An Approach to Enabling RDF Data in Querying to Invoke REST API for Complex Calculating

Presenter: Xianming Zhang, forzxm@163.com
Aviation Industry Development Center of China
Content

• Part 1 Motivation from a use case
• Part 2 Introduction to SPINx, its capability of executing JavaScript and defect
• Part 3 Driving SPINx framework to invoke REST API
• Part 4 Conclusion and discussion
RDF as well as related technologies are short in calculating, especially complex calculating. Although SPARQL, Jena rule, SWRL, being adopted widely, provide a number of in-built functions to implement calculating capacity, they can not accomplish many calculating tasks, reasons are:

1) In-built functions fail to perform many complex calculations, such as matrix, linear model-based ones, and so on.

\[ f(x) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \]
Motivation from a use case (2)

- It is not wise to write too many calculating steps in both query and rule statements.

```sql
SELECT ?value
WHERE {
  ?S ?P ?V
  BIND(f1(f2(f3(?V)))) AS ?value
}
```
Motivation from a use case (3)

3) Many calculations require external data which are not appropriate to be accessed freely, such as either valuable or of large volume experimental data. In the figure, the rdf source is external one that may not be accessed freely.
Motivation from a use case (4)

Here calculating for lift coefficient of airfoil is taken as a use case. An airfoil is designed to provide lift for airplane during flight, so it is necessary for users to query lift coefficient provided by a particular airfoil under given attack angle.

The formula is $\text{lift-coefficient} = \text{function(attack-angle)}$
In general, by means of searching RDF dataset in traditional way, we just get the result just like this figure. It is clear that we fail to get an explicit value under a given attack angle. The reason is that there is no explicit formula (or calculation script) to accurately calculate lift coefficient from attack angle.
Introducing SPINx, its capability of executing JavaScript and defect (1)

- SPIN (SPARQL Inferencing Notation) queries are stored in RDF format and together with RDF data, its idea is to link ontology classes in RDF with SPARQL queries that define constraints and rules formalizing the expected behavior of class members (instances).

**SPIN is an RDF Syntax for SPARQL**

SPIN provides a vocabulary to represent SPARQL queries as RDF triples.

```sparql
# Width and height must be equal
ASK WHERE {
    ?this ss:width ?width .
    ?this ss:height ?height .
    FILTER (?width != ?height) .
}
```

**Benefits**

- Stores SPARQL queries together with model
- Easy to share on the semantic web
- Referential integrity (true resource references)
- Namespaces are managed once, not for every query
Introduction to SPINx, its capability of executing JavaScript and defect (2)

- The SPINx framework included in SPIN makes it possible to define new SPARQL functions that are backed by JavaScript code. Whenever such new functions are invoked, a SPINx-aware SPARQL engine can look up the function’s body and execute it using a JavaScript interpreter.
2 Introduction to SPINx, its capability of executing JavaScript and defect (3)

- A user submits a SPARQL statement into SPINx framework, the statement's goal is to calculate lift coefficient with airfoil: CacuLiftCoefficient being the user-defined function (an instance of spin: Function) and 9 being the input attack angle.

<table>
<thead>
<tr>
<th>Segment of RDF Data</th>
<th>SPARQL Statement</th>
</tr>
</thead>
</table>
Introduction to SPINx, its capability of executing JavaScript and defect (4)

<table>
<thead>
<tr>
<th>Segment of RDF Data</th>
<th>SPARQL Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airfoil:CacuLiftCoefficient</td>
<td>SELECT ?VALUE</td>
</tr>
<tr>
<td>a spin:Function;</td>
<td>WHERE {</td>
</tr>
<tr>
<td>rdfs:subClassOf spin:Function;</td>
<td>BIND(Airfoil:CacuLiftCoefficient(9) AS ?VALUE).</td>
</tr>
<tr>
<td>spin:constraint</td>
<td>}</td>
</tr>
<tr>
<td>[ rdf:type spl:Argument;</td>
<td></td>
</tr>
<tr>
<td>rdfs:comment &quot;angle attack&quot;;</td>
<td></td>
</tr>
<tr>
<td>spl:predicate sp:arg1;</td>
<td></td>
</tr>
<tr>
<td>spl:valueType rdfs:Literal</td>
<td></td>
</tr>
<tr>
<td>] ;</td>
<td></td>
</tr>
<tr>
<td>spin:returnType xsd:float;</td>
<td></td>
</tr>
<tr>
<td>spinx:JavaScriptFile &quot;<a href="http://web-server/js/calculation.js">http://web-server/js/calculation.js</a>&quot;</td>
<td></td>
</tr>
</tbody>
</table>
2 Introduction to SPINx, its capability of executing JavaScript and defect (5)

- The figure illustrates the working principle of SPINx framework.
2 Introduction to SPINx, its capability of executing JavaScript and defect (6)

```javascript
function CacuLiftCoefficient(arg) {
    // for the sake of brevity, the code is omitted.
    return result;
}

return CacuLiftCoefficient(9);
```
2 Introduction to SPINx, its capability of executing JavaScript and defect (7)

- The defect is that the JavaScript file for calculation has to be deployed as a web URL accessible to other computers. In order to develop and deploy such JavaScript file, experimental data of confidential will have to be included in this exposed file and be updated frequently for future experiments, which both violates law of information security and is too expensive to maintain the JavaScript file.
Driving SPINx framework to invoke REST API(1)

• we can turn to an innovative method that is to develop a REST API that returns a small piece of JavaScript code containing resultant value after running. SPINx framework receives and executes this section of code quickly, namely extraction of resultant value in the code.

• REST API is typically a defined set of HTTP request messages along with a definition of the structure of response messages for system-to-system interactions (information exchange programmatically ), it is popular with more and more web applications that have deployed their own REST APIs.
Driving SPINx framework to invoke REST API (2)

- Users can access and invoke REST API by means of http://web-server/REST/CacuLiftCoefficient/{AttackAngle}, where {AttackAngle} can be replaced with any real number, such as 9, 9.6 and so on. The REST API for calculating of lift coefficient, developed in Spring Boot, can be written as below.

```java
@RestController
@RequestMapping(path="/REST")

public class Calculator {
    @RequestMapping(Path="/CacuLiftCoefficient/{AttackAngle}"，produce=MediaType.TEXT_HTML_VALUE)
    @ResponseBody
    public String CacuLiftCoefficient(@PathVariable("AttackAngle") float arg1) {
        String result;
        float value

        // for sake of brevity, details are omitted
        result="function CacuLiftCoefficient() {return "+(String)value+";}";
        return result;
    }
}
```
Driving SPINx framework to invoke REST API

```
http://web-server/REST/CacuLiftCoefficient/9

@RestController
@RequestMapping(path="/REST")
public class Calculator {
    @RequestMapping(Path="/CacuLiftCoefficient/{AttackAngle}" ,produce=MediaType.TEXT_HTML_VALUE)
    @ResponseBody
    public String CacuLiftCoefficient(@PathVariable("AttackAngle") float arg1) {
```
Driving SPINx framework to invoke REST API(3)

- After accessing to the REST API by URL, client can obtain a piece of code as HTTP response. For example, with http://web-server/REST/CacuLiftCoefficient/9 what client can obtain as follow:

  function CacuLiftCoefficient (){return 1.34 ;}
Driving SPINx framework to invoke REST API

1. Submitting SPARQL
2. SPINx framework loading RDF data
3. SPINx framework sending HTTP request
4. Receiving HTTP request
5. Invoking Web API
6. Responding with returned JavaScript code as HTTP response
7. SPINx frameworks executing returned content
Driving SPINx framework to invoke REST API(5)

``` ReasonML
Airfoil:CreateURL
    a spin:Function ;
    rdfs:label "create REST API URL"^^xsd:string ;
    rdfs:subClassOf spin:Functions ;
    spin:constraint
        [ rdf:type spl:Argument ;
            rdfs:comment "angle attack";
            spl:predicate sp:arg1 ;
            spl:valueType rdfs:Literal
        ] ;
    spin:returnType xsd:string;
    spinx:javaScriptCode "return 'http://web-server/REST/CacuLiftCoefficient/' + arg1"
```

```
DELETE
{ Airfoil:CacuLiftCoefficient spinx:javaScriptFile ?OLDURL 
}
INSERT
{ Airfoil:CacuLiftCoefficient spinx:javaScriptFile ?NEWURL 
}
WHERE{
    Airfoil:CacuLiftCoefficient spinx:javaScriptFile ?OLDURL.
    BIND(Airfoil:CreateURL(9) AS ?NEWURL ) }
```
Driving SPINx framework to invoke REST API(5)

Now the user can query resultant value by submitting

```
SELECT ?value WHERE {BIND( Airfoil:CacuLiftCoefficient() AS ?value). }
```

HTTP/1.1 200 OK
Content-Type: application/x-javascript
function CacuLiftCoefficient(){return 1.34;}

<table>
<thead>
<tr>
<th>Attack Angle</th>
<th>Lift Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td>5</td>
<td>1.1</td>
</tr>
<tr>
<td>10</td>
<td>1.4</td>
</tr>
<tr>
<td>15</td>
<td>1.7</td>
</tr>
<tr>
<td>20</td>
<td>1.6</td>
</tr>
<tr>
<td>22</td>
<td>0.6</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
</tr>
</tbody>
</table>

Angle attack=9
Coefficient=1.34
Conclusion and discussion

• With abundance in IT infrastructure today, the number of REST API is growing and RDF-based knowledge system should be constructed by fully taking advantage of this situation including REST API rather than from scratch. This paper discusses that usefulness and feasibility of using SPINx framework to invoke REST API for resultant value.

• The coordinating asynchronous requests, latency, availability and security must be taken into account, these problems should be solved effectively (at least in part) as the technologies for REST API, exemplified by SPRING BOOT, has made much effort to them from birth.
Thank you