Traceability

Master of Applied IT – Semester 2

Date	27-05-2024
Version	0.1
Student name	Lars van den Brandt
Student number	434565
Teachers	Nico Kuijpers (1st) & Casper Schellekens (2nd)

Contents

Context

NOMAD's purpose is to facilitate FAIR data management with good metadata handling for each experiment. Ensuring equipment traceability is vital for reproducibility. Logging all steps, including equipment usage, ensures experiment replication. Researched by prototyping a proof of concept of NOMAD Oasis, testing RobotLab's use cases to clarify its standardization and metadata handling features. When an entry is created, a parser is selected to parse and format the data.

1. How does NOMAD handle metadata and ensure consistency in data representation?
2. How does NOMAD handle traceability of used equipment?

Schema

Schemas in NOMAD (.archive.yaml format) are used to define the structure of data entries. They specify what combinations of primitive values, objects, and lists are allowed within the data.

Sections

Sections represent different parts of the data structure. They are like chapters or sections in a book. Each section has defined keys and possible values.

```yaml
sections:
    Chemical:
      description: |
        This is an example description for Chemical.
      quantities:
        cas_number:
          type: str
          m_annotations:
            eln:
              component: StringEditQuantity
        ec_number:
          type: str
          m_annotations:
            eln:
              component: StringEditQuantity
```

Code 1: Sections code snippet

In this code snippet, a section example is created. A section can always contain a description, and define different quantities.

Quantities

Quantities define the properties or attributes within a section. They specify the type of values (e.g., string, float), shape (e.g., scalar, list), and units (physical meaning) for each property.

List of supported quantity types:

type	description
string	-
str	-
float	-
integer	-
int	-
boolean	-
bool	-
np.int32	Numpy based integer with 32 bits.

np.int64	Numpy based integer with 64 bits.
np.float32	Numpy based float with 32 bits.
np.float64	Numpy based float with 64 bits.
Datetime	-
User	A type for NOMAD users as values.
{type_kind: Enum, type_data: []}	Use type_data to specify enum values as list of strings.
-	To define a quantity that is a reference to a specific section.

Table 1: Quantities supported types

```
Chemical:
  description: Details about the chemical used in the experiment.
  quantities:
    chemical_name:
      type: string
      description: The name of the chemical.
    molecular_weight:
      type: np.float64
      description: The molecular weight of the chemical.
      unit: g/mol
    purity:
      type: float
      description: The purity percentage of the chemical.
      unit: %
```

Code 2: Quantities code snippet

In this code snippet, the section Chemical is created, it contains 3 quantities which define the properties within the section.

Sub-sections

Sub-sections define relationships between sections, indicating that one section is part of another. They can be repeating to allow multiple instances of the same type within a section.

```
Experiment:
  description: Information about the experiment.
  quantities:
    experiment_id:
      type: string
      description: Unique identifier for the experiment.
    date:
      type: Datetime
      description: The date when the experiment was conducted.
  sub_sections:
    chemical_used:
      type: Chemical
      repeats: true
      description: List of chemicals used in the experiment.
    instrument_used:
      type: Instrument
      repeats: true
      description: List of instruments used in the experiment.
```

Code 3: Sub-sections code snippet

This code snippet contains a section definition for an experiment. It defines quantities and two different sub-sections. These sub-sections have a different type than the possible pre-defined values like a String or a Float. The sub-section chemical_used contains the type Chemical, which refers to different sections within the schema, the quantities and values of chemical is inherited.

Base sections

Base sections allow for specialization and inheritance in schema definitions. This enables creating more specialized definitions based on abstract ones.

References establish connections between different sections or even between different entries' archives. They can be uni-directional links between sections or reference to section definitions.

NOMAD provides built-in section definitions for common use cases like entry metadata, data, etc. These predefined sections facilitate consistent data modelling.

```yaml
sections:
   Chemical:
     base_sections:
       - 'nomad.datamodel.metainfo.eln.Chemical'
       - 'nomad.datamodel.data.EntryData'
     description: |
       This is an example description for Chemical.
       A description can contain **markdown** markup and TeX formulas
     quantities:
       form:
         type:
           type_kind: Enum
           type_data:
             - crystalline solid
             - powder
             - liquid
```

Code 4: Base-sections code snippet

This code snippet contains a section called Chemical, which contains two base_sections. These pre-defined base_sections contain pre-defined quantities which are inherited.

Base sections can be browsed on the central nomad: https://nomad.warwick.ac.uk/nomad-oasis/gui/analyze/metainfo/nomad.datamodel.datamodel.EntryArchive

All together

NOMAD created a UML with the data structure, to see the references between each definition.

- Packages contains one or more sections.
- Section definition contains definitions for subsections and quantities. Sections can inherit the properties of other sections (base_sections).
- Subsections allow for hierarchical definitions for other sections.
- Quantities can use section definitions (or other quantity definitions) as a type to define references.

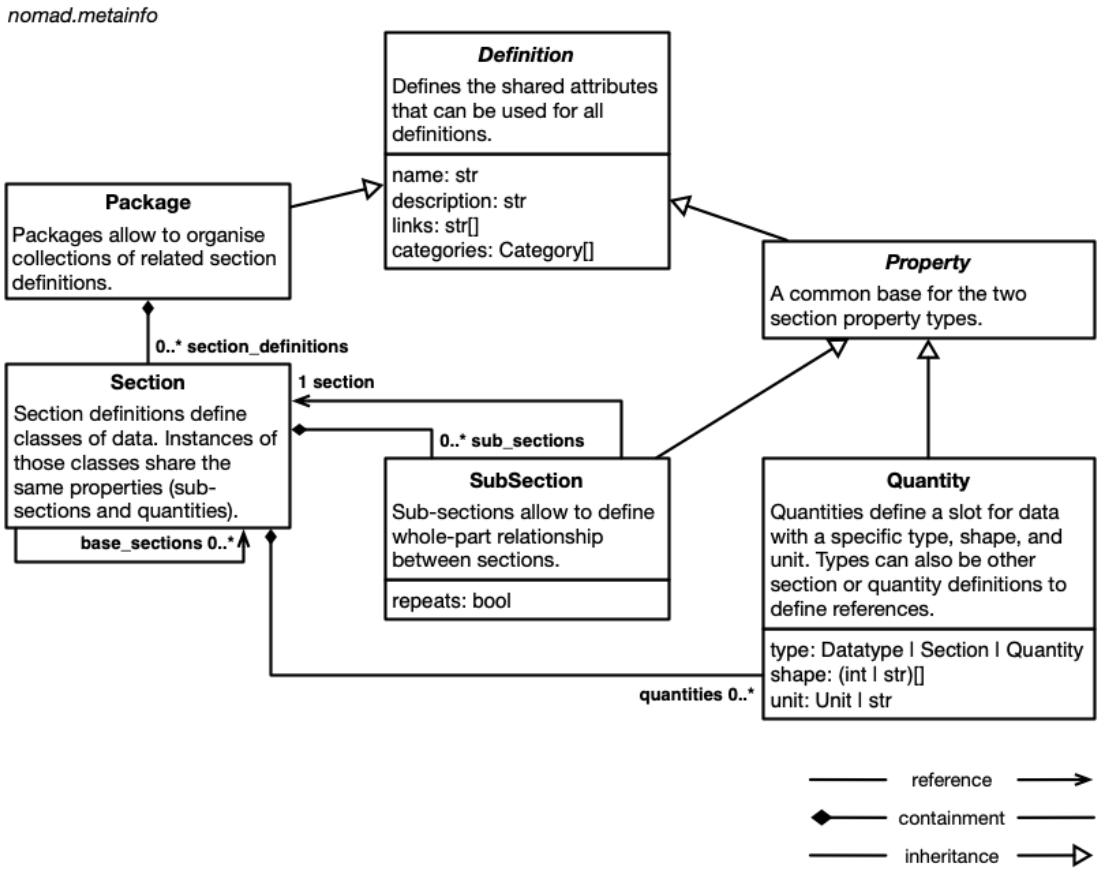

Figure 1: UML schema definitions

Designing a schema

Example schema

The following schema example is designed for testing and gaining knowledge about schema creation in the context of electronic lab notebooks (ELNs). It is based on the tutorials provided in the NOMAD documentation, which offer guidance on how to create such schemas.

This schema includes some additional elements to further illustrate its structure and functionality and help with exploring all schema features NOMAD has to offer. The schema defines several sections, including Chemical, Sample, Process, and Instrument, each with its own set of quantities and annotations. By understanding and utilizing this example, users can gain insights into effectively organizing and managing data within an ELN framework.

This schema is saved in the Git repository for NOMAD-RobotLab: https://git.fhict.nl/coe-htsm/nomad-oasis/-/tree/main/nomad-develop/schemas/Example%20ELN%20-%20mock%20data

Schema definition

Sections:

- Chemical
- Sample
- Process
- Instrument

Quantities:

- Sample has various quantities, including name, tags, chemicals, substrate_type, substrate_thickness, and sample_is_from_collaboration.
- Chemical has various quantities: form, cas_numbr and ec_number.
- Process has an instrument quantity referencing the Instrument section.
- pvd_evaporation, hotplate_annealing, and plate_reader_setup define specific quantities relevant to those processes.

Base Sections:

- Chemical inherits from nomad.datamodel.metainfo.eln.Chemical and nomad.datamodel.data.EntryData.
- Instrument inherits from nomad.datamodel.metainfo.eln.Instrument and nomad.datamodel.data.EntryData.
- Process inherits from nomad.datamodel.metainfo.eln.Process.
- Sample inherits from nomad.datamodel.metainfo.eln.Sample and nomad.datamodel.data.EntryData.
- pvd_evaporation, hotplate_annealing, and plate_reader_setup inherit from Process.

Subsections:

- Sample contains the subsection processes.
- processes contains the subsections pvd_evaporation, hotplate_annealing, and plate_reader_setup.

Unified Modelling Language (UML)

The UML class diagram is created using draw.io. The goal is to visually represents the schema definitions and their relationships as specified in the YAML schema file.

The diagram clearly illustrates the hierarchical structure of the sections Chemical, Instrument, Process, and Sample, along with their associated properties (quantities) and inheritance (base sections).
The Sample section contains a processes subsection, which further breaks down into detailed process sections like pvd_evaporation, hotplate_annealing, and plate_reader_setup.

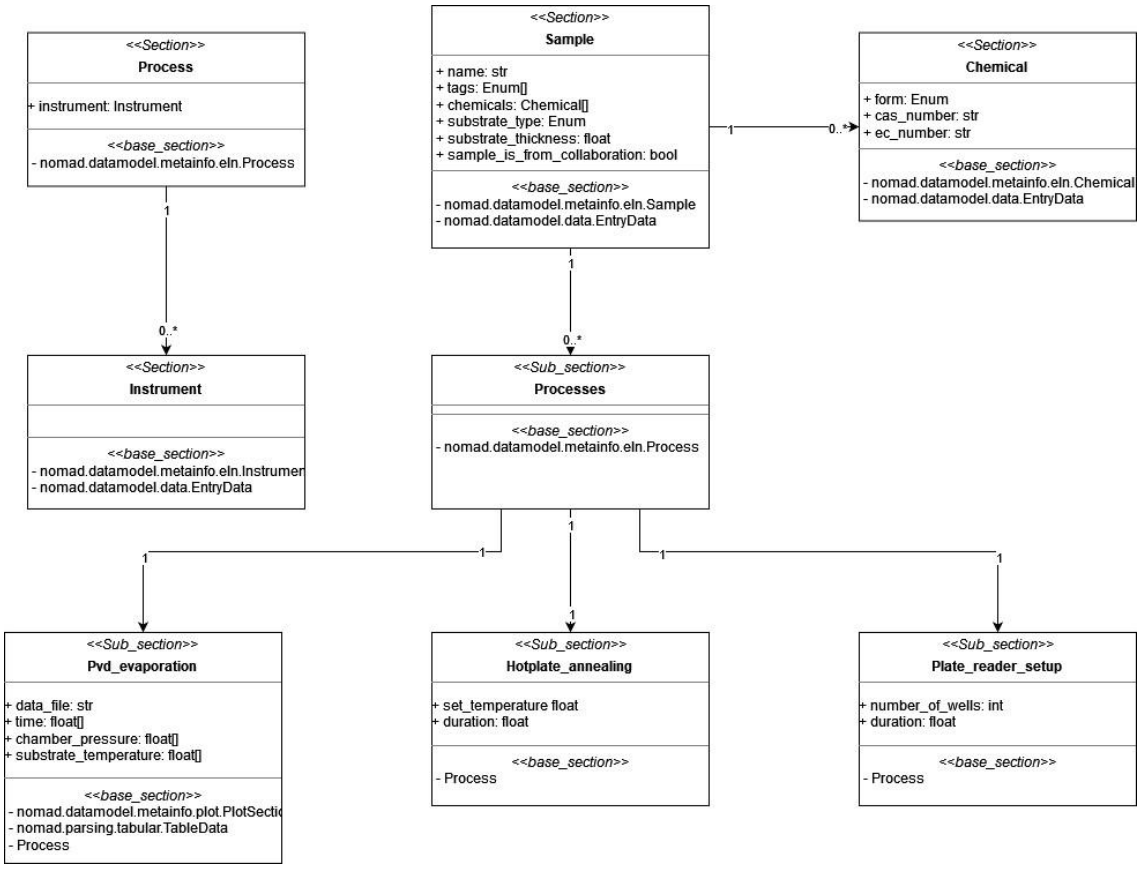

Figure 2: UML example schema

Use case TU/e Fluorescence experiment

This schema is saved in the Git repository for NOMAD-RobotLab: https://git.fhict.nl/coe-htsm/nomad-oasis/-/tree/main/nomad-develop/schemas/Example%20ELN%20-%20TU

Questionnaire for schema

To create the schema for the TU/e use case, all definitions have to be gathered. This questionnaire was discussed with Joost van Toll.

- **Chemical Section:**
  - *What types of chemicals will be involved in the experiment?*
    The chemicals will involve two reactants called 'X' (X) and 'X2' (X). Furthermore, demineralized water and salt (sodium chloride) will be added to the reaction mixture. The product that will be created up reaction between the two reactants is 'X' (X).
  - *What are the typical properties or attributes of these chemicals that need to be recorded?*
    The typical property of the product 'X' that requires recording is: fluorescence.

- **Instrument Section:**
  - *What instruments or equipment will be used in the experiment?*
    A plate reader that is able to measure 96 well plates. The specific type is: the Tecan Infinite M+ Nano.
  - *What information about each instrument is important to capture?*
    Important information about the instrument is that it captures fluorescence (280 -850 nm) and absorbance (230 – 1000 nm). Plate format from 6 – 384 wells. Optics: monochromator

- **Process Section:**
  - *What processes or procedures will be carried out during the experiment?*
    The aqueous samples will be prepared by pipetting all required components (reactants and salt) into a 96-well plate. The well-plate will be heated at a specific temperature and shaken for a specific amount of time. After a specific time point, the well-plate is cooled down to room temperature and will be transferred into the plate reader. The fluorescence will be measured of all filled wells. From the fluorescence, the yield will be calculated and noted in a notebook.
  - *What instruments or equipment will be involved in each process?*
    For pipetting: 50-1000 µL pipettes will be used. Well-plate: Nunclon "microwell" 96-Well, flat-bottom, black, delta-treated. For fluorescence measurement: the Tecan Infinite M+ Nano.

- **Sample Section:**
  - *What types of samples will be used in the experiment?*
    Aqueous samples will be used comprising reactants, demineralized water and salt. Samples will be distributed over wells.
  - *What are the key characteristics or attributes of each sample that need to be documented?*
    Key characteristics (or conditions) of each sample that should be reported: concentration of reactants and products, concentration of salt, pH of the solution, reaction time, reaction temperature. And as a result the fluorescence and yield of the reaction.
- **Specific Quantity Details:**
  - *For each section, what specific quantities or properties need to be documented?*
    I think I just answered this question in question 2 of 'sample section'.
- **Data Entry and Editing Components:**
  - *Are there any specific requirements for data formatting or units?*
    Yield is in percentage and fluorescence is expressed in relative fluorescence units or RFU (compared to reference).
- **Sub-Sections and Relationships:**
  - *Are there any hierarchical relationships between different sections or quantities?*
    From the RFU obtained for each sample and the maximum achievable RFU with full conversion, the yield is calculated. Thus, RFUsampel/RFUmax*100% = yield.
  - *Are there any sub-sections within a section that need to be defined?*
    Not that I know at this point.
- **Visualization Requirements:**
  - *Are there any specific visualization requirements for certain quantities or processes?*
    The yield should be plotted as a function of all the variables (concentration reactants, salt, pH, reaction time, temperature, etc.). Furthermore, if possible, a 2D/3D phase diagram with two or three variables on the axis could be plotted with fluorescence as the read-out.
  - Do any quantities need to be plotted over time or against each other?
    See previous answer in this section.
- **Any Additional Requirements or Preferences:**
  - *Are there any additional features, requirements, or preferences that should be considered when defining the schema and creating the example upload?*
    Not sure if there are any additional features that we would like besides that the results should be represented in an automatically generated csv file with all the possible results and data variables. This csv-file should be readable by python… Not sure how NOMAD can help with this process though…

Schema definition

Sections:

- Experiment
- Chemical
- Sample
- Process
- Instrument

Quantities:

- **Experiment**
  Name, Experiment Description, Experiment Visual and Sample
- **Sample**
  Name, tags and chemicals.
- **Chemical**
  Form and fluorescence.
- **Process**
  Instrument quantity referencing the Instrument section.
- **Pipetting**
  Size_of_pippet, number_of_wells, heated_to_temperature, time_shaken
- **Fluorescence_measurement**
  Plotted quantities

Base Sections:

- Experiment inherits from nomad.datamodel.data.EntryData.
- Chemical inherits from nomad.datamodel.metainfo.eln.Chemical and nomad.datamodel.data.EntryData.
- Instrument inherits from nomad.datamodel.metainfo.eln.Instrument and nomad.datamodel.data.EntryData.
- Process inherits from nomad.datamodel.metainfo.eln.Process.
- Sample inherits from nomad.datamodel.metainfo.eln.Sample and nomad.datamodel.data.EntryData.
- Pipetting and fluorescence_measurement inherit from Process.

Subsections:

- Sample contains the subsection processes.
- Processes contains the subsections pipetting and fluorescence measurement

Unified Modelling Language (UML)

The UML class diagram is created using draw.io. The goal is to visually represents the schema definitions and their relationships as specified in the YAML schema file.

The diagram clearly illustrates the hierarchical structure of the sections Chemical, Instrument, Process, and Sample, along with their associated properties (quantities) and inheritance (base sections).
The Sample section contains a processes subsection, which further breaks down into detailed process sections Fluorescence_measurement and pipetting. Both these processes contain the sub-section process, referring to the defined process with its reference to the instrument.

Each of these sections and subsections is annotated with relevant quantities, types, and their corresponding annotations, showcasing a well-organized and comprehensive schema design. This visual aid helps to better understand the complex relationships and data flow within the schema.

Figure 3: UML TU/e Fluorescence experiment

Conclusion

NOMAD facilitates consistent data representation by utilizing schemas that define the structure of data entries. These schemas, written in .archive.yaml format, specify allowed combinations of values, objects, and lists. By referencing base sections and sections within these schemas, NOMAD ensures uniformity across different datasets.

For instance, in our schema designed for an Electronic Lab Notebook (ELN) for a fluorescence experiment, the "Chemical" section references the base section nomad.datamodel.metainfo.eln.Chemical. This reference inherrits additional properties from the NOMAD Library, enabling consistent data saving. Consequently, users can run queries across all data entries, retrieving only ELNs containing specific elements. This approach ensures that metadata handling and data representation remain consistent and reliable.

NOMAD manages equipment traceability similarly by defining schemas that reference relevant base sections. In the ELN for the fluorescence experiment, the "Instrument" section refers to the base section nomad.datamodel.metainfo.eln.Instrument. Specific instruments, are linked to processes. This again allows users to later search for all experiments involving the Tecan instrument through targeted queries. Thus, by saving data with well-defined schemas, NOMAD ensures comprehensive traceability of the equipment used in experiments.

By implementing these schemas we make data FAIR (**F**indable, **A**ccessible, **I**nteroperable, **R**eusable):

- **Findable**
  The inclusion of sufficient metadata and unique identifiers ensures that data entries are easily searchable within NOMAD's repositories. For example, consistent data savage allows users to run precise queries, making data findable.
- **Accessible**
  By adhering to the structured schema format, both metadata and data remain accessible to humans and machines.
- **Interoperable**
  The common structure enforced by the schema ensures that data entries are interoperable. This standardization allows seamless integration and sharing of data across different platforms and research domains.
- **Reusable**
  Clearly defined data is reusable. The detailed documentation of the processes within the schema creates reusability for further research.

The UML class diagram created using draw.io serves as a visual representation of the schema definitions and their relationships. This diagram helps illustrate the hierarchical structure of sections such as Chemical, Instrument, Process, and Sample, along with their quantities and inheritance. By visualizing these references, it becomes easier to understand the complex relationships and data flow within the schema.

In conclusion, consistent data traceability can be achieved by defining a schema for each experiment. This methodical approach ensures that all aspects of an experiment, from chemicals and instruments to processes and samples, are documented and linked. As a result, data becomes more reliable, reproducible, and aligned with FAIR principles.