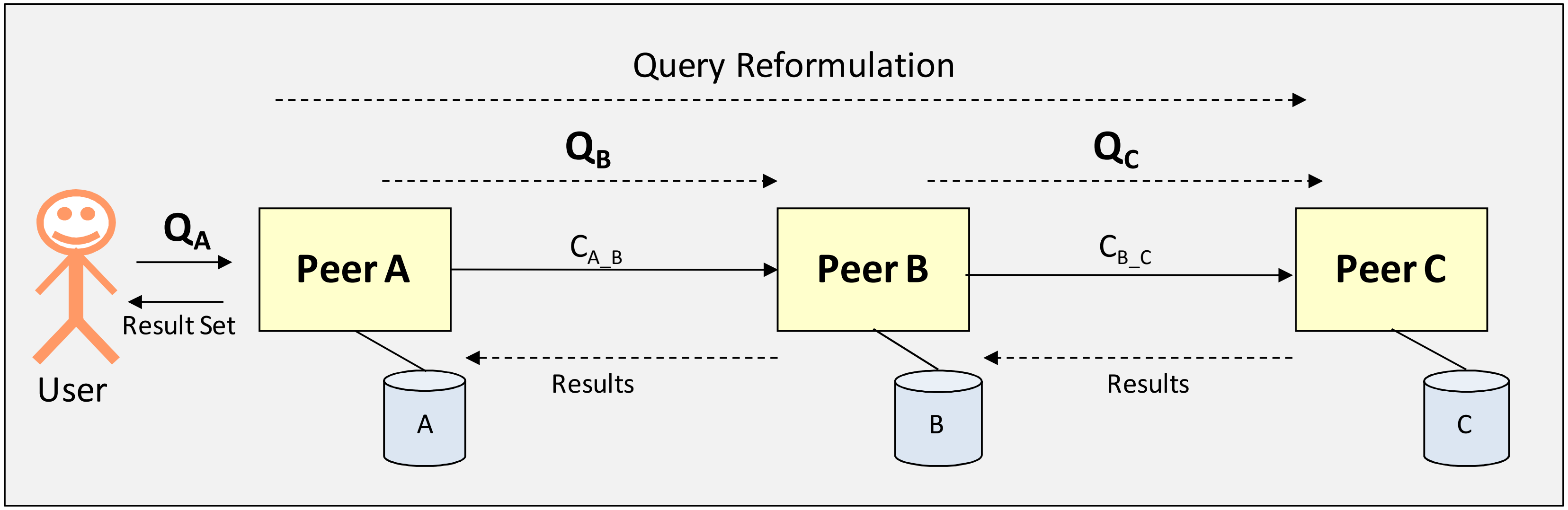
Using Semantics to Enhance Query Reformulation in Dynamic Environments

Damires Souza (damires@ifpb.edu.br)
Thiago Arruda (tan@cin.ufpe.br)
Ana Carolina Salgado (acs@cin.ufpe.br)
Patricia Tedesco (pcart@cin.ufpe.br)
Zoubida Kedad (zoubida.kedad }@prism.uvsq.fr)

VIRTUS IMPAVIDA
Centro de Informática U·F·P·E
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PARAÍBA
UNIVERSITE DE VERSAILLES SAINT-QUENTIN-EN-YVELINES
Laboratoire PRiSM

Motivation



- Dynamic distributed environments are composed by autonomous and heterogeneous data sources – ***peers***
- Peers are semantically connected to each other by means of mappings or ***correspondences***

Motivation – our focus

How to ***reformulate*** queries among the peers,
in such a way that the resulting ***set of answers***
expresses, as close as possible, what the ***users***
intended to obtain at query submission time,
considering the ***dynamicity*** of the environment ??

Query Reformulation

I. Querying distributed data sources should be ***useful*** for users

- ✓ Resulting query answers should be in conformance with users' preferences.
- ✓ It is not useful for users when they do not receive any answer at all.

II. Concepts from a source peer do not always have ***exact corresponding concepts*** in a target one

- ✓ ***Empty reformulation*** and, possibly, no answer to the user.

Outline

- Applying Semantics to Query Reformulation
- The ***SemRef*** Approach
- Experiments and Results
- Related Work
- Conclusions and Further Work

Our Rationale

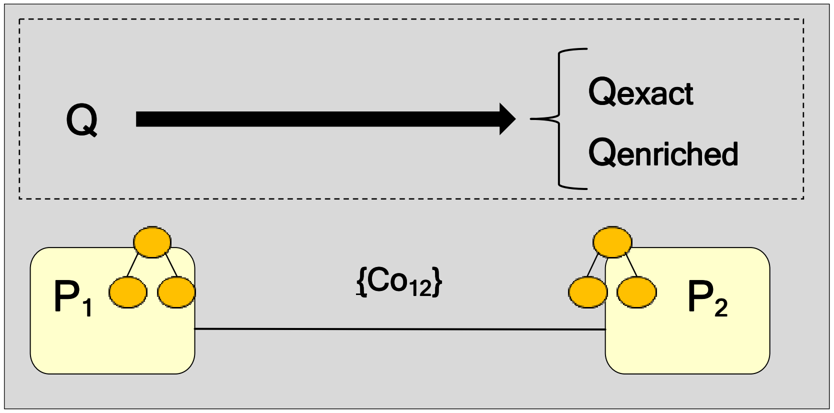
- Using ***semantics*** to enhance the reformulation
  I. ***Users' preferences, query semantics*** and the ***current status of the environment*** should be taken into account at query reformulation time

  We use ***contextual information***

  II. The original query should be ***adapted*** to bridge the gap between the two sets of concepts

  We use ***query enrichment***

Setting for our Approach

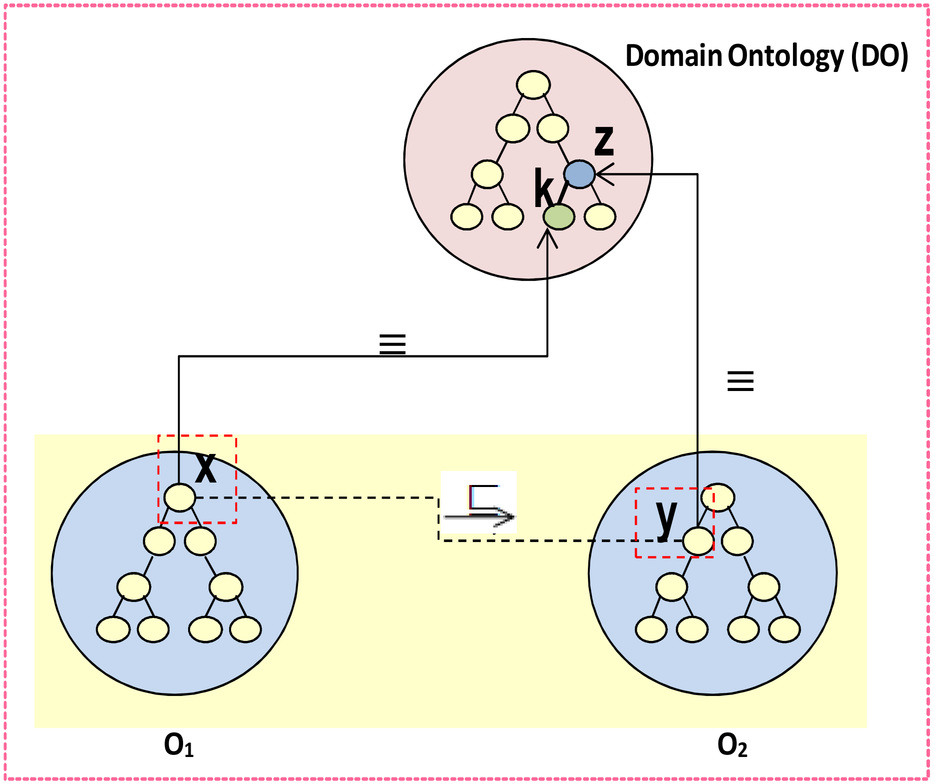


- Goal: find reformulated queries of ***Q*** expressed in terms of the concepts of O_2
- Schemas are represented as ontologies

- ***Two*** query reformulations may be produced:
  - ✓ an ***exact*** one, considering only equivalence correspondences; and
  - ✓ an ***enriched*** one, resulting from the set of other correspondences.
- Users will be provided with a set of ***expanded*** answers.

Using Domain Ontologies to Identify Semantic Correspondences

- We use ***Domain Ontologies – DO*** as background knowldege to identify semantic correspondences
  - To bridge the conceptual *differences* or *similarities* between two overlapping ontologies
  - Matching Ontologies are terminologically *normalized* according to the DO



Using Domain Ontologies to Identify Semantic Correspondences

- A ***Semantic Correspondence*** is defined as one of the following expressions:

  1. $O_1{:}x \xrightarrow{\equiv} O_2{:}y$, an *isEquivalentTo* correspondence
  2. $O_1{:}x \xrightarrow{\sqsubseteq} O_2{:}y$, an *isSubConceptOf* correspondence
  3. $O_1{:}x \xrightarrow{\sqsupseteq} O_2{:}y$, an *isSuperConceptOf* correspondence
  4. $O_1{:}x \xrightarrow{\trianglerighteq} O_2{:}y$, an *isPartOf* correspondence
  5. $O_1{:}x \xrightarrow{\triangleleft} O_2{:}y$, an *isWholeOf* correspondence
  6. $O_1{:}x \xrightarrow{\approx} O_2{:}y$, an *isCloseTo* correspondence
  7. $O_1{:}x \xrightarrow{\perp} O_2{:}y$, an *isDisjointWith* correspondence

  where x and y are elements (concepts/properties) belonging to the matching ontologies O_1 and O_2.

Using Domain Ontologies to Identify Semantic Correspondences

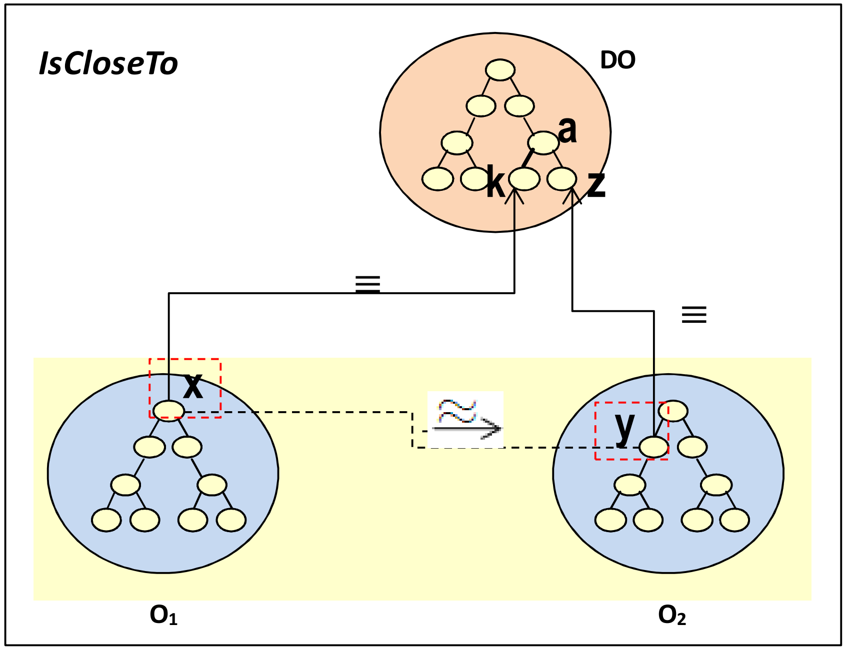
- An element O_1:x ***isCloseTo*** O_2:y If

  (O_1:x $\equiv$ DO:k and O_2:y $\equiv$ DO:z) and (DO:k $\sqsubseteq$ DO:a and DO:z $\sqsubseteq$ DO:a) and DO:a $\neq$ $\top$ and DO:a *isRigid* and $\neg$(DO:k $\perp$ DO:z) and (depth(DO:a, DO:$\top$) $\geqslant$ tresholdRoot)

  and

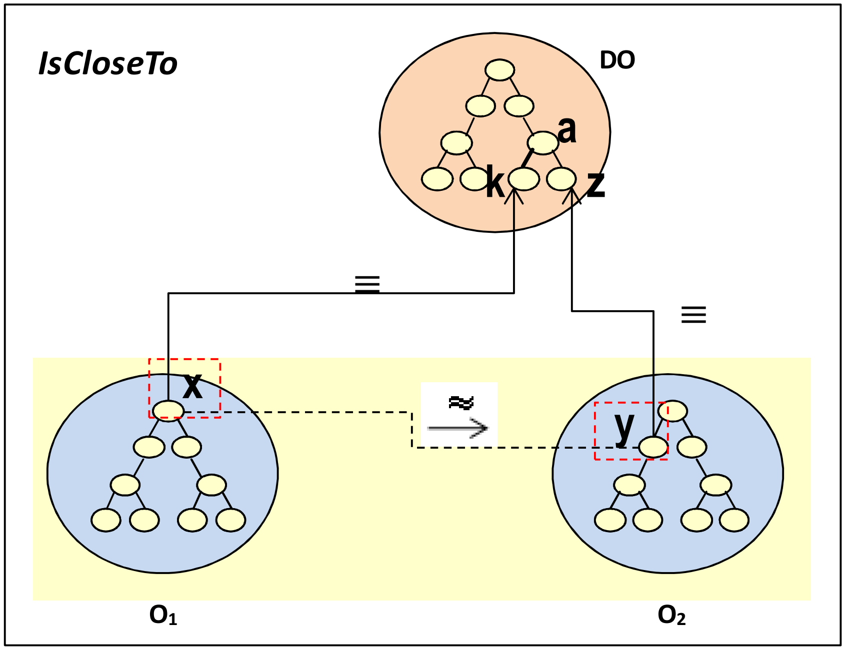
  (depth(DO:k,DO:a) $\leqslant$ thresholdCommonAncestor and

  depth(DO:z,DO:a) $\leqslant$ thresholdCommonAncestor)



Using a Domain Ontology to Define Semantic Correspondences

- O_1:k and O_2:z are close if
  - ✓ They share a common ancestor in the DO
  - ✓ The common ancestor is not the root
  - ✓ The concepts do not hold any subsumption nor disjointness
  - ✓ The measured depths are evaluated to true



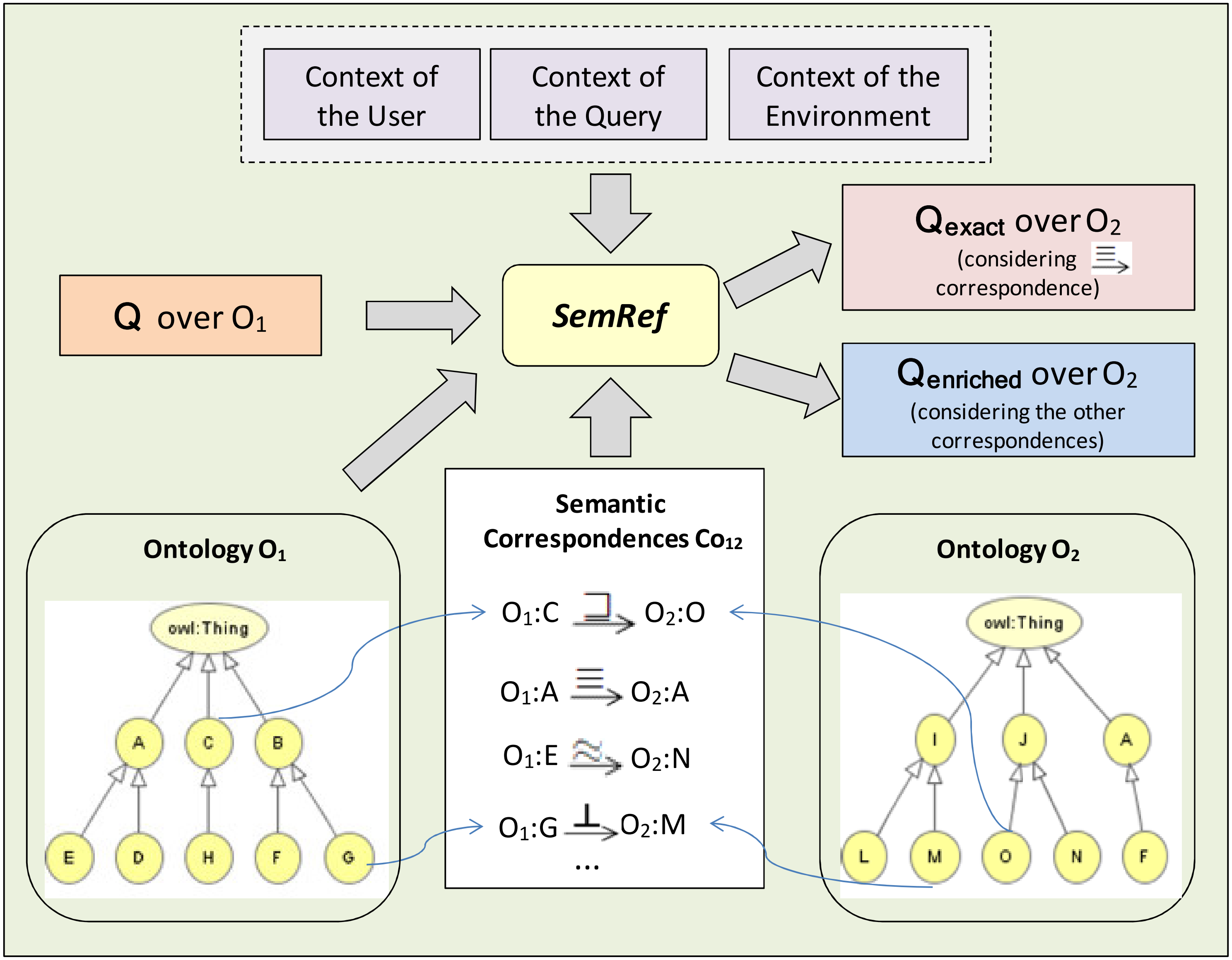
Example:

O_1.Notebook $\overset{\approx}{\rightarrow}$ O_2.MacintoshPC

Using Context in Query Reformulation

- ***User Context*** (preferences):
  - Exact reformulation is the default option
  - Enriching variables: ***Approximate***, ***Specialize***, ***Generalize***, *and* ***Compose***.
- ***Query Context:*** Query semantics + Query reformulation mode
  - ***Restricted***: the priority is to produce an exact reformulation, although if it results empty, then an enriched reformulation may be provided
  - ***Expanded***: exact and enriched reformulations are to be produced.
- ***Environment Context:*** *path_length (*number of subsequent reformulations) + submission peer's identification and its neighbors context.

General Principle of Applying Semantics to Query Reformulation



The *SemRef* Approach – The Algorithm

- Our approach has been encoded in $\mathcal{ALC}$-DL
- ***Query Q*** is a query expressed over P_i's ontology, having the following form:

$$Q = Q_1 \sqcup Q_2 \sqcup \ldots \sqcup Q_M$$

Where $Q_i = C_1 \sqcap C_2 \sqcap \ldots \sqcap C_N$

Each C_j is an atomic concept, a negated atomic concept of a quantified atomic concept (C_j, $\neg C_j$, $\forall R.C_j$ or $\exists R.C_j$)

$$Q_3 = [Teacher \sqcap Researcher] \sqcup [Student \sqcap Researcher]$$

The *SemRef* Approach – The Algorithm

Enriching Variables *Approximate* *Compose* *Specialize* *Generalize*	**Mode**		**Produced Reformulated Queries**
	Expanded	*Restricted*	
At least one is TRUE	TRUE	FALSE	Exact Enriched
All are FALSE	TRUE	FALSE	Exact
At least one is TRUE	FALSE	TRUE	Exact Enriched, if Exact is EMPTY
All are FALSE	FALSE	TRUE	Exact

The *SemRef* Approach – The Algorithm

SemRef ($Q, P_1, P_2, Co[O_1,O_2]$, MODE, REF_VAR, Q_{exact}, $Q_{enriched}$)

Input: $Q, P_1, P_2, Co[O_1,O_2]$, MODE, REF_VAR

Output: Q_{exact}, $Q_{enriched}$

```
For each conjunctive query Q_k in Q
      Find exact reformulation Q_k_exact of Q_k
      If (one of APPROXIMATE, COMPOSE, SPECIALIZE, GENERALIZE is TRUE)
            Then
                  Find enriched reformulation Q_k_enriched of Q_k
End For;
If (at least one of Q_k_exact ≠ ∅)
      Then
            Build final exact reformulation Q_exact of Q
      Else Q_exact ← ∅
If ((MODE is expanded) or (MODE is restricted and Q_exact is empty)) and
      (at least one of Q_k_enriched ≠ ∅)
      Then
            Build final enriched reformulation Q_enriched of Q
      Else Q_enriched ← ∅
End SemRef;
```

The *SemRef* Approach – Main steps through an Example

Approximate = TRUE + Query Reformulation Mode = Expanded

Q = Faculty ⊔ ¬AssistantProfessor

Q_1 = Faculty

S_1C_1, S_2C_1, Neg_S_2C_1 = { }

S_1C_1 = {Faculty}

S_2C_1= {AdministrativeStaff, Assistant}

Neg_S_2C_1 = { }

Co_{12} for O_1:Faculty
O_1:Faculty $\xrightarrow{\equiv}$ O_2:Faculty
O_1:Faculty $\xrightarrow{\sqsubseteq}$ O_2:Worker
O_1:Faculty $\xrightarrow{\approx}$ O_2:AdministrativeStaff
O_1:Faculty $\xrightarrow{\approx}$ O_2:Assistant

Q_1_exact = [Faculty]

Q_1_enriched = [AdministrativeStaff ⊔ Assistant]

The *SemRef* Approach – Main steps through an example

$Q_2 = \neg AssistantProfessor$

S_1C_1, S_2C_1, $Neg\_S_2C_1 = \{\ \}$

$S_1C_1 = \{\ \}$

$S_2C_1 = \{VisitingProfessor\}$

$Neg\_S_2C_1 = \{AssociateProfessor, FullProfessor\}$

Co_{12} for O_1:AssistantProfessor

O_1:AssistantProfessor $\xrightarrow{\approx}$ O_2:VisitingProfessor

O_1:AssistantProfessor $\xrightarrow{\sqsubseteq}$ O_2:Professor

O_1:AssistantProfessor $\xrightarrow{\perp}$ O_2:AssociateProfessor

O_1:AssistantProfessor $\xrightarrow{\perp}$ O_2:FullProfessor

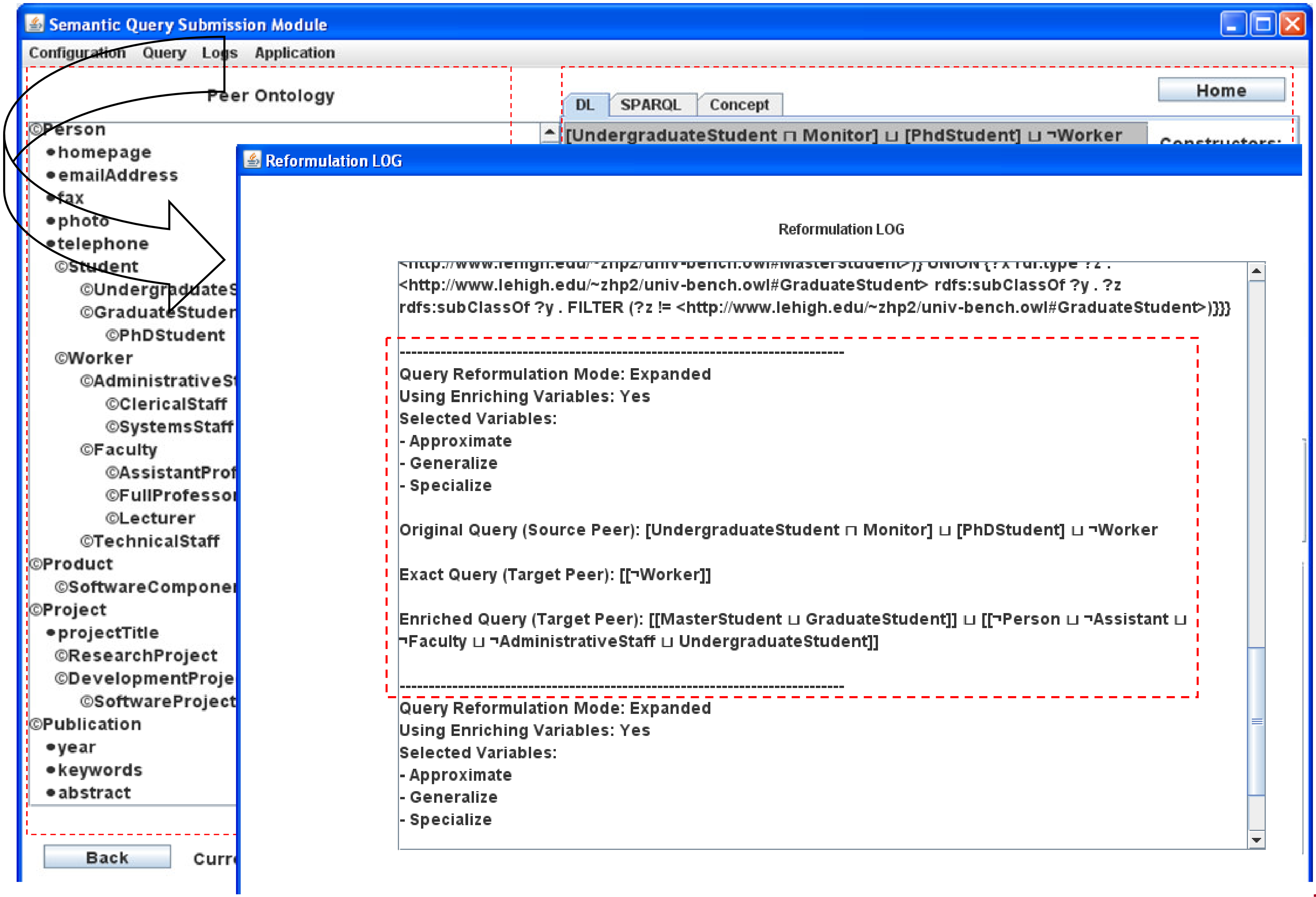
$Q_2\_exact = \emptyset$

$Q_2\_enriched = [\neg VisitingProfessor \sqcup AssociateProfessor \sqcup FullProfessor]$

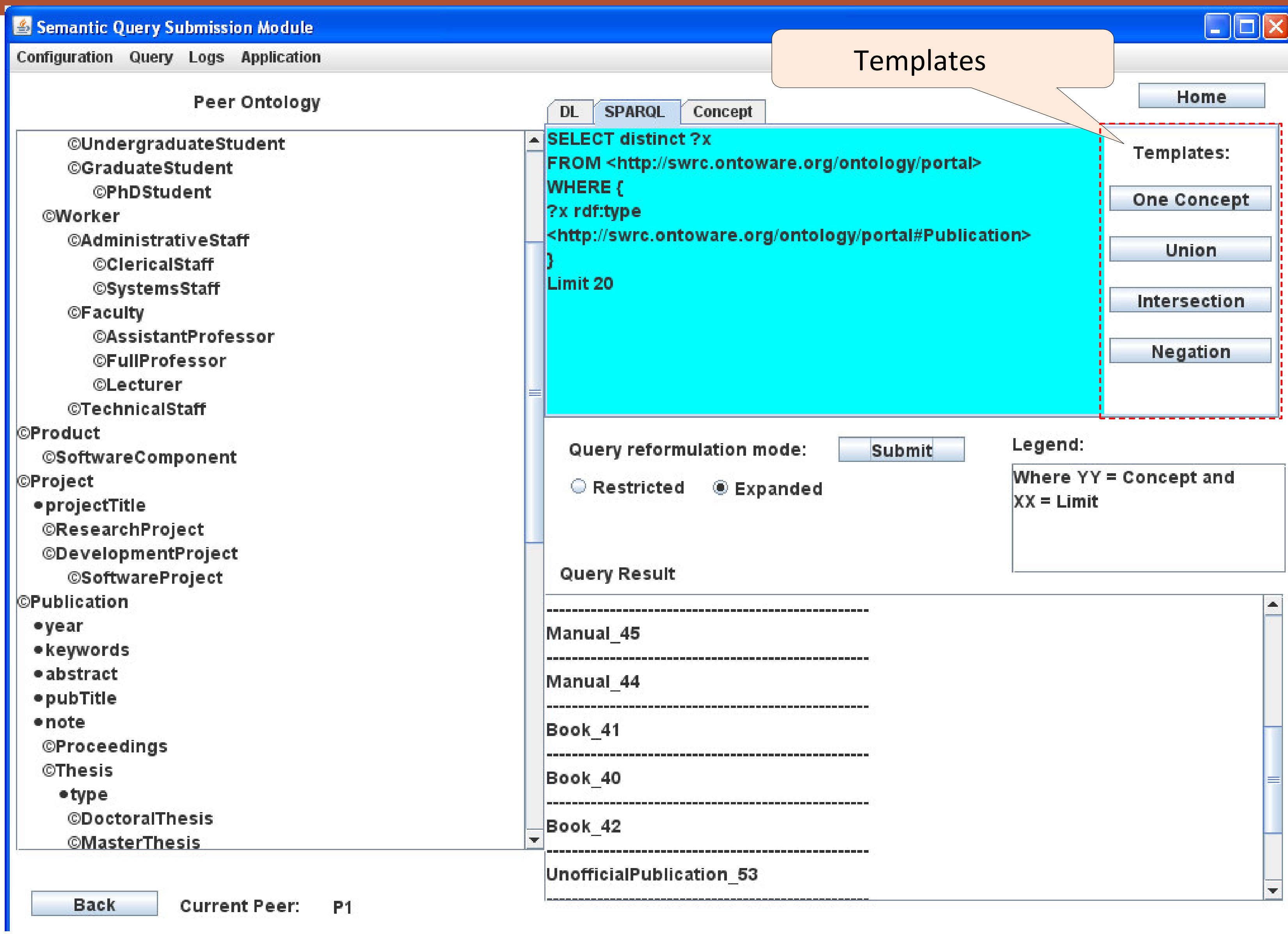
$Q_{exact} = [Faculty]$

$Q_{enriched} = [AdministrativeStaff \sqcup Assistant] \sqcup [\neg VisitingProfessor \sqcup AssociateProfessor \sqcup FullProfessor]$

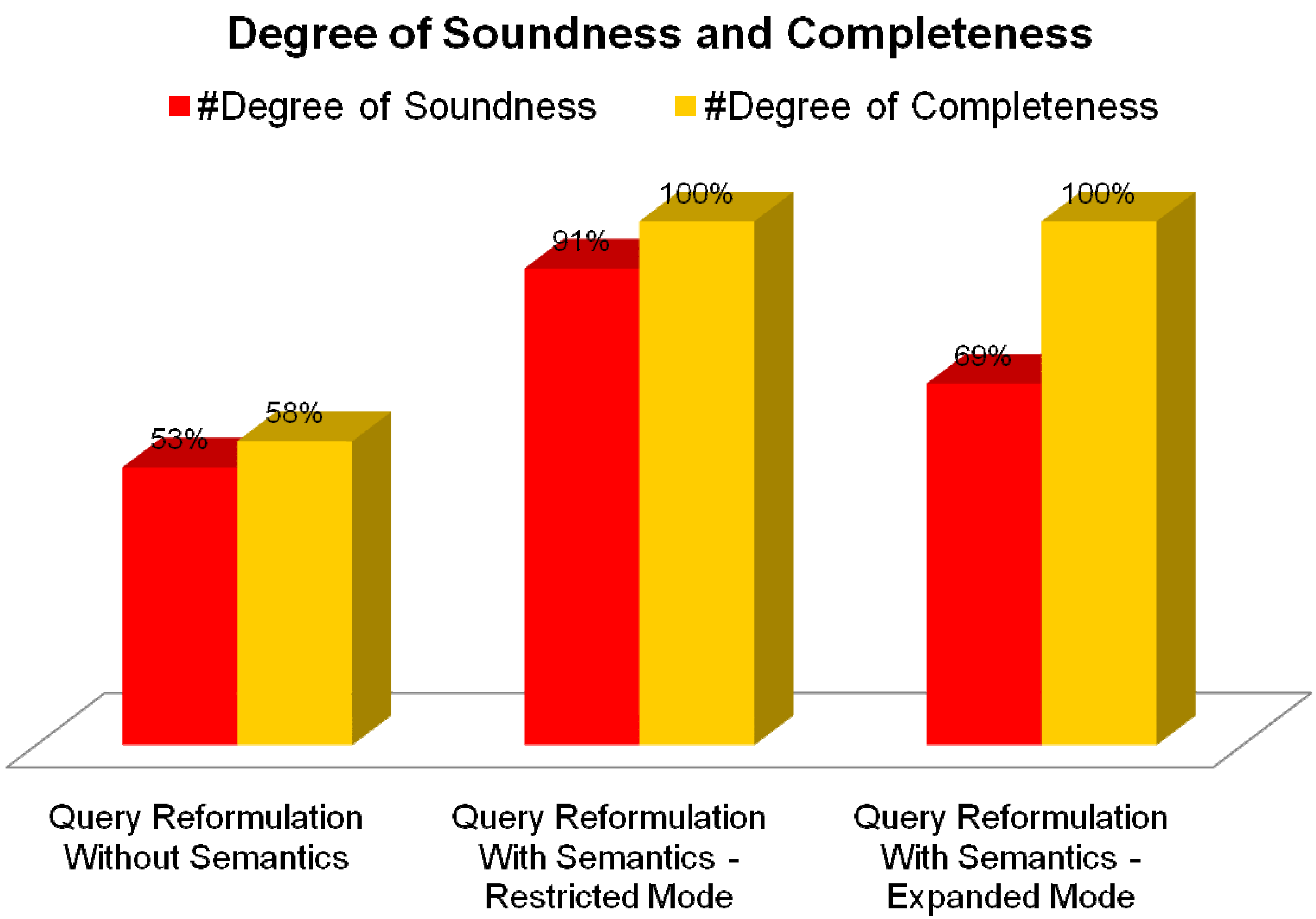
Implementation Issues – *SemRef* Module



Implementation Issues – *SemRef* Module



Experiments and Results



Related Work

Approach	Environment	Representation Model	Formalism	Query Language
[Necib 2007]	Single Databases	Relational	Term Rewriting Systems	SQL
[Kostadinov 2007]	Mediator-based System	Relational	Conjunctive Query	SQL
Piazza [Halevy et al. 2005]	PDMS	Relational and XML	Conjunctive Query	XQuery or Conjunctive Query
OPDMS [Xiao and Cruz 2006]	PDMS	RDF	FOL (First Order Logic)	Conjunctive RQL Query
WTA [Calvanese et al. 2004]	PDMS	Knowledge-based - FOL	FOL (First Order Logic)	FOL Query
SomeRDFS [Adjman et al. 2007]	PDMS/Semantic Web	RDF	DL and FOL	FOL Query
CA [Stuckenschmidt et al. 2005]	Weakly-Structured Environments	Terminological	DL	Boolean Query
SemRef	**Dynamic Distributed Environments; OPDMS**	**OWL**	**DL (Description Logics)**	$\mathcal{ALC}$ / **DL SPARQL**

Related Work

Approach	Mapping/Correspondence Type	Semantics Usage	Reformulation Rules
[Necib 2007]	Equivalence (between database schema and ontology	Ontology	Extension Rules Reduction Rules
[Kostadinov 2007]	LAV mappings	User Profiles	Enrichment Rules Translation rules using LAV
Piazza [Halevy et al. 2005]	Equivalence, Inclusion and Definitional Mappings	Metadata in a Catalog	Translation Rules, using GAV/LAV
OPDMS [Xiao and Cruz 2006]	Equivalence, Broader, Narrower, Union and Intersection	Mapping Ontology	Translation Rules
WTA [Calvanese et al. 2004]	Subsumption, Participation of classes in roles	-----	Translation Rules
SomeRDFS [Adjman et al. 2007]	Equivalence, Inclusion, Disjunction	-----	Translation Rules
CA [Stuckenschmidt et al. 2005]	Equivalence, Specialization, Generalization	Terminological reasoning and query relaxation	Concept Approximation in terms of Lower and Upper Bounds
SemRef	**Equivalence, Specialization, Generalization, Closeness, Disjointness, Aggregation (PartOf) and Aggregation (WholeOf)**	**Domain Ontology, Semantics underlying Correspondences and Contextual Information**	**Exactness and Enrichment Rules**

Conclusions

- This work has presented a **semantic-based query reformulation approach** instantiated in a PDMS that brings together both **query enrichment** and **query reformulation**.
  - ✓ ***SemRef*** approach goes beyond traditional correspondences usage (e.g., ***closeness***).
  - ✓ ***SemRef*** uses ***contextual information***.
  - ✓ ***SemRef*** prioritizes the generation of exact reformulations but it also generates an ***enriched version***
- Experiments carried out have shown that considering semantics, enriched reformulations are generated, providing **additional expanded reformulations.**

Further Work

- **Rules** are being developed to allow reasoning over the contextual information already instantiated in a specific **context ontology** [Souza et al. 2008].
  - ✓ This reasoning might improve the query reformulation and routing processes.
- We will instantiate **additional query reformulation scenarios** which may allow us to work with other different contextual settings and with larger datasets.

Using Semantics to Enhance Query Reformulation in Dynamic Environments

Damires Souza (damires@ifpb.edu.br)
Thiago Arruda (tan@cin.ufpe.br)
Ana Carolina Salgado (acs@cin.ufpe.br)
Patricia Tedesco (pcart@cin.ufpe.br)
Zoubida Kedad (zoubida.kedad }@prism.uvsq.fr)

The *SemRef* Approach – The Algorithm

- ***Exact Reformulation***. A reformulation Q' of a query Q is said to be exact (denoted as Q_{exact}) if each concept (or property) C' of Q' is related to a concept (or property) C of Q by a Co correspondence, where Co $\in$ $\{\xrightarrow{\equiv}\}$.

- ***Enriched Reformulation***. A reformulation Q' of a query Q is said to be enriched ($Q_{enriched}$) if each concept (or property) C' of Q' is related to a concept (or property) C of Q by a Co correspondence,
where Co $\in$ $\{\xrightarrow{\sqsubseteq}, \xrightarrow{\sqsupseteq}, \xrightarrow{\approx}, \xrightarrow{\rhd}, \xrightarrow{\lhd}, \xrightarrow{\perp}\}$.