Evaluating NOMAD Oasis to manage data for a self-driving laboratory

Lars van den Brandt
Fontys ICT
lars.vandenbrandt@student.fontys.nl

Nico Kuijpers
Fontys ICT
nico.kuijpers@fontys.nl

Teade Punter
Fontys ICT
teade.punter@fontys.nl

1 Abstract

Self-driving chemical laboratories seek to transform the chemical industry and research to an automated, AI-driven laboratory capable of independently designing and executing experiments. Automated laboratories generate substantial and complex data, necessitating an advanced data management framework. NOMAD, specifically its customizable extension NOMAD Oasis, is proposed as a solution. This research evaluates to what extend NOMAD Oasis can handle the data management needs of an automated chemical laboratory, focusing on data access control, traceability, and the management of diverse and voluminous data. Enhanced access control models were developed to meet specific requirements, ensuring secure and efficient data collaboration. Traceability was achieved through detailed schema definitions, facilitating reproducibility and data integrity. Practical tests confirmed NOMAD's scalability and adaptability to various data formats. Despite some challenges regarding documentation, NOMAD Oasis is a robust and flexible data management platform suitable for an academic self-driving laboratory of the future.

***Keywords*:** Self-driving lab, RobotLab, data management frameworks, artificial intelligence, data volume, various data, data access, data traceability, data integration, NOMAD, NOMAD Oasis.

2 Introduction

An automated laboratory that can design and conduct experiments by itself is envisioned by the RobotLab project. The RobotLab, as part of The Big Chemistry consortium [1, 2] aspires to improve efficiency in the current landscape of high throughput chemical research, specifically supramolecular and inorganic chemistry [2], by combining automated machinery with artificial intelligence (AI) [3]. This self-driving lab autonomously executes experiments and analyzes the results, a cycle known as closing the loop. The RobotLab specifically includes geographically dispersed execution of experiments, additionally, the lab results will be available to various parties [2].

One of the challenges is to manage large amounts of data generated by AI-driven experiments. The RobotLab requires a robust data management framework, capable of effectively handling the data generated by these experiments. A possible solution is NOMAD [4], a data management framework designed to handle the complexities of scientific data effectively [4]. Preliminary investigations within the RobotLab suggested NOMAD as a potential candidate, offering a platform capable of managing and organizing the data generated by robotic laboratory experiments.

NOMAD, which stands for Novel Materials Discovery [4], is an online infrastructure designed to manage and share scientific data efficiently. It offers a unified solution for the **F**indability, **A**ccessibility, **I**nteroperability, and **R**eusable (FAIR) [5] of data, crucial principles in modern scientific research [6]. NOMAD achieves this by processing files to extract structured data and rich metadata, enabling seamless access of scientific information.

NOMAD is open source [7] and offers the ability to create a local NOMAD instance; NOMAD Oasis [8]. NOMAD Oasis represents a customizable extension, allowing organizations to tailor the platform to their specific needs. This flexibility offers advantages for projects like RobotLab, which require specialized data management solutions to manage the unique characteristics of each experiment.

The primary goal of this research is to evaluate whether NOMAD, particularly its customizable extension NOMAD Oasis, is indeed a suitable data management platform for RobotLab. This research aims to assess the capabilities of NOMAD Oasis in handling the extensive and complex data generated by RobotLab's AI-driven experiments, and to determine its suitability in meeting the specific data management needs of an automated laboratory environment.

3 Methods

In assessing the compatibility of NOMAD, particularly NOMAD Oasis, with the data requirements of the RobotLab project, a structured methodology was employed. The research methodology consisted of several key steps, systematically addressing three critical aspects of the data management needs of the RobotLab project: ensuring data access control, maintaining traceability of experimental data and managing various and voluminous data. By systematically following this methodology, a comprehensive understanding of NOMAD's suitability for a self-driving lab like the RobotLab project is achieved. This helped with making an informed decision and supported potential enhancements to the NOMAD framework to better align with project's needs.

3.1 Access control

The necessity for robust data access control mechanisms arises from the varied stakeholders involved in the RobotLab project. This aspect was selected due to the need to ensure that data is securely accessible only to authorized users, especially considering that the RobotLab plans to rent out its facilities to external commercial entities such as Nouryon [9], which will use and generate data. All these external entities have diverse needs for accessing project data. This paper contains the key findings of the research to access control. More information can be found in [10].
The following steps were taken to research access control:

- **Literature Review**: A review of relevant literature to possible data access control methods and policies, to find the best method and policy to manage the RobotLab's data access [10].
- **Stakeholder Interviews**: Conducted interviews with stakeholders from the RobotLab project. The purpose was to understand the specific needs for data access for the RobotLab.
- **Assessment of Native Capabilities**: Investigated NOMAD's native support for data access control functionalities. This included reviewing the platform's documentation and code and performing initial tests to identify existing capabilities and potential gaps [10].
- **Script proof of concept**: Before modifying the code, several scripts were created. These scripts could run as a service before NOMAD [11].
- **Codebase Modifications**: Upon identifying gaps in NOMAD's capabilities, modifications were made to the NOMAD codebase. This involved enhancing the platform's data access control features to better suit the specific needs of the RobotLab [12].
- **Validation Tests**: Conducted tests to verify the effectiveness of these modifications. This included setting up various user roles and permissions and ensuring that data access was appropriately restricted based on these settings [10].

3.2 Traceability

Traceability of experimental data is crucial for reproducibility and data integrity, aligning with NOMAD's goal of making data FAIR. This aspect was selected based on input from field experts who highlighted the importance of traceability of data for the reproducibility of data and experiments. This paper contains the key findings of the research to traceability. More information can be found in [13].
The following steps were taken to research traceability:

- **Stakeholder Interviews:** Conducted interviews with stakeholders and academic partners of the RobotLab. The purpose was to create a structure for use case of a fluorescence experiment. Collaborated with experts to captures all necessary metadata for traceability of this experiment.
- **Schema Design**: Designing a schema based on the use case. This schema included details such as experimental conditions, equipment used, and procedural steps [13].
- **Prototyping**: Developed the schema for the fluorescence experiment, serving as a prototype implementation of the traceability schema within NOMAD [14].
- **Practical Testing**: Used data for the specific case involving fluorescence experiments from academic partners to test the traceability functionality, ensuring that the data could be accurately traced back to its source.

3.3 Various and voluminous data

Given that the RobotLab is designed to handle a high throughput of experiments in Supramolecular and Inorganic chemistry [2], which will generate large and diverse datasets, an important step involved is a comprehensive analysis of the data generated by these experiments. This aspect was chosen because the self-driving nature of the lab ensures a continuous loop of experiments [3], producing significant amounts of input and output data of varying sizes and types. This paper contains the key findings of the research to various and voluminous data. More information can be found in [15].
The following steps were taken to research various and voluminous data:

- **Stakeholder Interviews**: Conducted interviews with stakeholders and academic partners of the RobotLab. The purpose was to gather insights into the types and volumes of data generated by robotic laboratory experiments.
- **Literature Review**: A review of relevant literature and similar use cases to understand the data characteristics and challenges. Raw data was gathered from the supplementary data of this literature and use cases [15].
- **Practical Testing**: Uploaded various data sets to the NOMAD Oasis instances. These datasets were representative of the typical data generated by RobotLab experiments, including diverse file formats and large volumes. The platform's performance in managing, storing, and retrieving these data sets was analysed [15].

4 Results

The following sections contain the findings of the investigation into three critical areas for the project's success: access control, traceability, and management of various and voluminous data. By addressing these areas, this research aims to ensure secure, efficient, and reproducible data handling that aligns with the project's objectives and the FAIR principles.

4.1 Access control

Criteria for accessing data within the RobotLab project is identified through stakeholder interviews [10] and literature review [16, 2]. The criteria include the following:

1. **Multiple, dynamic user roles:** The model should seamlessly accommodate multiple access roles of users, allowing access controls to be deduced from their respective roles.
2. **Collaboration Rights**: Different stakeholders often need to collaborate on experiments and share data. The system should facilitate such collaboration by providing access control that allows users to share specific datasets with collaborators while keeping other data private. This is essential for enabling teamwork without compromising sensitive information.
3. **Ease of Specification**: The access control system must be user-friendly, allowing administrators to easily define and modify access policies. This reduces the administrative burden and ensures that access controls can be quickly adapted to changing project requirements.
4. **Efficient Storage**: Access control mechanisms should not significantly impact the storage performance of the data management system. Efficient storage ensures that the system remains scalable and responsive, even as the volume of data grows.
5. **Automation**: Automating access control processes is crucial for maintaining up-to-date security policies without requiring constant manual intervention.

NOMAD offers mechanisms to restrict access to the entire platform through whitelisting [17]. Through testing, it is confirmed that the whitelist mechanism functions as intended, effectively restricting access to un-authorized users [10]. This security measure provides reassurance that unauthorized individuals cannot breach the Oasis's defenses when the whitelist is implemented. However, while the whitelist mechanism proves to be a valuable security enhancement, it falls short of meeting the specific data access management needs of the RobotLab project. Even though it restricts access to unauthorized users, once a user gets access, they have access to all data within the Oasis. This solution does not provide the access control required by the RobotLab project, because it does not allow for the control over access by individuals to specific data or resources.

Research of literature identified three main classes of access control policies: discretionary (DAC), mandatory (MAC), and role-based (RBAC) [18, 10]. While each policy has its advantages and challenges, criteria review and interviews with stakeholders showed that none of the models fully meet the specific access criteria of the RobotLab project.

- **Mandatory (MAC) – Multileve**l: Mandatory policies focus on fixed security levels and categories. They lack the flexibility to accommodate dynamic user roles and the changing access needs typical in collaborative environments like the RobotLab (Criteria 1).
  Additionally, defining access based on security levels and categories is complex and does not align with the users understanding of the roles (Criteria 3).
- **Role-based (RBAC):** While roles can simplify specification compared to individual permissions, defining and updating roles to accurately reflect the needs of all users and projects can become complex (Criteria 3). Automating is possible but requires mechanisms to handle the dynamic roles of users, maintaining this will be challenging (Criteria 5).
- **Discretionary (DAC) – Access Matrix Model:** Handling dynamic role assignments in an Access Matrix Model is challenging, managing individual access control lists (ACLs) for each user-object pair can be hard to maintain (Criteria 1 & 3). Automation is difficult to implement because of the individual access controls (Criteria 5).

An enhanced access matrix model as part of DAC is proposed to address the limitations of existing access control policies. This model explicitly specifies permissions granted to users for each object (instead of specifying each object for all users at Access Matrix Model), this simplifies specification and enables automation, improving the efficiency of access control. It provides flexibility and control over data access rights, particularly in large collaborative environments like the RobotLab project [10].

Table 1 contains an example where three objects are identified. Each object contains one owner (main author), the owner can assign users to writer (co-author) and reader (reviewer) roles.

		Roles		
Objects		Main author Owner	Co-Author Writer	Reviewer Reader
	Formula x	User1		
	Text y	User2	User1	User3
	Video z	User1		User2, User3

Table 1: Enhanced Access Matrix model

NOMAD already restricts access to unpublished uploads based on Main author (Owner), Co-Author (Writer), and Reviewer (Reader) [19]. After an upload is published, it becomes visible to everyone. NOMAD is open source [8, 20], which allows for modifications to its source code to enhance access control functionalities, enabling tailored solutions for projects like the RobotLab. By modifying NOMAD's source code, it is proved possible to extend the access control functionalities beyond the pre-publishing phase. This enables the enforcement the enhanced access matrix policy, even after data is published, ensuring that only authorized users can access specific data within the Oasis. Code changes can be found in [12].

4.2 Traceability

Ensuring data traceability within the RobotLab project is important for the reproducibility and integrity of the experiments and data. Traceability can be ensured in NOMAD by using Schemas. Schemas are used to define the structure of data entries [21]. They specify what combinations of values, objects, and lists are allowed within the data.

The NOMAD development team created a UML class diagram (Figure 1) with the data structure to see the references between each definition [22].

- A package (or a schema) contains one or more sections.
- A section definition contains definitions for subsections and quantities. Sections can reference the properties of other sections; this is called a base_section.
- Subsections allow for hierarchical references for other sections.
- Quantities can use section definitions (or other quantity definitions) as a type to define references.

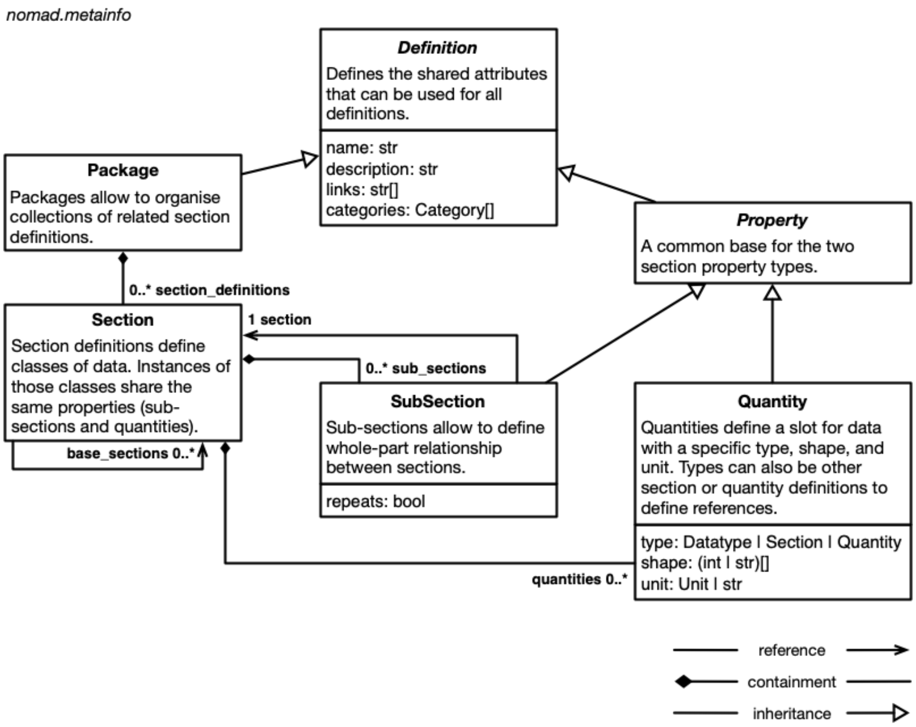

Figure 1: UML class diagram of NOMAD's schema structure [22]

Datasets and schema can be uploaded separately, or as a .zip or .tar.gz format [23]. When data is uploaded to NOMAD, the software interprets the files and determines which of them are the **main files**. Any other files in the upload can be viewed as **auxiliary files**. NOMAD recognizes main files by the extension; .archive.yaml. These main files can contain schema definitions, or data in a predefined structure.

To test NOMAD's capabilities for the traceability of data for a self-driving lab like the RobotLab, a use case was designed by a scientific domain expert of an academic partner of the RobotLab [13]. This use case involved a fluorescence measurement, where the result should be a plot of the absorbance is against the wavelength. The goal of this use case was to develop a custom schema that defines an Electronic Lab Notebook (ELN) and to create an example entry for this notebook.

Defining the schema

A questionnaire about the experiment is discussed with a domain expert [13]. This questionnaire runs through all schema definitions. In this schema, five main sections are defined. Each section defines its own quantities, base-sections and sub-sections. Table 2 contains the example for all definitions to design the schema for the fluorescence use case.

SECTION	QUANTITIES	BASE_SECTIONS	SUB_SECTIONS
Experiment	Name, Experiment Description, Experiment Visual, Sample	nomad.datamodel.data.EntryData	-
Chemical	Form	nomad.datamodel.metainfo.eln.Chemical nomad.datamodel.data.EntryData	-
Sample	Name, Tags, Chemicals	nomad.datamodel.metainfo.eln.Sample nomad.datamodel.data.EntryData	Processes: Pipetting, Fluorescence measurement
Process	Instrument	nomad.datamodel.metainfo.eln.Process	-
Instrument	-	nomad.datamodel.metainfo.eln.Instrument nomad.datamodel.data.EntryData	-
(sub)Pipetting	Size_of_pippet, Number_of_wells, heated_to_temperature, Time_shaken, explaination	Process	-
(sub)Fluorescence measurement	Explaination, data_file, explaination	Process nomad.parsing.tabular.TableData, nomad.datamodel.metainfo.plot.Plotsection	-

Table 2: Use case schema definitions

To visualize the relations between different sections from Table 2: Use case schema definitions, a UML diagram is created (Figure 2). The diagram illustrates the hierarchical structure of the sections Experiment, Chemical, Instrument, Process, and Sample, along with their associated properties (quantities) and inheritance (base sections).

The Experiment section can contain one to many Samples, illustrating a one-to-many relationship from Experiment to Sample. The Sample section can contain one to many Chemicals and one to many Processes, indicating two separate one-to-many relationships from Sample to Chemical and from Sample to Process. Processes have a one-to-one relation to the different process steps: Fluorescence_measurement and Pipetting. Both these processes inherit from the Process section, indicating that they are specialized types of processes. Furthermore, the Process section has a one-to-many relationship with the Instrument section, indicating that one Process can involve multiple Instruments. The Sample section contains a Processes subsection, which further breaks down into detailed process sections: Fluorescence_measurement and Pipetting. Each of these processes contains the subsection Process, referring to the defined process with its reference to the Instrument.

Each of these sections and subsections is defined with their 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 2: UML class diagram corresponding to the fluorescence use case

To validate the schema design, the actual schema for this fluorescence use case is developed. All files related to the development of this schema, including yaml schema's, json data and raw data can be found in [14]. The following aspects are examined:

- **Consistency in Data Representation:** The schema ensured that each experiment is logged with consistent metadata, facilitating uniform data representation across different experiments.
- **Equipment Traceability:** By linking processes to specific instruments within the schema, the usage of equipment can be tracked across multiple experiments. For example, the Tecan Infinite M+ Nano plate reader.

Schemas enables the logging of every significant step and equipment used, ensuring thorough traceability. Practical tests demonstrated that NOMAD could effectively manage the traceability requirements of the RobotLab project [13, 14]. Schemas provide a robust structure for capturing detailed metadata, ensuring that every aspect of the experiment is logged. The ability to reference different sections like instruments to each process step within the schema allows for precise tracking of equipment usage, ensuring the reproducibility of experiments.

Implementation of detailed schemas in NOMAD has proven to be an effective method for ensuring data traceability in the RobotLab project. By structuring data entries with comprehensive metadata, consistent and reliable documentation of experiments is achieved. This not only enhances the reproducibility of the experiments but also aligns with the FAIR principles, making the data **F**indable, **A**ccessible, **I**nteroperable, and **R**eusable.

4.3 Various and voluminous data

Investigation of literature and use cases revealed a spectrum of file formats including raw data, tabular data, images, videos, spectra files, and chemical structure files. These formats encapsulate different aspects of experimental data, ranging from numerical outputs to visual representations and molecular structures.

A standard NOMAD Oasis instance limits the size of each upload to 32GB, with a maximum of 10 unpublished uploads per user [24, 15]. If an upload is complete, the user can publish the upload, making it visible to any user in the Oasis. It is also possible to publish uploads with an embargo period, this can last up to 3 years [23]. The embargo may be lifted at any time. Embargoed data is visible to and findable by others, but it makes only some few metadata (e.g. chemical formula, instrument details) public. The raw-file and archive contents remain hidden (except to the owner and users with whom the data is explicitly shared) [25].

Findings indicate that the volume of data can vary significantly depending on the nature of the experiment. However, the largest experimental data includes videos from a Fluorescence signal from mCherry-LaminA [26]. The images were background corrected to display the C-iSCAT contrast. Each video was around 30 seconds with a size of 17MB. The complete experiment totaling 153MB in size [26], this volume is well within the limitations of NOMAD's upload capacity [15].

NOMAD is based on a bottom-up approach to data management. Instead of only supporting data in a specific predefined format, files are processed with parsers to extract information from an extendable variety of data formats [27]. Converting heterogenous files into homogeneous processed data is the basis to make data FAIR. NOMAD offers all its functionality through APIs, specifically RESTful HTTP APIs. This allows us to use NOMAD as a set of resources that can be uploaded, accessed, downloaded, and searched through HTTP requests. Various and voluminous data is tested by creating uploads through the API and GUI [15]. This showed that there are no limitations or exceptions found when uploading various file types to NOMAD, indicating that it supports various data types.

5 Discussion

The following discussion addresses the key findings of the RobotLab project in terms of access control, traceability, and management of various and voluminous data. Each section reflects on how the identified solutions and implemented modifications align with the project's needs, highlighting the benefits and addressing potential challenges.

5.1 Access control

Through the combination of an enhanced access matrix model and code modifications to NOMAD, the RobotLab project can achieve the necessary level of data access control [10]. This creates seamless collaboration among stakeholders while safeguarding sensitive information and aligning with the project's needs. The RobotLab project requires an enhanced matrix model to effectively manage access to its data [10]. This model implements the requirements for controlling access due to the involvement of various stakeholders with varying roles and responsibilities, with possible collaboration between them. While the standard NOMAD Oasis platform offers the capability to restrict access to the entire Oasis through whitelisting mechanisms, it falls short of providing the control over access to specific data resources for the RobotLab project [10].

Fortunately, NOMAD's open-source nature under the Apache License Version 2.0 [20] allows for the modification of its source code to tailor it to specific needs. NOMAD already incorporates roles such as main authors, co-authors, and reviewers to control access before an upload is published. Adjustments are made to NOMAD's codebase to test extended access control functionalities on uploads beyond the pre-publishing phase [10]. By modifying the code to enforce these roles even after publication, the access policy can be effectively implemented. This research updated the source-code, code changes can be found in [12]. This ensures that only authorized users with specific roles; main authors, co-authors, and reviewers, can access specific data within the Oasis, aligning with the requirements of the RobotLab project [10].

In essence, by enhancing NOMAD's access control mechanisms through code modifications, the RobotLab project can achieve the necessary level of data access control, enabling seamless collaboration while safeguarding sensitive information.

5.2 Traceability

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, making data FAIR.

- **Findable:** The inclusion of 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.

For instance, the Electronic Lab Notebook (ELN) schema [14], designed for a fluorescence experiment, contains the section "Chemical", this section references a base-section called nomad.datamodel.metainfo.eln.Chemical. This reference inherits additional properties from the NOMAD Library, enabling consistent data saving. Additionally, users can run advanced 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 [13]. 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 a specific instrument through targeted queries. Thus, by saving data with well-defined schemas, NOMAD ensures comprehensive traceability of the equipment used in experiments.

The UML class diagram 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 [13].

5.3 Various and voluminous data

The RobotLab project is characterized by its capacity to generate large and diverse datasets from high-throughput experiments in Supramolecular and Inorganic chemistry [2]. Addressing the compatibility of NOMAD, specifically NOMAD Oasis, with such varied and voluminous data necessitated a comprehensive analysis.

Experiments within the RobotLab generate significant volumes of data in various formats, including raw data, tabular data, images, videos, spectra files, and chemical structure files. For instance, fluorescence measurement experiments can produce extensive video data, which, despite its large size, remains within the manageable limits of NOMAD's upload capacity of 32GB per upload and up to 10 unpublished uploads per user. This capacity is generally sufficient for most of the experimental datasets observed in the literature review and case studies.

Through practical testing, NOMAD demonstrated its ability to handle various file types and volumes effectively. The platform's performance in managing, storing, and retrieving these datasets was positive, indicating its scalability and robustness. It is essential to note that while NOMAD supports a wide range of file types, the integration of new data formats may require the creation of additional custom parsers and schemas. This necessitates ongoing development efforts to ensure that NOMAD remains adaptable to the evolving experiments within the RobotLab project. Creating custom parsers for specific datasets was considered out of scope of the present research.

NOMAD's open-source [20] nature offers flexibility for customization. Adjustments to the codebase can be made to enhance its capacity for handling larger datasets or to support specific data requirements unique to the RobotLab. For instance, the upload size limit and the number of unpublished uploads can be modified to better accommodate the project's needs.

6 Conclusion

This research evaluated the suitability of NOMAD, specifically its customizable extension NOMAD Oasis, for the data management needs of the RobotLab project. The RobotLab aims to create a self-driving laboratory capable of executing and managing AI-driven chemical experiments, requiring a robust data management framework to handle the extensive and complex data generated.

In conclusion, this research supports NOMAD, particularly NOMAD Oasis, as a suitable data management framework for the RobotLab project. Its flexible and customizable nature, combined with enhanced access control mechanisms and robust traceability features, make it well-equipped to handle the extensive and complex data generated by a self-driving laboratory. However, adopting NOMAD Oasis does require a commitment of time and effort to create schemas for data representation. Additionally, as NOMAD is a relatively new concept and is still in development, some documentation is incomplete, which can be challenging to work with. Despite this, NOMAD Oasis holds significant potential for advancing data management for automated laboratory environments like the RobotLab.

7 Acknowledgements

I would like to extend my gratitude to several individuals and teams who have contributed significantly to this research:

Firstly, I would like to thank Joost van Toll (TU/e) for his domain expertise and for providing data for the fluorescence use case. Additionally, I would like to acknowledge the entire RobotLab team for their domain expertise and support during weekly standups. I want to thank the Radboud RobotLab team for providing domain expertise and raw data.

Lastly, I extend my sincere appreciation to Dirk van Zon for being an excellent colleague and fellow student and for his daily assistance. His collaboration and support have been greatly appreciated.

Thank you all for your invaluable contributions and support.

8 References

[1] "Big Ideas Big Chemistry Consortium," [Online]. Available: https://bigchemistry.nl/big-ideas/. [Accessed 29 05 2024].

[2] W. T. S. Huck and B. E. Meijer, "De revolutie van zelfdenkende moleculaire systemen".

[3] F. Hase, L. M. Roch and A. Aspuru-Guzik, "Next-Generation Experimentation with," *Trends in Chemistry,* vol. 1, no. 3, p. 10, 2019.

[4] "NOMAD Materials science data managed and shared," NOMAD-lab, [Online]. Available: https://nomad-lab.eu/nomad-lab/. [Accessed 15 05 2024].

[5] "FAIR Principles," [Online]. Available: https://www.go-fair.org/fair-principles/. [Accessed 15 05 2024].

[6] "NOMAD Share data, not files.," [Online]. Available: https://nomad-lab.eu/nomad-lab/nomad.html. [Accessed 15 05 2024].

[7] E. Bertino, G. Ghinita and A. Kamra, "Access Control for Databases: Concepts and Systems," *Foundations and Trends in Databases,* vol. 2, no. 1-2, pp. 1-148, 2010.

[8] "NOMAD Oasis Manage your research group's data," NOMAD-lab, [Online]. Available: https://nomad-lab.eu/nomad-lab/nomad-oasis.html. [Accessed 15 05 2024].

[9] "Nouryon," Nouryon, [Online]. Available: https://www.nouryon.com/. [Accessed 12 06 2024].

[10] L. v. d. Brandt, "GitHub NomadRobotLab," [Online]. Available: https://github.com/LarsvdBrandt/NomadRobotLab/blob/main/documentation/Data%20access.pdf. [Accessed 31 05 2024].

[11] L. v. d. Brandt, "NomadRobotLab/nomad-develop/scripts at main · LarsvdBrandt/NomadRobotLab," GitHub, [Online]. Available: https://github.com/LarsvdBrandt/NomadRobotLab/tree/main/nomad-develop/scripts. [Accessed 31 05 2024].

[12] L. v. d. Brandt, "NomadRobotLab/nomad-develop/nomad-source at main · LarsvdBrandt/NomadRobotLab," GitHub, [Online]. Available: https://github.com/LarsvdBrandt/NomadRobotLab/tree/main/nomad-develop/nomad-source. [Accessed 31 05 2024].

[13] L. v. d. Brandt, "Traceability Github NomadRobotLab," [Online]. Available: https://github.com/LarsvdBrandt/NomadRobotLab/blob/main/documentation/Traceability.pdf. [Accessed 31 05 2024].

[14] L. v. d. Brandt, "NomadRobotLab/nomad-develop/schemas at main · LarsvdBrandt/NomadRobotLab," GitHub, [Online]. Available: https://github.com/LarsvdBrandt/NomadRobotLab/tree/main/nomad-develop/schemas. [Accessed 31 05 2024].

[15] L. v. d. Brandt, "Various and vouminous Github Nomad RobotLab," [Online]. Available: https://github.com/LarsvdBrandt/NomadRobotLab/blob/main/documentation/Various%20and%20voluminous.pdf. [Accessed 31 05 2024].

[16] H. Shen and P. Dewan, "Access control for collaborative environments," 1992.

[17] "Restricting Access to Oasis," NOMAD-lab, [Online]. Available: https://nomad-lab.eu/prod/rae/docs/oasis.html?highlight=restricting#restricting-access-to-your-oasis. [Accessed 15 05 2024].

[18] P. Samarati and S. D. C. D. Vimercati, "Access Control: Policies, Models, and Mechanisms," *Lecture Notes in Computer Science,* pp. 137-196, 2001.

[19] "From files to data," NOMAD-lab, [Online]. Available: https://nomad-lab.eu/prod/v1/staging/docs/explanation/basics.html. [Accessed 15 05 2024].

[20] "GitHub nomad," [Online]. Available: https://github.com/nomad-coe/nomad/blob/develop/LICENSE. [Accessed 29 05 2024].

[21] "How to write a YAML schema package," [Online]. Available: https://nomad-lab.eu/prod/v1/staging/docs/howto/customization/basics.html.

[22] "Data structure," NOMAD-Lab, [Online]. Available: https://nomad-lab.eu/prod/v1/staging/docs/explanation/data.html. [Accessed 28 05 2024].

[23] "Quick start," NOMAD-Lab, [Online]. Available: https://nomad-lab.eu/prod/v1/staging/docs/examples/computational_data/uploading.html. [Accessed 28 05 2024].

[24] "How to upload data," NOMAD-lab, [Online]. Available: https://nomad-lab.eu/prod/rae/docs/upload.html. [Accessed 15 05 2024].

[25] "Support," NOMAD-lab, [Online]. Available: https://nomad-lab.eu/nomad-lab/support.html. [Accessed 29 05 2024].

[26] M. Kuppers, D. Albrecht, A. Kashkanova, J. Luhr and V. Sandoghadar, "Confocal interferometric scattering microscopy reveals 3D nanoscopic structure and dynamics in live cells," *Nature Communications,* 2023.

[27] "Supported Parsers," NOMAD-Lab, [Online]. Available: https://nomad-lab.eu/prod/v1/staging/docs/reference/parsers.html#supported-parsers. [Accessed 29 05 2024].