Virtual Knowledge Graphs for Data Access and Integration

Diego Calvanese

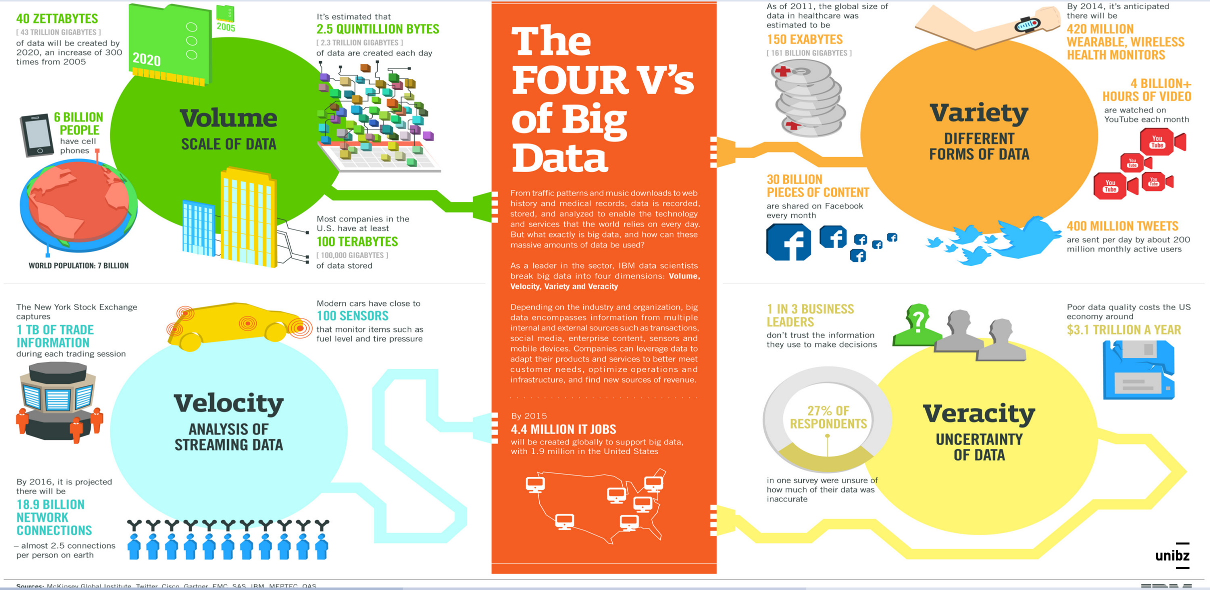
KRDB Research Centre for Knowledge and Data
Free University of Bozen-Bolzano, Italy

Department of Computing Science
Umeå University, Sweden

26th International Computer Conference
Computer Society of Iran

3 March 2021 – Tehran, Iran

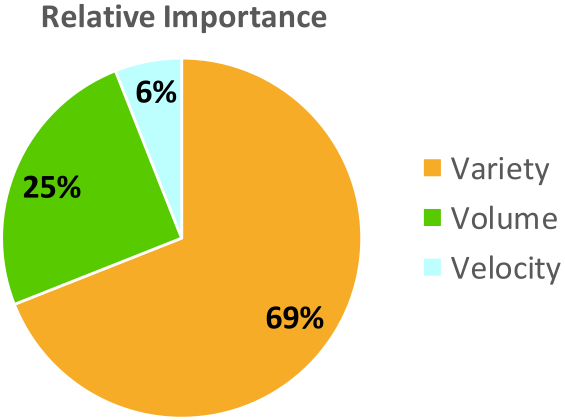
Challenges in the Big Data era



unibz

Variety, not volume, is driving Big Data initiatives

MIT Sloan Management Review (28 March 2016)



http://sloanreview.mit.edu/article/variety-not-volume-is-driving-big-data-initiatives/

How much time is spent searching for the right data?

Important problem: searching for data and establishing its quality

Example: in oil&gas, engineers spend 30–70% of their time on this (Crompton, 2008)

Challenge: Accessing heterogeneous data

Statoil (now Equinor) Exploration

Geologists at Statoil, prior to making decisions on drilling new wellbores, need to gather relevant information about previous drillings.

Slegge relational database:

- Terabytes of relational data
- 1,545 tables and 1727 views
- each with dozens of attributes
- consulted by 900 geologists

Problem: Translating information needs

Information need expressed by geologists

In my geographical area of interest, return all pressure data tagged with key stratigraphy information with understandable quality control attributes, and suitable for further filtering.

To obtain the answer, this needs to be translated into SQL[1]:

- main table for wellbores has 38 columns (with cryptic names)
- to obtain pressure data requires a 4-table join with two additional filters
- to obtain stratigraphic information requires a join with 5 more tables

[1] SQL is the standard DB query language.

Problem: Translating information needs

We would obtain the following SQL query:

```
SELECT WELLBORE.IDENTIFIER, PTY_PRESSURE.PTY_PRESSURE_S,
       STRATIGRAPHIC_ZONE.STRAT_COLUMN_IDENTIFIER, STRATIGRAPHIC_ZONE.STRAT_UNIT_IDENTIFIER
FROM WELLBORE,
     PTY_PRESSURE,
     ACTIVITY FP_DEPTH_DATA
        LEFT JOIN (PTY_LOCATION_1D FP_DEPTH_PT1_LOC
           INNER JOIN PICKED_STRATIGRAPHIC_ZONES ZS
              ON ZS.STRAT_ZONE_ENTRY_MD <= FP_DEPTH_PT1_LOC.DATA_VALUE_1_O AND
                 ZS.STRAT_ZONE_EXIT_MD >= FP_DEPTH_PT1_LOC.DATA_VALUE_1_O AND
                 ZS.STRAT_ZONE_DEPTH_UOM = FP_DEPTH_PT1_LOC.DATA_VALUE_1_OU
           INNER JOIN STRATIGRAPHIC_ZONE
              ON   ZS.WELLBORE = STRATIGRAPHIC_ZONE.WELLBORE AND
                 ZS.STRAT_COLUMN_IDENTIFIER = STRATIGRAPHIC_ZONE.STRAT_COLUMN_IDENTIFIER AND
                 ZS.STRAT_INTERP_VERSION = STRATIGRAPHIC_ZONE.STRAT_INTERP_VERSION   AND
                 ZS.STRAT_ZONE_IDENTIFIER = STRATIGRAPHIC_ZONE.STRAT_ZONE_IDENTIFIER)
           ON FP_DEPTH_DATA.FACILITY_S = ZS.WELLBORE AND
              FP_DEPTH_DATA.ACTIVITY_S  = FP_DEPTH_PT1_LOC.ACTIVITY_S,
     ACTIVITY_CLASS FORM_PRESSURE_CLASS
WHERE WELLBORE.WELLBORE_S = FP_DEPTH_DATA.FACILITY_S AND
      FP_DEPTH_DATA.ACTIVITY_S = PTY_PRESSURE.ACTIVITY_S AND
      FP_DEPTH_DATA.KIND_S = FORM_PRESSURE_CLASS.ACTIVITY_CLASS_S AND
      WELLBORE.REF_EXISTENCE_KIND = 'actual' AND
      FORM_PRESSURE_CLASS.NAME = 'formation pressure depth data'
```

Problem: Translating information needs

We would obtain the following SQL query:

```
SELECT WELL
       STR
FROM WELLB
     PTY_P
     ACTIV
        LE
```

This can be very time consuming, and requires knowledge of the domain of interest, a deep understanding of the database structure, and general IT expertise.

```
           INNER JOIN STRATIGRAPHIC_ZONE
              ON   ZS.WELLBORE = STRATIGRAPHIC_ZONE.WELLBORE AND
```

This is also very costly!

Equinor loses **50.000.000€** per year only due to this problem!!

```
     ACTIV
WHERE WELL
      FP_D
      FP_DEPTH_DATA.KIND_S = FORM_PRESSURE_CLASS.ACTIVITY_CLASS_S AND
      WELLBORE.REF_EXISTENCE_KIND = 'actual' AND
      FORM_PRESSURE_CLASS.NAME = 'formation pressure depth data'
```

Outline

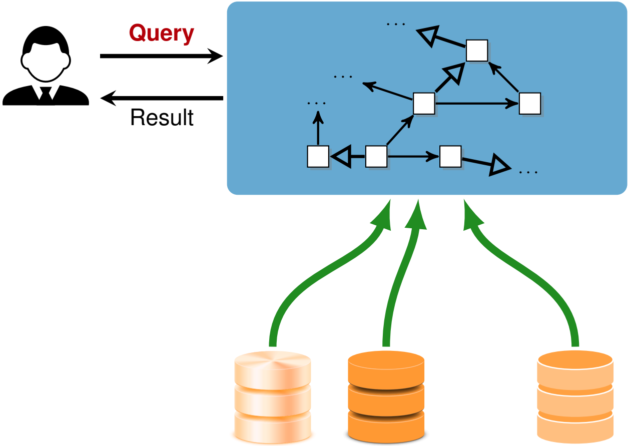
1. Motivations
2. Virtual Knowledge Graphs (VKGs) for data access
3. The *Ontop* system and the *Ontopic* spinoff
4. Ongoing and planned developments
5. Conclusions

unibz

Outline

unibz

Solution: Virtual Knowledge Graphs (VKGs) – Also known as OBDA



Ontology $\mathcal{O}$
data is viewed as a graph, vocabulary of the user

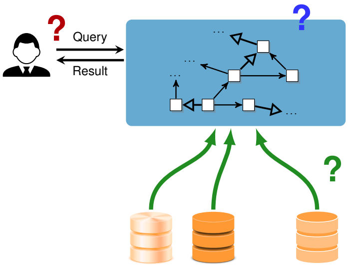
Mapping $\mathcal{M}$
how to populate the ontology from the data

Data Sources $\mathcal{S}$
autonomous and heterogeneous

Greatly simplifies the access to information, and frees end-users from the need to know the precise structure of information sources.

VKG framework – Which languages to use?

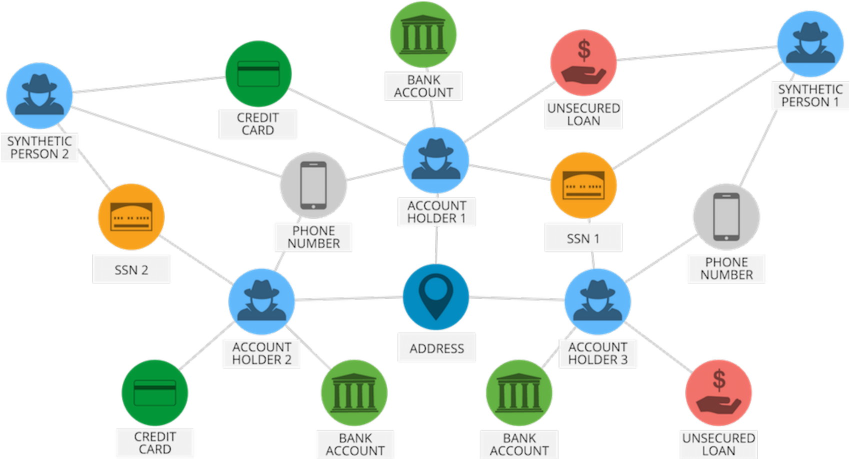
The choice of the right languages needs to take into account the tradeoff between expressive power and efficiency of query answering.



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**.

The OWL 2 QL ontology language

- **OWL 2 QL** is one of the three standard profiles of OWL 2. [W3C Rec. 2012]
- Is considered a lightweight ontology language:
  - controlled expressive power
  - efficient inference
- Optimized for accessing large amounts of data [C., De Giacomo, et al. 2007]
  - Queries over the ontology can be rewritten into SQL queries over the underlying relational database (First-order rewritability).
  - Consistency of ontology and data can also be checked by executing SQL queries.

Main constructs of OWL 2 QL

Class hierarchy: `rdfs:subClassOf` $(A_1 \sqsubseteq A_2)$

Example: `:MovieActor rdfs:subClassOf :Actor .`

Inference: `<person/2> rdf:type :MovieActor .`

$\Longrightarrow$ `<person/2> rdf:type :Actor .`

Domain of properties: `rdfs:domain` $(\exists P \sqsubseteq A)$

Example: `:playsIn rdfs:domain :MovieActor .`

Inference: `<person/2> :playsIn <movie/3> .`

$\Longrightarrow$ `<person/2> rdf:type :MovieActor .`

Range of properties: `rdfs:range` $(\exists P^- \sqsubseteq A)$

Example: `:playsIn rdfs:range :Movie .`

Inference: `<person/2> :playsIn <movie/3> .`

$\Longrightarrow$ `<movie/3> rdf:type :Movie .`

unibz

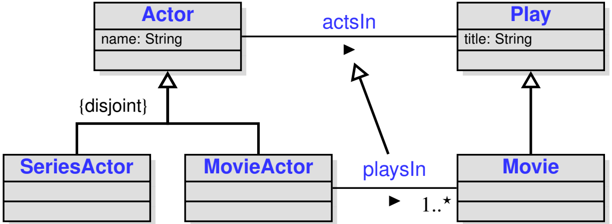
Other constructs of OWL 2 QL

- Class disjointness
- Inverse properties
- Property hierarchy
- Property disjointness
- Mandatory participation

Representing OWL 2 QL ontologies as UML class diagrams/ER schemas

There is a close correspondence between OWL 2 QL and conceptual modeling formalisms, such as UML class diagrams and ER schemas [Berardi, C. & De Giacomo 2005; Bergamaschi & Sartori 1992; Borgida 1995; C., Lenzerini & Nardi 1999; Lenzerini & Nobili 1990; Queralt et al. 2012].

SeriesActor $\sqsubseteq$ Actor	`rdfs:subClassOf`	subclass
SeriesActor $\sqsubseteq$ $\neg$MovieActor	`owl:disjointWith`	disjointness
$\exists$actsIn $\sqsubseteq$ Actor	`rdfs:domain`	domain
$\exists$actsIn$^-$ $\sqsubseteq$ Play	`rdfs:range`	range
MovieActor $\sqsubseteq$ $\exists$playsIn	`owl:someValuesFrom`	mandatory participation
playsIn $\sqsubseteq$ actsIn	`rdfs:subPropertyOf`	sub-association
...		

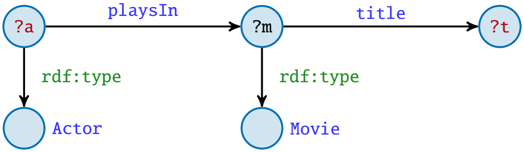


In fact, to visualize an OWL 2 QL ontology, we can use standard UML class diagrams.

SPARQL query language

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

```
SELECT ?a ?t
WHERE { ?a rdf:type Actor .
        ?a playsIn ?m .
        ?m rdf:type Movie .
        ?m title ?t .
      }
```



Additional language features (SPARQL 1.1):

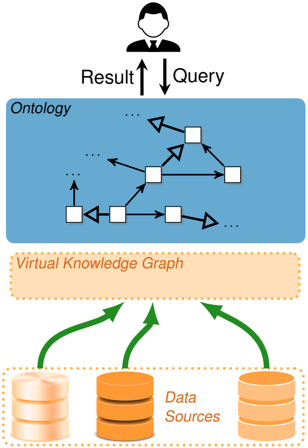
- `UNION`: matches one of alternative graph patterns
- `OPTIONAL`: produces a match even when part of the pattern is missing
- complex `FILTER` conditions
- `GROUP BY`, to express aggregations
- `MINUS`, to remove possible solutions
- property paths (regular expressions)
- ...

Use of mappings

In VKGs, the **mapping** $\mathcal{M}$ encodes how the data $\mathcal{D}$ in the sources should be used to create the virtual knowledge graph.

Virtual knowledge graph $\mathcal{V}$ **defined from** $\mathcal{M}$ **and** $\mathcal{D}$

- Queries are answered with respect to $\mathcal{O}$ and $\mathcal{V}$.
- The data of $\mathcal{V}$ is not materialized (it is virtual!).
- Instead, the information in $\mathcal{O}$ and $\mathcal{M}$ is used to translate queries over $\mathcal{O}$ into queries formulated over the sources.
- Advantage, compared to materialization: the graph is **always up to date** w.r.t. data sources.



Mapping language

The mapping consists of a set of assertions of the form

SQL Query $\rightsquigarrow$ Class

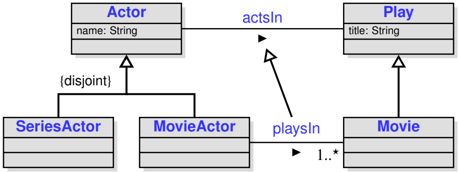
SQL Query $\rightsquigarrow$ Property

Impedance mismatch: values in the DB vs. objects in the knowledge graph

In the right-hand side of the mapping, we make use of **`iri-templates`**, which transform database values into object identifiers (IRIs).

Mapping language – Example

Ontology $\mathcal{O}$:



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} :playsIn :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 :playsIn :m/5118 .    :a/572 :playsIn :m/5118 .    :a/271 :playsIn :m/2281 .
```

unibz

Formalizing VKGs [Poggi et al. 2008; Xiao, C., et al. 2018]

VKG specification $\mathcal{P} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$ and **VKG instance** $\langle \mathcal{P}, \mathcal{D} \rangle$

- $\mathcal{O}$ is an ontology (expressed in OWL 2 QL),
- $\mathcal{M}$ is a set of (R2RML) mapping assertions,
- $\mathcal{S}$ is a (relational) database schema with integrity constraints,
- $\mathcal{D}$ is a database conforming to $\mathcal{S}$.

Semantics:

A first-order interpretation of the ontology predicates is a **model** of $\langle \mathcal{P}, \mathcal{D} \rangle$ if

- it satisfies all axioms in $\mathcal{O}$, and
- contains all facts in $\mathcal{M}(\mathcal{D})$, i.e., retrieved through $\mathcal{M}$ from $\mathcal{D}$.

Note:

- In general, $\langle \mathcal{P}, \mathcal{D} \rangle$ has infinitely many models, and some of these might be infinite.
- However, for query answering, we do not need to compute such models.

Query answering in VKGs – Certain answers

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

Consider our formalization of VKG and a VKG instance $\langle \mathcal{P}, \mathcal{D} \rangle$.

Certain answers

Given a VKG instance $\langle \mathcal{P}, \mathcal{D} \rangle$ and a query q over it, the certain answers to q are those answers that hold in all models of $\langle \mathcal{P}, \mathcal{D} \rangle$.

unibz

First-order rewritability

To make computing certain answers viable in practice, the VKG setting relies on reducing it to evaluating SQL (i.e., first-order logic) queries over the data.

Consider a VKG specification $\mathcal{P} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$.

First-order rewritability [Poggi et al. 2008]

A query R is a **first-order rewriting** of a query Q with respect to $\mathcal{P}$ if, for every source database $\mathcal{D}$,

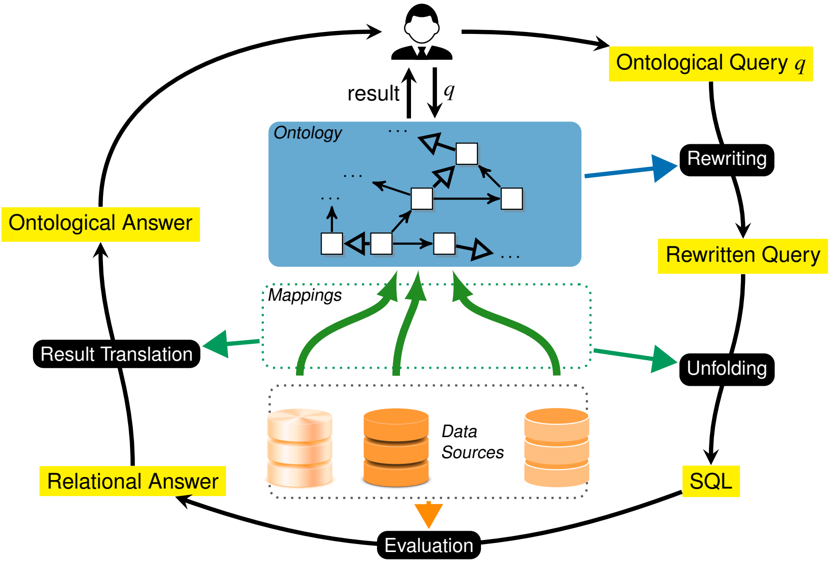
certain answers to Q over $\langle \mathcal{P}, \mathcal{D} \rangle$ = answers to R over $\mathcal{D}$.

For OWL 2 QL ontologies and R2RML mappings,
(core) SPARQL queries are first-order rewritable.

In other words, **in VKGs, we can compute the certain answers** to a SPARQL query by **computing its rewriting**, which is a SQL query, and **evaluating it over the sources**.

unibz

Query answering by query rewriting



Outline

unibz

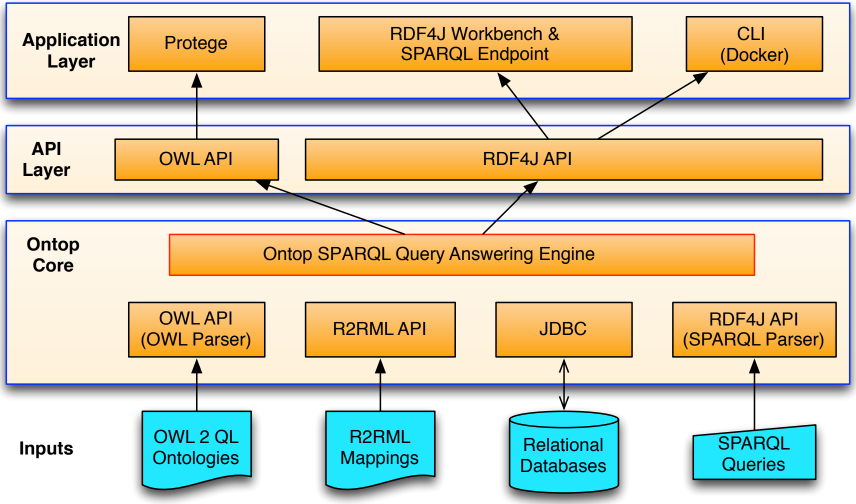
Ontop [C., Cogrel, et al. 2017, SWJ], [Xiao, Lanti, et al. 2020, ISWC]

https://ontop-vkg.org

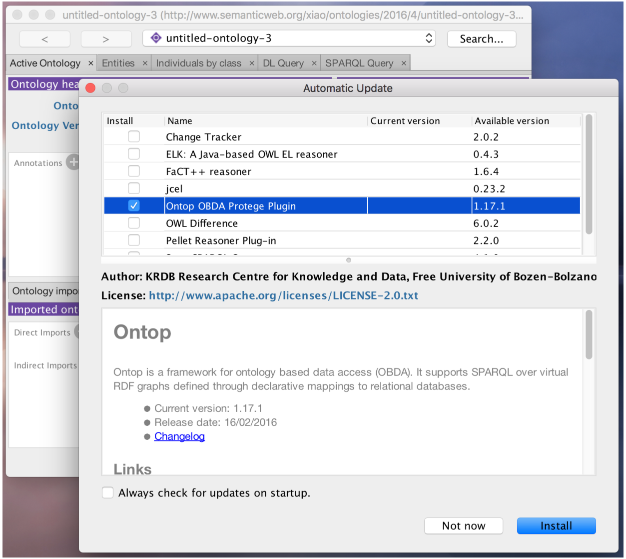
- State-of-the-art system for OBDA and VKGs.
- Compliant with the relevant W3C standard (RDF, OWL 2 QL, R2RML, and SPARQL).
- Supports all major relational DBs.

  Oracle, DB2, MS SQL Server, Postgres, MySQL, Denodo, Dremio, Teiid, etc.
- Open-source and released under Apache 2 license.
- Development of *Ontop*:
  - Development started in 2009.
  - Major v4 just released.
  - Already well established:
    - +200 members in the mailing list
    - +14000 downloads in the last year.
  - Main development carried out in the context of several local, national, and EU projects, and at a university spinoff.

Architecture of *Ontop*

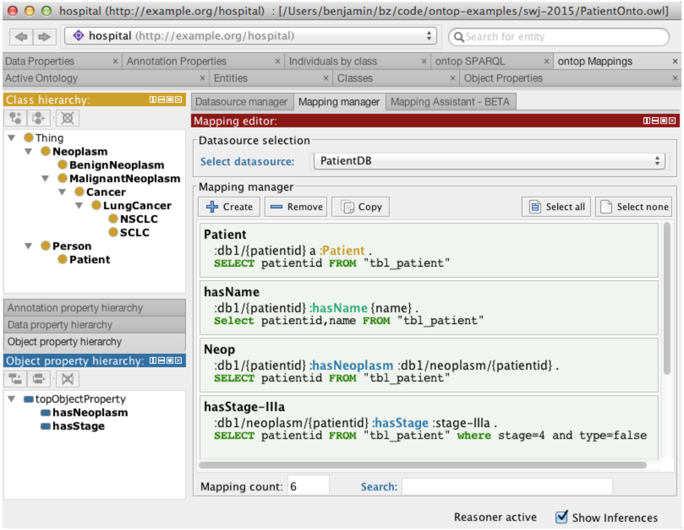


Ontop plugin available from *Protégé* plugin repository



unibz

Mapping editor in *Protégé*



Some use cases of *Ontop* – Research projects

- EU FP7 project Optique "Scalable End-user Access to Big Data" (11/2012 – 10/2016)
  - 10 Partners, including industrial partners Statoil, Siemens, DNV.
  - *Ontop* is core component of the Optique platform.
- EU project EPNet (ERC Advanced Grant) "Production and distribution of food during the Roman Empire: Economics and Political Dynamics"
  - Access to data in the cultural heritage domain .
- Euregio funded project KAOS "Knowledge-aware Operational Support" (06/2016 – 05/2019)
  - Preparation of standardized log files from timestamped log data for the purpose of process mining. [C., Kalayci, et al. 2017]
- EU H2020 project INODE "Intelligent Open Data Exploration" (11/2019 – 10/2022)
  - Development of techniques for the flexible interaction with data.

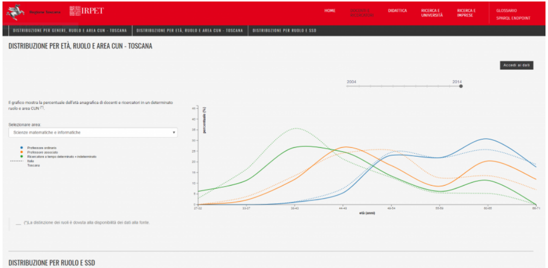
unibz

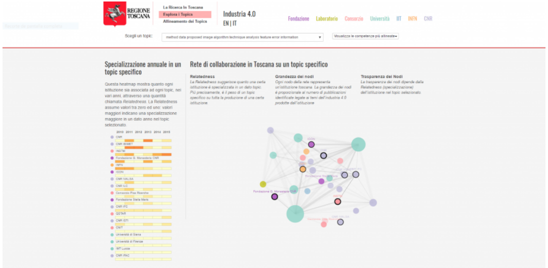
Commercial use cases of *Ontop* in which we are currently involved

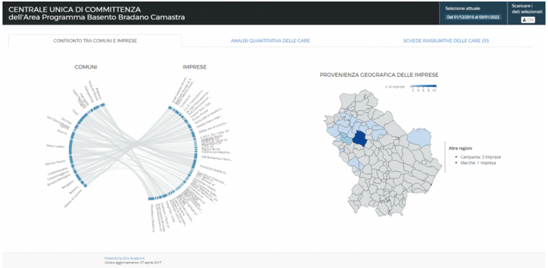
- NOI Techpark in Bolzano – Development of knowledge graph of South Tyrol data [Ding et al. 2020]
  - Tourism data
  - Mobility data
- Collaboration with SIRIS Academic (Barcelona) – Development of data integration and dashboards for data analysis over open data from public institutions
  - Tuscany's Observatory of Research and Innovation
  - Sorbonne University
- Robert Bosch GmbH – Product quality analysis of the Surface Mounting Process pipeline

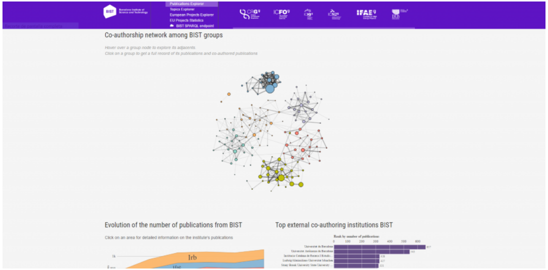
See [Xiao, Ding, et al. 2019] for a survey on VKG systems and use cases.

SIRIS Academic – UNiCS UNiversity AnalytiCS platform









http://www.sirisacademic.com/wb/siris-lab/unics/

unibz

The *Ontopic* spinoff of unibz

https://ontopic.biz/

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

- Ontopic Suite currently under development.
  - 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

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]

- 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*.

Geospatial extension [Bereta, Xiao & Koubarakis 2019]

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]

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].

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 transactions from a same account that are in two different locations with a distance longer than 1000 km and within 5 min.
- Example 2: find all customers with at least 3 temporal overlapping loans within the last 5 years.

Ontop-temporal

- A prototype extension *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 real-world scenarios mostly for the case of relational data sources.
- Also in that setting, 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

Further research directions

Theoretical investigations:

- Dealing with data inconsistency and incompleteness – Data quality!
- Addressing privacy and security issues.
- Ontology-based update.
- Coping with evolution of data in the presence of ontological constraints.

From a practical point of view, supporting technologies need to be developed to make the VKG technology easier to adopt:

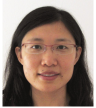
- Improving the support for multiple, heterogeneous data sources.
- Techniques for (semi-)automatic extraction/learning of ontology axioms and mapping assertions [C., Gal, et al. 2020].
- Techniques and tools for efficient management of mappings and ontology axioms, to support design, maintenance, and evolution $\rightsquigarrow$ Ontopic Suite
- User-friendly ontology querying modalities (graphical languages, natural language queries).

Thank you!

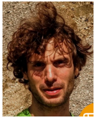
A great thank you to all my collaborators


Elena Botoeva

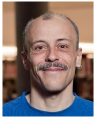

Julien Corman


Linfang Ding


Elem Güzel


Davide Lanti


Marco Montali


Alessandro Mosca


Mariano Rodriguez Muro


Guohui Xiao

Technion Haifa


Avigdor Gal


Roee Shraga

VKGs

Birkbeck College London

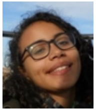

Roman Kontchakov


Vladislav Ryzhikov


Michael Zakharyaschev

Ontopic s.r.l.


Benjamin Cogrel


Sarah Komla Ebri

U. Roma “La Sapienza”


Giuseppe De Giacomo


Domenico Lembo


Maurizio Lenzerini


Riccardo Rosati


Antonella Poggi

unibz

References I

[1] Daniela Berardi, Diego C. & Giuseppe De Giacomo. "Reasoning on UML Class Diagrams". In: *Artificial Intelligence* 168.1–2 (2005), pp. 70–118.

[2] 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.

[3] Sonia Bergamaschi & Claudio Sartori. "On Taxonomic Reasoning in Conceptual Design". In: *ACM Trans. on Database Systems* 17.3 (1992), pp. 385–422.

[4] Alexander Borgida. "Description Logics in Data Management". In: *IEEE Trans. on Knowledge and Data Engineering* 7.5 (1995), pp. 671–682.

[5] 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.

[6] 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 International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018, 9:1–9:22. DOI: 10.4230/LIPIcs.ICDT.2018.9.

References II

[7] 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 2019)*. Vol. 147. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019, 7:1–7:15. DOI: 10.4230/LIPIcs.TIME.2019.7.

[8] 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.

[9] 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.

[10] Diego C., Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini & Riccardo Rosati. "Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family". In: *J. of Automated Reasoning* 39.3 (2007), pp. 385–429.

References III

[11] Diego C., Avigdor Gal, Davide Lanti, Marco Montali, Alessandro Mosca & Roee Shraga. "Mapping Patterns for Virtual Knowledge Graphs (A Report on Ongoing Research)". In: *Proc. of the 33rd Int. Workshop on Description Logics (DL)*. Vol. 2663. CEUR Workshop Proceedings, `http://ceur-ws.org/`. 2020.

[12] Diego C., Tahir Emre Kalayci, Marco Montali & Ario Santoso. "OBDA for Log Extraction in Process Mining". In: *Reasoning Web: Semantic Interoperability on the Web – 13th Int. Summer School Tutorial Lectures (RW 2017)*. Ed. by Giovambattista Ianni, Domenico Lembo, Leopoldo Bertossi, Wolfgang Faber, Birte Glimm, Georg Gottlob & Steffen Staab. Vol. 10370. Lecture Notes in Computer Science. Springer, 2017, pp. 292–345. DOI: `10.1007/978-3-319-61033-7_9`.

[13] 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`.

[14] Diego C., Maurizio Lenzerini & Daniele Nardi. "Unifying Class-Based Representation Formalisms". In: *J. of Artificial Intelligence Research* 11 (1999), pp. 199–240.

References IV

[15] Diego C., Pietro Liuzzo, Alessandro Mosca, Jose Remesal, Martin Rezk & Guillem Rull. "Ontology-Based Data Integration in EPNet: Production and Distribution of Food During the Roman Empire". In: *Engineering Applications of Artificial Intelligence* 51 (2016), pp. 212–229. DOI: 10.1016/j.engappai.2016.01.005.

[16] Linfang Ding, Guohui Xiao, Diego C. & Liqiu Meng. "A Framework Uniting Ontology-based Geodata Integration and Geovisual Analytics". In: *Int. J. of Geo-Information* (2020). To appear.

[17] 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.

[18] Maurizio Lenzerini & Paolo Nobili. "On the Satisfiability of Dependency Constraints in Entity-Relationship Schemata". In: *Information Systems* 15.4 (1990), pp. 453–461.

[19] Antonella Poggi, Domenico Lembo, Diego C., Giuseppe De Giacomo, Maurizio Lenzerini & Riccardo Rosati. "Linking Data to Ontologies". In: *J. on Data Semantics* 10 (2008), pp. 133–173. DOI: 10.1007/978-3-540-77688-8_5.

References V

[20] Anna Queralt, Alessandro Artale, Diego C. & Ernest Teniente. "OCL-Lite: Finite Reasoning on UML/OCL Conceptual Schemas". In: *Data and Knowledge Engineering* 73 (2012), pp. 1–22.

[21] 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.

[22] 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.

[23] 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.