The most prominent knowledge bases do not support SPARQL time-traversal queries

Verifiability, not truth. We need provenance

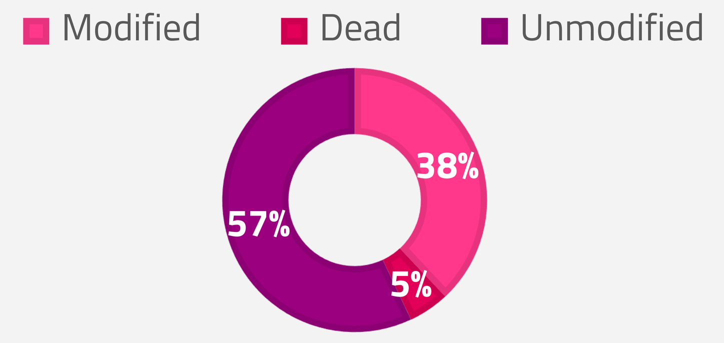
"The threshold for inclusion in Wikipedia is verifiability, not truth" (Garfinkel 2008)

- **Trustworthiness** must be evaluated by each application probing by the **context** of the statements
- **Provenance**: people, institutions, entities, and activities involved in producing, influencing, or delivering data

Data evolves. We need change-tracking

- Natural **evolution** of **concepts**, related to a change of place, jurisdiction, subjective perception of the receiver
- Correction of **mistakes**
- The latest version of knowledge may not be the most accurate

90,000 RDF DOCUMENTS MONITORED FOR 29 WEEKS (KÄFER ET AL. 2013)



We need provenance and change-tracking in RDF

- The **most extensive RDF datasets** – DBPedia, Wikidata, Yago, and the Dynamic Linked Data Observatory – do **not** use RDF to **track changes** and **provenance**. Some of them, such as YAGO 4, record provenance but not changes
- Therefore, such knowledge bases do not allow users to perform time-traversal SPARQL queries

Objective

- A methodology to perform **SPARQL time-traversal queries** on RDF datasets and **software** based on this procedure
- Enabling all the time-related **retrieval functionalities** defined by (Fernández et al. 2016) **live**

Time-related retrieval functionalities

- **Version materialization (VM)**: retrieves data using a query targeted at a single version
- **Delta materialization (DM)**: retrieves query's result change sets between two versions
- **Version query (VQ)**: annotates a query's results with the versions in which they are valid
- **Cross-version join (CV)**: joins the results of queries between different versions
- **Change materialization (CM)**: returns a list of versions in which a given query produces consecutively different results

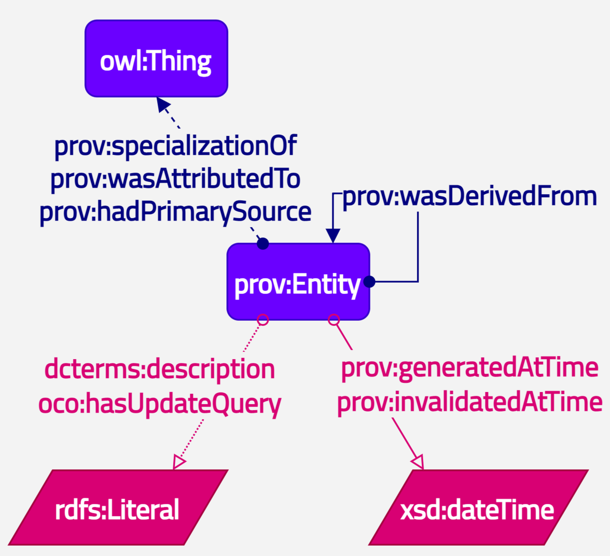
The OpenCitations provenance model was adopted to manage both provenance and change-tracking

Representing provenance in RDF

- Origin of the **first sin**: RDF 1.0 and **RDF Reification**
- Many alternatives have no concrete implementation and are only **theoretical models** (Sikos et al. 2020)
- **Named Graphs** and the **Provenance Ontology** are the most adopted approaches
- Recent promising solution: **RDF***

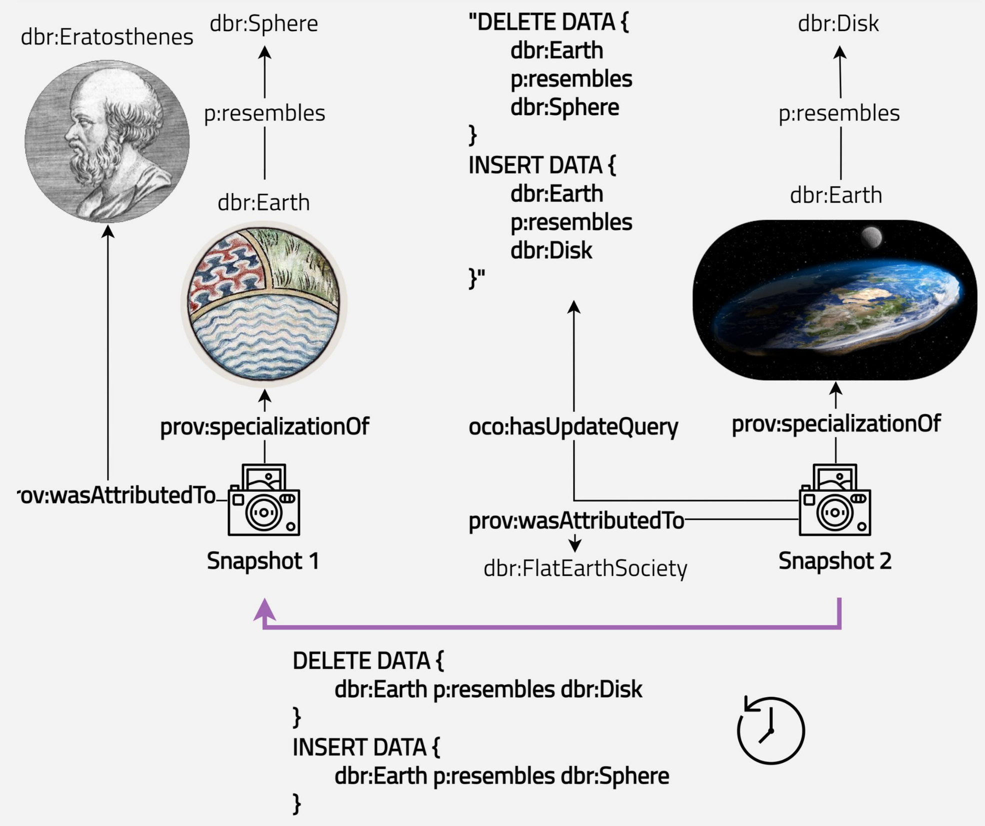
OpenCitations Data Model (OCDM)

- It combines **Named Graphs** and the **Provenance Ontology** (Daquino et al.)
- **Change-based** storage policy: it stores deltas using **SPARQL INSERT DATA** and **DELETE DATA** operations



How to rebuild past versions and achieve VM, VQ, and CV

- According to the OCDM, **only deltas** are **stored**; therefore, the dataset's **past conditions** must be **reconstructed** to query those states.
- However, restoring as many versions as snapshots would generate **massive** amounts of data, consuming time and storage. The adopted solution was to reconstruct only the past resources **significant** for the user's query
- To **recover** the status of an entity to a particular snapshot s_i, apply the **inverse update queries** from the most recent snapshot s_n to s_{i+1}



Delta and change materializations are straightforward

The procedure is similar to the one for version queries, but once the relevant entities have been found, there is no reason to rebuild their past conditions because **deltas** are **explicitly stored** according to the OCDM

Time-Agnostic Library enables all the time-related retrieval functionalities via SPARQL live

Time-agnostic-library

- Such methodology was implemented in a **Python package**, available **open-source** on GitHub under ISC License and downloadable via **pip**
- **Test driven development**: 141 tests (98% coverage)

Evaluation of time-agnostic-library

- Two benchmarks: **execution times** and **memory**
  - The **evaluation dataset** contains 4,960,087 data triples and 19,348,027 provenance triples
  - **10 queries** repeated **10 times** on **20 random entities** having variable number of **provenance snapshots** (from **2** to **35**, avg 20, stdev 8)
  - Benchmarks were performed on **Blazegraph**, **GraphDB Free Edition**, **Apache Jena Fuseki**, and **OpenLink Virtuoso**
  - The benchmarks are fully **reproducible** by simply running a bash script (Massari 2022)

Retrieval functionality	Time on Fuseki			RAM on Fuseki		
	Min	Median	Max	Min	Median	Max
VM, VQ	0.294s	10.9s	11.5s	33.5MB	94.5MB	186MB
CV	0.294s	14.3s	15.6s	33.4MB	157MB	519MB
CV with unkown subj	199s	237s	240s	262MB	264MB	696MB
DM, CM	0.414s	13.0s	13.8s	37.0MB	74.4MB	74.7MB
DM, CM with unkown subj	138s	177s	172s	74.2MB	89.5MB	192MB

- Like **R&Wbase**, **R43ples**, and **QuitStore, time-agnostic-library** achieves time-agnostic SPARQL queries **live**, that is, **without** requiring **pre-indexing processes**
- Unlike them, time-agnostic-library is compatible with any triplestore and is compliant with RDF 1.1 and SPARQL 1.1

Software	Queries	Live
SemVersion	VM, DM	+
X-RDF-3X	VM, DM	-
RBDMS	All	-
R&Wbase	**All**	**+***
R43ples	**All**	**+****
TailR	VM, VQ, CV, CM	+
V-RDFCSA	VM, DM, V	-
Dydra	All	-
OSTRICH	All	-
QuitStore	**All**	**+*****
Time-agnostic-library	**All**	**+**

* It works only on Fuseki and Virtuoso

** It saves data in 5-tuples and extends SPARQL

*** It uses an in-memory custom quadstore and has huge memory requirements

PERFORMING LIVE TIME-TRAVERSAL QUERIES ON RDF DATASETS

Arcangelo Massari
arcangelo.massari@unibo.it — https://orcid.org/0000-0002-8420-0696

Digital Humanities and Digital Knowledge, University of Bologna
Supervisor Prof. Silvio Peroni