Ontology-Based Data Access and OWL 2 QL

Roman Kontchakov

Department of Computer Science and Information Systems,
Birkbeck, University of London

http://www.dcs.bbk.ac.uk/~roman

Data access in industry

(from Norwegian Petroleum Directorate's FactPages)

show me the wellbores completed before 2008 where Statoil as a drilling operator sampled less than 10 meters of cores

5 days later:

```
SELECT DISTINCT cores.wlbName, cores.lenghtM, wellbore.wlbDrillingOperator, wellbore.wlbCompletionYear
FROM
    ( (SELECT wlbName, wlbNpdidWellbore, (wlbTotalCoreLength * 0.3048) AS lenghtM
       FROM wellbore_core
       WHERE wlbCoreIntervalUom = '[ft ]' )
      UNION
      (SELECT wlbName, wlbNpdidWellbore, wlbTotalCoreLength AS lenghtM
       FROM wellbore_core
       WHERE wlbCoreIntervalUom = '[m ]' )
    ) as cores,
    ( (SELECT wlbNpdidWellbore, wlbDrillingOperator, wlbCompletionYear
       FROM wellbore_development_all
      UNION
      (SELECT wlbNpdidWellbore, wlbDrillingOperator, wlbCompletionYear
       FROM wellbore_exploration_all )
      UNION
      (SELECT wlbNpdidWellbore, wlbDrillingOperator, wlbCompletionYear
       FROM wellbore_shallow_all )
    ) as wellbore
WHERE wellbore.wlbNpdidWellbore = cores.wlbNpdidWellbore
        ...
```

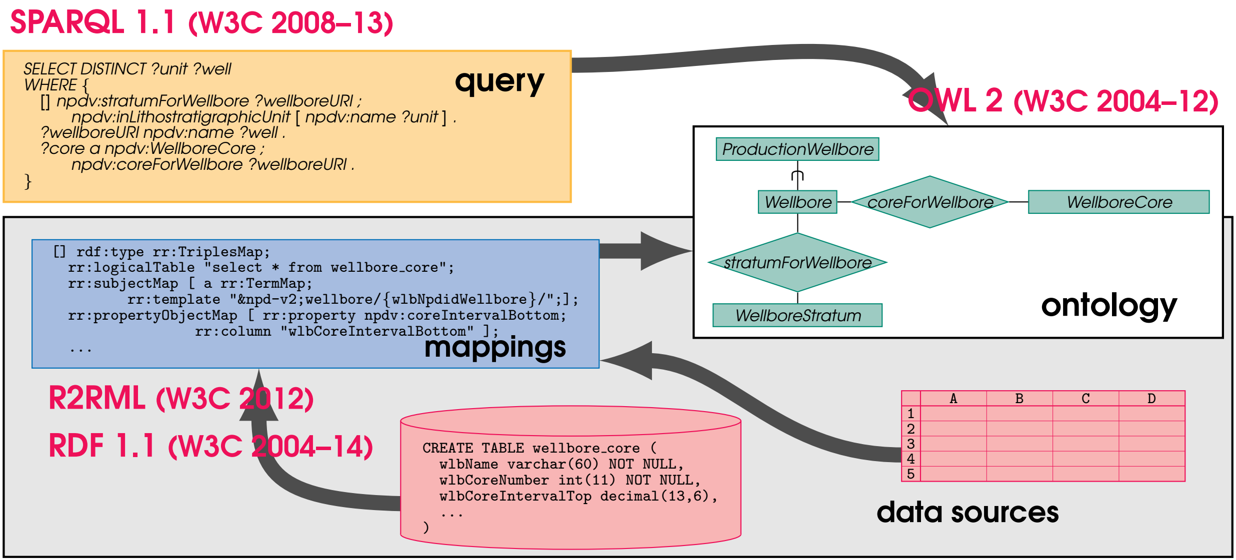
In STATOIL:

1,000 TB of relational data

1,500 tables

30–70% of time on data gathering

Ontology-Based Data Access (OBDA) Poggi et al. (JDS 2008)



ontology

- gives a high-level conceptual view of the data
- provides a convenient & natural vocabulary for user queries
- enriches incomplete data with background knowledge

Materialisation or ETL (Extract, Transform and Load)

translate mappings into rules:

$$\texttt{wellbore\_core}(t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}) \to$$
$$\texttt{npdv:coreIntervalBottom}(\texttt{URI}(\texttt{"\&npdv;wellbore/\{\}/core/\{\}"}, t_9, t_2), t_4)$$

translate the ontology onto rules:

$\texttt{npdv:production\_wellbore}(x) \to \texttt{npdv:wellbore}(x)$ `rdfs:subClassOf`

$\texttt{npdv:coreForWellbore}(x, y) \to \texttt{npdv:WellboreCore}(y)$ `rdfs:range`

`owl:someValuesFrom` (on the left-hand side of $\to$)

not every OWL 2 axiom can be translated into rules

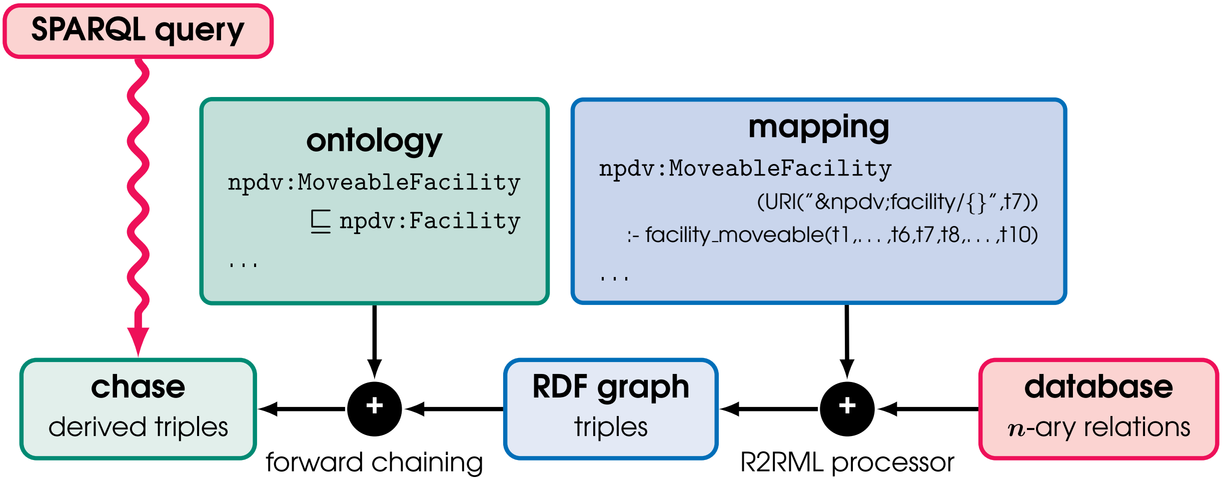
$\texttt{npdv:WellboreCore}(y) \to \exists x\, \texttt{npdv:coreForWellbore}(x, y)$ `owl:someValuesFrom`

(on the right-hand side of $\to$)

$$\texttt{npdv:StratigraphicUnit}(x) \to$$
$$\texttt{npdv:LithostratigraphicUnit}(x) \lor \texttt{npdv:ChronostratigraphicUnit}(x)$$

`owl:unionOf`

Forward Chaining and OWL 2 RL



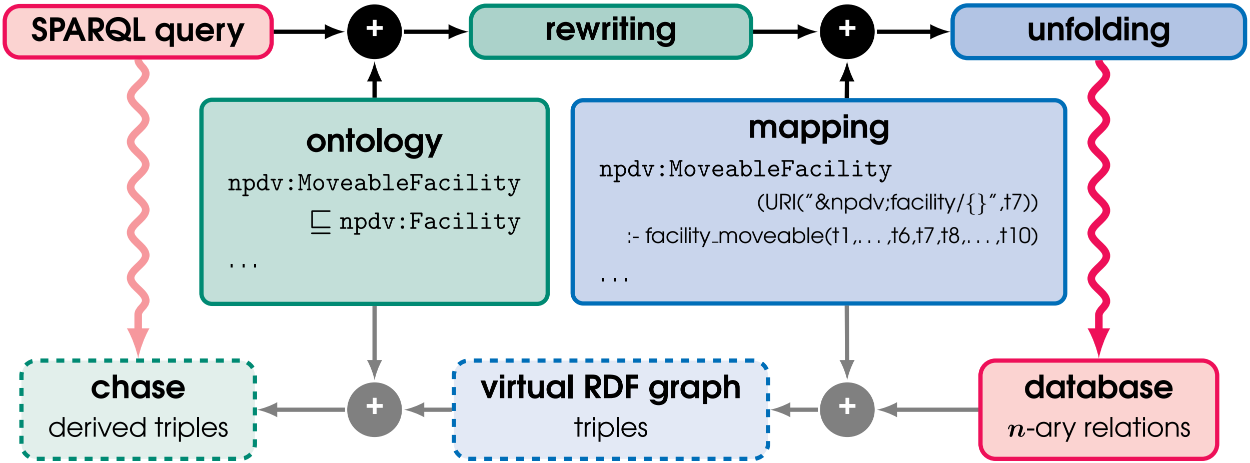
chase is defined only for Horn logics (**no disjunction**)

in general, even for Horn ontologies in OWL 2, the chase does not terminate
(**value invention** as in $\texttt{npdv:WellboreCore}(y) \to \exists x\, \texttt{npdv:coreForWellbore}(x, y)$)

OWL 2 RL is the largest Horn fragment of OWL 2 without value invention

Grosof et al. (WWW 2003), **ter Horst (Web Semantics, 2005)**

Backward Chaining and OWL 2 QL



OWL 2 QL is almost the largest fragment of OWL 2 that supports backward chaining

OWL 2 QL can encode UML class / ER diagrams **Artale et el. (ER 2007)**

data complexity of query answering in OWL 2 QL = the data complexity of database query evaluation (AC^0)

Forward v Backward Chaining

ontology: $\texttt{production\_wellbore}(x) \to \texttt{wellbore}(x)$

data: $\texttt{production\_wellbore}(\boldsymbol{a42}),\ \texttt{wellbore}(\boldsymbol{a92})$

query: $\boldsymbol{q}(\boldsymbol{x}) \leftarrow \texttt{wellbore}(\boldsymbol{x})$

forward chaining

1. 'apply' ontology to the data to obtain the chase:

$$\texttt{production\_wellbore}(\boldsymbol{a42}),\ \boxed{\texttt{wellbore}(\boldsymbol{a42})},\ \texttt{wellbore}(\boldsymbol{a92})$$

2. query the chase

backward chaining

1. 'apply' ontology to the query to obtain its rewriting:

$$\boldsymbol{q}(\boldsymbol{x}) \leftarrow \texttt{wellbore}(\boldsymbol{x})$$

$$\boldsymbol{q}(\boldsymbol{x}) \leftarrow \boxed{\texttt{production\_wellbore}(\boldsymbol{x})}$$

the head of the rule unifies with a query atom

$\Longrightarrow$ create a copy of the CQ with the atom replaced by the rule body

2. use the obtained UCQ to query the original data

Query Rewriting: Theory and Practice

UCQ rewritings are exponential $\Longrightarrow$ **very bad** in practice

in general, even PE- and NDL-rewritings are **exponential**

and FO-rewritings are superpolynomial unless NP/poly $\subseteq$ NC^1

for more details, see **Bienvenu et al. (2016)** `https://arxiv.org/abs/1605.01207`

using $\vee$ in UCQs (unions of semiconjucntive queries) helps

to deal with **class/property hierarchies** in practice

moreover, hierarchies can be compiled into mappings (**T-mappings**)

and optimised using database **integrity constraints**

implemented in ontop framework

Free University of Bozen-Bolzano
with some help from Birkbeck

Calvanese et al. (Semantic Web, 2017), **Rodriguez-Muro et al. (ISWC 2013)**