How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices

Platform MaterialDigital- Semantic Interoperability Team

PMDco workshop repository

https://github.com/HosseinBeygiNasrabadi/PMDco-workshop

Platform MaterialDigital Core Ontology (PMDco) Workshop

This repository provides PMDco workshop materials, featuring the HTTTO (High Temperature Tensile Test Ontology) as a hands-on use case. HTTTO is an application-level ontology built on top of the Platform MaterialDigital Core Ontology (PMDco) and is designed to support workshops on **How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices**

Workshop materials

- **Workshop slides** here
- **Tutorials 1 and 2:** Miro Board
- **Tutorial 3:** GitHub ODK template
- **Tutorial 4:** GitHub PMDco Workshop (this repository) and httto.ttl

PMDco documentation

- **PMDco GitHub repository**
- **PMDco documentation page** for detailed information on the ontology structure, beginner learning materials, and a complete user guide.
- **PMDco list of classes and properties in Widoco**
- **PMDco ontology design patterns**
- **PMDco-related publications**

How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices

Workshop content (10 topics, 4 tutorials):

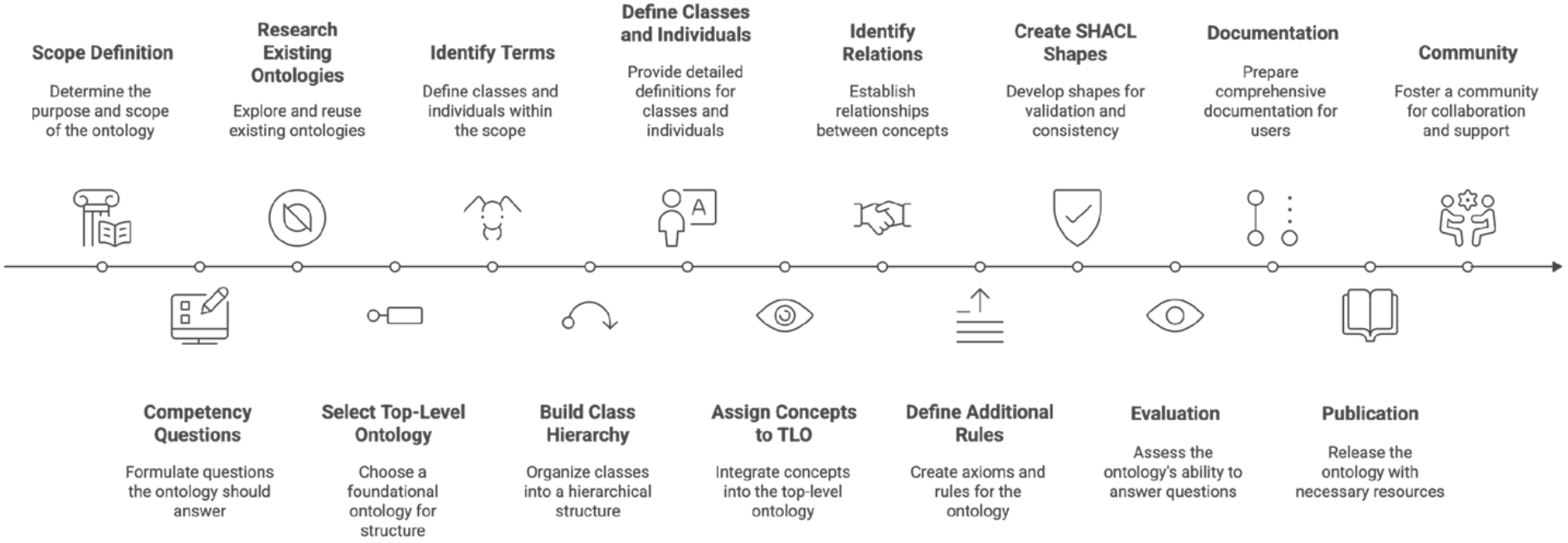
1- Ontology development and beginners learning materials
2- Ontology levels
3- Basic Formal Ontology (BFO) classes
4- Platform MaterialDigital Core Ontology (PMDco) classes
Tutorial 1: Structure given classes according to PMDco hierarchy (Miro board)
5- Platform MaterialDigital Core Ontology (PMDco) object properties
Tutorial 2: Using appropriate object properties (Miro board)
6- How to develop your application ontologies using PMDco and OBO+ODK best practices?
7- Ontology Development Kit (ODK)
Tutorial 3: Creating ODK repository for PMDco application ontologies (GitHub)
8- Collaborative ontology development workflow, adding taxonomy and axioms
9- PMDco Ontology Design Patterns (ODPs)
Tutorial 4: Ontology editing; adding classes, annotations and axioms (Protege)
10- Ontology evaluation, release, documentation and maintenance

1- Ontology development and beginners learning materials

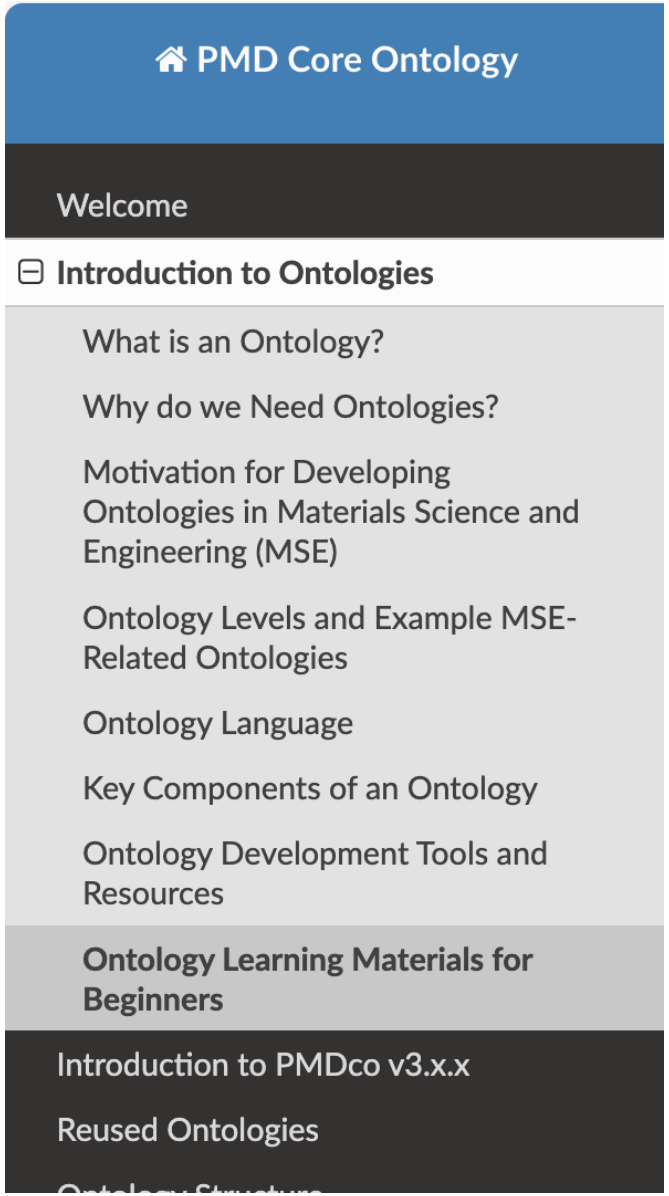
An ontology is an explicit, formal specification of a shared conceptualization.

conceptualization:	abstract model (domain, relevant terms, relations)
explicit:	meaning of all terms is clearly and unambiguously defined
formal:	interpretable by machines
shared:	consensus

Ontology development process:



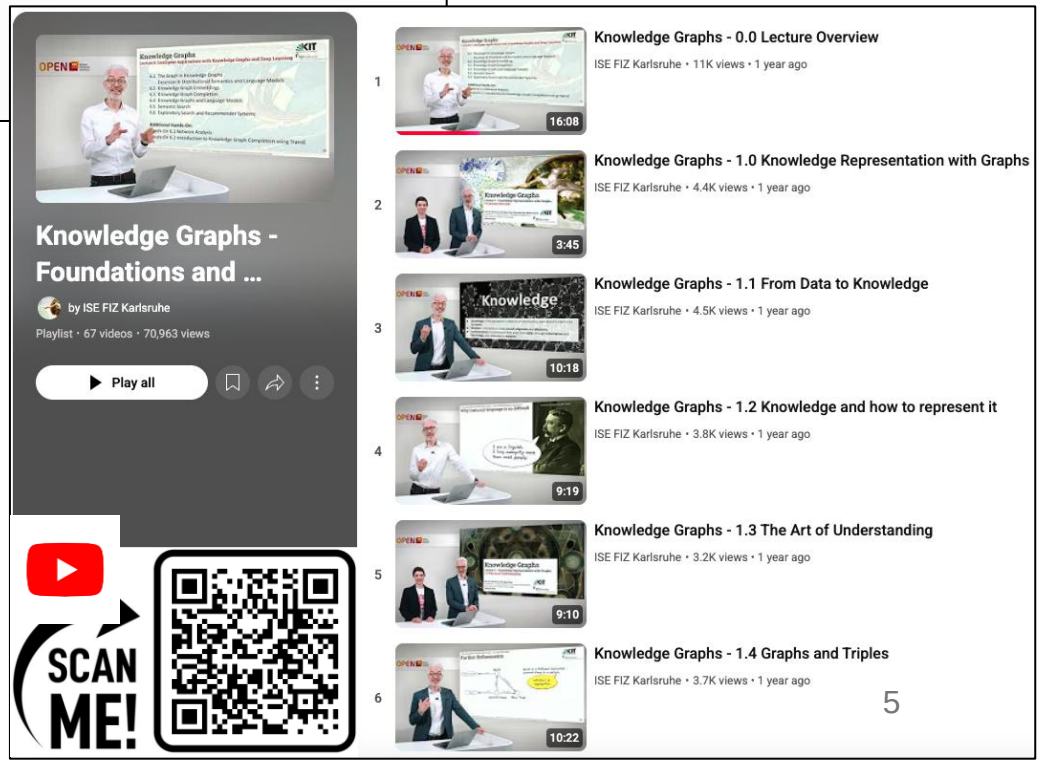
1- Ontology development and beginners learning materials



Ontology Learning Materials for Beginners

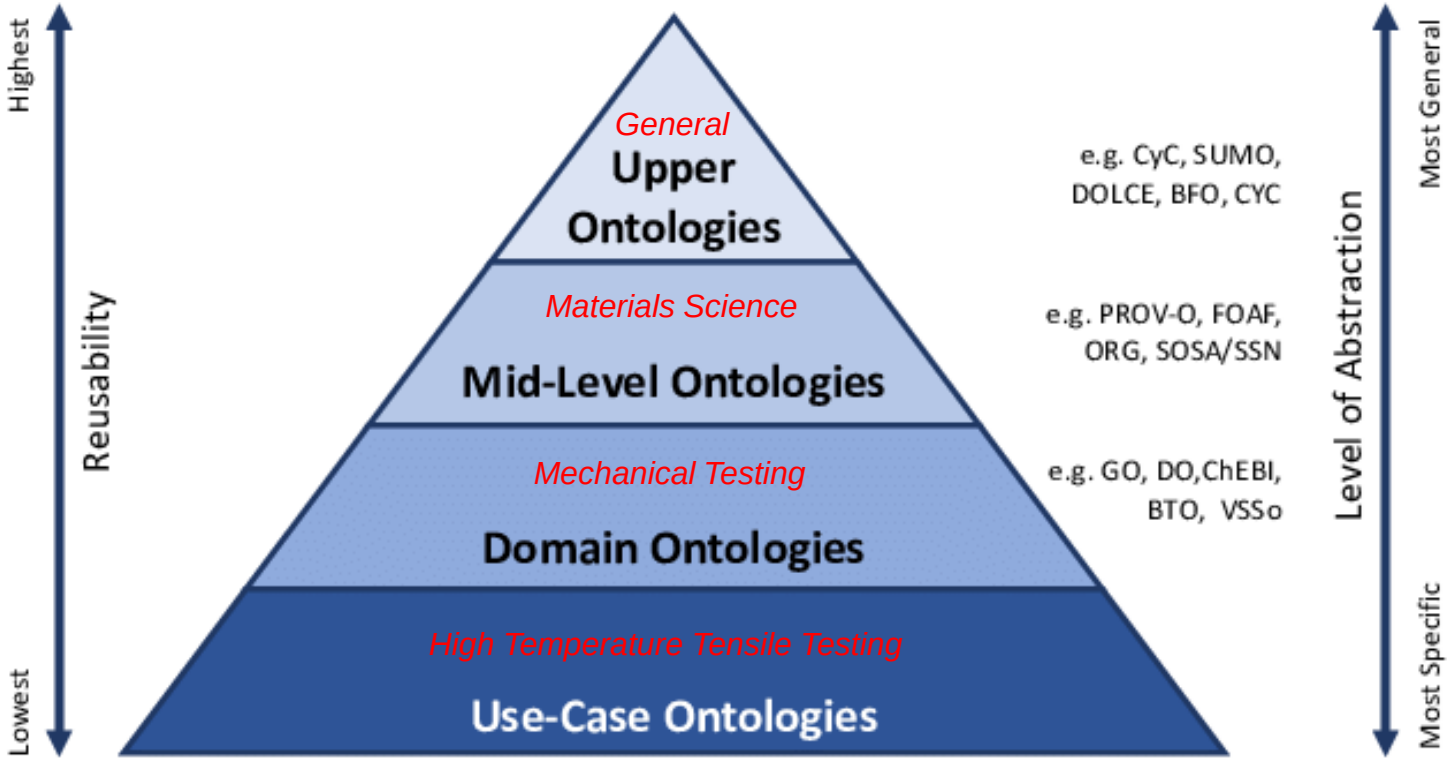
- A Practical Ontology Development Guide: In cooperation with participants in the MaterialDigital initiative, a guide for the development of ontologies in general was created, which contains basic aspects and recommended procedures. This guide may be expanded and developed further and is hosted at The Scientific Ontology Network.
- Ontology Development 101: Stanford University's guide provides a foundational, step-by-step methodology for creating your first ontology.
- Pizza Tutorial: A practical guide to building OWL ontologies using Protégé 5.5 and plugins
- OBO Academy's Ontology Design Course: An interactive, free curriculum that provides lessons on ontology construction, reasoning, and design principles (based on the OBO Foundry and Semantic Web standards)
- ISE FIZ YouTube channel: Playlist of Lectures "Knowledge Graphs - Foundations and Applications"
- Barry Smith's YouTube channel
- PMD YouTube channel

PMDco documentation page



2- Ontology levels

- **Top-Level Ontologies**: Define very general concepts that apply across all domains
- **Mid-Level Ontologies**: Provide reusable structures shared across several domains
- **Domain-Level Ontologies**: Describe concepts specific to a particular field
- **Application-Level Ontologies**: Highly specific to individual applications or projects



3- Basic Formal Ontology (BFO)

- ISO standard top-level ontology
- A small top-level ontology that is designed for use in supporting information retrieval, analysis and integration in scientific and other domains.
- Used by more than 800 ontology-driven endeavors throughout the world.


INTERNATIONAL STANDARD

ISO/IEC 21838-2

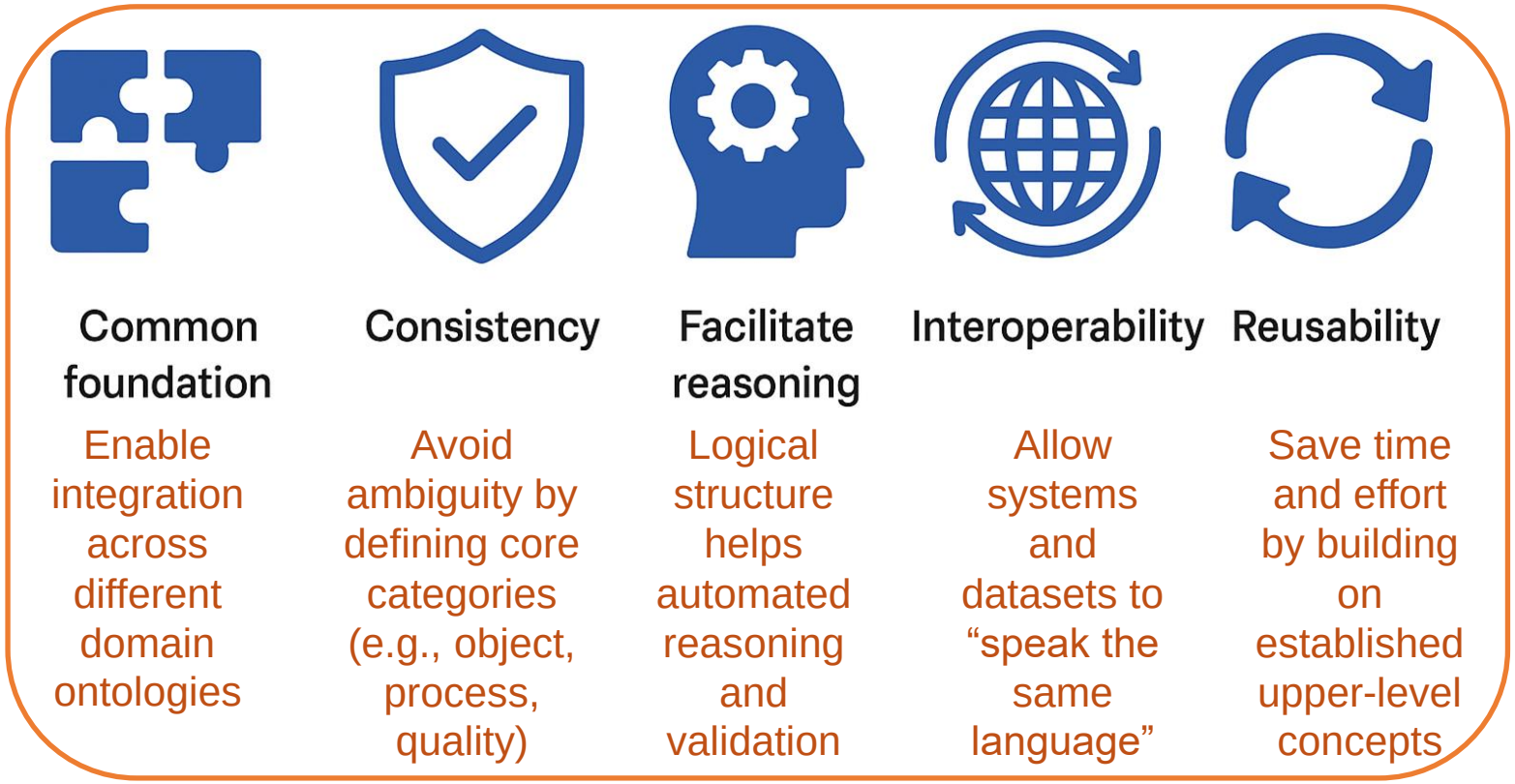
First edition 2021-11

ISO

Information technology — Top-level ontologies (TLO) —

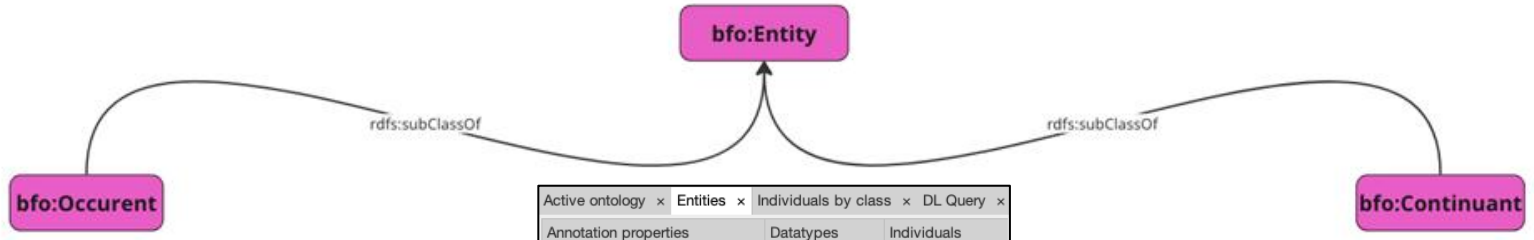
Part 2:
Basic Formal Ontology (BFO)

Technologies de l'information — Ontologies de haut-niveau (TLO) —
Partie 2: Ontologie formelle de base (BFO)



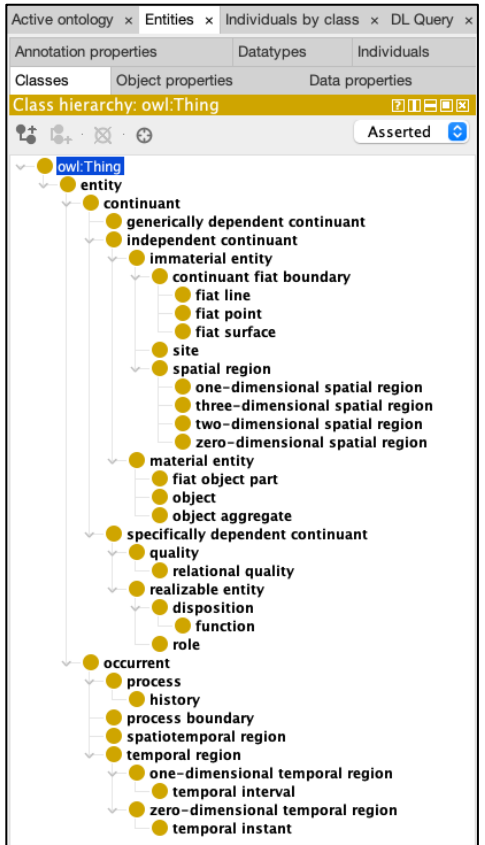
3- Basic Formal Ontology (BFO) classes: the very top

BFO sits at the very top as a domain-neutral upper ontology
Divides all entities into two fundamental categories:



Occurrents

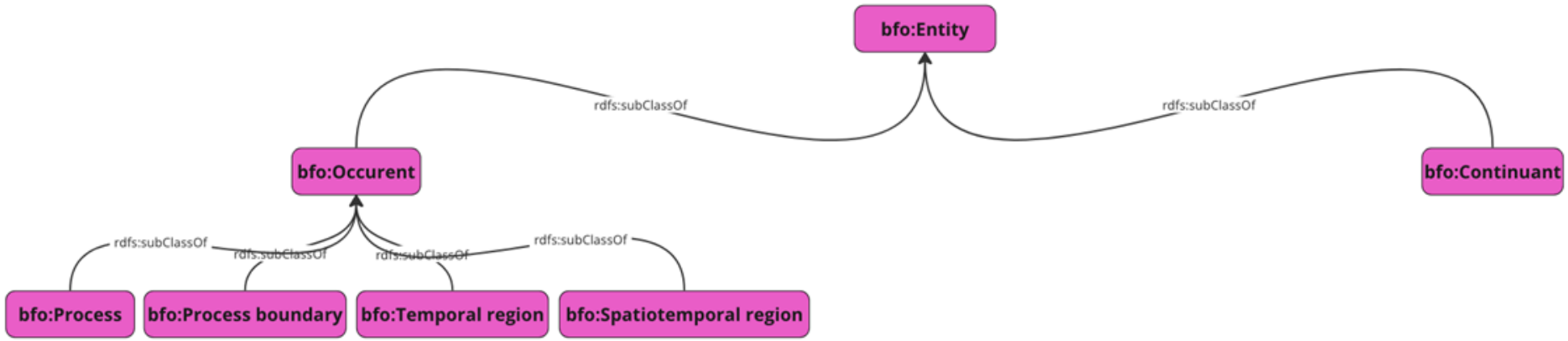
- Happen or unfold **over time**
- Have temporal parts and duration
- Examples: reading a book

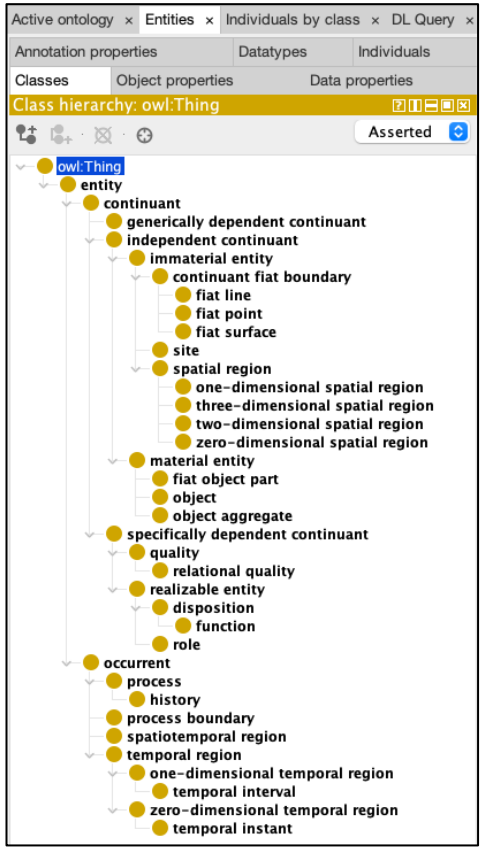


Continuants

- Persist through time while possibly undergoing change
- Exist **wholly at any moment** in time
- Example: a book

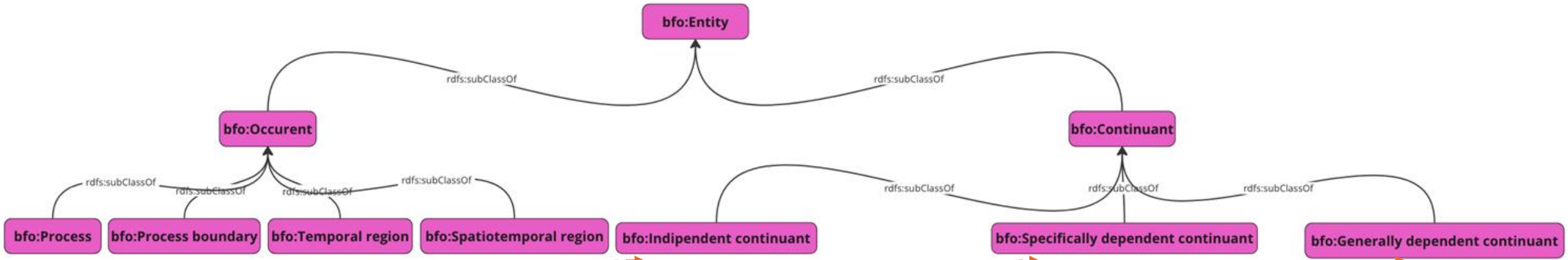
3- Basic Formal Ontology (BFO) classes: Occurrent





- **Process:** Ongoing event with temporal parts (e.g., a surgery, running)
- **Process Boundary:** Instant where a process starts or ends (e.g., moment of impact)
- **Temporal Region:** Span of time itself (e.g., a day, a year)
- **Spatiotemporal Region:** Combination of space and time (e.g., the area of a concert during its performance)

3- Basic Formal Ontology (BFO) classes: Continuant



Independent Continuants

- Exist on their own
- Do not depend on other entities for their existence
- Examples: a rock, a human body, a tree

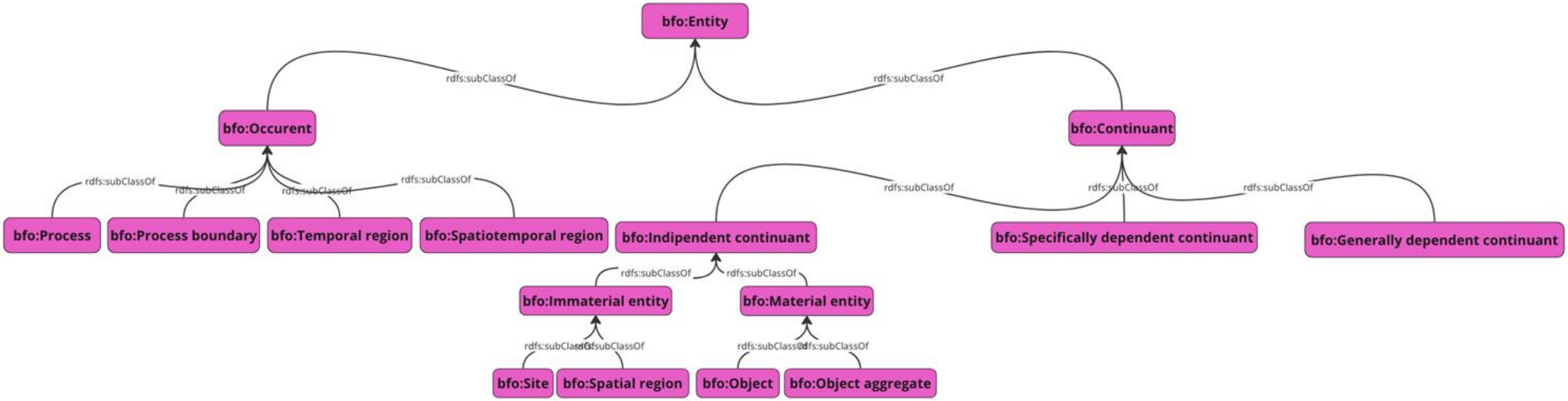
Specifically Dependent Continuants

- Depend on **one specific** bearer
- If the bearer ceases to exist, then its quality, function, role ceases to exist
- Examples: the color of *this* apple, my weight

Generically Dependent Continuants

- Can be **copied or instantiated** in different bearers
- Examples: a digital file, a PDF document, a gene sequence

3- Basic Formal Ontology (BFO) classes: Independent continuant



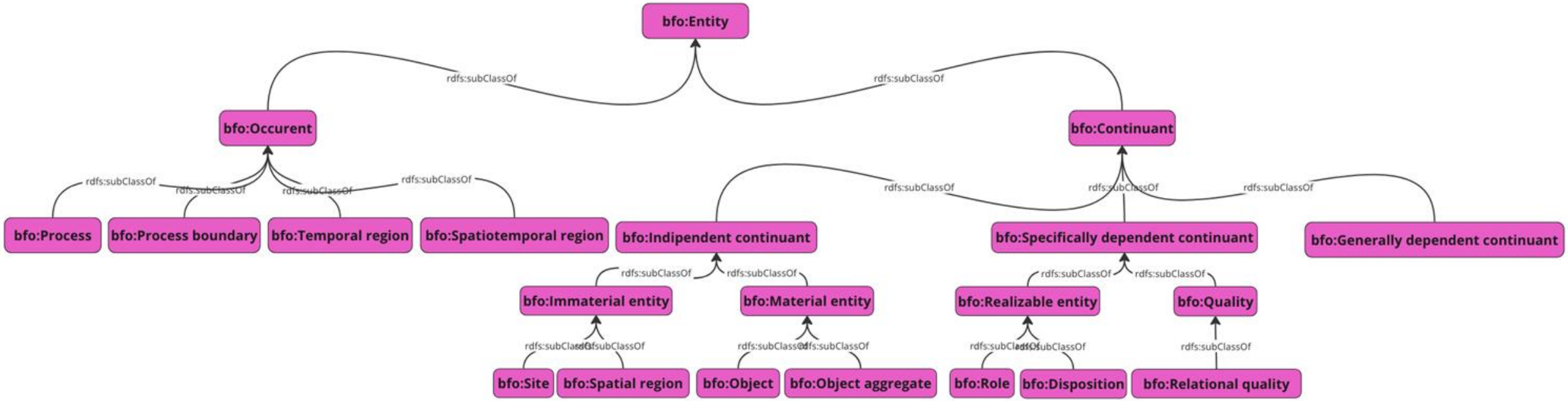
Immaterial Entity: Has no physical mass but still exists spatially

- **Site**: location occupied by something (e.g., parking space)
- **Spatial Region**: abstract space (e.g., 3D coordinates)
- **Fiat Boundary**: non-physical border (e.g., equator)

Material Entity: Has physical presence and occupies space

- **Object**: one whole, independent thing (e.g., a car)
- **Object Aggregate**: group of objects (e.g., a fleet of cars)
- **Fiat Object Part**: arbitrary part of an object (e.g., front half of a car)

3- Basic Formal Ontology (BFO) classes: Specifically Dependent Continuant



Realizable Entity: Exists as a potential; realized in certain conditions

- **Role:** A realizable entity that depends on external **social or** institutional context

Example: A person as a teacher, a molecule as a tracer

- **Disposition:** A realizable entity that reflects an inherent tendency

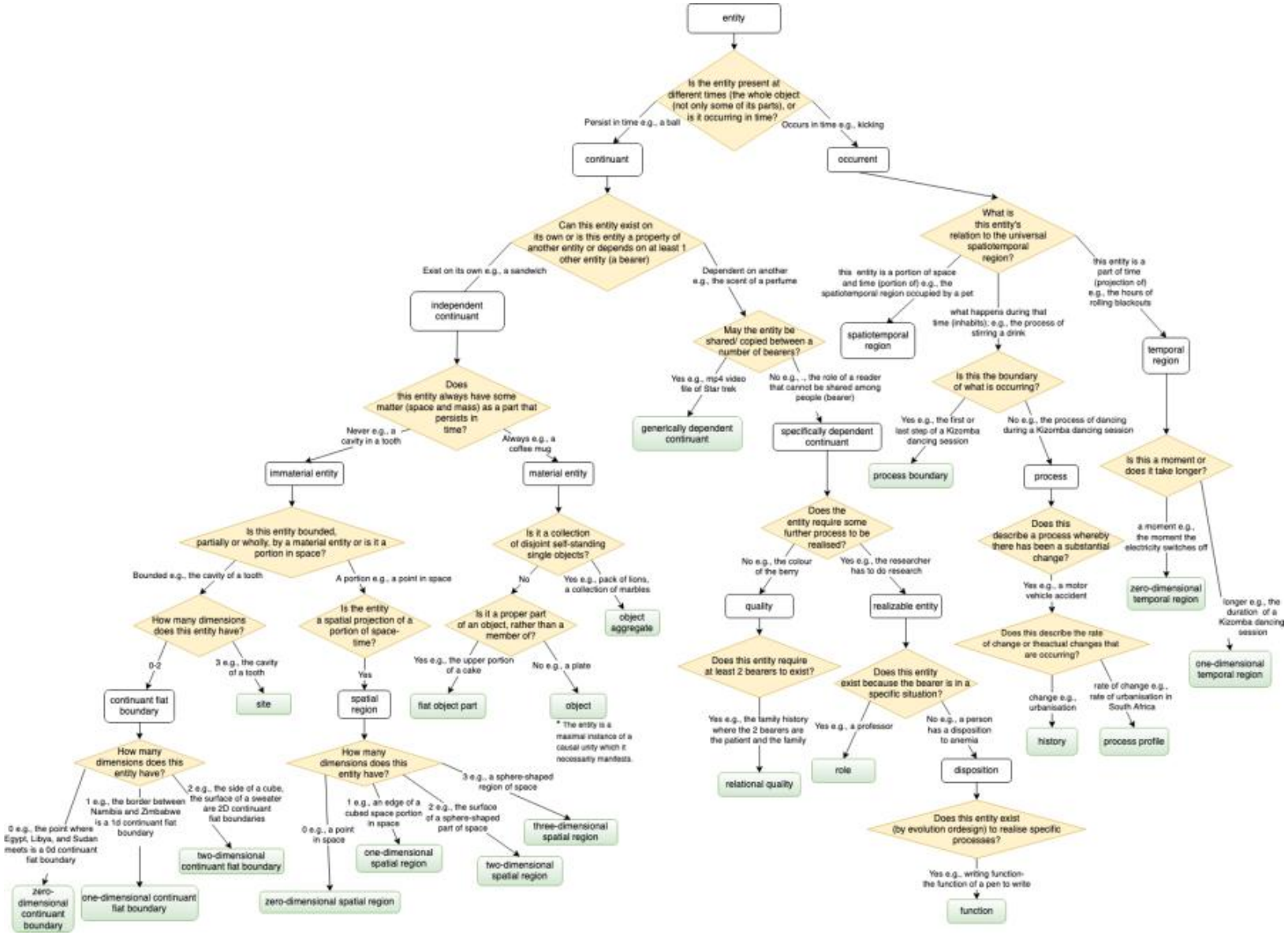
Example: Fragility of glass, flammability of gasoline

Quality: Inherent and always present in the bearer, Measurable or observable features

Examples: color of an apple, mass of a rock

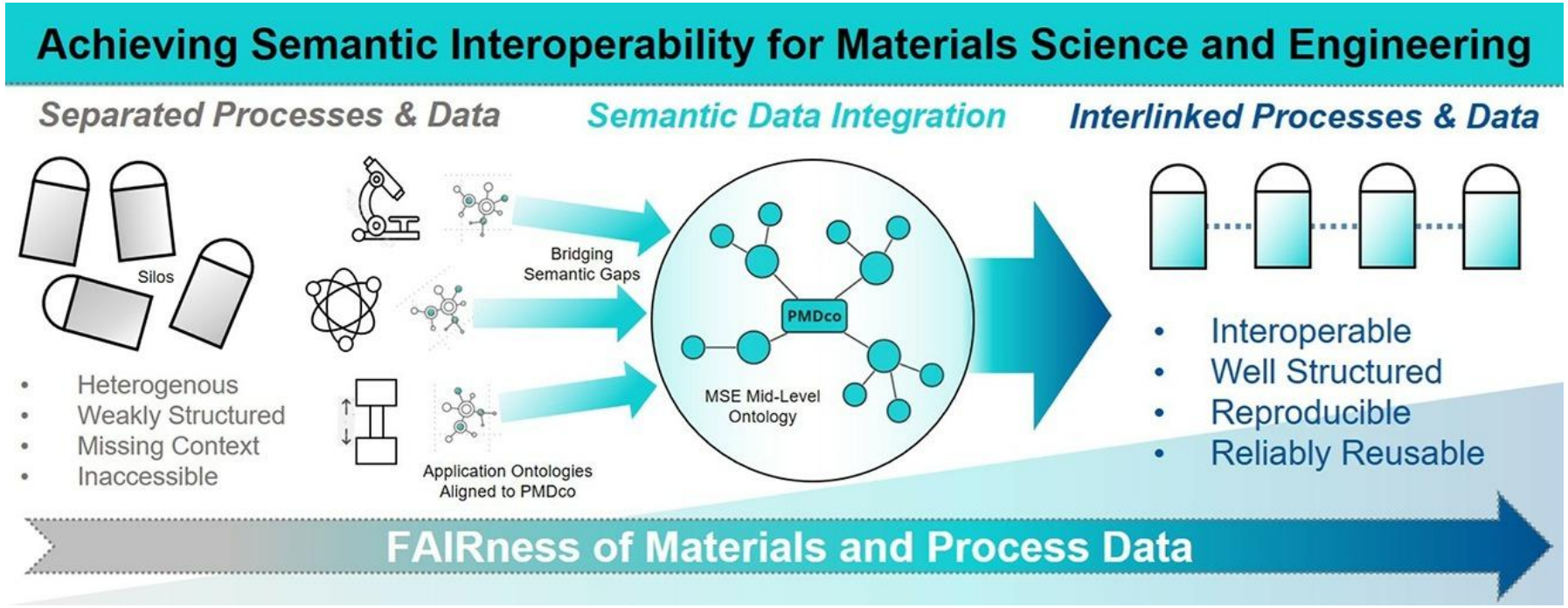
- **Relational Quality**: a quality that depends on another entity (e.g., distance between two objects)

BFO Classifier



4- Platform MaterialDigital Core Ontology (PMDco) classes

A mid-level ontology to support the digital transformation of the Materials Science and Engineering (MSE) domain.



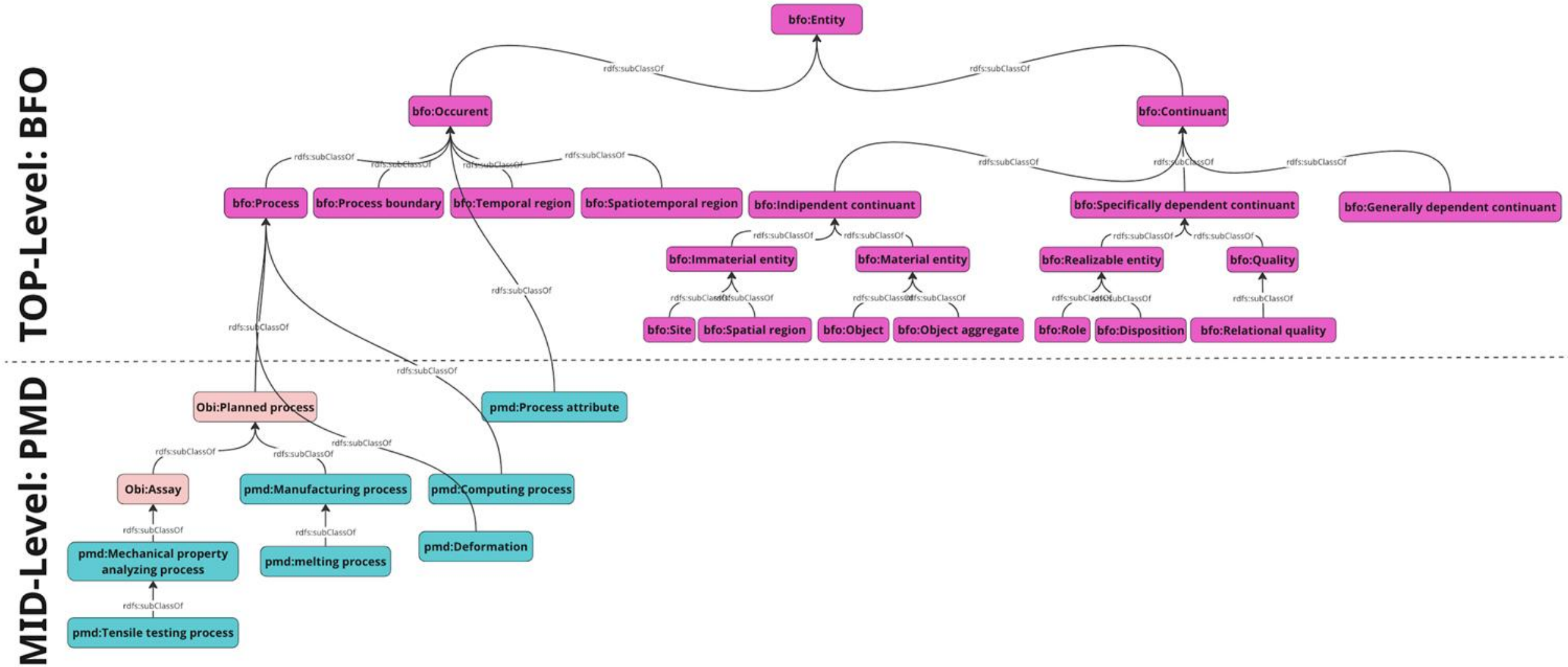
Objectives

- ✓ Semantic Interoperability
- ✓ FAIR Data Principles
- ✓ Workflow Modeling

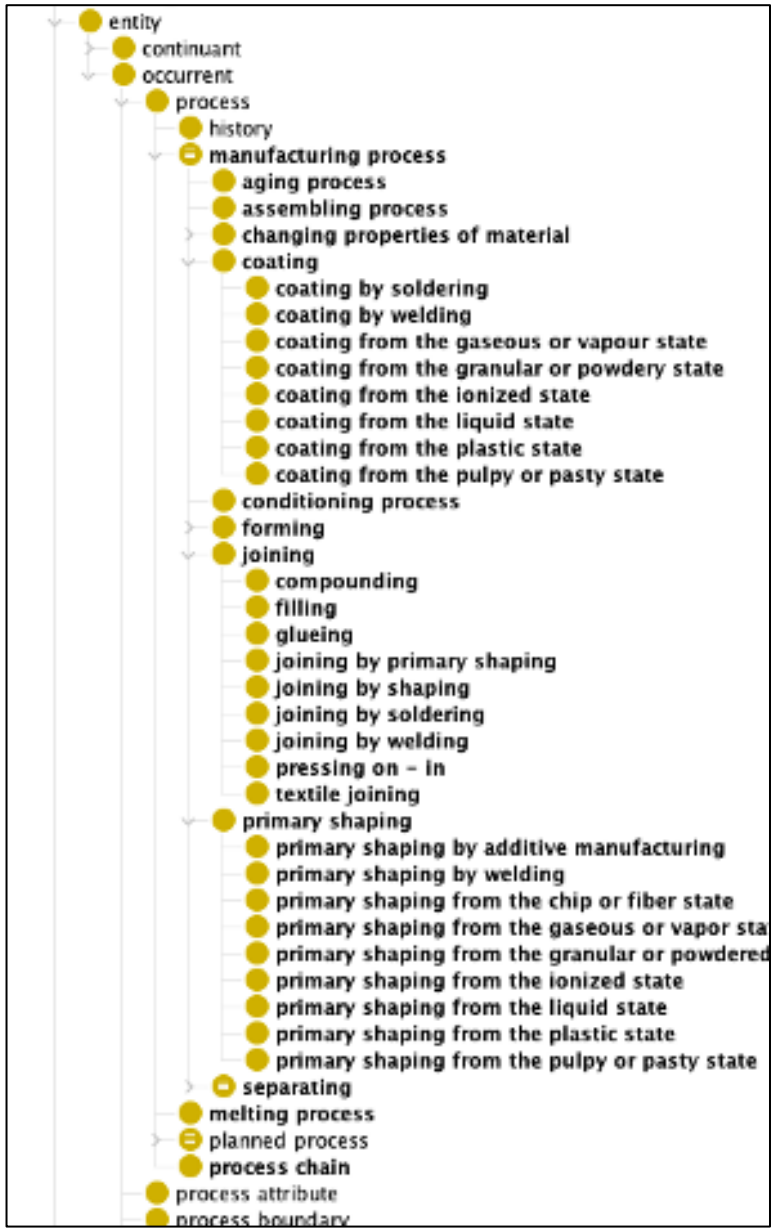
Applications

- ✓ Material Specifications
- ✓ Process Chains
- ✓ Data Integration and Analysis
- ✓ Device and Function Modeling

4- Platform MaterialDigital Core Ontology (PMDco) classes

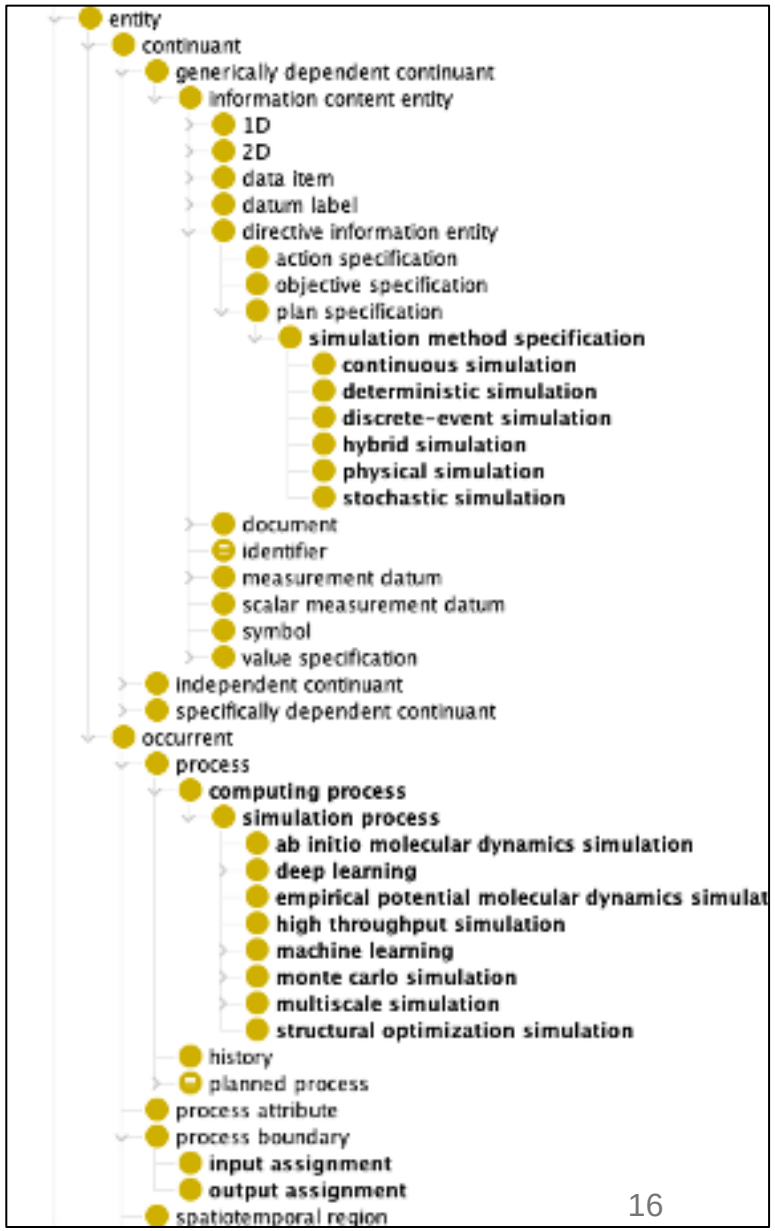


Manufacturing module

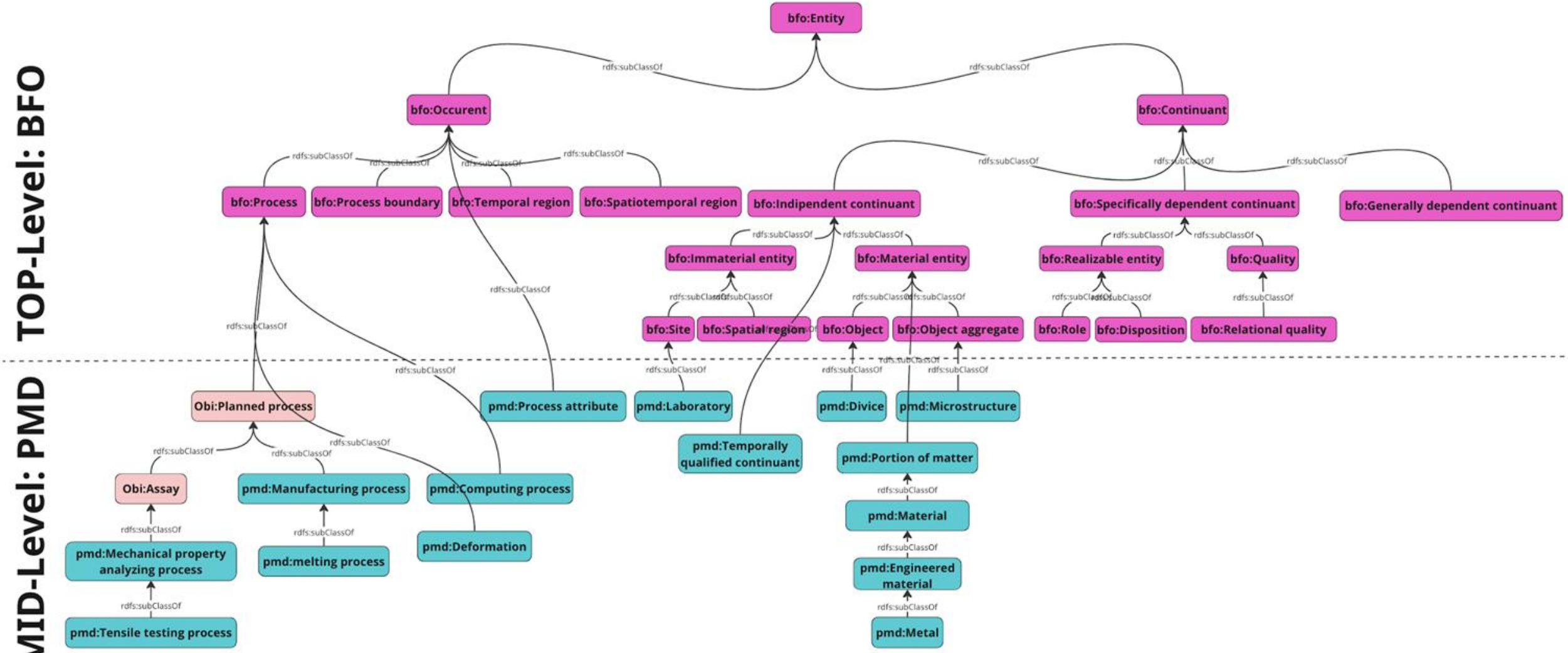


Material characterization module

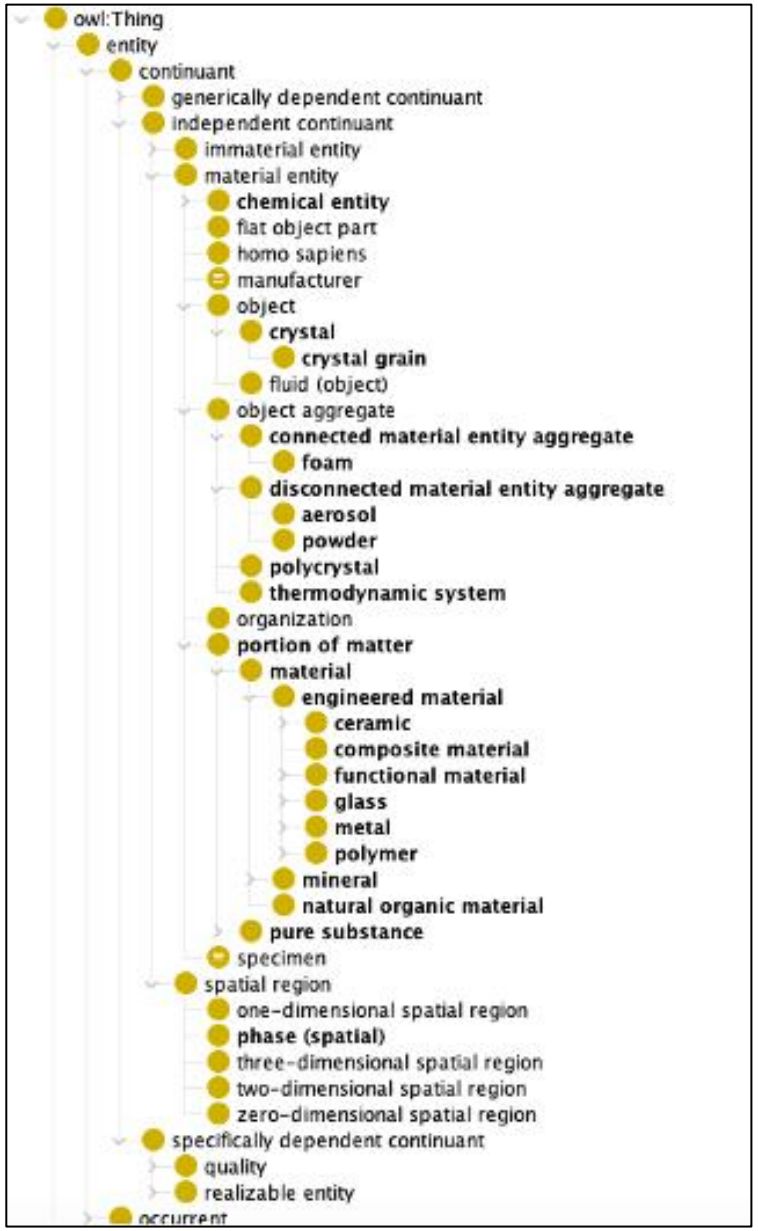
Data transformation module



4- Platform MaterialDigital Core Ontology (PMDco) classes

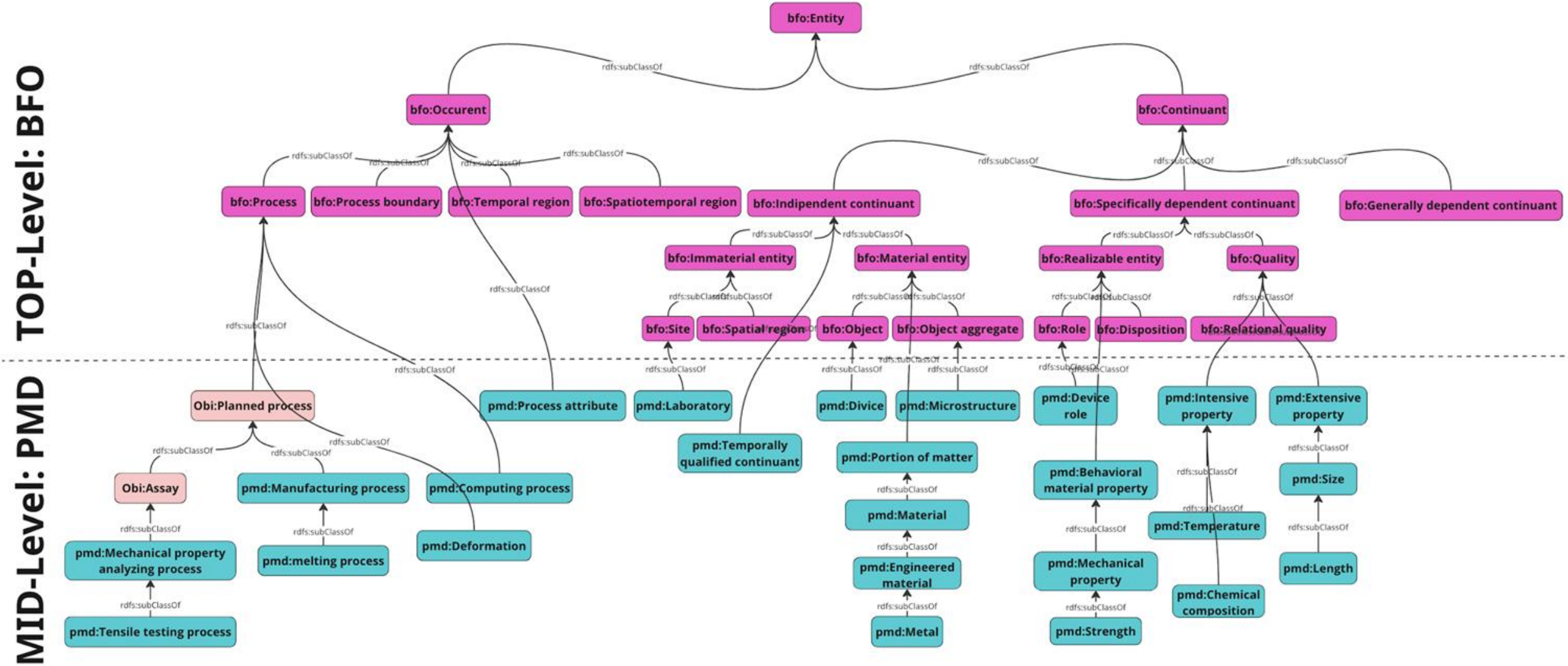


Materials module

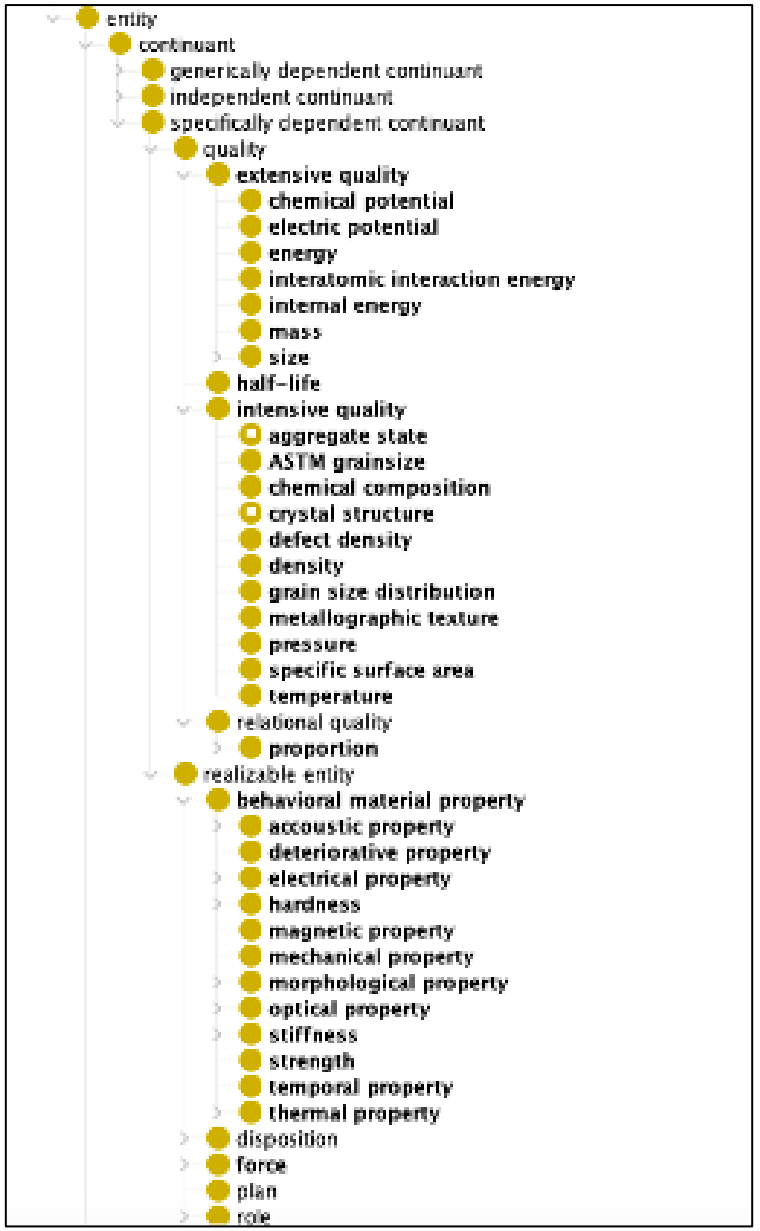


Devices module

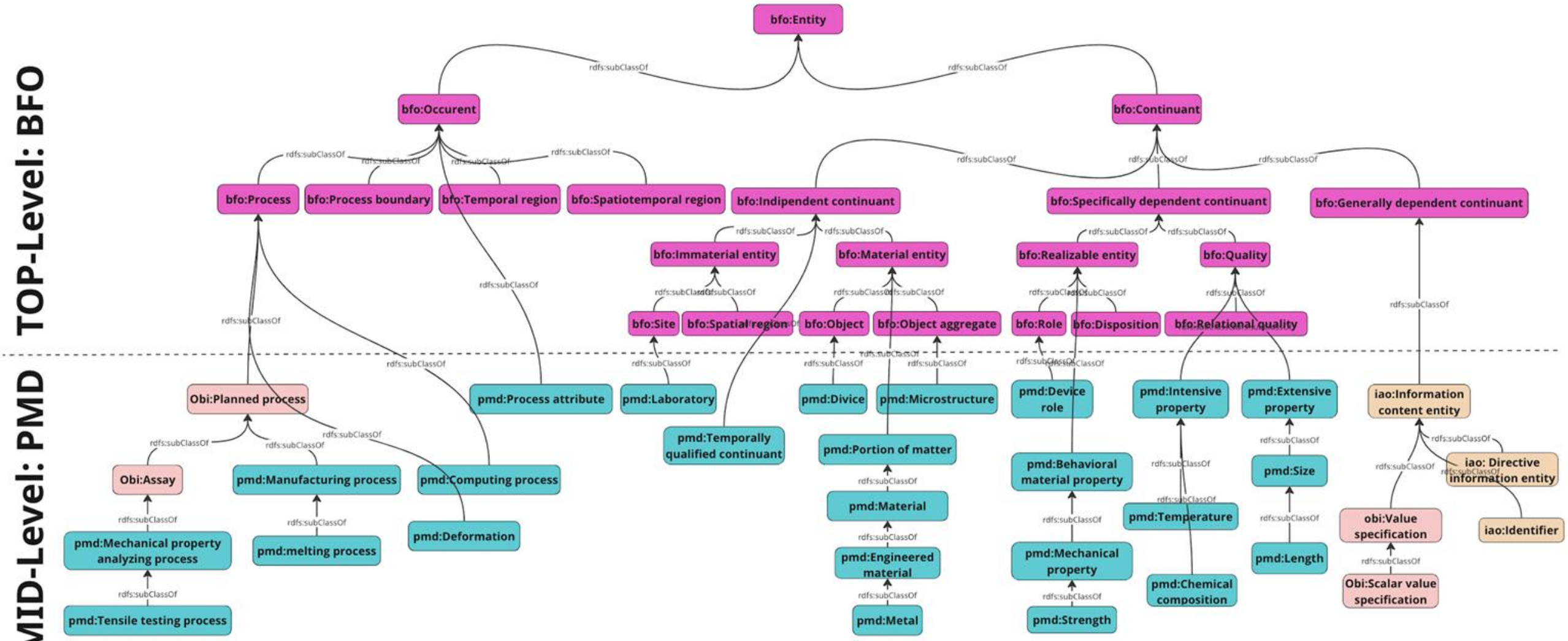
4- Platform MaterialDigital Core Ontology (PMDco) classes



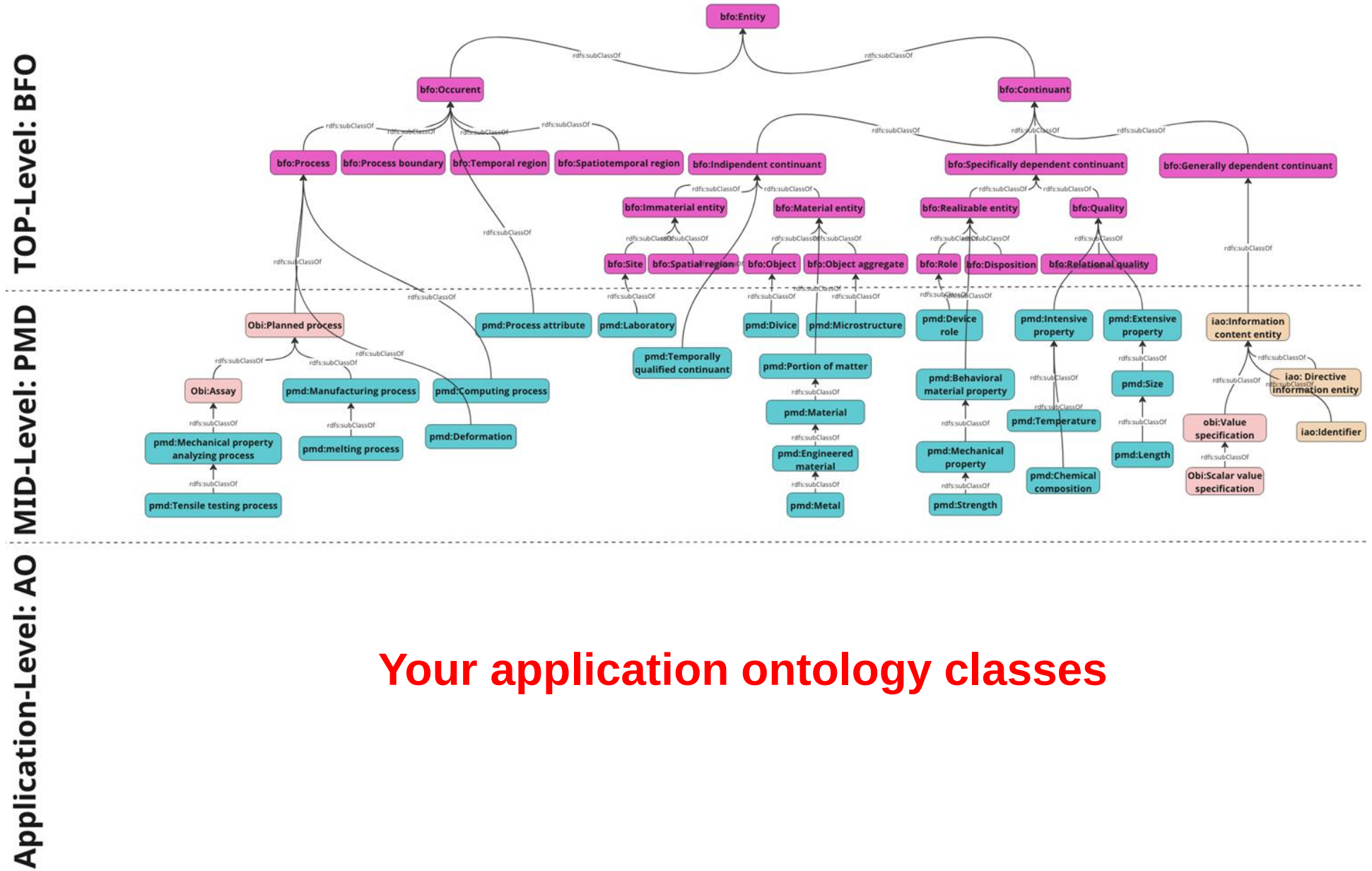
Qualities module



4- Platform MaterialDigital Core Ontology (PMDco) classes



4- Platform MaterialDigital Core Ontology (PMDco) classes



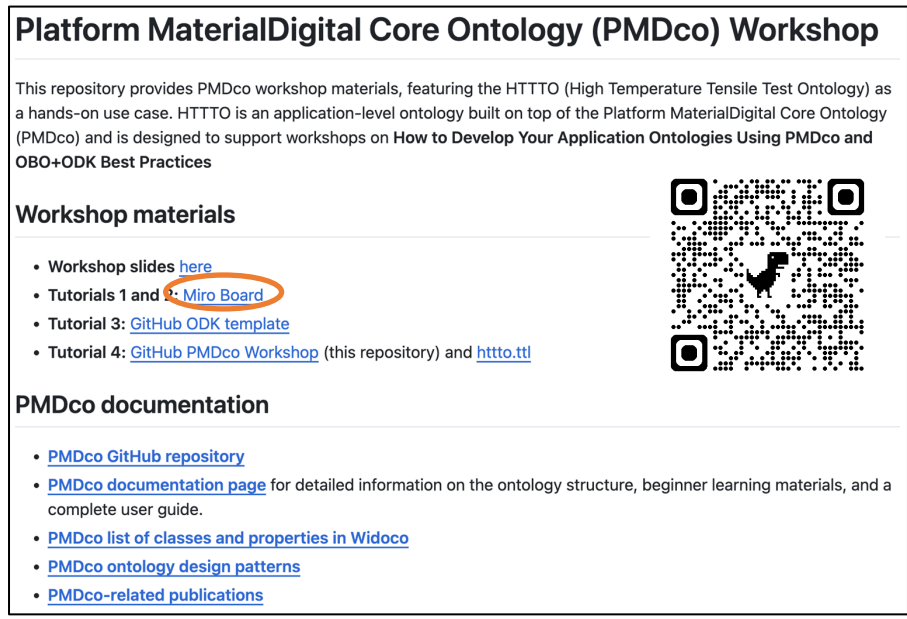
Your application ontology classes

Workshop example:
"High temperature tensile testing ontology" by reusing PMDco

Tutorial 1: Structure given classes according to PMDco hierarchy (Miro board)

** This tutorial designed based on PMDco v3.0.0.rc1*

Miro Board



Platform MaterialDigital Core Ontology (PMDco) Workshop

This repository provides PMDco workshop materials, featuring the HTTTO (High Temperature Tensile Test Ontology) as a hands-on use case. HTTTO is an application-level ontology built on top of the Platform MaterialDigital Core Ontology (PMDco) and is designed to support workshops on **How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices**

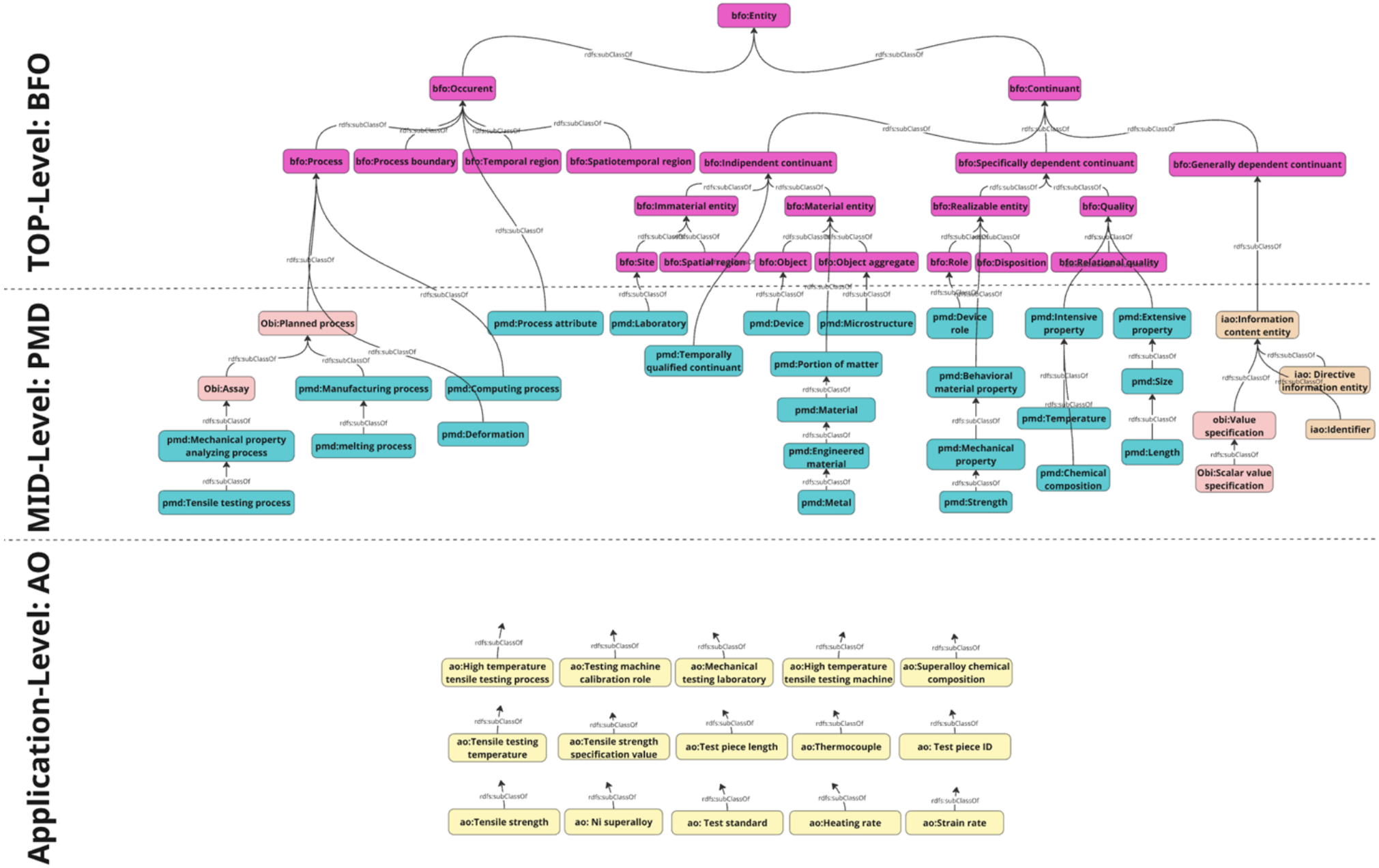
Workshop materials

- **Workshop slides** here
- **Tutorials 1 and 2:** Miro Board
- **Tutorial 3:** GitHub ODK template
- **Tutorial 4:** GitHub PMDco Workshop (this repository) and httto.ttl

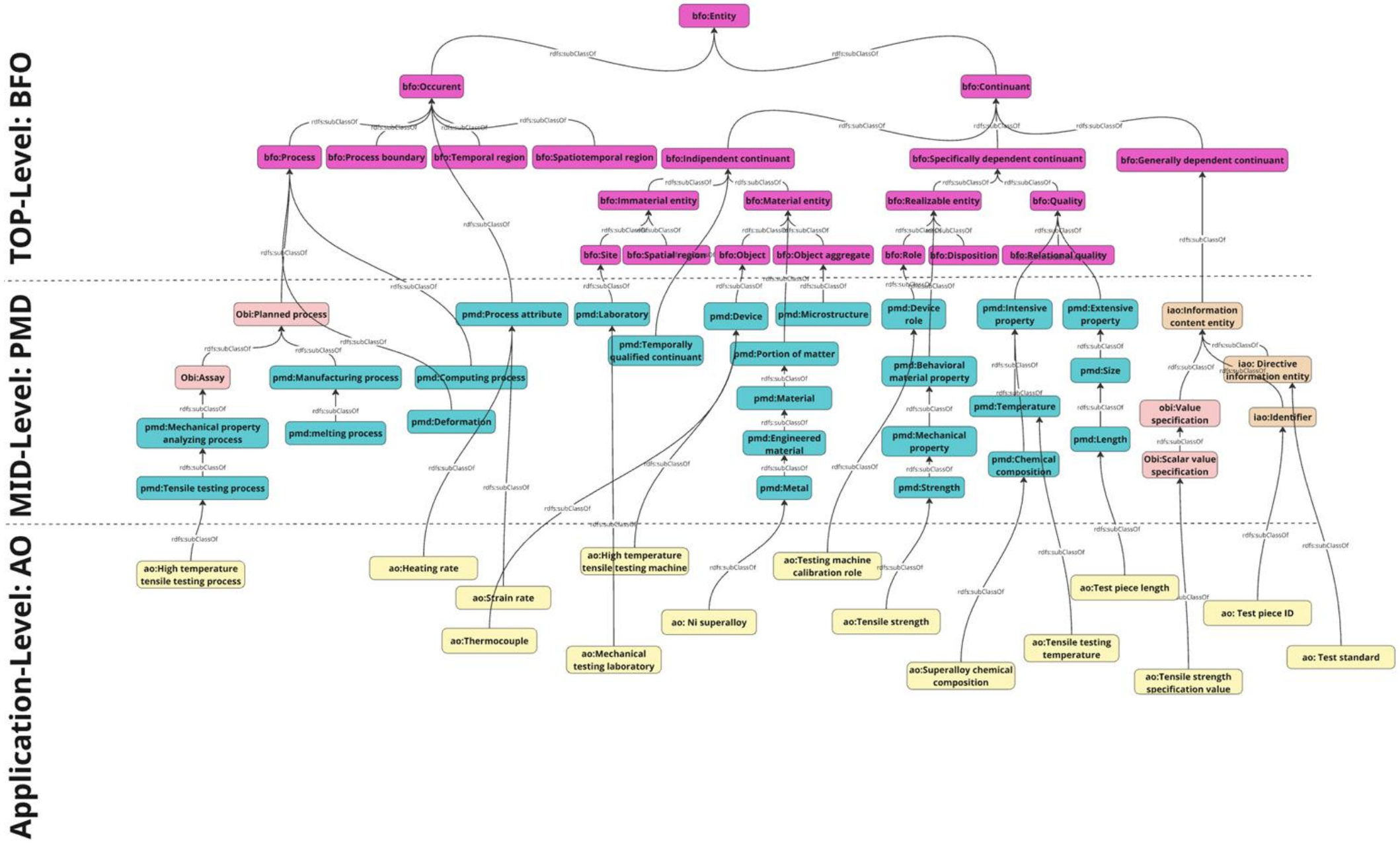
PMDco documentation

- PMDco GitHub repository
- PMDco documentation page for detailed information on the ontology structure, beginner learning materials, and a complete user guide.
- PMDco list of classes and properties in Widoco
- PMDco ontology design patterns
- PMDco-related publications

Tutorial 1: Structure given classes according to PMDco hierarchy



Tutorial 1: Answer

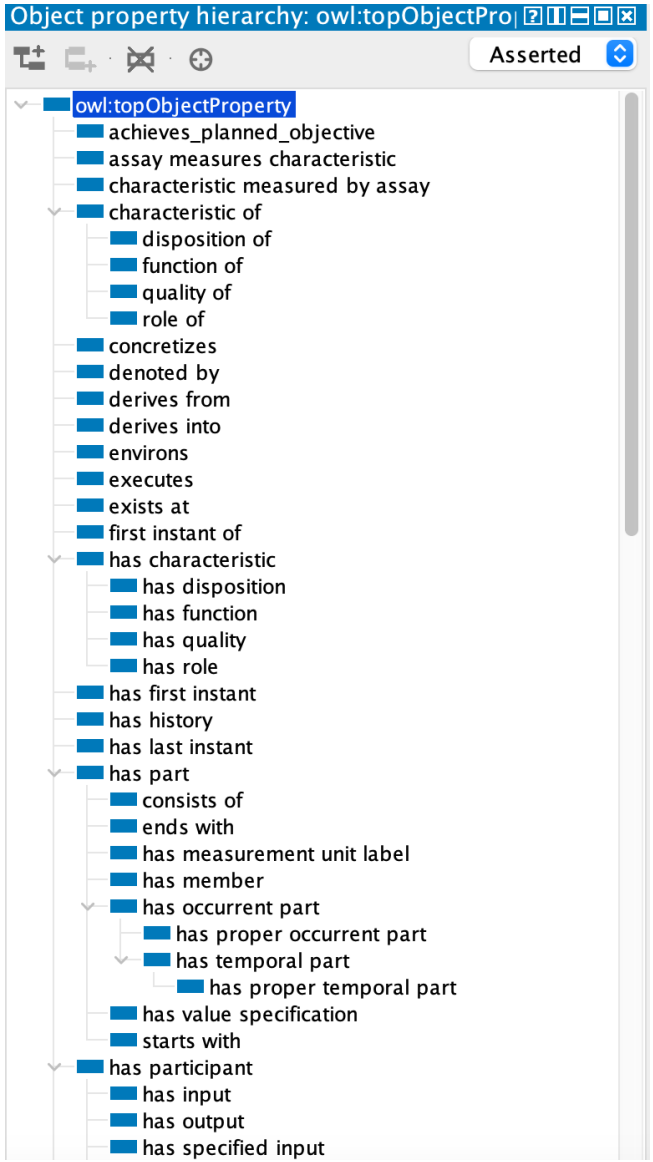


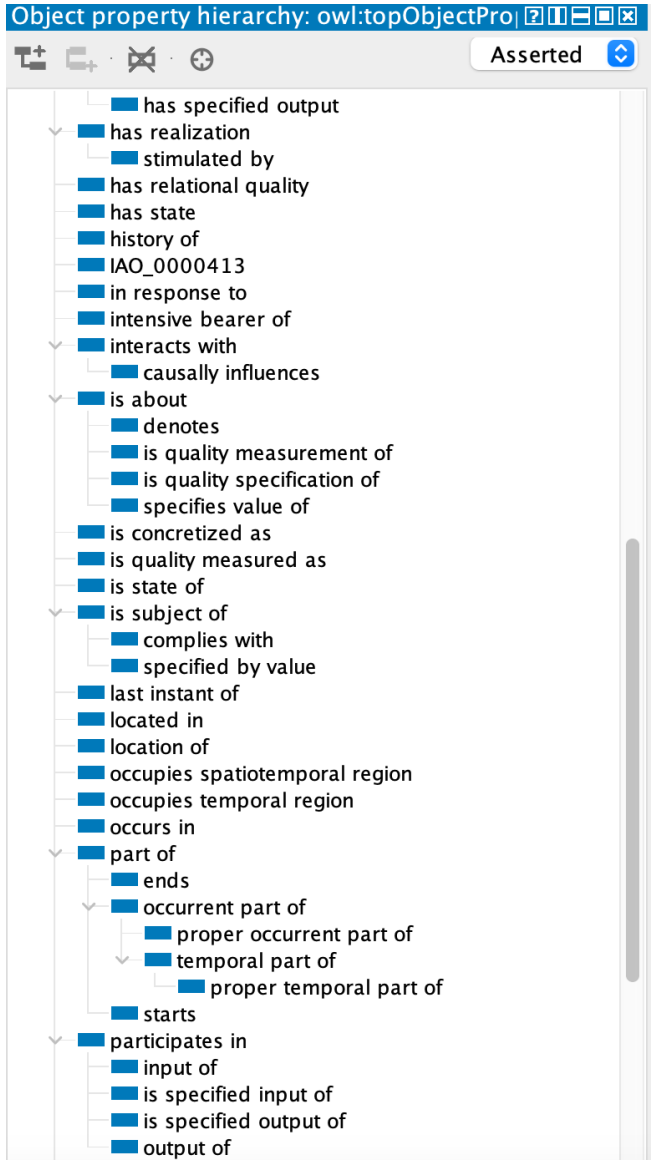
How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices

Workshop content (10 topics, 4 tutorials):

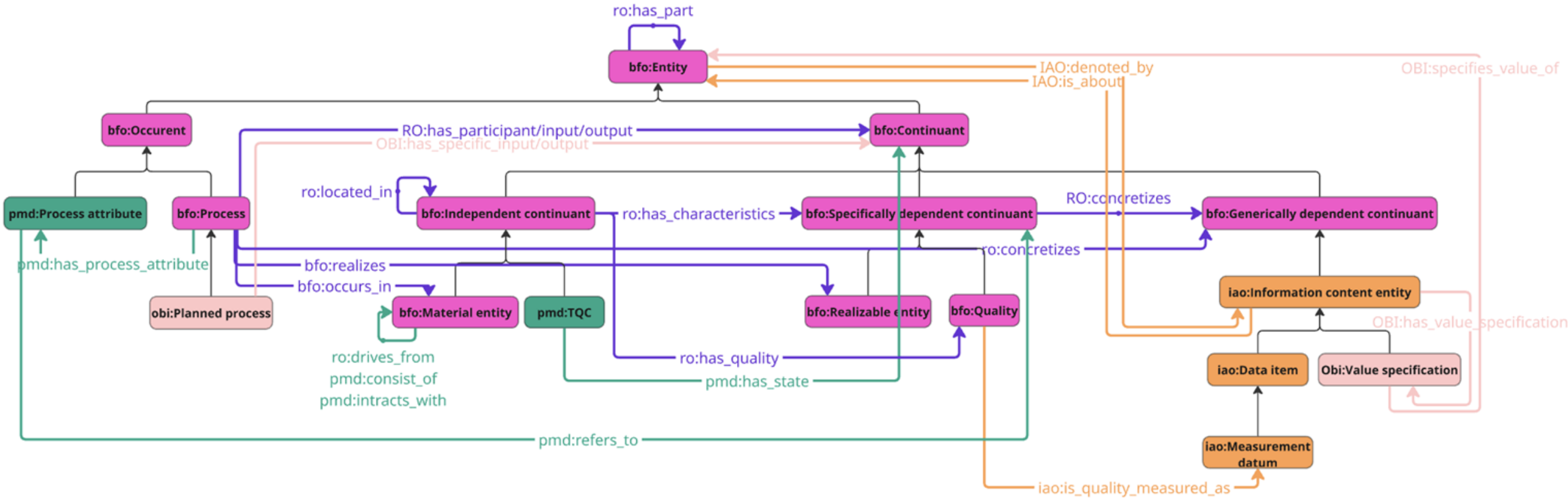
1- Ontology development and beginners learning materials
2- Ontology levels
3- Basic Formal Ontology (BFO) classes
4- Platform MaterialDigital Core Ontology (PMDco) classes
Tutorial 1: Structure given classes according to PMDco hierarchy (Miro board)
5- Platform MaterialDigital Core Ontology (PMDco) object properties
Tutorial 2: Using appropriate object properties (Miro board)
6- How to develop your application ontologies using PMDco and OBO+ODK best practices?
7- Ontology Development Kit (ODK)
Tutorial 3: Creating ODK repository for PMDco application ontologies (GitHub)
8- Collaborative ontology development workflow, adding taxonomy and axioms
9- PMDco Ontology Design Patterns (ODPs)
Tutorial 4: Ontology editing; adding classes, annotations and axioms (Protege)
10- Ontology evaluation, release, documentation and maintenance

5- Platform MaterialDigital Core Ontology (PMDco) object properties





5- Platform MaterialDigital Core Ontology (PMDco) object properties



Workshop example:
"High temperature tensile testing ontology" by reusing PMDco

Tutorial 2: Using appropriate object properties (Miro board)

** This tutorial designed based on PMDco v3.0.0.rc1*

Miro Board

Platform MaterialDigital Core Ontology (PMDco) Workshop

This repository provides PMDco workshop materials, featuring the HTTTO (High Temperature Tensile Test Ontology) as a hands-on use case. HTTTO is an application-level ontology built on top of the Platform MaterialDigital Core Ontology (PMDco) and is designed to support workshops on **How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices**

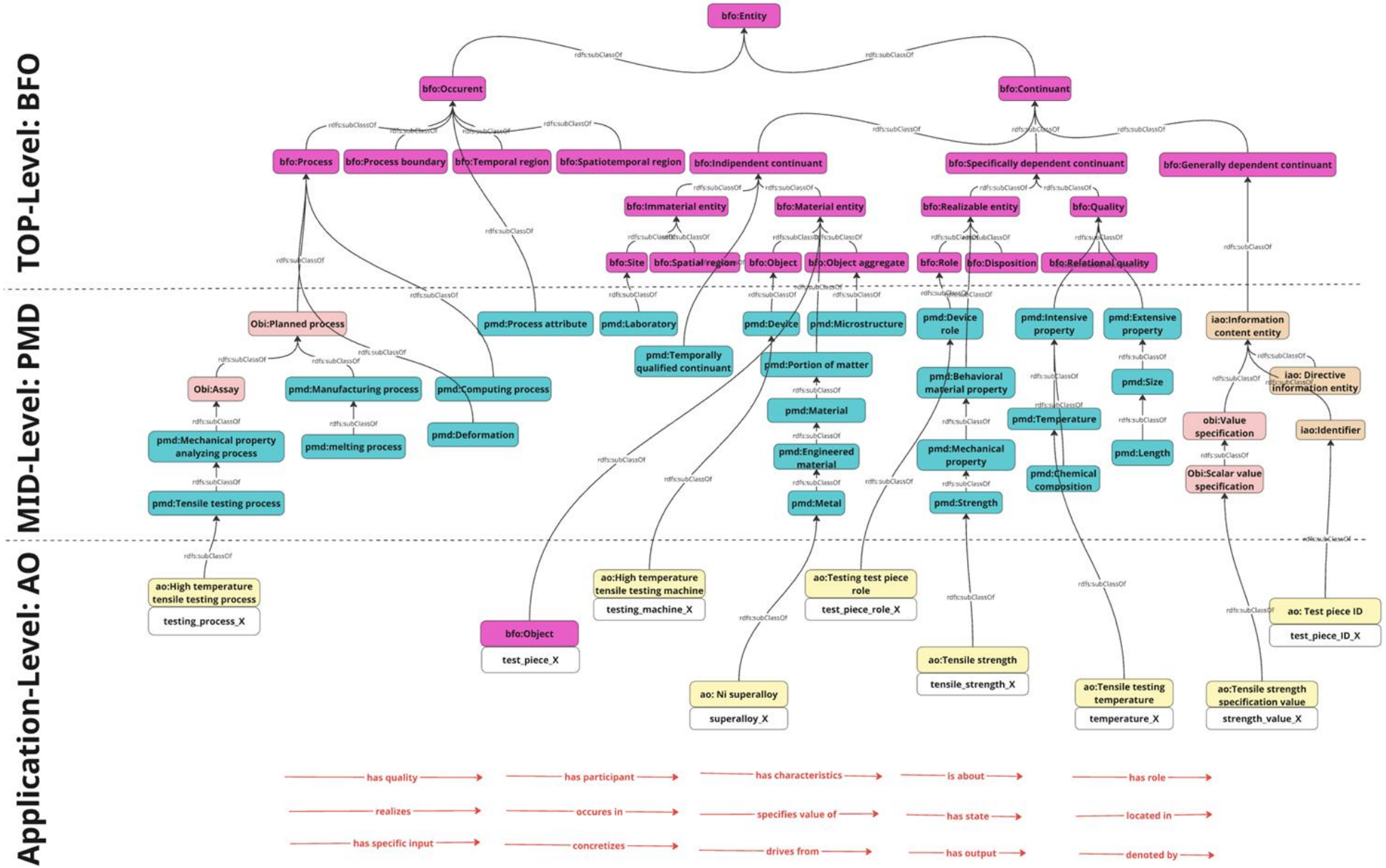
Workshop materials

- **Workshop slides** here
- **Tutorials 1 and 2:** Miro Board
- **Tutorial 3:** GitHub ODK template
- **Tutorial 4:** GitHub PMDco Workshop (this repository) and httto.ttl

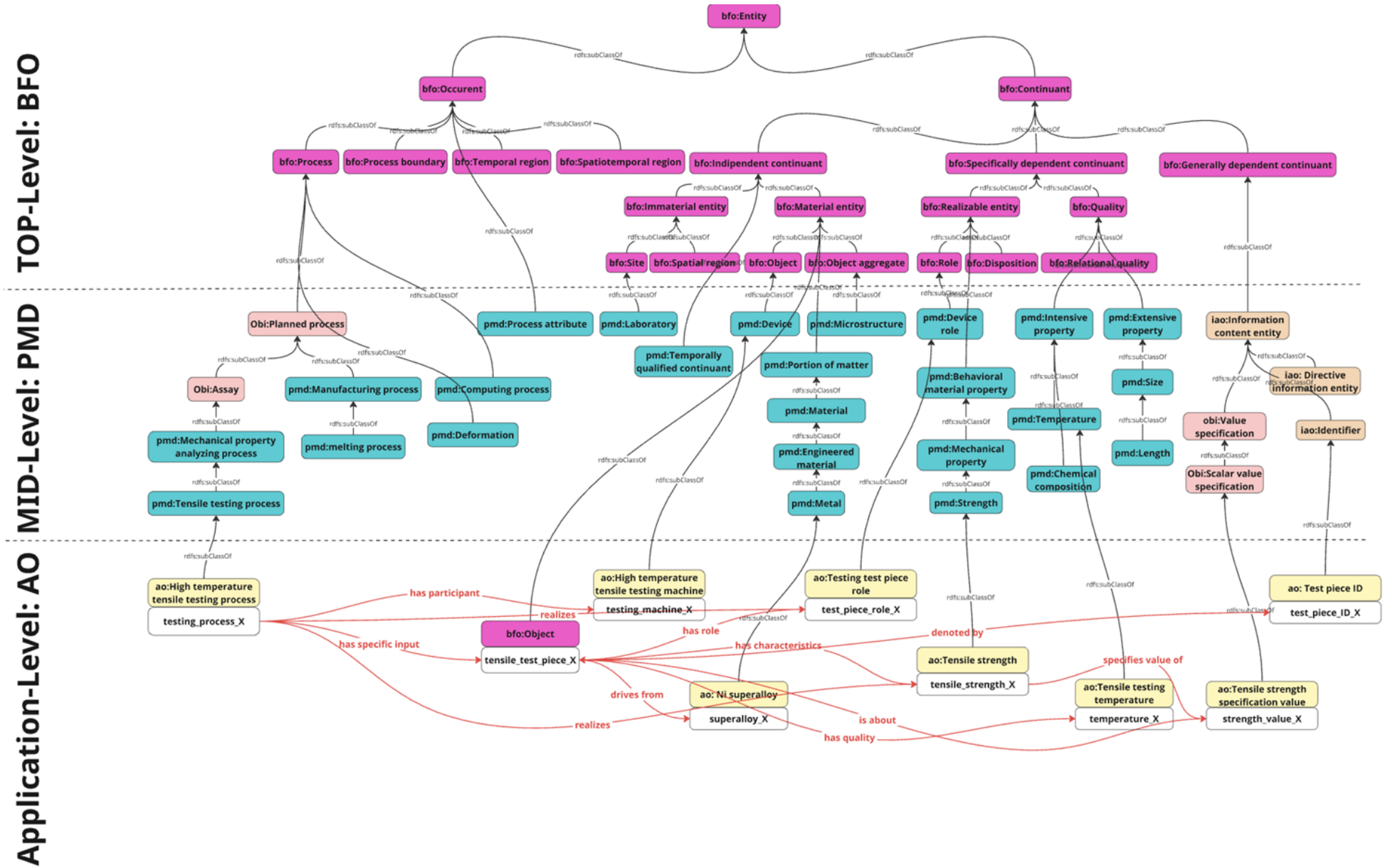
PMDco documentation

- PMDco GitHub repository
- PMDco documentation page for detailed information on the ontology structure, beginner learning materials, and a complete user guide.
- PMDco list of classes and properties in Widoco
- PMDco ontology design patterns
- PMDco-related publications

Tutorial 2: Using appropriate object properties



Tutorial 2: Answer

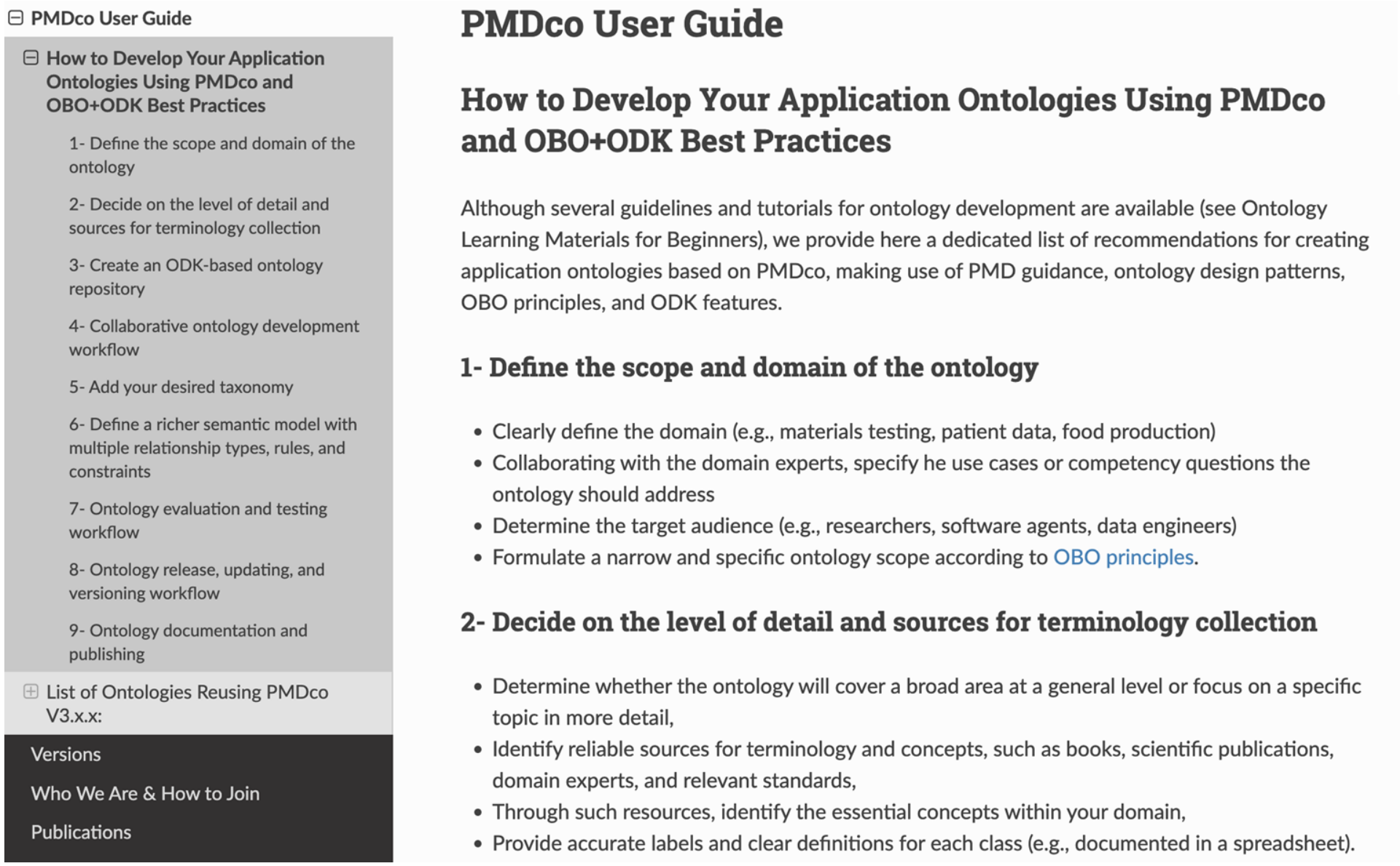


How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices

Workshop content (10 topics, 4 tutorials):

1- Ontology development and beginners learning materials
2- Ontology levels
3- Basic Formal Ontology (BFO) classes
4- Platform MaterialDigital Core Ontology (PMDco) classes
Tutorial 1: Structure given classes according to PMDco hierarchy (Miro board)
5- Platform MaterialDigital Core Ontology (PMDco) object properties
Tutorial 2: Using appropriate object properties (Miro board)
6- How to develop your application ontologies using PMDco and OBO+ODK best practices?
7- Ontology Development Kit (ODK)
Tutorial 3: Creating ODK repository for PMDco application ontologies (GitHub)
8- Collaborative ontology development workflow, adding taxonomy and axioms
9- PMDco Ontology Design Patterns (ODPs)
Tutorial 4: Ontology editing; adding classes, annotations and axioms (Protege)
10- Ontology evaluation, release, documentation and maintenance

6- How to develop your application ontologies using pmdco and OBO+ODK best practices?



7- Ontology Development Kit (ODK)

PMDco User Guide

How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices

- 1- Define the scope and domain of the ontology
- 2- Decide on the level of detail and sources for terminology collection
- **3- Create an ODK-based ontology repository**
- 4- Collaborative ontology development workflow
- 5- Add your desired taxonomy
- 6- Define a richer semantic model with multiple relationship types, rules, and constraints
- 7- Ontology evaluation and testing workflow
- 8- Ontology release, updating, and versioning workflow

3- Create an ODK-based ontology repository

- If you already know ODK, create a GitHub repository for ontology with a reasonable x.yaml file configuration and importing the latest PMDco version,
- If you don't know ODK or don't want to install it, we made a template for you. Simply create your repository following the instructions mentioned in "application-ontology-template". This template applies the same framework used for PMDco and mirrors PMDco with all its modules.

odk
ONTOLOGY DEVELOPMENT KIT

- Open-source toolkit and ontology life cycle management system for a standardized and transparent ontology development process.
- **Automation** of tasks like: import and release management, consistency checks, reasoning, and quality assurance.
- Supports us to comply to the **best-practices** of ontology development and avoid errors in imports and inconsistencies.
- **Modularized** approach through clearly separable components and modules.
- Uses technologies like ROBOT, Docker, GitHub Actions.
- Community-based tool with regular updates

Workshop example:
"High temperature tensile testing ontology" by reusing PMDco

Tutorial 3: Creating ODK repository for PMDco application ontologies (GitHub)

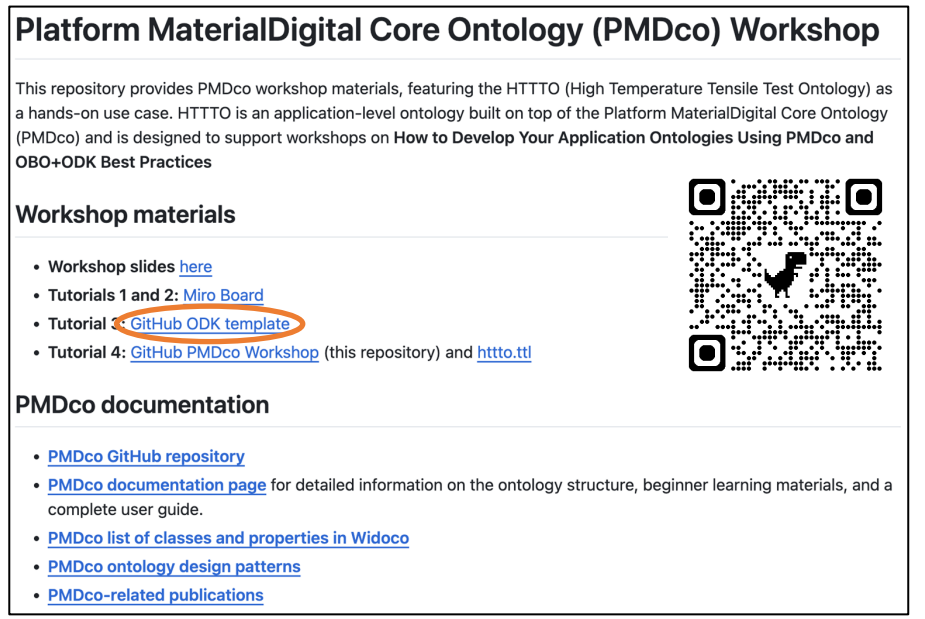
How to participate:

Case A) I plan developing professional AO:
We recommend learning ODK and use it for creating your repo. Now you can just watch how an ODK is looking like.

Case B) I want to test ODK, I may use it:
Participate in this tutorial and make your first ODK repo.

Case C) I don't plan developing AO and use ODK:
You can just watch how an ODK is looking like.

GitHub ODK template



Platform MaterialDigital Core Ontology (PMDco) Workshop

This repository provides PMDco workshop materials, featuring the HTTTO (High Temperature Tensile Test Ontology) as a hands-on use case. HTTTO is an application-level ontology built on top of the Platform MaterialDigital Core Ontology (PMDco) and is designed to support workshops on **How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices**

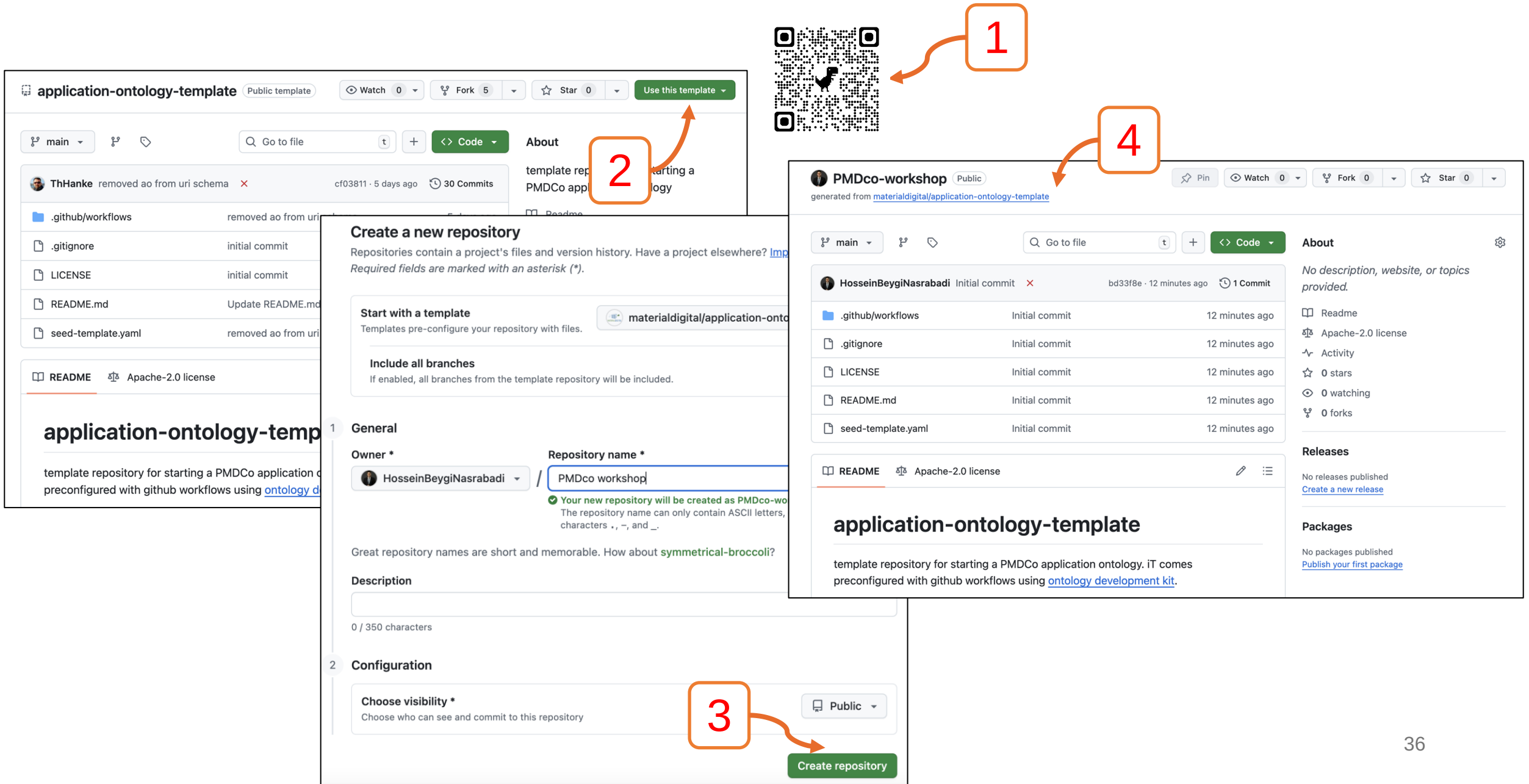
Workshop materials

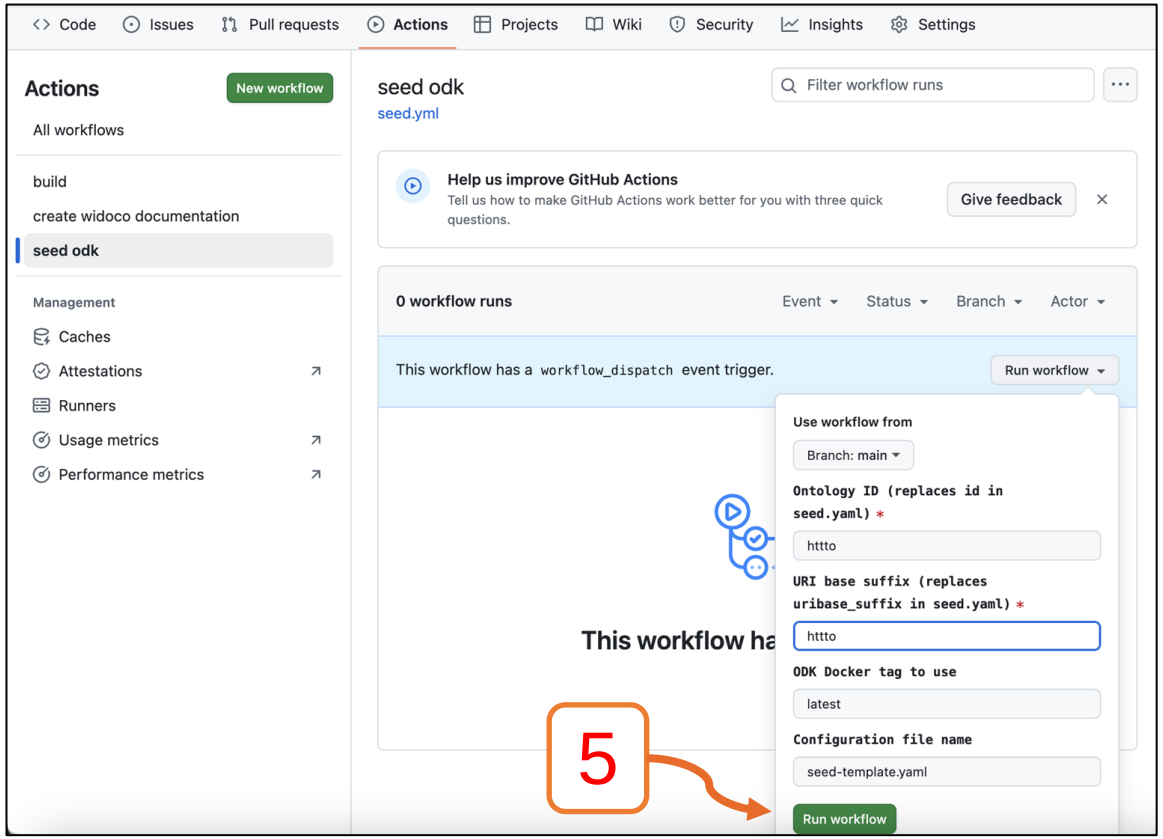
- **Workshop slides** here
- **Tutorials 1 and 2:** Miro Board
- **Tutorial 3:** GitHub ODK template
- **Tutorial 4:** GitHub PMDco Workshop (this repository) and httto.ttl

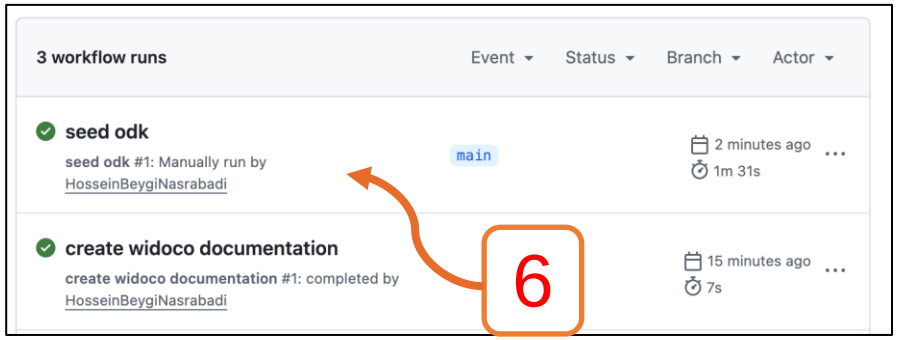
PMDco documentation

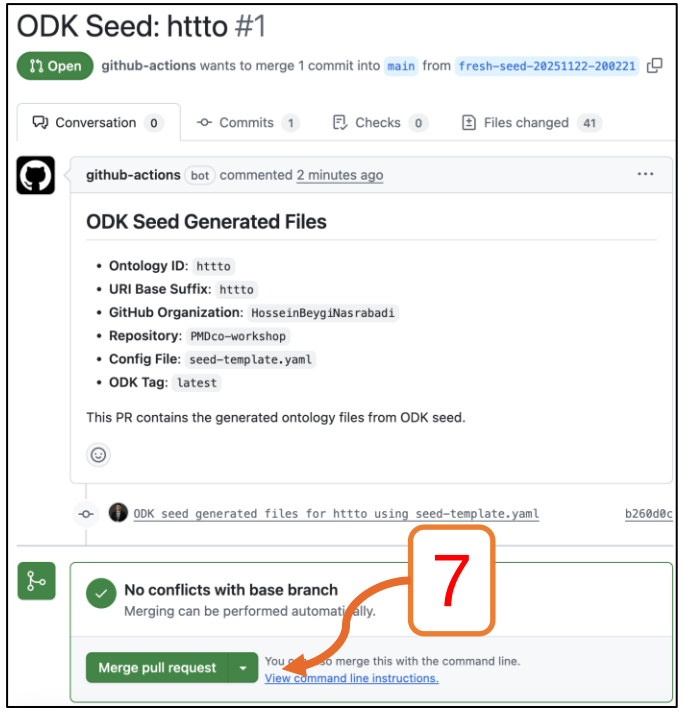
- PMDco GitHub repository
- PMDco documentation page for detailed information on the ontology structure, beginner learning materials, and a complete user guide.
- PMDco list of classes and properties in Widoco
- PMDco ontology design patterns
- PMDco-related publications

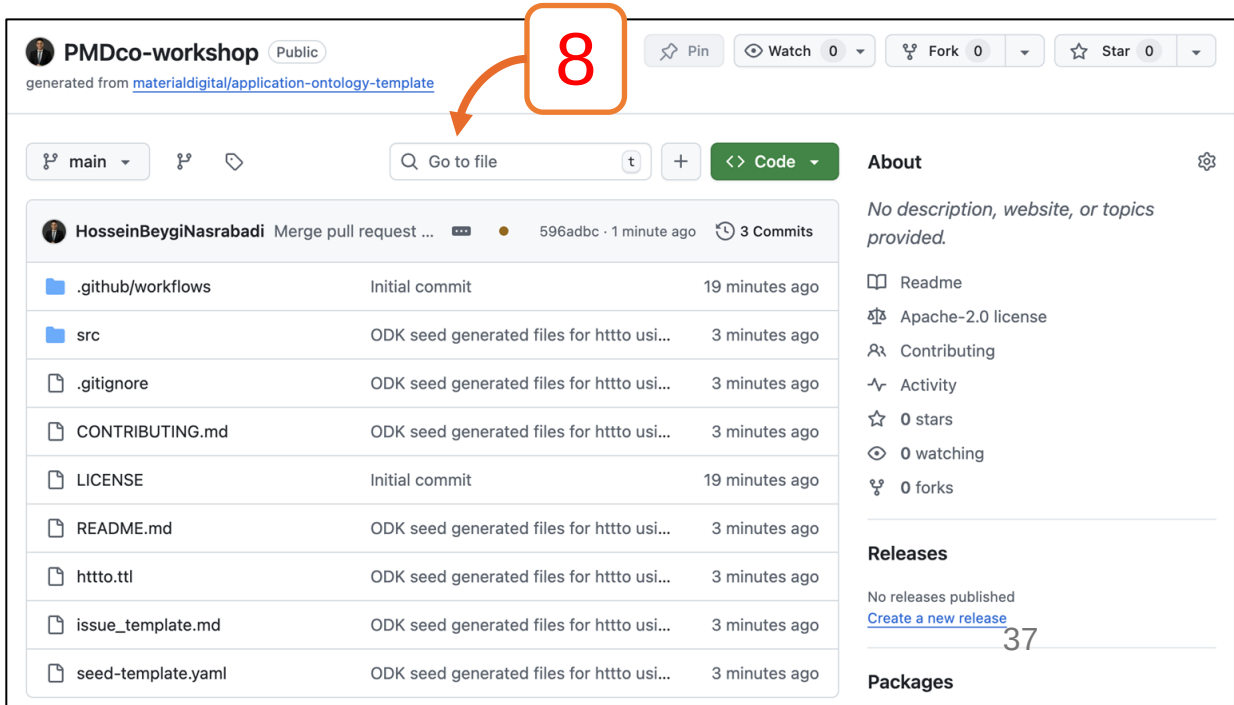
Tutorial 3: Creating ODK repository for PMDco application ontologies (GitHub)




Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Actions
New workflow
All workflows
build
create widoco documentation
seed odk
Management
Caches
Attestations
Runners
Usage metrics
Performance metrics
seed odk
seed.yml
Filter workflow runs
Help us improve GitHub Actions
Tell us how to make GitHub Actions work better for you with three quick questions.
Give feedback
0 workflow runs
Event
Status
Branch
Actor
This workflow has a workflow_dispatch event trigger.
Run workflow
Use workflow from
Branch: main
Ontology ID (replaces id in seed.yaml) *
httto
URI base suffix (replaces uribase_suffix in seed.yaml) *
httto
ODK Docker tag to use
latest
Configuration file name
seed-template.yaml
Run workflow
This workflow ha
5


3 workflow runs
Event
Status
Branch
Actor
seed odk
seed odk #1: Manually run by HosseinBeygiNasrabadi
main
2 minutes ago
1m 31s
create widoco documentation
create widoco documentation #1: completed by HosseinBeygiNasrabadi
15 minutes ago
7s
6


ODK Seed: httto #1
Open
github-actions wants to merge 1 commit into main from fresh-seed-20251122-200221
Conversation 0
Commits 1
Checks 0
Files changed 41
github-actions bot commented 2 minutes ago
ODK Seed Generated Files
Ontology ID: httto
URI Base Suffix: httto
GitHub Organization: HosseinBeygiNasrabadi
Repository: PMDco-workshop
Config File: seed-template.yaml
ODK Tag: latest
This PR contains the generated ontology files from ODK seed.
ODK seed generated files for httto using seed-template.yaml
b260d0c
No conflicts with base branch
Merging can be performed automatically.
Merge pull request
View command line instructions.
7


PMDco-workshop
Public
generated from materialdigital/application-ontology-template
Pin
Watch 0
Fork 0
Star 0
8
main
Go to file
Code
HosseinBeygiNasrabadi Merge pull request ...
596adbc · 1 minute ago
3 Commits
.github/workflows Initial commit 19 minutes ago
src ODK seed generated files for httto usi... 3 minutes ago
.gitignore ODK seed generated files for httto usi... 3 minutes ago
CONTRIBUTING.md ODK seed generated files for httto usi... 3 minutes ago
LICENSE Initial commit 19 minutes ago
README.md ODK seed generated files for httto usi... 3 minutes ago
httto.ttl ODK seed generated files for httto usi... 3 minutes ago
issue_template.md ODK seed generated files for httto usi... 3 minutes ago
seed-template.yaml ODK seed generated files for httto usi... 3 minutes ago
About
No description, website, or topics provided.
Readme
Apache-2.0 license
Contributing
Activity
0 stars
0 watching
0 forks
Releases
No releases published
Create a new release
Packages

How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices

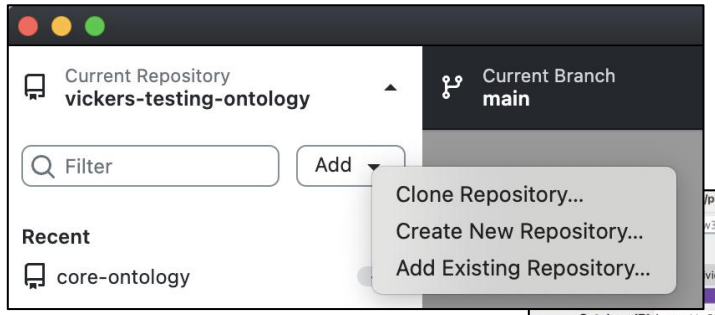
Workshop content (10 topics, 4 tutorials):

1- Ontology development and beginners learning materials
2- Ontology levels
3- Basic Formal Ontology (BFO) classes
4- Platform MaterialDigital Core Ontology (PMDco) classes
Tutorial 1: Structure given classes according to PMDco hierarchy (Miro board)
5- Platform MaterialDigital Core Ontology (PMDco) object properties
Tutorial 2: Using appropriate object properties (Miro board)
6- How to develop your application ontologies using PMDco and OBO+ODK best practices?
7- Ontology Development Kit (ODK)
Tutorial 3: Creating ODK repository for PMDco application ontologies (GitHub)
8- Collaborative ontology development workflow, adding taxonomy and axioms
9- PMDco Ontology Design Patterns (ODPs)
Tutorial 4: Ontology editing; adding classes, annotations and axioms (Protege)
10- Ontology evaluation, release, documentation and maintenance

8- Collaborative ontology development workflow, adding taxonomy and axioms

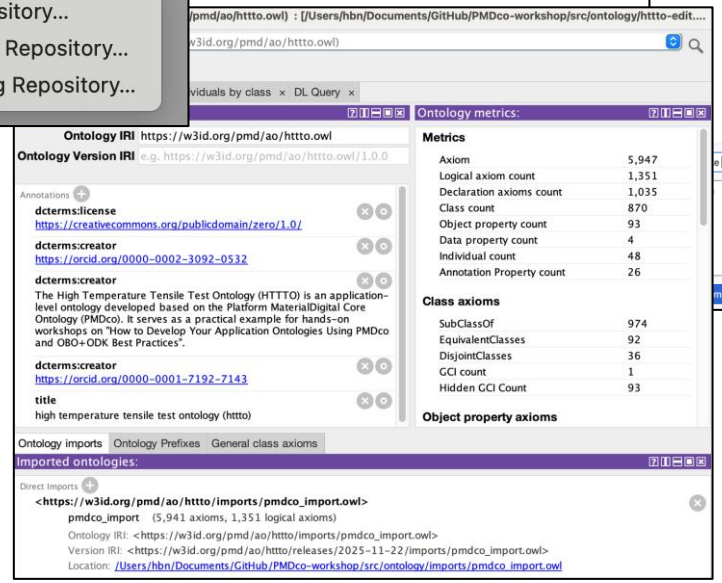
4- Collaborative ontology development workflow

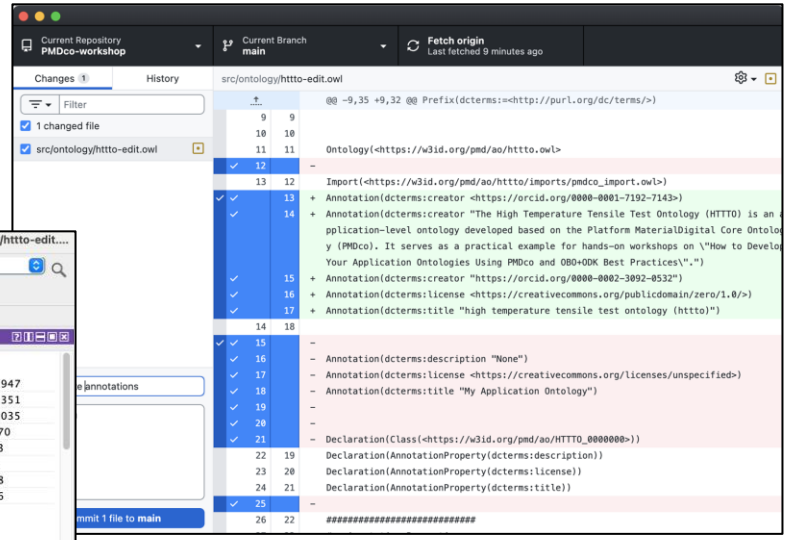
- Any person who wants to contribute to the collaborative development of ontology should clone the ODK repository to the local machine,
- In Protégé, open the editable file: src/ontology/*-edit.owl (Any editor can manage an assigned ID range),
- Edit the ontology, like add classes/properties, update definitions, and organize axioms,
- Save and validate locally,
- Commit your edits and push them to your branch,
- Create a Pull Request, submit a PR with a short summary and justification of your changes,
- Team members review; after approval, the PR is merged.



1- GitHub desktop: clone the repository

2- Protege: src/ontology/httto-edit.owl Edit Ontology





3- GitHub desktop: commit and push

8- Collaborative ontology development workflow, adding taxonomy and axioms

5- Add your desired taxonomy

- Edit ontology in protégé,
- Create new classes, properties, and individuals according to your terminology lists,
- Structure them hierarchically in alignment with reused top- and mid-level ontologies, where subclasses inherit properties from their superclasses,
- Evaluate other OBO Foundry ontologies (e.g., using Ontobee) to check for existing classes and to guide your classification. In the case you want to import entities from other ontologies, please follow the instructions here,
- Following OBO principles, build the ontology primarily by extending the class hierarchy and minimizing the introduction of new object properties.
- Add rich annotations for new entities (such as definitions, references, examples, and editorial notes).
- Include comprehensive ontology annotation metadata (e.g., title, label, description, creators, contributors, version info, previous versions, license, and citation).

6- Define a richer semantic model with multiple relationship types, rules, and constraints

- Define logical axioms and property constraints (e.g., domain and range),
- Use graphical tools (e.g., Ontopanel or Miro) to visually discuss the modeling issues,
- Use our Ontology Design Patterns as a guideline for the semantic representation of your concepts,
- See other PMDco-based application ontologies (listed below this page) for similar modeling concepts.
- If you still need support for specific modelling issues, please feel free to mention it in our GitHub issues or discussion forum.

9- PMDco Ontology Design Patterns (ODPs)

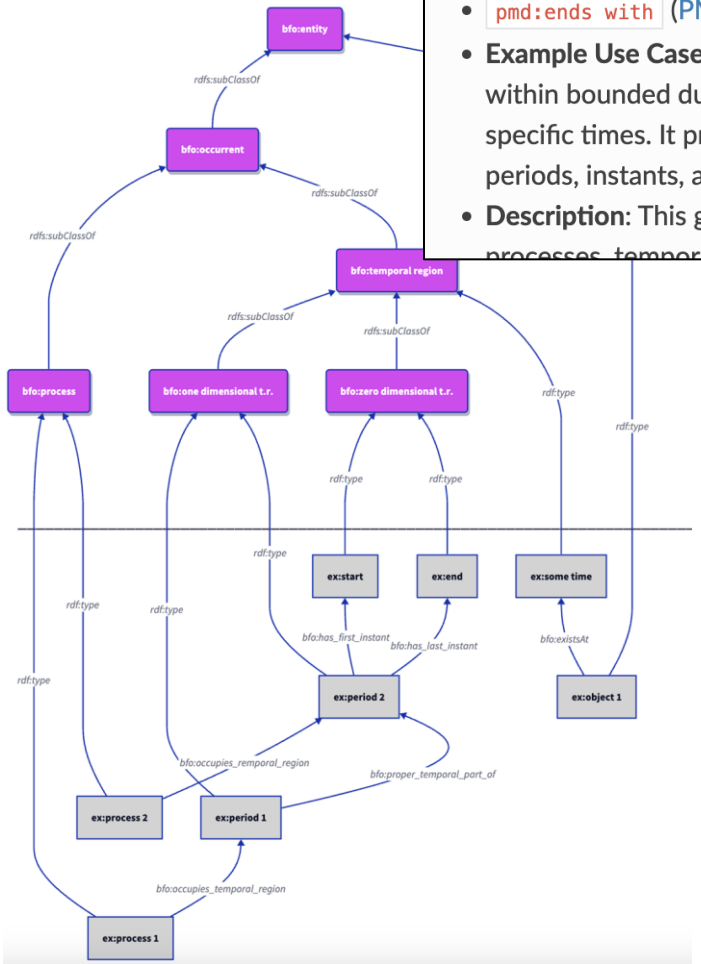
https://github.com/materialdigital/core-ontology/tree/main/patterns

Patterns

- **Pattern 1 - Temporal Region**
- Pattern 2 - Process Chain
- Pattern 3 - Process Inputs and Outputs
- Pattern 4 - Realizable Entities (Role)
- Pattern 5 - Realizable Entities (Qualities)
- Pattern 6 - Scalar Measurement
- Pattern 7 - Scalar Value Specification
- Pattern 8 - Categorical Value Specification
- Pattern 9 - Material and Device Specification

Pattern 1 - Temporal Region

- **Purpose**: Specifying the boundaries of a process on the time axis.
- **Core Properties**:
- `bfo:occupies temporal region` (BFO_0000199)
- `bfo:proper temporal part of` (BFO_0000136)
- `bfo:has first instant` (BFO_0000222)
- `bfo:has last instant` (BFO_0000224)
- `pmd:ends with` (PMD_0060003)
- **Example Use Case**: the pattern defines a temporal ontology where processes unfold within bounded durations, th
specific times. It provides a d
periods, instants, and enduri
- **Description**: This graph descr
processes, temporal regions




```
@prefix : <https://w3id.org/pmd/co/test#> .
@prefix ex: <https://www.example.org/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <https://w3id.org/pmd/co/test#> .

@prefix process: <http://purl.obolibrary.org/obo/BFO_0000015>  . #Class
@prefix continuant: <http://purl.obolibrary.org/obo/BFO_0000002> .  #Class
@prefix temporal_region: <http://purl.obolibrary.org/obo/BFO_0000008> . #Class
@prefix one-dimensional_temporal_region: <http://purl.obolibrary.org/obo/BFO_0000038>
@prefix zero-dimensional_temporal_region: <http://purl.obolibrary.org/obo/BFO_0000148

@prefix occupies_temporal_region: <http://purl.obolibrary.org/obo/BFO_0000199> . #Obj
@prefix proper_temporal_part_of: <http://purl.obolibrary.org/obo/BFO_0000136> . #Obje
@prefix has_first_instant: <http://purl.obolibrary.org/obo/BFO_0000222> . #ObjectProp
@prefix has_last_instant: <http://purl.obolibrary.org/obo/BFO_0000224> . #ObjectPrope
@prefix exists_at: <http://purl.obolibrary.org/obo/BFO_0000108> . #ObjectProperty

<https://w3id.org/pmd/co/test/shape/temporal_region> rdf:type owl:Ontology .

ex:process_1 a process: .
ex:process_2 a process: .
ex:period_1 a one-dimensional_temporal_region: .
ex:period_2 a one-dimensional_temporal_region: .
ex:some_time a temporal_region: .
ex:object_1 a continuant: .
ex:start a zero-dimensional_temporal_region: .
ex:end a zero-dimensional_temporal_region: .

ex:process_1 occupies_temporal_region: ex:period_1 .
ex:process_2 occupies_temporal_region: ex:period_2 .
ex:period_2 has_first_instant: ex:start .
ex:period_2 has_last_instant: ex:end .
ex:period_1 proper_temporal_part_of: ex:period_2 .
ex:object_1 exists_at: ex:some_time .
```

Workshop example:
"High temperature tensile testing ontology" by reusing PMDco

Tutorial 4: Ontology editing; adding classes, annotations and axioms (Protege)

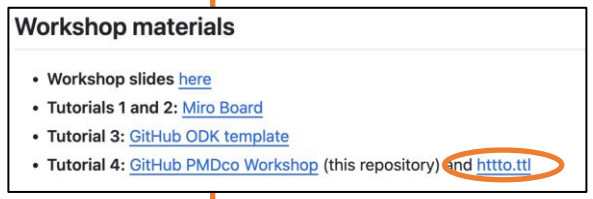
How to participate:

Option A (commit and push possibility):
1- GitHub desktop: clone the "PMDco Workshop" repository (QR code link below)
2- Protégé: open file> src/ontology/httto-edit.owl

GitHub PMDco Workshop

Option B (faster, just for participating in tutorial):
1- Protégé: open from URL> QR code link below)

httto.ttl

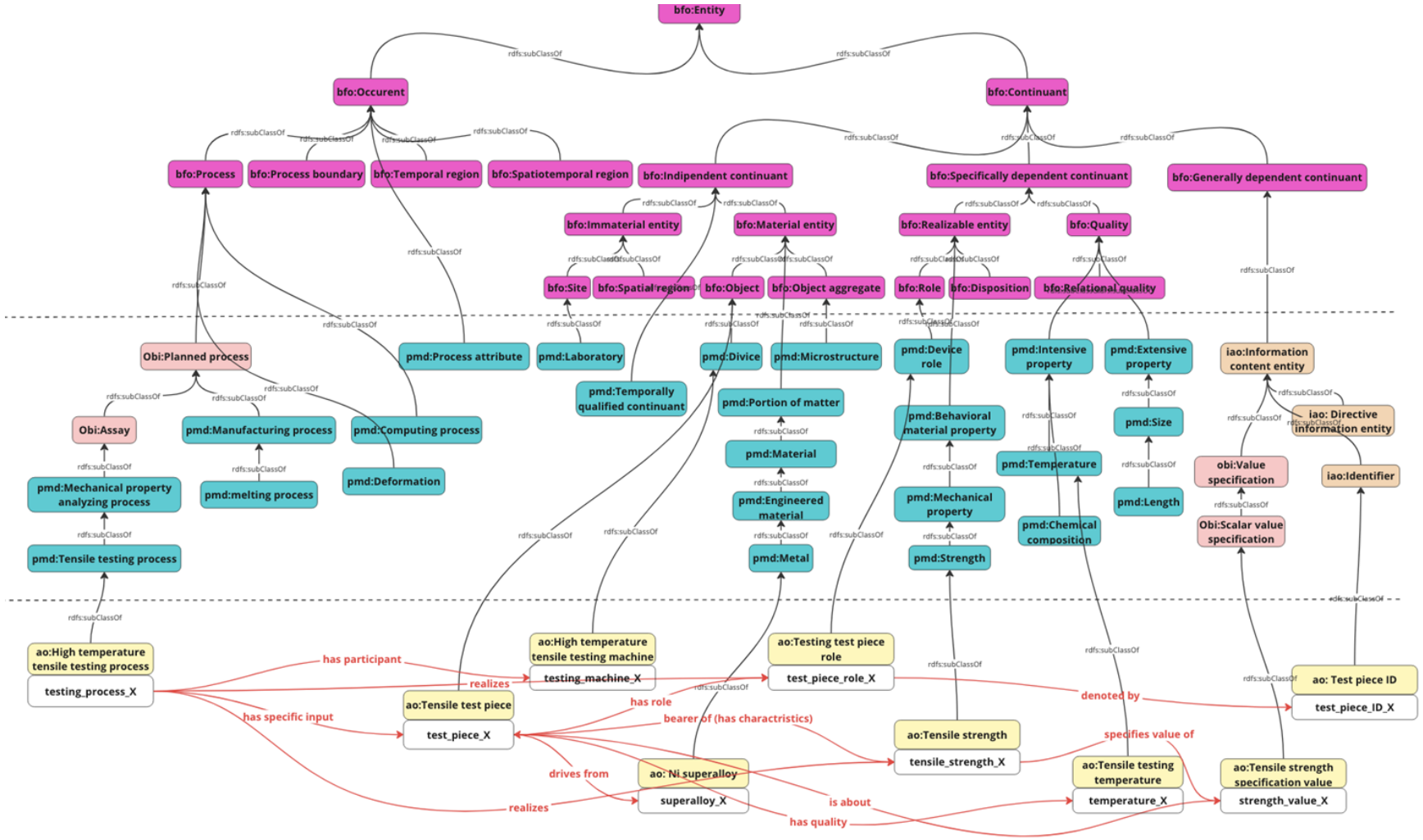


Tutorial 4: Ontology editing; adding classes, annotations and axioms (Protege)

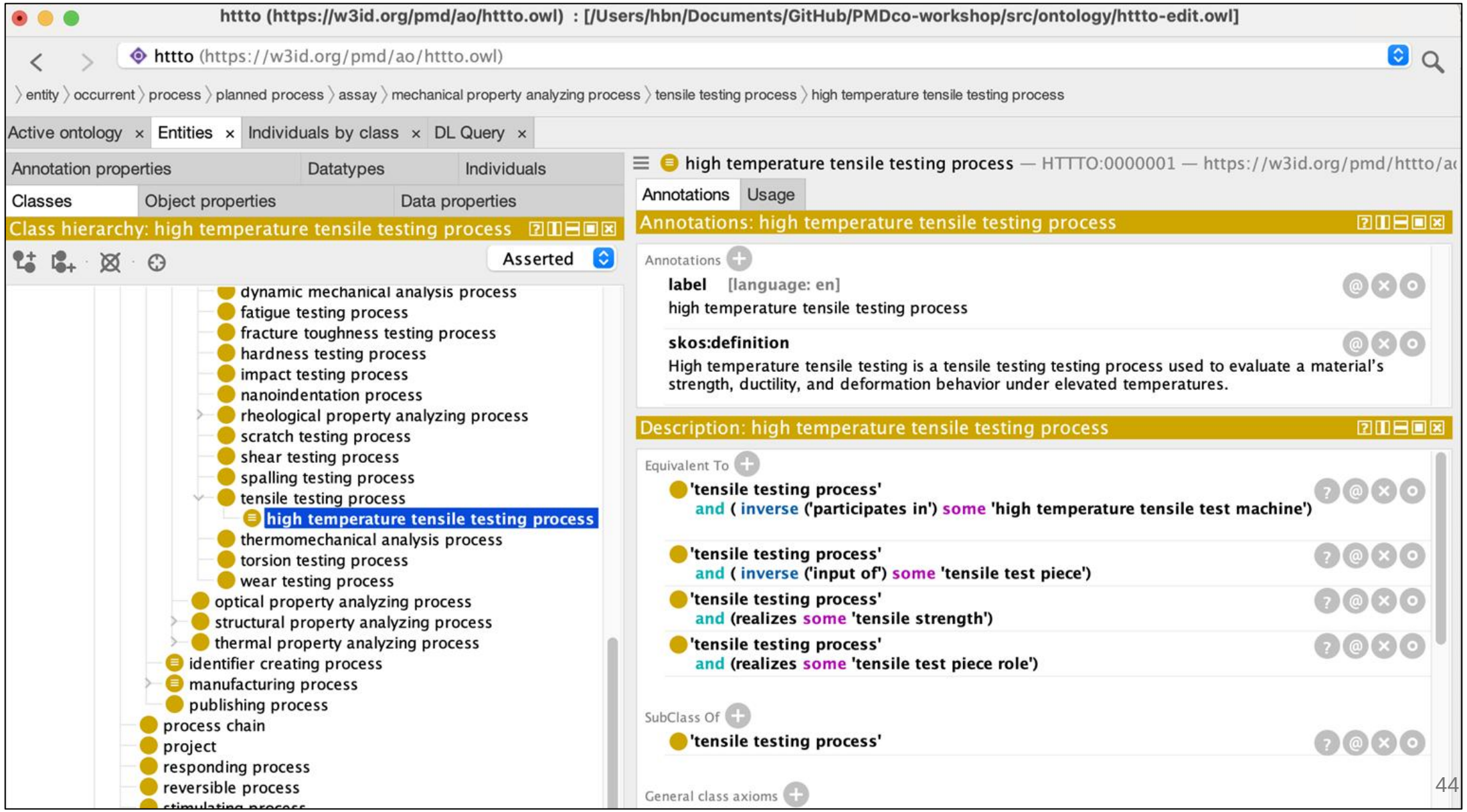
1- Add ao classes (yellow ones) in their designed hierarchy

2- Create annotations (at least definitions) for your created classes

3- Create axioms (as shown in the graph) for your created classes



Tutorial 4: Example adding classes, annotations, and axioms



How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices

Workshop content (10 topics, 4 tutorials):

1- Ontology development and beginners learning materials
2- Ontology levels
3- Basic Formal Ontology (BFO) classes
4- Platform MaterialDigital Core Ontology (PMDco) classes
Tutorial 1: Structure given classes according to PMDco hierarchy (Miro board)
5- Platform MaterialDigital Core Ontology (PMDco) object properties
Tutorial 2: Using appropriate object properties (Miro board)
6- How to develop your application ontologies using PMDco and OBO+ODK best practices?
7- Ontology Development Kit (ODK)
Tutorial 3: Creating ODK repository for PMDco application ontologies (GitHub)
8- Collaborative ontology development workflow, adding taxonomy and axioms
9- PMDco Ontology Design Patterns (ODPs)
Tutorial 4: Ontology editing; adding classes, annotations and axioms (Protege)
10- Ontology evaluation, release, documentation and maintenance

10- Ontology evaluation, release, documentation and maintenance

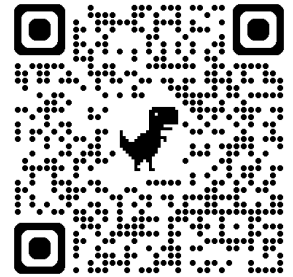
7- Ontology evaluation and testing workflow

- Plan regular meetings with both ontology and MSE domain experts in your project to assess the ontology's structure, resolve modeling issues, and ensure its logical correctness,
- Use ODK evaluation and testing workflows through GitHub Actions Continuous Integration to validate the ontology's structure, syntax, annotation completeness, and logical consistency,
- Use reasoning tools (e.g., reasoners HermiT and Pellet) to ensure logical and technical consistency and identify potential redundancies,
- Analyzes the ontology for common modeling pitfalls, structural issues, and best-practice violations using tools like OOPS! (Ontology Pitfall Scanner!),
- We will be pleased to invite you to present your ontology in one of our "Ontology Playground meetings" (See Section Who We Are & How to Join)

8- Ontology release, updating, and versioning workflow

- Use ODK automated workflow for the release process, which follows these steps: running the release with ODK, reviewing the output, merging changes into the main branch, and creating a GitHub release.
- Track actively the GitHub issue to gather community feedback, recognize bugs, and identify necessary additions and refinements.
- Implement a curation process to assess and prioritize updates based on community needs.
- Employ the robust versioning system hosted on GitHub to ensure clear tracking of changes over time, management of previous versions, and clear tracking of the ontology's evolution over time.

10- Ontology evaluation, release, documentation and maintenance



9- Ontology documentation and publishing

- Create detailed documentation, including best practices, examples, and guidelines in your ontology GitHub repository.
- Use Widoco template, a documentation generator for ontologies that automatically produces human-readable HTML documentation including metadata, hierarchies of classes, properties, and annotations,
- If interested, create MkDocs documentation (like this repository).
- If you are using "application-ontology-template", your ontology is automatically documented in obofoundary.org,
- To enhance your ontology accessibility, please also publish your ontology in MatPortal and PMD DataPortal.

List of Ontologies Reusing PMDco V3.x.x:

- **Logistic Application Ontology (LOG)**
- PMDCo application ontology for logistics and supply chain adopted from iof-supplychain-module
- **Vickers Testing Ontology (VTO)**
- An Ontology for representing the Vickers testing process, testing equipment requirements, test pieces characteristics, and related testing parameters and their measurement procedure according to DIN EN ISO 6507-1 standard.
- **Tensile testing Ontology (TTO)**
- An ontology for representing tensile testing of metals at room temperature in accordance with the associated testing standard ISO 6892-1:2019-11.
- **Heat Treatment Ontology (HTO)**
- An application ontology of PMDco to model heat treatment processes with a focus on metals.

How to Develop Your Application Ontologies Using PMDco and OBO+ODK Best Practices

Workshop content (10 topics, 4 tutorials):

1- Ontology development and beginners learning materials
2- Ontology levels
3- Basic Formal Ontology (BFO) classes
4- Platform MaterialDigital Core Ontology (PMDco) classes
Tutorial 1: Structure given classes according to PMDco hierarchy (Miro board)
5- Platform MaterialDigital Core Ontology (PMDco) object properties
Tutorial 2: Using appropriate object properties (Miro board)
6- How to develop your application ontologies using PMDco and OBO+ODK best practices?
7- Ontology Development Kit (ODK)
Tutorial 3: Creating ODK repository for PMDco application ontologies (GitHub)
8- Collaborative ontology development workflow, adding taxonomy and axioms
9- PMDco Ontology Design Patterns (ODPs)
Tutorial 4: Ontology editing; adding classes, annotations and axioms (Protege)
10- Ontology evaluation, release, documentation and maintenance

Invitation to PMD Ontology Playground Meetings

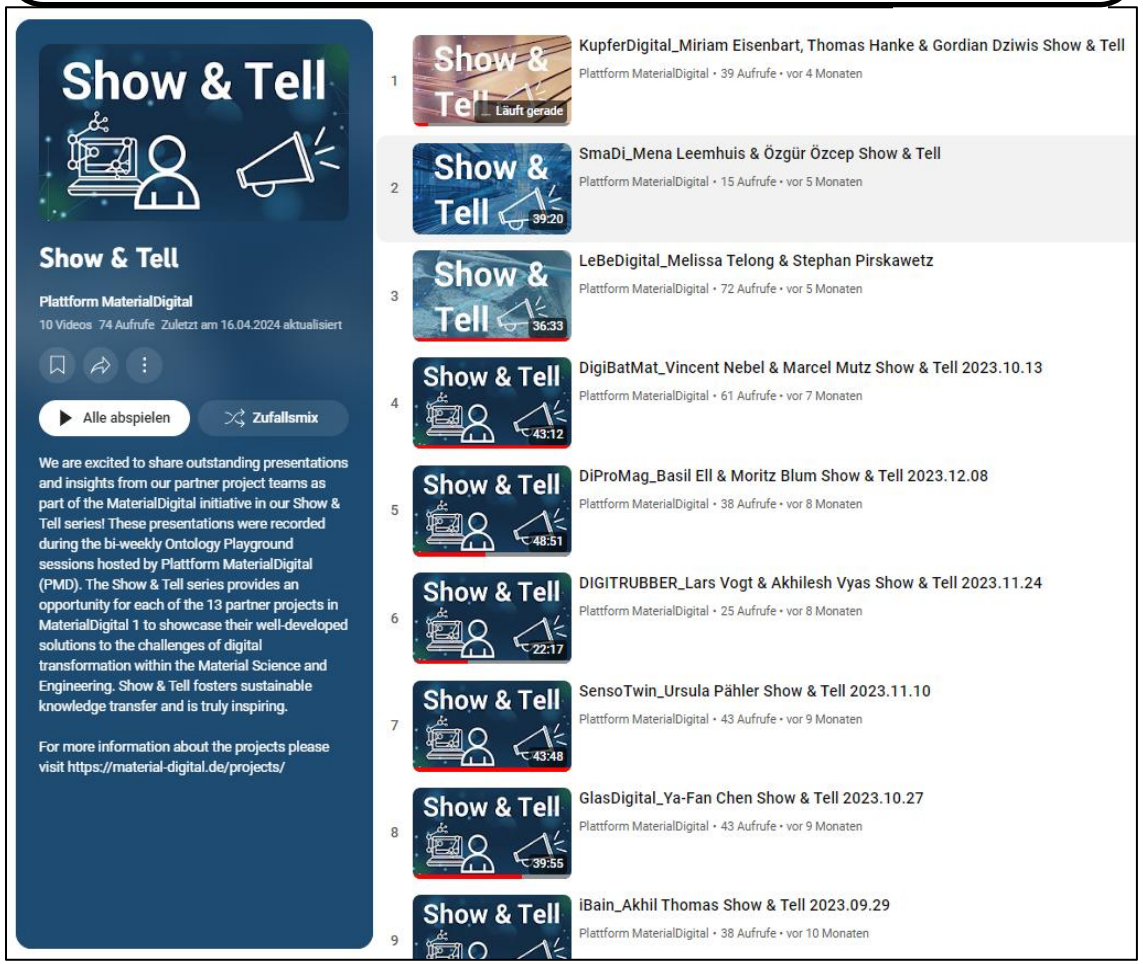
Bi-weekly public sessions
Fridays, 1-2 pm

Audience
Ontology practitioners & MSE domain experts

Key activities
- Knowledge transfer
- Experience exchange
- Modeling challenges – User insights
- Collaborative PMDco enhancement

Participate in our PMD Playground Meetings
Please register via our mailing list.

Check out our YouTube channel



Thank You!

PMDco workshop materials have been prepared by Platform MaterialDigital- Semantic Interoperability Team.

For more information, please contact us via hossein.beygi_nasrabadi@fiz-karlsruhe.de.

The Material Digitalization Platform - a joint project by:

Karlsruher Institut für Technologie
BAM Bundesanstalt für Materialforschung und -prüfung
Max Planck Institute for Sustainable Materials
Fraunhofer IWM
IWT Leibniz-Institut für Werkstofforientierte Technologien
FIZ Karlsruhe Leibniz-Institut für Informationsinfrastruktur
Bundesministerium für Forschung, Technologie und Raumfahrt

Contact us!

info@material-digital.de

www.material-digital.de

The Material Digitalization Platform - a joint project by:

Karlsruher Institut für Technologie

BAM Bundesanstalt für Materialforschung und -prüfung

MAX PLANCK INSTITUTE FOR SUSTAINABLE MATERIALS

Fraunhofer IWM

IWT Leibniz-Institut für Werkstofforientierte Technologien

FIZ Karlsruhe Leibniz-Institut für Informationsinfrastruktur

Bundesministerium für Forschung, Technologie und Raumfahrt