SWIRRL & C4I
Reproducible Notebooks in Climate4Impact

Alessandro Spinuso, Ian van der Neut
Hans Verhoef, Friedrich Striewski, Mats Veldhuizen

R&D Data Technology and Observations
Climate4Impact Re-design

- **GUI usability** (Search, Selection, Subsetting)

- **Flexible analysis features** (Integration of Notebooks, ICCLIM and Workflows/Batch-processing)

- **Automated reproducibility mechanisms** (Datasets Version/GitHub/Binder/Provenance)

- **Making FAIRness of Data and Methods accessible**

- **New front-end technology**
- access distributed raw data

- develop, document and reuse methods for processing and visualisation.

- update/extend raw data and software

- Track changes and rollback (Traceability/Recovery)

- keep old versions of the data after updates (Reproducibility)

- snapshot and restore the state of a workspace software (Reproducibility)
Interactive & reproducible Workspaces

Objective: Extend C4I with Data Driven & Reproducible Workspaces

Climate4Impact Search for CMIP5/6 Cordex Data (Distributed Data)

Incremental data staging/subsetting onto customisable and Reproducible Notebooks (extensible to other tools..)
Objective: Extend C4I with Data Driven & Reproducible Workspaces

Climate4Impact Search for CMIP5/6 Cordex Data (Distributed Data)

Incremental data staging/subsetting onto customisable and Reproducible Notebooks (extensible to other tools..)

Interactive & reproducible Workspaces

SWIRRL-API

Trace Changes to Software and Data
Restore Environments
Objective: Extend C4I with Data Driven & Reproducible Workspaces

Climate4Impact Search for CMIP5/6 Cordex Data (Distributed Data)

Incremental data staging/subsetting onto customisable and Reproducible Notebooks (extensible to other tools..)

Interactive & reproducible Workspaces

Software and Environment to Git

Trace Changes to Software and Data

Restore Environments

SWIRRL-API
Objective: Extend C4I with Data Driven & Reproducible Workspaces

Climate4Impact Search for CMIP5/6 Cordex Data (Distributed Data)

Incremental data staging/subsetting onto customisable and Reproducible Notebooks (extensible to other tools...)

Interactive & reproducible Workspaces

Software and Environment to Git

Trace Changes to Software and Data

Restore Environments

SWIRRL-API

MyBinder Reproduce

Data
SWIRRL Jupyter Lab Extension

- **Monitor Jobs**

- **Snapshot Controls**

- **Trace Activities and trigger rollback actions**
SWIRRL Jupyter Lab Extension

- **Monitor Jobs**
- **Snapshot Controls**
- **Trace Activities and trigger rollback actions**
Demo
Provenance-aware Workspaces
SWIRRL-API

A Web API (high-level piece of infrastructure) to:

- Manage **Working Sessions** offering Notebook and Visualisation Services
  - Run Workflows (CWL) for data staging and preprocessing onto the Working Session
  - Keep data staging history
  - Provenance-aware
  - Restore SW Environments to a state in the past
  - On-demand Binder Snapshots to GitHub (Environment, methods, data references)

- Jupyter Lab Extension!

[Resources]
- https://gitlab.com/KNMI-OSS/swirrl/swirrl-api
- https://zenodo.org/record/4264852#.X7ZeqNv_qNZ
"Update" a Notebook Environment

Luc Moreau et al. A Templating System to Generate Provenance


https://eprints.soton.ac.uk/405025/1/provtemplate.pdf

https://github.com/EnvriPlus-PROV/ProvTemplateCatalog
“Run” a Workflow/Container

Deployment Info

Association with users and systems

Execution

Volume State
Restoring Notebook Libraries (Prov Graph)

1 Create
2 Update
3 Update (J3)
4 Restore
Ongoing and Future Work

- Integration of OpenDAP data-staging workflows with remote subsetting
- Improved Deployment Performances
- Manage multiple Workspaces per User
- Integration of Datasets’ metadata Pages (DOI)
- Selection of Model’s Members
- Activities view and Restoring actions in Jupiter Lab
- Alpha testers (volunteers?)
- AAI C4I/SWIRRL