Towards a more human-friendly knowledge graph generation & publication*

Dylan Van Assche(✉), Thomas Delva,
Pieter Heyvaert, Ben De Meester, and Anastasia Dimou(✉)

IDLab, Department of Electronics and Information Systems,
Ghent University – imec Technologiepark-Zwijnaarde 122, 9052 Ghent, Belgium
{dylan.vanassche,thomas.delva,pieter.heyvaert,
ben.demeester,anastasia.dimou}@ugent.be

Abstract. Human-friendly representations of mapping languages, e.g., YARRRML or ShExML, allow seamlessly integrating heterogeneous data into knowledge graphs. However, they do not describe how a generated knowledge graph should be exported or published, leaving the knowledge graph generation and publication disconnected. To address this, we recently introduced the Logical Target in RML, but an alignment with the human-friendly representations of mapping languages is still pending. In this demo paper, we (i) align RML's Logical Target with YARRRML to provide a more human-friendly syntax for knowledge graph's generation and publication and (ii) demonstrate our approach in YARRRML's editor Matey. YARRRML users are now able to describe how their generated knowledge graphs are exported to one or multiple targets, paving the path towards a more reproducible and human-friendly workflow for generating and publishing knowledge graphs.
Demo: https://w3id.org/yarrrml/matey
Screencast: https://w3id.org/yarrrml/matey/screencast-target
Repository: https://w3id.org/yarrrml/matey/repository

1 Introduction

Human-friendly representations of mapping languages, such as YARRRML [5], or ShExML [4], abstract mapping languages' machine-readable syntax e.g. W3C Recommendation R2RML [2] for generating knowledge graphs (KG) from relational databases represented in the Resource Description Framework (RDF) [1], or the RDF Mapping Language (RML) [3] which broadens the scope of R2RML to heterogeneous data sources. These human-friendly representations facilitate the integration of heterogeneous data sources into a knowledge graph in a human-friendly way. Other mapping languages, such as SPARQL-Generate [6], claim to be less complex by using other Semantic Web recommendations as their basis, e.g., SPARQL [6] for SPARQL-Generate.

These human-friendly representations need to follow the evolution of mapping languages to make all their features available. For instance, we recently extended RML with the Logical Target to describe how a KG should be exported to one or multiple targets [7]. Such new features may be more easily adopted if these human-friendly representations support such changes, since most users rely on such human-friendly representations to define how KGs should be generated.

In the case of Logical Target, neither YARRRML nor ShExML, the two most common human-friendly representations of RML, were not extended yet to define how and where to export a KG. Consequently, their users still resort to hard-coded scripts to bridge this gap, leaving the KG generation and publication disconnected, even though it is addressed by the underlying mapping language.

In this demo paper, we (i) introduce a YARRRML syntax for RML's Logical Target[1] and (ii) demonstrate our approach in Matey, an editor for YARRRML [5]: http://w3id.org/yarrrml/matey. Thanks to our approach, not only mapping languages, but also users of their human-friendly representations, can specify how their generated KG should be exported and published to one or multiple targets.

2 Logical Target alignment with YARRRML

We extend YARRRML's syntax to define how and to where a KG, represented in the RDF [1], should be exported. We added a `targets` map to specify Logical Targets in YARRRML with a similar description as a `sources` map (Listing 1.1). This approach is straightforward because Logical Target uses a similar description as Logical Source in RML [7]. We integrated this YARRRML extension with RML's Logical Target in the YARRRML specification: http://w3id.org/yarrrml/spec.

Targets can be defined in the same way as sources, both describe how a source or target is accessed (Listing 1.1). YARRRML mappings may refer to targets in `subjects`, `predicates`, `objects`, or `graphs` maps to describe where RDF triples of the generated KG are exported.

Each target contains at least an access description and type to specify how the target should be accessed (Listing 1.1: lines 5-6, 9). The serialization format and compression are optional (Listing 1.1: lines 7-8). By default, the serialization format is N-Quads and no compression is applied, following the DataIO specification[2]. In Listing 1.1, we define: (i) a local file[3] to which the triples are dumped with Turtle as serialization format and GZip compression, and (ii) a SPARQL endpoint to which part of the KG is exported to using SPARQL UPDATE.

[1] RML's Logical Target alignment is integrated in the YARRRML specification: http://w3id.org/yarrrml/spec
[2] DataIO specification: http://rml.io/specs/dataio
[3] VoID Dataset: http://www.w3.org/TR/void

Listing 1.1. Targets follow a similar syntax as sources. The SPARQL endpoint target is written in YARRRML's shortcut syntax.

```
sources:
  people-source: ["/data/people.json~jsonpath", "$.[*]"]
targets:
  file-target:
    access: /data/file.ttl.gz
    type: void
    serialization: turtle
    compression: gzip
  sparql-target: ["https://example.org/sparql~sparql"]
```

Each YARRRML mapping may contain a `targets` map with one or multiple targets to describe where each RDF triple of the generated KG is exported to.

Triples are exported to the targets as specified in the mappings (Listing 1.2: lines 6, 10). If no targets are specified for a certain triple, the triple is exported to the mapping language processor's default target. When multiple targets are specified for the same triple, it is exported to all its targets. This approach allows fine grained control over where each triple is exported to, following RML's Logical Target extension [7]. In Listing 1.2, the local file target is referred to in `subjects` map of the `people` mappings (Listing 1.2: line 6), all triples containing these subjects will be exported to that local file. The SPARQL endpoint target is only specified in `predicates` map of the `people` mapping, only the triples containing the predicate `foaf:name` (Listing 1.2: line 10) are exported to the SPARQL endpoint.

Listing 1.2. Triples generated by a mapping matching multiple targets are exported to all targets. Triples without a target are exported to the default target.

```
mappings:
  people:
    sources: people-source
    subjects:
      - value: "http://example.org/person/$(id)"
        targets: file-target
    predicateobjects:
      - predicates:
          - value: foaf:name
            targets: sparql-target
        objects: "$(name)"
      - predicates: foaf:age
        objects: "$(age)"
```

3 Matey: Human-friendly KG generation & publication

We extended YARRRML's editor Matey to demonstrate how a KG is exported to multiple targets such as local files or SPARQL endpoints. We added support for (i) RML's Logical Target extension in YARRRML and (ii) multiple outputs (Figure 1). Each target is shown in Matey in the output tab, the drop-down menu provides the user the possibility to show different targets. Matey is available at http://w3id.org/yarrrml/matey and a demo screencast at http://w3id.org/yarrrml/matey/screencast-target which shows how targets are

defined and how this affects the KG publication. Our extensions to Matey are available under the MIT license: http://w3id.org/yarrrml/matey/repository.

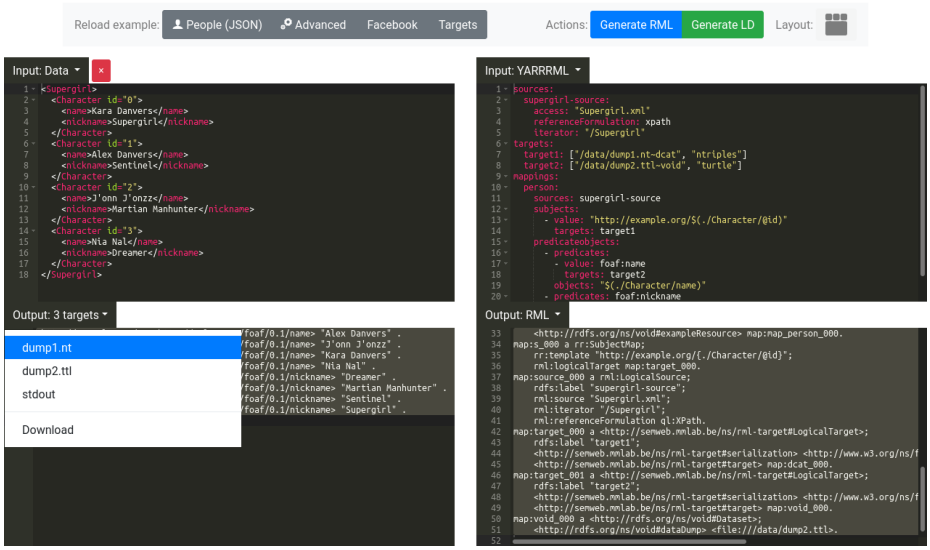

Fig. 1. Targets are chosen in Matey through the output tab's drop-down menu.

4 Conclusion

Through RML's Logical Target alignment with YARRRML, users of these human-friendly representations of mapping languages do not need to resort to hard-coded scripts anymore to export their KG to various targets, but they can specify how their generated KG should be exported. Currently, we implemented a few targets such as local files or SPARQL endpoints, but our implementation can be extended to support more targets. Future research is needed to make sure that these human-friendly representations follow the evolution of the languages they represent. In the future, our approach can inspire adjustments to other human-friendly representations of mapping languages, such as ShExML.

References

1. Cyganiak, R., Wood, D., Lanthaler, M.: RDF 1.1 Concepts and Abstract Syntax. Recommendation, World Wide Web Consortium (W3C) (2014), http://www.w3.org/TR/rdf11-concepts/
2. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF Mapping Language. Working group recommendation, World Wide Web Consortium (W3C) (2012), http://www.w3.org/TR/r2rml/

3. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In: Proceedings of the 7th Workshop on Linked Data on the Web. CEUR Workshop Proceedings, vol. 1184. CEUR-WS.org (2014)
4. García-González, H., Boneva, I., Staworko, S., Labra-Gayo, J.E., Lovelle, J.M.C.: ShExML: improving the usability of heterogeneous data mapping languages for first-time users. PeerJ Computer Science **6** (2020)
5. Heyvaert, P., De Meester, B., Dimou, A., Verborgh, R.: Declarative Rules for Linked Data Generation at your Fingertips! In: The Semantic Web: ESWC 2018 Satellite Events. Lecture Notes in Computer Science, vol. 11155. Springer, Cham (2018)
6. Lefrançois, M., Zimmermann, A., Bakerally, N.: A SPARQL extension for generating RDF from heterogeneous formats. In: The Semantic Web 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 – June 1, 2017, Proceedings. Springer International Publishing (2017)
7. Van Assche, D., Haesendonck, G., De Mulder, G., Delva, T., Heyvaert, P., De Meester, B., Dimou, A.: Leveraging Web of Things W3C Recommendations for Knowledge Graphs Generation. In: Web Engineering. Lecture Notes in Computer Science, vol. 12706. Springer (2021)