Supporting EnviroInfo Systems and Services Realization with the Geo-Spatial and Streaming Dimensions of the Semantic Web

Emanuele Della Valle
DEI - Politecnico di Milano
emanuele.dellavalle@polimi.it

Alessio Carenini
CEFRIEL
alessio.carenini@cefriel.it

http://emanueledellavalle.org

ENVIP @ EnviroInfo2010, Bonn/Cologne, October 6-8, 2010

Agenda

- The problem
- A case study
- Background
- Efficient **Geospatial Analysis** for the Semantic Web
  - GIS-to-RDF (G2R) approach
- Continuous Processing of **Data Streams** for the Semantic Web
  - Continuous-SPARQL (C-SPARQL) approach
- **Combining** the Two Approaches with LarKC
- Conclusions

The Problem

- EnviroInfo Systems and Services require to support a large number of concurrent decision processes
- Critical factors are the ability to **seamlessly**
  - cope with **geo-spatial features** of the environment
  - process in real time huge and possibly noisy **data streams**

A Case Study

" A typical oil production platform is equipped with about **400.000 sensors** for measuring environmental and technical parameters. "

– Einar Landre - STATOIL, 2010

- **Typical questions** oil operation engineers have to answer in the decision processes
  - Given this brand of turbine, what is the expected time to failure when the barring starts to vibrate as now detected?
  - How do I detect weather events from the observation data of the sensors spread around in the environment?
  - Which sensors are observing a blizzard within a 100 mile radius of a given location?

Background

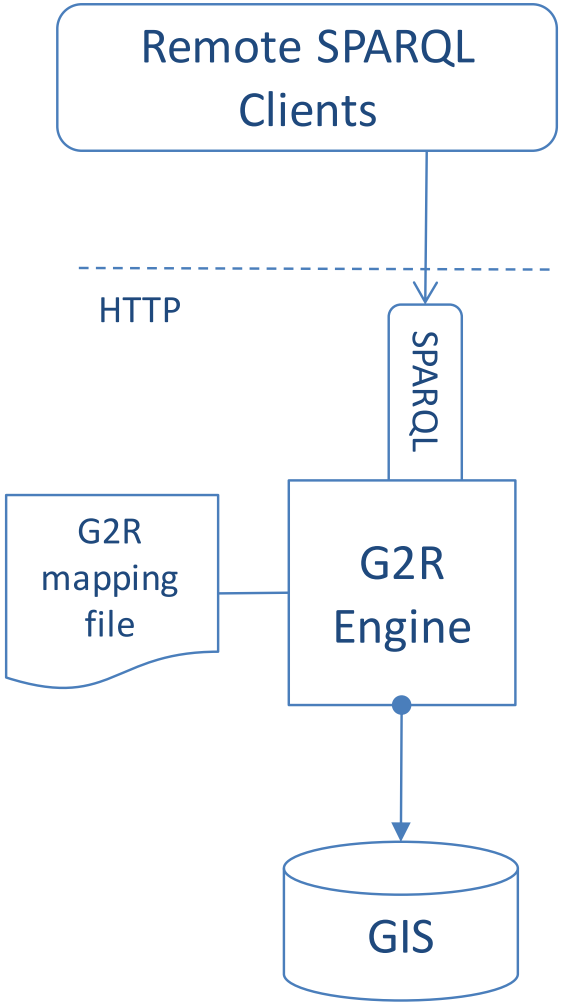
- Technologies are available
  - Geographic Information System (GIS)
  - Data Stream Management Systems (DSMS) and Complex Event Processors (CEP)
- Barriers
  - Seamless integrated usage of GIS and DSMS/CEP
- Proposed solutions
  - Semantic Web (i.e., Linked Data) as the standard approach for data integration
  - Adding two dimensions to the Semantic Web
    - Efficient Geo-Spatial Analysis
    - Data Streaming processing

Efficient Geospatial Analysis for the Semantic Web

- Semantic Web: no built in geo-spatial feature
- There is a growing need of Semantic Web practitioners to efficiently perform geo-spatial analysis
- Available solutions (e.g., Virtuoso or AllegroGraph) offer a limited support if compared to the rich features normally available in a GIS
- Proposed solution:
  - Idea: **Treating GIS as Virtual RDF Graphs**
  - Implementation: GIS to RDF (**G2R**)

GIS-to-RDF (G2R) Approach

- **G2R is**
  - **a**n extension of D2RQ **declarative language to describe mappings** between GIS schemata and OWL/RDFS ontologies, and
  - **a set of extended value testing functions for SPARQL** that leverage SQL/MM spatial function implementation in existing GIS



GIS-to-RDF (G2R) at Work 1/2

Operator Question: detects the platforms within oil-fields in which more than 10 blizzards were detected in the last month

SPARQL query:

```
SELECT ?oilField ?platform
FROM WHERE {
  ?oilField ex:hasSurface ?oilFieldSurface .
  ?platform ex:hasSurface ?platformSurface .
  ?sensor grs:point ?sensorPosition ;
          so:generatedObservation [ a w:blizzard] ;
          so:samplingTime ?time .
  FILTER(g2r:contains(?oilFieldSurface,?sensorPosition)
          && g2r:overlaps(?oilFieldSurface,?platformSurface))
  FILTER(?time >= "2010-10-01T00:00:00Z^^xsd:dateTime")
  FILTER(?time <= "2010-09-01T00:00:00Z^^xsd:dateTime")}
GROUP BY ?oilField
HAVING (COUNT(?sensor) > 10)
```

GIS-to-RDF (G2R) at Work 2/2

Rewritten SQL MM/Spatial query:

```
SELECT o.ID, p.ID,
FROM platform AS p, oilFields AS o, sensors AS s
WHERE s.generatedObservation = "blizzard" AND
        p.area.ST_Within(s.position) = 1 AND
        b.area.ST_Overlaps(o.area) = 1 AND
        s.samplingTime >= "2010-09-01T00:00:00Z" AND
        s.samplingTime <= "2010-10-01T00:00:00Z”
GROUP BY o.ID
HAVING COUNT (s.generatedObservation) > 10
```

Mapping declared:

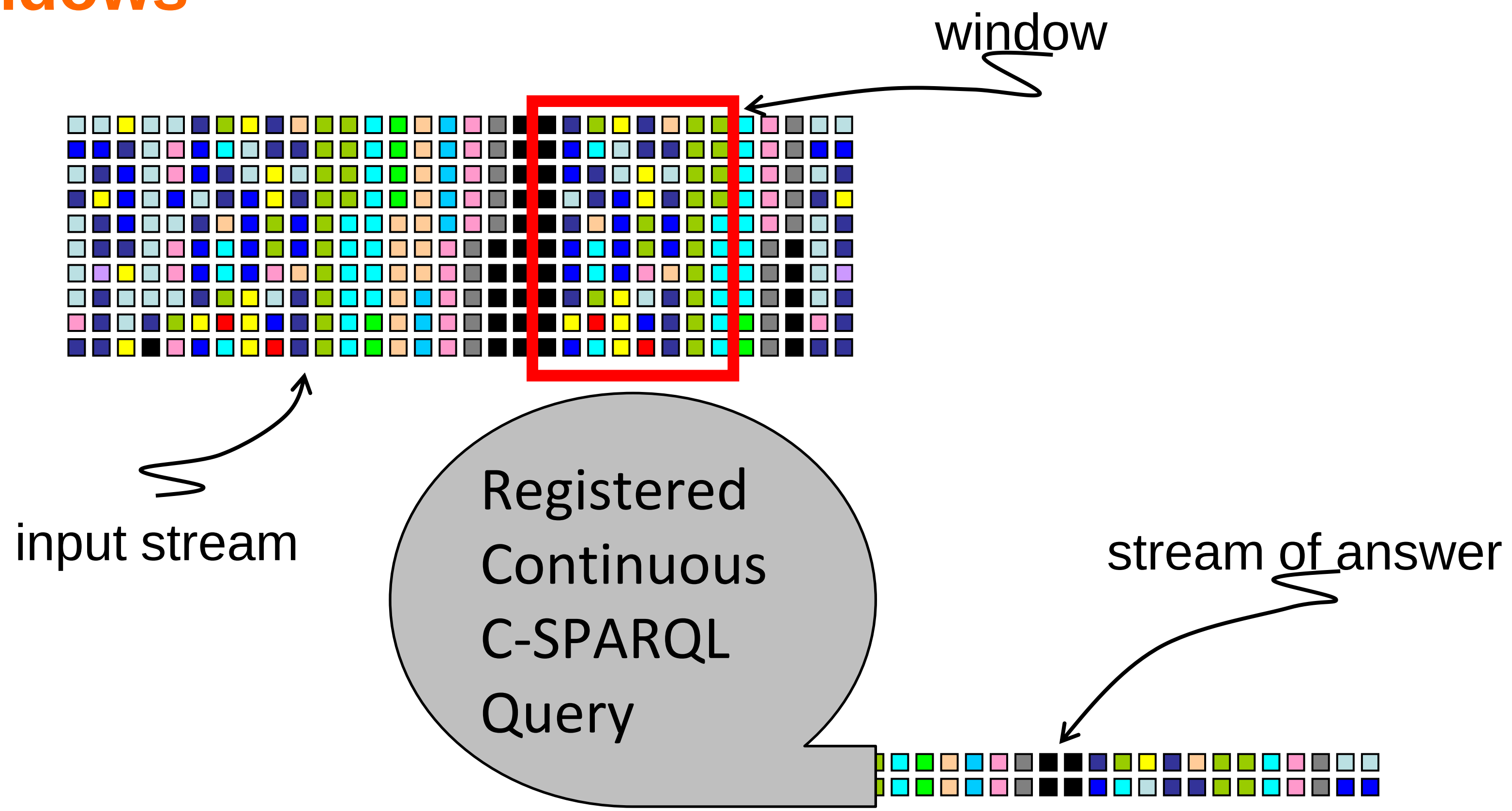
```
map:area a g2r:SpatialPropertyBridge ;
        d2rq:belongsToClassMap map:platform ;
        d2rq:property ex:hasSurface ;
        g2r:spatialColumn "area” ;
        d2rq:datatype g2r:Polygon .
```

Continuous Processing of Data Streams for the Semantic Web

- The Semantic Web comes with no standard solution for processing of continuous data flows
- Reasoning on rapidly evolving knowledge has been neglected (Belief-Revision is far to computational intensive to cope with gigantic data streams)
- **Stream Reasoning** has been investigated in the last years as an approach to reasoning on rapidly evolving information and rich background knowledge.
- A number of competing solutions are appearing
  - Streaming SPARQL
  - Time Annotated SPARQL (TA-SPARQL)
  - Continuous SPARQL (**C-SPARQL**)

Basic Idea Behind C-SPARQL

- **C-SPARQL enables continuous queries registered** over streams that are observed trough **windows**



C-SPARQL at Work

Operator Question: detect a blizzartd

```
REGISTER STREAM BlizzardDetection COMPUTE EVERY 10m AS
CONSTRUCT { ?sensor so:generatedObservation [a ...] ;
                                   so:samplingTime fn:now() .}
FROM <http://oilprod.org/weatherStations.rdf>
FROM STREAM <http://oilprod.org/weatherObs.trdf> [RANGE 3h STEP 10m]
WHERE { ?sensor so:generatedObservation [a w:SnowfallObservation] .
{ SELECT ?sensor
WHERE { ?sensor so:generatedObservation ?o1 .
            ?o1 a w:TemperatureObservation ;
                  so:observedProperty w:AirTemperature ;
                  so:result [ so:value ?temperature ] .
} GROUP BY ( ?sensor ) HAVING (AVG(?temperature)<"0.0"^^xsd:float) }
{ SELECT ?sensor
WHERE { ?sensor so:generatedObservation ?o2 .
            ?o2 a w:WindObservation ;
                  so:observedProperty w:WindSpeed ;
                  so:result [ so:value ?speed ] .
} GROUP BY ( ?sensor ) HAVING (MIN(?speed)> "40.0"^^xsd:float) } }
```

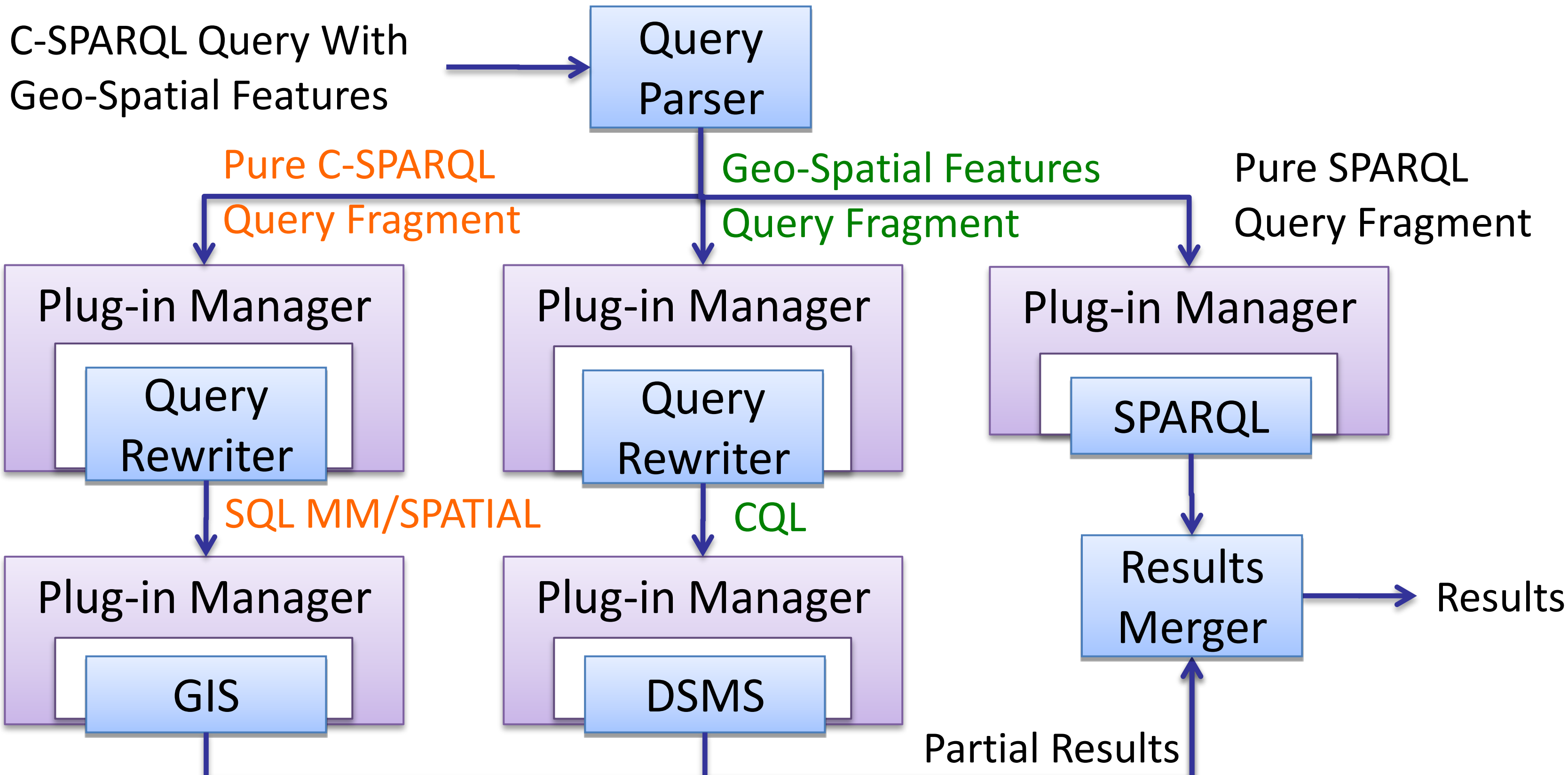
Query registration (for continuous execution)

FROM STREAM clause

WINDOW

Combining the Two Approaches with LarKC

- **LarKC** is a **pluggable platform for reasoning on massive heterogeneous information** integrating techniques from various areas



Combining the Two Approaches with LarKC

```
REGISTER STREAM BlizzardAreaDetection COMPUTE EVERY 30m AS
CONSTRUCT { [] a w:blizzard ;   ex:hasArea g2r:convexHull(?sensorPoint) . }
FROM <http://oilprod.org/weatherStations.rdf>
FROM STREAM <http://oilprod.org/BlizzardDetection.trdf>
                  [RANGE 3h STEP 30m]
WHERE {
?sensor so:generatedObservation [a w:blizzard] ; grs:point ?sensorPosition . }

REGISTER QUERY PlatformToAlertForPotentialBlizzard
   COMPUTE EVERY 30m AS
SELECT ?platform
FROM <http://oilprod.org/weatherStations.rdf>
FROM STREAM <http://oilprod.org/BlizzardAreaDetection.trdf>
                  [RANGE 3h STEP30m]
WHERE {
  ?blizzard a w:blizzard ; ex:hasArea ?blizzardArea .
  ?platform ex:hasSurface ?platformSurface .
  FILTER(g2r:overlaps(
              g2r:buffer(?blizzardArea ,"20"^^g2r:km),?platformSurface)) }
```

Conclusions

- Ongoing efforts are extending the Semantic Web standards with the ability
  - to cope with the geospatial features, e.g., **G2R**
  - to process in real time huge and possibly noisy sensor data streams, e.g., **C-SPARQL**
  - to seamlessly integrate these extensions, e.g., **LarKC**
- C-SPARQL and G2R are potentially usable in the context of oil production
- The path towards systems able to support in real-time the decision making processes of hundreds of concurrent users (e.g., the controllers on the platform and in the onshore control rooms) is still long.