Ontology-based Data Access and Integration – Relational Data and Beyond

Diego Calvanese

KRDB Research Centre for Knowledge and Data
Free University of Bozen-Bolzano, Italy

Ontopic s.r.l.

Department of Computing Science
Umeå University, Sweden

SAP Inspiration Sessions

15 December 2021 – Online

Data integration

Databases are great!
They let us manage efficiently huge amounts of data . . .
. . . assuming you have put all data into your schema.

However, the reality is much more complicated and **heterogeneous**:

- Data sets were created independently.
- Data are often stored across different sources.
- Data sources are controlled by different people / organizations.

Goal of data integration

To put together different data sources,
created for different purposes,
and controlled by different people,
making them accessible in a uniform way.

unibz

Why heterogeneity?

- **Data model heterogeneity**: Relational data, graph data, xml, json, csv, text files, . . .
- **System heterogeneity**: Even when systems adopt the same data model, they are not always fully compatible.
- **Schema heterogeneity**: Different people see things differently, and design schemas differently!
- **Data-level heterogeneity**: e.g., ‘IBM’ vs. ‘Int. Business Machines’ vs. ‘International Business Machines’.

How to address heterogeneity?

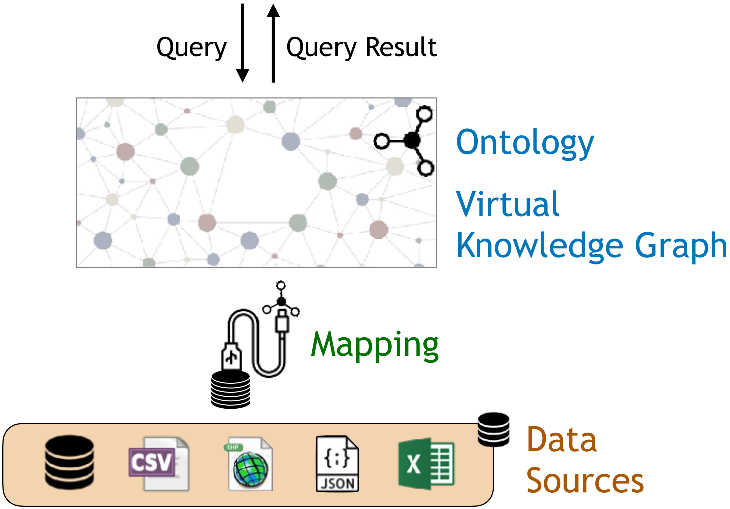
We combine three key ideas:

1. Use a global (or integrated) schema and **map the data sources to the global schema**.
2. Adopt a very flexible data model for the global schema
   $\leadsto$ **Knowledge Graph** whose vocabulary is expressed in an **ontology**.
3. Exploit **virtualization**, i.e., the KG is not materialized, but kept virtual.

This gives rise to the **Virtual Knowledge Graph** (**VKG**) approach to data access / integration, also called **Ontology-based Data Access / Integration** (**OBDA**).
[Xiao, C., et al. 2018, IJCAI]

unibz

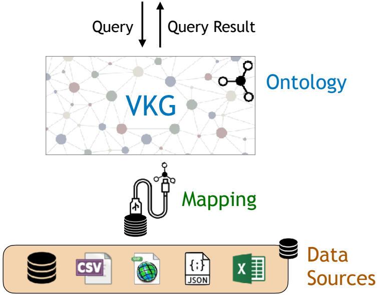
Virtual Knowledge Graph (VKG) architecture



unibz

Why an ontology?

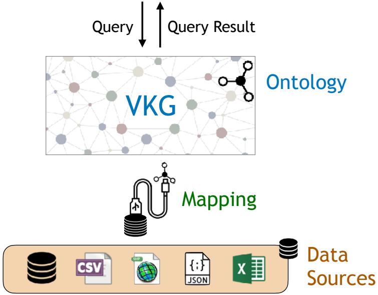
An ontology is a structured formal representation of concepts and their relationships that are relevant for the domain of interest.



- In the VKG setting, the ontology has a twofold purpose:
  - It defines a vocabulary of terms to denote classes and properties that are familiar to the user.
  - It extends the data in the sources with background knowledge about the domain of interest, and this knowledge is machine processable.
- One can make use of custom-built domain ontologies.
- In addition, one can rely on standard ontologies, which are available for many domains.

unibz

Why a Knowledge Graph for the global schema?



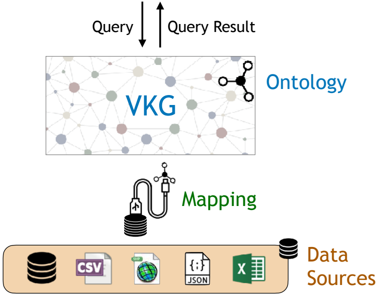
The traditional approach to data integration adopts a relational global schema.

A Knowledge Graph, instead:

- Does not require to commit early on to a specific structure.
- Can better accommodate heterogeneity.
- Can better deal with missing / incomplete information.
- Does not require complex restructuring operations to accommodate new information or new data sources.

unibz

Why mappings?

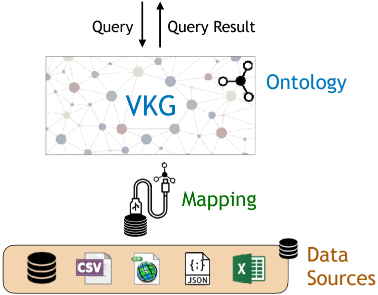


The traditional approach to data integration relies on mediators, which are specified through complex code.

Mappings, instead:

- Provide a declarative specification, and not code.
- Are easier to understand, and hence to design and to maintain.
- Support an incremental approach to integration.
- Are machine processable, hence are used in query answering and for query optimization.

Why virtualization?



Materialized data integration relies on extract-transform-load (ETL) operations, to load data from the sources into an integrated data store / data warehouse / materialized KG.

In the virtual approach, instead:

- The data stays in the sources and is only accessed at query time.
- No need to construct a large and potentially costly materialized data store and keep it up-to-date.
- Hence the data is always fresh wrt the latest updates at the sources.
- One can rely on the existing data infrastructure and expertise.
- There is better support for an incremental approach to integration.

unibz

Outline

unibz

Outline

Applications of the VKG approach

Adopted in many academic and industrial use cases from different application areas.
See also [Xiao, Ding, et al. 2019, Data Intelligence].

- Industry 4.0
- Analytical processing / Business Intelligence
- Geospatial data

unibz

Applications of the VKG approach

Adopted in many academic and industrial use cases from different application areas.
See also [Xiao, Ding, et al. 2019, Data Intelligence].

- **Industry 4.0**
  - Ability to deal with data coming from different vendors, or with historical heterogeneous data.

  Examples: Equinor, Siemens, Bosch

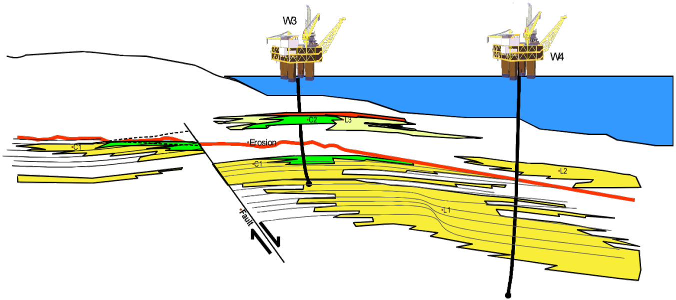
- Analytical processing / Business Intelligence

- Geospatial data

unibz

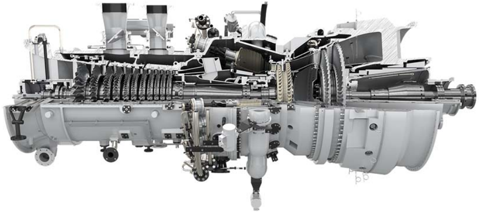
Surface exploration at Statoil [Kharlamov, Hovland, et al. 2017, J. of Web Semantics]

- Statoil (now Equinor) is Norway's largest (oil and gas) company. Statoil has been a use case partner in the EU project Optique.
- Exploration domain: analyze existing relevant data in order to find exploitable accumulations of oil or gas.
- Improve the efficiency of the information gathering routine for geologists.
- Efficient, creative data collection from multiple large volume data sources.



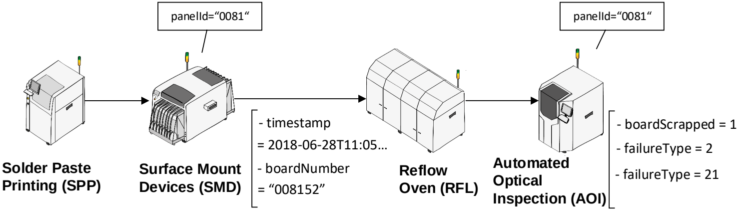
unibz

Siemens Energy Services [Kharlamov, Mailis, et al. 2017, J. of Web Semantics]



- Use case partner in the EU project Optique.
- Siemens produces huge appliances (e.g., gas turbines) and installs them in plants.
- Siemens service centers:
  - over 50 service centers world-wide
  - each center is responsible for several thousands of appliances
  - offer constant monitoring and diagnostics services
- Monitoring and diagnostics tasks
  - reactive and preventive diagnostics: offline, after an issue is detected
  - predictive analyses: real-time, to avoid issues while appliance is functioning

Failure detection for surface mounting at Bosch [Kalayci et al. 2020, ISWC]



- The Surface Mounting Process at Bosch consists of four separate phases involving different machines.
- The involved machines come from different suppliers and rely on distinct formats.
- Failure detection fundamentally relies on the integration and analysis of data generated in the different phases.

Applications of the VKG approach

Adopted in many academic and industrial use cases from different application areas.
See also [Xiao, Ding, et al. 2019, Data Intelligence].

- Industry 4.0
  - Ability to deal with data coming from different vendors, or with historical heterogeneous data.

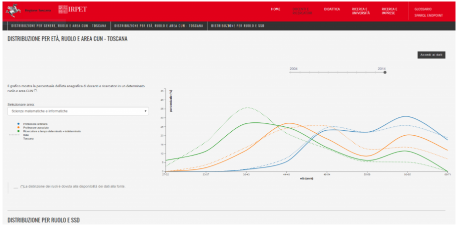
  Examples: Equinor, Siemens, Bosch

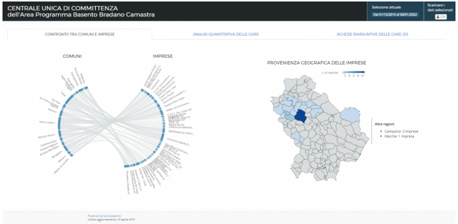
- **Analytical processing / Business Intelligence**
  - Combine internal data, manual processes (e.g., Excel), and external data.
  - Data privacy issues / GDPR: we need to avoid data copies

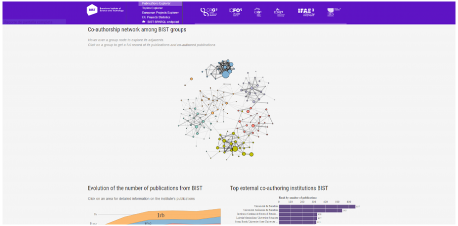
  Examples: Toscana Open Research, a large European university, a large TLC company

- Geospatial data

Toscana Open Research







http://www.toscanaopenresearch.it/en/

unibz

A large European university

- Internal data
  - Research funding, HR, teaching, etc.
  - Redundant applications due to the merge of several universities.
  - Operational data store and data warehouse.
  - Many processes are still using Excel.
- External data
  - Open Data (from the ministry, EU commission and public initiatives).
  - Commercial bibliometric data.
  - Mainly for benchmarking.

unibz

VKG over scientific documentation for a large TLC company

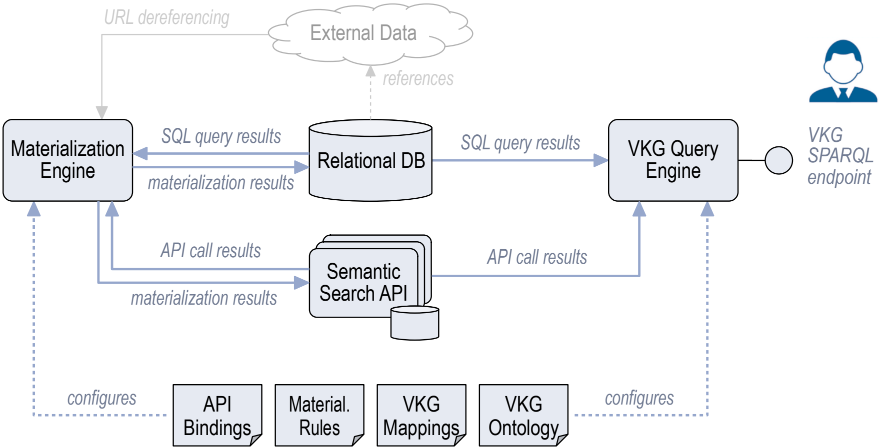
Goal: build a **virtual knowledge graph** integrating structured data in a proprietary platform and the results of information extraction from related semi-structured data.

- Structured data is provided by a relational database.
- **Semi-structured data** consisting of text with little (if any) structure or markup, such as natural language text, HTML documents, PDF files.
- **Information extraction (IE)** aims at extracting structured information from semi-structured data, possibly leveraging natural language processing (NLP) techniques.

Motivations:

- Provide an unambiguous formalization of the knowledge in the platform, to ease exploitation.
- Provide an integrated, queryable, up-to-date view over all available information.
- Enable more advanced services, such as intelligent search and intelligent recommendation.

High-Level architecture of VKG over semi-structured data



unibz

Applications of the VKG approach

Adopted in many academic and industrial use cases from different application areas.
See also [Xiao, Ding, et al. 2019, Data Intelligence].

- Industry 4.0
  - Ability to deal with data coming from different vendors, or with historical heterogeneous data.

  Examples: Equinor, Siemens, Bosch

- Analytical processing / Business Intelligence
  - Combine internal data, manual processes (e.g., Excel), and external data.
  - Data privacy issues / GDPR: we need to avoid data copies

  Examples: Toscana Open Research, a large European university, a large TLC company

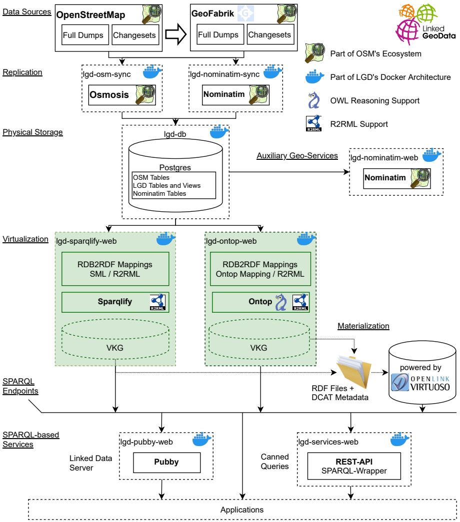
- **Geospatial data**
  - GeoSPARQL over PostGIS

  Examples: LinkedGeoData.org, South Tyrolean Open Data Hub

unibz

LinkedGeoData.org

- LinkedGeoData.org (LGD) converts OpenStreetMap to RDF.
- Is one of the most important Geospatial Knowledge Graphs.
- Ongoing project in collaboration with University of Leipzig to develop a new version of LGD based on the Ontop VKG system. [Ding et al. 2022, J. of Web Semantics]



unibz

LinkedGeoData.org

LinkedGeoData.org

endpoint address: http://localhost:8080/sparql | ontop v4.1.0-beta-1-SNAPSHOT

Playground | Example Queries

Query x | Query 1 x | Query 2 x | Query 3 x | Query 4 x | Query 6 x | Query 7 x | road segment x | isHostedBy x | Query 11 x | Query 10 x | Query 12 x | Query 13 x | Query 5 x | Query 8 x | Query 9 x | Query 14 x | +

```
SELECT ?x ?wkt ?wktLabel ?wktColor WHERE {
  {  ?x a lgdo:University ; geo:asWKT ?wkt . OPTIONAL {?x rdfs:label ?wktLabel . FILTER (LANG(?wktLabel) = '')}
    BIND('red' AS ?wktColor)
  }
  UNION {
    ?u a lgdo:University ; geo:asWKT ?uWkt . OPTIONAL {?u rdfs:label ?uktLabel . FILTER (LANG(?uktLabel) = '')}
    ?x a lgdo:Restaurant ; geo:asWKT ?wkt ; rdfs:label ?wktLabel . FILTER (LANG(?wktLabel) = '')
    FILTER(geof:distance(?wkt, ?uWkt, uom:metre) < 200)
    BIND('blue' AS ?wktColor)
  }
  UNION {
    ?u a lgdo:University ; geo:asWKT ?uWkt . OPTIONAL {?u rdfs:label ?uktLabel . FILTER (LANG(?uktLabel) = '')}
    BIND(geof:buffer(?uWkt, 200, uom:metre) AS ?wkt)  BIND('red' AS ?wktColor)
}
```

Table | Response | Pivot Table | Google Chart | Geo

VKG over the South Tyrolean Open Data Hub (ODH)

- The *South Tyrolean Open Data Hub* (ODH) publishes tourism, mobility, and weather data from different providers through a JSON-based Web API.
- ODH is developed by NOI, a South Tyrolean company managing a Techpark in Bolzano and providing services to companies and research institutions.
- The backend of ODH relies on a PostgreSQL database.
- Ongoing project between Ontopic and NOI on extending ODH with a Virtual Knowledge Graph.

https://sparql.opendatahub.bz.it/

unibz

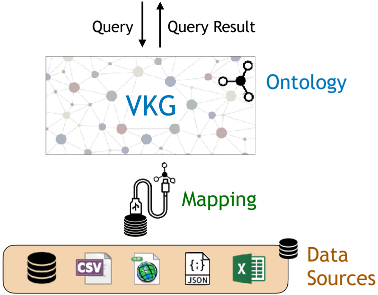
Outline

unibz

Components of the VKG framework

We consider now the main components that make up the VKG framework, and the languages used to specify them.

In defining such languages, we need to consider the **tradeoff between expressive power and efficiency**, where the key point is efficiency with respect to the data.

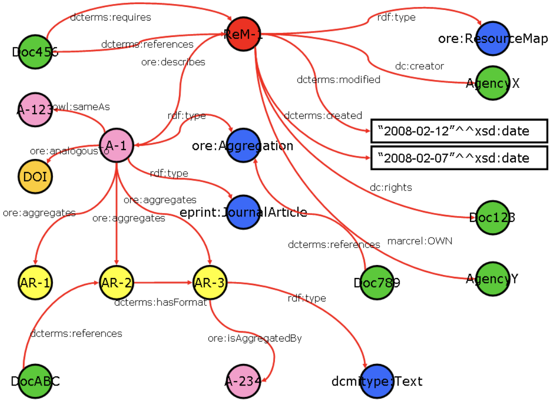


The W3C has standardized languages that are suitable for VKGs:

1. Knowledge graph: expressed in **RDF** [W3C Rec. 2014] (v1.1)
2. Ontology $\mathcal{O}$: expressed in **OWL 2 QL** [W3C Rec. 2012]
3. Mapping $\mathcal{M}$: expressed in **R2RML** [W3C Rec. 2012]
4. Query: expressed in **SPARQL** [W3C Rec. 2013] (v1.1)

RDF – Data is represented as a graph

The graph consists of a set of **subject-predicate-object triples**:



Class membership:

```
<WB-2025> rdf:type :Wellbore .
```

Object property:

```
<WB-2025> :hasMeasurement <M-48> .
```

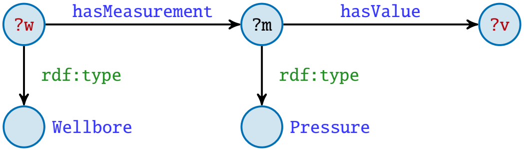
Data property:

```
<M-48> :hasDate "2008-02-07" .
```

SPARQL query language

- Is the standard query language for RDF data. [W3C Rec. 2008, 2013]
- Core query mechanism is based on **graph matching**.

```
SELECT ?w ?v
WHERE { ?w rdf:type Wellbore .
        ?w hasMeasurement ?m .
        ?m rdf:type Pressure .
        ?m hasValue ?v .
      }
```



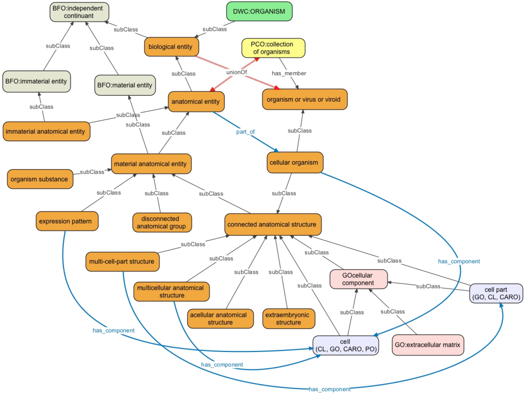
Additional language features (SPARQL 1.1):

- `UNION`: matches one of alternative graph patterns
- `OPTIONAL`: produces a match even when part of the pattern is missing
- complex `FILTER` conditions
- `GROUP BY`, to express aggregations
- `MINUS`, to remove possible solutions
- property paths (regular expressions)

unibz

What is an ontology?

- An ontology conceptualizes a domain of interest in terms of **concepts/classes**, (binary) **relations**, and their **properties**.
- It typically organizes the concepts in a hierarchical structure.
- Ontologies are often represented as graphs.



What is an ontology?

- An ontology conceptualizes a domain of interest in terms of **concepts/classes**, (binary) **relations**, and their **properties**.
- It typically organizes the concepts in a hierarchical structure.
- Ontologies are often represented as graphs.
- However, an ontology is actually a **logical theory**, expressed in a suitable fragment of first-order logic

$\forall x.\, \mathsf{Pressure}(x) \rightarrow \mathsf{Measurement}(x)$
$\forall x.\, \mathsf{Porosity}(x) \rightarrow \mathsf{Measurement}(x)$
$\forall x.\, \mathsf{Permeability}(x) \rightarrow \mathsf{Measurement}(x)$
$\forall x.\, \mathsf{Temperature}(x) \rightarrow \mathsf{Measurement}(x)$
$\forall x.\, \mathsf{Pressure}(x) \rightarrow \neg\mathsf{Porosity}(x) \wedge \neg\mathsf{Permeability}(x) \wedge \neg\mathsf{Temperature}(x)$
$\forall x.\, \mathsf{Porosity}(x) \rightarrow \neg\mathsf{Permeability}(x) \wedge \neg\mathsf{Temperature}(x)$
$\forall x.\, \mathsf{Permeability}(x) \rightarrow \neg\mathsf{Temperature}(x)$

$\forall x.\, \mathsf{HydrostaticPressure}(x) \rightarrow \mathsf{Pressure}(x)$
$\forall x.\, \mathsf{FormationPressure}(x) \rightarrow \mathsf{Pressure}(x)$
$\forall x.\, \mathsf{PorePressure}(x) \rightarrow \mathsf{Pressure}(x)$
$\forall x.\, \mathsf{HydrostaticPressure}(x) \rightarrow \neg\mathsf{FormationPressure}(x) \wedge \neg\mathsf{PorePressure}(x)$
$\forall x.\, \mathsf{FormationPressure}(x) \rightarrow \neg\mathsf{PorePressure}(x)$

$\forall x, y.\, \mathsf{hasFormationPressure}(x, y) \rightarrow \mathsf{Wellbore}(x) \wedge \mathsf{FormationPressure}(y)$
$\forall x, y.\, \mathsf{hasDepth}(x, y) \rightarrow \mathsf{FormationPressure}(x) \wedge \mathsf{Depth}(y)$
$\forall x.\, \mathsf{FormationPressure}(x) \rightarrow \exists y.\, \mathsf{hasDepth}(x, y)$

$\forall x, y.\, \mathsf{hasFormationPressure}(x, y) \rightarrow \mathsf{hasMeasurement}(x, y)$

$\forall x, y.\, \mathsf{completionDate}(x, y) \rightarrow \mathsf{Wellbore}(x) \wedge \mathsf{xsd{:}dateTime}(y)$
$\forall x.\, \mathsf{Wellbore}(x) \rightarrow (\sharp\{y \mid \mathsf{completionDate}_{\mathsf{wb}}(x, y)\} \leq 1)$
$\forall x, y.\, \mathsf{wellboreTrack}_{\mathsf{wb}}(x, y) \rightarrow \mathsf{Wellbore}(x) \wedge \mathsf{xsd{:}string}(y)$
$\forall x.\, \mathsf{Wellbore}(x) \rightarrow (\sharp\{y \mid \mathsf{wellboreTrack}_{\mathsf{wb}}(x, y)\} \leq 1)$

$\forall x, y.\, \mathsf{hasCoreSample}(x, y) \rightarrow \mathsf{Core}(x) \wedge \mathsf{CoreSample}(y)$
$\forall x.\, \mathsf{CoreSample}(x) \rightarrow \exists y.\, \mathsf{hasCoreSample}(y, x) \wedge \mathsf{Core}(y)$
$\ldots$

What is an ontology?

- An ontology conceptualizes a domain of interest in terms of **concepts/classes**, (binary) **relations**, and their **properties**.
- It typically organizes the concepts in a hierarchical structure.
- Ontologies are often represented as graphs.
- However, an ontology is actually a **logical theory**, expressed in a suitable fragment of first-order logic, or better, in **description logics**.

$$
\begin{array}{rcl}
\mathsf{Pressure} & \sqsubseteq & \mathsf{Measurement}\\
\mathsf{Porosity} & \sqsubseteq & \mathsf{Measurement}\\
\mathsf{Permeability} & \sqsubseteq & \mathsf{Measurement}\\
\mathsf{Temperature} & \sqsubseteq & \mathsf{Measurement}\\
\mathsf{Pressure} & \sqsubseteq & \neg\mathsf{Porosity} \sqcap \neg\mathsf{Permeability} \sqcap \neg\mathsf{Temperature}\\
\mathsf{Porosity} & \sqsubseteq & \neg\mathsf{Permeability} \sqcap \neg\mathsf{Temperature}\\
\mathsf{Permeability} & \sqsubseteq & \neg\mathsf{Temperature}\\
\mathsf{HydrostaticPressure} & \sqsubseteq & \mathsf{Pressure}\\
\mathsf{FormationPressure} & \sqsubseteq & \mathsf{Pressure}\\
\mathsf{PorePressure} & \sqsubseteq & \mathsf{Pressure}\\
\mathsf{HydrostaticPressure} & \sqsubseteq & \neg\mathsf{FormationPressure} \sqcap \neg\mathsf{PorePressure}\\
\mathsf{FormationPressure} & \sqsubseteq & \neg\mathsf{PorePressure}\\
\exists\mathsf{hasFormationPressure} & \sqsubseteq & \mathsf{Wellbore}\\
\exists\mathsf{hasFormationPressure}^- & \sqsubseteq & \mathsf{FormationPressure}\\
\exists\mathsf{hasDepth} & \sqsubseteq & \mathsf{FormationPressure}\\
\exists\mathsf{hasDepth}^- & \sqsubseteq & \mathsf{Depth}\\
\mathsf{FormationPressure} & \sqsubseteq & \exists\mathsf{hasDepth}\\
\mathsf{hasFormationPressure} & \sqsubseteq & \mathsf{hasMeasurement}\\
\exists\mathsf{completionDate}_{\mathsf{wb}} & \sqsubseteq & \mathsf{Wellbore}\\
\exists\mathsf{completionDate}_{\mathsf{wb}}^- & \sqsubseteq & \mathsf{xsd{:}dateTime}\\
\mathsf{Wellbore} & \sqsubseteq & (\leq 1\, \mathsf{completionDate}_{\mathsf{wb}})\\
\exists\mathsf{wellboreTrack}_{\mathsf{wb}} & \sqsubseteq & \mathsf{Wellbore}\\
 & \cdots &
\end{array}
$$

The OWL 2 QL ontology language

- **OWL 2 QL** is one of the three standard sub-languages of the very expressive standard ontology language OWL 2. [W3C Rec. 2012]
- It is considered a lightweight ontology language:
  - controlled expressive power
  - efficient inference
- Optimized for accessing large amounts of data
  - Queries over the ontology can be rewritten into SQL queries over the underlying relational database (First-order rewritability).
  - Logical consistency of ontology and data can also be checked by executing SQL queries over the underlying database.

Constructs of OWL 2 QL

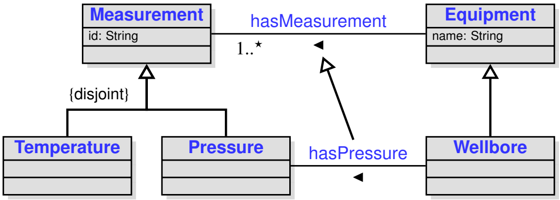
In an OWL 2 QL ontology, one can express knowledge about the classes and properties in the domain of interest by means of various types of assertions.

- Subclass assertions — Pressure `rdfs:subClassOf` Measurement
- Class disjointness — Pressure `owl:disjointWith` Temperature
- Domain of a property — hasPressure `rdfs:domain` Wellbore
- Range of a property — hasPressure `rdfs:range` Pressure
- Subproperty assertions — hasPressure `rdfs:subPropertyOf` hasMeasurement
- Inverse properties — hasMeasurement `owl:inverseOf` isMeasurementOf
- Mandatory participation to a property expression — ... `owl:someValuesFrom` ... in superclass

Representing OWL 2 QL ontologies as UML class diagrams

There is a close correspondence between OWL 2 QL and conceptual modeling formalisms, such as UML class diagrams and ER schemas.

Pressure `rdfs:subClassOf` Measurement	subclass
Pressure `owl:disjointWith` Temperature	disjointness
hasPressure `rdfs:domain` Wellbore	domain
hasPressure `rdfs:range` Pressure	range
hasPressure `rdfs:subPropertyOf` hasMeasurement	sub-association
... `owl:someValuesFrom` ...	mandatory participation



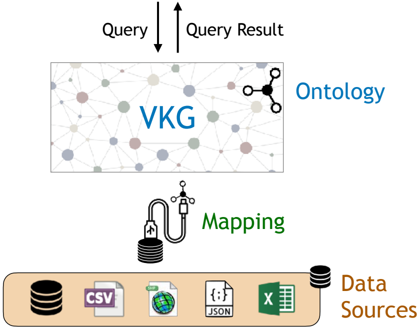
In fact, to visualize an OWL 2 QL ontology, we can use standard UML class diagrams.

Use of mappings

In the VKG framework, the mapping encodes how the data in the sources should be used to create the Virtual Knowledge Graph, which is formulated in the vocabulary of the ontology.

VKG defined from the mapping and the data.

- Queries are answered with respect to the ontology and the data of the VKG.
- The data of the VKG is not materialized (it is virtual!).
- Instead, the information in the ontology and the mapping is used to translate queries over the ontology into queries formulated over the sources.



Note: The graph is **always up to date** wrt the data sources.

unibz

Mapping language

The mapping consists of a set of assertions of the form

SQL Query $\leadsto$ Class membership assertion

SQL Query $\leadsto$ Property membership assertion

Intuition behind the mapping

The answers returned by the SQL Query in the left-hand side are used to create the objects (and values) that populate the Class / Property in the right-hand side.

Note: The mapping contains also a mechanism to transform values retrieved from the database into objects of the VKG (thus solving the so-called impedance mismatch).

unibz

Mapping language – Example

Ontology $\mathcal{O}$:

```
:actsIn rdfs:domain :MovieActor .
:actsIn rdfs:range :Movie .
:title rdfs:domain :Movie .
:title rdfs:range xsd:string .
```

Mapping $\mathcal{M}$:

m_1:
```
SELECT mcode, mtitle FROM MOVIE
WHERE type = "m"
  ⇝ :m/{mcode} rdf:type :Movie .
    :m/{mcode} :title {mtitle} .
```

m_2:
```
SELECT M.mcode, A.acode FROM MOVIE M, ACTOR A
WHERE M.mcode = A.pcode AND M.type = "m"
  ⇝ :a/{acode} :actsIn :m/{mcode} .
```

Database $\mathcal{D}$:

MOVIE

mcode	*mtitle*	*myear*	*type*	···
5118	The Matrix	1999	m	···
8234	Altered Carbon	2018	s	···
2281	Blade Runner	1982	m	···

ACTOR

pcode	*acode*	*aname*	···
5118	438	K. Reeves	···
5118	572	C.A. Moss	···
2281	271	H. Ford	···

The mapping $\mathcal{M}$ applied to database $\mathcal{D}$ generates the (virtual) knowledge graph $\mathcal{V} = \mathcal{M}(\mathcal{D})$:

```
:m/5118 rdf:type :Movie .   :m/5118 :title "The Matrix" .
:m/2281 rdf:type :Movie .   :m/2281 :title "Blade Runner" .
:a/438 :actsIn :m/5118 .   :a/572 :actsIn :m/5118 .   :a/271 :actsIn :m/2281 .
```

unibz

Query answering in VKGs

In VKGs, we want to answer queries formulated over the ontology, by using the data provided by the data sources through the mapping.

- The ontology contains **domain knowledge** that can be used to enrich answers.

  Example: Suppose that our data contains WB-2025 among the Wellbores, and that the ontology states that each Wellbore is an Equipment.
  If we ask for all Equipments, we should return also WB-2025, considering both the data and the knowledge in the ontology.

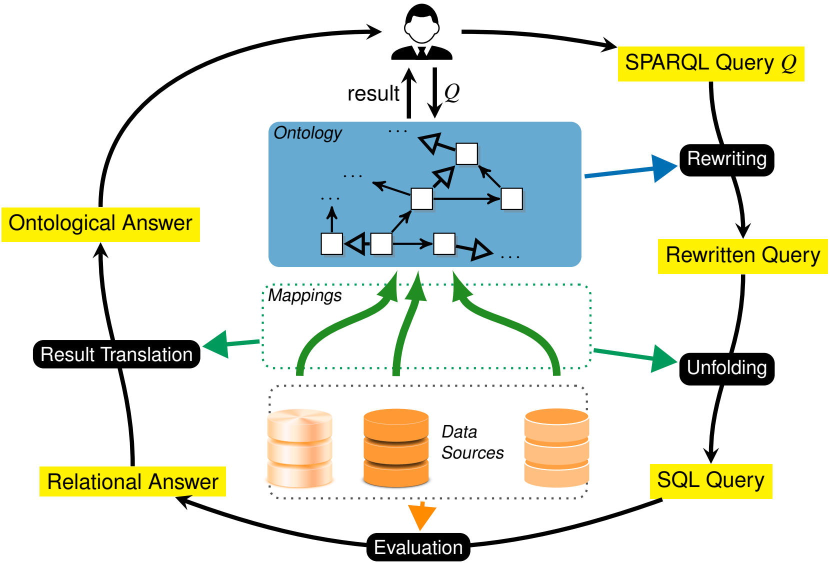
- The **mapping** encodes the information of how to translate a query over the ontology into a query over the **database**.

A VKG query answering engine has to take into account all these types of information.

Query answering by query rewriting

unibz

Query answering by query rewriting



Outline

unibz

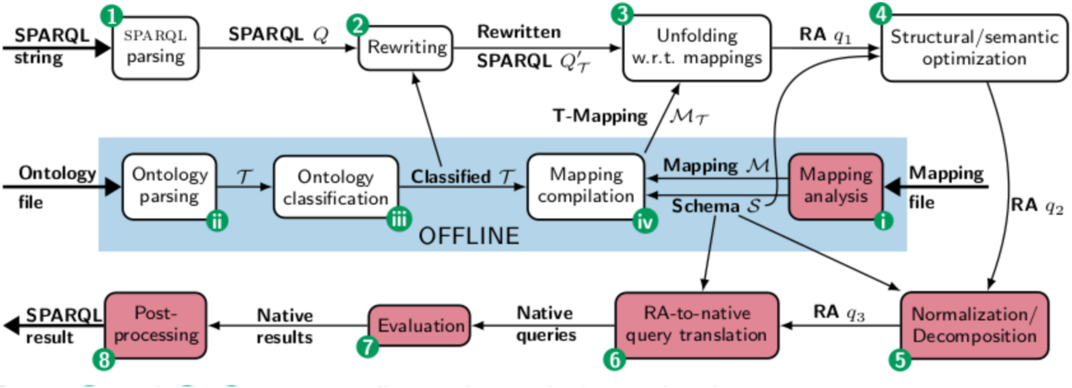
The *Ontop* system [C., Cogrel, et al. 2017, Semantic Web J.], [Xiao, Lanti, et al. 2020, ISWC20]

ontop

https://ontop-vkg.org/

- State-of-the-art VKG system.
- Addresses the key challenges in query answering of scalability and performance.
- Compliant with all relevant Semantic Web standards:
  RDF, RDFS, OWL 2 QL, R2RML, SPARQL, and GeoSPARQL.
- Supports all major relational DBMSs:
  Oracle, DB2, MS SQL Server, Postgres, MySQL, Teiid, Dremio, Denodo, etc.
- Open-source and released under Apache 2 license.

Query answering in *Ontop*



Developer community

unibz

ONTOPIC

Birkbeck UNIVERSITY OF LONDON

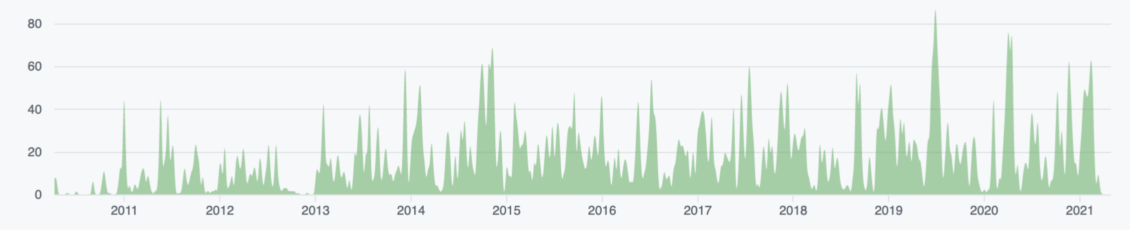
UiO : University of Oslo

HELLENIC REPUBLIC
National and Kapodistrian University of Athens

UNIVERSITÄT LEIPZIG

ontotext

POLITECNICO MILANO 1863



The *Ontopic* spinoff of unibz

https://ontopic.ai/

Funded in April 2019 as the first spin-off of the Free University of Bozen-Bolzano.

- Ontopic Studio just released
  - Ensures scalability, reliability, and cost-efficiency at design and runtime of VKG solutions.
  - Strong focus on usability.
- Technical services
  - Technical support for Ontop and Ontopic Suite.
  - Customized developments.
- Consulting on adoption of VKG-based solutions for data access and integration.

Outline

unibz

Support data analytics in VKGs

Supporting data analytics is currently a top priority for us.

Main challenges addressed in Ontop v4:

- Semantics: computing aggregation functions correctly, in particular those depending on cardinalities (`SUM`, `COUNT`, `AVG`) – bag vs. set semantics is an issue.
- Performance: efficient computation of aggregates, by delegating their execution to the database whenever possible.
- Expressiveness: support user-defined aggregation functions beyond the ones in SPARQL 1.1 (Ongoing).

unibz

Provenance and explanation [C., Lanti, et al. 2019, IJCAI]

- The base version of *Ontop*, does not provide any information about how query answers are constructed.
- In many cases, we are interested in:
  - which data from which relation/source has been used to obtain an answer
  - which mappings have been activated
  - which ontology axioms have contributed to the answer
- We have developed a framework for provenance/explanation in VKGs, building on provenance semi-rings in relational databases.
- We have a prototype extension of *Ontop* that supports this framework.
- We are currently incorporating the framework in the latest release of *Ontop*.

unibz

Geospatial extension [Bereta, Xiao & Koubarakis 2019, J. of Web Semantics]

Spatial data play an important role in many scenarios.

Geo-spatial extension on *Ontop*

- *Ontop* 4 provides full support for accessing geospatial data.
- Supports GeoSPARQL query language standardized by Open Geospatial Consortium (OGC).
- Translates GeoSPARQL functions into functions supported by PostGIS.
- Use cases: urban development, land management, disaster management.

noSQL data sources [Botoeva, C., Cogrel, Corman, et al. 2019, Intelligenza Artificiale]

Prototype extension of *Ontop* over mongoDB. databases.

MongoDB

- Most popular noSQL DBMS.
- Stores data as collections of **JSON** documents.
- Comes with an expressive (low-level) query language: Mongo Aggregate Queries.

Benefits of virtual VKGs over MongoDB:

- **Interface**: higher-level query language (SPARQL) for the end-user.
- **Performance**: *Ontop* delegates query execution to the MongoDB engine
  ⇒ leverages document-based storage.
- Query translation relies on a correspondence between nested-relational algebra and Mongo Aggregate Queries [Botoeva, C., Cogrel & Xiao 2018, ICDT].

Temporal extension [Brandt, C., et al. 2019; Brandt, Güzel Kalayci, et al. 2018; Güzel Kalayci et al. 2019]

Temporal data plays an important role in many scenarios.

- Example 1: find all drillings using the same equipment that are in two different locations with a distance longer than 200 km and within 2 months.
- Example 2: find all customers with at least 3 temporal overlapping loans within the last 5 years.

Ontop-temporal

- A prototype extension of *Ontop* for accessing temporal data.
- Can express complex temporal patterns.
- Use cases: turbine diagnoses, medical records.

Outline

Conclusions

- VKGs are by now a mature technology to address the challenges related to data access and integration.
- It has been well-investigated and applied in many different scenarios mostly for the case of relational data sources.
- The technology is general purpose, and it can be tailored towards specific domains, relying also on standard ontologies.
- Performance and scalability w.r.t. larger datasets (volume), larger and more complex ontologies (variety, veracity), and multiple heterogeneous data sources (variety, volume) is a challenge.
- Recently VKGs have been investigated for alternative types of data, such as **temporal data**, **noSQL** and tree structured data, **linked open data**, and **geo-spatial data**.
- Performance and scalability are even more critical for these more complex domains.

unibz

Thank you!

- E: calvanese@inf.unibz.it
- H: `http://www.inf.unibz.it/~calvanese/`

- *Ontop* website: `https://ontop-vkg.org/`
- Github: `http://github.com/ontop/ontop/`
- Facebook: `https://www.facebook.com/obdaontop/`
- Twitter: @ontop4obda
- *Ontopic* website: `https://ontopic.biz/`

References I

[1] Konstantina Bereta, Guohui Xiao & Manolis Koubarakis. “Ontop-spatial: Ontop of Geospatial Databases”. In: *J. of Web Semantics* 58 (2019). DOI: 10.1016/j.websem.2019.100514.

[2] Elena Botoeva, Diego C., Benjamin Cogrel, Julien Corman & Guohui Xiao. “Ontology-based Data Access – Beyond Relational Sources”. In: *Intelligenza Artificiale* 13.1 (2019), pp. 21–36. DOI: 10.3233/IA-190023.

[3] Elena Botoeva, Diego C., Benjamin Cogrel & Guohui Xiao. “Expressivity and Complexity of MongoDB Queries”. In: *Proc. of the 21st Int. Conf. on Database Theory (ICDT)*. Vol. 98. Leibniz Int. Proc. in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018, 9:1–9:22. DOI: 10.4230/LIPIcs.ICDT.2018.9.

[4] Sebastian Brandt, Diego C., Elem Güzel Kalayci, Roman Kontchakov, Benjamin Mörzinger, Vladislav Ryzhikov, Guohui Xiao & Michael Zakharyaschev. “Two-Dimensional Rule Language for Querying Sensor Log Data: A Framework and Use Cases”. In: *Proc. of the 26th Int. Symp. on Temporal Representation and Reasoning (TIME)*. Vol. 147. Leibniz Int. Proc. in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019, 7:1–7:15. DOI: 10.4230/LIPIcs.TIME.2019.7.

References II

[5] Sebastian Brandt, Elem Güzel Kalayci, Vladislav Ryzhikov, Guohui Xiao & Michael Zakharyaschev. "Querying Log Data with Metric Temporal Logic". In: *J. of Artificial Intelligence Research* 62 (2018), pp. 829–877.

[6] Diego C., Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro & Guohui Xiao. "Ontop: Answering SPARQL Queries over Relational Databases". In: *Semantic Web J.* 8.3 (2017), pp. 471–487. DOI: 10.3233/SW-160217.

[7] Diego C., Davide Lanti, Ana Ozaki, Rafael Peñaloza & Guohui Xiao. "Enriching Ontology-based Data Access with Provenance". In: *Proc. of the 28th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. IJCAI Org., 2019, pp. 1616–1623. DOI: 10.24963/ijcai.2019/224.

[8] Linfang Ding, Guohui Xiao, Albulen Pano, Claus Stadler & Diego C. "Towards the Next Generation of the LinkedGeoData Project using Virtual Knowledge Graphs". In: *J. of Web Semantics* (2022). To appear.

unibz

References III

[9] Elem Güzel Kalayci, Sebastian Brandt, Diego C., Vladislav Ryzhikov, Guohui Xiao & Michael Zakharyaschev. "Ontology-based Access to Temporal Data with Ontop: A Framework Proposal". In: *Applied Mathematics and Computer Science* 29.1 (2019), pp. 17–30. DOI: 10.2478/amcs-2019-0002.

[10] Elem Güzel Kalayci, Irlan Grangel González, Felix Lösch, Guohui Xiao, Anees ul-Mehdi, Evgeny Kharlamov & Diego C. "Semantic Integration of Bosch Manufacturing Data Using Virtual Knowledge Graphs". In: *Proc. of the 19th Int. Semantic Web Conf. (ISWC)*. Vol. 12507. Lecture Notes in Computer Science. Springer, 2020, pp. 464–481. DOI: 10.1007/978-3-030-62466-8_29.

[11] Evgeny Kharlamov, Dag Hovland, et al. "Ontology Based Data Access in Statoil". In: *J. of Web Semantics* 44 (2017), pp. 3–36. DOI: 10.1016/j.websem.2017.05.005.

[12] Evgeny Kharlamov, Theofilos Mailis, et al. "Semantic Access to Streaming and Static Data at Siemens". In: *J. of Web Semantics* 44 (2017), pp. 54–74. DOI: 10.1016/j.websem.2017.02.001.

References IV

[13] Guohui Xiao, Diego C., Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati & Michael Zakharyaschev. "Ontology-Based Data Access: A Survey". In: *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. IJCAI Org., 2018, pp. 5511–5519. DOI: 10.24963/ijcai.2018/777.

[14] Guohui Xiao, Linfang Ding, Benjamin Cogrel & Diego C. "Virtual Knowledge Graphs: An Overview of Systems and Use Cases". In: *Data Intelligence* 1.3 (2019), pp. 201–223. DOI: 10.1162/dint_a_00011.

[15] Guohui Xiao, Davide Lanti, Roman Kontchakov, Sarah Komla-Ebri, Elem Güzel-Kalayci, Linfang Ding, Julien Corman, Benjamin Cogrel, Diego C. & Elena Botoeva. "The Virtual Knowledge Graph System Ontop". In: *Proc. of the 19th Int. Semantic Web Conf. (ISWC)*. Vol. 12507. Lecture Notes in Computer Science. Springer, 2020, pp. 259–277. DOI: 10.1007/978-3-030-62466-8_17.

unibz