RSLT
RDF Stylesheet Language Transformations
Silvio Peroni & Fabio Vitali
University of Bologna
Summary

• RSLT is a language for creating visualisations (in HTML) of RDF graphs using transformational templates
• Inspired by XSLT for XML documents, with a few adaptations to the specificities of RDF graphs
• An implementation has been created within the browser and can be tested at the address
  
  http://www.fabiovitali.it/rslt/
  http://github.org/fvitali/rslt/
An example
Tripelstore: http://two.eelst.cs.unibo.it:8181/data/query

```xml
<rslt
  triplestore="http://two.eelst.cs.unibo.it:8181/data/query">
  <prefix ns="">
    rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns# ...
  </prefix>
  <template match="?person -> foaf:Person">
    <p>Found
      {{ count(person.pro_holdsRoleInTime) }}
      papers authored by
      {{ person.foaf_givenName ?g }}
      {{ person.foaf_familyName ?f }}:
    </p>
    <ul>
      <applyTemplates select='?person pro:holdsRoleInTime / pro:relatesToDocument ??work'></applyTemplates>
    </ul>
  </template>
...
</rslt>
It is possible to get the value specified in a data property statement by using the element “valueOf”

An alternative syntax for valueOf exists, based on AngularJS syntax for bindable markup

```html
<template match="?person -> foaf:Person">
  <p>Found
    {{ count(person.pro_holdsRoleInTime) }}
    papers authored by
    {{ person.foaf_givenName }}
    {{ person.foaf_familyName }}:
    ...
</p>
```

Variables are objects with dot notation connecting them to properties. Colon separating prefixes from property names is substituted with underscore because the colon is restricted in Javascript.
<applyTemplates>

• The basic recursion aid of the RSLT language

<template match="?person -> foaf:Person">
  <p>Found
  {{ count(person.pro_holdsRoleInTime) }}
  papers authored by
  {{ person.foaf_givenName ?g }}
  {{ person.foaf_familyName ?f }}:
</p>
  <ul>
    <applyTemplates select='?person pro:holdsRoleInTime / pro:relatesToDocument ??work'></applyTemplates>
  </ul>
</template>

Starting from the person into consideration, select all the works that involve her with some role (e.g., as author) and apply the appropriate template
To distinguish triples from RSLT entities, we introduce an extension to the simple sparql syntax, which we call DQM (double question mark):

```sparql
??person foaf:familyName 'Horrocks'.
```

selects the RSLT entity whose familyName is Horrocks, and not simply that specific triple. A RSLT entity is defined as all the triples with the same subject. Arbitrarily complex graphs can be described through this syntax, and only those variable specified through double question marks will be considered for returning RSLT entities.
RSLT handles three types of selectors in templates

- Individual RDF statements
  - `<template match='?person foaf:familyName "Horrocks"'>`
  - `<template match='?person foaf:familyName ?string'>`

- Resources identified by their URI
  - `<template match='http://www.semanticlancet.eu/resource/person/ian-horrocks'> ... </template>`

- Resources identified by their type
  - `<template match='?person -> foaf:Person'>`
    ... </template>
  - The -> syntax selects all triples whose subject is a resource of the specified type.
Future work

• Documentation
• Implement a full mechanism to avoid infinite loops
• Implement support for RDF files using the RDF-Ext library
• Support SPARQL endpoints based on SPARQL 1.0 (many Linked Data endpoints are still 1.0)
  – Current implementation relies heavily on BIND
Open questions

• What is the best way to deal with SPARQL 1.0 endpoints? Ignore them? Adopt them as lesser sons? Create queries that work on both 1.0 and 1.1?

• Knowledge is best understood when expressed in natural language texts (a.k.a documents). Why is it so difficult to convert RDF graphs in texts?