



Extensions of SPARQL towards Heterogeneous Sources and Domain Annotations

Nuno Lopes

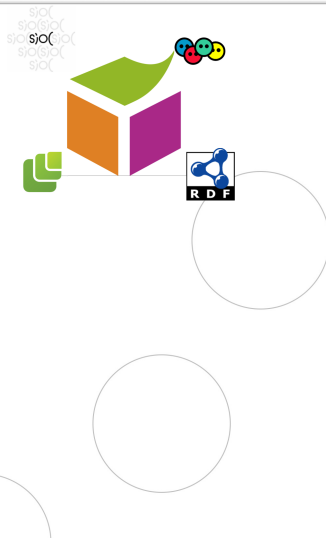
November 09, 2010



NUI Galway
OÉ Gaillimh



Scenario: Integration of Heterogeneous Sources



Scenario: Integration of Heterogeneous Sources

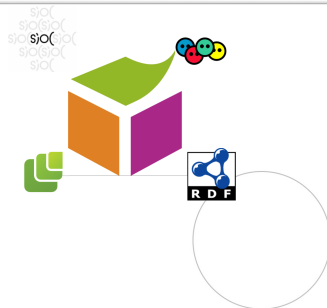


www.deri.ie



<XML/>

SOAP/WSDL

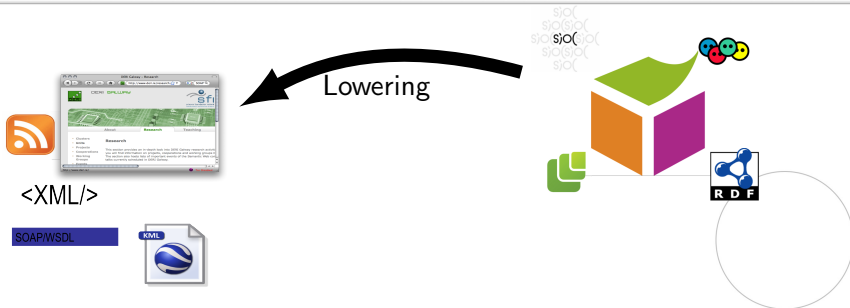


Scenario: Integration of Heterogeneous Sources

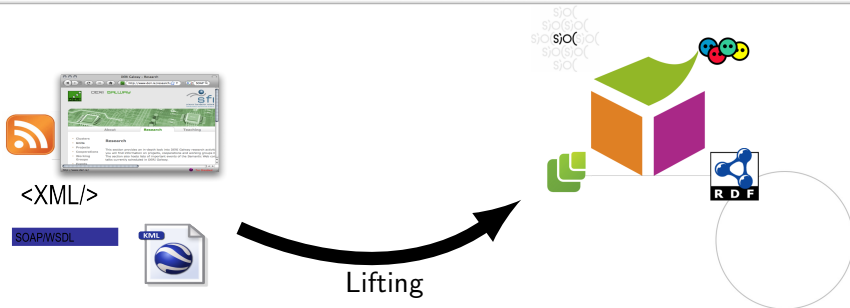


www.deri.ie

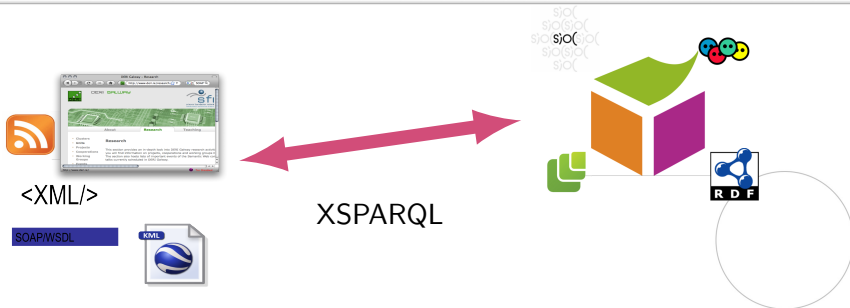
Digital Enterprise Research Institute



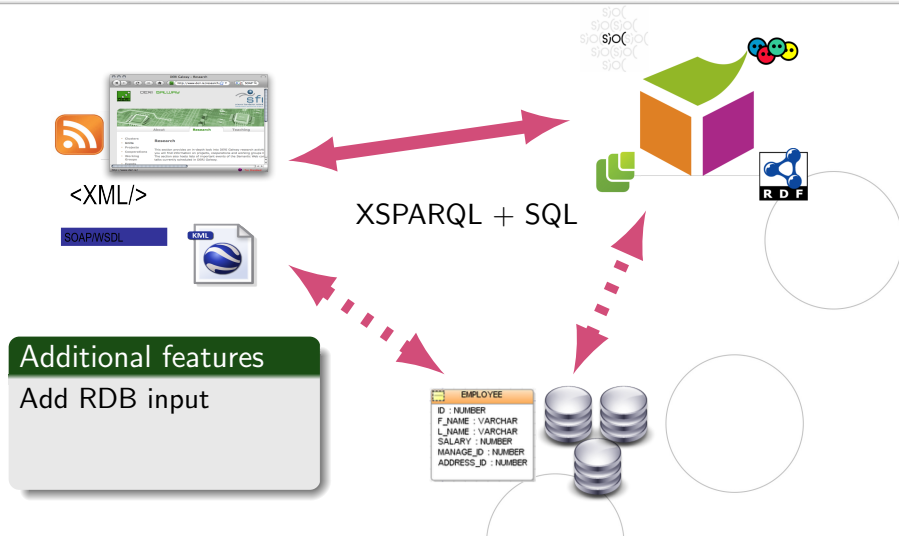
Scenario: Integration of Heterogeneous Sources



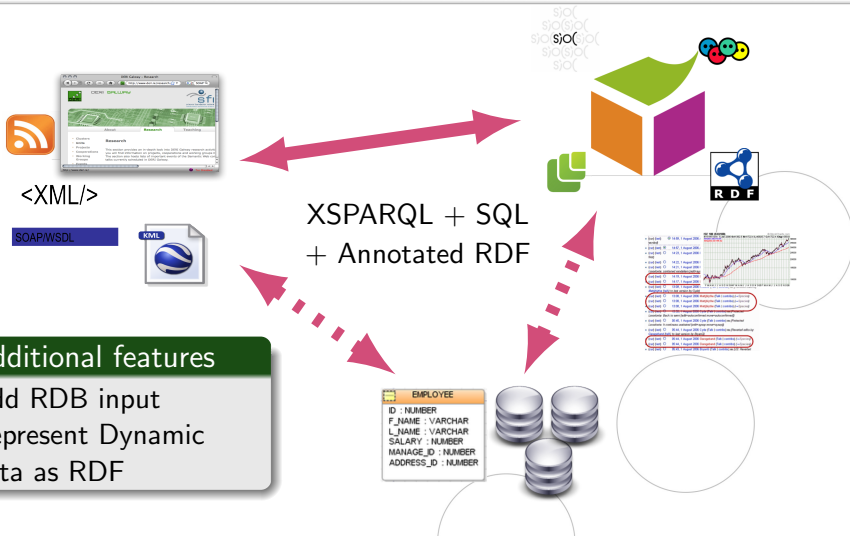
Scenario: Integration of Heterogeneous Sources



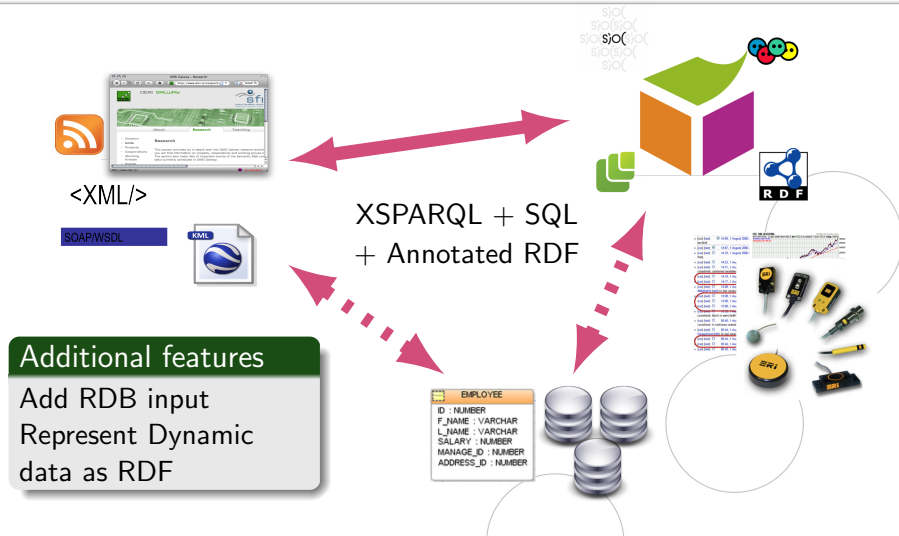
Scenario: Integration of Heterogeneous Sources



Scenario: Integration of Heterogeneous Sources



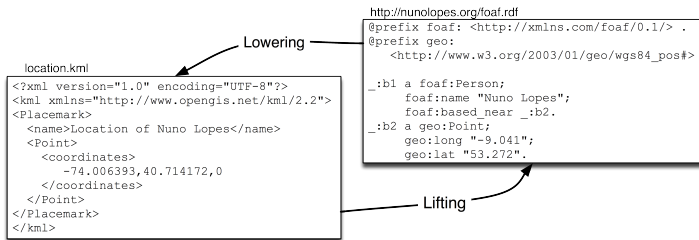
Scenario: Integration of Heterogeneous Sources



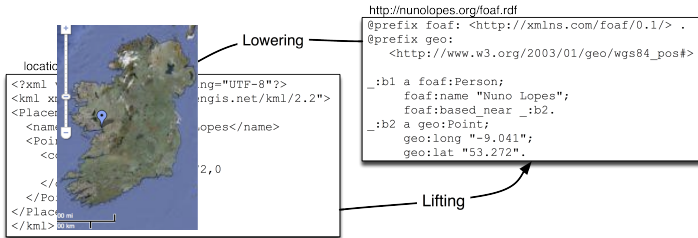
Additional features
Add RDB input
Represent Dynamic data as RDF

ID	F_NAME	L_NAME	SALARY	MANAGE_ID	ADDRESS_ID
100	DEHAENE	CHRISTOPHER	11000	100	100
101	DEHAENE	CHRISTOPHER	11000	100	100
102	DEHAENE	CHRISTOPHER	11000	100	100
103	DEHAENE	CHRISTOPHER	11000	100	100
104	DEHAENE	CHRISTOPHER	11000	100	100
105	DEHAENE	CHRISTOPHER	11000	100	100
106	DEHAENE	CHRISTOPHER	11000	100	100
107	DEHAENE	CHRISTOPHER	11000	100	100
108	DEHAENE	CHRISTOPHER	11000	100	100
109	DEHAENE	CHRISTOPHER	11000	100	100

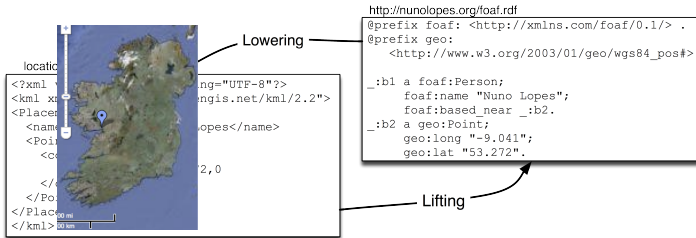
Motivation: Lifting and Lowering



Motivation: Lifting and Lowering



Motivation: Lifting and Lowering



Transformations between XML and RDF are not easy
Mainly due to the heterogeneity of RDF/XML serialisations
Objective: language capable of integrating heterogeneous sources for the Semantic Web

XQuery + SPARQL = XSPARQL



- Syntactic extension of XQuery
- With a formally defined semantics
- includes XML and RDF sources



- Syntactic extension of XQuery
- With a formally defined semantics
- includes XML and RDF sources

Query example (Lowering)

```
prefix foaf : <http://xmlns.com/foaf/0.1/>
prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
<kml xmlns="http://www.opengis.net/kml/2.2">
{ for $person $name $long $lat from <http://nunolopes.org/foaf.rdf>
  where { $person a foaf:Person; foaf:name $name;
          foaf:based_near [ a geo:Point; geo:long $long; geo:lat $lat ] }
return <Placemark>
  <name>{fn:concat("Location of ", $name)}</name>
  <Point><coordinates>{fn:concat($long, ",", $lat, ",0")}
  </coordinates></Point>
</Placemark>
}</kml>
```

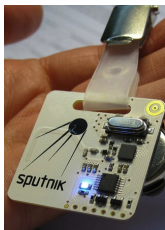


- Syntactic extension of XQuery
- With a formally defined semantics
- includes XML and RDF sources

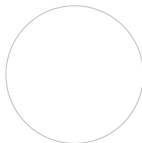
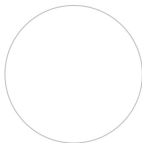
Query example (Lowering)

```
prefix foaf : <http://xmlns.com/foaf/0.1/>
prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
<kml xmlns="http://www.opengis.net/kml/2.2">
{ for $person $name $long $lat from <http://nunolopes.org/foaf.rdf>
  where { $person a foaf:Person; foaf:name $name;
          foaf:based_near [ a geo:Point; geo:long $long; geo:lat $lat ] }
return <Placemark>
  <name>{fn:concat("Location of ", $name)}</name>
  <Point><coordinates>{fn:concat($long, ",", $lat, ",0")}</coordinates></Point>
</Placemark>
}</kml>
```

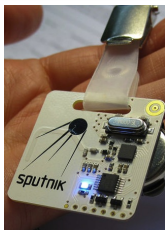
Usecase: Using XSPARQL to expose location data



sensor tags are assigned to people



Usecase: Using XSPARQL to expose location data

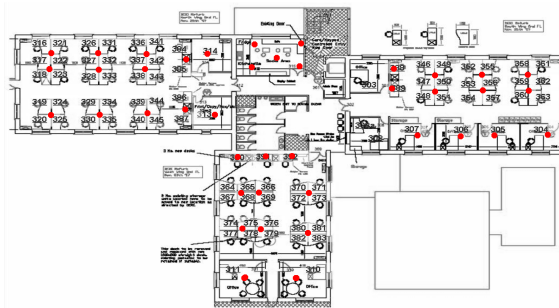


sensor tags are assigned to people

tag proximity is registered by *base stations*

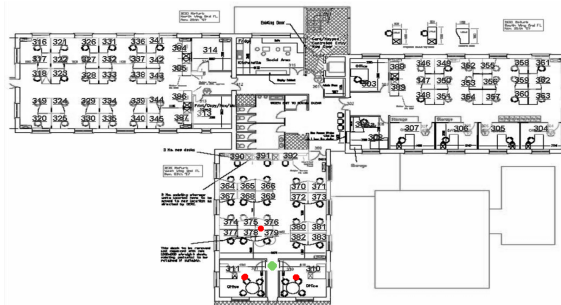


Usecase: Using XSPARQL to expose location data



base stations are deployed throughout a building

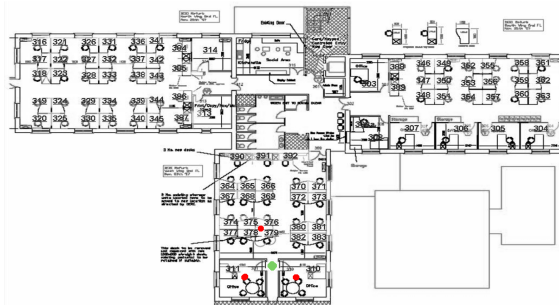
Usecase: Using XSPARQL to expose location data



sensor data readings:

timestamp	ip	tag	ssi
2010-11-09 14:57:51	10.254.2.15	4302	83
2010-11-09 14:57:51	10.254.3.1	4302	83
2010-11-09 14:57:51	10.254.2.6	4302	83

Usecase: Using XSPARQL to expose location data

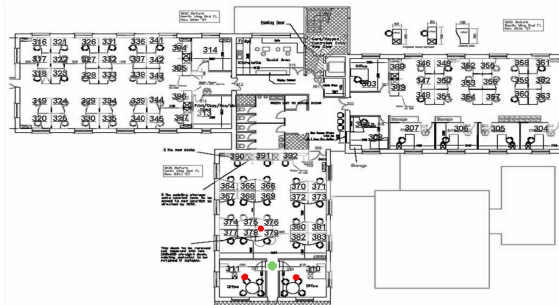


sensor data readings:

timestamp	ip	tag	ssi
2010-11-09 14:57:51	10.254.2.15	4302	83
2010-11-09 14:57:51	10.254.3.1	4302	83
2010-11-09 14:57:51	10.254.2.6	4302	83



Usecase: Using XSPARQL to expose location data

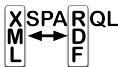
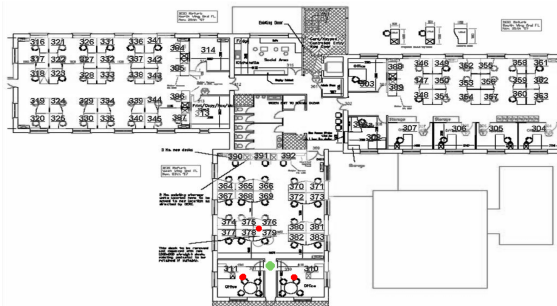


sensor data readings:

timestamp	ip	tag	ssi
2010-11-09 14:57:51	10.254.2.15	4302	83
2010-11-09 14:57:51	10.254.3.1	4302	83
2010-11-09 14:57:51	10.254.2.6	4302	83



Usecase: Using XSPARQL to expose location data



Use XSPARQL to create a webpage from the annotated data



EMPLOYEE	
ID	: NUMBER
F_NAME	: VARCHAR
L_NAME	: VARCHAR
SALARY	: NUMBER
MANAGE_ID	: NUMBER
ADDRESS_ID	: NUMBER



Next steps

- Defining the syntax for querying RDBs
- Extending XSPARQL's semantics

EMPLOYEE	
ID	: NUMBER
F_NAME	: VARCHAR
L_NAME	: VARCHAR
SALARY	: NUMBER
MANAGE_ID	: NUMBER
ADDRESS_ID	: NUMBER



Next steps

- Defining the syntax for querying RDBs
- Extending XSPARQL's semantics

RDB Query syntax example (draft)

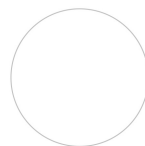
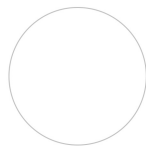
```
prefix : <http://example.org/>
```

```
for locations.tag as $tag, locations.ip as $ip
```

```
from locations
```

```
construct { <{$tag}> :locatedIn <{$ip}> }
```


- Each RDF triple is given an annotation
- Annotations refer to a specific **domain**



- Each RDF triple is given an annotation
- Annotations refer to a specific **domain**

Temporal:

```
:tag4302 :locatedIn :room311 . "2010-11-09 14:57:51"
```

- Each RDF triple is given an annotation
- Annotations refer to a specific **domain**

Temporal:

```
:tag4302 :locatedIn :room311 . "2010-11-09 14:57:51"
```

Fuzzy:

```
:tag4302 :locatedIn :room311 . "0.9"
```

- Each RDF triple is given an annotation
- Annotations refer to a specific **domain**

Temporal:

```
:tag4302 :locatedIn :room311 . "2010-11-09 14:57:51"
```

Fuzzy:

```
:tag4302 :locatedIn :room311 . "0.9"
```

Annotated SPARQL Queries:

"When were two people in the same room?"

"Who is closer to room 311?"

- Each RDF triple is given an annotation
- Annotations refer to a specific **domain**

Temporal:

```
:tag4302 :locatedIn :room311 . "2010-11-09 14:57:51"
```

Fuzzy:

```
:tag4302 :locatedIn :room311 . "0.9"
```

Annotated SPARQL Queries:

```
SELECT ?l ?person WHERE {?tag1 :assignedTo :nuno .  
    ?tag1 :locatedIn :room311 . ?l  
    ?tag2 :assignedTo ?person .  
    ?tag2 :locatedIn :room311 . ?l }
```

Query language integrating heterogeneous sources

- Relational Databases
- W3C RDB2RDF Working Group

Query language integrating heterogeneous sources

- Relational Databases
- W3C RDB2RDF Working Group

Integration of annotated data (Temporal, Fuzzy, ...)

- Higher entailment regimes?

Query language integrating heterogeneous sources

- Relational Databases
- W3C RDB2RDF Working Group

Integration of annotated data (Temporal, Fuzzy, ...)

- Higher entailment regimes?

Optimisations

- Based on results from RDBs and XQuery?

Query language integrating heterogeneous sources

- Relational Databases
- W3C RDB2RDF Working Group

Integration of annotated data (Temporal, Fuzzy, . . .)

- Higher entailment regimes?

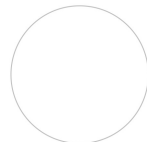
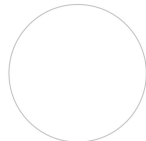
Optimisations

- Based on results from RDBs and XQuery?

Update language

- How to handle the heterogeneous sources?

Above all... Avoid creating



Above all... Avoid creating



Thank you! Questions?