1. BI and the Semantic Web

- Business Intelligence tools need to analyze data published on the Web
- OLAP-style analysis of Linked Data may help in better decision making

2. System Architecture

- Global Conceptual Schema (GSC) – high-level view of the system (expressed in QB4OLAP)
- Source Discovery/Schema Builder – discovery of data sources and construction of the GCS
- Federated Query Processor (FQP) – retrieval, in parallel, data from several federated data sources
- Semantic Query Processor – conversion of a user query to SPARQL which is sent to the FQP

3. Processing Aggregate Queries in a Federation of SPARQL Endpoints

3.1 Data Structure

```
SELECT 3?placeId 3?regName WHERE {
  ?placeId gn:parentFeature/gn:name ?regName .
} GROUP BY ?placeId
```

3.2 Federated Query

```
SELECT 3?regName 3?value WHERE {
  ?regName rdfs:label "OLAP" .
} GROUP BY ?regName
```

3.3 Result Size Estimation

```
\text{Cost Model}
```

```
\text{C}_{\text{overall}} = \text{C}_{\text{com}} + \text{C}_{\text{comm}} + \text{C}_{\text{res}} + \text{C}_{\text{agg}} + \text{C}_{\text{result}}
```

```
\text{Cost Model}
```

```
\text{Cost Model}
```

```
\text{Cost Model}
```

4. Improving Performance of Aggregate Queries using Materialized RDF Views

4.1 Data Cube Query Example

```
SELECT 3?cityid 3?month (SUM(?total) AS ?T) WHERE {
  ?ex:OrderDate ?o ; ex:Customer ?c ;
  ?o ev:place ?place ;
} GROUP BY ?cityid ?month
```

4.2 Defining Views

```
- View query consists of 2 parts: SELECT query specifies the desired lattice node, CONSTRUCT query creates RDF triples from SELECT query results
```

4.3 Cost Model

```
\text{Cost Model}
```

```
\text{Cost Model}
```

5. Publications

Published:

- D. Ibragimov, K. Hose, T. B. Pedersen, E. Zimányi. Towards Exploratory OLAP over Linked Open Data – A Case Study. BIIRTE 2014
- D. Ibragimov, K. Hose, T. B. Pedersen, E. Zimányi. Executing Aggregate SPARQL Queries over Federated Endpoints

Future Work:

- Analyzing the Performance of Complex Aggregate SPARQL Queries with Intermediate Results Materialization
- Answering Aggregate SPARQL Queries over Materialized Views with Inferred Knowledge