MAX PLANCK INSTITUTE FOR DYNAMICS OF COMPLEX TECHNICAL SYSTEMS MAGDEBURG

CSC COMPUTATIONAL METHODS IN SYSTEMS AND CONTROL THEORY

TA 2

MaRDI

Universität Münster

Description and Design of FAIR CSE Workflows: A Multi-Layered Approach

Pavan L. Veluvali[1], Jan Heiland[1], Peter Benner[1]

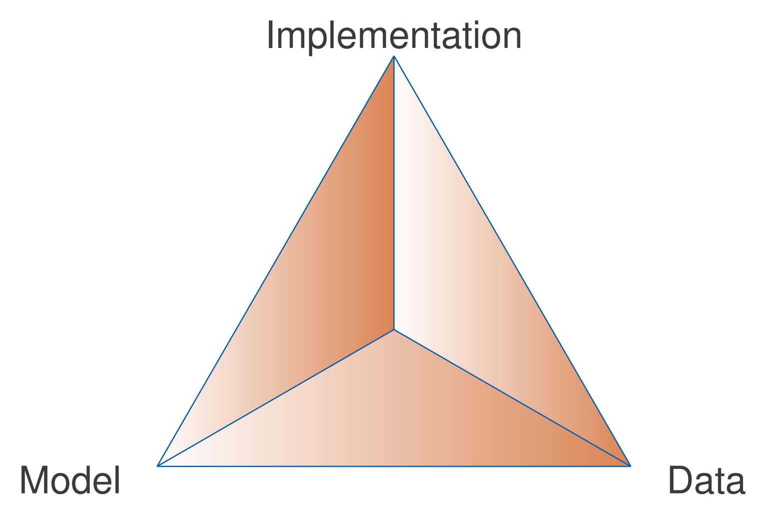
[1]Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany

Key Features, Principles and Structure of Workflow Tools

MaRDIFlow

– a framework for the abstraction of FAIR CSE workflows.

- Integrates execution and environmental configurations, representing them through multi-layered descriptions with customizable abstraction levels
- Every building block described differently for modular integration
- Interchangeable and redundant components defined by I/O behavior
- Combines models, code, and data to describe workflows



Matrix of Solutions and Features

Workflow tools	Meta-data documentation	Abstraction level	Standardized I/O
MaRDIFlow	✓	high	✓
Galaxy	○	high	○
Kepler	○	high	✓
Askalon	○	low	○
Nextflow	○	high	✓
Jupyter	✓	low	○

FAIR Features of Workflow Tools:

Meta-data Documentation Enforced: Does the workflow tool, by design, enforce the documentation of building blocks including the meta-data? *Jupyter* has the opportunity to include descriptions but the code runs independent of it.

Level of Abstraction: Can the building blocks be considered a certain realization of an otherwise high-level entity? And, thus, be easily interchangeable? *Askalon* acts as scheduler for jobs and lacks components that describe mathematical models.

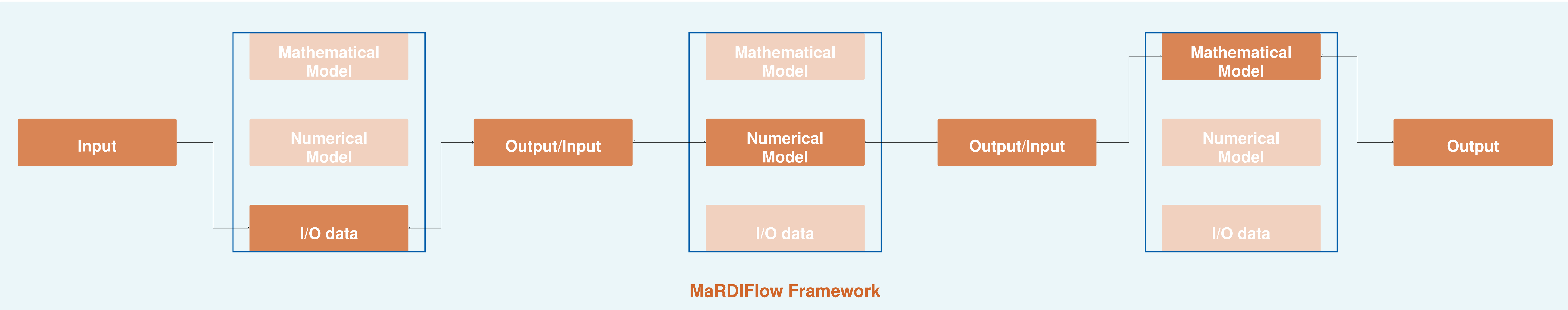
Standardized I/O: Does the description of the interfaces adhere to general meta-data schemes? Like *Jupyter* notebooks don't have any structure in the interfaces.

References:

- P. L. Veluvali, J. Heiland, and P. Benner, *MaRDIFlow: A CSE workflow framework for abstracting meta-data from FAIR computational experiments*, arXiv preprint, 2024, `https://doi.org/10.48550/arXiv.2405.00028`.
- The MaRDI consortium, *MaRDI: Mathematical Research Data Initiative proposal*, Zenodo, 2022, `https://doi.org/10.5281/zenodo.6552436`.

arXiv preprint

Demonstrator



Design Objectives & User Interface in MaRDIFlow

Design Objectives:

- Addresses **Reproducibility** in FAIR
- Modular building blocks
- Enhanced reproducibility of research data as a result of vertical dimension of workflow components.
- Provide a specialized programming environment for users to orchestrate a computer-based experiment
- As a service: A framework that can handle both documentation and realization

- Meta-data realized as:

Mathematical Model
... or ...
Finite Element Model
... or ...
Finite Difference Model
... or ...
I/O Data

User Interface:

- A command line interface (CLI) tool for users
- Version 1.0: A Python-based implementation
- Metadata input with standardized JSON file
- Community driven use-case implementation
- Users can execute & document computer-based experiments
- Includes built-in unit tests
- `MaRDIFlow -config config_CH.ini`

```
[DEFAULT]
workflow-title = Provide workflow title here
input  = files/CH_input.json
output-directory = demos/Output_usecase_CH/
component = TRUE
math_solver = Cahn-Hilliard
display =
display_pdf  =
display_html =

[DISPLAY]
display = TRUE
display_html = TRUE

inputmarkdown = file/path/to/workflow_CH.md
outputmarkdown = documentation_scheme_output.html
```

```
{
        "nx"     : 48,
        "ny"     : 48,
        "dx"     : 2.0e-9,
        "dy"     : 2.0e-9,
        "czero" : 0.5,
        "R"      : 8.314,
        "temp"  : 673.0,
        "nsteps": 1800,
        "ac"     : 3.0e-14,
        "La"     : 13943.0
}
```

Example Use Case

Cahn-Hilliard Model:

- Free and open source example
- Free energy functional:

$$F = \int_V \left[\frac{K}{2} (\nabla \phi)^2 + H f_{dw}(\phi) \right] \mathrm{d}V$$

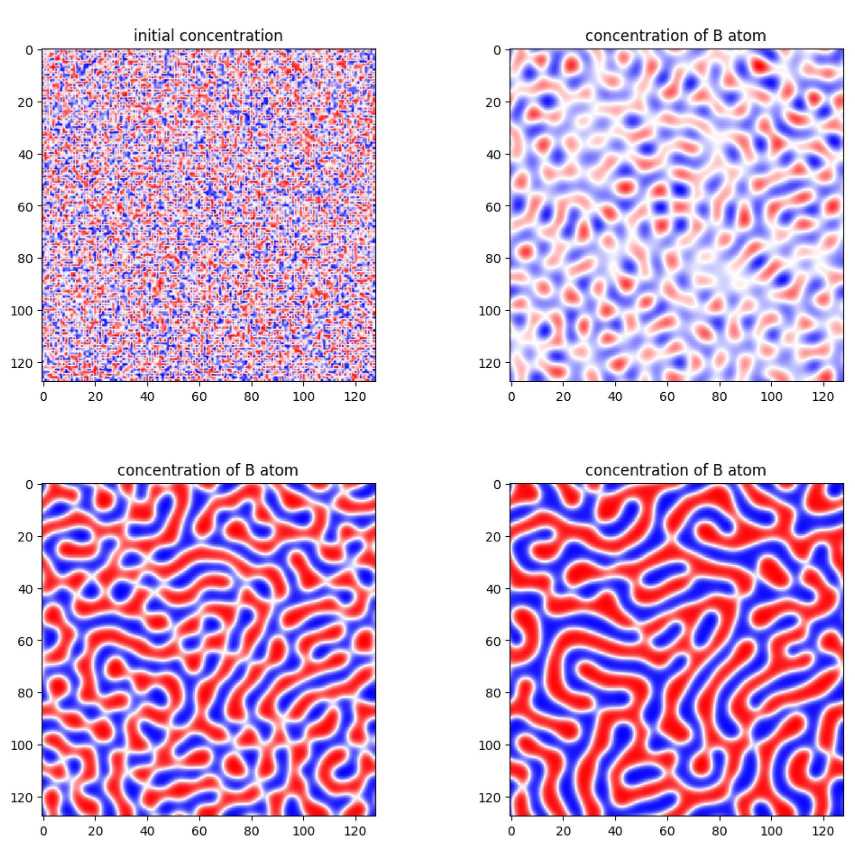
$$f_{dw} = \phi^2 (1 - \phi)^2$$

- Interface Profile:

$$\phi(x) = \frac{1}{2} \left[1 + \tanh \left(\frac{x}{2\epsilon} \right) \right]$$

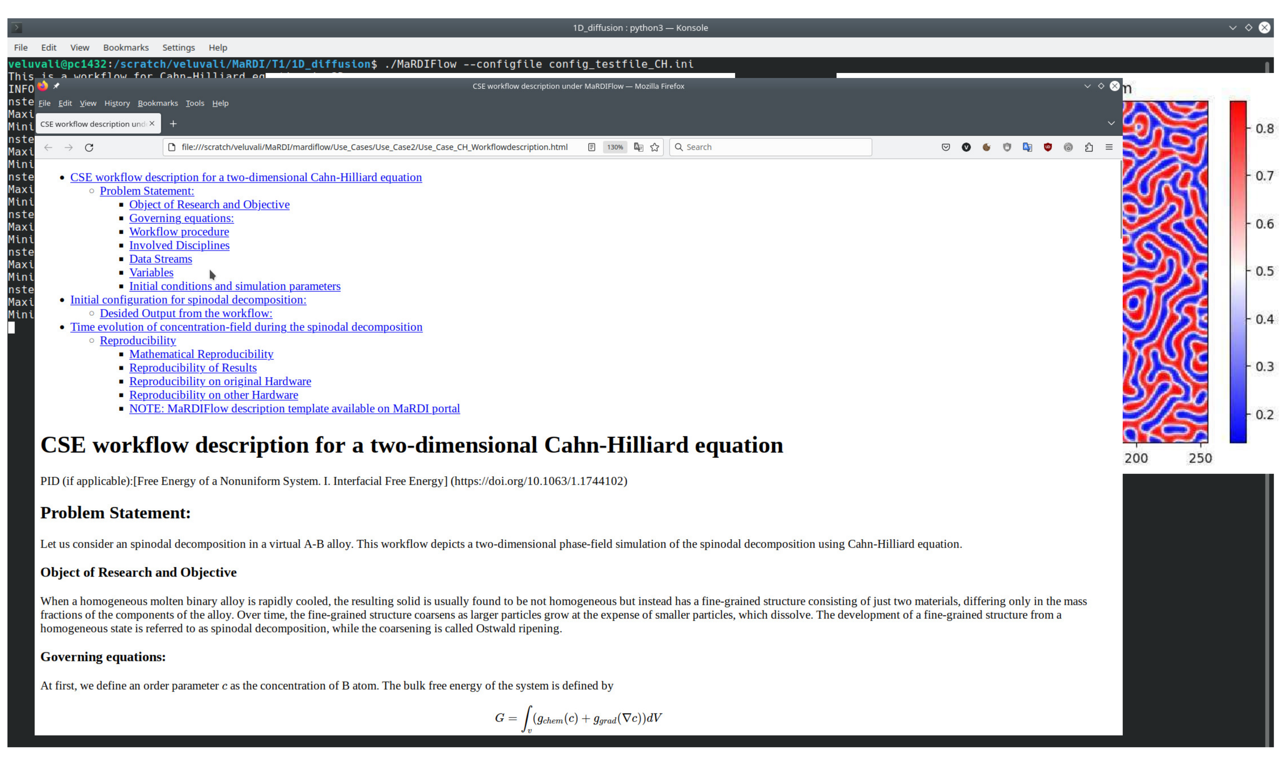
- Governing equations:

$$\frac{\partial \phi}{\partial t} = -M \frac{\delta F}{\delta \phi} \quad \text{or} \quad \frac{\partial \phi}{\partial t} = \nabla \cdot \left(M \nabla \frac{\delta F}{\delta \phi} \right)$$



Documentation Scheme:

- In cooperation with **TA4**



- Other implemented use cases: Methanization Reactor Model — In collaboration with **NFDI4Cat**