Knowledge Graph publication and browsing using Neo4J*

Ghislain Atemezing$^{1[0000-0003-1562-6922]}$ and Anh Huynh^{1}

MONDECA, 18 rue de Londres, 75009, Paris, France.
`firstname.lastname@mondeca.com`

Abstract. Publishing 5-stars datasets on the Web requires RDF triple stores as the backend for managing knowledge graphs. Some most popular endpoints for serving RDF include OpenLinkSW Virtuoso, Apache Jena/Fuseki, Ontotext GraphDB, and Stardog. However, in some cases in enterprise settings, publishers using graph databases such as property graphs (PG) also need to publish their RDF data assets on the Web. This paper proposes a Knowledge Browser (aka `KBrowser`), a tool for publishing and browsing multilingual RDF knowledge graphs using Neo4J as backend. The first experiments on loading RDF files show promising results on sample relatively large open RDF datasets. We report two successful deployments of `KBrowser` for publishing taxonomies on the Web.

Keywords: Knowledge Browser · Neo4J · Taxonomy Portal · RDF · Graph Databases.

1 Introduction

Knowledge graphs (KGs) are critical to many enterprises as they provide the structured data and factual knowledge to enable smart applications [6]. Semantic Web technologies provide open standards to create reference schema or ontology, and the RDF model to generate KGs in terms of triples generally stored in triple stores. In the recent years, graph databases using property graph (PG) models [1] have contributed to partially solved some weaknesses in RDF landscape, such as the ability to easily annotate properties, to make traversal queries and visual appealing graph relations and nodes. RDF triple stores and property graph databases are two approaches for data management which are based on modeling, storing and querying graph-like data. Our research question is the following: How can we integrate a PG database in the publication pipeline of an enterprise KG encoded in RDF? Our contribution is three-fold: (1) We propose a set of mappings from RDF to Neo4J, (2) We implement and evaluate an RDF loader to Neo4J and (3) We showcase two terminology portals on the Web using our proposal.

This paper presents `KBrowser`, a pipeline tool leveraging a PG database (aka Neo4J) to publish and visualize any KG encoded in RDF. Section 2 presents

* Copyright © 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

the mappings used for our implementation, followed by the description of the architecture (section 3). Section 4 presents the loading experiments, completed by applications in Section 5. We conclude the paper in Section 7 after a brief discussion (Section 6).

2 Mapping rules

Our model defines three main rules for mappings from the RDF dataset to Neo4J PG model.

Rule 1 *Subjects of triples are mapped to nodes in Neo4j. A node in Neo4j representing an RDF resource will be labeled :Resource and have a property uri with the resource's URI.* $(S, P, O) => (: Resource\{value : S, kind : uri\})...$

Example 1. Let's consider the following triples[1]

```
dbr:Mondeca rdf:type dbo:Agent, dbo:Organisation .
```

When applying rule 1 above, we obtain the following Mondeca subject resource:

```
(:Resource {value:'dbr:Mondeca',kind:'uri'})
```

Rule 2 *Predicates of triples are mapped to relationships in Neo4j if the object of the triple is a literal.* $(S, P, O)\&\&isLiteral(O) => (: Resource\{value : S, kind : uri\}) - [: P] - > (: Resource\{value : O, kind : literal\})$

This rule is generic enough to handle multi valued properties, data types of literals, multilingual values in literals. Therefore, it is much easier to add in a given node the properties of Neo4J such as language tag, and type of a literal.

Example 2 (Representing multilingual information). In the following triples:

```
dbr:Mondeca rdfs:label "Mondeca"@fr ,
"Mondeca"@en.
```

When applying this rule, we get the output below:

```
(:Resource {value:'dbr:Mondeca',kind:'uri'})-[:'rdfs:label'`]->
(:Resource {value:'Mondeca',kind:'literal',lang:'fr'}   )
(:Resource {value:'dbr:Mondeca',kind:'uri'})-[:'rdfs:label']->
(:Resource {value:'Mondeca',kind:'literal',lang:'en'}   )
```

Rule 3 *Predicates of triples are mapped to relationships in Neo4J if the object of the triple is a resource.* $(S, P, O)\&\&!isLiteral(O) => (: Resourcevalue : S, kind : uri)-[: P]- > (: Resource\{value : O, kind :' uri'\})$

Example 3. In the following example:

[1] Namespaces used are the ones defined in the DBpedia N3 file

```
dbr:Mondeca dbo:location dbr:Paris.
```

. After applying the third rule, we get the following output:

```
(:Resource {value:‘dbr:Mondeca‘,kind:‘uri‘})−[:‘dbo:location‘]−>
(:Resource {value:‘dbr:Paris‘,kind:‘uri‘})
```

3 `KBrowser` architecture overview

`KBrowser` is deployed on a Web Server. The Graph Database is Neo4J, with Cypher used as the internal query language. It uses the implementation to load RDF data defined with the mappings in Section 2. Elasticsearch (ES) is used as the search engine. The REST API module allows transparent management of the repositories, graphs and search strategies. RDF is the exchange data model between the components. The overall architecture is depicted in Figure 1. Our implementation of the RDF import into Neo4J is bundled into `KBrowser` before the indexing phase by ES.

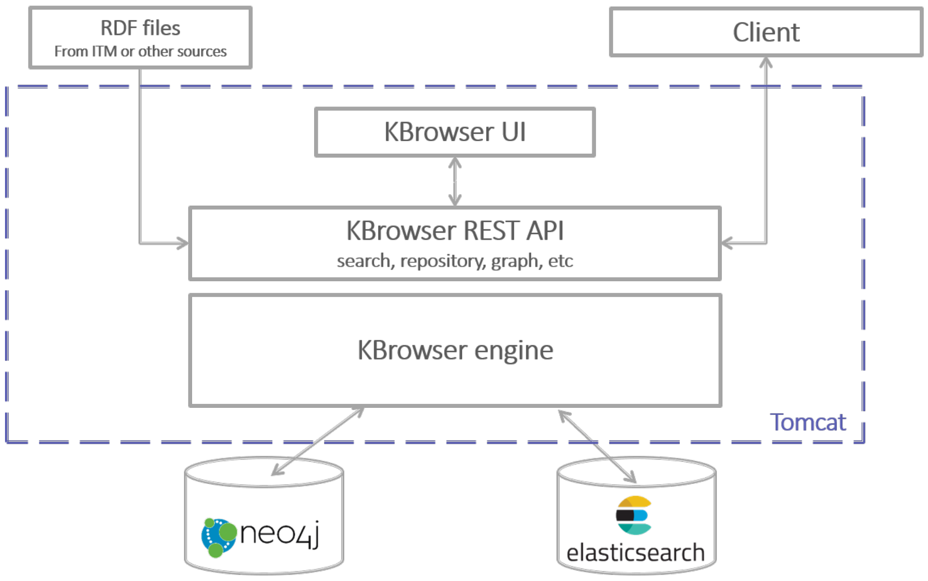


Fig. 1: `KBrowser` Architecture Overview

4 Experiments

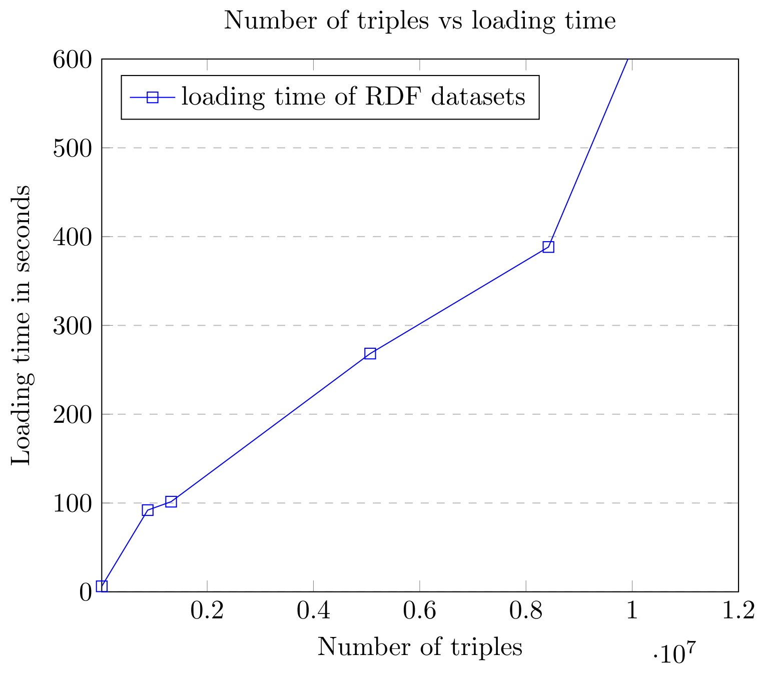
We report an evaluation of loading time with our first implementation in Java of the plugin to load RDF sample data into Neo4J. In this experiment, we use Neo4J 3.5.6 community edition with the following configuration of the memory heap size of 8GB and page cache memory of 1GB. For each loading process, we go through the following steps:

– Step 1: Connect to Cypher shell with an empty database.
– Step 2: Creation of 5 indexes as follows:

```
create index on :INDEXED(p); create index on :INDEXED(c);
create index on :INDEXED(cp);create index on :INDEXED(kind);
create index on :INDEXED(value);
```

– Step 3: Call the procedure with three parameters: the file, the serialization, and the number of triples per transaction set to 50K triples.

We use for testing our import procedure one ontology (schema.org)[2] and random sample RDF dumps datasets of DBpedia from 2015[3].



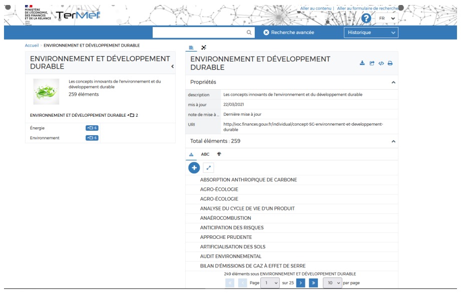
5 Applications scenario

KBrowser is mainly used as a web-based portal to provide read-only access to large audiences of users who need to visualize, search, navigate and browse collections of enterprise data and content. This section reports two successful deployments portals in the domain of finance (TerMef) and eHealth (Bioloinc). The portals are powered by KBrowser and are accessible online.

[2] https://schema.org/docs/developers.html
[3] http://downloads.dbpedia.org/2015-10/core-i18n/en/

5.1 TerMef portal

The TerMEF[4] `KBrowser` is published by the French Ministry of Finance. It supports the publication and dissemination on the Web of the ministry's controlled vocabularies to a broad audience of analysts and domain experts.


Ministère de l'Économie, des Finances et de la Relance
TerMef
Aller au contenu | Aller au formulaire de recherche
FR
Recherche avancée
Historique
Accueil > ENVIRONNEMENT ET DÉVELOPPEMENT DURABLE

ENVIRONNEMENT ET DÉVELOPPEMENT DURABLE

Les concepts innovants de l'environnement et du développement durable
259 éléments

ENVIRONNEMENT ET DÉVELOPPEMENT DURABLE 2

Énergie	6
Environnement	6

ENVIRONNEMENT ET DÉVELOPPEMENT DURABLE

Propriétés

description	Les concepts innovants de l'environnement et du développement durable
mis à jour	22/03/2021
note de mise à …	Dernière mise à jour
URI	http://voc.finances.gouv.fr/individual/concept-SG-environnement-et-developpement-durable

Total éléments : 259

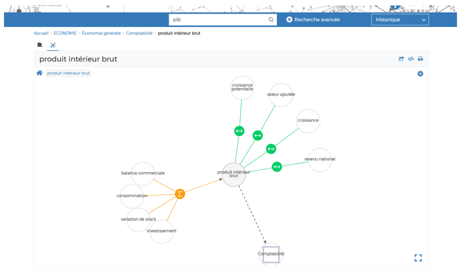
ABC

ABSORPTION ANTHROPIQUE DE CARBONE
AGRO-ÉCOLOGIE
AGRO-ÉCOLOGIE
ANALYSE DU CYCLE DE VIE D'UN PRODUIT
ANAÉROCOMBUSTION
ANTICIPATION DES RISQUES
APPROCHE PRUDENTE
ARTIFICIALISATION DES SOLS
AUDIT ENVIRONNEMENTAL
BILAN D'ÉMISSIONS DE GAZ À EFFET DE SERRE

249 éléments sous ENVIRONNEMENT ET DÉVELOPPEMENT DURABLE

Page 1 sur 25 10 par page

(a) Tabular view of a concept



(b) Graph visualization

The portal offers to users a list of groups of concepts, facets lists and a navigation view as depicted in Figures 2a and 2b. It is composed of 21 `SKOS:ConceptScheme` representing different domains. The portal contains nearly $200K$ triples.

5.2 Bioloinc portal

ASIP Santé (French Agency for Digital Health)'s `KBrowser` instance is used as the French reference Terminology Server for the publication, dissemination and reuse of the international BIOLOINC terminology. LOINC is a reference terminology created and maintained by the Regenstrief Institute [4]. Nowadays, LOINC is a clinical terminology used for recording health measurements, observations, and documents. The portal[5] supports health stakeholders in implementing consistent indexing of patient records as defined by the nationwide shared medical records initiative. The deployed server uses a maximum memory of 4GB, with less than 1 million triples. `KBrowser` serves approximatively $54,500$ French LOINC terms and almost $1,019$ nomenclature of medical biology procedures.

6 Discussion

This work was inspired by the work of Neo4J Labs [3] and the Tinkerpop sail view approach.[6]. The main difference of our mappings compared to the work in [3] is the generic approach for handling multilingual information contained in the RDF. Additionally, we argue that the proposed mappings deal with any type of RDF asset, including OWL ontologies. Our mapping is also closed to

[4] http://terminologie.finances.gouv.fr/
[5] https://bioloinc.fr/
[6] https://github.com/tinkerpop/blueprints/wiki/Sail-Implementation

the database schema independent matching from RDF to PG described in [2]. Nevertheless, we do not have yet a mechanism to validate the data inserted into Neo4J. This can be mitigated by doing such tasks (reasoning, validation) in the RDF before loading into Neo4J. The time spent for loading is promising with less than a hundred million triples. Nevertheless, we need to improve our loader implementation to cope with billions of triples with a relatively decent memory. `KBrowser` usage as Web portals support search functions and visual graph visualizations appealing to users.

7 Conclusions

This work contributes to bridge the gap of consuming RDF datasets in Neo4J. We proposed a set of mappings from RDF to Neo4J, with an implementation to load any serialization of RDF for querying in Cypher. `KBrowser` is used as a pipeline for publishing RDF enterprise assets, browsing and searching on the Web. We present two use cases of portals supporting our approach.

Among the limitations of the system presented in this paper we can mention: the hypothesis of not handling Blank Nodes; the high memory consumption and speed of the loader when handling billions of triples. Future works include the integration of loading RDF-star datasets according to the recent specifications [5], and a thorough evaluation of existing proposals in the literature. The adoption of RDF-star will reduce the gap between RDF and PG databases, but it will not necessarily solve use cases of integrating RDF data into graph databases, or making fully accessible SPARQL endpoints with PG backends.

Acknowledgments. This work has been partially funded by the project "Disinformation Identification in Evolving Knowledge Bases" (DIEKB) under grant n° DGA01D19018444.

References

1. Angles, R.: The property graph database model. In: AMW (2018)
2. Angles, R., Thakkar, H., Tomaszuk, D.: Mapping rdf databases to property graph databases. IEEE Access **8**, 86091–86110 (2020)
3. Barrasa, J.: A step-by-step example of rdf to property graph transformation (2016), https://jbarrasa.com/2016/09/09/quickgraph3-a-step-by-step-example-of-rdf-to-property-graph-transformation/
4. Forrey, A.W., Mcdonald, C.J., DeMoor, G., Huff, S.M., Leavelle, D., Leland, D., Fiers, T., Charles, L., Griffin, B., Stalling, F., et al.: Logical observation identifier names and codes (loinc) database: a public use set of codes and names for electronic reporting of clinical laboratory test results. Clinical chemistry **42**(1), 81–90 (1996)
5. Hartig, O., al. (eds): Rdf-star and sparql-star (June 2021), https://w3c.github.io/rdf-star/cg-spec
6. Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J.: Industry-scale knowledge graphs: lessons and challenges: five diverse technology companies show how it's done. Queue **17**(2), 48–75 (2019)