Small can be Beautiful in the Semantic Web

Marie-Christine ROUSSET
Univ. Paris-Sud & CNRS (LRI)
INRIA(Futurs)
The Semantic Web today

- Methodologies, formal languages, platforms and standards for building (domain) ontologies
  - Methontology, On-To-Knowledge
  - Description Logics, F-logic
  - Kaon, OilEd, OntoEdit, Ontolingua, OntoSaurus, Protege2000, WebOde, WebOnto
  - XML, RDF, OWL

- With domain specific applications
  - Knowledge management
  - Thematic portals
    - Information integration systems related to a same domain
The current Semantic Web vision

- A « big is beautiful » vision of the ontologies supposed to be required for data integration
  - the main current application of SW technologies
- Expressivity is favoured against efficiency or even feasibility of machine processing
  - OWL Full is undecidable
  - OWL Lite is ExpTime-complete
- Cannot not scale up to the Web
  - « semantic » Google does not exist
Summary of this talk

a « small is beautiful » vision of ontologies

- for an easy deployment of thematic portals
  - both for humans and for machines
  - report on the PICSEL project

- for a Semantic Web viewed as a huge semantic peer-to-peer data management system
  - based on simple ontologies and mappings distributed at Web scale
  - implemented in the SomeWhere infrastructure
Thematic portals

- Provide a single entry point for querying a collection of distributed pre-existing data sources related to a same domain
  - tourism, medicine, biology, finance, education

- Underlying model and algorithms
  - Mediator-based information integration
  - Query rewriting using views
The mediator model

3 levels of language

mediated schema between users queries and views describing the sources content

data sources
Query answering in a mediator model

Query rewriting using views

Domain ontology

Query engine

Source description

Query plan

V1 V3

data sources
Query answering in a mediator model

Query plan execution through wrappers

Domain ontology

Query engine

Source description

Source description

Query plan

wrapper

V1

V3

wrapper

wrapper

wrapper

wrapper

data sources
Overview of existing systems

- **Relational** versus **Object-oriented** approach

- **Global as Views** versus **Local as Views**
  - **GAV**: mediated schema: views on sources schemas
    - query reformulation: simple unfolding
  - **LAV**: each source schema: views on the mediated schema
    - query reformulation: *query rewriting using views*
Query rewriting using views

- Extensively studied in relational DB theory
  - central for query optimization and information integration
  - decidability and complexity results
    - depending on the languages used for the queries, rewritings and views
    - NP-hard when queries, rewritings and views are conjunctive queries

- Little studied when queries and views are defined w.r.t an ontology
  - report on PICSEL experience
Picsel

- Generic environment for developing thematic portals based on domain ontologies
  - applied to the tourism domain
  - also used in electronic commerce (MKBeem)
- funded by France Telecom R&D
  - two patented pieces of software
    - OntoClass, OntoQuery
- Joint work with
  - A. Léger
  - F. Goasdoué, C. Reynaud, B. Safar
Choices made in Picsel

- a « simple » DL ontology language
  - ALN : less expressive than OWL Lite
  - Polynomial complexity
- a conjunctive query language
  - over concepts and roles
  - a sublanguage of CARIN (combining DL and Horn rules)
- a restricted language of views
  - No combination between DL and rules in views
Illustration on the tourism domain

Class hierarchy automatically built from ALN definitions:

Queries correspond to CARIN-ALN rules:

\[ Q(X) :- \text{Hotel}(X) \land \text{Location}(X, \text{london}) \]

\[ Q'(X) :- \text{Flight}(X) \land \forall \text{Stop}.\text{AmCity}(X) \land \text{Airline}(X,Y) \land \text{AmCompany}(Y) \]
Source descriptions

**Source 1:** provides *Flights* having atmost one *Stop*
\[ v1(X) : - (\text{Flight} \cap (\leq 1 \text{ Stop}))(X) \]

**Source 2:** provides *Flights* whose *Airline* is *American*
\[ v2(X) : - (\text{Flight} \cap (\geq 1 \text{ Airline}) \cap (\forall \text{ Airline.AmCompany}))(X) \]

**Source 3:** provides *American Cities on the East Coast*
\[ v3(X) : - (\text{AmCity} \cap (\forall \text{ Located.Eastcoast}))(X) \]

**Source 4:** provides pairs of *Flights* and *Stops*
\[ v4(X,Y) : - \text{Stop}(X,Y) \]
Query rewriting in Picsel by example

\[ Q'(X) :- \text{Flight}(X) \land \forall \text{Stop}.\text{AmCity}(X) \land \text{Airline}(X,Y) \land \text{AmCompany}(Y) \]

\[ Q'a(X) :- \text{Flight}(X) \land \forall \text{Stop}.\text{AmCity}(X) \land (\geq 1 \text{ Airline}) \land (\forall \text{ Airline}.\text{AmCompany})(X) \]

\[ v1(X) :- (\text{Flight} \cap (\leq 1 \text{ Stop}))(X) \]

\[ v2(X) :- (\text{Flight} \cap (\geq 1 \text{ Airline}) \land (\forall \text{ Airline}.\text{AmCompany}))(X) \]

\[ v3(X) :- (\text{AmCity} \land (\forall \text{ Located.Eastcoast}))(X) \]

\[ v4(X,Y) :- \text{Stop}(X,Y) \]

\[ P'a(X) :- v1(X) \land v4(X,Y) \land v3(Y) \land v2(X) \]

\[ (\text{Flight} \cap (\leq 1 \text{ Stop}))(X) \land \text{Stop}(X,Y) \land (\text{AmCity} \land (\forall \text{ Located.Eastcoast}))(Y) \land (\geq 1 \text{ Airline}) \land (\forall \text{ Airline}.\text{AmCompany})(X) \]
Lessons learnt from PICSEL

- A tractable DL can lead to complex query rewriting
  - polynomial in the number of views
  - exponential in the size of the query and in the maximal size of the views
  - $O(n^n)$ where $n$ is the biggest $k$ s.t. $(\leq k \ r)$ appears in the views

- Full ALN not handled by human users

- Manual building of a domain ontology is time-consuming

- PICSEL2:
  - ALN replaced by AL
  - semi-automatic construction of AL ontologies from a set of XML schemas
Thematic portals: summary

- A centralized vision of mediation based on single domain ontologies
  - appropriate for integrating a few dozens of sources
  - not flexible enough to scale up to the whole web
- Requirement for the Semantic Web viewed as a (huge) Peer Data Management System
  - a distributed P2P mediation
A PDMS

- Coalition of information servers
  - each server can play the role of:
    - a data (or service) provider
    - a mediator for queries
- Knowledge required at each server
  - its own ontology (its local schema)
  - the description of the data stored locally (or the services offered locally)
  - semantic mappings between its ontology and the ontologies of some of its peers (its acquaintances in the network)
The PDMS model

- Extension of the P2P model of existing file sharing systems
  - Gnutella, Kazaa, Chord
- Richer description of data and more complex queries
- Same principle: no central server
  - any peer is a possible entry point in the network
    - for a user who wants to query the whole network
    - for a new peer which wants to join the network
Web ontologies in that setting

- Should be simple
  - to be human understandable and machine processable at a large scale
    - example: taxonomies of atomic classes
      - a tractable fragment of OWL
      - formal semantics easy to understand by humans

- personalized
  - just like personal file systems, mail files or bookmarks

- distributed

- semantically connected by mappings
SomeWhere

- A PDMS infrastructure based on propositional logic
  - for defining ontologies and mappings
  - based on a sublanguage of OWL DL
- Experimental study of its scalability
- Joint work with
  - P. Adjiman, P. Chatalic, F. Goasdoué, L. Simon
SomeWhere in a nutshell

- topology is not fixed
- a new peer joins the PDMS via some acquaintances:
  - by declaring mappings between its ontology and the ontologies of some peers in the PDMS that it knows
SomeWhere Data Model

Ontology: hierarchy of intentional classes

Storage description: extensional classes

More complex inclusion statement: $\text{St}_A_3 \sqsubseteq A_1 \sqcap \neg A_2$
SomeWhere Data Model

Mappings:

Logical combination of class literals: \( A_1 \bigcap \neg A_3 \bigcap A_2 \)

\( A_1 \bigcap (A_2 \bigcup \neg B_3) \subseteq B_1 \bigcap \neg B_3 \)

\( B \bigcup B_1 \bigcup B_2 \bigcup B_3 \)
A simple example
Query rewriting: illustration

Rewritings

- $St_{Pop\_Rock}$
- $St_{Francais} \sqcap St_{Pop}$
- $St_{US} \sqcap St_{Pop}$
- $St_{Mouv} \sqcap St_{Pop}$

Rewritings

- $St_{Pop\_Rock}$
- $St_{Francais} \sqcap St_{Pop}$
- $St_{US} \sqcap St_{Pop}$
- $St_{Mouv} \sqcap St_{Pop}$

Rewritings

- $St_{Pop\_Rock}$
- $St_{Francais} \sqcap St_{Pop}$
- $St_{US} \sqcap St_{Pop}$
- $St_{Mouv} \sqcap St_{Pop}$
Query rewriting problem

- Can be reduced to a distributed reasoning problem in propositional logic
  - Ontologies encoded as a set of clauses of length 2
  - Mappings encoded as a set of clauses
- A message-passing algorithm
  - anytime
    - complete with no restriction on the structure of the network
- Scalability experiments
  - on randomly generated networks of 1000 peers
  - having the topology of small worlds
    - Close to the topology of the web
Varying topologies

1000 peers Ring, 10 neighbours/peer

P: probability of redirecting an edge

P = 0.01
Model of Watts and Strogatz

P = 0.1
Small world

P = 1
Random graph
Scalability results

- **Additional parameters**
  - a varying number of mappings per edge
  - a varying complexity of mappings
    - ratio of clauses of length 3 (0%, 20%, 100%)
  - a 30 seconds timeout per query

- **Depth of query processing**
  - A majority of runs have a small depth (less than 7) even on the hard cases

- **Time to produce a number of answers**
  - In 90% cases, the first answer is produced within 2 seconds
  - Easy cases (simple mappings):
    - Few answers per query (5 on average)
    - very fast (less than 0.1s) to compute all the answers without timeouts
  - Hard cases (complex and more mappings per edge)
    - around 1000 answers per query (but > 30% queries not complete: timeouts)
    - quite fast to obtain them (less than 20s)
Ongoing work

- Connection with the SomeOne project of FT R&D
- Plug SomeWhere in CHORD infrastructure for optimizing the query routing
Conclusion

- Same message for the Semantic Web as 20 years ago for Knowledge Representation

[Patel-Schneider 84]: Small can be beautiful in Knowledge Representation
  - limit the expressive power of ontologies in order to make SW technologies usable as part of larger systems
  - towards a « semantic » Google: replacing words by terms of a taxonomy
    - any user is free to use his own taxonomy to annotate his web resources but must attach his taxonomy to the resources he has annotated as a context of interpretation of the terms
    - SomeWhere: a possible infrastructure to implement such a semantic google