

#### RDF Stream Processing and the Role of Semantics

@jpcik

#### Jean-Paul Calbimonte

LSIR EPFL



Stream Reasoning Workshop 2015

Vienna, 10.11.2015

## Why Streams?



Internet of Things Sensor Networks Mobile Networks Smart Devices Participatory Sensing Transportation

"It's a streaming world!"

Financial Data Social Media Urban Planning Health Monitoring Marketing

2



"The man who is swimming against the stream knows the strength of it."



W. Wilson



# 

#### input stream

 $\phi\phi\phi$ 

continuous results





## Why Streams?

- Web standards
   Data discovery
   Data sharing
   Web queries
- Query languages Query answering Efficient processing Query Federation
- Semantics Vocabularies Data Harvesting Data linking Matching
- Ontologies
   Expressivity
   Inference
   Rule processing
   Knowledge bases



Workinprogress

#### The RSP Community





## The RSP Community





#### W3C RSP Community Group

http://www.w3.org/community/rsp

discuss standardize Effort to formalize evangelize

our work on RDF stream processing



#### Linked Data on the Web



#### Web of Data Linked Data

pest





8

## Querying the Web of Data



Use Linked Data principles for representing, querying and processing RDF streams?



#### **RDF** Streams



stream of RDF data



A bit of semantics in the data Well defined ontology models Web standards





#### Continuous extensions of RDF

 As you know, "RDF -> standard model for data interchange on the Web" (http://www.w3.org/RDF/)

> <sub<sub>1</sub> pred<sub>1</sub> obj<sub>1</sub>> <sub<sub>2</sub> pred<sub>2</sub> obj<sub>2</sub>>

Simple model: Subject Predicate Object

- Extend RDF to model data streams
- A data stream is an (infinite) ordered sequence of data items
- A data item is a self-consumable informative unit



#### Data items

- With **data item** we can refer to:
  - 1. A triple

<:alice :isWith :bob>

2. A graph

<:alice :posts :p>
<:p :who :bob>
<:p :where :redRoom>

:graph1



#### RDF stream model



- A commonly adopted RDF stream model
  - A RDF triple is an event
  - Application time: point-based

<:alice :isWith :bob>:[1]
<:alice :isWith :carl>:[3]
<:bob :isWith :diana>:[6]



### **RSP Data Model**

Timestamped Graph

:g1 {:axel :isIn :RedRoom. :darko :isIn :RedRoom} {:g1 prov:generatedAtTime "2001-10-26T21:32:52"}

Allows:

- Many/One-triple graphs
- ightarrow Multiple time predicates
- ightarrow Implicit timestamp
- Different timestamp representations
- Contemporaneity

#### **RDF Stream**

A RDF stream S consists of a sequence of timestamped graphs (with a partial order)

:g1 {:axel :isln :RedRoom. :darko :isln :RedRoom} {:g1,prov:generatedAtTime,t1} :g2 {:axel :isln :BlueRoom. } {:g2,prov:generatedAtTime,t2} :g3 {:minh :isln :RedRoom. } {:g3,prov:generatedAtTime,t3} ...

Substream, time-bounded substream, window, window function, ...



https://github.com/streamreasoning/RSP-QL/blob/master/Semantics.md

## Querying RDF: SPARQL

#### Where is alice?

<:alice :posts :p>

<:p :who :bob>

<:p :where :redRoom>

#### How about streams?

SELECT ?room WHERE{
 :alice :posts ?post
 ?post :where ?room
}

dow many posts about bob?
SELECT COUNT(?post) WHERE{
 ?person :posts ?post
 :post :who :bob
}



#### Querying data streams – The CQL model



\*Stream operators



#### CQL-like extensions for RDF stream queries





#### Ontology-based access for data streams



#### Similar (not equals!) query languages

```
om-owl:result [ om-owl:floatValue ?value ] . }
GROUP BY ?sensor HAVING ( AVG(?value) >= "74"^^xsd:float )
```

SELECT ?sensor
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 )



#### Classification of existing systems

	Model	Continuous execution	Union, Join, Optional, Filter	Aggregates	Time window	Triple window	R2S operator	Sequence, Co- ocurrence
TA-SPARQL	TA-RDF	Х	$\checkmark$	Limited	Х	Х	Х	Х
tSPARQL	tRDF	Х	$\checkmark$	Х	Х	Х	Х	Х
Streaming SPARQL	RDF Stream	1	$\checkmark$	Х	$\checkmark$	$\checkmark$	X	Х
C-SPARQL	RDF Stream	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	Rstream only	time function
CQELS	RDF Stream	~	$\checkmark$	√	$\checkmark$	$\checkmark$	lstream only	Х
SPARQLStrea m	(Virtual) RDF Stream	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	Х	$\checkmark$	Х
EP-SPARQL	RDF Stream	$\checkmark$	$\checkmark$	$\checkmark$	Х	Х	Х	$\checkmark$
Instans	RDF	$\checkmark$	$\checkmark$	$\checkmark$	Х	Х	X	Х

Disclaimer: other features may be missing

20

COLE POLYTECHNIQUE



Continuously look for bars where people are falling in love (because of a book )



#### Semantics of SPARQL for Streams

• EP-SPARQL: a unified language for event processing and stream reasoning. Anicic, D., Fodor, P., Rudolph, S., & Stojanovic, N. In WWW (p. 635-644). ACM. 2011.

• C-SPARQL: a Continuous Query Language for RDF Data Streams. Barbieri, D. F., Braga, D., Ceri, S., Della Valle, E., & Grossniklaus, M. Int. J. Semantic Computing, 4(1), 3-25. 2010.

• Enabling query technologies for the semantic sensor web. Calbimonte, J.-P., Jeung, H., Corcho, Ó., & Aberer, K. Int. J. Semantic Web Inf. Syst., 8(1), 43-63. 2012.

• A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data. Phuoc, D. L., Dao-Tran, M., Parreira, J. X., & Hauswirth, M.In ISWC (Vol. 7031, p. 370-388). Springer. 2011.

RSP-QL Semantics: a Unifying Query Model to Explain Heterogeneity of RDF Stream Processing Systems.
D. Dell'Aglio, E. Della Valle, J.-P. Calbimonte, O. Corcho. Int. J. Semantic Web Inf. Syst, 10(4). (in press).
2015.

• LARS: A Logic-based Framework for Analyzing Reasoning over Streams. Beck, H., Dao-Tran, M., Eiter, T., Fink, M. In AAAI. 2015.



## Querying the Web of Data



Use Linked Data principles for representing, querying and processing RDF streams?



## URIs as Names of Things

http://mysensorweb.me/mytemperature

n observation http://mysensorweb.me/mytemperature/20151110Z10:00:00

http://mysensorweb.me/mytemperature/latest

many observations http://mysensorweb.me/mytemperature/lastMinute

http://mysensorweb.me/mytemperature/lastMonth

http://mysensorweb.me/mytemperature/avgLastMonth

**Different concepts** 

**Different granularities** 

**Different uses** 



#### HTTP URIs

#### http://mysensorweb.me/mytemperature/latest



Internet of Things

How about XMPP, CoAP, MQTT? Websockets?



#### De-referenceable URIs

#### GET http://mysensorweb.me/mytemperature/latest

:Obs1 a my:TemperatureObservation; my:hasValue 33.5 ; my:hasUnit u:Celsius; my:atTime "20151110Z10:00:00".

GET http://mysensorweb.me/mytemperature

Get the whole stream?

GET http://mysensorweb.me/mytemperature/lastMonth

Get continuous updates?



## Link to other URIs

• Broken links?

http://mysensorweb.me/mytemperature/20151110Z10:00:00

- Mix streaming and stored data
- Persist or not persist?
- Volatile links?



## Looking back 10 years ago...



"8 requirements of real-time stream processing"<sup>[2]</sup> Keep data moving
Query with stream SQL
Handle imperfections
Predictable outcomes
Integrate stored data
Data safety & availability
Partition & scale
Respond Instantaneously

Do we address them? Do we have more requirements? Do we need to do more?



## Reactive Systems



*Events:* Event-Driven *Load:* Scalable *Failure:* Resilient *Users:* Responsive

Keep data moving
Query with stream SQL
Handle imperfections
Predictable outcomes
Integrate stored data
Data safety & availability
Partition & scale
Respond Instantaneously

Do we address them? Do we have more requirements? Do we need to do more?



#### *Reactive* RSPs

8 Requirements

Keep data moving Query with stream SQL Handle imperfections Predictable outcomes Integrate stored data Data safety & availability Partition & scale Respond Instantaneously We go beyond only these **Data Heterogeneity Data Modeling Stream Reasoning** Data discovery **Stream data linking Query** optimization ... more

#### Reactive Principles

Needed if we want to build relevant systems



30

#### Reactive RSP workflows



Minimal agreements: standards, serialization, interfaces Formal models for RSPs and reasoning Working prototypes/systems!





#### Muchas gracias!

@jpcik

## Jean-Paul Calbimonte

