SA-REST: Using Semantics to Empower RESTful Services and Smashups with Better Interoperability and Mediation

Karthik Gomadam  
*Wright State University - Main Campus*

Amit P. Sheth  
*Wright State University - Main Campus*, amit.sheth@wright.edu

Follow this and additional works at: [http://corescholar.libraries.wright.edu/knoesis](http://corescholar.libraries.wright.edu/knoesis)

Part of the Bioinformatics Commons, Communication Technology and New Media Commons, Databases and Information Systems Commons, OS and Networks Commons, and the Science and Technology Studies Commons

Repository Citation  
[http://corescholar.libraries.wright.edu/knoesis/758](http://corescholar.libraries.wright.edu/knoesis/758)
Software is a Service

Key Features:
- Customizable
- Ubiquitous
- Component Driven
- Device Independent
- Data as “Intel Inside”
Services Act 1: Starring…

- WSDL
- WS-Metadata Exchange
- SOAP
- BPEL
- WS-Policy
- WS-Transactions
- WS-Notification
- WS-Addressing
- WS-Reliable Messaging
- WS-Resource Lifetime
- WS-Security
And Then …

I'VE SPENT A BUNCH OF MILLIONS OF DOLLARS FOR THE SOA YOU TOLD ME IT'S A MUST TO HAVE. AM I FINISHED NOW?

ALMOST. YOU JUST HAVE TO THROW ALL THE HEAVY WEIGHT BLOATED SOAP-WSDL-CRAP OUT. SOA WAS YESTERDAY. TODAY IS REST.

OH MY DEAR. THAT COSTS ME ANOTHER 2 YEARS AND A COUPLE OF MILLION BUCKS.

DO YOU WANNA BE AGILE OR DO YOU NOT WANNA BE AGILE?

Services Act 2: Starring

<table>
<thead>
<tr>
<th>HTTP</th>
<th>XML</th>
<th>RSS / ATOM</th>
<th>JSON</th>
</tr>
</thead>
</table>

and
Services on the Web =

• Democracy
• Innovation
• But as with any democratic process
AGREEMENT CAN BE VERY HARD...VERY VERY HARD
SA-REST:
Using Semantics to Empower RESTful Services and Smashups with Better Interoperability and Mediation

Karthik Gomadam, Researcher
Dr. Amit P. Sheth, Lexis-Nexis Eminent Scholar
Services Research Lab,
Kno.e.sis Center, Wright State University, Dayton, OH.
Acknowledgement: Ajith Ranabahu kno.e.sis center
SA-REST is

Approach to

- Create services that are more interoperable
- Smart Mashups (Smashups)
- Demystify mashups
- Enable device independent applications

Not quite…
The Road Ahead…

• Foundations of SA-REST
  – MREF, SAWSDL

• Microformats

• SA-REST
  – SA-REST Microformat

• SA-REST Benefits
  – Data Mediation and Mediatability
  – SA-REST to Recipes
SA-REST : The roots

Mref
SAWSDL
MREF (Metadata REFerence links)

- Representing and Correlating information at a meta- or semantic level
- Abstraction on top of RDF and XML
- `href` for logical relationships.
- Virtual Resource
  - Can be embedded in HTML or linked
MRef: Continued...

```xml
<?namespace href="http://www.fooo.com/IQ" as="IQ"?>
<?namespace href="http://www.w3.org/schemas/rdf-schema" as="RDF"?>
<RDF:serialization>
  <RDF:bag id="MREF:12345">
    <IQ:keyword>
      <RDF:resource id="constraint_001">
        <IQ:threshold>0.5</IQ:threshold>
        <RDF:PropValue>winter rose</RDF:PropValue>
      </RDF:resource>
    </IQ:keyword>
    <IQ:attribute>
      <RDF:resource id="constraint_002">
        <IQ:name>color</IQ:name>
        <IQ:type>string</IQ:type>
        <RDF:PropValue>red</RDF:PropValue>
      </RDF:resource>
    </IQ:attribute>
    <IQ:attribute>
      <RDF:resource id="constraint_003">
        <IQ:name>fragrance</IQ:name>
        <IQ:type>string</IQ:type>
        <RDF:PropValue>slight</RDF:PropValue>
      </RDF:resource>
    </IQ:attribute>
  </RDF:bag>
</RDF:serialization>
```
SAWSDL: Semantic Annotations for WSDL and XML Schema

• Defined as WSDL-S [Sivashanmugam et. Al, Adding Semantics to Web Service Standards, ICWS, 2003]
• Evolutionary approach to add semantics to services
• WSDL + modelreference = SAWSDL!!!!
  – Little Semantics…Indeed goes a long way
SAWSDL: ModelReference

• Defines how to add semantic annotations to various parts of a WSDL document
  – Interface, Operations, Input and Output

• XML Schema
  – Element Declarations
  – Attribute Declarations
Semantics:

• ontology classes
  – discovery, composition
  – filtering, ranking
• lifting/lowering mappings
  – mediation, invocation
• functionality categories
  – publishing, discovery, composition
• anything, really

Image Courtesy:

SAWSDL

• Grounded to semantic meta-models
  – Independent of ontology / meta-model specification languages

• Lifting and Lowering
  – Systematic approach to data mediation
  – Mediation at the schema level
  – XSLT driven
The Road Ahead…

• Foundations of SA-REST
  – MREF, SAWSDL

• Microformats

• SA-REST
  – SA-REST Microformat

• SA-REST Benefits
  – Data Mediation and Mediatability
  – SA-REST to Recipes
Supporting SA-REST: Microformats
Microformats

- Designed for humans first and machines second
- Simple open formats built upon existing standards 😊
- Easier to add markups to via POSH (Plain Old Semantic HTML)
Design Patterns in Microformats

• **abbr-design-pattern**
  – Human friendly text along with machine processible text

• **Class-design-pattern**
  – Indicate Semantic meaning

• **rel-design-pattern**
  – Indicate meaning of a link

• **Others..**
  – But we concern ourselves with only those rel-to-SA-REST
SA-REST

Microformats

MREF
SAWSDL
Services Act 3: SA-REST

• Microformat approach
  – Add more meaning to service descriptions
  – What messaging formats, What methods…
  – Semantic grounding to concepts
    • Domain of an API
    • Annotated inputs and outputs
Breaking Down SA-REST

• **input**
  – Block markup
  – Markups within this block relate to the input
  – Pattern: Class

• **output**
  – Block markup
  – Markups within this block relate to output
  – Pattern: Class
Breaking Down SA-REST

• **domain-rel**
  – The domain(s) of the API
  – Can be used at the API level
    • markup on the body element
  – block level
    • the domain of a given block
  – Pattern: abbr

• **method**
  – Captures Get or Post; Method for accessing a resource
  – Pattern: Class
Breaking Down SA-REST

• **p-lang-binding**
  – Programming language binding
  • Useful describing the languages supported by an API
  • Pattern: Class

• **sem-rel**
  – Describes a link in an API
  – An XSD schema link
  – Pattern: Abbr
Breaking Down SA-REST

• **sem-class**
  – Meta description for content in the API
  – Ala SAWSDL’s modelreference
  – Pattern: Abbr

• **data-format**
  – Data format descriptors (XML, RSS / ATOM, Gdata,…)
  – Pattern: Class

• **Protocol**
  – SOAP / REST
  – Pattern: Class
SA-REST: The Vehicles

- RDFa
  - SA-REST elements can be used along with RDFa

```xml
<div xmlns:sarest=http://knoesis.wright.edu/srl/sarest
     xmlns:apihutTax=http://apihut.com/facetedTaxonomy>
  <div about="sarest:input">
    The input is an <span property="sarest:sem-class" value="apihutTax:address">address</span>. The schema is described in <a href="http://foo.xsd" rel="sarest:sem-rel" value="apihutTax:address">Address.xsd</a>.
  </div>
</div>

<div xmlns:sarest=http://knoesis.wright.edu/srl/sarest
     xmlns:apihutTax=http://apihut.com/facetedTaxonomy>
Using <span property="sarest:p-lang-binding">PHP</span>/MySQL with Google <span property="sarest:domain" value="apihutTax:Mapping">Maps</span>
</div>
```
SA-REST: Vehicles

• GRDDL
  – Use the SA-REST microformat as it is
  – Extract the RDF using GRDDL
  – Make sure the resource is “gleanable”
  – XSLT your way to RDF
Rules of Thumb

• The text *unambiguously* allows the system to identify the concept in the meta-model
  – Use `<class>` in microformat version
  – No value in RDFa

• All other cases
  – Use `<attr>` in microformat
  – Value in RDFa
Example

```html
<div class="input">
  <abbr class="sem-class" title="http://apihut.com/facetedTaxonomy#Address">
    Address
  </abbr>
</div>
```
Faceted Search for APIs

Data mediation

SA-REST

Smarter Mashups

Microformats

MREF SAWSDL
SA-REST: Benefits

• Data Mediation
  – Systematic mediation similar to SAWSDL
  – Upcast and Downcast
  – Specify “Application Data Model”
    • Services map to the ADM
  – How much effort will it take me to mediate?
    • Mediatability computation
SA-REST: Benefits

• Smarter Mashups
  – More dynamism for mashups
  – Why only craigslist and Google maps?
    • One glove never fits all
  – Meta level specification of Mashups
    • Specify application at meta-level and go from there
    • Demo
SA-REST: Benefits

• Demystifying mashups
  – Better searching for API’s in a more faceted manner
  – Better API integration
    • Better mediation
    • Code generation

• PROGRAMMABLE WEB FOR THE MASSES?
SA-REST: Heads Up

• Taxonomies available for APIs
  – Programmableweb.com (more user created)
    • 55 categories
  – ApiHut.com/taxonomy (User assisted)
    • 60 Categories
    • 4 different facets
      – Functional, Message Format, Protocol and Programming language bindings
      – Available in RDFS
SA-REST: A Walkthrough

• The example will be made available at
  – http://knoesis.wright.edu/research/srl/standards/sa-rest
Mashup using SA-REST

• The example will be made available at
  – http://knoesis.wright.edu/research/srl/standards/sa-rest
ApiHut.com : Find and Bind

• ApiHut.com is a framework for performing faceted API search
• ApiHut uses SA-REST internally for classification
• Plans to support assisted user annotation
• Public alpha expected soon.
  – http://apihut.com
What Next?

• SWS Testbed Incubator activity (ongoing):
  • To be followed by submission to W3C
  • [http://www.w3.org/2005/Incubator/swsc/](http://www.w3.org/2005/Incubator/swsc/)
  • Collaborators and contributors welcome
Contribute

• Blogs
• Use cases
• Open source implementations
• Monitor the progress
  – http://knoesis.wright.edu/research/srl/standards/sarest
  – http://www.w3.org/2005/Incubator/swsc/