

# Making the Local-As-View approach feasible in Semantic Web and Application with Open Data and DBpedia

COUTABLE Guillaume, GAUTIER Gaëtan, RULLIER Noémie

1<sup>er</sup> mai 2013

# Table of contents

- 1 Introduction
- 2 Problematic
- 3 Proposed solutions

# Introduction

- Web semantic and open data
  - Lots of data
  - Need to query
- Data integration
  - Warehousing
  - Mediation (GAV and LAV)
    - Rewriting algorithm (Bucket, Minicon)

# Data integration

## Warehousing

- The data sources are aggregated into a global basis
- The sources are not interviewed in a query

### Difficulties

- Effectively detect and perform the update of data
- Not applicable to a large number of sources
- Add new source

# Data integration

## Warehousing

- The data sources are aggregated into a global basis
- The sources are not interviewed in a query

### Difficulties

- Effectively detect and perform the update of data
- Not applicable to a large number of sources
- Add new source

# Data integration

## Mediation

The mediator asks necessary data sources to the query and assembles the sub-results.

# Data integration

## Mediation Global-As-View

The global schema was expressed according with the data sources.

### Example

$$Product = Source1 \cup Source2 \quad (1)$$

# Data integration

## Mediation Local-As-View

- The data sources are expressed according with the global schema.
- For each request, the mediator search the sources which can provide results. This allows the source to change regularly.

### Example

*Source1* < include or equal > *Products* < join > *Vendors* (2)

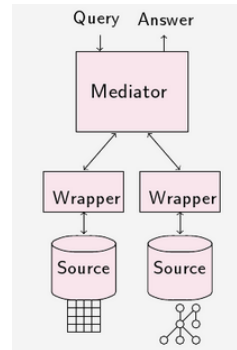


# Problematic

Making the Local-As-View approach feasible in Semantic Web

# Proposed solutions

- GUN
  - Detection of relevant views
  - Rewritings only on those views
- SemLAV
  - Abandonment of rewritings
  - Immediate execution of view's body
- Comparison with other approaches



# GUN

Main feature : only few rewritings are executed.

- 1 Use Bucket algorithm to calculate the relevant views
- 2 ??????????????????

# SemLAV

Improvement of GUN algorithm : no rewritings are executed.

- ① Use Bucket algorithm to calculate the relevant views and build the buckets (No change here)
- ② Storage of views based on the number of their appearances

Consequence : The bulk of the result is calculate faster