

The RDM System LARA

- semantics through automation from bottom up

Mark Doerr¹[\[https://orcid.org/0000-0003-3270-6895\]](https://orcid.org/0000-0003-3270-6895), Stefan T. Maak¹[\[https://orcid.org/0009-0000-9930-0182\]](https://orcid.org/0009-0000-9930-0182),
Marian J. Menke¹[\[https://orcid.org/0000-0002-5887-3045\]](https://orcid.org/0000-0002-5887-3045), and
Uwe T. Bornscheuer¹[\[https://orcid.org/0000-0003-0685-2696\]](https://orcid.org/0000-0003-0685-2696)

¹University Greifswald, Greifswald, Germany

Abstract: LARAsuite (<https://gitlab.com/larasuite>) is a free and open source research data management system that addresses the problematic of manual data insertion and metadata assignment into the corresponding databases by a radically automated processes. Data and Metadata is mainly not inserted by humans, but by the machines producing the data and automatically transferring this generated data and metadata to the LARA RDM system. This data transfer is achieved through the intensive utilisation of the free and open lab-automation standard SiLA (Standardisation in Labautomation) (<https://sila-standard.org>), combined with a simple to note process description language (pythonLab) and orchestration, scheduling, data aggregation and evaluation. Many LARA instances can be selectively synchronised to form a decentralised network of data infrastructures across labs and institutions. Research data and meta-data can be queried by a SPARQL endpoint. As the LARAsuite comes close to an ideal FAIR-principles based RDM System¹, is generic to the most common natural science applications, is open source (python based), modular and easy deployable, it can be used to showcase NFDI goals for a wide range of scientists.

Keywords: RDMS, Semantics, Ontologies, Automation, SiLA, JSON-LD, Workflow-description

1 Introduction

1.1 Motivation of the LARA system

The majority of recent research data management systems (RDMS) are suffering from the fact that most data and metadata - and it's meaning (semantics) need to be entered manually. Also all relations between the data needs to be assigned by the hand of the researcher. The LARAsuite[2] addresses this by a radical automation approach. It uses the free and open source lab communication standard SiLA[3] and the underlying wire protocol gRPC.

¹LARAsuite tries to achieve full maturity levels according to the *RDA FAIR data maturity model Working Group's* maturity levels F1-4, A1-2, I1-3 and R1 (1)

1.2 Key features of the LARA system

- fully open source and (mostly) written in python - so any scientist can adjust the system to the needs
- modular and extensible due to its modular architecture / (partly) micro-services new functionality can be added
- easy deployable through docker containers / docker-compose or kubernetes

1.3 Workflow with the LARA system

Projects and experiments are designed with a web-interface or a python notation. Processes can be formulated in the pythonLab process description language [4], which is interpreted by the LabOrchestrator [5]. The LabOrchestrator orchestrates the whole experiment, executed by machines and humans (also in collaboration between humans and machines). Which parts of processes are executed by which entity (machine or human) is described in the *pythonLab procedures* defining how a process is executed. Humans have always the option to intervene any running process by pausing, interrupting / resuming it². For the scheduling of all the experimental steps the LabScheduler[6] can be used. All data is collected in the LARA databases and automatically semantically enriched, using modular ontologies. Data can then automatically evaluated and visualised or further processed by interfacing to the jupyter infrastructure.

2 Architecture of LARA

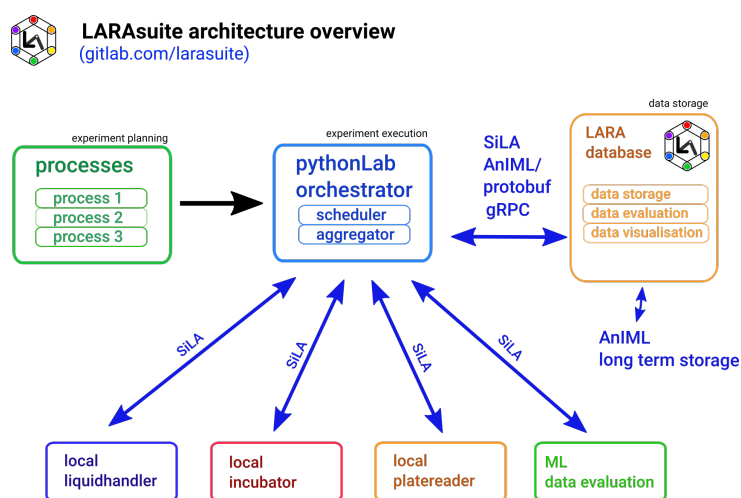


Figure 1. The LARA architecture.

The LARA suite has currently components to model the following entities of projects and experiments

- Projects and Experiments
- User
- Processes
- Data (including associated Metadata)
- Material (Parts, Devices, Device Setups, Labware)- including an inventory
- Substances (all chemical substances and mixtures) - including an inventory

²Modification of a running process is currently under development

- Sequences (in case substances have a sequence, like DNA, RNA, plasmids, peptides, proteins,
- Organism (Bacteria, Plants, Animals, ...)

More details can be found on the LARAsuite Webpage: <https://github.com/larasuite/lara>.

Data availability statement

The full source code of the LARAsuite [2] is available on gitlab: <https://gitlab.com/larasuite> - GPL-3 licensed. The ontologies, semantic web tools and SiLA servers are also freely available <https://gitlab.com/opensourclab>[7].

Underlying and related material

Videos of the University LARA robotic platform, which runs with the LARAsuite, can be viewed here: <https://lara.uni-greifswald.de>

Author contributions

Conceptualization: M.D.; Methodology: M.D, S.T.M; Software: M.D, S.T.M; Writing – Original Draft: M.D., Writing – Review Editing: M.D.; Visualization: M.D.; Project-Funding: U.B., M.D.

Competing interests

The authors declare that they have no competing interests.

Funding

The Deutsche Forschungsgemeinschaft (DFG) is acknowledged for funding this research as part of the Nationale Forschungsdateninfrastruktur (NFDI) initiative (grant No.: NFDI/2-1 - 2021).

The Bundesministerium für Forschung und Wissenschaft (BMBF) - KIWI-biolab project.

Acknowledgements

We would like to acknowledge Stefan Born (TU-Berlin) for his very valuable comments.

References

- [1] R. F. data maturity model Working Group. "Fair data maturity model: Specification and guidelines." (2023), [Online]. Available: <https://www.rd-alliance.org/group/fair-data-maturity-model-wg/outcomes/fair-data-maturity-model-specification-and-guidelines-0> (visited on 07/25/2023).
- [2] M. Doerr. "The lara suite." (2023), [Online]. Available: <https://github.com/LARAsuite> (visited on 04/26/2023).
- [3] T. S. organisation. "Sila - standards in lab automation." (2023), [Online]. Available: <https://sila-standard.org> (visited on 04/26/2023).

- [4] M. Doerr. "The pythonlab process description language." (2023), [Online]. Available: <https://github.com/opensourcelab/pythonlab> (visited on 04/26/2023).
- [5] M. Doerr and S. T. Maak. "The laborchestrator." (2023), [Online]. Available: <https://github.com/opensourcelab/laborchestrator> (visited on 04/26/2023).
- [6] S. T. Maak and M. Doerr. "The labscheduler." (2023), [Online]. Available: <https://github.com/opensourcelab/labscheduler> (visited on 04/26/2023).
- [7] M. Doerr. "The opensource lab project." (2023), [Online]. Available: <https://github.com/OpenSourceLab> (visited on 04/26/2023).