Ontology-based Data Access: Theory and Practice

Guohui Xiao

KRDB Research Centre

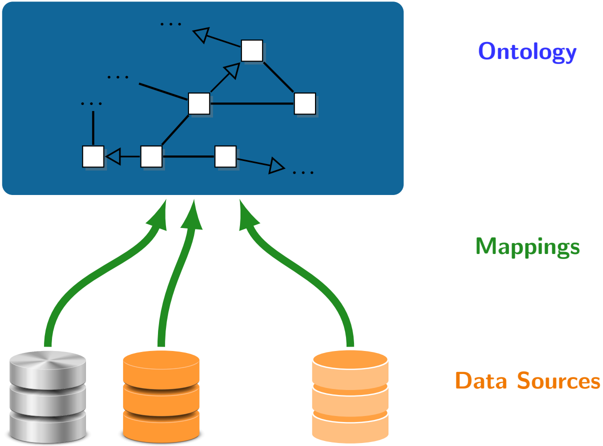
Free University of Bozen-Bolzano

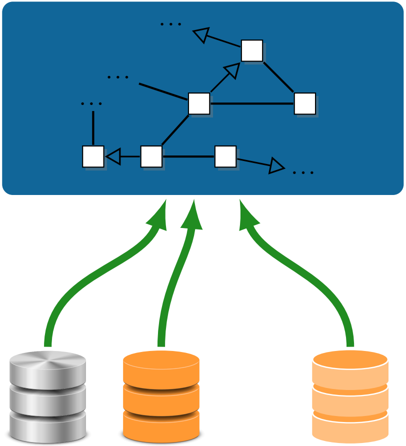
Roman Kontchakov

Department of Computer Science & Inf. Systems

Birkbeck, University of London

`http://ontop.inf.unibz.it/ijcai-2018-tutorial`


...
...
...
...
Ontology
Mappings
Data Sources

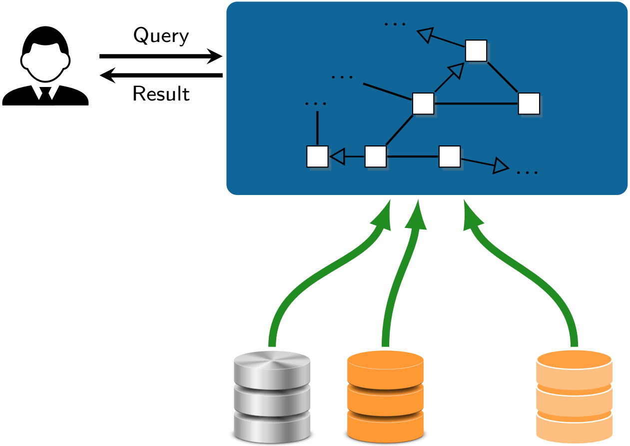


Ontology
provides uniform common vocabulary and conceptual view of data

Mappings
how to populate the ontology from the data

Data Sources
external and heterogeneous

Ontology-based data integration (OBDI)

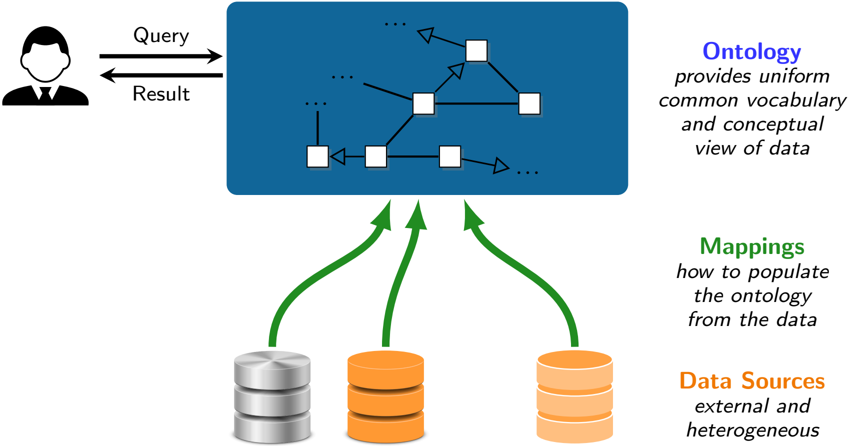


Ontology
provides uniform common vocabulary and conceptual view of data

Mappings
how to populate the ontology from the data

Data Sources
external and heterogeneous

Ontology-based data integration (OBDI)



- OBDI is an extension of OBDA in which data are in multiple datasources
- The conceptual architectures of OBDA and OBDI are the same
- Query execution relies on a database federation engine, e.g., Teiid, Exareme, Denodo

The information about one real-world entity can be distributed over several data sources.

Entity resolution

Understand which records actually represent the same real world entity.

We assume that this information is available and/or known to the integration system designer.

Integrated querying

Answer queries that require to integrate data items representing the same entity, but coming from different data sources.

This is the topic .

The information about one real-world entity can be distributed over several data sources.

Entity resolution

Understand which records actually represent the same real world entity.

We assume that this information is available and/or known to the integration system designer.

Integrated querying

Answer queries that require to integrate data items representing the same entity, but coming from different data sources.

This is the topic .

The information about one real-world entity can be distributed over several data sources.

Entity resolution

Understand which records actually represent the same real world entity.

We assume that this information is available and/or known to the integration system designer.

Integrated querying

Answer queries that require to integrate data items representing the same entity, but coming from different data sources.

This is the topic .

OBDI – Example

Consider two databases nat and corp with one table each (keys in blue):

nat.wellbore		
name	wbField	opPurpose
2-1	BLANE	WILDCAT
3-1		WILDCAT
3-10	OSELVAR	APPRAISAL
4-2	EKOFISK	WILDCAT

corp.drillingops		
name	driStDt	reason
NO-2-1	20-03-1989	WILDCAT
NO-3-1	06-07-1968	WILDCAT
NO-3-A	22-07-2011	PRODUCTION
NO-4-2	18-09-1969	

Mapping assertions

1. SELECT name, wbField, opPurpose FROM nat.wellbore
   $\leadsto$:NatWB/{name} :inField {wbField} ; :purpose {opPurp} .
2. SELECT name, driStDt, reason FROM corp.drillingops
   $\leadsto$:CorpWB/{name} :drillingStarted {driStDt} ; :purpose {reason} .

Some triples in the ABox defined by the DBs and mapping

:NatWB/2-1	:inField	BLANE .
:NatWB/2-1	:purpose	WILDCAT .
:CorpWB/NO-2-1	:drillingStarted	20-03-1989 .
:CorpWB/NO-2-1	:purpose	WILDCAT .

Consider two databases `nat` and `corp` with one table each (keys in blue):

nat.wellbore		
name	wbField	opPurpose
2-1	BLANE	WILDCAT
3-1		WILDCAT
3-10	OSELVAR	APPRAISAL
4-2	EKOFISK	WILDCAT

corp.drillingops		
name	driStDt	reason
NO-2-1	20-03-1989	WILDCAT
NO-3-1	06-07-1968	WILDCAT
NO-3-A	22-07-2011	PRODUCTION
NO-4-2	18-09-1969	

Mapping assertions

1. `SELECT name, wbField, opPurpose FROM nat.wellbore`
   $\rightsquigarrow$ `:NatWB/{name} :inField {wbField} ; :purpose {opPurp} .`
2. `SELECT name, driStDt, reason FROM corp.drillingops`
   $\rightsquigarrow$ `:CorpWB/{name} :drillingStarted {driStDt} ; :purpose {reason} .`

Some triples in the ABox defined by the DBs and mapping

:NatWB/2-1	:inField	BLANE .
:NatWB/2-1	:purpose	WILDCAT .
:CorpWB/NO-2-1	:drillingStarted	20-03-1989 .
:CorpWB/NO-2-1	:purpose	WILDCAT .

Consider two databases nat and corp with one table each (keys in blue):

nat.wellbore		
name	wbField	opPurpose
2-1	BLANE	WILDCAT
3-1		WILDCAT
3-10	OSELVAR	APPRAISAL
4-2	EKOFISK	WILDCAT

corp.drillingops		
name	driStDt	reason
NO-2-1	20-03-1989	WILDCAT
NO-3-1	06-07-1968	WILDCAT
NO-3-A	22-07-2011	PRODUCTION
NO-4-2	18-09-1969	

Mapping assertions

1. SELECT name, wbField, opPurpose FROM nat.wellbore
   $\rightsquigarrow$:NatWB/{name} :inField {wbField} ; :purpose {opPurp} .
2. SELECT name, driStDt, reason FROM corp.drillingops
   $\rightsquigarrow$:CorpWB/{name} :drillingStarted {driStDt} ; :purpose {reason} .

Some triples in the ABox defined by the DBs and mapping

:NatWB/2-1	:inField	BLANE .
:NatWB/2-1	:purpose	WILDCAT .
:CorpWB/NO-2-1	:drillingStarted	20-03-1989 .
:CorpWB/NO-2-1	:purpose	WILDCAT .

Integrated querying – Example

nat.wellbore		
name	wbField	opPurpose
2-1	BLANE	WILDCAT
3-1		WILDCAT
3-10	OSELVAR	APPRAISAL
4-2	EKOFISK	WILDCAT

corp.drillingops		
name	driStDt	reason
NO-2-1	20-03-1989	WILDCAT
NO-3-1	06-07-1968	WILDCAT
NO-3-A	22-07-2011	PRODUCTION
NO-4-2	18-09-1969	

Some triples in the ABox defined by the DBs and mapping

```
:NatWB/2-1       :inField          BLANE .
:NatWB/2-1       :purpose          WILDCAT .
:CorpWB/NO-2-1   :drillingStarted  20-03-1989 .
:CorpWB/NO-2-1   :purpose          WILDCAT .
```

Intuitively, 2-1 in nat and NO-2-1 in corp represent the same wellbore.
Hence the SPARQL query

```
SELECT * WHERE { ?w :inField ?f.  ?w :drillingStarted ?d .  }
```

should return an answer, e.g.,

$$\{?w \mapsto \texttt{:NatWB/2-1}, \quad ?f \mapsto \texttt{BLANE}, \quad ?d \mapsto \texttt{20-3-1989}\}.$$

Integrated querying – Example

nat.wellbore		
name	wbField	opPurpose
2-1	BLANE	WILDCAT
3-1		WILDCAT
3-10	OSELVAR	APPRAISAL
4-2	EKOFISK	WILDCAT

corp.drillingops		
name	driStDt	reason
NO-2-1	20-03-1989	WILDCAT
NO-3-1	06-07-1968	WILDCAT
NO-3-A	22-07-2011	PRODUCTION
NO-4-2	18-09-1969	

Some triples in the ABox defined by the DBs and mapping

```
:NatWB/2-1       :inField          BLANE .
:NatWB/2-1       :purpose          WILDCAT .
:CorpWB/NO-2-1   :drillingStarted  20-03-1989 .
:CorpWB/NO-2-1   :purpose          WILDCAT .
```

Intuitively, 2-1 in nat and NO-2-1 in corp represent the same wellbore.
Hence the SPARQL query

SELECT * WHERE { ?w :inField ?f. ?w :drillingStarted ?d . }

should return an answer, e.g.,

$\{?w \mapsto \texttt{:NatWB/2-1},\quad ?f \mapsto \texttt{BLANE},\quad ?d \mapsto \texttt{20-3-1989}\}$.

Integrated querying – Example

nat.wellbore		
name	wbField	opPurpose
2-1	BLANE	WILDCAT
3-1		WILDCAT
3-10	OSELVAR	APPRAISAL
4-2	EKOFISK	WILDCAT

corp.drillingops		
name	driStDt	reason
NO-2-1	20-03-1989	WILDCAT
NO-3-1	06-07-1968	WILDCAT
NO-3-A	22-07-2011	PRODUCTION
NO-4-2	18-09-1969	

Some triples in the ABox defined by the DBs and mapping

```
:NatWB/2-1        :inField           BLANE .
:NatWB/2-1        :purpose           WILDCAT .
:CorpWB/NO-2-1    :drillingStarted   20-03-1989 .
:CorpWB/NO-2-1    :purpose           WILDCAT .
```

Intuitively, `2-1` in nat and `NO-2-1` in corp represent the same wellbore.
Hence the SPARQL query

```
SELECT * WHERE { ?w :inField ?f.  ?w :drillingStarted ?d .  }
```

should return an answer, e.g.,

$\{?w \mapsto \texttt{:NatWB/2-1},\quad ?f \mapsto \texttt{BLANE},\quad ?d \mapsto \texttt{20-3-1989}\}$.

Virtually merge the data using `owl:sameAs` and mappings [Calvanese et al., 2015]

Example owl:sameAs mapping

central.masterTable		
id	natName	corpName
2	2-1	NO-2-1
3	3-1	NO-3-1
4	4-2	NO-4-2
5		NO-3-A
6	3-10	

- `SELECT natName, corpName FROM central.masterTable` $\rightsquigarrow$ `:NatWB/{natName} owl:sameAs :CorpWB/{corpName} .`
- Mapping for owl:sameAs can rely on master tables, but may use arbitrary SQL queries to ordinary tables.

- `sameAs` is the standard way of dealing with identity resolution in OWL.
- We assume that there is no sameAs relation within a datasource $\rightsquigarrow$ the length of sameAs chain is bounded by the number of datasources
- Semantics of `sameAs` may cause an exponential number of query results:
  - detrimental for performance
  - redundancy makes query answers difficult to understand

$\rightsquigarrow$ *Not feasible or desirable in practice!*

Virtually merge the data using `owl:sameAs` and mappings [Calvanese et al., 2015]

Example owl:sameAs mapping

central.masterTable		
id	natName	corpName
2	2-1	NO-2-1
3	3-1	NO-3-1
4	4-2	NO-4-2
5		NO-3-A
6	3-10	

- `SELECT natName, corpName`
  `FROM central.masterTable`
  $\leadsto$ `:NatWB/{natName} owl:sameAs`
  `:CorpWB/{corpName} .`
- Mapping for owl:sameAs can rely on master tables, but may use arbitrary SQL queries to ordinary tables.

- `sameAs` is the standard way of dealing with identity resolution in OWL.
- We assume that there is no sameAs relation within a datasource $\leadsto$ the length of sameAs chain is bounded by the number of datasources
- Semantics of `sameAs` may cause an exponential number of query results:
  - detrimental for performance
  - redundancy makes query answers difficult to understand

$\leadsto$ *Not feasible or desirable in practice!*

- Idea: for the entities with different IRI, assign a canonical one
- Now the mapping $\mathcal{M}$ includes **assertions $\mathcal{M}^c$ populating** `canIriOf`.
- Such mappings should satisfy some properties:
  e.g., each IRI has at most one canonical IRI

Example master table and mapping

central.masterTable		
id	natName	corpName
2	2-1	NO-2-1
3	3-1	NO-3-1
4	4-2	NO-4-2
5		NO-3-A
6	3-10	

- SELECT id, natName
  FROM central.masterTable
  $\leadsto$ `:WB/{id} canIriOf :NatWB/{natName} .`
- SELECT id, corpName
  FROM central.masterTable
  $\leadsto$ `:WB/{id} canIriOf :CorpWB/{corpName} .`

- Idea: for the entities with different IRI, assign a canonical one
- Now the mapping $\mathcal{M}$ includes **assertions $\mathcal{M}^c$ populating** `canIriOf`.
- Such mappings should satisfy some properties:
  e.g., each IRI has at most one canonical IRI

Example master table and mapping

central.masterTable		
id	natName	corpName
2	2-1	NO-2-1
3	3-1	NO-3-1
4	4-2	NO-4-2
5		NO-3-A
6	3-10	

- SELECT id, natName
  FROM central.masterTable
  $\rightsquigarrow$ `:WB/{id} canIriOf :NatWB/{natName} .`
- SELECT id, corpName
  FROM central.masterTable
  $\rightsquigarrow$ `:WB/{id} canIriOf :CorpWB/{corpName} .`

- We propose a practical method based on compiling the consequences of canonical IRI semantics into mappings $\rightsquigarrow$ **Mapping rewriting**
- We replace individuals and IRI-templates in the mapping by their canonical representation.
- Again, this is inspired by the mapping saturation algorithm.

Mapping rewriting – Example

Mapping $\mathcal{M}^t$

1. `SELECT name, wbField, opPurpose FROM nat.wellbore`
   $\leadsto$ `:NatWB/{name} :inField {wbField} ; :purpose {opPurp} .`
2. `SELECT name, driStDt, reason FROM corp.drillingops`
   $\leadsto$ `:CorpWB/{name} :drillingStarted {driStDt} ; :purpose {reason} .`

Mapping $\mathcal{M}^c$

1. `SELECT id, natName FROM central.masterTable`
   $\leadsto$ `:WB/{id} canIriOf :NatWB/{natName} .`
2. `SELECT id, corpName FROM central.masterTable`
   $\leadsto$ `:WB/{id} canIriOf :CorpWB/{corpName} .`

Canonical-iri rewriting

1. `SELECT wlbFld, opPurp, id`
   `FROM nat.wellbore, central.masterTable WHERE name = natName`
   $\leadsto$ `:WB/{id} :inField {wlbField} ; :purpose {opPurp} .`
2. `SELECT driStDt, reason, id`
   `FROM corp.drillingops, central.masterTable WHERE name = corpName`
   $\leadsto$ `:WB/{id} :drillingStarted {driStDt} ; :purpose {reason} .`

Calvanese, D., M. Giese, D. Hovland, and M. Rezk (2015). "Ontology-based Integration of Cross-linked Datasets". In: *Proc. of ISWC*. Vol. 9366. LNCS. Springer, pp. 199–216.

Xiao, G., D. Hovland, D. Bilidas, M. Rezk, M. Giese, and D. Calvanese (2018). "Efficient Ontology-Based Data Integration with Canonical IRIs". In: *Proc. of ESWC*. Springer.