Hands-on session

Lift your data with RMLMapper

Franck MICHEL

UNIVERSITÉ CÔTE D'AZUR cnrs Inria

Photo: https://unsplash.com/fr/photos/2-personnes-portant-un-jean-bleu-GDWmu0bFfS4

RMLMapper

Various RML implementations:
https://rml.io/tools/

RMLMapper is the most mature
46 releases since 2018, last 7.3.3 Apr. 2025
22 contributors
https://github.com/RMLio/rmlmapper-java

RMLStreamer for very large sources
19 releases since 2019, last 2.5.0 Jun. 2023
12 contributors
https://github.com/RMLio/RMLStreamer

Issues:

- Limited/not up-to-date documentation
  What is implemented/how to use:
  test cases are the most up-to-date reference
- Poor error messages,
  notably about syntax errors
- Transitioning to “RML2” makes things confusing: namespaces,
  features not/partly implemented

Environment

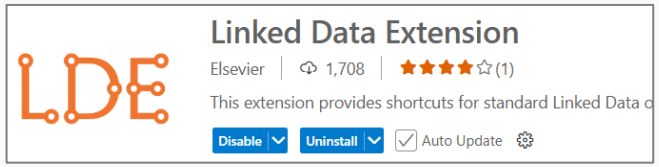
Docker

https://docs.docker.com/engine/install/
https://docs.docker.com/get-started/introduction/get-docker-desktop/

Jar version

https://github.com/RMLio/rmlmapper-java/releases/tag/v7.3.3
https://github.com/RMLio/rmlmapper-java?tab=readme-ov-file#cli

VS Code +



SPARQL endpoint/client: Corese (GUI), GraphDB, YasGUI…

Examples

https://github.com/Wimmics/rml-training
https://rml.io/docs/rml/examples/
https://github.com/RMLio/rmlmapper-java/tree/master/src/test/resources

How to start?

```
# Get the latest version of the RMLMapper
docker pull rmlio/rmlmapper-java

# Check CLI options
docker run  rmlio/rmlmapper-java:latest -h

usage: java -jar mapper.jar <options>
options:
 -c,--configfile <arg>     path to configuration file
 -d,--duplicates           remove duplicates in the HDT, N-Triples, or N-Quads output
 -h,--help                 show help info
 -m,--mappingfile <arg>    one or more mapping file paths and/or
                           strings (multiple values are concatenated).
                           r2rml is converted to rml if needed using the r2rml
                           arguments. RDF Format is determined based on extension.
 -o,--outputfile <arg>     path to output file (default: stdout)
 -s,--serialization <arg>  serialization format (nquads (default),
                           turtle, trig, trix, jsonld, hdt)
 -v,--verbose              show more details in debugging output
```

How to start?

Linux

```
# Run RMLMapper with inline parameters
docker run --rm -v $(pwd):/data \
    rmlio/rmlmapper-java:latest \
    --mappingfile   mapping.ttl \
    --outputfile    output.ttl \
    --serialization turtle
```

Powershell/CMD

```
docker run --rm -v .:/data `
    rmlio/rmlmapper-java:latest `
    --mappingfile   mapping.ttl `
    --outputfile    output.ttl `
    --serialization turtle
```

Your mission

Your mission (1/2)

- Figure out a use case to integrate 2 independent data sources:
  - Select source data: CSV/JSON files (local FS or over http/Web API) or RDB
  - Select and understand target vocabularies:
    - Schema.org, DBpedia, https://lov.linkeddata.es/dataset/lov/ …
- Translate both sources into RDF using RML
  - Define a resource naming strategy: how to construct the resource URIs
  - Write an example of the RDF you would like to generate
  - Define how to join the 2 sources: name? identifier? etc.
  - Pre-process the files if needed (e.g. in python) or use RML functions: remove outliers, fix syntax variations, etc.
  - Write and execute the RML mappings

Examples

- Find accommodations in the city where music festivals take place

  List of festivals per city + list of accommodations

  Possible source: https://www.data.gouv.fr/datasets/search
- Find companies names after Pokemons

  Possible source: https://datasetsearch.research.google.com/

Your mission (2/2)

- Execute a SPARQL query (e.g. in Corese) that involves both generated RDF files
- Start prototyping an LLM-based approach to query the graph
  - **Goal: translate a natural language question into an equivalent SPARQL query**
  - Use an LLM of your choice
    The important is the method, not the fact that the result will be the right one.
  - Propose 3 competency questions that query different aspects of the graph.
  - Discuss the results: What works? What does not work? What could improve?

Work to be submitted

- Files you produced: mappings + CSV/JSON/RDB facelift results
- Snapshot of SPARQL queries execution
- Report:
  - **No ChatGPT fluff!** Don't tell me how Docker works…
  - Useful information: scenario, modeling choices, methodology, difficulties

GenAI coding is handy but stupid. Be smart!