The eNanoMapper Ontology

An application ontology to enable data integration for nanomaterial risk assessment

Javier Millán Acosta

BiGCaT - Maastricht University

September 19, 2022

Overview

1. Introduction
   - eNanoMapper
   - Engineered nanomaterials and safety
   - Ontologies
   - Ontologies - example of a class
2. eNanoMapper Ontology
   - Design of the eNanoMapper ontology
3. Migrating the eNM ontology CI/CD
   - Migrating the ontology development
   - A. Migrating current setup to GitHub actions
   - B. Ontology Development Kit
   - Comparison

Overview

1. Introduction
   - eNanoMapper
   - Engineered nanomaterials and safety
   - Ontologies
   - Ontologies - example of a class
2. eNanoMapper Ontology
3. Migrating the eNM ontology CI/CD

eNanoMapper

- **eNanoMapper** is a broader project which aims to address data and model interoperability challenges for engineered nanomaterial safety. [1]
- **The eNanoMapper ontology** is an application ontology and reuses parts of several ontologies to describe the full domain of nanomaterial safety assessment. [2]. It was continued by NanoCommons and NanoSolveIT.

Figure 1: The eNanoMapper logo

Engineered nanomaterials and safety

- Engineered nanomaterials (eNMs) are broadly defined as compounds that exist on a scale of 1–100 nm. in at least one of their dimensions. [2]
- Their safety assessment must cover the identification of:
  - Physicochemical properties
  - Biological properties and activity [1]
  - Diffusion behaviour into the natural environment [2]

Ontologies

- A formal representation of a set of concepts and the rules constraining how they become structured within a knowledge domain. [3]
- They can be used to provide metadata (a computer-readable set of information that describes other data), or to reason over a domain.
- It consists of three syntactical categories: **Entities**, **Expressions** and **Axioms**, which can be given annotations for further description. [4]

Maastricht University

Ontologies

- All entities (classes, object properties, named individuals...) are uniquely identified by a sequence of characters called IRI (International Resource Identifier, an extension of URI).

Figure 2: An owl:Class in an ontology text file (the ontology document). IRIs in blue.

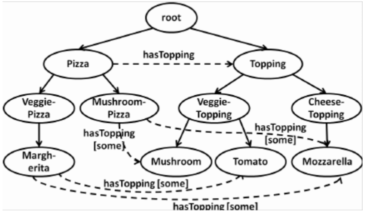
```
<!-- http://purl.obolibrary.org/obo/PATO_0001464 -->

<owl:Class rdf:about="http://purl.obolibrary.org/obo/PATO_0001464">
<rdfs:subClassOf rdf:resource="http://purl.obolibrary.org/obo/PATO_0001018"/><rdfs:subClassOf
rdf:resource="http://purl.obolibrary.org/obo/BFO_0000016"/>
<obo:IAO_0000115 rdf:datatype="http://www.w3.org/2001/XMLSchema#string">A quality that is equal to the
potential energy per unit charge associated with a static (time-invariant) electric field, also called the electrostatic
potential.</obo:IAO_0000115>
<oboInOwl:hasOBONamespace rdf:datatype="http://www.w3.org/2001/XMLSchema#string
">quality</oboInOwl:hasOBONamespace>
<oboInOwl:id rdf:datatype="http://www.w3.org/2001/XMLSchema#string">PATO:0001464</oboInOwl:id>
<oboInOwl:inSubset rdf:resource="http://purl.obolibrary.org/obo/pato#attribute_slim"/><oboInOwl:inSubset
rdf:resource="http://purl.obolibrary.org/obo/pato#scalar_slim"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">electric potential</rdfs:label>
</owl:Class>
```

Ontologies

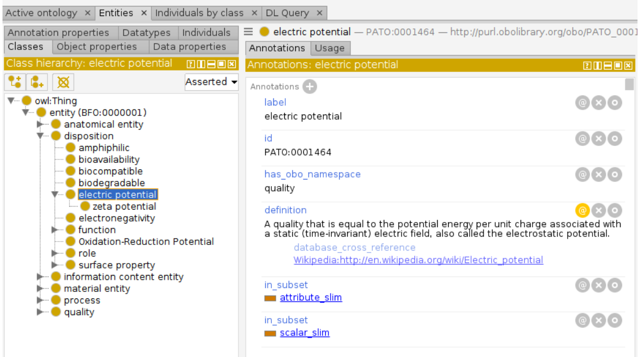
- Most ontologies use the W3C standard language for ontologies, Web Ontology Language **OWL**.
- OWL ontologies are mainly stored in `.owl` files, which are a sort of **RDF** document.
- **RDF** (Resource Description Framework) is a standard for data exchange. It defines **triples** of **(subject, predicate, object)**. [5]
- These triples form labeled graphs where the edge (predicate) represents the link between two resources (subject and object)

Figure 3: The pizza ontology, visualized as a graph [6]



Ontologies - example of a class

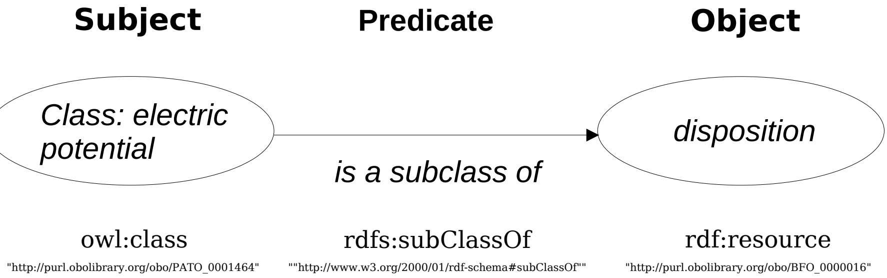
Figure 4: A view of the eNanoMapper ontology class hierarchy, showing the class "electric potential"



Maastricht University

Ontologies - example of a class

Figure 5: A graph with two nodes (Subject and Object) and a triple connecting them (Predicate): the class with class description "electric potential" is a subclass of disposition.



Ontologies - example of a class

Figure 6: The triple in the previous figure as expressed in the `.owl` document file of the ontology it is contained in.

```
<!-- http://purl.obolibrary.org/obo/PATO_0001464 -->

<owl:Class rdf:about="http://purl.obolibrary.org/obo/PATO_0001464">
<rdfs:subClassOf rdf:resource="http://purl.obolibrary.org/obo/BFO_0000016"/>

 (...)
</owl:Class>
```

Maastricht University

Ontologies - example of a class

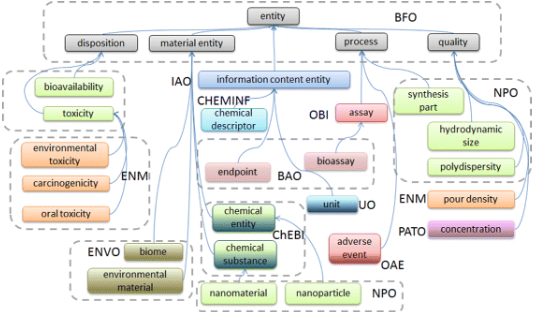
- **Foundation/Upper/Top-level ontologies**
- **Application and domain ontologies**

Overview

1. Introduction
2. eNanoMapper Ontology
   - Design of the eNanoMapper ontology
3. Migrating the eNM ontology CI/CD

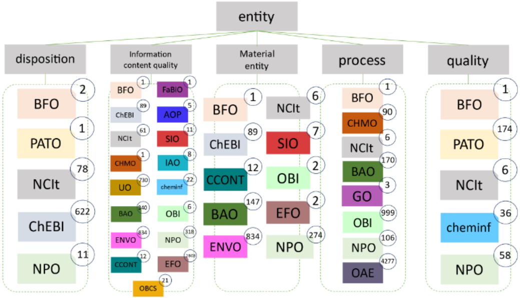
Design of the eNanoMapper ontology

Figure 7: An overview of the upper levels, imports and manually annotated content going into the eNanoMapper ontology [2]



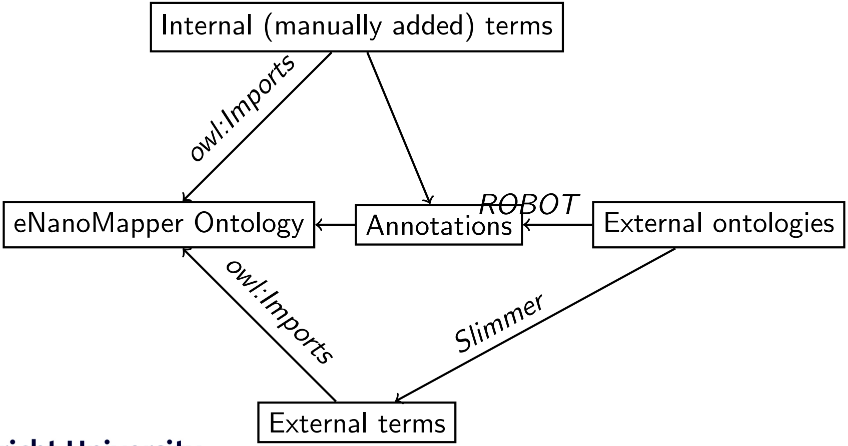
Design of the eNanoMapper ontology

Figure 8: Distribution of the number of classes imported from each ontology into eNM [7]



Design of the eNanoMapper ontology

The current approach uses in-house software (*Slimmer*) to slim imports, and ROBOT to extract annotations.



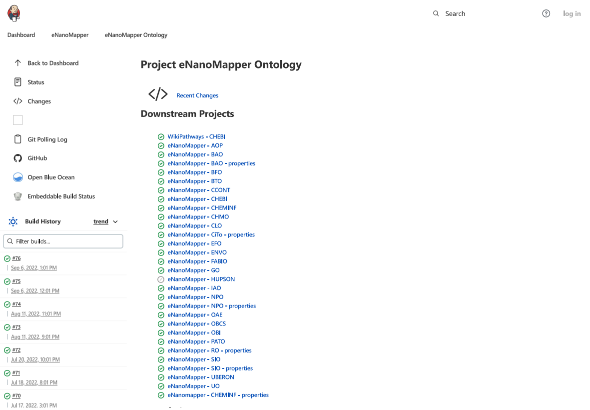
Maastricht University

Design of the eNanoMapper ontology

- **SLIMMER** Uses the OWLAPI to create slims of the ontologies to be imported, and determine the class hierarchy of the imports. [2]
- **ROBOT** is a wrapper for the OWL API used by developers in the Open Biomedical Ontologies (among others) for development and quality control. [8]

Design of the eNanoMapper ontology

Figure 9: The current Jenkins setup for eNanoMapper ontology CI/CD



Overview

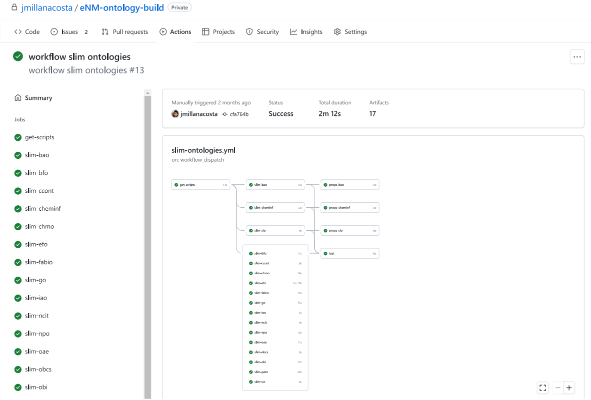
1. Introduction
2. eNanoMapper Ontology
3. Migrating the eNM ontology CI/CD
   - Migrating the ontology development
   - A. Migrating current setup to GitHub actions
   - B. Ontology Development Kit
   - Comparison

Migrating the ontology development

- Current setup: Jenkins for CI in own server (to be dropped soon)
- Alternatives:
  - Literal migration from Jenkins to GitHub actions
  - Migration to the Ontology Development Kit-ROBOT - rethinking the whole process

A. Migrating current setup to GitHub actions

Figure 10: A test run for the GitHub actions-based setup for eNanoMapper ontology CI



A. Migrating current setup to GitHub actions

- A toolkit developed for the development of biomedical ontologies following the software development approach (a method preceded by the eNanoMapper/jenkins/Slimmer setup)
- It is delivered in a docker image and uses ROBOT (a wrapper for the OWL API) and a series of scripts to standardize and automate steps like preparing ontology releases, continuous quality control checking, and dependency management. [9]

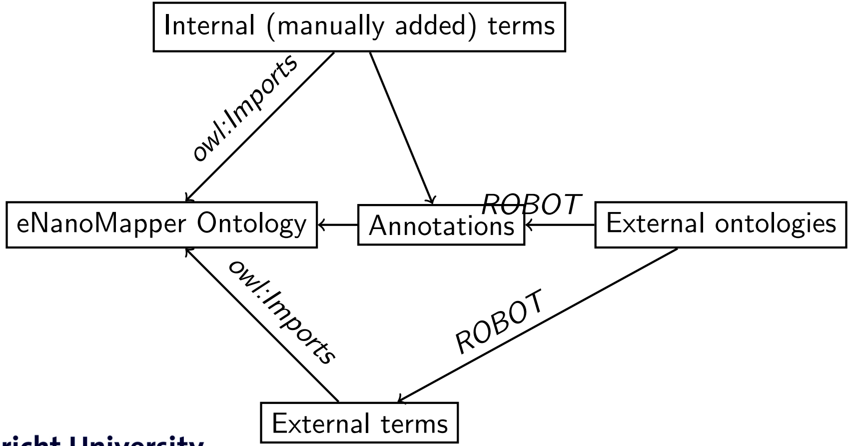
Figure 11: The ODK logo

A. Migrating current setup to GitHub actions

The challenge: replicating the same class hierarchies after substituting Slimmer with ROBOT.

B. Ontology Development Kit

The ODK approach relies on ROBOT to handle ontologies and serialization of several steps through standardized Makefiles. [10]



Comparison

Table 1: Comparison of the two alternatives for eNanoMapper CI/CD

	GitHub actions	Ontology Development Kit
Uses tools	GH actions, slimmer, maybe ROBOT for annotations, in-house scripts	ROBOT and ODK scripts
Keeps current class hierarchy	Yes	Perhaps, but will be a challenge
Sticks to OBO recommendations	Yes, after minimal adaptations if Slimmer is kept	Yes
Hosted on	GitHub	GitHub
Time to develop	Ready at any point	Not optimal for migrating existing ontologies [10]

Maastricht University

References I

[1] Nina Jeliazkova, Charalampos Chomenidis, Philip Doganis, Bengt Fadeel, Roland Grafström, Barry Hardy, Janna Hastings, Markus Hegi, Vedrin Jeliazkov, Nikolay Kochev, Pekka Kohonen, Cristian R Munteanu, Haralambos Sarimveis, Bart Smeets, Pantelis Sopasakis, Georgia Tsiliki, David Vorgrimmler, and Egon Willighagen. The eNanoMapper database for nanomaterial safety information. *Beilstein Journal of Nanotechnology*, 6:1609–1634, July 2015. ISSN 2190-4286. doi: 10.3762/bjnano.6.165. URL `https://www.beilstein-journals.org/bjnano/articles/6/165`.

[2] Janna Hastings, Nina Jeliazkova, Gareth Owen, Georgia Tsiliki, Cristian R Munteanu, Christoph Steinbeck, and Egon Willighagen. eNanoMapper: harnessing ontologies to enable data integration for nanomaterial risk assessment. *Journal of Biomedical Semantics*, 6(1):10, December 2015. ISSN 2041-1480. doi: 10.1186/s13326-015-0005-5. URL `http://www.jbiomedsem.com/content/6/1/10`.

References II

[3] Markus Krötzsch, Frantisek Simancik, and Ian Horrocks. A Description Logic Primer, June 2013. URL http://arxiv.org/abs/1201.4089. arXiv:1201.4089 [cs].

[4] OWL - Semantic Web Standards, . URL https://www.w3.org/OWL/.

[5] RDF - Semantic Web Standards, . URL https://www.w3.org/RDF/.

[6] Nick Drummond and Alan Rector. Pizza Ontology. URL https://protege.stanford.edu/ontologies/pizza/pizza.owl.

[7] Laurent Winckers, Chris Evelo, and Egon Willighagen. Expanding the enanomapper ontology - icbo2020, September 2020. URL https://doi.org/10.5281/zenodo.4032809.

References III

[8] Rebecca C. Jackson, James P. Balhoff, Eric Douglass, Nomi L. Harris, Christopher J. Mungall, and James A. Overton. ROBOT: A Tool for Automating Ontology Workflows. *BMC Bioinformatics*, 20(1):407, December 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-3002-3. URL `https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3002-3`.

[9] Nicolas Matentzoglu, Damien Goutte-Gattat, Shawn Tan, James Balhoff, Seth Carbon, Anita Caron, William Duncan, Joe Flack, Melissa Haendel, Nomi Harris, William Hogan, Charles Hoyt, Rebecca Jackson, HyeongSik Kim, Huseyin Kir, Martin Larralde, Julie McMurry, James Overton, Bjoern Peters, and David Osumi-Sutherland. Ontology development kit: a toolkit for building, maintaining, and standardising biomedical ontologies, 07 2022.

References IV

[10] Nicolas Matentzoglu, Chris Mungall, and Damien Goutte-Gattat. Ontology Development Kit, July 2021. URL https://github.com/INCATools/ontology-development-kit/issues/298.