# COMMON WORKFLOW DESCRIPTION

## AN EXPERIMENTAL VIEWPOINT


S. Brinckmann, F. Chen, T. Dahmen, T. Hickel, L. Huber, S. Menon, J. Murugan, J. Neugebauer, R. Schwaiger, H. Tsybenko, O. Waseda

IMD-1: Microstructure and Properties of Materials

JÜLICH Forschungszentrum

# ALLOW FOR COMPARISON AND REUSE OF WORKFLOWS

Experiments 1

1. Prepare Al sample
2. Measure sample
3. Tensile test
4. Analyse curves
5. Calculate Modulus

# ALLOW FOR COMPARISON AND REUSE OF WORKFLOWS

Experiments 1

1. Prepare Al sample
2. Measure sample
3. Tensile test
4. Analyse curves
5. Calculate Modulus

Experiments 2

1. Prepare polymer
2. Measure sample
3. Tensile test
4. Analyse curves
5. Calculate Modulus

# ALLOW FOR COMPARISON AND REUSE OF WORKFLOWS

| Experiments 1 | Experiments 2 | Simulations |
|---|---|---|
| 1. Prepare Al sample | 1. Prepare polymer | 1. Build model |
| 2. Measure sample | 2. Measure sample | 2. Apply forces |
| 3. Tensile test | 3. Tensile test | 3. Calculate |
| 4. Analyse curves | 4. Analyse curves | 4. Analyse curves |
| 5. Calculate Modulus | 5. Calculate Modulus | 5. Calculate Modulus |

# MANY WORKFLOW LANGUAGES EXIST

Dedicated workflow languages

- Common workflow language
- Snakemake
- Nextflow
- Nexus
- Kadi4Mat - json

...

Programming languages:

- C
- python
- bash

...

## TASK: PICK ONE AND USE FOR SIMULATIONS AND EXPERIMENTS

# FORMAL WORKFLOW DESCRIPTION EXISTS

## BUSINESS PROCESS MODELING

- Visual representation
- Process analysis
- Communication and documentation
- Simulation and automation


Two layers of information:
- Top level: short description
- Bottom level: details

Di Ciccio, et. al 10.1007/978-3-319-94214-8_4

# FORMAL WORKFLOW DESCRIPTION EXISTS

## BUSINESS PROCESS MODELING

- Visual representation
- Process analysis
- Communication and documentation
- Simulation and automation

# GOAL: WORKFLOW STANDARDIZATION

Experiments have two layers:

- Workflow step incl. description
- Standard Operating Procedure

Simulations have two layers:

- Function calls
- Each function is a graph node

# WORKFLOW EXAMPLE

```
from common_workflow_description import Storage, Sample, step
from analysis_steps import plot_curves, calc_E

wf = Workflow('Sandia Fracture Challenge 3')
proceduresLibrary = urlparse('https://…')
storage=Storage(proceduresLibrary)

sample = Sample('AM_NA_05')
wf.step1 = step(storage, sample, 'metallography', {})
wf.step2 = step(storage, sample, 'light microscopy', {})
…
file_name = [v for k,v in list(wf....items())...][0][1]
wf.step6 = plot_curves(file_name, 'Strain (Gauge0)', 'Engr. Stress')
wf.step7 = calc_E(file_name, 'Strain (Gauge0)', 'Engr. Stress')

print('Output: ','\n '.join([f"{k}: {v}" for k,v in list(wf…)]))
wf.draw().render(filename="io_demo", format="png", cleanup=True)
```

Header

Define workflow

Experiments

Analysis

Footer

same as pyiron-workflow style

# PROCEDURE IN MARKDOWN+

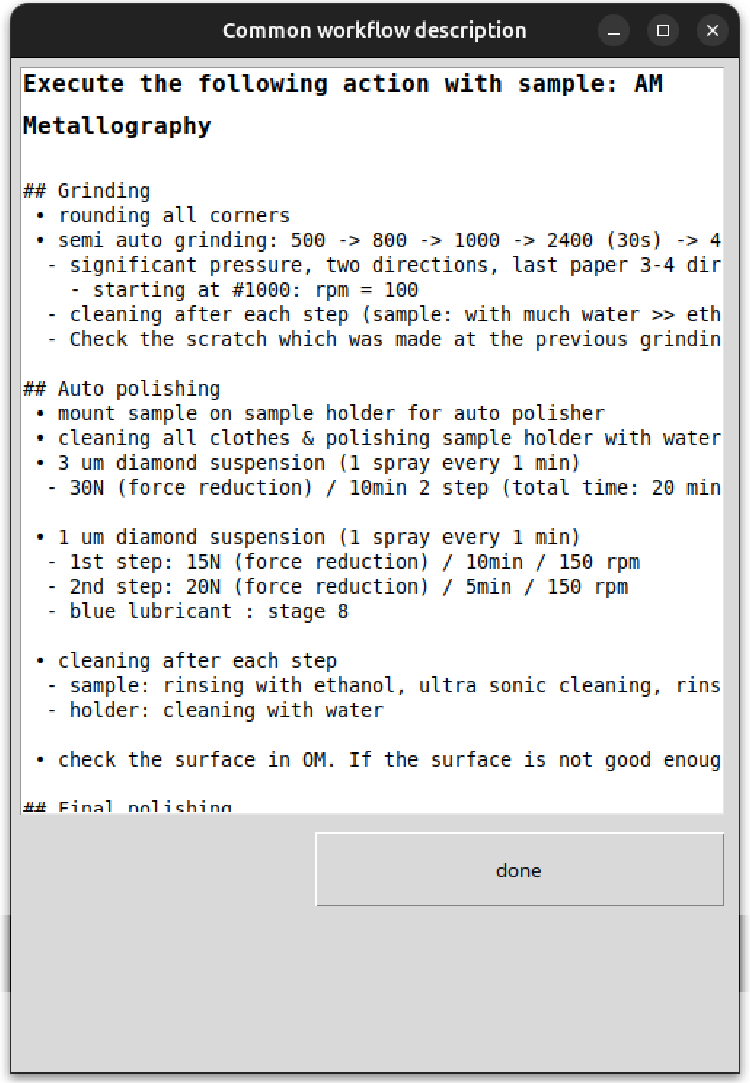
```
# Zeiss SEM
- Vent
- Open chamber
- Mount sample and close chamber
- Use it at |voltage|20| kV
- Use a aperture of |aperture|3|


Author: Steffen Brinckmann, IMD-1, FZJ
<!--- version:1.0 |filename|| -->
```

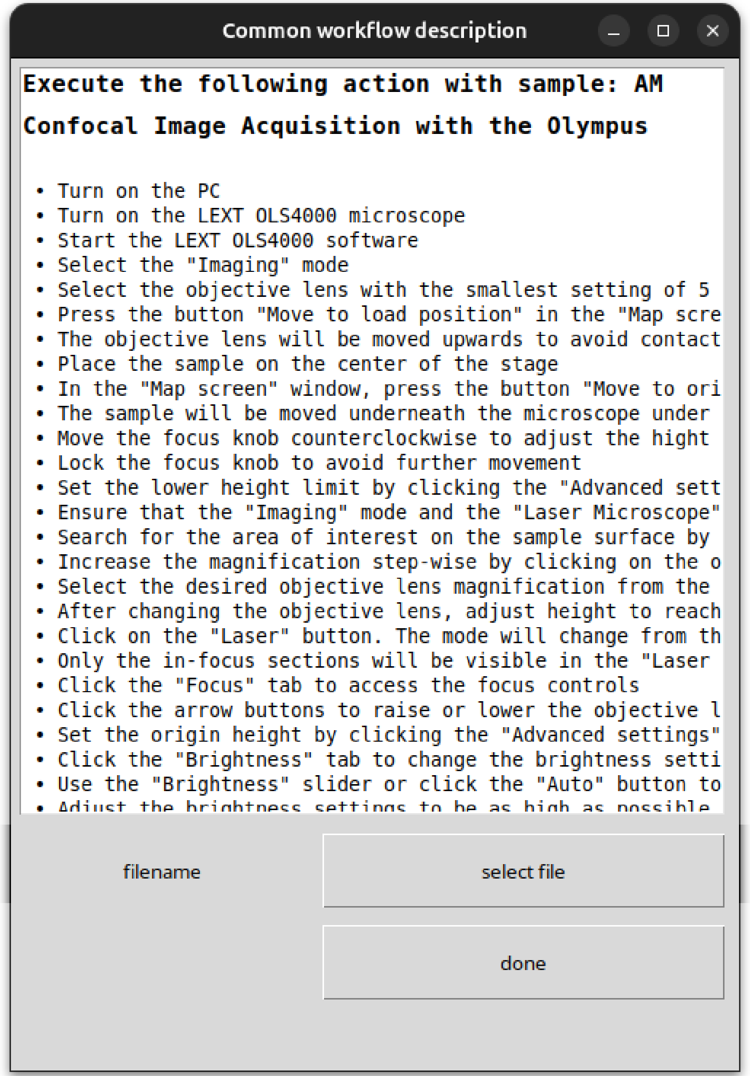
# EXAMPLE: INSTRUCTIONS

Instructions can have:

- no forms

- only ask for a file



# INSTRUCTION OPTIONS

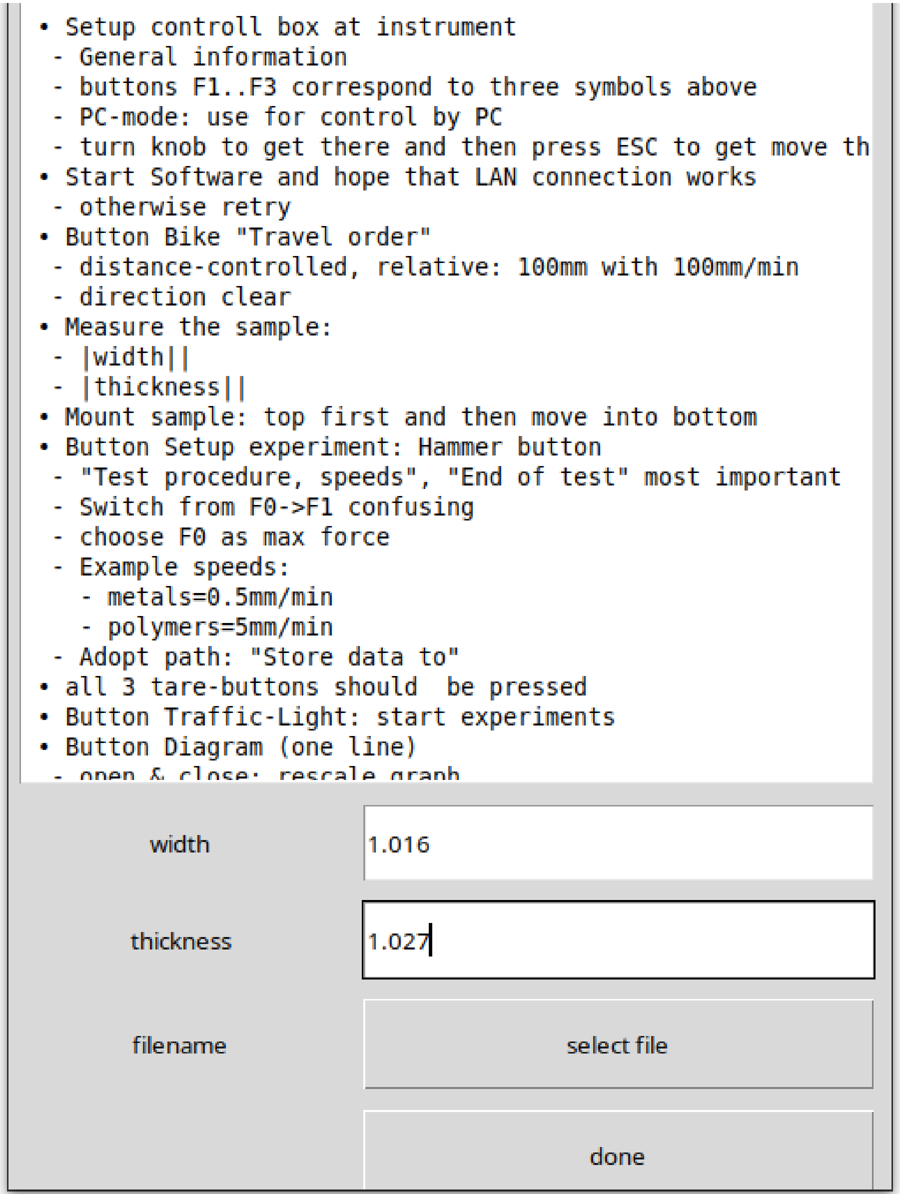
Instructions can include parameters

- geometry

- real process parameter

```
prescribed flow rate: 16
real flow rate: 10
```

- experimental metadata that is not in output file





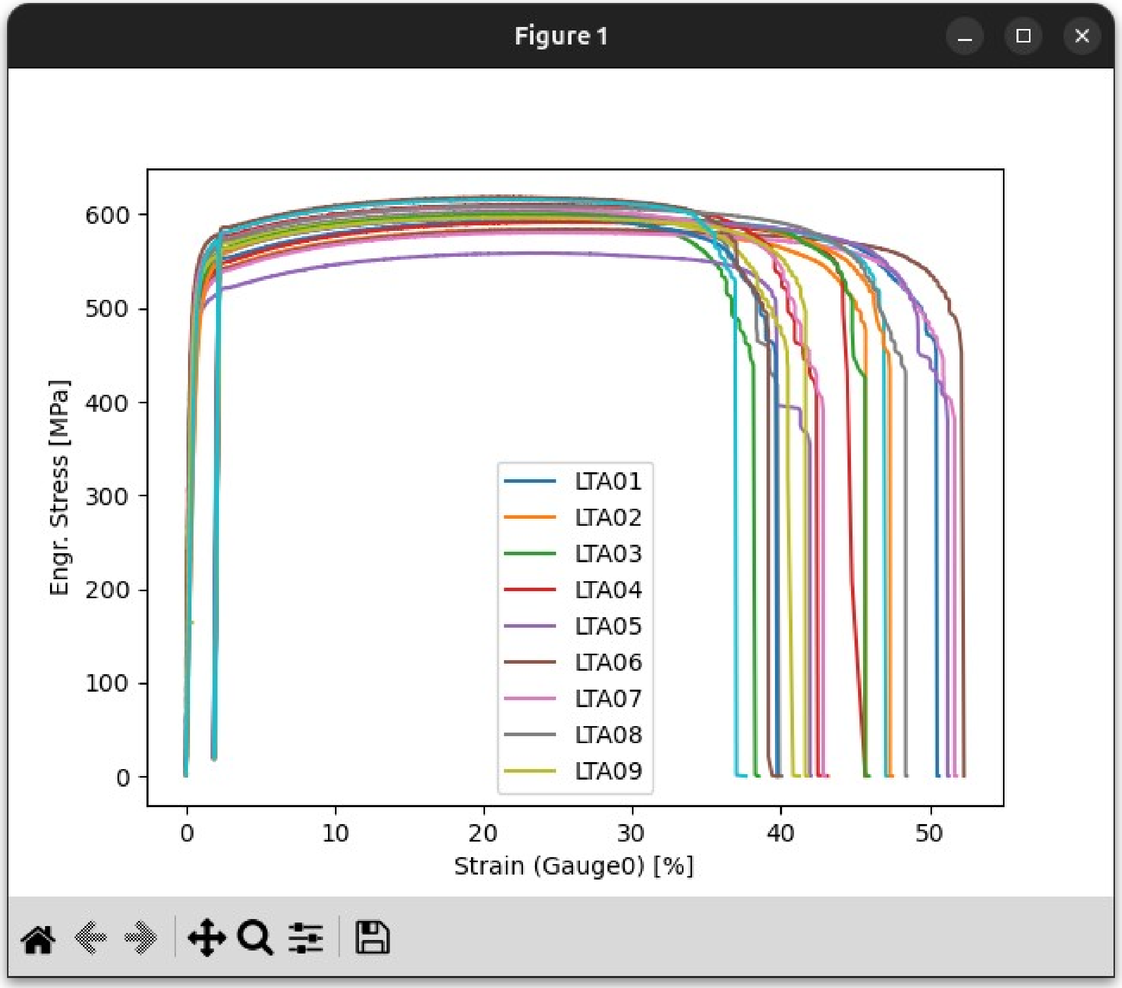

JÜLICH Forschungszentrum

# EXAMPLE: ANALYSIS STEP

All possibilities exist for:

- graphical plotting (tensile curves)
- calculating material properties
- simulation steps

```
Young's modulus in GPa:
    average=159.82
    std-dev=9.05
```



# OUTPUT AND LOGFILE

```
Started minimalistic workflow engine
Output:
  step1: [Name: AM05,'',{}]
  step2: [Name: AM05,'',{'magnification': '5'}]
  step3: [Name: AM05,'....xlsx',{'width': '1.016'}]
  step4: [Name: AM05,'',{'magnification': '5'}]
  step5a:[Name: AM05,'',{'voltage':'30','aperture':'3'}]
  step5b:[Name: AM05,'',{'voltage':'30','aperture':'2'}]
```

```
08-22 10:52:39|INFO:Start workflow
08-22 10:56:31|INFO:Start step
  sample:{AM_NA_05}
  procedure-name:{tensile test}
  sha256:{ae9c351e9fcbd41fc39d0e3831d9dc8d3388bd7e714869aa050b870dc21ac341}
  Parameters: {{}}
08-22 10:57:40|INFO:Save step
  file-name:...LongitudinalTensileOverall.xlsx
  metadata: {"width": "1.016"}
08-22 10:57:40|INFO:End step
```

# TWO ENGINES EXIST TO EXECUTE WORKFLOWS

Pyiron-workflow:

- Many features: graphics, super-computer...

- Allow for adoptive of workflows

**Same output and logging files**

Minimalistic engine:

- Can only run common-workflow-description

- No installation required

# A COMMON WORKFLOW DESCRIPTION EXISTS FOR EXPERIMENTAL AND NUMERICAL MATERIALS SCIENCE

- Most basic version exists
    - creates a receipt of all experimental steps
    - unified way to write procedures
- You can use pyiron-workflow to adopt it
- **GUI version with drag-drop will exist**

**Looking for scientists feedback: What is missing?**

# NON UNIQUE DEFINITION: WHAT IS "A STEP" (same in simulations)

Smallest step that make scientific sense to execute individually (might depend on others)

- SEM
- EBSD  (requirement of SEM before)
- EDX (requirement of SEM before)

Issue: requirements can only be resolved if there is step types (SEM procedures, …) → much more complicated (second version)



JÜLICH Forschungszentrum

# KNOWN SHORTCOMMING

Validation of experimental input:

“What happens if I enter a non-nonsensical value?”

Issue: How to integrate that into procedure that it is easy to write?

- allowed values
- only integers can be entered
- regex needed?

# NOT A SHORTCOMMING

Allow for comment / remark in each step?

- one can enter unforeseeable events

Very easy fix: just add “|comment||” at end of file