This work is licensed under the Creative Commons Attribution 3.0 Unported License.

Your are free:

- **to Share** — to copy, distribute and transmit the work
- **to Remix** — to adapt the work

Under the following conditions

**Attribution** — You must attribute the work by inserting
- “[source http://streamreasoning.org/sr4ld2013]” at the end of each reused slide
- a credits slide stating

To view a copy of this license, visit [http://creativecommons.org/licenses/by/3.0/](http://creativecommons.org/licenses/by/3.0/)
Agenda

- Revisiting the research challenges
  - Relation with DSMSs and CEPs
    - Reasoning on RDF streams
    - Dealing with incomplete & noisy data
    - Engineering Stream Reasoning Applications
- What's next?
- More on Stream Reasoning at ISWC 2013
### Research Challenges

- **Relation with DSMSs and CEPs**
  - Just as RDF relates to data-base systems?

- **Data types and query languages for semantic streams**
  - Just RDF and SPARQL but with continuous semantics?

- **Reasoning on Streams**
  - Theory: formal semantics
  - Efficiency
  - Scalability and approximation

- **Dealing with incomplete & noisy data**
  - Even more than on the current Web of Data

- **Distributed and parallel processing**
  - Streams are parallel in nature, data stream sources are distributed, ...

- **Engineering Stream Reasoning Applications**
  - Development Environment
  - Integration with other technologies
  - Benchmarks as rigorous means for comparison
Revisiting the research challenges
Relation with DSMSs and CEPs

- **Achievement**
  - Somehow just as RDF, SPARQL, and OWL relate to data-base systems

<table>
<thead>
<tr>
<th>DB → Semantic Web</th>
<th>DSMS/CEP → Semantic Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relational data → RDF</td>
<td>Data streams → RDF Streams</td>
</tr>
<tr>
<td>SQL → SPARQL</td>
<td>CQL/EPL/… → C-SPARQL/EP-SPARQL/…</td>
</tr>
<tr>
<td>Schema → OWL</td>
<td>Schema → OWL</td>
</tr>
</tbody>
</table>

- But with some differences
  - Queries are registered → opportunity for query optimizations
  - Many application requires a network of queries → opportunity for inter-query optimizations

- **Issues**
  - It is time to bring Stream Reasoning to the Web
    - Volatile URIs
    - Serialization of RDF streams
    - Protocols: HTTP, Web sockets
Revisiting the research challenges
Data types for semantic streams - Achievements

- **RDF streams** introduced as new data type in the Semantic Web and Linked Data research
- W3C RDF stream processor **community group started** to jointly work out a recommendation in 2014
  - [http://www.w3.org/community/rsp/](http://www.w3.org/community/rsp/)
Revisiting the research challenges

Data types for semantic streams - Issues

- Multiple notions of RDF stream proposed
  - Ordered sequence (implicit timestamp)
  - One timestamp per triple (point in time semantics)
  - Two timestamps per triple (interval base semantics)

- Comparison between existing approaches

<table>
<thead>
<tr>
<th>System</th>
<th>Data item</th>
<th>Time model</th>
<th># of timestamps</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTANS</td>
<td>triple</td>
<td>Implicit</td>
<td>0</td>
</tr>
<tr>
<td>C-SPARQL</td>
<td>triple</td>
<td>Point in time</td>
<td>1</td>
</tr>
<tr>
<td>SPARQL\text{stream}</td>
<td>triple</td>
<td>Point in time</td>
<td>1</td>
</tr>
<tr>
<td>CQELS</td>
<td>triple</td>
<td>Point in time</td>
<td>1</td>
</tr>
<tr>
<td>Sparkwave</td>
<td>triple</td>
<td>Point in time</td>
<td>1</td>
</tr>
<tr>
<td>Streaming Linked Data</td>
<td>RDF graph</td>
<td>Point in time</td>
<td>1</td>
</tr>
<tr>
<td>ETALIS</td>
<td>triple</td>
<td>Interval</td>
<td>2</td>
</tr>
</tbody>
</table>

- More investigation is required to agree on an RDF stream model
Languages for continuous querying of and event processing on RDF streams proposed

Window base selection outperforms filter base selection

Dynamic optimization of query plans and incremental evaluation is possible

Multiple RDF stream processor prototypes implemented and deployed

W3C RDF stream processor community group started to jointly work out a recommendation in 2014

http://www.w3.org/community/rsp/
Revisiting the research challenges
Query languages for semantic streams - Issues

- Different syntax for S2R operator
- Semantics of query languages is similar, but not identical
- Lack of R2S operator in some cases
- Different support for time-aware operators
Revisiting the research challenges
Query languages for semantic streams - Issues

- Comparison between existing approaches

<table>
<thead>
<tr>
<th>System</th>
<th>S2R</th>
<th>R2R</th>
<th>Time-aware</th>
<th>R2S</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTANS</td>
<td>Based on time events</td>
<td>SPARQL update</td>
<td>Based on time events</td>
<td>Ins only</td>
</tr>
<tr>
<td>C-SPARQL Engine</td>
<td>Logical and triple-based</td>
<td>SPARQL 1.1 query</td>
<td>timestamp function</td>
<td>Batch only</td>
</tr>
<tr>
<td>SPARQL_stream</td>
<td>Logical and triple-based</td>
<td>SPARQL 1.1 query</td>
<td>no</td>
<td>Ins, batch, del</td>
</tr>
<tr>
<td>CQELS</td>
<td>Logical and triple-based</td>
<td>SPARQL 1.1 query</td>
<td>no</td>
<td>Ins only</td>
</tr>
<tr>
<td>Sparkwave</td>
<td>Logical</td>
<td>SPARQL 1.0</td>
<td>no</td>
<td>Ins only</td>
</tr>
<tr>
<td>Streaming Linked Data</td>
<td>Logical and graph-based</td>
<td>SPARQL 1.1</td>
<td>no</td>
<td>Batch only</td>
</tr>
<tr>
<td>ETALIS</td>
<td>no</td>
<td>SPARQL 1.0</td>
<td>SEQ, PAR, AND, OR, DURING, STARTS, EQUALS, NOT, MEETS, FINISHES</td>
<td>Ins only</td>
</tr>
</tbody>
</table>

- Is it time to converge on a standard?
The *existing engines*

- adopts **different architectural** choices and it is still unclear when each choice is best
  - C-SPARQL, ETALIS, SPARQL\_stream are wrappers for existing systems thus they are more reliable and maintainable
  - CQELS, Streaming Linked Data, INSTANS, Sparkwave are native implementations, thus they are more efficient and offer optimizations not possible in the other system

- They have **different operational semantics**
  - for more information check out the ISWC 2013 evaluation track for "*On Correctness in RDF stream processor benchmarking*" by Daniele Dell’Aglio, Jean-Paul Calbimonte, Marco Balduini, Oscar Corcho and Emanuele Della Valle
Stream Reasoning research field is getting momentum

Efficient **continuous reasoning algorithm** on RDF streams for RDFS, RDFS++, EL++, Answer Set Programming were proposed

Multiple Stream Reasoning **proofs of concept** were implemented
Issues

- Theory still largely based on one-time semantics
  - Continuous reasoning for the following topics requires more investigations
    - Continuous conjunctive queries under OWL2QL entailment regime
    - Union of Continuous conjunctive queries under OWL2QL entailment regime
    - Continuous queries including negation (in all its possible forms)
    - Continuous recursive querying under expressive entailment regimes
    - Modelling in the ontology aggregates and functions
  - Logic based time-management
    - More expressive specification, e.g., calendar algebra
    - Windows that logically resize at runtime
- Lack of prototypes that go beyond proof of concept
- Explore more reasoning form beyond Q/A
Data streams are incomplete and noisy!

Achievements
- Reasoning can help dealing with incompleteness
- Initial works on inductive stream reasoning explored relation learning as a way to cope with those problematic aspects

Issues
- More research required!
Data streams are parallel and distributed in nature!

Achievements
• Proof of concept implemented on S4 and Storm

Issues
• More research required!
Revisiting the research challenges
Engineering Stream Reasoning Applications

Achievements
- Deployments for
  - semantic sensor networks
  - social media analytics
  - City Data Fusion
- Multiple benchmarks proposed

Issues
- It is still unclear when and where it is convenient to adopt Stream Reasoning solutions
- Benchmarks too focused on throughput; correctness and memory allocation cost, too
Revisiting the research challenges
Wrap-up

- Data types and query languages for semantic streams
  - Notion of RDF stream
  - Languages for continuous querying
  - Prototypes
  - Standardization

- Reasoning on RDF streams
  - Theory
  - Algorithms
  - Prototypes

- Dealing with incomplete & noisy data
  - Theory
  - Algorithms
  - Prototypes

- Engineering Stream Reasoning Applications
  - Deployments
  - Benchmarks
**What's next? order matters!**

- Observation: order reflects recency, relevance, trustability ...

<table>
<thead>
<tr>
<th>Types of orders</th>
<th>Combinations</th>
<th>Continuous top-k Q/A</th>
<th>Order-aware reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recency</td>
<td>Top-k Q/A</td>
<td>Top-k Reasoning</td>
<td></td>
</tr>
<tr>
<td>Indexes</td>
<td>DSMS/CEP</td>
<td>Stream reasoning</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional solutions</td>
<td>Scalable reasoning</td>
</tr>
</tbody>
</table>

---

More on Stream Reasoning at ISWC 2013

- Tuesday Afternoon - **OrdRing 2013**
  - 2\textsuperscript{nd} International *Workshop* on Ordering and Reasoning
  - Open Door Meeting of the *W3C RDF Stream Processing Community Group*

- Wednesday Evening - Poster session
  - M. Balduini et al. *A Restful Interface for RDF Stream Processors*
  - L. Fischer et al. *Network-Aware Workload Scheduling for Scalable Linked Data Stream Processing*

- Thursday - 11:00-12:40 **Track on Streams**
  - M. Balduini et al. *Social listening of City Scale Events using the Streaming Linked Data Framework*
  - D. Le Phuoc et al. *Elastic and scalable processing of Linked Stream Data in the Cloud*
  - S. Tallevi-Diotalleli et al. *Real-time Urban Monitoring in Dublin using Semantic and Stream Technologies*
  - D. Dell'Aglion et al. *In Correctness in RDF stream processor benchmarking*
  - D. Gerber et al. *Real-time RDF extraction from unstructured data streams*
Stream Reasoning For Linked Data
M. Balduini, J-P Calbimonte, O. Corcho, D. Dell'Aglio, E. Della Valle, and J.Z. Pan
http://streamreasoning.org/sr4ld2013

Wrap-up and conclusions
Emanuele Della Valle
emanuele.dellavalle@polimi.it
http://emanueledellavalle.org