Aggregating Linked Data

Tomáš Knap
XML and Web Engineering Research Group (XRG)
Department of Software Engineering
Charles University, Prague, Czech Republic

21.6.2012

Outline

Motivation

Linked Data Framework

Data Aggregation

Linked Data

Set of best practises for publishing structured data on the Web, Tim Berners-Lee presented four principles:

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
- Include links to other URIs. so that they can discover more things.

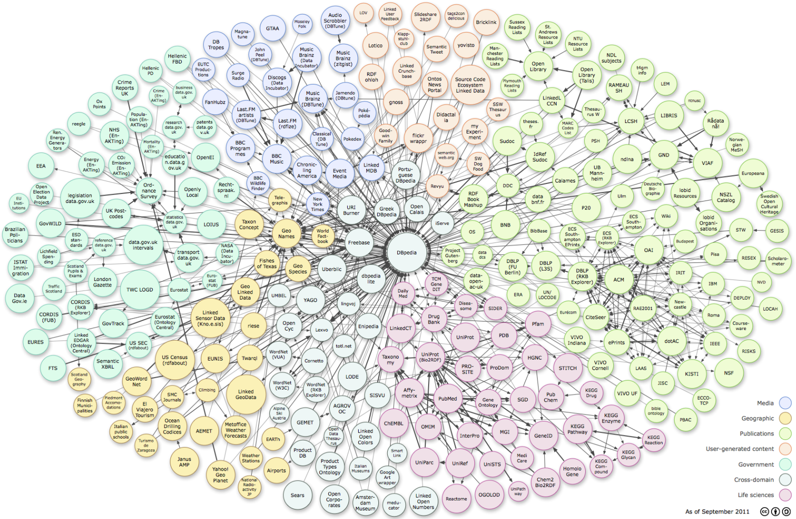
See: `http://www.w3.org/DesignIssues/LinkedData.html`

RDF

- Sample RDF statement (triple):
  - (http://dbpedia.org/resource/Berlin http://dbpedia.org/ontology/populationTotal ”3450889”)
  - (http://dbpedia.org/resource/Berlin http://www.w3.org/2002/07/owlsameAs http://rdf.freebase.com/ns/en.berlin)
- RDF data are represented as typed statements – *triples* $(s, p, o) \in U^3$ – consisting of a *subject s*, a *predicate (property) p* and an *object (value) o*.
  - U = all possible nodes, URI resources or literals (optionally typed)
  - The RDF data model can be viewed as a directed graph where edges, labeled with a predicate, lead from a subject to an object.
- A triple may be part of a *named graph* – a set of triples identified by an URI
  - Triples can be then extended to *quads* $(s, p, o, g) \in Q$ where $g \in G$ is the named graph (its URI) to which the data belongs

.

Linked Data Cloud

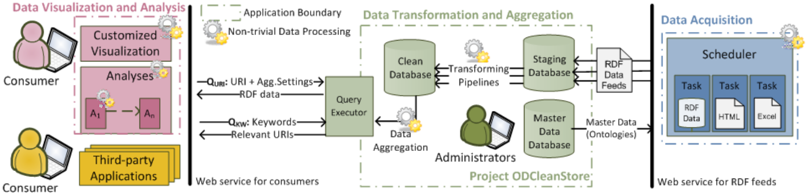


Obrázek: Linked Data Cloud

Outline

Linked Data Framework

- Data acquisition
- Data transformation and aggregation
- Data visualization and analysis



Obrázek: Linked Data Aggregation Framework

ODCleanStore - Core concepts

- Staging database
  - Incoming data inserted via web service
- Pipelines
  - Applied to the incoming data based on the identifier or the extraction feed
  - Transformers
    - Cleaners
    - Linkers
    - Quality assessment
    - Custom transformers
- Clean database
  - Transformed data are inserted to the clean database
- Data aggregation on top of the clean database
  - Design time vs. query time data aggregation

Outline

Motivational Scenario

- Suppose we have in the clean database data about the German city Berlin coming from multiple sources – DBpedia, GeoNames, and Freebase
  - `http://dbpedia.org/resource/Berlin`
  - `http://sws.geonames.org/2950159/`
  - `http://rdf.freebase.com/ns/en.berlin`

  .
- Consumer would like to get data about the resource `http://dbpedia.org/resource/Berlin`
- Tasks:
  - Discover and follow `owl:sameAs` links between resources representing the same concepts
  - Discover that meaning of the predicates `geo:lat` and `fb:location.geocode.latitude` is the same
  - Compute average value for the values of the properties `geo:long` and `geo:lat`
  - Select the best value (with the highest aggregate quality) for `rdfs:label`
  - Select the maximum (latest) value from the values of the property `dbpedia:populationTotal`

Data Aggregation - Basics

- Schema mapping
  - Enabled by proper mappings between ontologies in the master data database
- Duplicate detection
  - Enabled by proper linker
- Data fusion
  - Instance level conflicts (data conflicts)

J. Bleiholder and F. Naumann. Data fusion. ACM Comput. Surv., 2009.

Definitions - Conflicts

- Conflicting quads
  - Suppose $g_1, g_2 \in G$; quads (s, p, o_1, g_1) and (s, p, o_2, g_2) are called *conflicting quads* if $o_1 \neq o_2$
- Duplicate quads
  - Suppose $g_1, g_2 \in G$; quads (s, p, o_1, g_1) and (s, p, o_2, g_2) are called *duplicate quads* if $o_1 = o_2$

Conflict Handling Strategies

- Conflict resolution
  - Data conflicts are resolved according to the set of conflict resolution policies
- Conflict ignorance
  - Data conflicts are tolerated
  - Fusing, Non-fusing
- Conflict avoidance
  - If the data conflict occurs, all the conflicting quads are removed from the aggregated view

Data Fusion Algorithm - Inputs/Outputs

- Inputs:
  - A collection of quads from the clean database to be fused – the quads (x,*,*,*),(*,*,x,*), where *x* is the URI in the consumer's query
  - Data fusion settings – selected conflict handling strategy and set of selected conflict resolution policies (for conflict resolution strategy) - global or per property
  - `owl:sameAs` links between URI resources occurring in the quads (result of deduplication and schema mapping part of the aggregation - linkers, master data database mappings)
  - Quality scores for named graphs of the quads.
- Outputs:
  - Collection of aggregated triples enriched with the aggregate quality and source named graphs for each quad.

Quality Assessment - Obtaining Quality Scores for Named Graphs

- The quality assessment (QA) transformer checks whether the processed named graph g (feed) satisfies the set of custom *QA policies*
  - sample policies are: "Property x has an object (value) satisfying the regular expression y", "Property z exists".
- *quality score* $s(g)$ of the graph (feed) g, $s : G \rightarrow [0, 1]$
  - Based on the successful application of QA policies to the named graph g and based on the successful application of QA policies to other named graphs published by the same data source (e.g. DBpedia)
  - `www.ksi.mff.cuni.cz/~knap/files/method.pdf`

Phase 1 of Data Fusion Algorithm - Overview

Step 1.1) Replace URIs of resources representing the same entity (i.e. connected by the `owl:sameAs` links) with a single URI. Prefer URI in the consumer's query.

Step 1.2) Remove duplicate quads.

Step 1.3) Group quads to sets of conflicting quads.

Phase 1 of Data Fusion Algorithm - Detailed

Algorithm 1 Phase 1

```
Create graph H = (V, E) from the given owl:sameAs links
   between deduplicated (linked) resources; edges E are the
   owl:sameAs predicates, vertices V ⊂ U the URI resources they
   are connecting.
Find the set of weakly connected components 𝒞 in H.
For each connected component C ∈ 𝒞 do
   Choose uri(C), a single URI from the component, preferring
   URIs given in the consumer's query.
end for
For each input quad (s, p, o, g) do
   Replace s with uri(C_s), s ∈ C_s, C_s ∈ 𝒞. Do the same with
   predicate p and object o.
end for
Remove duplicate quads that might have appeared.
Group quads into sets of conflicting quads Q_{s,p}, i.e. having the
   same subject s and predicate p. (skipped for non-fusing conflict
   ignorance strategy)
```

Phase 2 of Data Fusion Algorithm - Overview

- Phase 2 of the algorithm is applied to each set of *conflicting quads* $Q_{s,p} = \{q_1, \ldots, q_n\}$, $q_i = (s, p, o_i, g_i)$, $i \in \{1, \ldots, n\}$;
  - Context of Phase 2 is given by the collection of objects o_i; subject s and predicate p are constant for one execution of Phase 2
- Let us denote $V = (v_1, \ldots, v_n)$ the collection of object values $v_i = o_i$ from the named graph g_i.

- **Conflict resolution strategy**

  Step 2.1) Choose and apply a conflict resolution policy
  Step 2.2) Compute aggregate quality for the conflict resolved quads
  Step 2.3) Create resulting aggregated triples enriched with the aggregate quality and sources of each triple.
- **Conflict ignorance strategy**

  Step 2.2) Compute aggregate quality for all conflicting quads
  Step 2.3) Create resulting aggregated triples enriched with the aggregate quality and sources of each triple.
- **Conflict avoidance strategy**

  Step 2.0) If the set of conflicting quads, $Q_{s,p}$, contains at least two conflicting quads, Phase 2 ends.
  Step 2.2) Compute aggregate quality for the single quad
  Step 2.3) Create resulting aggregated triple enriched with the aggregate quality and sources of the triple.

Further, Phase 2 is described only for conflict resolution strategy.

Application of a Conflict Resolution Policy (Step 2.1)

- In Step 2.1, a set of conflicting quads $Q_{s,p}$ is fused by the application of a *conflict resolution policy* defined for the predicate p in the data fusion settings.
- Conflict Resolution Policies:
  - Deciding - selects one or more values
    - ANY,MIN,MAX,SHORTEST,LONGEST – an arbitrary value, minimum, maximum, shortest, or longest is selected from the conflicting values V
    - BEST – the value with the highest aggregate quality is selected
    - LATEST – the value with the newest time is selected
  - Mediating - computes new values
    - AVG, MEDIAN, CONCAT – computes the average, median, or concatenation of conflicting values
- Let us introduce the set A holding such selected or computed values.

Computation of the Aggregate Quality (Step 2.2)

- The goal: compute the aggregate quality of values $v \in A$, denoted $q(v)$
- Three factors of the aggregate quality computation:
  - Quality scores $s(g_i)$ of the source named graphs g_i
  - Size of $agree(v)$, $agree(v) = \{g_i \mid v_i = v\}$, set of graphs that agree on a value $v \in V$
  - Difference between value v and other (conflicting) values from V.

Formula q_1 – Scores of the sources

- First, we calculate aggregate quality $q_1(v)$ based on the quality scores of the sources.
- A value $v \in A$ may
  - (a) be calculated from all the sources (in case of conflict resolution policies AVG, MEDIAN, CONCAT)
  - (b) come from named graphs containing a quad (s, p, v, g_i) (in case of other conflict resolution policies)

$$q_1(v) = \begin{cases} \mathrm{avg}\,\{s(g) \mid g \in \{g_1, \ldots, g_n\}\} & \text{(a)} \\ \max\,\{s(g) \mid g \in agree(v)\} & \text{(b)} \end{cases}$$

Formula q_2 – Conflicting values

- In the second step, we compute aggregate quality $q_2(v)$ based on $q_1(v)$ and differences of conflicting values V.
- we use a metric $d : U \times U \to [0, 1]$ satisfying $d(v, v) = 0$.
  - $d(x, y) = |(x - y) \; / \; \mathrm{avg}(x, y)|$ in case of numeric literals
  - normalized Levenshtein distance in case of string literals
  - $d(x, y) = 1$, where $x \neq y$, for URI resources and nodes of a different type
- If there are conflicting values different from v, the aggregate quality of v is reduced increasingly with the value of metric d and the score of the source of the conflicting value:

$$q_2(v) = q_1(v) \cdot \left(1 - \frac{\sum_{i=1}^{n} s(g_i) d(v, v_i)}{\sum_{i=1}^{n} s(g_i)}\right)$$

- Consumer can set a parameter called *multivalue* in the data fusion settings which instructs the data fusion algorithm to use $q_2(v) \equiv q_1(v)$ instead.

Formula q_3 – Confirmation by multiple sources

- Intuitively, if multiple different sources agree on a single value, we should trust this value more than each of the sources individually. We reflect this in the final phase of aggregate quality computation $q_3(v)$ ($C \in \mathbb{N}$ is a constant):

$$q_3(v) = q_2(v) + \\ + (1 - q_2(v)) \cdot \min\left(\frac{-q_1(v) + \sum_{g \in agree(v)} s(g)}{C}, 1\right)$$

Final Formula q

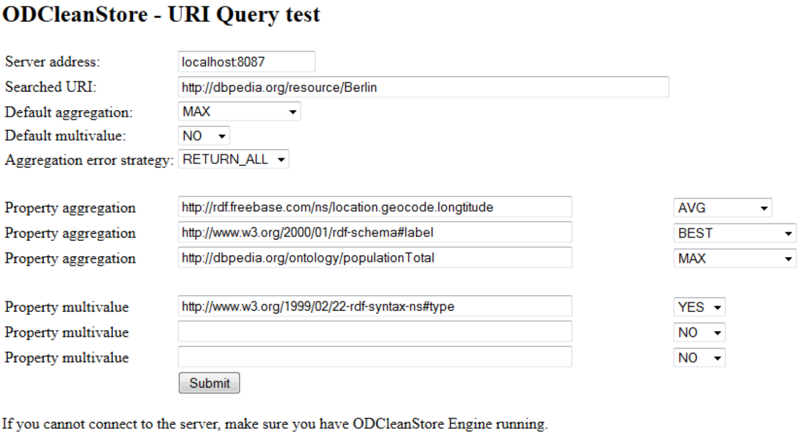
The aggregate quality $q(v)$ is computed as:

- $q(v) \equiv q_1(v)$ for conflict resolution policy CONCAT
- $q(v) \equiv q_2(v)$ if the selected mediated value v is not equal to some v_i
- $q(v) = q_3(v)$, in other cases

$q(v) = q_3(v)$ satisfies the following constraints:

- If there is a named graph g asserting a non-conflicting value v, the aggregate quality (based just on the value v) should be at least $s(g)$.
- $q(v)$ is increasing with quality scores of source named graphs v was selected from or calculated from.
- $q(v)$ is decreasing with difference of other values $v_i \in V$, taking their quality scores $s(g_i)$ into consideration.
- If multiple sources agree on the same value, the aggregate quality is increased.

Applying Algorithm to Motivational Scenario - DEMO



ODCleanStore - URI Query test

Server address:	localhost:8087	
Searched URI:	http://dbpedia.org/resource/Berlin	
Default aggregation:	MAX	
Default multivalue:	NO	
Aggregation error strategy:	RETURN_ALL	
Property aggregation	http://rdf.freebase.com/ns/location.geocode.longtitude	AVG
Property aggregation	http://www.w3.org/2000/01/rdf-schema#label	BEST
Property aggregation	http://dbpedia.org/ontology/populationTotal	MAX
Property multivalue	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	YES
Property multivalue		NO
Property multivalue		NO
	Submit	

If you cannot connect to the server, make sure you have ODCleanStore Engine running.

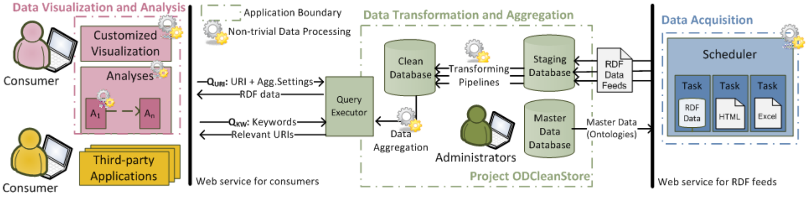
Obrázek: Demo

Experiments

- Measuring completeness, conciseness, and consistency of the original data sets and the data set created by aggregating the data
  - `www.ksi.mff.cuni.cz/~knap/files/aggregation.pdf`
- Execution times needed to accomplish various conflict resolution policies
  - `www.ksi.mff.cuni.cz/~knap/files/method.pdf`

Conclusions

- Linked Data Framework
  - Data Aggregation - Data Fusion



Obrázek: Linked Data Aggregation Framework

Thank You!