Warehousing Web Data

Ioana Manolescu-Goujot

Leo team
INRIA Saclay–Île-de-France & LRI, Université de Paris Sud-11

DESWeb workshop, April 11, 2011

Supported by: ANR-08-DEFIS-004 CODEX, INRIA ADT ViP2P & ViRDF
Overview

Work and results

Two very concrete efforts, with connections to the Semantic Web:

1. P2P warehousing of Web content
   - ViP2P (Views in Peer-to-Peer) platform for sharing XML and RDF
   - Focus on scalability and content dissemination and re-use

2. Large RDF databases
   - RDFViewS (RDF View Selection) model and algorithms
   - Focus on decoupling the storage from the logical RDF model

Some perspective and applications

Linked open public data
Part I

P2P warehousing of Web content
Distributed data management: old goal (1970)
Distributed data management: old goal (1970)

- distributed versions of industrial-strength DBMSs
- map/reduce style systems for massively parallel computations
Distributed data management: old goal (1970)
- distributed versions of industrial-strength DBMSs
- map/reduce style systems for massively parallel computations

Still missing: the flexible federation
- high independence of the sites: when to be in, what to store
- data distribution transparency
- . . . with the usual performance requirements
Distributed warehouses of Web content

Web content
structured documents, schemas, annotations, concepts, mappings, Web services, inter-document links

Web content warehouse
Distributed database of selected content, whose users may:

- publish / update
- query resources (subscription + snapshot mode)
- connect (annotate, map, link...) existing resources
- re-publish derived resources
INRIA annual activity report in XML

https://irabot.inrialpes.fr/RA2010/leo.xml

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE raweb PUBLIC "raweb2.dtd">
<raweb year="2009">
  <identification id="leo" isproject="true">
    <shortname>leo</shortname>
    <domaine>Perception, Cognition, Interaction</domaine>
    <UR name="Saclay"/>
  </identification>
  <team>
    <person>
      <web>http://www-roc.inria.fr/~manolesc</web>
      <firstname>Ioana</firstname>
      <lastname>Manolescu</lastname>
    </person>...
  </team>...
</raweb>
(Very) popular data exchange format
Many W3C standards + efficient tools
XML and the Semantic Web

XML

```xml
<raweb>
  <identification>
    <@id>gemo</@id>
    <@isproject>true</@isproject>
    <domaine>Perception, Cognition, Interaction</domaine>
    <UR>Saclay</UR>
  </identification>
  <person>
    <web>www-roc.inria.fr/~manolesc</web>
    <firstname>Ioana</firstname>
    <lastname>Manolescu</lastname>
  </person>
</raweb>
```

RDF

```xml
<http://leo.saclay.inria.fr>
  <wroteReport>
    https://irabot.inrialpes.fr/RA2010/leo.xml
  </wroteReport>
  <coordinates>
    http://codex.saclay.inria.fr
  </coordinates>
  <hasType>
    http://www.inria.fr/types.xsd#ContratFR
  </hasType>
</http://leo.saclay.inria.fr>
```
XML and the Semantic Web

Possible queries

- European projects involving at least two INRIA teams?
- Most frequent keywords for each INRIA theme?
- PhD students co-supervised with a foreign University?
Distributed warehouses of Web content

Distributed XML

https://irabot.inrialpes.fr/RA2010/leo.xml

domaine
Perception, Cognition, Interaction

@id
@gemo
team
name
web
firstname
lastname

ioana
Loana
Manolescu

http://leo.saclay.inria.fr
http://codex.saclay.inria.fr

Distributed RDF

INRIA Tech. Info. Serv.

http://François Bancilhon

collaboratesWith

https://leo.saclay.inria.fr

organizes

http://desweb2011.unimore.it

Data Publica

http://leo.saclay.inria.fr
possibleCollaborator

http://data-publica.com/
Motivation

Distributed warehouses of Web content

Distributed XML

https://irabot.inrialpes.fr/RA2010/leo.xml

raweb

identification

@id @isproject

gemo true

Perception, Cognition, Interaction

domaine

UR

team

person

www-roc.inria.fr/~manolesc

Ioana Manolescu

INRIA http://leo.saclay.inria.fr

Tech. Info. Serv.

collaboratesWith

François Bancilhon

http://disi.unitn.it/~velgias

organizes

http://desweb2011.unimore.it

Data Publica

http://data-publica.com/

possibleCollaborator

uni. Trento researchers involved in IEEE workshop organization?

Possible collaborators for a public data LOD project?
Distributed hash tables
Distributed hash tables
Distributed hash tables
Distributed hash tables

\[ \text{put}(k_1, v_1) \]
Distributed hash tables

$\text{put}(k_1, v_1)$

$(k_1, v_1)$
Distributed hash tables

\[
\begin{align*}
(k_1, v_1) \\
(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8)
\end{align*}
\]

\[
\begin{align*}
\text{put}(k_1, v_1) \\
\text{put}(k_1, v_2)
\end{align*}
\]
Distributed hash tables

\[(k_1, \{v_1, v_2\})\]

\[
\begin{align*}
&\text{put}(k_1, v_1) \\
&\text{put}(k_1, v_2)
\end{align*}
\]
Distributed hash tables

\[
\begin{aligned}
&\text{put}(k_1, v_1) \\
&\text{put}(k_1, v_2) \\
&\text{get}(k_1) \\
&(k_1, \{v_1, v_2\})
\end{aligned}
\]
Distributed hash tables

\[ (k_1, \{v_1, v_2\}) \]

\[ \text{put}(k_1, v_1) \]

\[ \text{put}(k_1, v_2) \]

\[ \text{get}(k_1) \]
Distributed hash tables

\[ \text{put}(k_1, v_1) \rightarrow p_8 \]

\[ \text{put}(k_1, v_2) \rightarrow p_6 \]

\[ \text{get}(k_1) \rightarrow p_1 \]

\[ \{v_1, v_2\} \rightarrow p_1 \]

\[ (k_1, \{v_1, v_2\}) \rightarrow p_3 \]
DHTs: what they provide and what is missing

Provided: dynamic peer networks

- each peer is assigned an id $\Rightarrow$ re-adjustable address range
- bound of $\log_2(N)$ hops to route a message to peer
DHTs: what they provide and what is missing

Provided: dynamic peer networks

- each peer is assigned an id $\Rightarrow$ re-adjustable address range
- bound of $\log_2(N)$ hops to route a message to peer

(Key, value) stores = basis for content sharing

- index the resources by keys
- look up resources by keys
DHTs: what they provide and what is missing

Provided: dynamic peer networks
- each peer is assigned an id ⇒ re-adjustable address range
- bound of $\log_2(N)$ hops to route a message to peer

(Key, value) stores = basis for content sharing
- index the resources by keys
- look up resources by keys

Functionalities to add
- data storage and indexing
- query processing within / across distributed peers (query optimization, . . .)
Trade-offs in DHT indexing and query processing

Level of detail of resource indexing
- lookup precision $\uparrow \Rightarrow$ execution time $\downarrow$
- data publication time $\uparrow$, possibly execution time $\uparrow$

If data is re-placed in the P2P network
- fewer peers contacted for a query (message no. $\downarrow$, execution time $?$)
- data transfers in the absence of queries (message no. $\uparrow$, total message size $\uparrow$)
ViP2P is joint work with

**Students**

- Konstantinos Karanasos, Asterios Katsifodimos, Spyros Zoupanos (PhD 2009)
- Martin Goodfellow (U. Strathclyde), Silviu Julean (UVT, Romania), Varunesh Mishra (IIT Kanpur), Alexandra Roatis (UVT, Romania), Mimma Sileo (U. Basilicata, Italy)

**Engineers**

Jesús Camacho-Rodriguez, Julien Leblay, Alin Tilea

**Friends from abroad**

Angela Bonifati (CNR Italy), Vasilis Vassalos (AUEB, Greece)

http://vip2p.saclay.inria.fr
Web data sharing in ViP2P

Web data
- XML documents
- RDF annotations
Web data sharing in ViP2P

Web data
- XML documents
- RDF annotations

Architecture [MZ09]
- peers retain control over the data they store/publish
  - documents published independently
  - triples which may connect content (annotations)
- data is re-distributed: peers accumulate results of long-running queries
Web data sharing in ViP2P

Web data
- XML documents
- RDF annotations

Architecture [MZ09]
- peers retain control over the data they store/publish
  - documents published independently
  - triples which may connect content (annotations)
- data is re-distributed: peers accumulate results of long-running queries
- peers share their accumulated results with others

long running queries ⇔ materialized views
Publishing and querying in ViP2P
Publishing and querying in ViP2P

The peers may store:
- documents,
- annotations
The peers may store:

- documents, annotations
- views
Publishing and querying in ViP2P

When $q$ arrives:

- view definition
- lookup
Publishing and querying in ViP2P

When \( q \) arrives:
- view definition
- lookup
- rewriting
Publishing and querying in ViP2P

When $q$ arrives:
- view definition
- lookup
- rewriting
- execution of physical plan
Publishing and querying in ViP2P

When $d$ arrives:

- Search view definitions for which $v_i(d) \neq \emptyset$.
- Compute $v_i(d)$.
- Send results.
When $d$ arrives:
- search view definitions for which $v_i(d) \neq \emptyset$
Publishing and querying in ViP2P

When \( d \) arrives:

- search view definitions for which \( v_i(d) \neq \emptyset \)
- compute \( v_i(d) \)
Publishing and querying in ViP2P

When $d$ arrives:

- search view definitions for which $v_i(d) \neq \emptyset$
- compute $v_i(d)$
- send results
Distributed materialized views in ViP2P
Distributed materialized views in ViP2P

Long-running query = subscription = view

View results may be found in

- a published document
- a published RDF dataset
- several documents and/or RDF data sets, published independently over a period of time
Distributed materialized views in ViP2P

Long-running query = subscription = view

View results may be found in

- a published document
- a published RDF dataset
- several documents and/or RDF data sets, published independently over a period of time
Distributed materialized views in ViP2P

Long-running query = subscription = view

View results may be found in
- a published document
- a published RDF dataset
- several documents and/or RDF data sets, published independently over a period of time

Data publication = adding to the interested views
Distributed materialized views in ViP2P

Long-running query = subscription = view

View results may be found in
- a published document
- a published RDF dataset
- several documents and/or RDF data sets, published independently over a period of time

Data publication = adding to the interested views

Query planning = rewriting queries using views

1. Find relevant views ⇒ rewrite the query (find a plan combining views and producing the same results as the query) [MKVZ11]
2. Optimize the rewriting plan
Sample views

All article abstracts

<table>
<thead>
<tr>
<th>URI</th>
<th>abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://peer1.com/doc1.xml">http://peer1.com/doc1.xml</a></td>
<td>Databases...</td>
</tr>
<tr>
<td><a href="http://peer2.com/f2.xml">http://peer2.com/f2.xml</a></td>
<td>XML processing...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Titles and abstracts of all articles

- article
  - abstract
  - title

<table>
<thead>
<tr>
<th>URI</th>
<th>abstract</th>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Sample views

Titles of all articles having “P2P” in the abstract

```
<table>
<thead>
<tr>
<th>URI</th>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
```
Sample views

URIs of all documents containing articles having “P2P” in the abstract
Sample views

IDs and authors of all articles having “P2P” in the abstract

<table>
<thead>
<tr>
<th>URI</th>
<th>article.ID</th>
<th>author.val</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

ViP2P views
Sample views

Homepages of all workshop organizers

workshop
  organizers
    person
    lab
homepage\textsubscript{cont}

<table>
<thead>
<tr>
<th>URI1</th>
<th>URI2</th>
<th>homepage\textsubscript{cont}</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Ioana Manolescu (Leo, Saclay)
AnnoViP [KZ10]: sharing RDF annotations of XML documents

Sample view over XML and RDF data

ViP2P views

Ioana Manolescu (Leo, Saclay)
How does the document publisher find the view?
Indexing views for view materialization (1)

How does the document publisher find the view?

Case 1: the view is defined before the document is published.
Indexing views for view materialization (1)

How does the document publisher find the view?

Case 1: the view is defined before the document is published.

View indexing

For each view $v$

- For each node label $l$ in $v$, put $(l, v)$
- For each constant $k$ in $v$, put $(k, v)$
Indexing views for view materialization (1)

How does the document publisher find the view?

Case 1: the view is defined before the document is published.

View indexing

For each view $v$

- For each node label $l$ in $v$, put $(l, v)$
- For each constant $k$ in $v$, put $(k, v)$

View lookup

1. For each document $d$ and node label $l$ in $d$, get($l$)
2. Evaluate the (filtered) view set on $d$
How does the document publisher find the view?
How does the document publisher find the view?
Case 2: the view is defined after the document is published
How does the document publisher find the view?
Case 2: the view is defined after the document is published

Document publishers periodically look up with all their search terms (messages ➔ ➔)!
Indexing views for view materialization (2)

How does the document publisher find the view?

Case 2: the view is defined after the document is published

1. Document publishers periodically look up with all their search terms (messages \( \rightarrow \rightarrow \))!

2. Better: divide the history into view publication intervals. Peers look up last-interval views.

\[
\begin{align*}
&d_1 & v_1 & v_2 & v_3 & d_2 \\
&\quad \quad \bullet \quad \bullet \quad \bullet \quad \bullet \quad \bullet \quad \bullet \\
&t_i & t_{i+1} & t_{i+2} & t_{i+3} & t_{i+4} & t_{i+5}
\end{align*}
\]
Sample rewritings

Views

\( v_1 \)

\[ \text{article}_{ID} \]

\[ \text{title}_{val} \]

\[ \text{author}_{val} \]

\( v_2 \)

\[ \text{article}_{ID} \]

Query

Rewriting

\[ \text{article} \]

\[ \text{title}_{val} \]

\[ \text{author}_{val} \]

\[ \bowtie \text{article}_{ID} \]

\[ v_1 \]

\[ v_2 \]
Sample rewritings

Views

\[ \nu_1 \]

proceedings

\[ \nu_2 \]

article_{ID}

title_{ID,val}

author_{val}

Query

proceedings

article

title_{val}

author_{val}

Rewriting

\[ \Box \diamond article.ID \prec title.ID \]

\[ \nu_1 \]

\[ \nu_2 \]
Sample rewritings

Views

<table>
<thead>
<tr>
<th>( \nu )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{proceedings}_{cont} )</td>
</tr>
</tbody>
</table>

Query

<table>
<thead>
<tr>
<th>( \text{proceedings} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{article} )</td>
</tr>
</tbody>
</table>

Rewriting

<table>
<thead>
<tr>
<th>( \text{nav} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{article} )</td>
</tr>
<tr>
<td>( \text{title}_{val} )</td>
</tr>
<tr>
<td>( \text{author}_{val} )</td>
</tr>
<tr>
<td>( \nu )</td>
</tr>
</tbody>
</table>
Sample physical rewriting plan

project@
bordeveau-25.bordeaux.grid5000.fr

receive@
bordeveau-25.bordeaux.grid5000.fr

hashJoin@
pastel-79.toulouse.grid5000.fr

scan(4nodesView)@
pastel-79.toulouse.grid5000.fr

receive@
pastel-79.toulouse.grid5000.fr

scan(3nodesView)@
griffon-92.nancy.grid5000.fr
ViP2P platform

- 70,000 Java lines (300 classes)
- Berkeley DB v3.3.75 to store view data
- FreePastry v2.1 as the DHT
- Experiments on Grid5000 using 250 machines / 1000 peers
ViP2P platform

- 70,000 Java lines (300 classes)
- Berkeley DB v3.3.75 to store view data
- FreePastry v2.1 as the DHT
- Experiments on Grid5000 using 250 machines / 1000 peers

Algorithms inside:

- SAX-based custom processor for computing $v(d) +$ identifiers
- Iterator-based execution engine + structural joins, send/receive ...
ViP2P platform

- 70,000 Java lines (300 classes)
- Berkeley DB v3.3.75 to store view data
- FreePastry v2.1 as the DHT
- Experiments on Grid5000 using 250 machines / 1000 peers

Algorithms inside:
- SAX-based custom processor for computing $v(d) +$ identifiers
- Iterator-based execution engine + structural joins, send/receive . . .

**Scalability** (mostly publishing and XML querying):
- published up to 160 GB
- data disseminated up to 230 MB/s depending on configuration

Many experiments described in technical report + [MKVZ11].
Part II

RDFViewS: selecting materialized views in an RDF database
Joint work with

Faculty
François Goasdoué

Students
Konstantinos Karanasos, Julien Leblay and Stamatis Zampetakis

http://rdfviews.saclay.inria.fr
Warehousing RDF data

RDF: complex graph-structured data for describing *resources*
Warehousing RDF data

RDF: complex graph-structured data for describing *resources*

- Query performance issues
- Recent work on RDF data management platforms (HexaStore, RDF3X, ...)

RDF implicit triples: part of the RDF and RDF-Schema (RDF-S) specifications. In particular:
- classes and properties
- class inclusion (subClassOf), property inclusion (subPropertyOf),
- domain and range typing of properties (domain and range)

RDF query results must reflect implicit triples

Query for person must return student, also carOwner
Warehousing RDF data

RDF: complex graph-structured data for describing resources

- Query performance issues
- Recent work on RDF data management platforms (HexaStore, RDF3X, . . .)

RDF implicit triples: part of the RDF and RDF-Schema (RDF-S) specifications. In particular:

- classes and properties
- class inclusion (subClassOf), property inclusion (subPropertyOf), domain and range typing of properties (domain and range)
Warehousing RDF data

RDF: complex graph-structured data for describing *resources*

- Query performance issues
- Recent work on RDF data management platforms (HexaStore, RDF3X, ...)

RDF implicit triples: part of the RDF and RDF-Schema (RDF-S) specifications. In particular:

- classes and properties
- class inclusion (subClassOf), property inclusion (subPropertyOf), domain and range typing of properties (domain and range)

RDF query results must reflect implicit triples

- Query for *person* must return *student*, also *carOwner*
Warehousing RDF data

Problem statement

**Given** query workload $\mathcal{W}$

- cost function $cf$ (view + workload costs)

**Find** set of views to materialize so as to minimize $cf$
- taking also into account implicit triples
Problem statement

Given query workload $\mathcal{W}$

Cost function $cf$ (view + workload costs)

Find set of views to materialize so as to minimize $cf$

taking also into account implicit triples

- Centralized setting
- Complete query rewriting, i.e. $\mathcal{W}$ can be evaluated based only on the views
- Meant to complement an RDF query processor (adapt $cf$)
RDF view selection as a search problem

In the style of the relational warehouse work [TLS01, TS97]

- **Initial state** recommended views = $\mathcal{W}$
- **Transition** a change to one or several views, possibly adding or removing a view
  - selection cut $EC$: remove a $\sigma$ edge
  - join cut $JC$: remove a $\Join$ edge, may split the view in two disjoint fragments, or leave it intact
  - view break $VB$: split a view in overlapping fragments
  - view fusion $VF$: merge two views into one
### Sample initial state

**v1**

- v1:n1.p=hasName
- v1:n1.s=n3.s
- v1:n1.s=n2.s
- v1:n2.o=postImpres
- v1:n3.p=hasPainted
- v1:n3.o=n4.s
- v1:n3.o=n5.s
- v1:n4.p=hasTitle
- v1:n5.o=moma
- v1:n5.p=isExpIn
- v1:n6.o=n6.s
- v1:n6.p=isNamed

**v2**

- v2:n7.p=hasCountry
- v2:n7.o=france
- v2:n7.s=n9.s
- v2:n8.s=n9.s
- v2:n8.o=postImpres
- v2:n9.p=hasPainted
- v2:n9.s=n10.s
- v2:n9.o=n11.s
- v2:n10.p=hasTitle
- v2:n10.s=n11.s
- v2:n11.p=isExpIn
- v2:n11.o=n12.s
- v2:n11.s=n13.s
- v2:n12.p=isNamed
- v2:n12.s=n13.s
- v2:n13.p=isLocatIn
- v2:n13.o=europe
$v_3$ is obtained from $v_2$
Search space for a very small workload

\[ Q = \{ q \} \]

\[ q(Y, Z): - \]

\[ t(X, Y, c_1), \]

\[ t(X, Z, c_2) \]

\[ V_0 \{ q(Y, Z): - t(X, Y, c_1), t(X, Z, c_2) \} \]

\[ V_1 \{ q_1(X_1, Y): - t(X_1, Y, c_1); \ q_2(X_2, Z): - t(X_2, Z, c_2) \} \]

\[ V_2 \{ q(Y, Z, W_1): - t(X, Y, W_1), t(X, Z, c_2) \} \]

\[ V_3 \{ q(Y, Z, W_2): - t(X, Y, c_1), t(X, Z, W_2) \} \]

\[ V_4 \{ q(Y, Z, W_1, W_2): - t(X, Y, W_1), t(X, Z, W_2) \} \]

\[ V_5 \{ q_1(X_1, Z, W_1): - t(X_1, Z, W_1); \ q_2(X_2, Z): - t(X_2, Z, c_2) \} \]

\[ V_6 \{ q_1(X_1, Z): - t(X_1, Z, c_1); \ q_2(X_2, Z, W_2): - t(X_2, Z, W_2) \} \]

\[ V_7 \{ q_1(X_1, Z, W_1): - t(X_1, Z, W_1); \ q_2(X_2, Z, W_2): - t(X_2, Z, W_2) \} \]

\[ V_8 \{ q(X, Y, Z): - t(X, Y, Z) \} \]
Search space, strategies and heuristics

Search space size: \( NS(Q, n) \leq \sum_{k=1}^{n} 2^{kn^2} \mu(n, k) B_k \)
- \( n \) atoms in the workload
- \( m \) edges in the workload (\( \sigma \)s or \( \bowtie \)s)
- \( \mu(n, k) \) the number of minimal \( k \)-covers of a set of size \( n \)
- \( B_k \) the \( k \)-th Bell number

Stop conditions specific to RDF:
- Reject views without constants or URIs
- Do not develop views with Cartesian products

Smart heuristic search strategies
Saturate the database

Available in many tools
Maintenance difficult when the data or the schema change
Not always possible (write access to the database)
RDF view selection and implicit triples

Saturate the database
Available in many tools
Maintenance difficult when the data or the schema change
Not always possible (write access to the database)

Reformulate the queries

- Query reformulation algorithm for conjunctive SPARQL based on RDFS [GKLM11]
- Pre-reformulation: Run view search on reformulated workload ⇒ search space explosion...
- Post-reformulation: Include implicit triples in statistics, reformulate recommended views in the end
Results

Implemented the search in Java 1.6
Materialized the views in Postgres
Compared against triple table with dictionary encoding and HexaStore indices

Main results

- Our heuristic search strategies scale up to 200 queries / 10 atoms
- Cost reductions of many orders of magnitude
- Custom materialized views allow much faster query evaluation
- Post-reformulation more efficient and effective than pre-reformulation
Part III

Perspectives
2004, Val Tannen (approx.): “DE is surely the kind of thing the Semantic Web community needs”
2004, Val Tannen (approx.): “DE is surely the kind of thing the Semantic Web community needs”
2010, I. Horrocks (approx.): “Such a paper would smell database-ish and the (ISWC) reviewers will reject it!”

My conclusion
I will continue working to apply data engineering to the Semantic Web and data visualization.
2004, Val Tannen (approx.): “DE is surely the kind of thing the Semantic Web community needs”

2010, I. Horrocks (approx.): “Such a paper would smell database-ish and the (ISWC) reviewers will reject it!”

2011, senior DB INRIA person (approx.): “You should stay within the DB field! Don’t defect to AI! Besides, AI is poorly considered in INRIA. You can work with data visualization people, though - that’s cool”
2004, Val Tannen (approx.): “DE is surely the kind of thing the Semantic Web community needs”

2010, I. Horrocks (approx.): “Such a paper would smell database-ish and the (ISWC) reviewers will reject it!”

2011, senior DB INRIA person (approx.): “You should stay within the DB field! Don’t defect to AI! Besides, AI is poorly considered in INRIA. You can work with data visualization people, though - that’s cool”

My conclusion

I will continue working to apply data engineering to the Semantic Web and data visualization.
France (and other EU countries) have a legal obligation to share public data administration with the public

- how many pupils in school per year per age . . . and how many professors
- how many tax cuts for the car industry . . . and how many cars sold . . . and how many jobs in carmaking
- political party affiliation of local and regional leaders across the years . . . and their budget decisions

This data is being collected by private operators (e.g., datapublica) or public ones (e.g., Etalab)
France (and other EU countries) have a legal obligation to share public data administration with the public

- how many pupils in school per year per age . . . and how many professors
- how many tax cuts for the car industry . . . and how many cars sold . . . and how many jobs in carmaking
- political party affiliation of local and regional leaders across the years . . . and their budget decisions

This data is being collected by private operators (e.g., datapublica) or public ones (e.g., Etalab) . . . under the form of a set of Excel and PDF tables!
Add public administration data to the LOD cloud

Democracy ↔ information ↔ usable data
Add public administration data to the LOD cloud

Democracy $\iff$ information $\iff$ usable data

- Searchable, navigable, structured (à la Google FusionTables, but strong heterogeneity should be OK)
Add public administration data to the LOD cloud

Democracy ⇐ information ⇐ usable data

- Searchable, navigable, structured (à la Google FusionTables, but strong heterogeneity should be OK)
- It should be possible to discover and record links
Add public administration data to the LOD cloud

Democracy ⇐ information ⇐ usable data

- Searchable, navigable, structured (à la Google FusionTables, but strong heterogeneity should be OK)
- It should be possible to discover and record links
- It should be possible to approach the data visually
Add public administration data to the LOD cloud

Democracy ⇐ information ⇐ usable data
- Searchable, navigable, structured (à la Google FusionTables, but strong heterogeneity should be OK)
- It should be possible to discover and record links
- It should be possible to approach the data visually

What’s in for the CS scientists?
- Data integration with a purpose!
Add public administration data to the LOD cloud

Democracy ⇐ information ⇐ usable data

- Searchable, navigable, structured (à la Google FusionTables, but strong heterogeneity should be OK)
- It should be possible to discover and record links
- It should be possible to approach the data visually

What’s in for the CS scientists?

- Data integration with a purpose!
- Wrapper induction, learning, schema extraction…
Add public administration data to the LOD cloud

Democracy ⇐ information ⇐ usable data

- Searchable, navigable, structured (à la Google FusionTables, but strong heterogeneity should be OK)
- It should be possible to discover and record links
- It should be possible to approach the data visually

What’s in for the CS scientists?

- Data integration with a purpose!
- Wrapper induction, learning, schema extraction. . .
- Schema matching + mapping, entity resolution. . .
Add public administration data to the LOD cloud

Democracy ⇔ information ⇔ usable data
- Searchable, navigable, structured (à la Google FusionTables, but strong heterogeneity should be OK)
- It should be possible to discover and record links
- It should be possible to approach the data visually

What’s in for the CS scientists?
- Data integration with a purpose!
- Wrapper induction, learning, schema extraction...
- Schema matching + mapping, entity resolution...
- RDF analytics: visual vs. languages and technical underpinnings (cubes...)
Part IV

Summary
Summary

P2P warehousing of Web Data

ViP2P: efficient publishing and querying of data based on a DHT network

Ongoing work:

1. XR: combining XML and RDF (K. Karanasos, F. Goasdoué, J. Leblay, S. Zampetakis)
2. Self-adaptive XML views (A. Katsifodimos, V. Vassalos)
3. RDF saturation vs. reformulation (F. Goasdoué, A. Roatis)
4. RDF and XML data management in a cloud (J. Camacho-Rodriguez, F. Bugiotti, N. Bidoit, D. Colazzo, F. Goasdoué)
5. OLAP for RDF (F. Goasdoué, D. Colazzo)
6. Turning French public administration data into an RDF database
Closest related works

DHT-based sharing of relations  [LHSH04]
DHT-based XML indexing  [GWJD03, BC06, SHA05, AMP+08, RM09]
DHT-based shared XML caches  [LP08]
XPath query rewriting  [BOB+04, XO05, CDO08, TYÖ+08]
  - XPath: wildcard *, union
  - Rewritings: intersection, navigations, joins

Rewriting with structural constraints  [ABMP07]
  - Centralized setting
  - Dataguide [GW97] constraints

Layered architecture for Web content warehousing  [AAC+08]
RDF querying and reasoning on DHT  [KMK08, LIK06]
Part V

Thank you!
WebContent: efficient P2P warehousing of web data.  

Structured materialized views for XML queries.  

XML processing in DHT networks.  

[BC06] Angela Bonifati and Alfredo Cuzzocrea.  
Storing and retrieving XPath fragments in structured P2P networks.
Algebraic incremental maintenance of XML views.
In *EDBT*, 2011.

A framework for using materialized XPath views in XML query processing.

XPath rewriting using multiple views: Achieving completeness and efficiency.

Materialized view-based processing of RDF queries.
Bases de Données Avancées, 2011.


[LHSH04] Boon Thau Loo, Ryan Huebsch, Ion Stoica, and Joseph M. Hellerstein.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Authors</th>
<th>Title</th>
<th>Conference</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>[MZ09]</td>
<td>Ioana Manolescu and Spyros Zoupanos</td>
<td>XML materialized views in P2P.</td>
<td>DataX workshop (not in the proceedings)</td>
<td>2009</td>
</tr>
</tbody>
</table>

In *OTM Conferences (2)*, 2005.


Multiple materialized view selection for XPath query rewriting.

[XO05] W. Xu and M. Ozsoyoglu.
Rewriting XPath queries using materialized views.
In *VLDB*, 2005.