

Accès aux données à un niveau conceptuel avec Ontop

Benjamin Cogrel

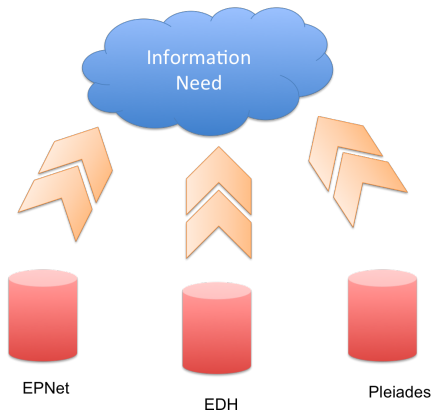
KRDB Research Centre for Knowledge and Data
Université Libre de Bozen-Bolzano, Italie

 **unibz**
Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen-Bolzano

Optique[®]

Journées Ontologie en Sciences Humaines et Sociales,
Tours, le 9 novembre 2015

Un cas d'usage d'Ontop : le projet EPNET



Bases de données relationnelles

- EPNET et EDH : objets antiques (amphores, monuments, etc.)
- Pleiades : informations géographiques sur des sites archéologiques

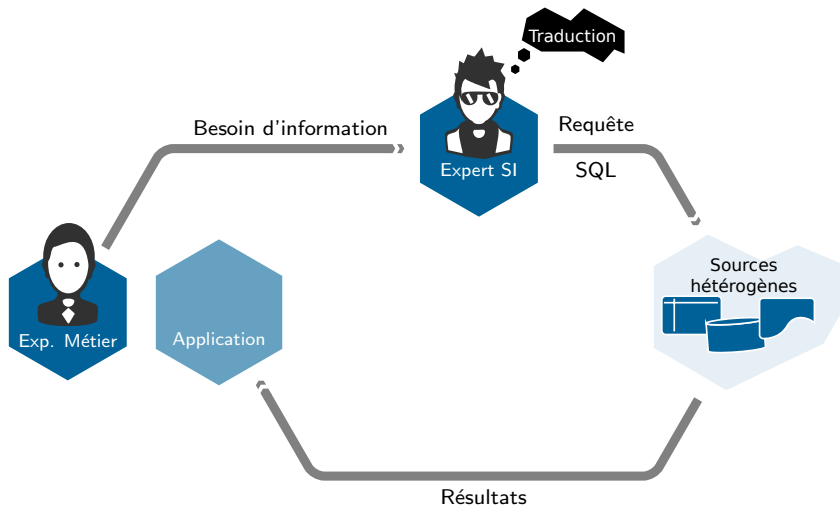
Besoins d'information

Donne moi toutes les transcriptions des inscriptions écrites sur des amphores retrouvées dans telle ville



Définition d'une nouvelle requête SQL

Sans système OBDA



Fédération SQL

Base de données fédérées

- Sources de données hétérogènes. . .
- . . . fédérées en une seule base de données virtuelle

Systèmes supportés par Ontop

- Teiid (LGPL) : <http://teiid.jboss.org/>
- Exareme (licence libre, partenaire Optique) : <http://exareme.org/>

Exemple de requête SQL (extrait)

```

SELECT *
FROM (
SELECT QVIEW8."simplifiedtranscription" AS "transcription"
FROM
Inscription QVIEW1,
InformationCarrier QVIEW2,
LinguisticObject_Inscription QVIEW3,
Finding QVIEW4,
Activity_Location QVIEW5,
Location_FallsWithin QVIEW6,
GeographicLocation QVIEW7,
LinguisticObject QVIEW8
WHERE
(QVIEW1."carrier" = QVIEW2."id") AND
QVIEW1."id" IS NOT NULL AND
QVIEW1."carrier" IS NOT NULL AND
(QVIEW1."id" = QVIEW3."inscription") AND
QVIEW3."linguisticobject" IS NOT NULL AND
(QVIEW2."finding" = QVIEW4."id") AND
(QVIEW2."finding" = QVIEW5."activity") AND
QVIEW5."location" IS NOT NULL AND
(QVIEW5."location" = QVIEW6."location1") AND
QVIEW6."location2" IS NOT NULL AND
(QVIEW6."location2" = QVIEW7."id") AND
(QVIEW7."name" = 'Barcelona') AND
(QVIEW3."linguisticobject" = QVIEW8."id") AND
QVIEW8."simplifiedtranscription" IS NOT NULL
UNION ALL

```

```

SELECT QVIEW1."taggedtranscription" AS "transcription"
FROM
edh.inscriptions QVIEW1,
edh.findingPlaces QVIEW2,
edh.places QVIEW3,
edh.places_not_in_corpus2 QVIEW4,
edh.places QVIEW5,
edh.places QVIEW6
WHERE
(QVIEW1."objtypeurl" =
'http://www.eagle-network.eu/voc/objtyp/lod/140') AND
QVIEW1."id" IS NOT NULL AND
QVIEW1."taggedtranscription" IS NOT NULL AND
(QVIEW1."ancientfindingplace" = QVIEW2."id") AND
(QVIEW2."spot" = QVIEW3."id") AND
(QVIEW3."numberid" = QVIEW4."id") AND
QVIEW3."pleiadesid" IS NULL AND
QVIEW3."numberid" IS NOT NULL AND
(QVIEW3."numberid" = QVIEW5."numberid") AND
(QVIEW3."numberid" = QVIEW6."numberid") AND
(QVIEW6."name" = 'Barcelona')
) SUB_QVIEW

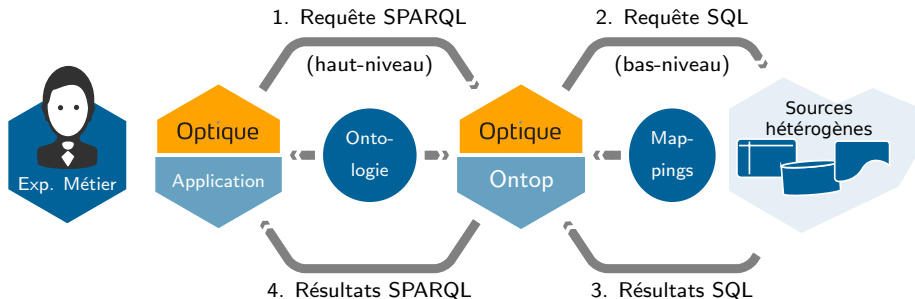
```

[... 12 autres sous-requêtes ...]

Automatisation avec la plateforme Optique

Ontology-Based Data Access (OBDA)

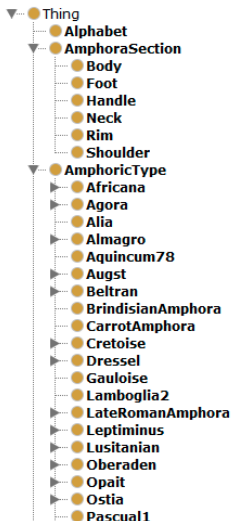
Accès aux données à un niveau conceptuel



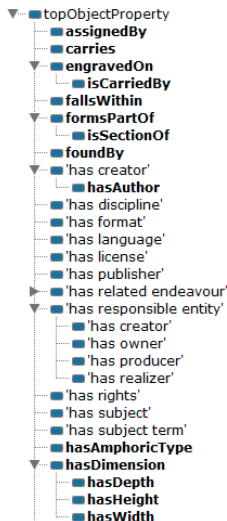
Extrait de l'ontologie du projet EPNET

Hiérarchies

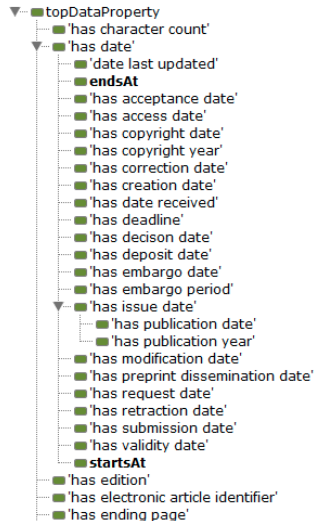
Classes:



Object properties:



Data properties:



Requête SPARQL de haut-niveau

Niveau conceptuel

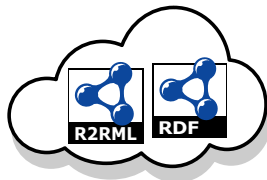
- Indépendant du stockage
- Vue homogène
- Terminologie du domaine formalisée (ontologie)
- Requêtes concises

```
SELECT ?transcription
WHERE {
    ?inscription a :Inscription ;
                :engravedOn ?amphora ;
                :isTranscribedBy ?trans .
    ?amphora a :Amphora ;
            :hasFindingPlace ?findingPlace .
    ?findingPlace :fallsWithin ?city .
    ?city dcterms:title "Barcelona" .
    ?trans :hasTranscription ?transcription .
}
```

SPARQL et les bases de données relationnelles

Transformation d'une BD relationnelle en un graphe RDF

- Pour être requêtée avec SPARQL
- Mappings SQL-RDF (R2RML)
- Contraintes de l'ontologie (opt.)
- Virtuelle ou matérielle



Graphe RDF matérialisé

- Matérialisation du graphe RDF saturé
- Maintenance
- + Profils d'ontologie expressifs (comme OWL 2 RL)

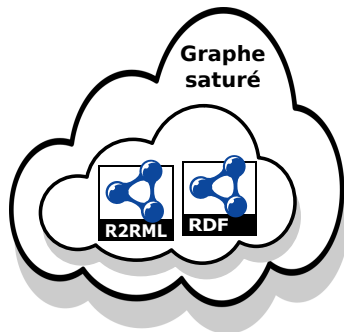
Graphe RDF virtuel

- Reformulation de requête
- + Aucune matérialisation
- Profil d'ontologie moins expressif (OWL 2 QL) *

SPARQL et les bases de données relationnelles

Transformation d'une BD relationnelle en un graphe RDF

- Pour être requêtée avec SPARQL
- Mappings SQL-RDF (R2RML)
- Contraintes de l'ontologie (opt.)
- Virtuelle ou matérielle



Graphe RDF matérialisé

- Matérialisation du graphe RDF saturé
- Maintenance
- + Profils d'ontologie expressifs (comme OWL 2 RL)

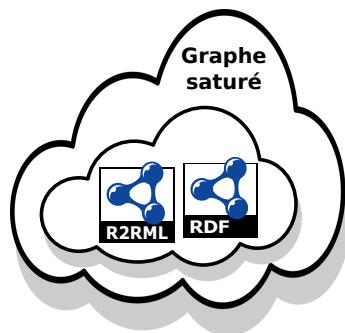
Graphe RDF virtuel

- Reformulation de requête
- + Aucune matérialisation
- Profil d'ontologie moins expressif (OWL 2 QL) *

SPARQL et les bases de données relationnelles

Transformation d'une BD relationnelle en un graphe RDF

- Pour être requêtée avec SPARQL
- Mappings SQL-RDF (R2RML)
- Contraintes de l'ontologie (opt.)
- Virtuelle ou matérielle



Graphe RDF matérialisé

- Matérialisation du graphe RDF saturé
- Maintenance
- + Profils d'ontologie expressifs (comme OWL 2 RL)

Graphe RDF virtuel

- Reformulation de requête
- + Aucune matérialisation
- Profil d'ontologie moins expressif (OWL 2 QL) *

(*) Inclut toutefois un mécanisme d'inférence absent du profil OWL 2 RL

Mapping SQL-RDF

Format natif d'Ontop (similaire au standard R2RML)

AmphoraT

amId	place	inscription
A24	Latium	'aaaa'
A52	Barcelona	'bbbb'

Source (SQL)

```
SELECT amId, inscription  
FROM AmphoraT
```

Cible (RDF, Turtle-like)

```
:db1/{amId} a :Amphora ;  
             :hasInscription "{inscription}"^^xsd:string .
```

Mappings SQL-RDF

Rendre les informations plus accessibles

Classification de périodes par gouvernement

- Exemple : le gouvernement *Caligula* est défini par la période 37-41
- Permet de requêter un gouvernement plutôt que des dates

M_63

```
:Amphora-{ic_id} :producedAt :Tiberius-Government .  
select ic.id as ic_id  
from  
  InformationCarrier ic  
  join  
  Producing n
```

M_64

```
:Amphora-{ic_id} :producedAt :Caligula-Government .  
select ic.id as ic_id  
from  
  InformationCarrier ic  
  join  
  Producing n
```

M_65

```
:Amphora-{ic_id} :producedAt :Claudius-Government .  
select ic.id as ic_id  
from  
  InformationCarrier ic  
  join  
  Producing n
```

M_66

```
:Amphora-{ic_id} :producedAt :Nero-Government .  
select ic.id as ic_id  
from  
  InformationCarrier ic  
  join  
  Producing n
```

Alignement des données

Réutilisation d'IRIs

- IRIs construites à partir de données pivots
- Attention à l'indexation !

Tables de correspondances (expérimental)

- Propriétés `owl:sameAs` spécifiées par les mappings
- Patrons d'IRIs différents entre les données alignées
- Phase de réécriture de la requête SPARQL
- *Ontology-based Integration of Cross-linked Datasets* (ISWC 15)

Reformulation de requête



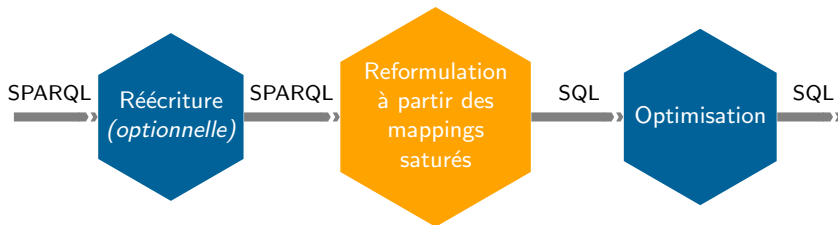
Rôle de l'ontologie OWL 2 QL

- Marginal : réécriture de la requête SPARQL (*cas très particuliers*)
- Principal : saturation des mappings (*hors-ligne*)

Perspective : mieux capturer le pouvoir expressif des mappings

- Inférence de mappings supplémentaires
- Autres formes de contraintes ontologiques (SWRL restreint)

Reformulation de requête



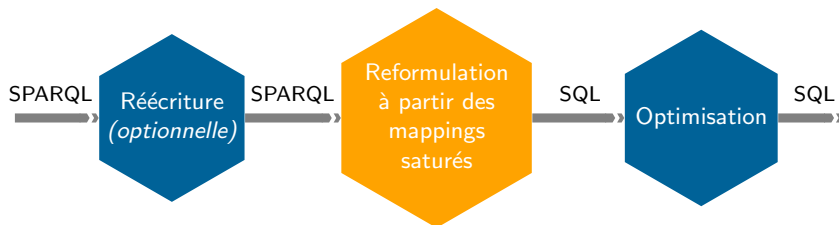
Rôle de l'ontologie OWL 2 QL

- Marginal : réécriture de la requête SPARQL (*cas très particuliers*)
- Principal : saturation des mappings (*hors-ligne*)

Perspective : mieux capturer le pouvoir expressif des mappings

- Inférence de mappings supplémentaires
- Autres formes de contraintes ontologiques (SWRL restreint)

Reformulation de requête



Rôle de l'ontologie OWL 2 QL

- Marginal : réécriture de la requête SPARQL (*cas très particuliers*)
- Principal : saturation des mappings (*hors-ligne*)

Perspective : mieux capturer le pouvoir expressif des mappings

- Inférence de mappings supplémentaires
- Autres formes de contraintes ontologiques (SWRL restreint)

Profil d'ontologie OWL 2 QL

Principales fonctionnalités

- Hiérarchies de classes (`rdfs:subClassOf`) et de propriétés (`rdfs:subPropertyOf`)
- Signature de propriétés (`rdfs:domain`, `rdfs:range`)
- Propriétés inverses (`owl:inverseOf`)
- Disjonction de classes (`owl:disjointWith`)
- Participation obligatoire (avancé)

Manques

- Identité d'individus (`owl:sameAs`)
- Contraintes de cardinalité (propriété fonctionnelle, etc.)
- Et bien d'autres

En résumé

- Ontologies légères
- Un peu plus que RDFS
- Traductibilité garantie vers SQL

Optimisation de la requête SQL

Objectif : obtenir une requête SQL...

- Proche de celles écrites par des humains
- Adaptée aux planificateurs existants

Optimisations structurelles

- Jointures d'unions vers unions de jointures
- Décomposition des IRIs pour faciliter les jointures

Optimisations sémantiques

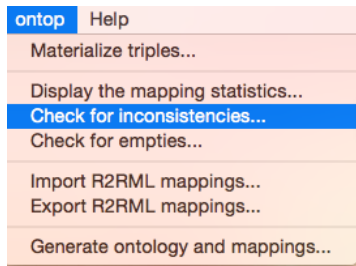
- Élimination des unions et des jointures redondantes
- À partir de contraintes fonctionnelles

Contraintes fonctionnelles

- Clés primaires, uniques ou étrangères
- Implicites dans les processus métiers (*Statoil*)
- Vitales pour la reformulation de requête !

Détection d'incohérences

Vérifier que certaines classes restent disjointes dans le graphe RDF virtuel



Conclusion

Ontop

- Logiciel libre (Apache 2)
- Extension Protégé
- Point d'accès SPARQL

Quelques expérimentations

- owl:sameAs via les mappings
- Support partiel de SWRL
- Agrégation SPARQL
- NoSQL

Intégration

- Plateforme Optique (FluidOps)
- Stardog 4.0 (graphes virtuels)

Liens

- Github : [ontop/ontop](https://github.com/ontop/ontop)
- ontop4obda@googlegroups.com
- Twitter : @ontop4obda
- <http://optique-project.eu>

Démonstration

<http://136.243.8.213/epnet-pleiades-edh/>

Conclusion

Ontop

- Logiciel libre (Apache 2)
- Extension Protégé
- Point d'accès SPARQL

Quelques expérimentations

- owl:sameAs via les mappings
- Support partiel de SWRL
- Agrégation SPARQL
- NoSQL

Intégration

- Plateforme Optique (FluidOps)
- Stardog 4.0 (graphes virtuels)

Liens

- Github : [ontop/ontop](https://github.com/ontop/ontop)
- ontop4obda@googlegroups.com
- Twitter : @ontop4obda
- <http://optique-project.eu>

Démonstration

<http://136.243.8.213/epnet-pleiades-edh/>