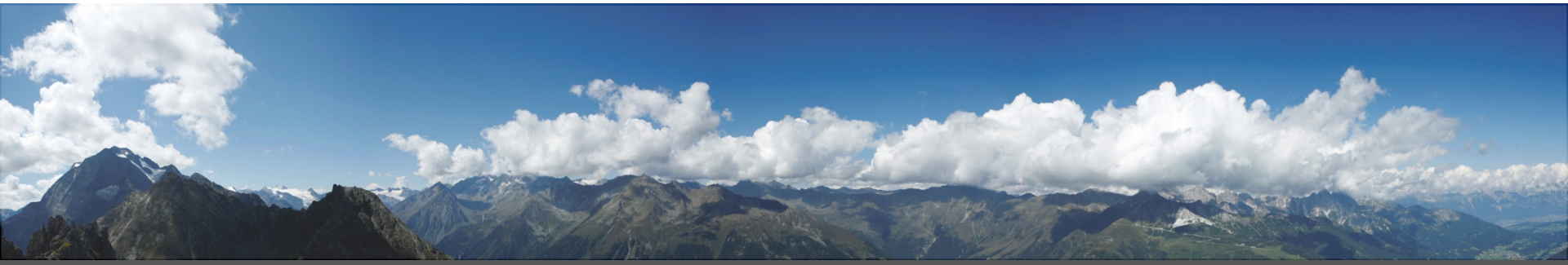
STI · INNSBRUCK

Towards Efficient Schema-Enhanced Pattern Matching over RDF Data Streams

Srdjan Komazec and Davide Cerri



1st International Workshop on Ordering and Reasoning @ ISWC'11, October 23rd, Bonn, Germany

Events and the Web

STI · INNSBRUCK

- **Data streams** are becoming common on the Web
  - Stock exchange tickers
  - Weather information
  - Sensor readings
  - Social networking activity notifications
- **Collecting, integrating and sharing** data streams
  - Pachube platform
- Interpreting data streams as **event sources**
  - Vision of the “Web of Events”

Technologies for stream/event processing

- **(Complex) Event Processing (CEP)**
  - Established computing paradigm providing approaches and techniques to **timely** process streams of events
  - Built around the notion of **event patterns**
  - E.g. Tibco Business Events, Oracle CEP, Esper, Streambase, etc.

- **Data Stream Management Systems (DSMS)**
  - A system which enables querying and maintenance of data in data streams
  - Built around the notion of **continuous queries** against data streams
  - E.g., Aurora and Borealis, STREAM, TelegraphCQ, NiagaraST, etc.

Processing data streams/events on the Web

STI · INNSBRUCK

- Stream/event processing on the Web needs to cope with **openness**, **heterogeneity** and required **scalability**

- **Semantic Web technologies**
  - Facilitating data integration by using machine processable descriptions to reconcile heterogeneities (e.g., Semantic Sensor Web)

- Impact of SW technologies over stream processing and vice versa
  - Entailments and data/event stream processing
  - Data volatility (temporality) and Semantic Web technologies

ETALIS

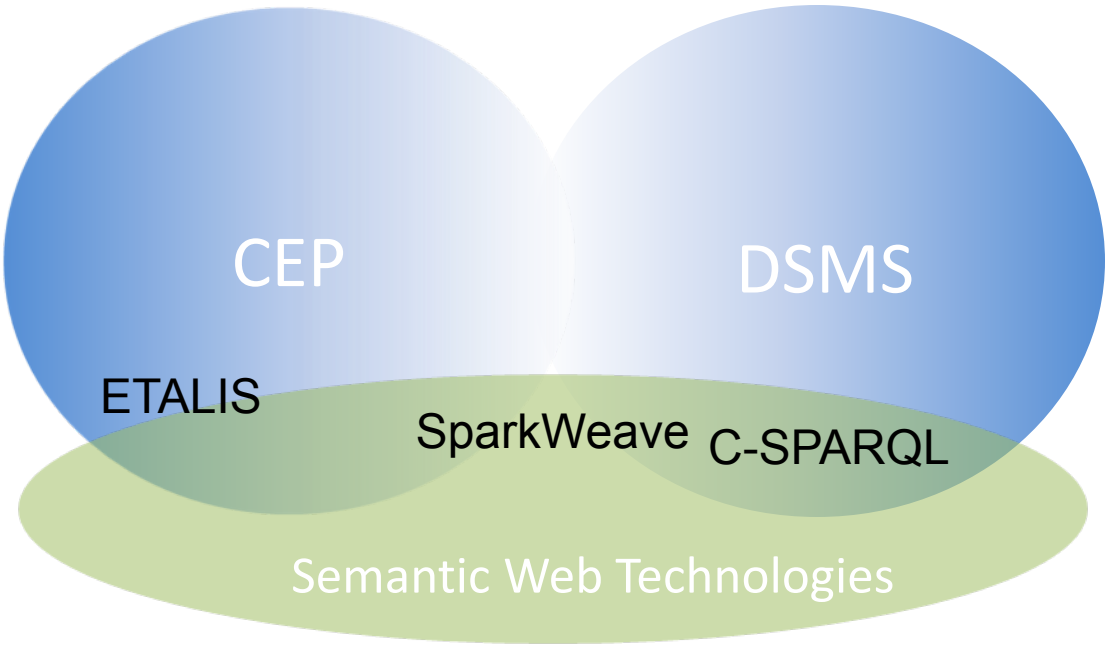
STI · INNSBRUCK

- Event-driven Transaction Logic Inference System
  - A CEP system bringing together the power of logic rule inference and event processing
- Feature rich system
  - Out-of-order events evaluation, aggregate functions computation, dynamic pattern insertion and retraction, several garbage collection policies and consumption strategies, event operators, count-based sliding windows, …
- Relies on a Prolog engine
- Support for RDF data through EP-SPARQL
  - Extends SPARQL with temporal operators at the level of RDF graphs
  - RDFS ontologies and graph patterns are transformed into Prolog statements

C-SPARQL

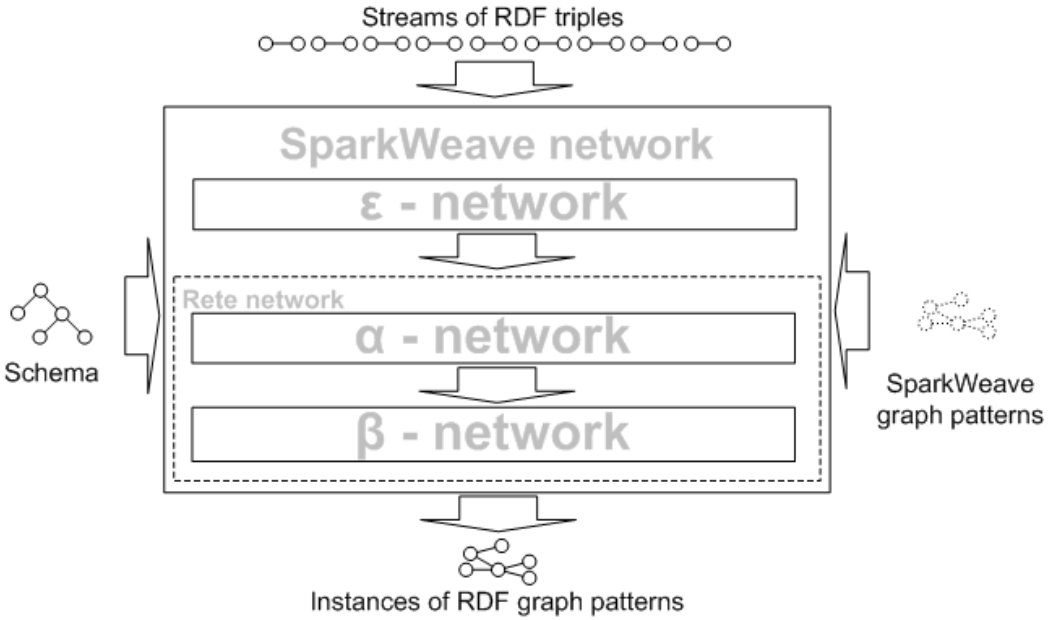
- Continuous SPARQL
  - A SPARQL-based DSMS representative enhanced with reasoning capabilities
- Features
  - Extends SPARQL with the notions of RDF streams, time windows, aggregation functions, handling of multiple streams, …
  - Suitable for situations when a significant amount of static knowledge needs to be combined with data streams to enable time-constrained reasoning
- Performs incremental RDF Schema-based materialisation of RDF snapshots on top of a triple repository
- Limited support for temporal operators
- Doesn't provide truly continuous querying (snapshotting)

SparkWeave position



Introduction

- Providing efficient pattern matching functionalities against RDF streams
  - Enabling expression of temporal RDF graph dependencies
  - Taking into account RDF Schema entailments
- Uses **Rete algorithm** as the foundation
- Extends Rete with **networking structures** necessary to **compute entailments**
- Supports **rich set of time-based features**

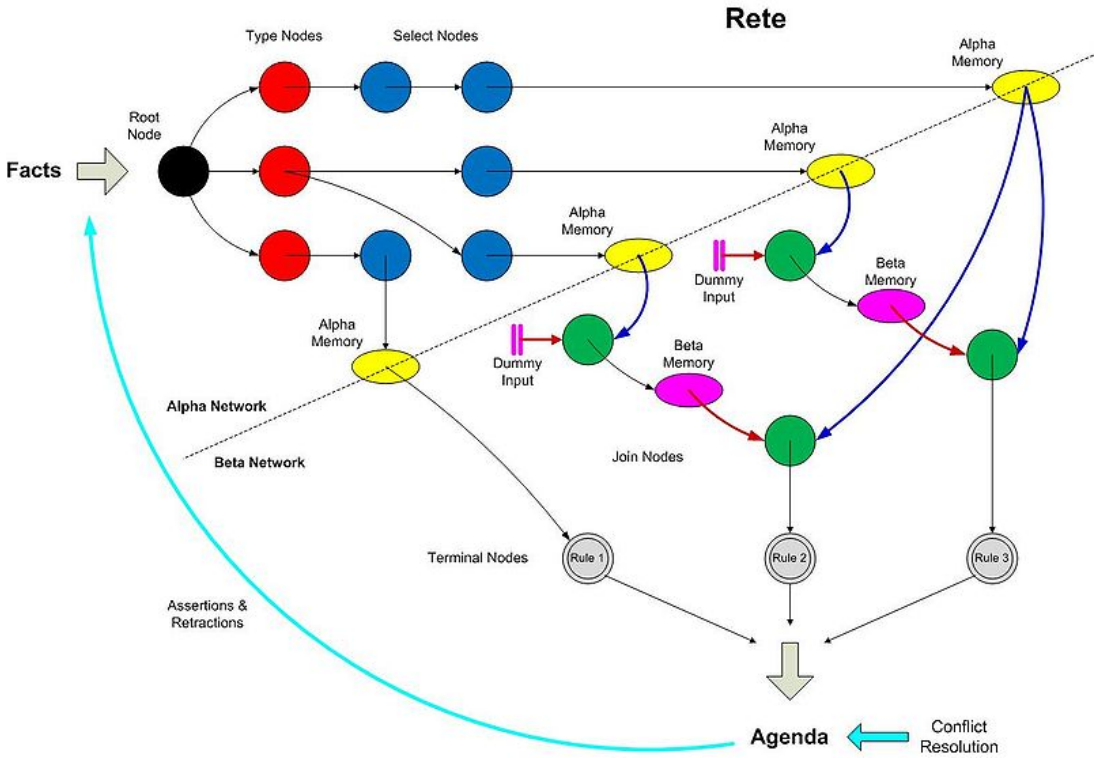


Constraints

- SparkWeave operates over a **fixed schema**
  - Schema definitions are rarely transported over stream (e.g., sensor readings)
- SparkWeave **does not support background knowledge**
  - The system exploits a memory-intensive approach
- SparkWeave **does not provide generic reasoning capabilities**
  - The system is built for RDF Schema (and few OWL constructs)

Rete algorithm as the foundation

- Pattern matching algorithm for implementing production rule systems in many object/many pattern situations.
  - The basis for many popular expert system shells, including CLIPS, Jess, Drools, BizTalk, Rules Engine and Soar.



Basic Rete topography, taken from `http://en.wikipedia.org/wiki/Rete_algorithm`

ε- network - Concept

STI · INNSBRUCK

- Classical Rete architecture is extended with an additional network of entailment nodes called **ε – network**
- **ε – network** follows Rete principles
  - Dataflow organization
  - Usage of tokens to keep state of computation
- **ε – network** is built on top of schema and optimized in accordance to RDF pattern descriptions
  - The network topology is governed by the property hierarchies (with domain and range definitions) and class hierarchies
- Outputs of **ε – network** are connected to inputs of **α – network**

Schema entailment

Rule	If	Then add
rdf1	(x p y)	(p rdf:type rdf:Property)
rdfs2	(p rdfs:domain c) (x p y)	(x rdf:type c)
rdfs3	(p rdfs:range c) (x p y)	(y rdf:type c)
rdfs4a	(x p y)	(x rdf:type rdfs:Resource)
rdfs4b	(x p y)	(y rdf:type rdfs:Resource)
rdfs5	(p rdfs:subPropertyOf q) (q rdfs:subPropertyOf r)	(p rdfs:subPropertyOf r)
rdfs6	(p rdf:type rdf:Property)	(P rdfs:subPropertyOf p)
rdfs7	(p rdfs:subPropertyOf q) (x p y)	(x q y)
rdfs8	(c rdf:type rdfs:Class)	(c rdfs:subClassOf rdfs:Resource)
rdfs9	(c rdfs:subClassOf d) (x rdf:type c)	(x rdf:type d)
rdfs10	(c rdf:type rdfs:Class)	(c rdfs:subClassOf c)
rdfs11	(c rdfs:subClassOf d) (d rdfs:subClassOf e)	(c rdfs:subClassOf e)
rdfs12	(p rdf:type rdfs:ContainerMembershipProperty)	(p rdfs:subPropertyOf rdfs:Member)
rdfs13	(x rdf:type rdfs:Datatype)	(x rdfs:subClassOf rdfs:Literal)
	(p owl:inverseOf q)	(q owl:inverseOf p)
	(p owl:inverseOf q) (x p y)	(y q x)
	(p rdf:type owl:SymmetricProperty) (x p y)	(y p x)

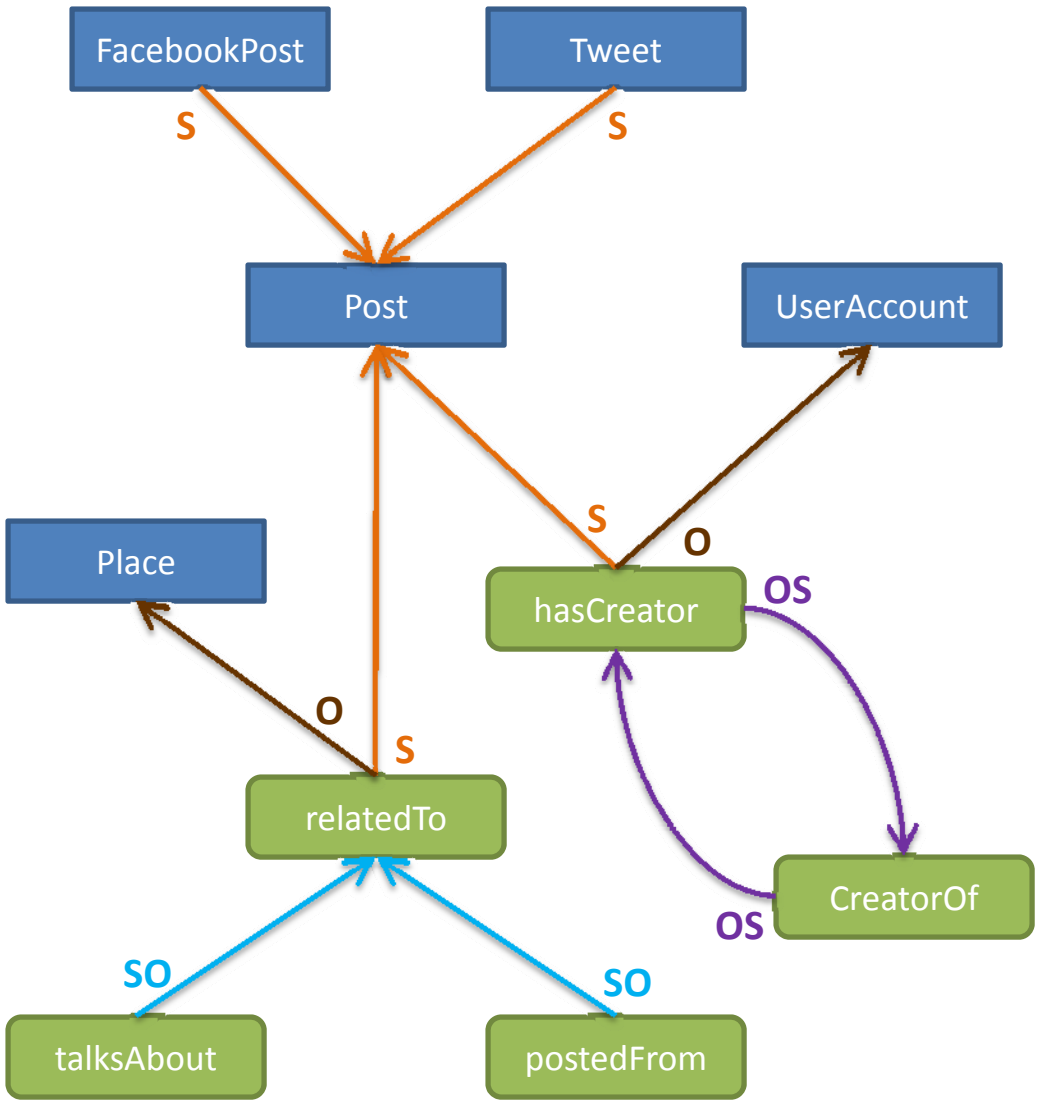
schema only

data + schema

not relevant

ε- network - Example

```
:UserAccount a rdfs:Class .

:Post a rdfs:Class .

:Tweet a rdfs:Class ;
       rdfs:subClassOf :Post .

:FacebookPost a rdfs:Class ;
              rdfs:subClassOf :Post .

:Place a rdfs:Class .

:hasCreator a rdf:Property ;
            rdfs:domain :Post ;
            rdfs:range :UserAccount .

:creatorOf a rdf:Property ;
           owl:inverseOf :hasCreator .

:relatedTo a rdf:Property ;
           rdfs:domain :Post ;
           rdfs:range :Place .

:talksAbout a rdf:Property ;
            rdfs:subPropertyOf :relatedTo .

:postedFrom a rdf:Property ;
            rdfs:subPropertyOf :relatedTo .
```



Time-based features

STI · INNSBRUCK

- **Time window** support
  - Extensions of β-node behaviour for executing time-window boundary checks
- **Temporal operators**
  - Focusing on the interval-based semantics
  - Extensions of join-node behaviour
- **Garbage collection** of discarded triples/tokens
  - Calculation of an object lifetime
  - Incremental and parallel process
- **Event consumption strategies**
  - Extensions of join-node behaviour
  - Currently supports UNRESTRICTED strategy

Conclusions and Further Work

STI · INNSBRUCK

- SparkWeave
  - Schema-enhanced pattern matcher operating aginst RDF data streams
  - Built on top of Rete algorithm extended with an entailment (**ε**) network
  - Supports RDFS + a few OWL constructs

- Future work
  - Integration and implementation of temporal extensions
  - Performance evaluation