



13th East-European Conference on
ADVANCES IN DATABASES AND
INFORMATION SYSTEMS

September 7-10, 2009

Riga, Latvia =

Using Semantics to Enhance Query Reformulation in Dynamic Environments

Damires Souza (damires@ifpb.edu.br)

Thiago Arruda (tan@cin.ufpe.br)

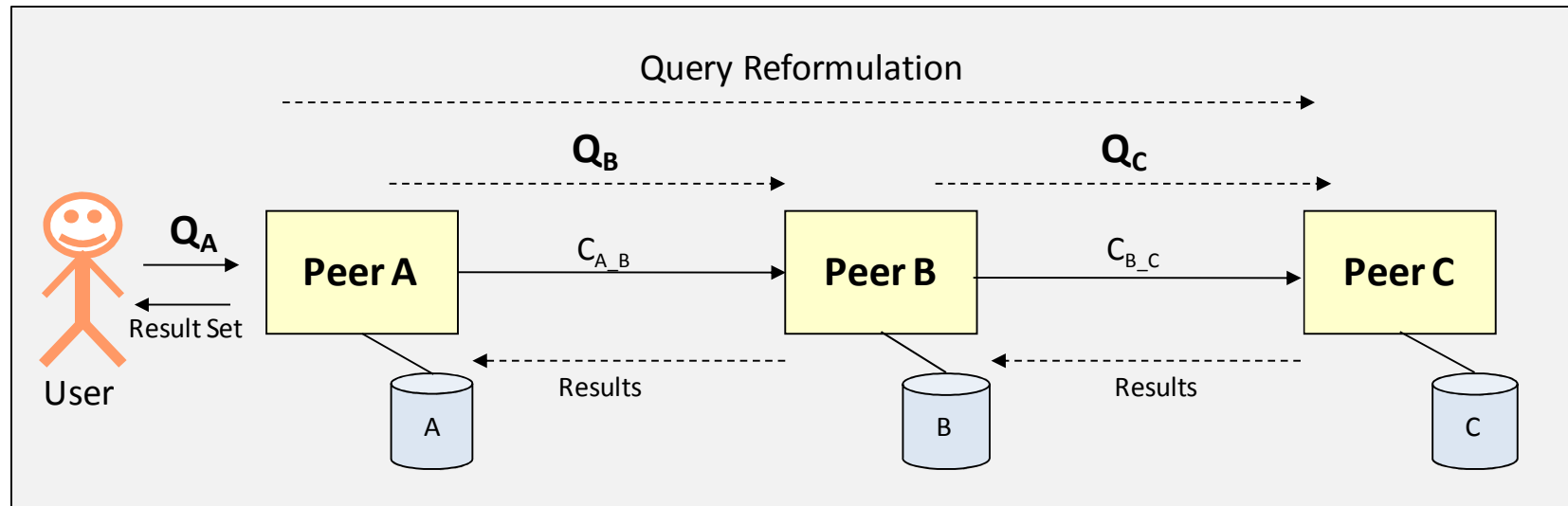
Ana Carolina Salgado (acs@cin.ufpe.br)

Patricia Tedesco (pcart@cin.ufpe.br)

Zoubida Kedad ([zoubida.kedad }](mailto:zoubida.kedad}@prism.uvsq.fr)@prism.uvsq.fr)



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

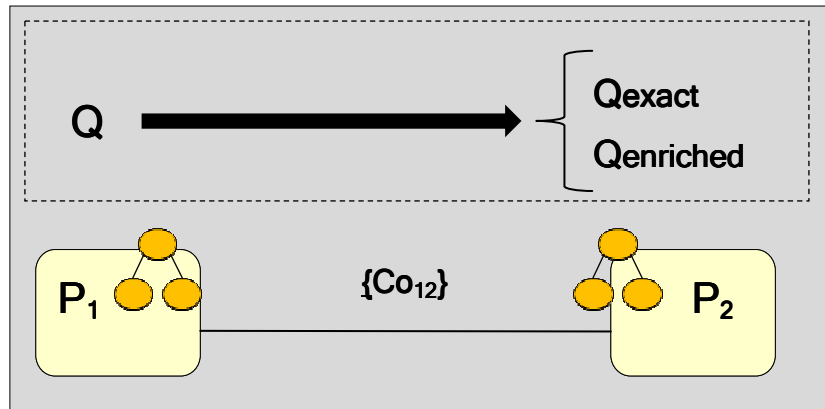
- Using ***semantics*** to enhance the reformulation
 - I. ***Users' preferences, query semantics*** and the ***current status of the environment*** should be taken into account at query reformulation time

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

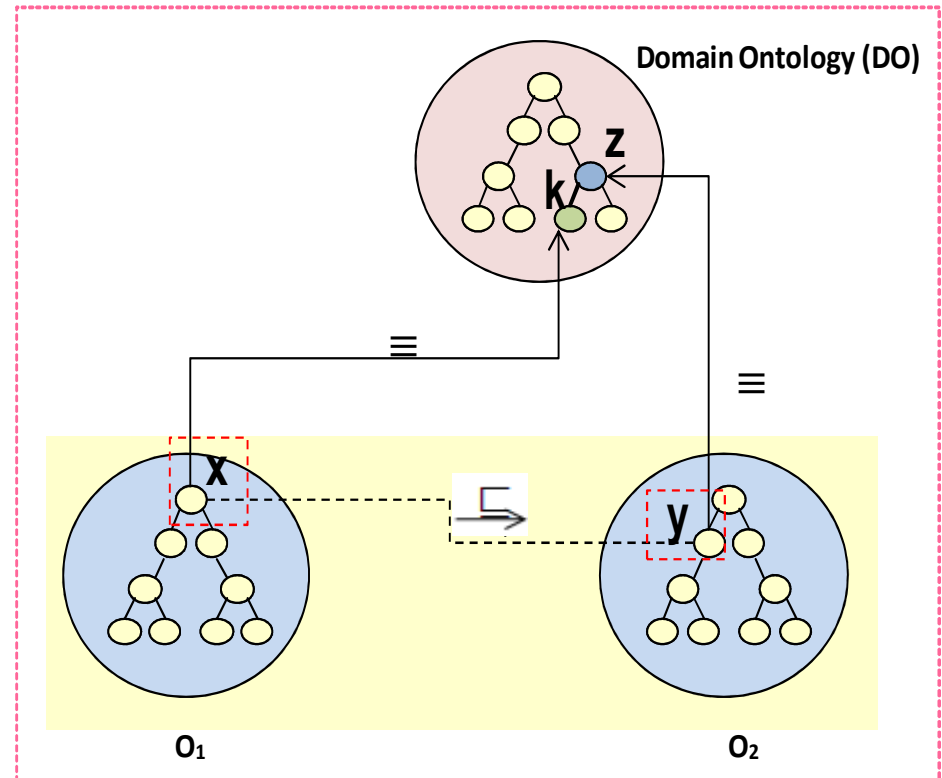


- Goal: find reformulated queries of Q expressed in terms of the concepts of O_2
- Schemas are represented as **ontologies**

- **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 knowledge 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. $O_1:x \xrightarrow{\triangleright} O_2:y$, 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 .

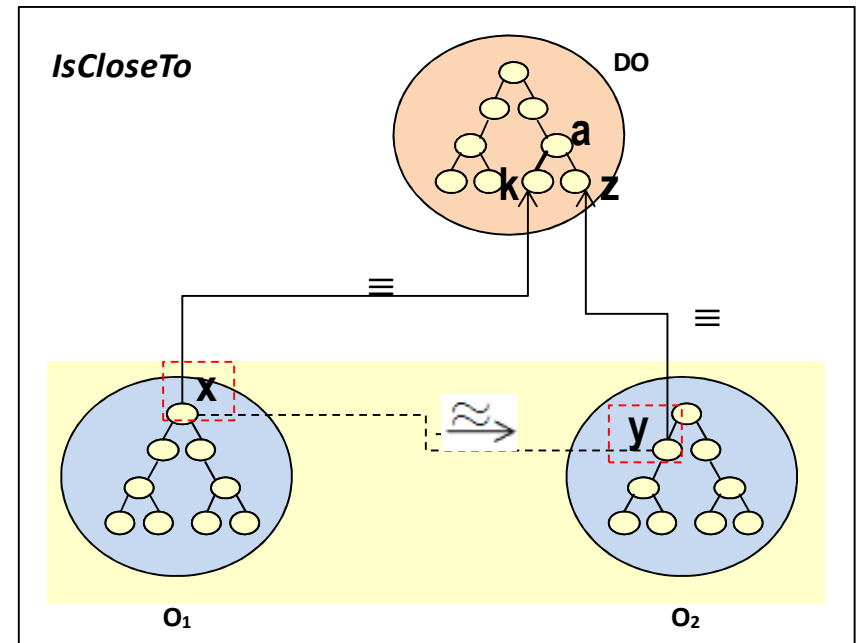
Using Domain Ontologies to Identify Semantic Correspondences

➤ An element $O_1:x$ *isCloseTo* $O_2:y$
 if

$(O_1:x \equiv DO:k \text{ and } O_2:y \equiv DO:z)$ and
 $(DO:k \sqsubseteq DO:a \text{ and } DO:z \sqsubseteq DO:a)$
 and $DO:a \neq \top$ and $DO:a$ *isRigid* and
 $\neg(DO:k \perp DO:z)$ and $(\text{depth}(DO:a, DO:\top) \geq \text{thresholdRoot})$

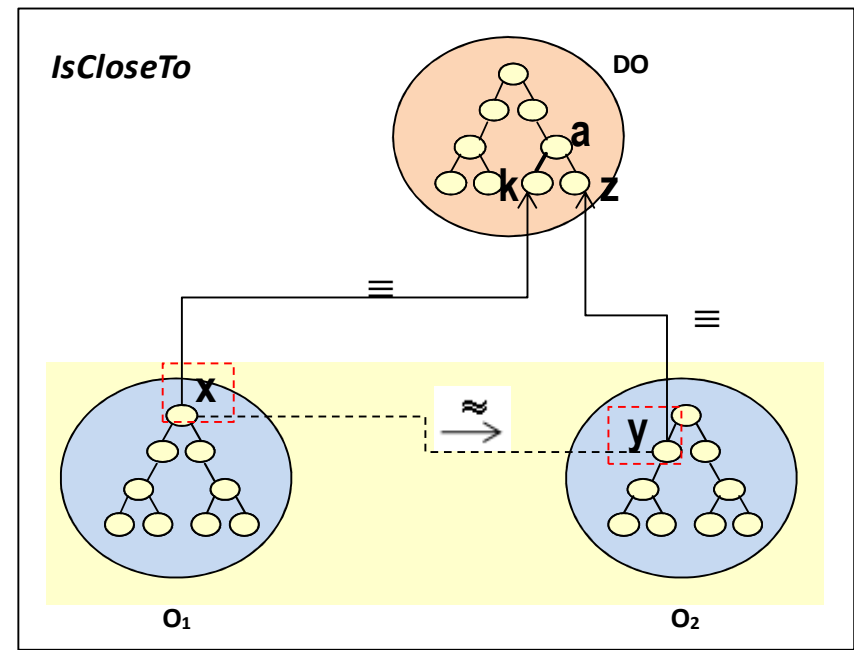
and

$(\text{depth}(DO:k, DO:a) \leq \text{thresholdCommonAncestor}$ and
 $\text{depth}(DO:z, DO:a) \leq \text{thresholdCommonAncestor})$



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**



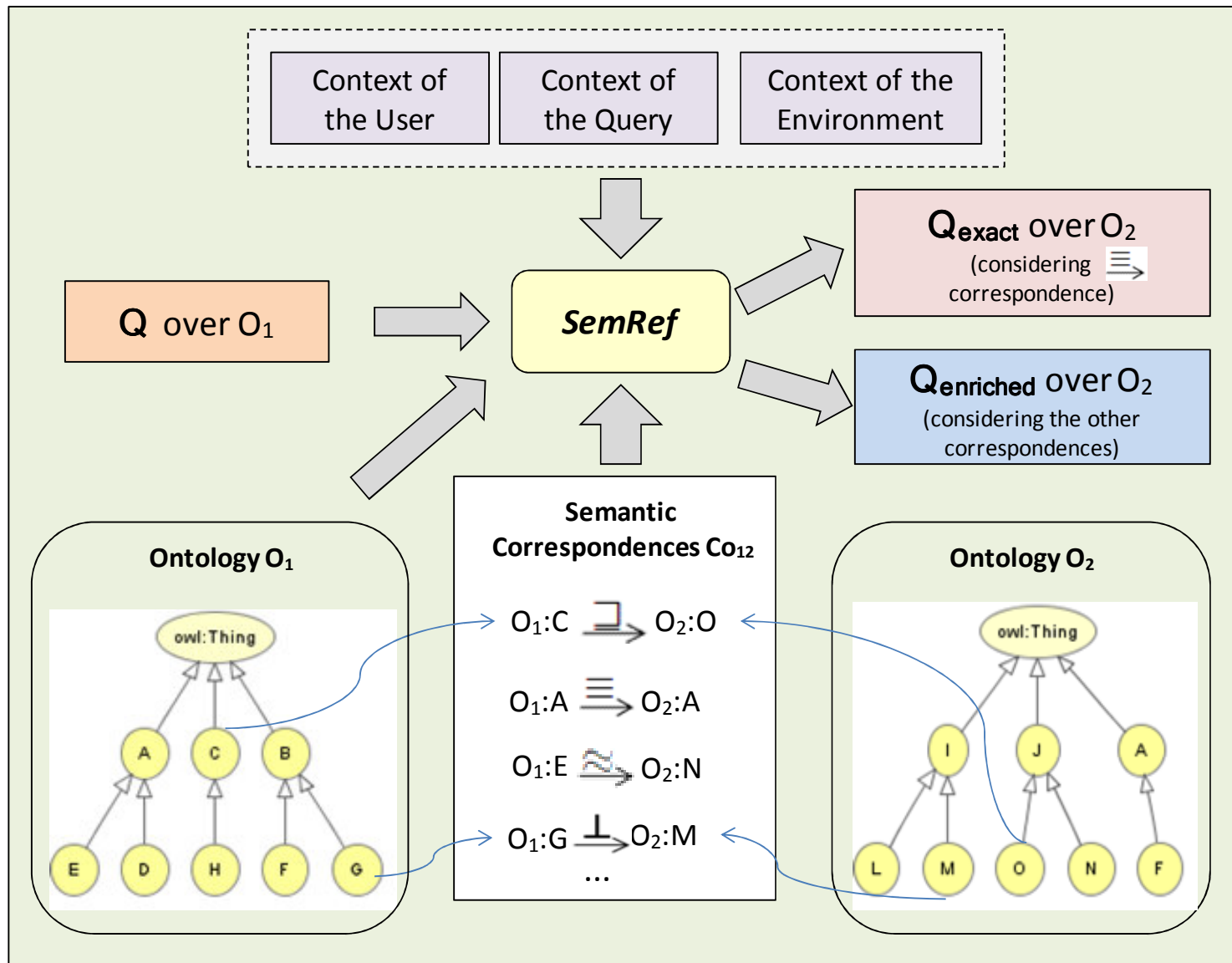
Example:

O_1 .Notebook \approx 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 *ALC-DL*
- **Query Q** is a query expressed over P_i 's ontology, having the following form:

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

$$\text{Where } Q_i = C_1 \sqcap C_2 \sqcap \dots \sqcap C_N$$

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

$$Q_3 = [\text{Teacher} \sqcap \text{Researcher}] \sqcup [\text{Student} \sqcap \text{Researcher}]$$

The *SemRef* Approach – The Algorithm

Enriching Variables <i>Approximate</i> <i>Compose</i> <i>Specialize</i> <i>Generalize</i>	Mode		Produced Reformulated Queries
	<i>Expanded</i>	<i>Restricted</i>	
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: $Q_{exact}, Q_{enriched}$

1. For each conjunctive query Q_k in Q
2. Find exact reformulation Q_{k_exact} of Q_k
3. If (one of APPROXIMATE, COMPOSE, SPECIALIZE, GENERALIZE is TRUE)
4. Then
5. Find enriched reformulation $Q_{k_enriched}$ of Q_k
6. End For;
7. If (at least one of $Q_{k_exact} \neq \emptyset$)
8. Then
9. Build final exact reformulation Q_{exact} of Q
10. Else $Q_{exact} \leftarrow \emptyset$
11. If ((MODE is expanded) or (MODE is restricted and Q_{exact} is empty)) and
12. (at least one of $Q_{k_enriched} \neq \emptyset$)
13. Then
14. Build final enriched reformulation $Q_{enriched}$ of Q
15. Else $Q_{enriched} \leftarrow \emptyset$
16. End *SemRef*,

The *SemRef* Approach – Main steps through an Example

Approximate = TRUE + Query Reformulation Mode = Expanded

$Q = \text{Faculty} \sqcup \neg \text{AssistantProfessor}$

$Q_1 = \text{Faculty}$

$S_1C_1, S_2C_1, \text{Neg_}S_2C_1 = \{ \}$

$S_1C_1 = \{ \text{Faculty} \}$

$S_2C_1 = \{ \text{AdministrativeStaff}, \text{Assistant} \}$

$\text{Neg_}S_2C_1 = \{ \}$

Co₁₂ for O₁:Faculty

O₁:Faculty \equiv O₂:Faculty

O₁:Faculty \sqsupseteq O₂:Worker

O₁:Faculty \approx O₂:AdministrativeStaff

O₁:Faculty \approx O₂:Assistant

Q_{1_exact} = [Faculty]

Q_{1_enriched} = [AdministrativeStaff \sqcup Assistant]

The *SemRef* Approach – Main steps through an example

$Q_2 = \neg\text{AssistantProfessor}$

$S_1C_1, S_2C_1, \text{Neg_}S_2C_1 = \{ \}$

$S_1C_1 = \{ \}$

$S_2C_1 = \{\text{VisitingProfessor}\}$

$\text{Neg_}S_2C_1 = \{\text{AssociateProfessor}, \text{FullProfessor}\}$

Co₁₂ for O₁:AssistantProfessor

O₁:AssistantProfessor $\xrightarrow{\approx}$ O₂:VisitingProfessor

O₁:AssistantProfessor $\xrightarrow{\sqsubseteq}$ O₂:Professor

O₁:AssistantProfessor $\xrightarrow{\perp}$ O₂:AssociateProfessor

O₁:AssistantProfessor $\xrightarrow{\perp}$ O₂:FullProfessor

Q_{2_exact} = \emptyset

Q_{2_enriched} = [$\neg\text{VisitingProfessor} \sqcup \text{AssociateProfessor} \sqcup \text{FullProfessor}$]

Q_{exact} = [Faculty]

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

Implementation Issues – *SemRef* Module

The screenshot displays the Semantic Query Submission Module interface. On the left, a tree view shows the Peer Ontology structure, including classes like Person, Student, Worker, and Faculty. A red dashed box highlights the Peer Ontology section. On the right, a Reformulation LOG window is open, showing the original query and its reformulation. The log includes the following text:

```
http://www.lehigh.edu/~zhp2/univ-bench.owl#MasterStudent}} UNION {?x rdf:type ?z .  
<http://www.lehigh.edu/~zhp2/univ-bench.owl#GraduateStudent> rdfs:subClassOf ?y . ?z  
rdfs:subClassOf ?y . FILTER (?z != <http://www.lehigh.edu/~zhp2/univ-bench.owl#GraduateStudent>)}}}
```

Query Reformulation Mode: Expanded
Using Enriching Variables: Yes
Selected Variables:
- Approximate
- Generalize
- Specialize

Original Query (Source Peer): [UndergraduateStudent \sqcap Monitor] \sqcup [PhDStudent] \sqcup \neg Worker

Exact Query (Target Peer): [[\neg Worker]]

Enriched Query (Target Peer): [[MasterStudent \sqcup GraduateStudent]] \sqcup [[\neg Person \sqcup \neg Assistant \sqcup \neg Faculty \sqcup \neg AdministrativeStaff \sqcup UndergraduateStudent]]

Query Reformulation Mode: Expanded
Using Enriching Variables: Yes
Selected Variables:
- Approximate
- Generalize
- Specialize

Implementation Issues – *SemRef* Module

The screenshot displays the 'Semantic Query Submission Module' interface. On the left, a tree view shows the 'Peer Ontology' with categories like UndergraduateStudent, GraduateStudent, Worker, Faculty, Product, Project, and Publication. The main area is divided into three tabs: DL, SPARQL, and Concept. The SPARQL tab is active, showing a query: `SELECT distinct ?x FROM <http://swrc.ontoware.org/ontology/portal> WHERE { ?x rdf:type <http://swrc.ontoware.org/ontology/portal#Publication> } Limit 20`. A callout box labeled 'Templates' points to a 'Templates' panel on the right, which contains buttons for 'One Concept', 'Union', 'Intersection', and 'Negation'. Below the query, there is a 'Query reformulation mode' section with 'Restricted' and 'Expanded' options, and a 'Legend' section with the text 'Where YY = Concept and XX = Limit'. The 'Query Result' section shows a list of results: Manual_45, Manual_44, Book_41, Book_40, Book_42, and UnofficialPublication_53. At the bottom, there is a 'Back' button and the text 'Current Peer: P1'.

Semantic Query Submission Module

Configuration Query Logs Application

Peer Ontology

- UndergraduateStudent
- GraduateStudent
 - PhDStudent
- Worker
 - AdministrativeStaff
 - ClericalStaff
 - SystemsStaff
- Faculty
 - AssistantProfessor
 - FullProfessor
 - Lecturer
 - TechnicalStaff
- Product
 - SoftwareComponent
- Project
 - projectTitle
 - ResearchProject
 - DevelopmentProject
 - SoftwareProject
- Publication
 - year
 - keywords
 - abstract
 - pubTitle
 - note
 - Proceedings
 - Thesis
 - type
 - DoctoralThesis
 - MasterThesis

DL SPARQL Concept

SELECT distinct ?x
FROM <http://swrc.ontoware.org/ontology/portal>
WHERE {
?x rdf:type
<http://swrc.ontoware.org/ontology/portal#Publication>
}
Limit 20

Home

Templates:

- One Concept
- Union
- Intersection
- Negation

Query reformulation mode: Submit

Restricted Expanded

Legend:
Where YY = Concept and
XX = Limit

Query Result

Manual_45

Manual_44

Book_41

Book_40

Book_42

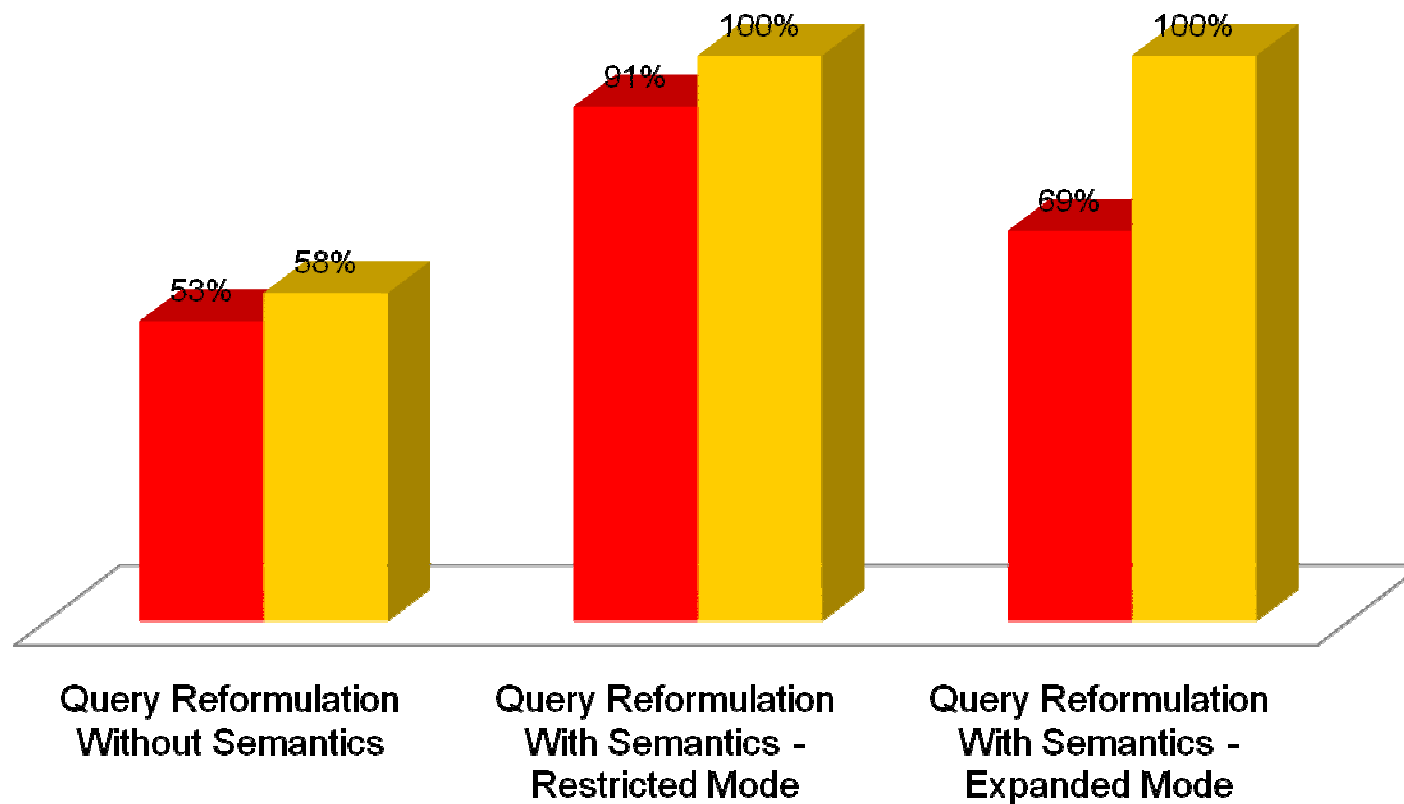
UnofficialPublication_53

Back Current Peer: P1

Experiments and Results

Degree of Soundness and Completeness

■ #Degree of Soundness ■ #Degree of Completeness



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
<i>SemRef</i>	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.



13th East-European Conference on
ADVANCES IN DATABASES AND
INFORMATION SYSTEMS

September 7-10, 2009

Riga, Latvia =

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](mailto: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 $Co \in \{\equiv\}$.
- ***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 $Co \in \{\sqsubseteq, \sqsupseteq, \approx, \triangleright, \triangleleft, \perp\}$.