FAIRmaterials: Ontology Tools with Data FAIRification in Development

08 July 2025

Summary

The bilingual `FAIRmaterials` package simplifies the creation and visualization of materials and data science ontologies. `FAIRmaterials`, available in the Python and R languages, addresses the complexities associated with traditional ontology editors based on manual user input such as `Protege` (Musen 2015) with an intuitive workflow and easy-to-use templates, making it accessible to users both experienced and inexperienced with ontologies.

The `FAIRmaterials` package is its ability to programatically convert simple and structured CSV inputs into rich, well-defined ontologies. This capability is designed to support the findability, accessibility, interoperability, and reusability (FAIR) (Wilkinson et al. 2016) of research data and serve as a tool in the process of data FAIRification.

Its additional features, such as automated ontology merging, static visualizations, and comprehensive documentation for outputs extend its utility, making it a valuable tool for any researcher engaged in knowledge management.

Statement of need

`Protege` (Musen 2015) is currently the most widely-used open-source tool for ontology creation and development. Its main capabilities include manually creating and editing ontological terms and relationships, visualizing ontologies, checking the logical consistency of ontologies, and querying ontologies for specific information. Unfortunately, the complexity of the interface is a barrier for those who have little experience with ontology creation. Another well-known software is `Owlready2` (Lamy 2017), a package for manipulating ontologies in Python. This package allows for manipulation of classes, instances, and annotations, as well as reasoning using a HermiT reasoner (B. Motik et al. 2017). However, like `Protege`, `Owlready2` is complex to use, requiring users to have extensive knowledge about OWL and Description Logics (W3C OWL Working Group 2012). This complexity prevents many researchers from creating and integrating ontologies with their own datasets entirely. Therefore, there is a need for a tool that can create ontologies with an interface that is easily understandable and provides ample documentation on how to use it. `FAIRmaterials` seeks to lower the barrier of entry for scientists entering the world of ontology development and evolution. The package provides a baseline CSV ontology template with built-in and easy-to-follow instructions on how to design an ontology which can be found here. Preliminary applications of the package in additive manufacturing (Hernandez et al. 2024) and PV systems (Nihar et al. 2021) suggest that `FAIRmaterials` offers valuable capabilities for researchers.

Variable Name	Belongs to Ontology	Parent Variable	Definition of Variable	Alternative Name(s)
The variable that you would like to represent in your ontology schema.	If the variable exists in an ontology below, select the ontology from the dropdown menu. Otherwise, please leave blank.	Please type the variable from the Parent Ontology that you would like to connect your variable to. If the variable is already in a selected ontology, please leave blank.	Please provide the definition the variable. For reccomendation on a definition, search the term in https://schema.org. If the variable is already in a selected ontology, please leave blank.	Please provide any alternative names for the variable in the literature. If the variable is already in a selected ontology, please leave blank ***Optional***

Figure 1: Empty Variable CSV Template Sheet for the `FAIRmaterials` Package. The CSV template sheet includes specific instructions on how to fill out every row to correctly generate the ontology. Template is split in half for readability.

Key Features

Ontology creation from template CSVs

The primary function of the `FAIRMaterials` package is to convert the term, relationship, and value specifications from the CSV template into an ontology. An overview of the sheets descriptive headers is illustrated in Figure 1.

Ontology serialization into multiple syntaxes

The package automatically converts the CSV sheets into an RDF object using `RDFlib` (Carl Boettiger 2024; RDFLib Development Team 2025) and then serializes the object into two syntaxes: Turtle and JSON-LD. The ontology is serialized into two syntaxes because of the unique advantages that each syntax provides.

Static visualization output of ontology

Determining the correctness of an ontology is difficult if its representation is in a textual format. For this reason, the package outputs a visualization in both the R and Python versions. The optional Python flag `include_graph_valuetype` can be used to include value type nodes in the output visualization. The visualization is generated using the `Graphviz` (Graphviz Development Team 2025) software in the Python version and `DiagrammeR` (Iannone and Roy 2025) in R. Both outputs are modeled after the popular `WebVOWL` (Lohmann et al. 2015) ontology visualization tool to make it easier for users to inherently understand the color schema and format.

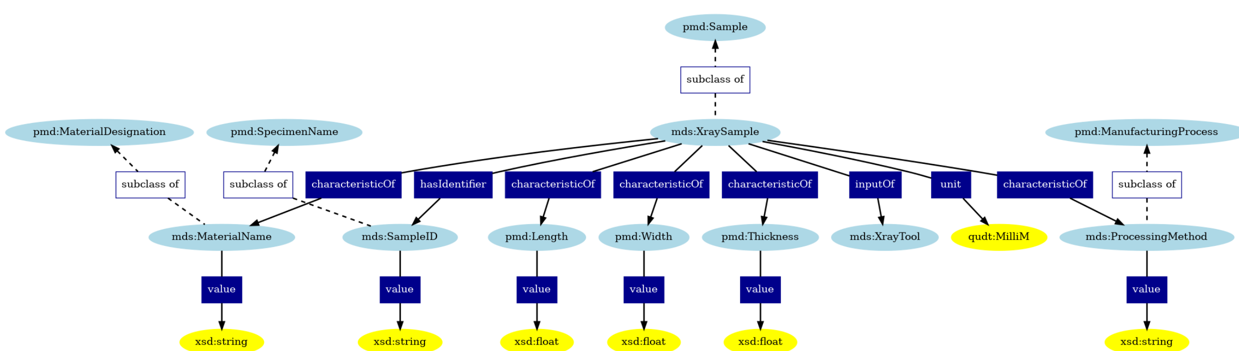

Figure 2: The X-ray sample ontology. The light-blue icons represent ontology terms, with the prefix (i.e. pmd) indicating the ontology that the term was created in. The dark-blue squared boxes indicate relationships created between entities. The yellow round boxes either indicate the type of the value stored in each subclass or the unit that the value is expressed in, with the prefix indicating the ontology the unit definition belongs to or the schema language that the value type is defined in.

Ontology merging

Both the R and Python versions of the `FAIRmaterials` package feature an ontology merging capability. The package processes all CSV files within a specified folder and its subdirectories, merging them into one ontology created in the main folder path. For each subdirectory containing a complete set of CSV sheets, the package generates separate, unmerged outputs. The merged output can also include customized metadata such as title, authors, version, URI, and description.

Corresponding documentation output for ontology

One important aspect of ontologies is that they are easily readable by humans as well as machines. The HTML documentation provides an intuitive interface for humans to understand the terms and relationships stored in ontologies. The Python version of the package leverages this by using `RDFLib` to output a PyLode HTML file. Unfortunately, the R version does not have the same capability because the R version of the RDFlib package does not create HTML files.

Typical Usage

It is recommended that users first design an ontology schema that includes all the vocabulary needed to describe a dataset. This ensures explicit connections to the Basic Formal Ontology (BFO) or another top-level ontology, ensuring its interoperability with other existing ontologies. Every variable in the ontology schema should be tagged as a subclass of an already-existing ontology term or it should be a new term. Other top-level terms should be used within the schema when necessary, such as using a QUDT ontology term when associating a certain measurement term with a standardized unit. An example of an ontology schema is showed in Figure 3.

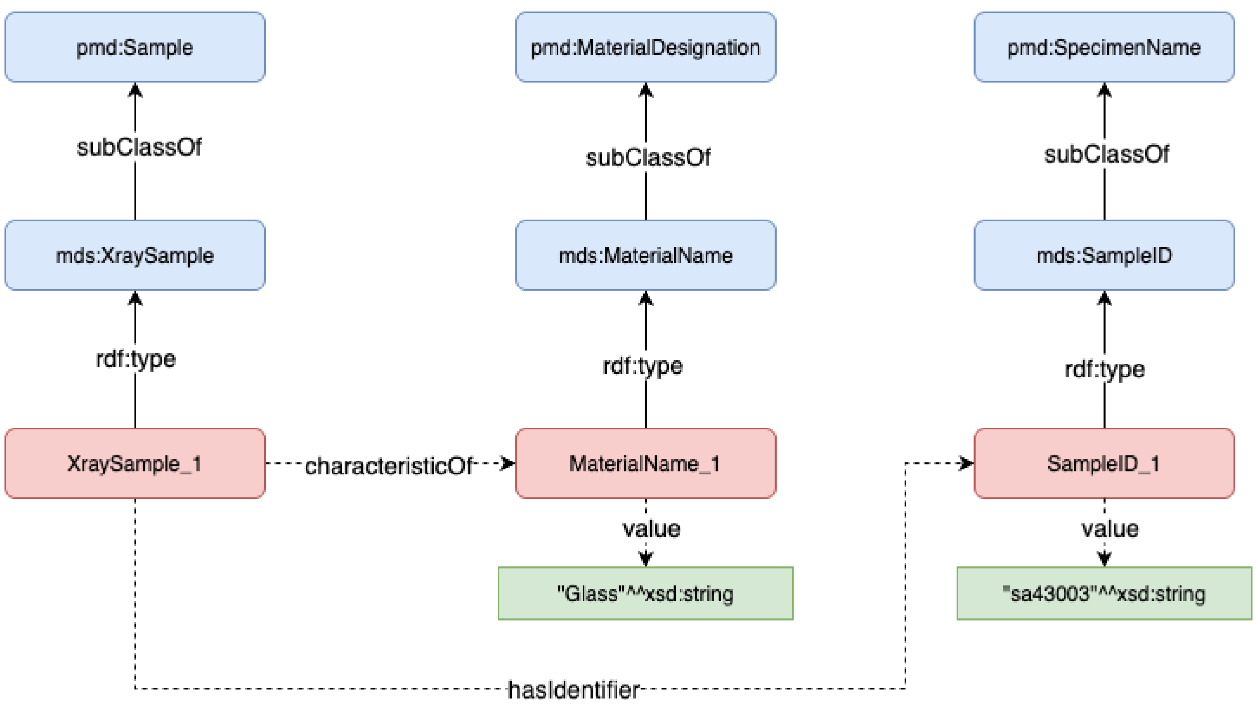

Figure 3: Example of Schema.

Post-execution, users should review the output to ensure accuracy and make necessary adjustments. This streamlined workflow facilitates effective ontology development without requiring extensive technical expertise.

Code Availability

To install Python version of `FAIRmaterials`, simply search for it on the The Python Package Index (PyPI) (Python Software Foundation 2025) website or click here. The `FAIRmaterials` R version can be easily accessed on the Comprehensive R Archive Network (CRAN) (R Project 2025). To install the package, simply search for `FAIRmaterials` on the CRAN website or click here. The code for both versions can also be accessed through a public GitHub found here and more documentation for the packages can be found here.

Acknowledgements

The development and research of the `FAIRmaterials` package was made possible through generous support from multiple sources. This work was supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office (SETO) through Agreement Numbers DE-EE0009353 and DE-EE0009347. Additional support was provided by the Department of Energy (National Nuclear Security Administration) under Award Number DE-NA0004104 and Contract Number B647887 and from the U.S. National Science Foundation under Award Number 2133576.

The authors would like to sincerely thank these organizations for their financial assistance as well as all of the individuals who participated in the project.

Appendix

Example full set of completed ontology sheets for the mds-XrayToolChess ontology

A	B	C
Prefix Name	**Ontology URL**	**Ontology File**
Create a prefix for the ontology that you would like to import into your own ontology	Enter the URL of the ontology's OWL file	Enter the URI of the location of the ontology OWL online
pmd	https://w3id.org/pmd/co/	
qudt	http://qudt.org/2.1/vocab/unit#	

Figure 4: Namespace Sheet: This sheet is used to define the namespace which connects ontology prefixes to the ontology URL. This sheet aids in preventing conflicts and maintaining clarity across the ontology's vocabulary.

A	B
Ontology Name *Name of the ontology*	mds-XraySample
Ontology URI *Base URI of the ontology (the ontology URI, minus the name of the ontology)*	https://cwrusdle.bitbucket.io/xraySample#
Ontology Version *Version of the ontology*	0.2
Ontology Author(s) *Authors of the ontoloy (seperate multiple authors by a comma)*	Alexander C. Harding Bradley, Balashanmuga Priyan Rajamohan, Mohommad Redad Mehdi, Weiqi Yue, Finley Holt, Pawan K. Tripathi, Erika I. Barcelos, Matthew Willard, Frank Ernst, Roger H. French
Ontology Description *Description of the ontology and ontology domain*	XRD Sample Ontology for the FAST Beamline at CHESS.

Figure 5: Ontology Info Sheet: Contains essential metadata about the ontology including title, creator, and version. This sheet sets the foundational attributes that describe and contextualize the ontology.

A	B	C	D	E	F	G
ValueType Name	**Belongs to Ontology**	**Domain**	**Range**	**Definition of Property**	**Logical Axioms**	**Alternative Name(s)**
The name of the valuetype of a variable that you would like to include in your ontology schema.	If the valuetype already exists in an ontology below, select the ontology from the dropdown menu. Otherwise, please leave blank.	Please enter the term that the relationship starts from. (Example: If you would like to define the relationship that a tool term "outputs" an image term, select the tool term).	Please enter the value type you would like to attach to the term ***Only fill out for Data Property relationships***	Please provide the definition the relationship. For reccomendation on a definition, search the term in https://schema.org. If the term is already in a selected ontology, please leave blank.	Please provide any logic you would like to attach to your term. For information and examples on logical axioms, please visit https://www.w3.org/TR/daml+oil-axioms/ ***Optional***	Please provide any alternative names for the relationship in the literature. ***Optional***
value	pmd	SampleID	xsd:string			
value	pmd	ProcessingMethod	xsd:string			
value	pmd	MaterialName	xsd:string			
value	pmd	**pmd:Length**	**xsd:float**			
value	pmd	**pmd:Width**	**xsd:float**			
value	pmd	**pmd:Thickness**	**xsd:float**			

Figure 6: Value Type Sheet: Specifies the types of values associated with ontology terms, used for data consistency and semantic accuracy in ontology modeling.

A	B	C	D	E	F	G
Relationship Name	**Belongs to Ontology**	**Domain**	**Range**	**Definition**	**Logical Axioms**	**Alternative Name(s)**
The name of the relationship between two terms that you would like to use in your ontology schema.	If the relationship already exists in an ontology below, select the ontology from the dropdown menu. Otherwise, please leave blank.	Please enter the term that the relationship starts from. (Example: If you would like to define the relationship that a tool term "outputs" an image term, select the tool term).	Please enter the term that the relationship goes to. Example: If you would like to define the relationship that a tool term "outputs" an image, select the image variable.	Please provide the definition the relationship. For reccomendation on a definition, search the term in https://schema.org. If the term is already in a selected ontology, please leave blank.	Please provide any logic you would like to attach to your term. For information and examples on logical axioms, please visit https://www.w3.org/TR/daml+oil-axioms/ ***Optional***	Please provide any alternative names for the relationship in the literature. ***Optional***
hasIdentifier	**pmd**	**XraySample**	**SampleID**			
characteristicOf	**pmd**	**XraySample**	**ProcessingMethod**			
characteristicOf	**pmd**	**XraySample**	**MaterialName**			
characteristicOf	**pmd**	**XraySample**	**pmd:Length**			
characteristicOf	**pmd**	**XraySample**	**pmd:Width**			
characteristicOf	**pmd**	**XraySample**	**pmd:Thickness**			
inputOf	**pmd**	**XraySample**	**XrayTool**			

Figure 7: Relationship Definition Sheet: Outlines the various relationships between terms within the ontology, facilitating a structured approach to defining how ontology elements interconnect.

A	B	C	D	E	F	G
Variable Name	**Belongs to Ontology**	**Parent Variable**	**Definition of Variable**	**Alternative Name(s)**	**Unit**	**Logical Axioms**
The variable that you would like to represent in your ontology schema.	If the variable exists in an ontology below, select the ontology from the dropdown menu. Otherwise, please leave blank	Please type the variable from the Parent Ontology that you would like to connect your variable to. If the variable is already in a selected ontology, please leave blank	Please provide the definition the variable. For reccomendation on a definition, search the term in https://schema.org. If the variable is already in a selected ontology, please leave blank.	Please provide any alternative names for the variable in the literature. If the variable is already in a selected ontology, please leave blank ***Optional***	Please provide the unit that the variable is expressed in your data. For a dictionary of standardized units, please visit https://qudt.org. If the variable is already in a selected ontology, please leave blank ***Optional***	Please provide any logic you would like to attach to your variable. For information and examples on logical axioms, please visit https://www.w3.org/TR/daml+oil-axioms/ If the variable is already in a selected ontology, please leave blank
Sample	pmd					
SpecimenName	**pmd**					
SampleID		**pmd:SpecimenName**	**A human-labeled sample identifier.**			
ManufacturingProcess	**pmd**					
ProcessingMethod		pmd:ManufacturingP…	The manufacturing method by ehich the sample was created.			
MaterialDesignation	**pmd**					
MaterialName		pmd:MaterialDesigna…	Name of the material.			
Length	pmd		**Length of the sample.**		qudt:MilliM	
Width	pmd		**Width of the sample.**		qudt:MilliM	
Thickness	**pmd**		Thickness of the sample.		qudt:MilliM	
XraySample		**pmd:Sample**				
XrayTool						

Figure 8: Variable Definition Sheet: This sheet details the individual variables within the ontology, defining their attributes and how they relate to the ontology's broader structure.

References

B. Motik, I. Horrocks et al. 2017. "HermiT OWL Reasoner." http://hermit-reasoner.com.

Carl Boettiger, Anna Krystalli, Bryce Mecum. 2024. "rdflib: Tools to Manipulate and Query Semantic Data." https://cran.r-project.org/web/packages/rdflib/index.html. https://doi.org/10.32614/CRAN.package.rdflib.

Graphviz Development Team. 2025. "Graphviz Python Package." https://pypi.org/project/graphviz/.

Hernandez, K. J., E. I. Hernandez, J. C. Jimenez, et al. 2024. "A Data Integration Framework of Additive Manufacturing Based on FAIR Principles." *MRS Advances* 9: 844–51. https://doi.org/10.1557/s43580-024-00874-5.

Iannone, Rich, and Olivier Roy. 2025. "DiagrammeR: Graph/Network Visualization." https://cran.r-project.org/web/packages/DiagrammeR/index.html. https://doi.org/10.32614/CRAN.package.DiagrammeR.

Lamy, J. B. 2017. "Owlready: Ontology-Oriented Programming in Python with Automatic Classification and High Level Constructs for Biomedical Ontologies." *Artificial Intelligence in Medicine* 80: 11–28.

Lohmann, Steffen, Vincent Link, Eduard Marbach, and Stefan Negru. 2015. "WebVOWL: Web-Based Visualization of Ontologies." In *Knowledge Engineering and Knowledge Management*, edited by Patrick Lambrix, Eero Hyvönen, Eva Blomqvist, Valentina Presutti, Guilin Qi, Uli Sattler, Ying Ding, and Chiara Ghidini, 154–58. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-17966-7_21.

Musen, Mark A. 2015. "The Protégé Project: A Look Back and a Look Forward." *AI Matters* 1 (4): 4–12. https://doi.org/10.1145/2757001.2757003.

Nihar, A. et al. 2021. "Toward Findable, Accessible, Interoperable and Reusable (FAIR) Photovoltaic System Time Series Data." In *2021 IEEE 48th Photovoltaic Specialists Conference (PVSC)*, 1701–6. Fort Lauderdale, FL, USA. https://doi.org/10.1109/PVSC43889.2021.9518782.

Python Software Foundation. 2025. "PyPI: The Python Package Index." https://pypi.org.

R Project. 2025. "CRAN: The Comprehensive R Archive Network." https://cran.r-project.org.

RDFLib Development Team. 2025. "RDFLib: Python Library for working with RDF." https://pypi.org/project/rdflib/.

W3C OWL Working Group. 2012. "OWL 2 Web Ontology Language Document Overview (Second Edition)." https://www.w3.org/TR/owl2-overview/.

Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data* 3 (1): 160018. https://doi.org/10.1038/sdata.2016.18.