QB4OLAP: A New Vocabulary for OLAP Cubes on the Semantic Web

Lorena Etcheverry[1] Alejandro A. Vaisman[2]

[1] *lorenae@fing.edu.uy*
Instituto de Computación
Universidad de la República, Uruguay

[2] *avaisman@ulb.ac.be*
Université Libre de Bruxelles, Belgium

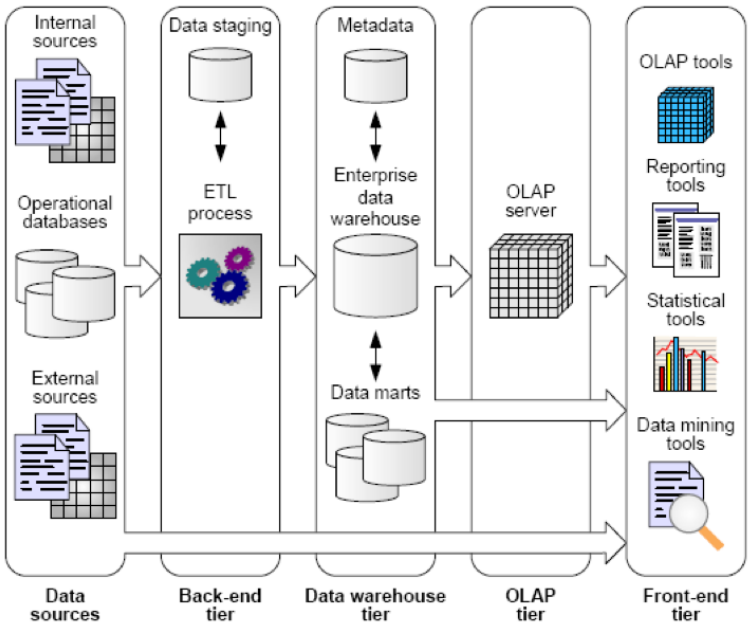
COLD 2012 - 11th International Semantic Web Conference, Boston, 2012

Outline

Outline

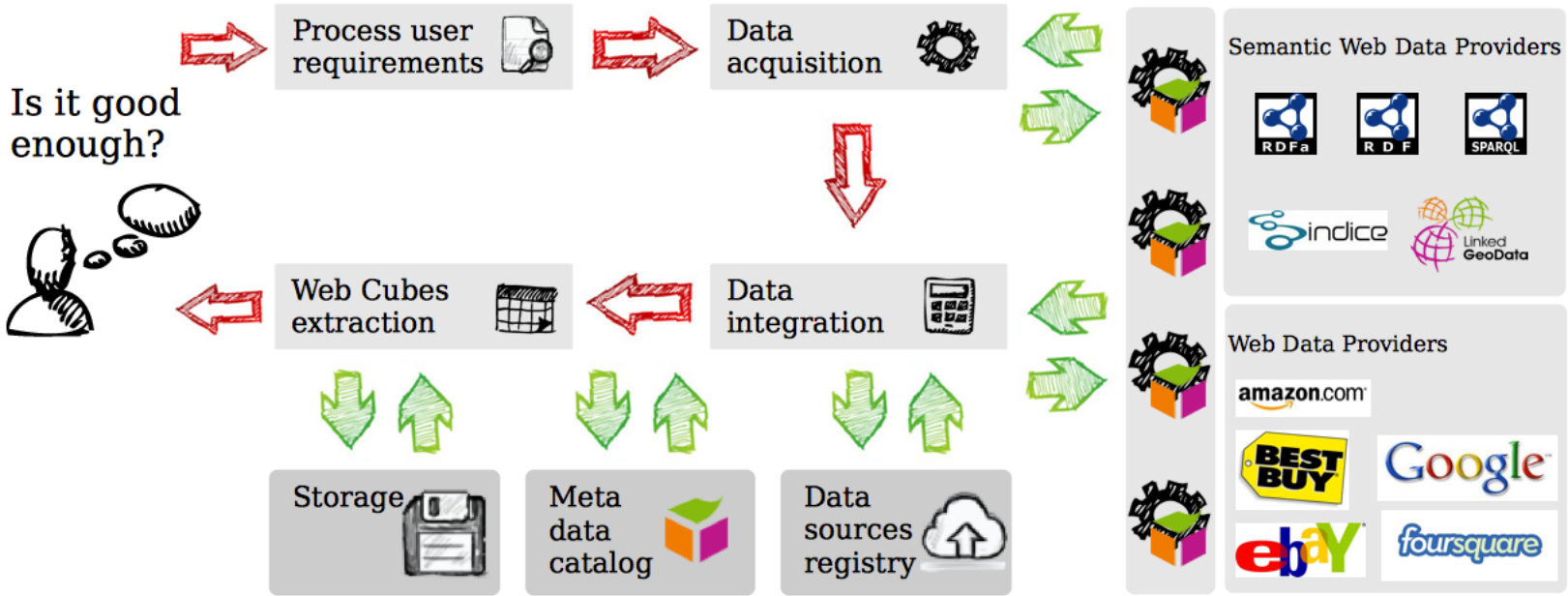
Motivation

- OLAP (On-line Analytical Processing) allows analyzing huge amounts of data for decision-making.
- Multidimensional data are seen as data cubes (DC).
- ETL (Extract,Transform, Load) process initially loads the DW; then, data is refreshed periodically
- ETL is costly and resource-consuming.



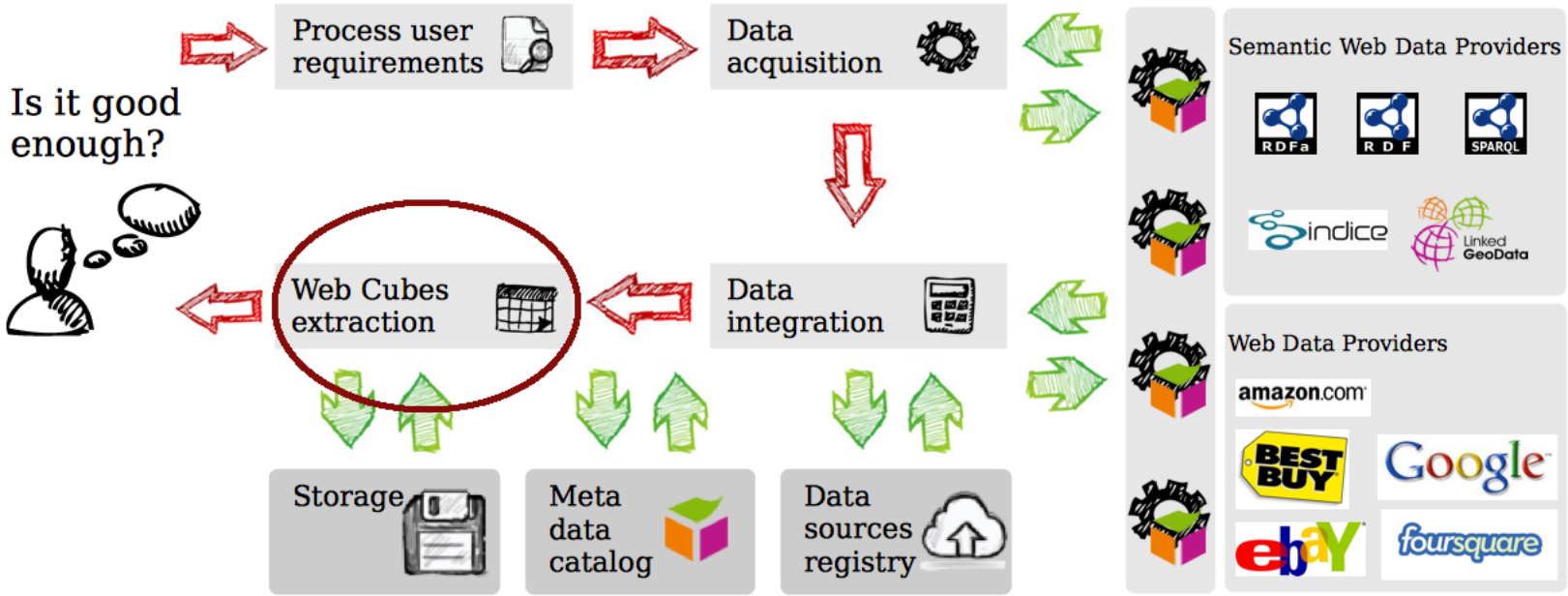
DW Architecture, Malinowski & Zimányi, 2008

A Possible Architecture



[Fusion cubes: Towards self-service business intelligence, Abelló et al. Dagstuhl seminar 2011 *"Data Warehousing: from Occasional OLAP to Real-time Business Intelligence"*]

A Possible Architecture



[Fusion cubes: Towards self-service business intelligence, Abelló et al. Dagstuhl seminar 2011 *"Data Warehousing: from Occasional OLAP to Real-time Business Intelligence"*]

The Story so Far

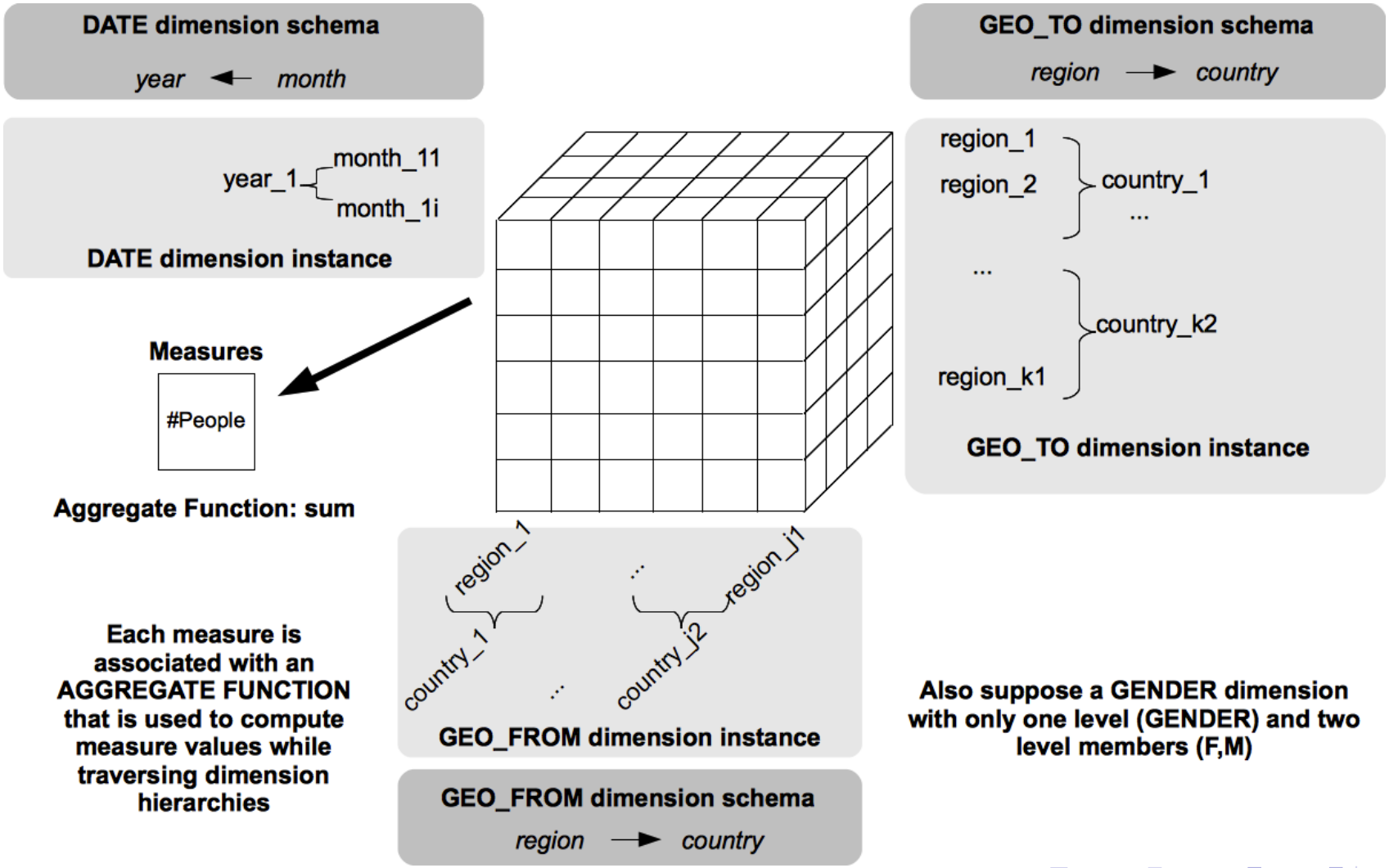
- The RDF Data Cube Vocabulary (QB) [Cyganiak et al. 2012] (W3C Working Draft) does not directly support the classical multidimensional model for OLAP.
  - Oriented to statistical data analysis.
  - Does not represent dimension structure.
  - Does not bind measures to aggregate functions.
  - Dimension hierarchies not accounted for directly.
  - **Consequence:** OLAP operators are difficult to define over QB (see Kämpgen et al., ILD, ESWC 2012).
- The OpenCube vocabulary (OC) [Etcheverry and Vaisman. ESWC 2012] fully represents OLAP model, but ...
  - no relationship is provided among concepts in OC and QB
  - **Consequence:** is not possible to use OC operators over data already published in QB. Data must be rewritten.

Contribution

- An RDF vocabulary (QB4OLAP), that fully represents the classical multidimensional model, based on OC and QB.
- A set of OLAP operators implemented as SPARQL queries.
- Algorithms that:
  - automatically build the SPARQL queries that implement OLAP operators.
  - build QB4OLAP cubes from QB cubes, allowing the **reuse** of published data without rewriting observations.

Outline

Multidimensional Model



Instance Example

Number of people that migrated between regions in Spain[1]

		2006		2007	
FROM	TO	Fem	Masc	Fem	Masc
Galicia	Com.Valenciana	–	–	678	–
Navarra	Com.Valenciana	242	266	318	427
Com.Valenciana	Galicia	780	–	930	–

OLAP Operators

		2006		2007	
FROM	TO	F	M	F	M
Galicia	C.Valenciana	–	–	678	–
Navarra	C.Valenciana	242	266	318	427
C.Valenciana	Galicia	780	–	930	–

(a) Cube C

FROM	TO	2006	2007
Galicia	C.Valenciana	–	678
Navarra	C.Valenciana	508	745
C.Valenciana	Galicia	780	930

(b) *Slice*(*C*, *Gender*)

		2006		2007	
FROM	TO	F	M	F	M
Spain	C.Valenciana	242	266	996	427
	Galicia	780	–	930	–

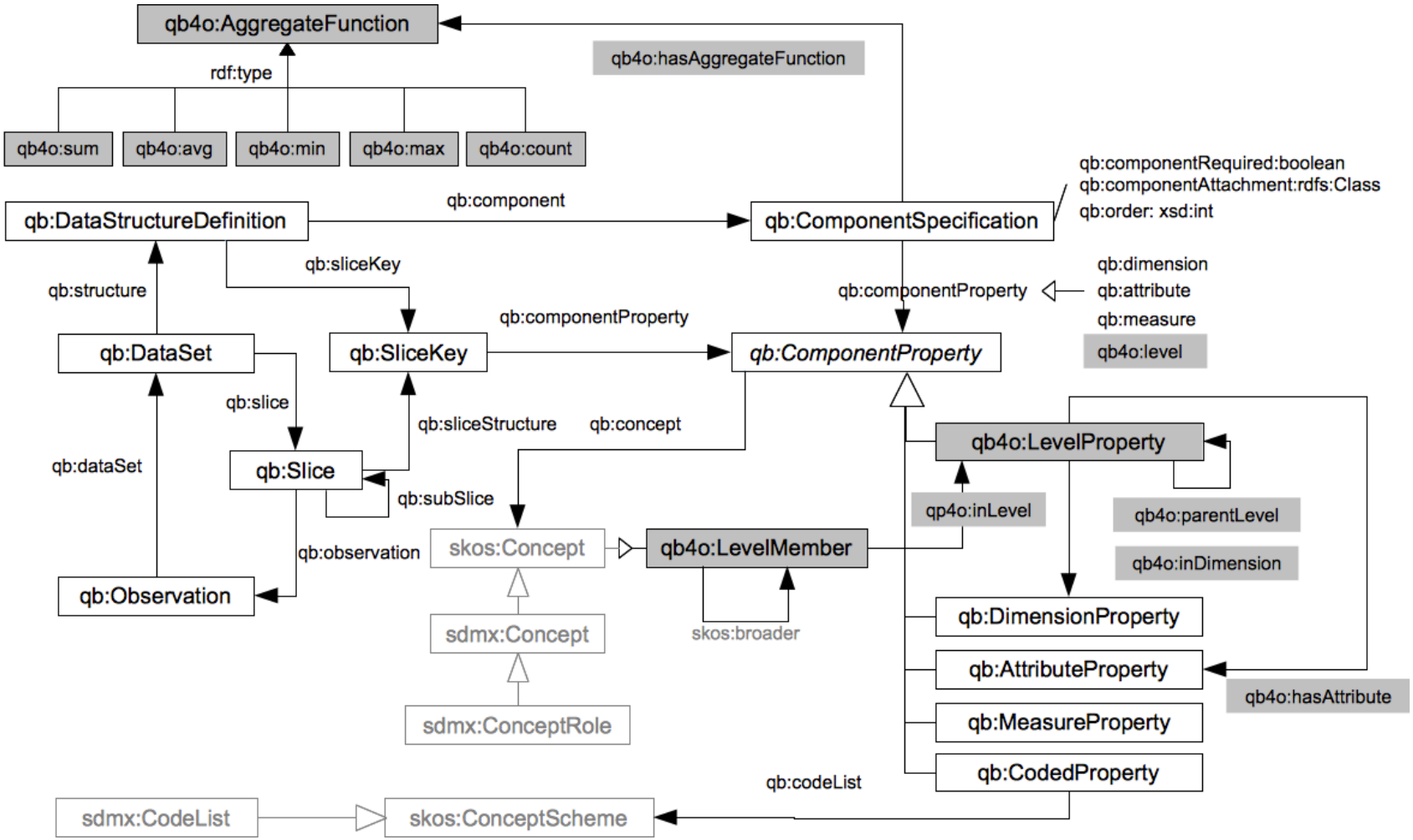
(c) *RollUp*(*C*, *From*, *country*)

		2006		2007	
FROM	TO	F	M	F	M
Galicia	C.Valenciana	–	–	678	–
C.Valenciana	Galicia	780	–	930	–

(d) *Dice*(*C*, #*people* > 500)

Outline

The QB4OLAP Vocabulary



Schema Examples

Dimension Schema

```
eg:geoFrom a qb:DimensionProperty.
eg:country a qb4o:LevelProperty;
        qb4o:inDimension eg:geoFrom.
eg:region a qb4o:LevelProperty;
        qb4o:inDimension eg:geoFrom;
        qb4o:parentLevel eg:country.
```

Fact Schema

```
eg:migr_2es a qb:DataStructureDefinition ;
    qb:component [ qb:dimension eg:date] ;
    qb:component [ qb:dimension eg:geoFrom] ;
    qb:component [ qb:dimension eg:geoTo] ;
    qb:component [ qb:dimension eg:gender] ;
    qb:component [qb:measure eg:numPeople;
        qb4o:hasAggregateFunction qb4o:sum] .

eg:dataset_migr qb:structure eg:migr_2es.
```

Instance Examples

Dimension Instance[ab]

```
geo:ES qb4o:inLevel eg:country.
geo:ES11 qb4o:inLevel eg:region;
rdf:label "Galicia";
skos:broader geo:ES .
geo:ES22 qb4o:inLevel eg:region;
rdf:label "Navarra";
skos:broader geo:ES .
geo:ES52 qb4o:inLevel eg:region;
rdf:label "Comunidad Valenciana";
skos:broader geo:ES .
```

Fact Instance

```
eg:migr_r_2es1 a qb:Observation;
  qb:dataSet eg:dataset_migr;
  eg:date 2007;
  eg:geoFrom geo:ES11;
  eg:geoTo geo:ES52;
  eg:gender sex:F;
  eg:numPeople 678.0.
```

[a] prefix geo: `http://eurostat.linked-statistics.org/dic/geo`

[b] prefix sex: `http://eurostat.`

Roll-Up Example

RollUp(Migr2es,From,country)

```
CONSTRUCT { ?id qb:dataSet eg:dataset_migrCountry. ?id eg:date ?date .
  ?id eg:geoFrom ?countryFrom . ?id eg:geoTo ?regionTo .
  ?id eg:gender ?gender . ?id eg:numPeople ?numPCountry .}
WHERE{ {
 SELECT ?date ?countryFrom ?regionTo ?gender (SUM(?numPeople) AS ?numPCountry)
  (iri(f(?date,?from,?regionTo,?gender)) AS ?id)
 WHERE {
    ?o qb:dataSet eg:dataset_migr . ?o eg:date ?date .
    ?o eg:from ?regionFrom . ?o eg:to ?regionTo . ?o eg:gender ?gender .
    ?o eg:numPeople ?numPCountry .
    ?regionFrom skos:broader ?countryFrom . ?countryFrom qb4o:inLevel eg:country
  }GROUP BY ?date ?countryFrom ?regionTo ?gender}}
```

Creating a QB4OLAP schema from a QB schema

Given a schema in QB, a schema in QB4OLAP can be built, that allows to apply OLAP operators over existent QB observations. The algorithm does not modify the set of observations.
INPUT:

- a schema in QB
- for each measure in the schema, an aggregate function
- for each dimension in the schema, a set of levels and a partial order among them (parent level)
- for each dimension level member, its correspondent dimension level and its corresponding level member in the parent level

Creating a QB4OLAP schema from a QB schema II

1: **for all** $d_i \in D_1$ (d_i a `qb:DimensionProperty`) **do**
2: Create a new dimension $d_j \in D_2$ (d_j a `qb:DimensionProperty`)
3: Add triples (d_i a `qb4o:LevelProperty`) and (d_i `qb4o:inDimension` d_j)
4: Add a triple (dsd_2 `qb:component` [`qb4o:level`d_i])
5: Obtain a hierarchy of levels hl_i and a hierarchy of level members hm_i for d_i
6: **for all** $l_i \in hl_i$ **do**
7: Add triples (l_i a `qb4o:LevelProperty`) and (l_i `qb4o:inDimension` d_j)
8: **for all** $lm_i \in hm_i$ such that lm_i belongs to level l_i **do**
9: Add a triple (lm_i `qb4o:inLevel` l_i).
10: **end for**
11: **end for**
12: **for all** $(l_j, l_k) \in hl_i$ such that $l_j \rightarrow l_k$ **do**
13: Add a triple (l_j `qb:parentLevel` l_k)
14: **end for**
15: **end for**
16: **for all** m_i such that (dsd_1 `qb:component` [`qb:measure` m_i]) **do**
17: Add a triple (dsd_2 `qb:component` [`qb:measure` m_i;`qb:hasAggregateFunction` ag_i])
18: **end for**

COMPLEXITY: the main source of complexity is the size of the set of dimension level members, which is usually small, compared to the set of observations (facts).

Outline

Conclusion

- An RDF vocabulary for representing the classical multidimensional model s.t.
  - ANSI architecture is supported (conceptual, logical and physical levels clearly identified).
  - OLAP applications and operators can be implemented naturally and easily maintained and extended.
- A set of OLAP operators implemented as SPARQL queries.
- Algorithms that automatically build the SPARQL queries that implement such OLAP operators.
- Preliminary tests over proof-of-concept prototype.
- More info: `https://code.google.com/p/publishing-multidimensional-data/`

Future Work

- Extend the operator set (e.g., Drill-Across).
- Perform stress tests with real data
- Query processing and optimization.
- Incorporate all of these into the general framework.

Thanks for your attention.
Questions?
Contact:

- Lorena Etcheverry *lorenae@fing.edu.uy*
- Alejandro A. Vaisman *avaisman@ulb.ac.be*

References I


A. Abelló, J. Darmont, L. Etcheverry, M. Golfarelli, J.-N. Mazón, F. Naumann, T. B. Pedersen, S. Rizzi, J. Trujillo, P. Vassiliadis, and G. Vossen.
Fusion cubes: Towards self-service business intelligence.
Submitted to IJDWM, 2012.


R. Cyganiak and D. Reynolds.
The RDF Data Cube Vocabulary, March 2012.


L. Etcheverry and A. A. Vaisman.
Enhancing OLAP Analysis with Web Cubes.
In *ESWC 2012*, pages 53–62, 2012.

References II


S. Gómez, L. Gómez, and A. A. Vaisman.
A Generic Data Model and Query Language for Spatiotemporal OLAP Cube Analysis.
In *EDBT 2012*, 2012.


C. A. Hurtado, A. O. Mendelzon, and A. A. Vaisman.
Maintaining Data Cubes under Dimension Updates.
In ICDE '99, pages 346–355, Washington, DC, USA, 1999. IEEE Computer Society.


B. Kämpgen and A. Harth.
Transforming statistical linked data for use in OLAP systems.
In I-Semantics '11, pages 33–40, New York, NY, USA, 2011. ACM.

References III


B. Kämpgen, S. O'Riain, and A. Harth.
Interacting with Statistical Linked Data via OLAP Operations.
In *Proceedings of the International Workshop on Interacting with Linked Data (ESWC)*. CEUR-WS.org, 2012.