Linking Data from RESTful services

Rosa Alarcon
Computer Science Department
Pontificia Universidad Catolica de Chile

Erik Wilde
School of Information
UC Berkeley

LDOW 2010

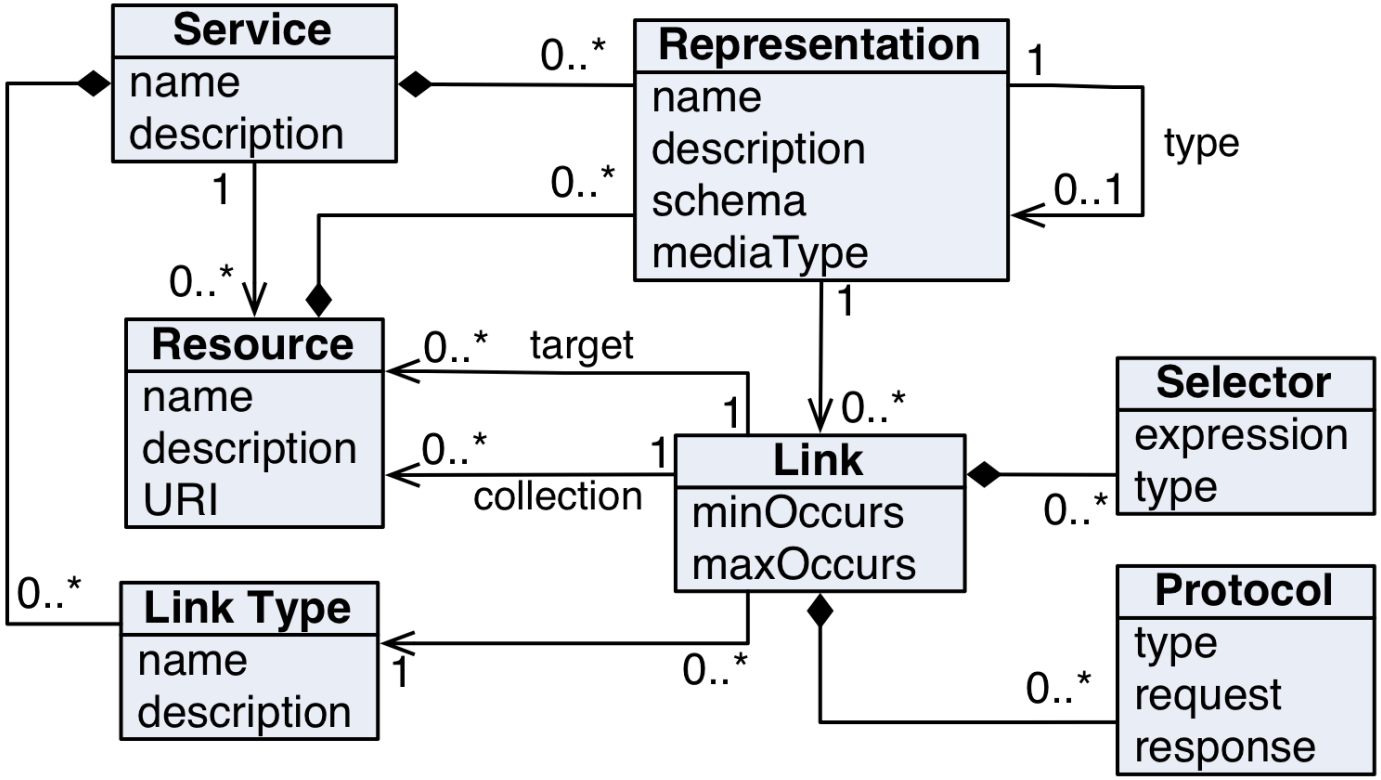
Abstract

- Semantic Web goal: extend current human-readable Web resources with semantic information encoded in machine-processable form
- Web of Data: Linked Data principles, several data sources compliant with the Semantic Web technologies, such as, RDF triple stores, and SPARQL endpoints
- REST: a set of the architectural principles that underlie the human-readable Web
- To provide a mechanism for describing REST (i.e. human-readable Web) resources and transform them into semantic resources
- Lower entry barrier

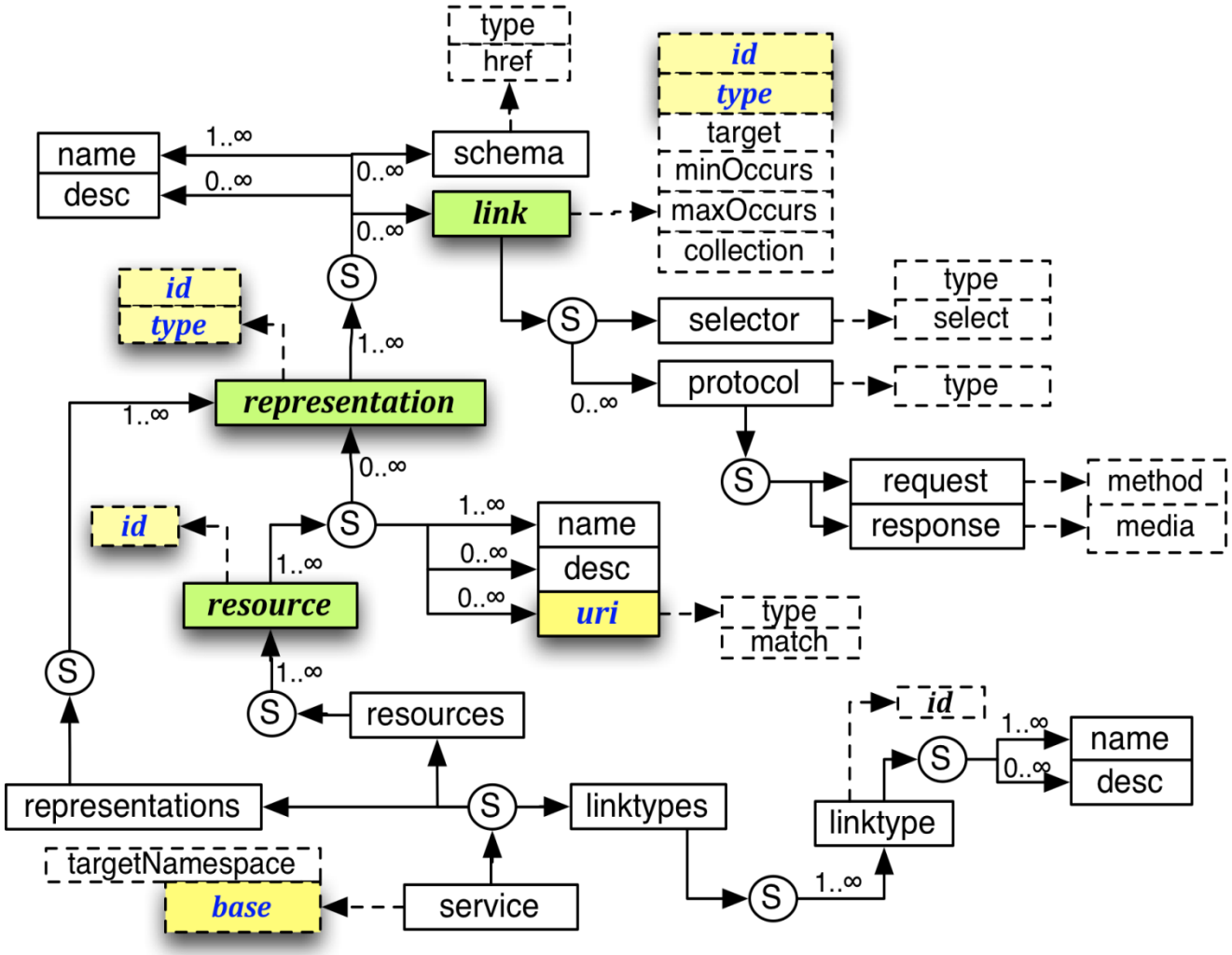
REST (Representational State Transfer)

- *Resources have unique, opaque identifiers*
  - Avoid coupling between clients and servers
- Uniform interface: methods with know semantics that change the state of resources
  - HTTP: GET, PUT, DELETE, POST, OPTIONS
- Resources (conceptual) have multiple representations
  - JSON, XML, XHTML, etc.
- Hyperlinks
  - Related resources + State change

ReLL: Resource Linking Language



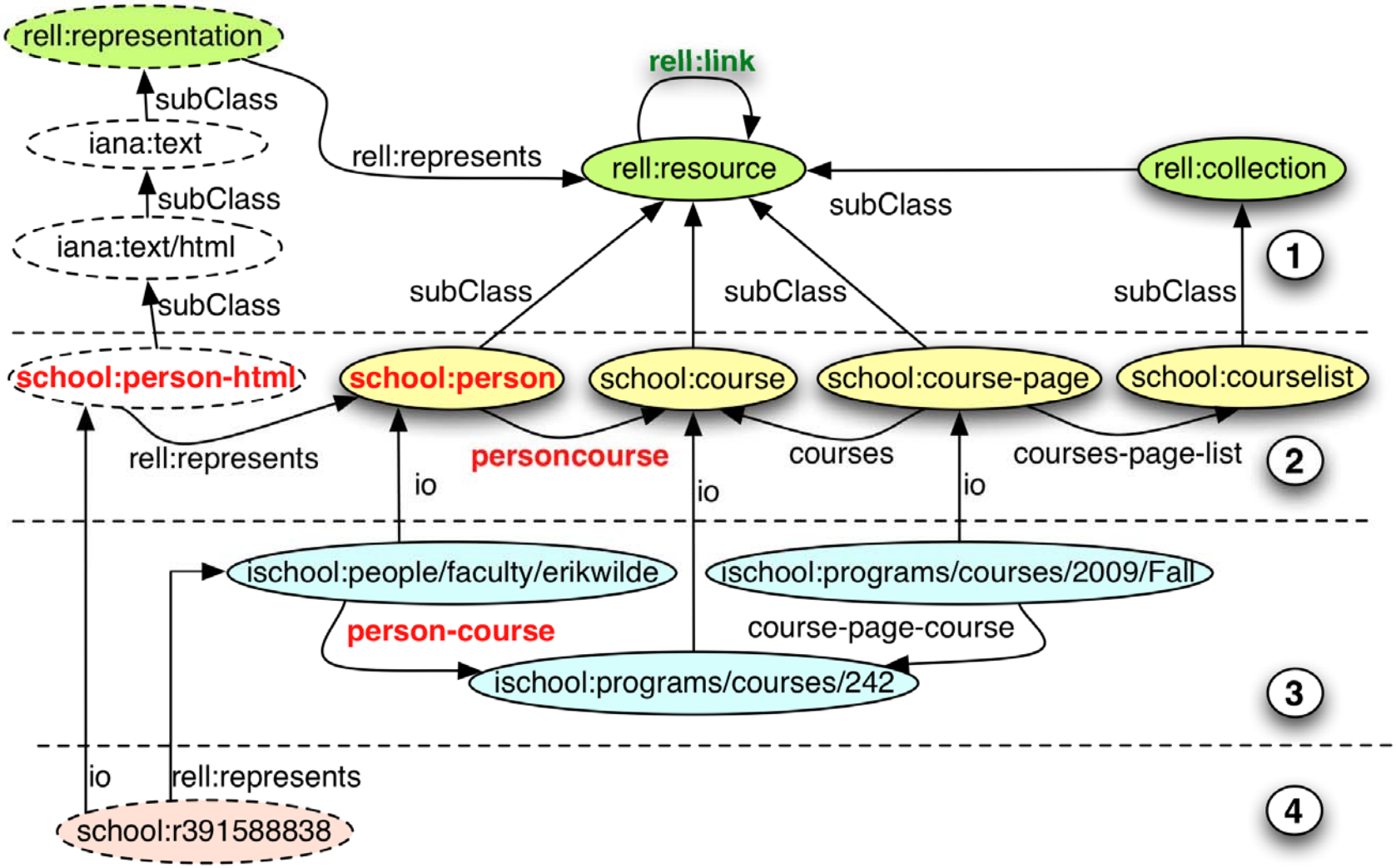
ReLL Description Schema



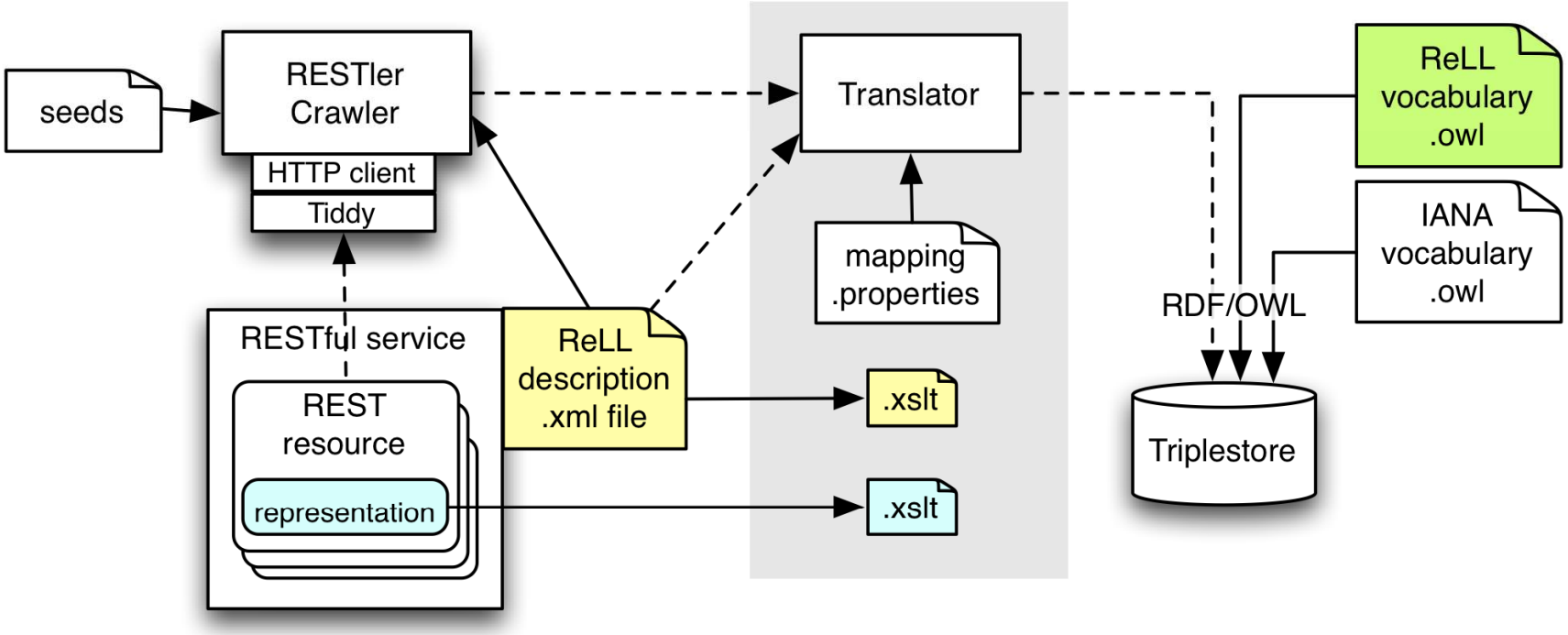
ReLL Snippet: Describing the ISchool


```
<service  … targetNamespace = "http://rell.org/school/”  … >
    <resource xml:id = "person">
        <name> …
        <desc> …
        <uri match="http://.*?/people/(faculty|students|staff|visitors)/[a-zA-Z]+" type="regex"/>
        <representation xml:id="person-html" type="iana:text/html">
            <name> …
            <link xml:id="person-website" type="website">
                <selector select="//div[@class = 'field-field-person-website']//a/@href"
    type="xpath"/>
            </link>
            <link xml:id="person-course" type="personcourse" target="course">
                <selector select="//span[@class = 'views-field-title']/a/@href" type="xpath"/>
                <protocol type="http">
                    <request method="get"/>
                    <response media="iana:html"/>
                </protocol>
            </link>
        </representation>
    </resource>
```

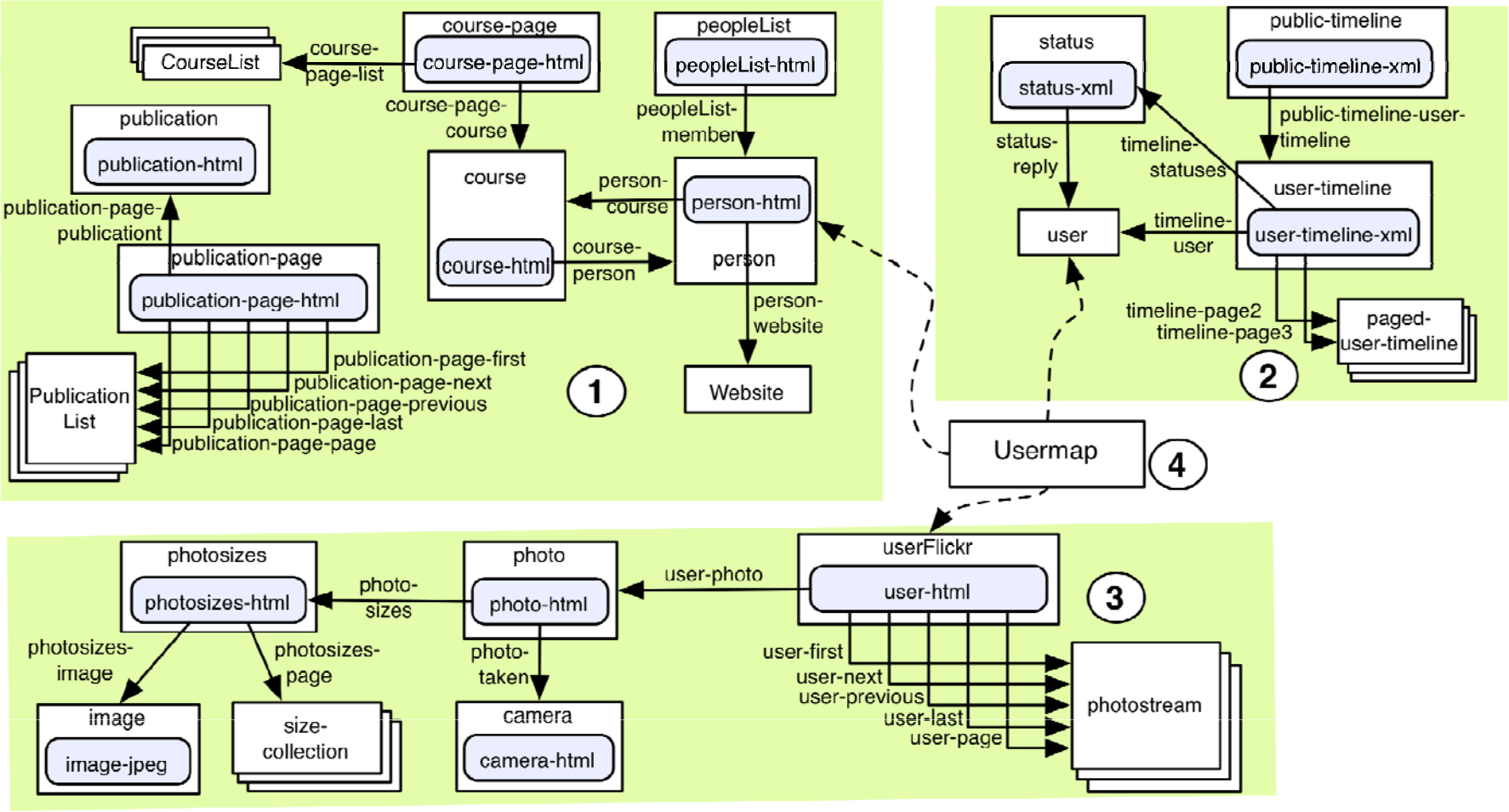
From ReLL to RDF



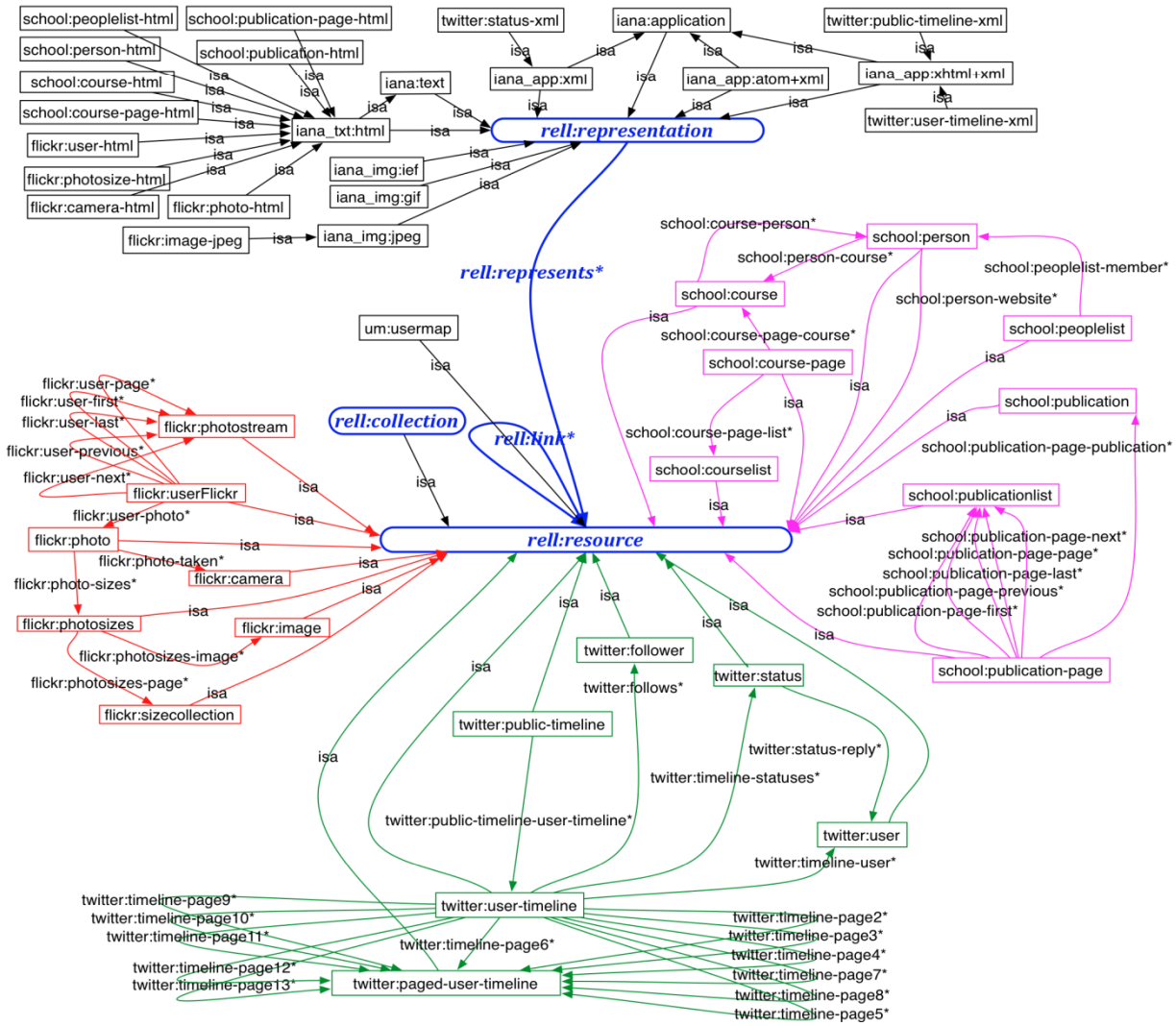
Implementation: RESTler



Crawled REST Services



Crawled REST Resources



Generated RDF

```
@prefix rell: <http://rell.org/rell/> .

rell:representation a rdfs:Class .
rell:resource a rdfs:Class .
rell:collection rdfs:subClassOf rell:resource .

rell:link a owl:ObjectProperty ;
   rdfs:domain rell:resource ;
   rdfs:range rell:resource .

rell:represents a owl:ObjectProperty ;
   rdfs:domain rell:representation ;
   rdfs:range rell:resource .
```

(a)

```
@prefix iana: <http://www.iana.org/assignments/media-types/> .

<iana:text/html> rdfs:subClassOf rell:representation , <iana:text> ;
   rdfs:label "The \"text/html\" Media Type" .
<iana:image/jpeg> rdfs:subClassOf rell:representation , <iana:image> ;
   rdfs:label "The \"image/jpeg\" Media Type" .
<iana:application/atom+xml> rdfs:subClassOf rell:representation ,
   <iana:application> ;
   rdfs:label "The \"application/atom+xml\" Media Type" .
<iana:application/xml> rdfs:subClassOf rell:representation ,
   <iana:application> ;
   rdfs:label "The \"application/xml\" Media Type" .
```

(b)

```
@prefix school: <http://rell.org/school/> .
@prefix iana: <http://www.iana.org/assignments/media-types/> .
school:person a rdfs:Class ;
   rdfs:subClassOf rell:resource ;
school:course a rdfs:Class ;
   rdfs:subClassOf rell:resource ;
school:course-page a rdfs:Class ;
   rdfs:subClassOf rell:resource ;
school:courselist a rdfs:Class ;
   rdfs:subClassOf rell:collection ;
school:person-html a rdfs:Class ;
   rdfs:subClassOf <iana:text/html> ;
   rell:represents school:person .
```

(c)

```
school:course-page-list a owl:ObjectProperty ;
   rdfs:domain school:course-page ;
   rdfs:range school:courselist ;
   rdfs:subPropertyOf school:page .

school:personcourse a owl:ObjectProperty ;
   rdfs:subPropertyOf rell:link ;

school:courses a owl:ObjectProperty ;
   rdfs:subPropertyOf rell:link ;
```

(d)

```
school:person-course a owl:ObjectProperty ;
   rdfs:domain school:person ;
   rdfs:range school:course ;
   rdfs:subPropertyOf school:personcourse .

school:course-page-course a owl:ObjectProperty ;
   rdfs:domain school:course-page ;
   rdfs:range school:course ;
   rdfs:subPropertyOf school:courses .
```

(e)

```
<http://www.ischool.berkeley.edu/people/faculty/erikwilde> a school:person ;
   school:person-website <http://dret.net/netdret/> ;
   school:person-course <http://www.ischool.berkeley.edu/programs/courses/242> .

<http://www.ischool.berkeley.edu/programs/courses/242> a school:course ;
   school:course-person <http://www.ischool.berkeley.edu/people/faculty/erikwilde> .
```

(f)

```
school:r391588838 a school:person-html ;
   rell:represents <http://www.ischool.berkeley.edu/people/faculty/erikwilde> .
```

(g)

Getting RDF from Resources


```
<http://www.ischool.berkeley.edu/people/faculty/erikwilde> a school:person ;
   vCard:FN "Erik Wilde" ;
   vCard:ADR _:node14m5kienpx1603 ;
   vCard:TITLE "Adjunct Professor" ;
   vCard:ORG _:node14m5kienpx1604 ;
   vCard:EMAIL _:node14m5kienpx1606 ;
   vCard:TEL _:node14m5kienpx1607 ;
   vCard:URL <http://dret.net/netdret/> ;
   vCard:PHOTO <http://www.ischool.berkeley.edu/files/imagecache/profile-pic/DSC_0176.JPG> ;
   school:person-website <http://dret.net/netdret/> ;
   school:person-course <http://www.ischool.berkeley.edu/programs/courses/242> ,
     <http://www.ischool.berkeley.edu/programs/courses/152> ,
     <http://www.ischool.berkeley.edu/programs/courses/190-waim> ,
     <http://www.ischool.berkeley.edu/programs/courses/290-wa> .
```

ISchool Camera SPARQL


```
<http://www.ischool.berkeley.edu/people/faculty/erikwilde> a school:person ;
   owl:sameAs <http://www.flickr.com/photos/dret/> , <http://twitter.com/dret> ,
   <http://twitter.com/users/show/dret.xml> ;
```

(a)

```
PREFIX dcterms:<http://purl.org/dc/terms/>
PREFIX flickr:<http://rell.org/flickr/>

SELECT DISTINCT ?person ?camera
WHERE
{?person owl:sameAs ?flickruser .
 ?picture dcterms:creator ?flickruser .
 ?picture flickr:photo-taken ?camera}
```

(b)

```
<school:people/faculty/erikwilde> <flickr:cameras/apple/iphone_3g/>
<school:people/faculty/erikwilde> <flickr:cameras/panasonic/dmc-tz5/>
<school:people/faculty/erikwilde> <flickr:cameras/panasonic/dmc-tz1/>
<school:people/faculty/erikwilde> <flickr:cameras/nikon/d80/>
```

(c)

Conclusions

- Do RESTful services even should be described?
  - Descriptions introduce coupling between service provider and consumer.
  - Shared set of assumptions & preconditions that facilitates documentation, understanding, & change identification (e.g. new ids, access schemes or representation format), so that clients and serves can react appropriately (e.g. alerting the client manager, attempting a fallback, or abort).
  - Mostly useful for automatic agents that translate the contract into RDF triples.
- Hence, ReLL can achieve loose coupling and still allow clients to behave when the unexpected occurs.

Conclusions

- Limitations
  - Static description of RESTful services (new resources, and changes ignored)
  - Design a ReLL document for each REST service (e.g. Virtuoso's Sponger)
  - Design a specific XSLT for each resource type to harvest information
- On the bright side
  - Web technologies (e.g. XPath, XSLT and XML) familiar to Web developers
- Future work
  - Dynamic and automatic generation of ReLL descriptions; may require information retrieval, text mining and probably machine learning techniques
  - Complex REST models (e.g. methods such as PUT, DELETE and POST)
  - Include Linked Data vocabularies