FAIR Data Infrastructure for Condensed-Matter and Chemical Physics

FAIRmat

Application Definition

Nomad lab

Andrea Albino

July 5, 2022

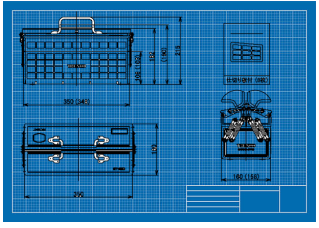
Application Definition

Covering Multiple Fields

Schema and Template concepts

SCHEMA

A formal description of data, data types, and data file structures, such as XML files.



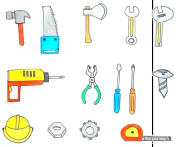
The blueprint of a toolbox

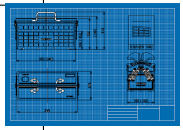
TEMPLATE

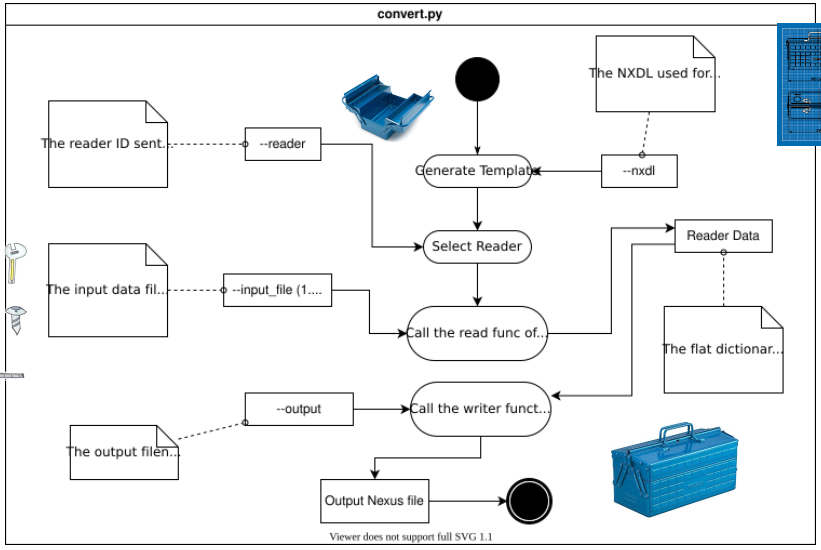
A physical object from which other objects are based or derived.

A toolbox that can be filled with a set of tools fitting in it

Area B Tools






convert.py
The reader ID sent...
--reader
The NXDL used for...
Generate Template
--nxdl
Select Reader
Reader Data
The input data fil...
--input_file (1....
Call the read func of...
The flat dictionar...
--output
Call the writer funct...
The output filen...
Output Nexus file
Viewer does not support full SVG 1.1

Area C, Area B and Area A

We can use pretty much the same tools.
Area B refers to communities using standard schemas (NeXus)
Area A needs to define some “standard” schema for Synthesis.
Area C could contaminate and enrich the Sample standard schema.

A standard enables and enhances exchange!

Even more: standard classes can be made *searchable* in NOMAD

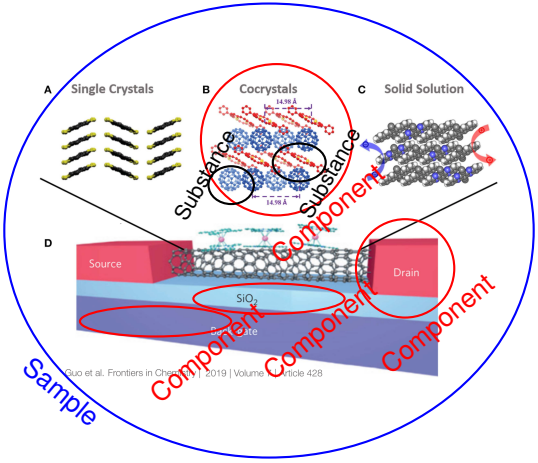
The SAMPLE base class

NXsample:

- exists: [min, 1, max, unbounded]
- sample_id(NXid):
- method: [experimental, simulation, declared by vendor]
- qualification:
- set of measured properties...
- COMPONENT(NXsample):

- The sample tree hierarchy may be visualized in some way inside ELNs to have the whole composition of the sample at a glance

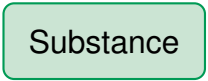
The SAMPLE base class



Implementing the SAMPLE base class

General Info

Qualifying Info



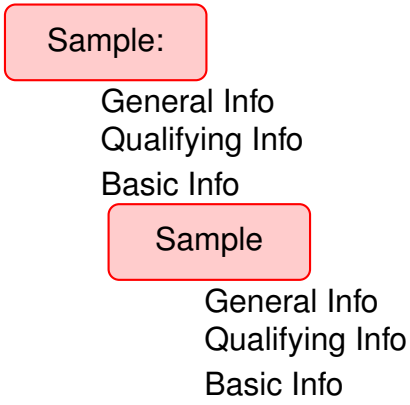
Basic Info

ID
name
user
reference_doi
method: [exp, sim, declared]
properties ...

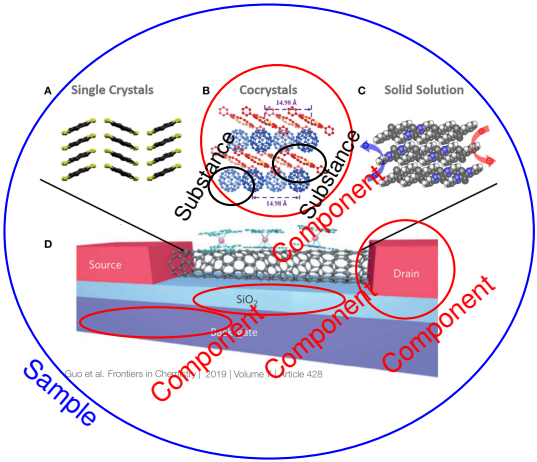
geometry
physical state
role_in_experiment
amount or conc. (extension of Component class)
properties ...

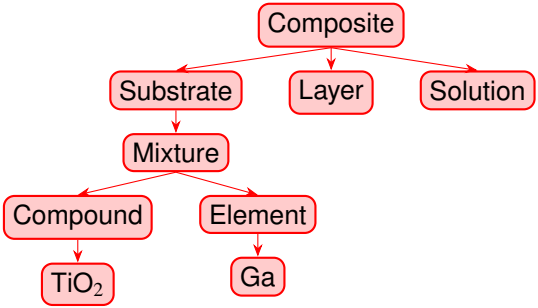
INCHI
CAS
IUPAC_name
molecular_mass
concentration
amount or conc. (extension of Substance class)
properties ...

Implementing the SAMPLE base class



The SAMPLE base class





Base classes

- Sample:
  - Component
    - Substance
- Instrument
  - Source
  - Monochromator
  - Detector
- Process
  - Manually Filled
    - Time Dependent
    - Time Independent
  - Machine Filled
    - Time Dependent
    - Time Independent
- User
  - Name
  - Affiliation
  - Email
- Data
  - Axes
  - Scaling Factor
  - Plot

Composing an Application Definition

How to combine these entities to compose our application definition?

Data | Instrument | Sample | Process | User

Multiple concatenations can be envisioned.

We need to instantiate a "master class" that represents our application definition containing the already mentioned base classes:

Synthesis Experiment

Simulation Experiment

Measurement Experiment

Composing an Application Definition

Data Instrument Sample Process User

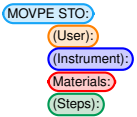
Measurement
Simulation
Synthesis

```
my_experiment:
    operator(NXuser):
    NXinstrument
    (NXprocess)
        NXsynthesis_step:
            parent(NXsample):
            child(NXsample):
            parameters(NXdata):
        post processing (NXprocess)
        Characterization(NXentry):
        plot(NXdata)
```

```
my_experiment:
    operator(NXuser):
    NXinstrument
    raw_materials(NXsample):
    (NXprocess)
        NXsynthesis_step:
            parameters(NXdata):
            obtained_sample(NXsample):
        post processing (NXprocess)
        Characterization(NXentry):
        plot(NXdata)
```

```
my_experiment:
    operator(NXuser):
    NXinstrument
    NXsample:
        my_synthesis(NXsynthesis):
            (NXprocess)
                precursor(NXsample):
                NXsynthesis_step:
                    parameters(NXdata):
            post processing (NXprocess)
    plot(NXdata)
```

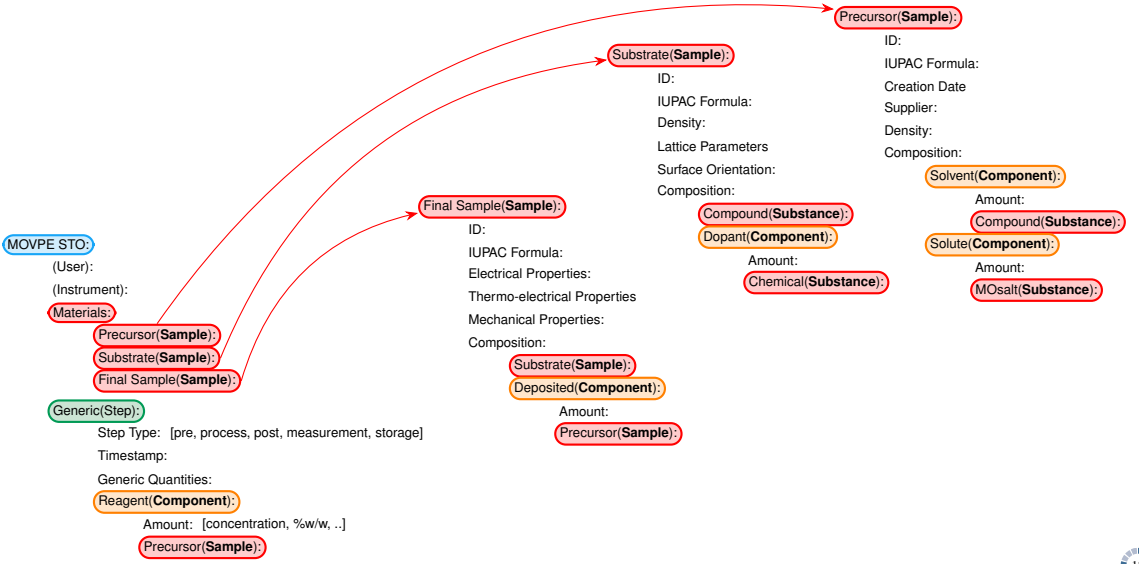
Linking Multiple Archive Files



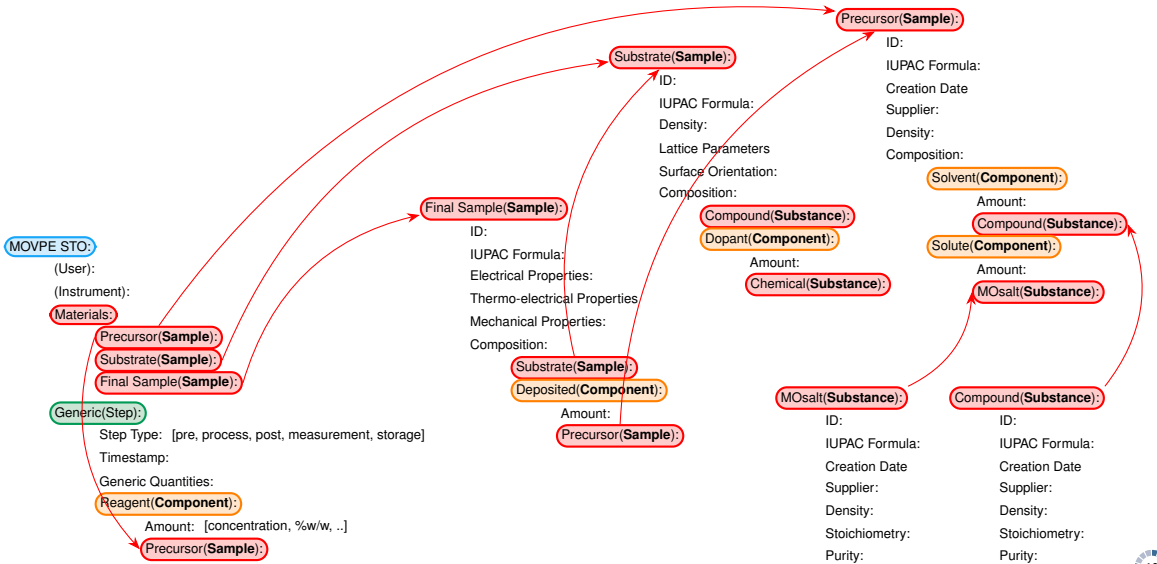
Linking Multiple Archive Files

MOVPE STO:
- (User):
- (Instrument):
- Materials:
  - Precursor(**Sample**):
  - Substrate(**Sample**):
  - Final Sample(**Sample**):
- Generic(Step):
  - Step Type: [pre, process, post, measurement, storage]
  - Timestamp:
  - Generic Quantities:
  - Reagent(**Component**):
    - Amount: [concentration, %w/w, ..]
    - Precursor(**Sample**):

Linking Multiple Archive Files



Linking Multiple Archive Files



Covering Multiple Fields

Linking Multiple Archive Files

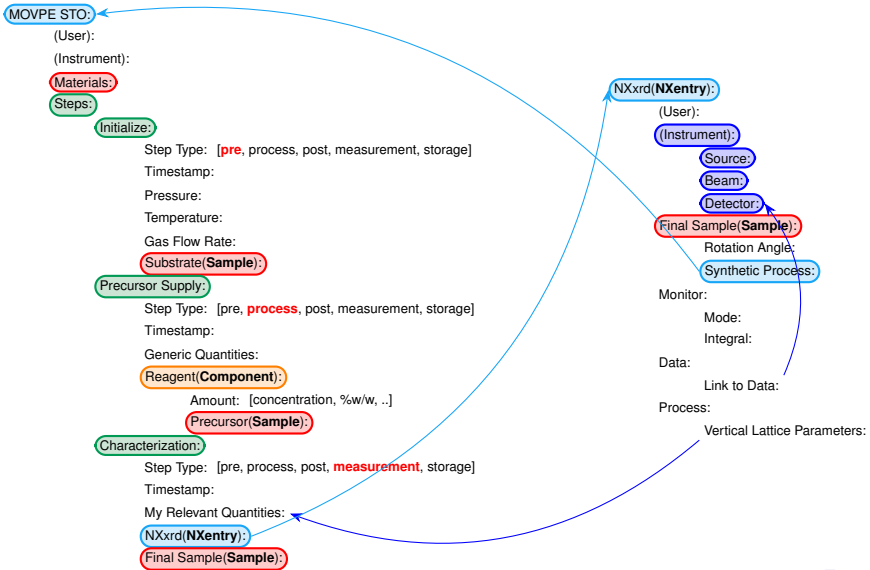
MOVPE STO:
(User):
(Instrument):
Materials:
Steps:
Initialize:
Step Type: [**pre**, process, post, measurement, storage]
Timestamp:
Pressure:
Temperature:
Gas Flow Rate:
Substrate(**Sample**):
Precursor Supply:
Step Type: [pre, **process**, post, measurement, storage]
Timestamp:
Generic Quantities:
Reagent(**Component**):
Amount: [concentration, %w/w, ..]
Precursor(**Sample**):
Characterization:
Step Type: [pre, process, post, **measurement**, storage]
Timestamp:
My Relevant Quantities:
NXxrd(**NXentry**):
Final Sample(**Sample**):

Linking Multiple Archive Files

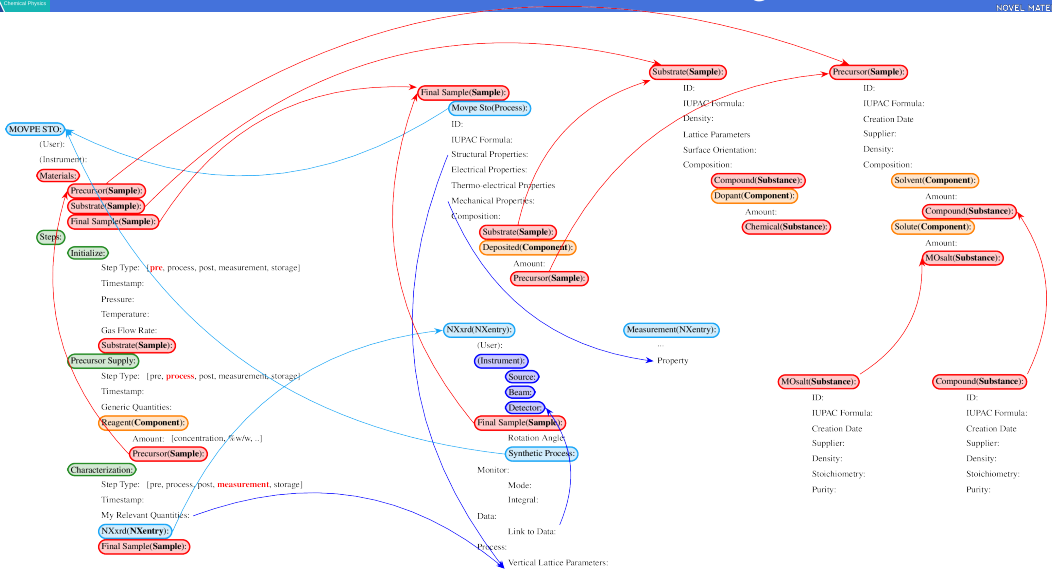
MOVPE STO:
(User):
(Instrument):
Materials:
Steps:
Initialize:
Step Type: [**pre**, process, post, measurement, storage]
Timestamp:
Pressure:
Temperature:
Gas Flow Rate:
Substrate(**Sample**):
Precursor Supply:
Step Type: [pre, **process**, post, measurement, storage]
Timestamp:
Generic Quantities:
Reagent(**Component**):
Amount: [concentration, %w/w, ..]
Precursor(**Sample**):
Characterization:
Step Type: [pre, process, post, **measurement**, storage]
Timestamp:
My Relevant Quantities:
NXxrd(**NXentry**):
Final Sample(**Sample**):

NXxrd(**NXentry**):
(User):
(Instrument):
Source:
Beam:
Detector:
Final Sample(**Sample**):
Rotation Angle:
Synthetic Process:
Monitor:
Mode:
Integral:
Data:
Link to Data:
Process:
Vertical Lattice Parameters:

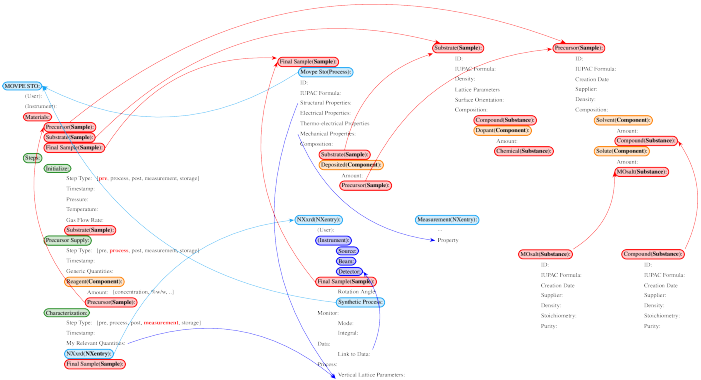
Linking Multiple Archive Files



Redundant Linking Structure

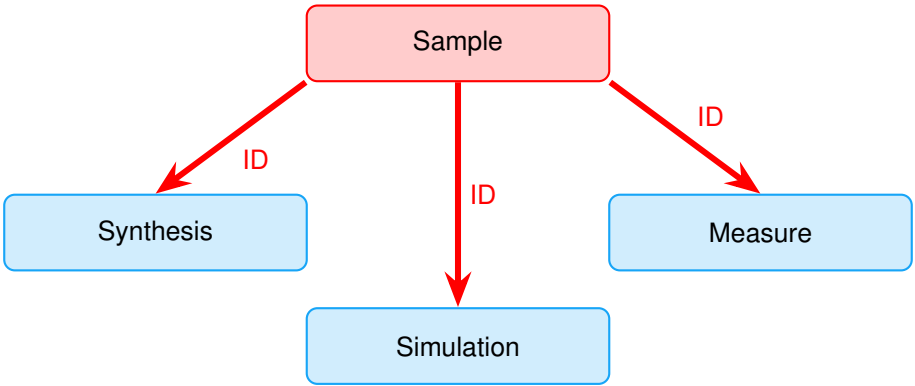


Nearest Neighbours



Covering Multiple Fields

Connect the parts and the whole



Collecting all these use cases and looking at their schemas could lead us to generalize it!