Ontology-Based Integration of Streaming & Static RDBs

### Diagnostic Queries with STARQL

```
PREFIX ex: <http://www.siemens.com/onto/gasturbine/>
CREATE PULSE examplePulse WITH START = NOW, FREQUENCY = 1sec
CREATE STREAM StreamOfSensorsInCriticalMode AS
CONSTRUCT GRAPE NOW (?sensor a :InCriticalMode)
FROM STATIC ONTOLOGY ex:sensorOntology, DATA ex:sensorStaticData
WHERE (?sensor a :Reliable)
FROM STREAM sensorMeasurements [NOW - 1sec, NOW] ->> 1sec,
USING PULSE examplePulse
LIMIT 100
SEQUENCE BY StandardSequencing AS MergedSequenceOfMeasurements
HAVING EXISTS (?sensor ex:hasValue Ty ex:refSensor ex:hasValue Ty)
HAVING PearsonCorrelation(Ty, Ty) > 0.75
```

### Main Features of STARQL
- **Query language over ontologies**
  - Syntax: extension of SPARQL
    - Basic graph patterns
    - Typical mathematical, statistical, and event pattern features needed in real-time diagnostic scenarios
- **Semantics**
  - Combination of open and closed world reasoning
  - Extends snapshot semantics for window operators with sequencing semantics that can handle integrity constraints such as functionality assertions
- **Efficient query enrichment and transformation**
  - Enrichment: PTime in the size of OWL 2QL ontology
  - Unfolding: in EXASTREAM hybrid queries

### Demo Scenarios
- **Demo Description**
  - Siemens diagnostics tasks
    - e.g., calculate the Pearson correlation coefficient between turbine data streams
  - Siemens data
    - 950 turbines, 2002 – 11 years
  - Anonymised
  - Data distribution
    - From 1 to 128 nodes
    - Each node: 2 proc., 4GB RAM

### Optique Platform

- **Main features**
  - End-to-end OBDA system
    - Fully integrated
  - For IT specialists
    - Whole OBDA lifecycle
    - Flexible configuration
  - For end-users
    - Intuitive query formulation
    - Monitoring dashboards
    - Integrated with GIS systems

- **OBDA query answering**
  - Q1: Onto Query
  - Q2: Enriched Onto Query
  - Q3: DB Query

---

**Research Challenges**

- **Deployment support**
  - Semi-automatic for ontologies and mappings
- **Query language**
  - Over ontologies, streaming and static data
  - Efficient query enrichment and transformation
- **Backend**
  - To optimise large numbers of queries
  - Efficiently execute over distributed streaming and static data

---

**Ontology Based Data Access**

- **Ontology:** conceptual domain model
- **Mappings:** ontological terms to DBs

**Demo Scenarios**

- **Diagonstics with our deployment**
- **Performance showcase of our deployment**
- **Diagonstics with user’s deployment**

---

**Semantic Access to Databases**

- **Large enterprise databases**
  - Many complex different schemata
  - Siemens
    - About 100s turbines produce data
    - Life, archived streams, static RDBs
    - Data access is hard:
      - Up to 80% of analytics time

- **Ontology Based Data Access**
  - Ontology: conceptual domain model
  - Mappings: ontological terms to DBs

---

**Stream-Static Query Processing with EXASTREAM**

- **Main Features**
  - Highly optimised query processing system
  - Supported queries
    - Extension of SQL
    - Hybrid stream-static
  - High-throughput

- **User Defined Functions**
  - For complex stream processing
  - Arbitrary user code

- **Architecture**
  - Parallelism by distributing Q. processing across multiple nodes
  - Query preprocessing
    - Registered at Gateway Server
    - Passed through Parser
    - Fed into Scheduler
  - Query execution
    - Scheduler finds Worker Nodes based on their load
    - Scheduler places stream & relations on selected Workers
    - Worker Nodes execute queries