International Semantic Web Conference ISWC 2012

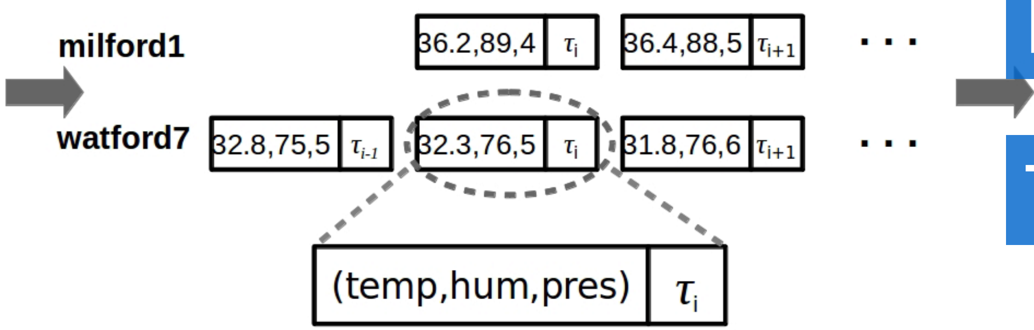
SRBench: A Streaming RDF/SPARQL Benchmark

Ying Zhang[1], Pham Minh Duc[1], Oscar Corcho[2], Jean-Paul Calbimonte[2]

[1]Centrum Wiskunde & Informatica, Amsterdam
[2]Ontology Engineering Group, Universidad Politécnica de Madrid.

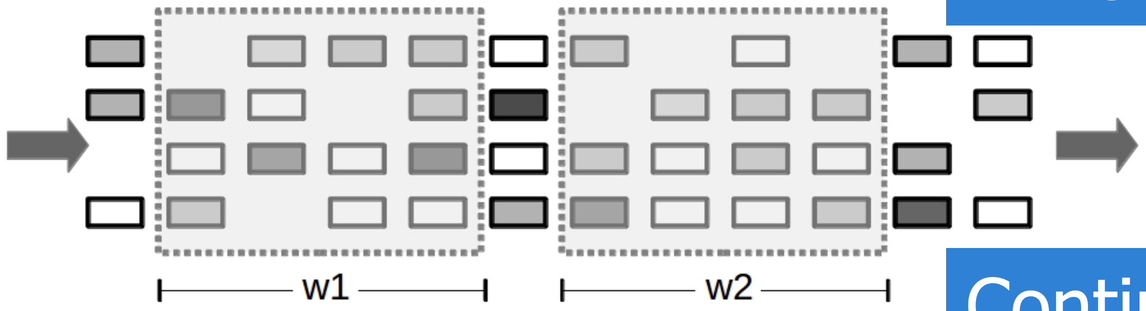
jp.calbimonte@upm.es

Date: 14/11/2012



Data streams

Timestamped tuples



Time windows

Continuous evaluation

e.g. **Data Stream Management Systems (DSMS)**

Sensor Web

"too much **(streaming) data** but not enough (tools to gain and derive) **knowledge**"*

Source: H Patni, C Henson, A Sheth

Why?

Annotate sensor data **with semantic metadata**

Apply **Linked Data principles** to publish streaming data

Interlink streaming data with **existing** datasets

Integrate data stream processing **+ reasoning**

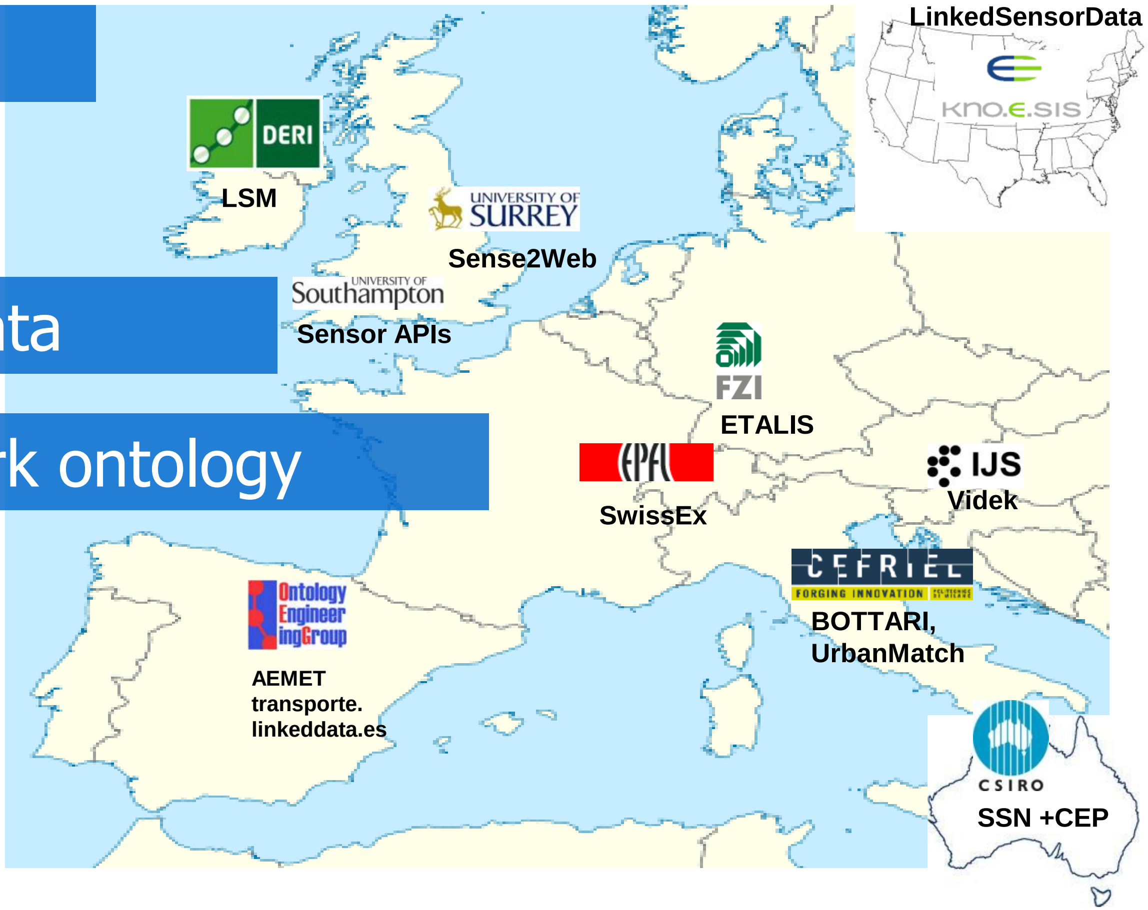
Raise the query abstraction level with **ontologies** "too much **(streaming) data** but not enough (tools to gain and derive) **knowledge**"*

Sensor data publishing

Linked Data

Semantic sensor metadata

Semantic Sensor Network ontology



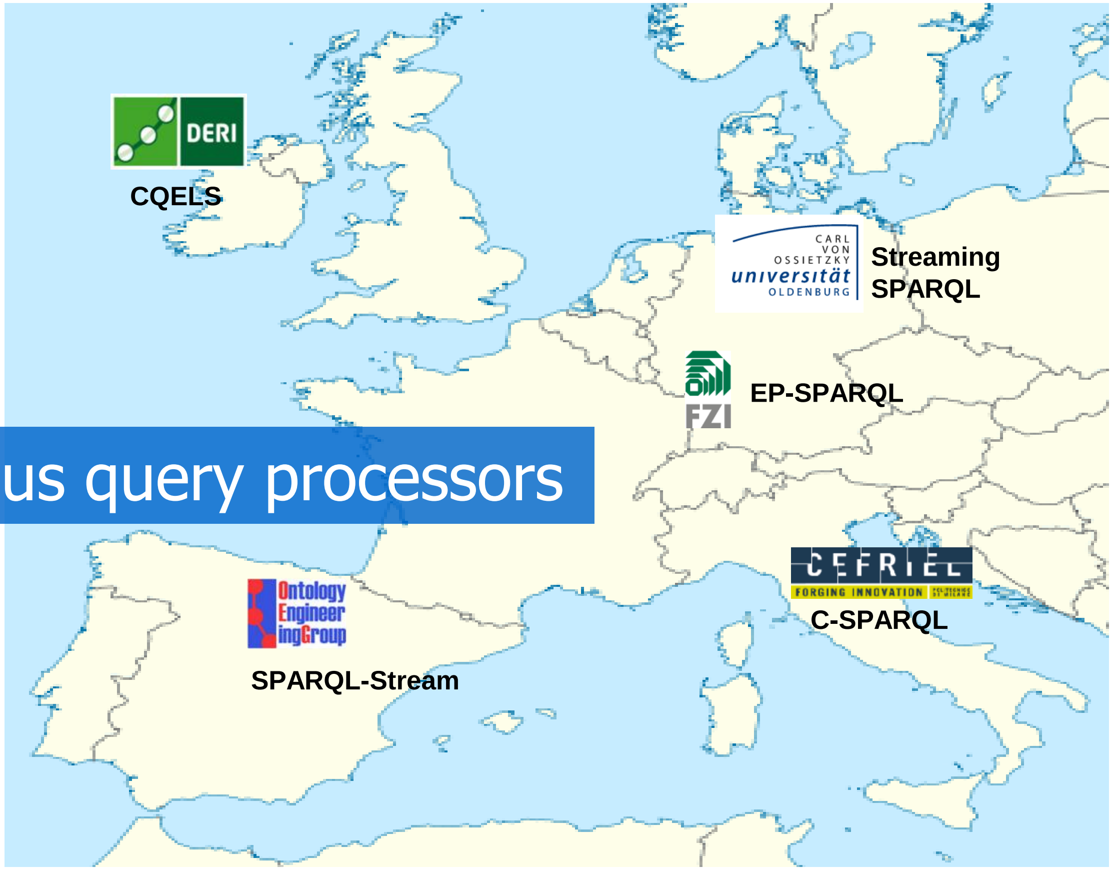
* Sheth et al. 2008, Semantic Sensor Web

Querying **semantic streaming data**

RDF Streams

SPARQL extensions

SPARQL-based continuous query processors



Approaches

Extend RDF for streaming data

Extend SPARQL for streaming RDF

~Similarities

Apply **reasoning** on streaming RDF

Query rewriting to DSMS or CEP

Logic-programming based query evaluation

RDF Streaming engine **from scratch**

Divergence

RDF Streaming Processing Challenges

How to **specify** queries?

↓

Standard query language extensions

✕ **not now**

How to **compare** systems?

↓

Streaming RDF/SPARQL benckmarks

First benchmark for streaming RDF engines

Streaming **RDF/SPARQL** benchmark

Assess engines abilities of dealing with streaming data

Based on **real-world datasets**

Functional evaluation

missing?

crucial?

distinctive?

Existing Benchmarks?

Linear Road Benchmark

relational-based model ≠ RDF graph model

no interlinking data with other datasets

no reasoning

RDF /SPARQL benchmarks

- LUBM, BSBM, SP^2Bench, ...

    meant for stored data

    one off-queries

    single static pre-generated dataset

    do not exploit LOD datasets

    no SPARQL 1.1 features*, no reasoning

* Now the BSBM BI use case includes aggregates

SRBench Challenges

Proper benchmark **dataset** → relevant | realistic | semantically valid | interlinkable

A **concise set** of features → time-bounded | summarization | continuous | data abstraction | contextual data

No **standard** query language → descriptive definition | C-SPARQL | CQELS | SPARQL-Stream

LinkedSensorData

real-world U.S. weather data[1]

first & largest sensor dataset in LOD

LinkedSensorMetadata

~20k US weather stations, ~100k sensors

links to locations in GeoNames nearby

LinkedObservationData

hurricane & blizzard observations in US

~1.73 billion RDF triples

~159 million observations

Name	Storm Type	Date	#Triples	#Observations	Data size
Bill	Hurricane	Aug. 17 – 22, 2009	231,021,108	21,272,790	~15 GB
Ike	Hurricane	Sep. 01 – 13, 2008	374,094,660	34,430,964	~34 GB
Gustav	Hurricane	Aug. 25 – 31, 2008	258,378,511	23,792,818	~17 GB
Bertha	Hurricane	Jul. 06 – 17, 2008	278,235,734	25,762,568	~13 GB
Wilma	Hurricane	Oct. 17 – 23, 2005	171,854,686	15,797,852	~10 GB
Katrina	Hurricane	Aug. 23 – 30, 2005	203,386,049	18,832,041	~12 GB
Charley	Hurricane	Aug. 09 – 15, 2004	101,956,760	9,333,676	~7 GB
	Blizzard	Apr. 01 – 06, 2003	111,357,227	10,237,791	~2 GB

1 http://mesowest.utah.edu

SRBench Datasets

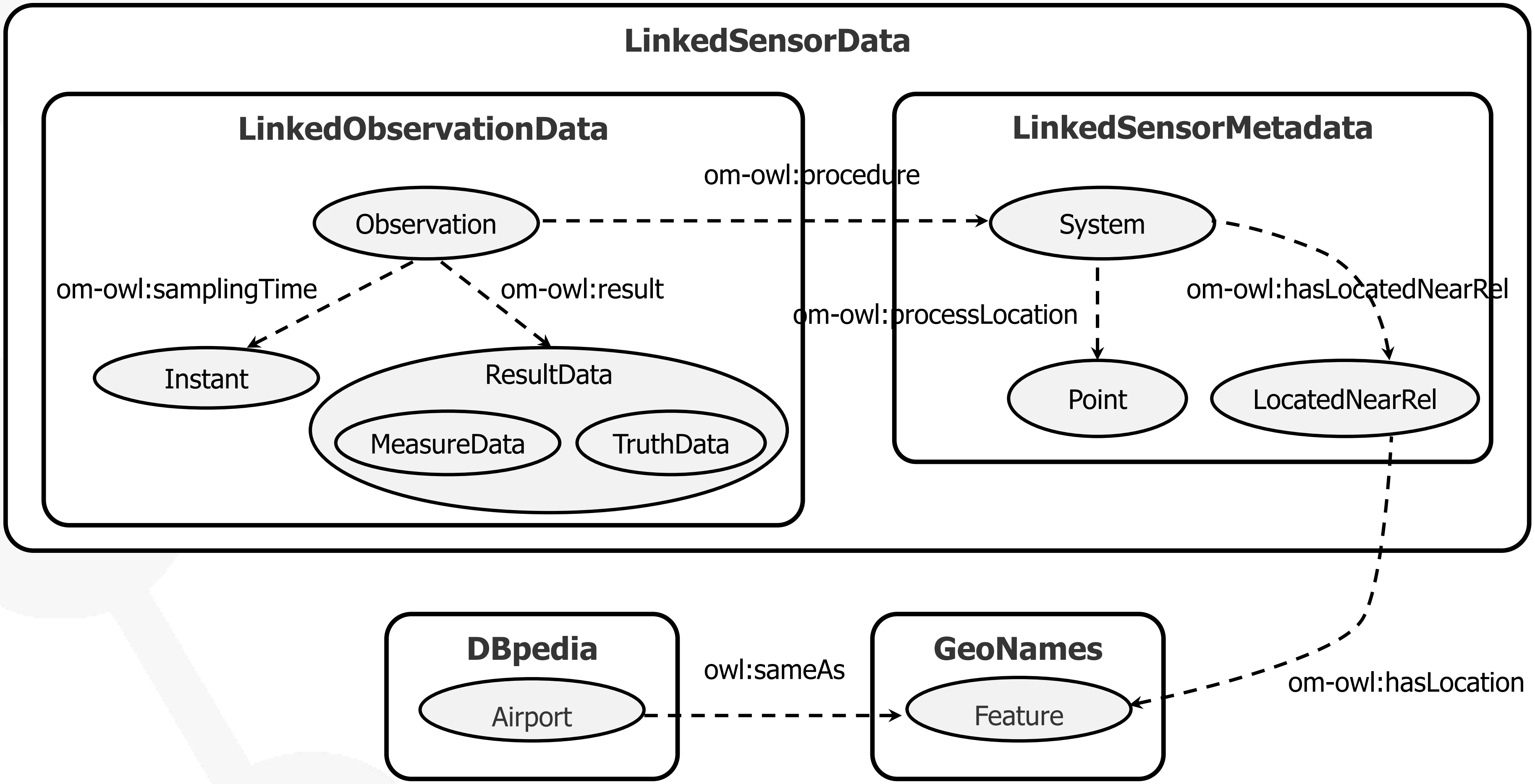
GeoNames

- geographical database, >8M places
- ~8M geographic features
- ~146M RDF triples
- ~10GB on disk

DBpedia

- largest & most popular dataset in LOD
- structured information from Wikipedia
- links to GeoNames through `owl:sameAs`
- we only use the English language collection
  - ~181M RDF triples
  - ~27GB on disk

SRBench Dataset model



17 queries

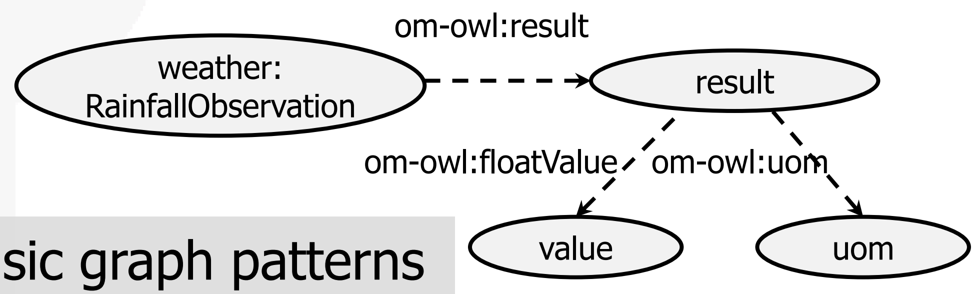
graph pattern matching	and, filter, union, optional
solution modifier	projection, distinct
query form	select, construct, ask
SPARQL 1.1	aggregate, subquery select expr, property path
reasoning	subclass, subproperty, sameAs
streaming	time window, istream dstream, rstream
data access	observations, sensor metadata geonames, dbpedia

Query Features

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17
1.Graph pattern matching	A	A,F,O	A	A,F	A	A,F,U	A	A	A	A	A,F	A,F,U	A,F	A,F,U	A,F	A,F	A,F
2. Solution modifier	P,D	P,D	P	P	P	P	P,D	P	P	P,D	P,D	P	P	P,D	P	P	P
3. Query form	S	S	A	S	C	S	S	S	S	S	S	S	S	S	S	S	S
4. SPARQL 1.1		F,P	A	A,E,M ,F	A,S		N	A,E,M	A,E,M		A,S,M ,F	A,S,E, M,F,P	A,E,M ,F,P	F,P	A,E,M ,P	P	P
5. Reasoning		C	R												C	A	C
6. Streaming	T	T	T	T	T	T	T,D	T	T	T	T	T	T	T	T		
7. Dataset	O	O	O	O	O	O	O	O,S	O,S	O,S	O,S	O,S,G	O,S,G	O,S,G	O,S,D	O,S,G ,D	S

1. **A**nd, **F**ilter, **U**nion, **O**ptional
2. **P**rojection, **D**istinct
3. **S**elect, **C**onstruct, **A**sk
4. **A**ggregate, **S**ubquery, **N**egation, **E**xpr in SELECT, assign**M**ent, **F**unctions&operators, **P**ropertyPath
5. sub**C**lassOf, subp**R**opertyOf, owl:same**A**s
6. **T**ime-based window, **I**stream, **D**stream,Rstream
7. Linked**O**bservationData, Linked**S**ensorMetadata, **G**eoNames, **D**bpedia

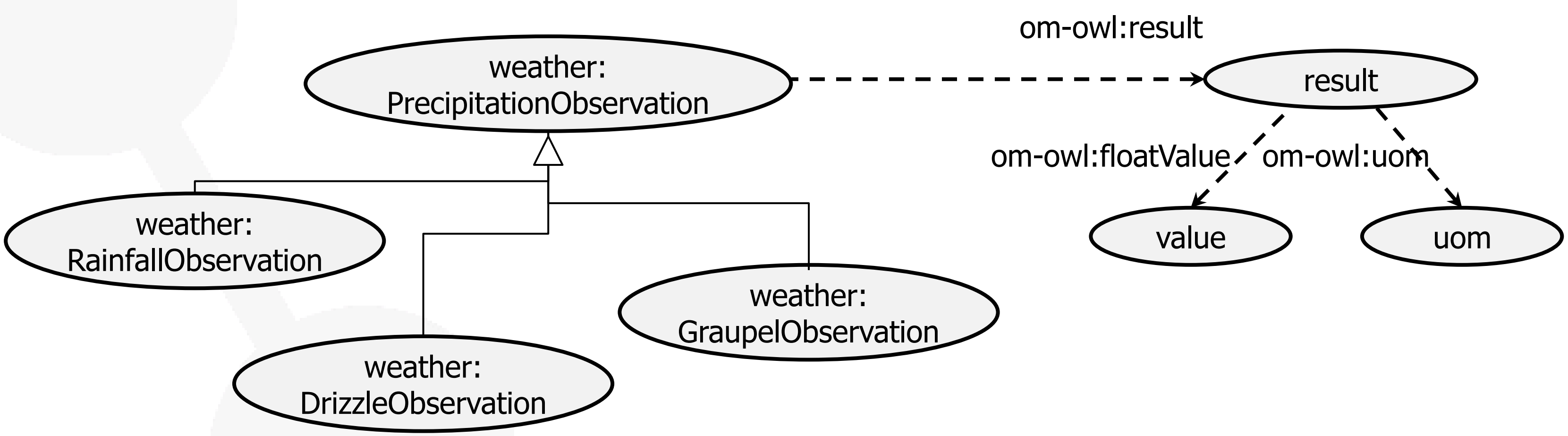
Q1. Get the rainfall observed once in an hour



time-window

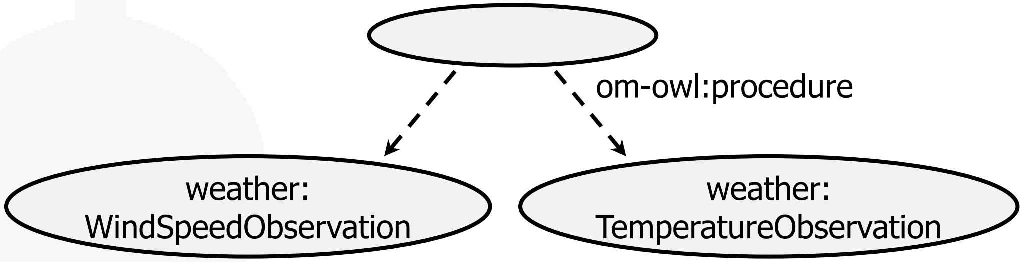
basic graph patterns

Q2. Get all precipitation observed once in an hour

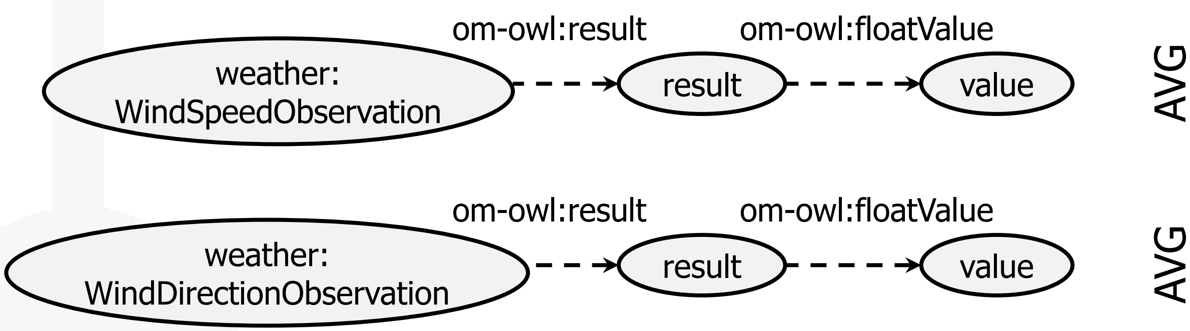


Q4. Get average wind speed at stations where the air temperature is >32 deg. in the last hour every 10 min

aggregates | filter | time-window



Q9. Get the daily average wind force and direction observed by the sensor at a given location.



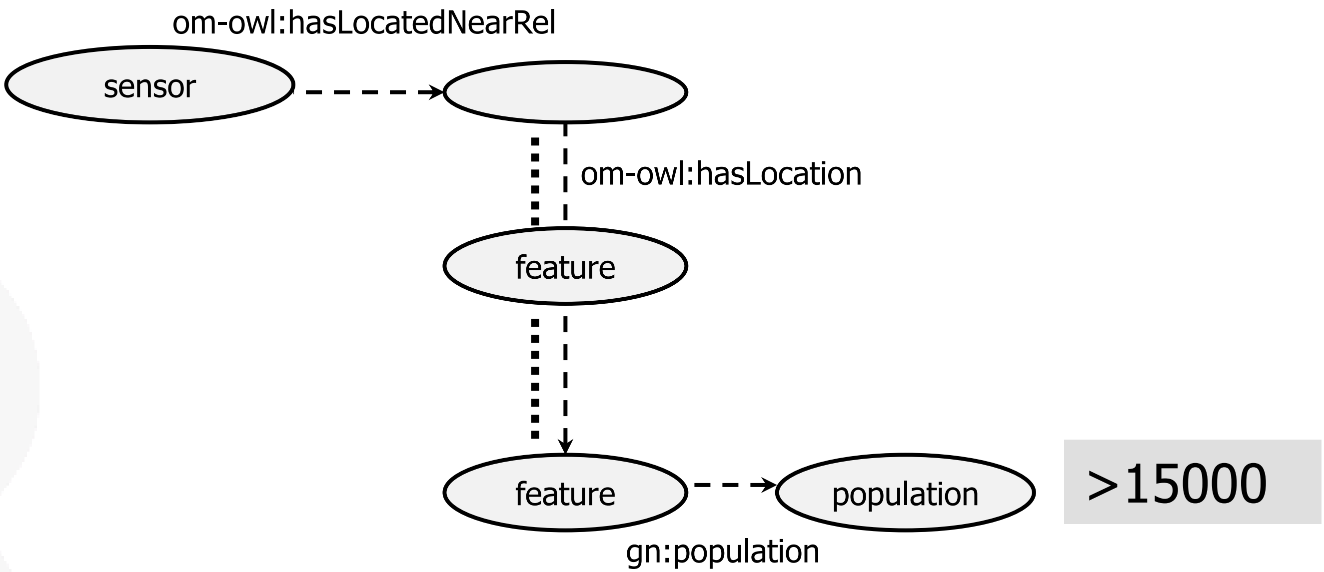
< 1 → 0
< 4 → 1
< 8 → 2
<13 → 3
…

Beaufort scale

Some semantics to bare wind speed numbers

Post process qualified triple patterns

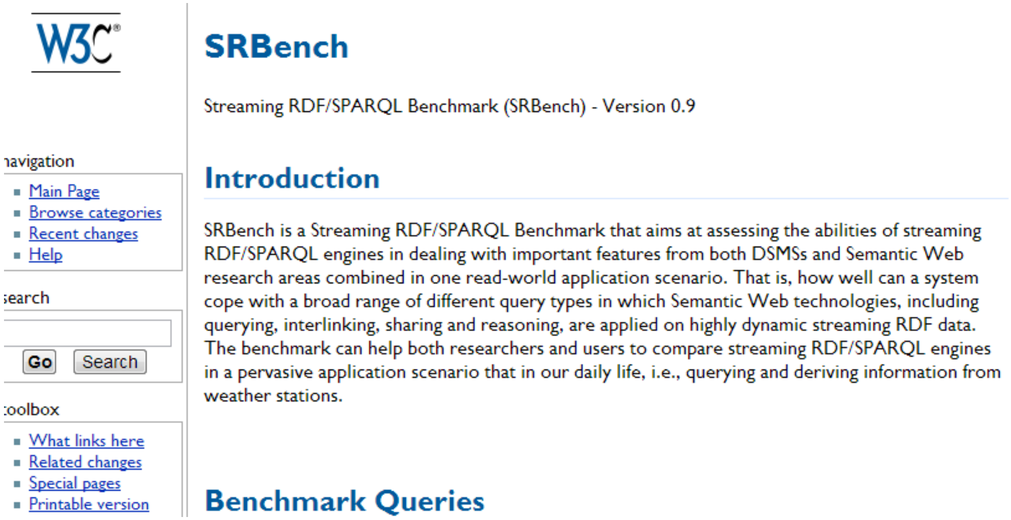
Q12. Get the hourly average air temperature and humidity of large cities



http://www.w3.org/wiki/SRBench

- **C-SPARQL**
- **SPARQLStream**
- **CQELS**

Not exhaustive!



SRBench

Streaming RDF/SPARQL Benchmark (SRBench) - Version 0.9

Introduction

SRBench is a Streaming RDF/SPARQL Benchmark that aims at assessing the abilities of streaming RDF/SPARQL engines in dealing with important features from both DSMSs and Semantic Web research areas combined in one read-world application scenario. That is, how well can a system cope with a broad range of different query types in which Semantic Web technologies, including querying, interlinking, sharing and reasoning, are applied on highly dynamic streaming RDF data. The benchmark can help both researchers and users to compare streaming RDF/SPARQL engines in a pervasive application scenario that in our daily life, i.e., querying and deriving information from weather stations.

Benchmark Queries

Q6. Get the stations that have observed extremely low visibility in the last hour.

SPARQLStream

```
SELECT ?sensor
FROM NAMED STREAM <http://www.cwi.nl/SRBench/observations> [NOW - 1 HOURS]
WHERE {
  {  ?observation om-owl:procedure ?sensor ;  a weather:VisibilityObservation ;
                    om-owl:result [om-owl:floatValue ?value ] . FILTER ( ?value < "10"^^xsd:float) }
  UNION
  {  ?observation om-owl:procedure ?sensor ;  a weather:RainfallObservation ;
                    om-owl:result [om-owl:floatValue ?value ] .  FILTER ( ?value > "30"^^xsd:float) }
  UNION
  {  ?observation om-owl:procedure ?sensor ;  a weather:SnowfallObservation . } }
```

C-SPARQL

```
SELECT ?sensor
FROM NAMED STREAM <http://www.cwi.nl/SRBench/observations>[RANGE 1h TUMBLING]
WHERE {
  { ?observation om-owl:procedure ?sensor ; a weather:VisibilityObservation ;
                    om-owl:result [om-owl:floatValue ?value ] . FILTER ( ?value < "10"^^xsd:float) }
  UNION
  { ?observation om-owl:procedure ?sensor ; a weather:RainfallObservation ;
                    om-owl:result [om-owl:floatValue ?value ] . FILTER ( ?value > "30"^^xsd:float) }
  UNION
  { ?observation om-owl:procedure ?sensor ; a weather:SnowfallObservation . } }
```

CQELS

```
SELECT ?sensor
WHERE {
STREAM <http://www.cwi.nl/SRBench/observations> [RANGE 3600s] {
{ ?observation om-owl:procedure ?sensor ; a weather:VisibilityObservation ;
                    om-owl:result [om-owl:floatValue ?value ] . FILTER ( ?value < "10"^^xsd:float)}
UNION
{ ?observation om-owl:procedure ?sensor ; a weather:RainfallObservation ;
                    om-owl:result [om-owl:floatValue ?value ] . FILTER ( ?value > "30"^^xsd:float) }
UNION
{ ?observation om-owl:procedure ?sensor ; a weather:SnowfallObservation . } } }
```

Q3. Detect if a hurricane has been observed

« A hurricane has a sustained wind (for more than 3 hours) of at least 33 metres per second or 74 miles per hour (119 km/h) »

```
ASK FROM NAMED STREAM
    <http://www.cwi.nl/SRBench/observations> [NOW - 3 HOURS SLIDE 10 MINUTES]
WHERE {
  ?observation om-owl:procedure ?sensor ; om-owl:observedProperty weather:WindSpeed ;
               om-owl:result [ om-owl:floatValue ?value ] . }
GROUP BY ?sensor HAVING ( AVG(?value) >= "74"^^xsd:float )
```

SPARQLStream

```
ASK FROM STREAM
     <http://www.cwi.nl/SRBench/observations> [RANGE 1h STEP 10m]
WHERE {
  ?observation om-owl:procedure ?sensor ; om-owl:observedProperty weather:WindSpeed ;
                         om-owl:result [ om-owl:floatValue ?value ] . }
GROUP BY ?sensor HAVING ( AVG(?value) >= "74"^^xsd:float )
```

C-SPARQL

```
ASK WHERE {
  STREAM <http://www.cwi.nl/SRBench/observations> [RANGE 10800s SLIDE 600s] {
    ?observation om-owl:procedure ?sensor ; om-owl:observedProperty weather:WindSpeed ;
                 om-owl:result [ om-owl:floatValue ?value ] .} }
GROUP BY ?sensor HAVING ( AVG(?value) >= "74"^^xsd:float )
```

CQELS

Q2. Get all precipitation observed once in an hour

```
SELECT DISTINCT ?sensor ?value ?uom
FROM NAMED STREAM
        <http://www.cwi.nl/SRBench/observations> [NOW - 1 HOURS]
WHERE {
  ?observation om-owl:procedure ?sensor ;
                    rdf:type/rdfs:subClassOf* weather:PrecipitationObservation ;
                    om-owl:result ?result .
  ?result om-owl:floatValue ?value .
  OPTIONAL {  ?result om-owl:uom ?uom . }
}
```

Functional Evaluation

System	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17
SPARQLStream	●	PP	A	G	G	●	●	G	G,IF	SD	SD	PP,SD	PP,SD	PP,SD	PP,SD	PP,SD	PP,SD
CQELS	●	PP	A	●	●	●	D/N	●	IF	●	●	PP	PP	PP	PP	PP	PP
C-SPARQL	●	PP	A	●	●	●	D	●	IF	●	●	PP	PP	PP	PP	PP	PP

Ask
Dstream
Group by and aggregations
IF expression
Negation
Property Path
Static **D**ataset

Evaluation:Discussion

- the graph pattern matching features
- solution modifiers
- SELECT and CONSTRUCT query forms

- property path expressions are not supported
- lack of support for the ASK
- DSTREAM, alternatively NOT EXISTS

- Lack of reasoning
- C-SPARQL→simple RDF entailment
- SPARQLStream → ontology-based query rewriting
- CQELS → Native implementation

Ongoing work: Tests criteria

- Correctness
  - query results validated
  - possible variations in ordering
  - possibly multiple valid results per query
  - mismatch, precision/recall
- Throughput:
  - maximal number data items a strRS engine is able to process per time unit
- Scalability:
  - increasing number of incoming streams
  - Increasing number of continuous queries to be processed
- Response time:
  - minimal elapsed time between a data item entering the system and being returned as output of a query
  - mainly relevant for queries allowing immediate query results upon receiving of a data item

Evaluation Issues

- Correctness, Throughput, Scalability
  - Different outputs
  - Differences in query semantics?
  - Very different query evaluation approaches
  - Reasoning
  - Execution parameters

- Framework with a toolset for evaluating Linked Stream Data engines

  Linked Stream Data Processing Engines: Facts and Figures.
  Le-Phuoc et al. ISWC 2012

- Subset of functionalities
  - Missing functions, reasoning, property paths, window-to-stream,
- Synthetic data
  - but flexibility
- First set of performace tests
- Showcase benefits of using semantic technologies?

Close to the end

- SRBench: the first benchmark for streaming RDF engines
- Version 1
  - SRBench specification
  - Functional evaluation
- Much room left for improvements
  - Streaming RDF processing is an evolving topic
  - Exploiting more reasoning possibilities on semantic data
  - Performance evaluation in Version 2

Questions, please.

jp.calbimonte@upm.es