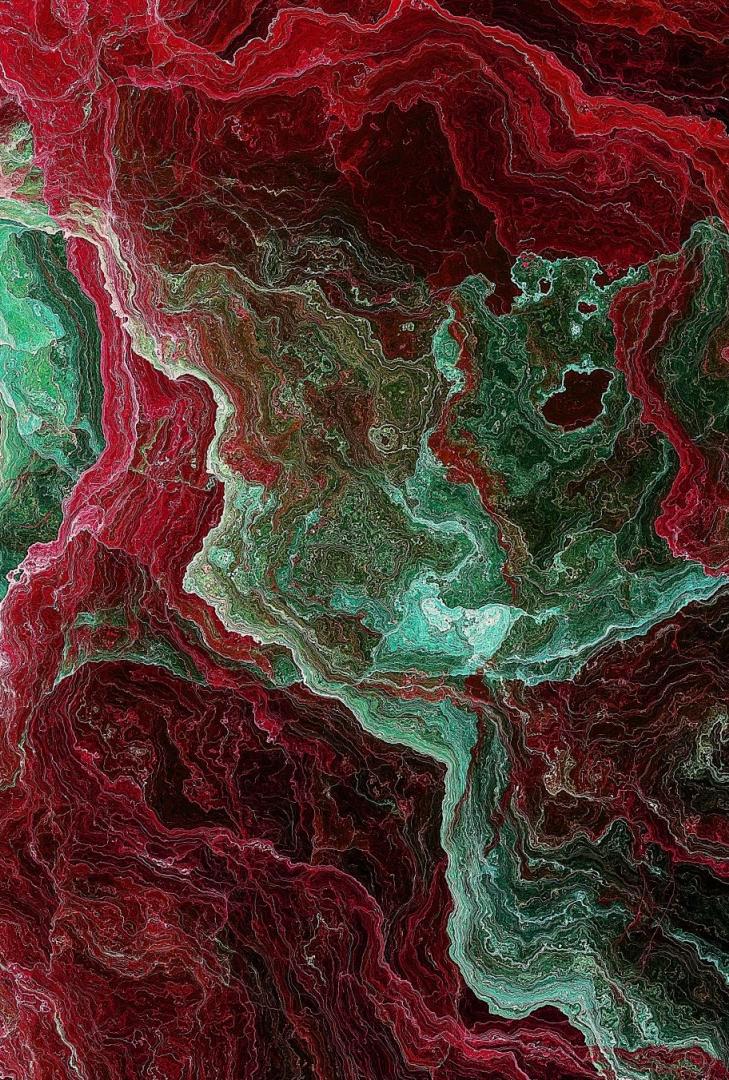




ASSERT: A SOFTWARE SUITE FOR EARTH-SYSTEMS REGRESSION TESTING
SUMMER 2023 INTERNSHIP

Deon Kouatchou



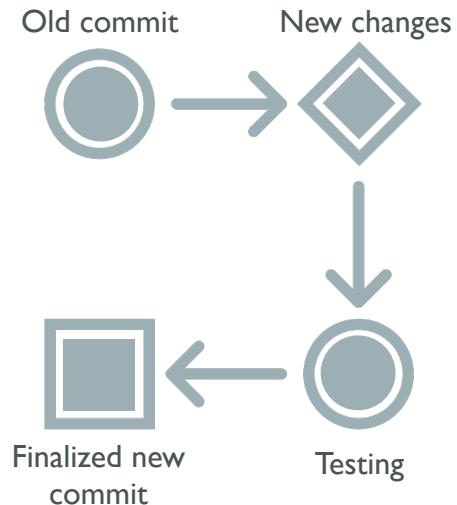


BACKGROUND

REGRESSION TESTING

Regression testing is a technique that tests software functionality after code updates. It ensures that previously developed and tested software still performs as expected.

Testing plays a crucial role for scientists working with Earth-systems models. For example, if a new commit is made to the model, various tests are run to ensure the new version of the model is still functional.



ABOUT ASSERT

Currently, many earth-systems models have their own independent testing tools.

ASSERT is a centralized, Python-based regression testing toolkit that aims to allow users to build regression testing workflows for numerous earth-systems models.

Key features:

- Test case management 
- Configuration using YAML files 
- Compatibility with Git repositories 
- Report generation 
- Easy integration of new models 



PROJECT RESPONSIBILITIES

When joining this project, though there was a lot of thorough effort put towards establishing documentation, Gitlab management, and prospective designs for ASSERT.

My primary responsibility was implementing the code infrastructure.





INFRASTRUCTURE I

Unit testing using PyTest

```
(base) deon.kouatchou@SAMDA-S1149 assert % python -m pytest
=====
test session starts =====
platform darwin -- Python 3.9.12, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/deon.kouatchou/ASSERT/Develop/assert
collected 113 items

test/lib/test_earthsystems_TestCase.py ...
test/lib/utils/test_access_repo.py .....
test/lib/utils/test_config.py .....
test/lib/utils/test_datatypes.py .....
test/lib/utils/test_forecasting_metrics.py .....
test/lib/utils/test_paths.py .....
test/lib/utils/test_time.py .....
test/models/test_model_e_reg.py .

=====
113 passed in 1.46s =====
(base) deon.kouatchou@SAMDA-S1149 assert %
```

Example of logging in Python

```
DEBUG:root:Debug message
INFO:root:Information message
WARNING:root:Warning message
ERROR:root>Error message
CRITICAL:root:Critical message
```

Source: [StackOverflow](#)

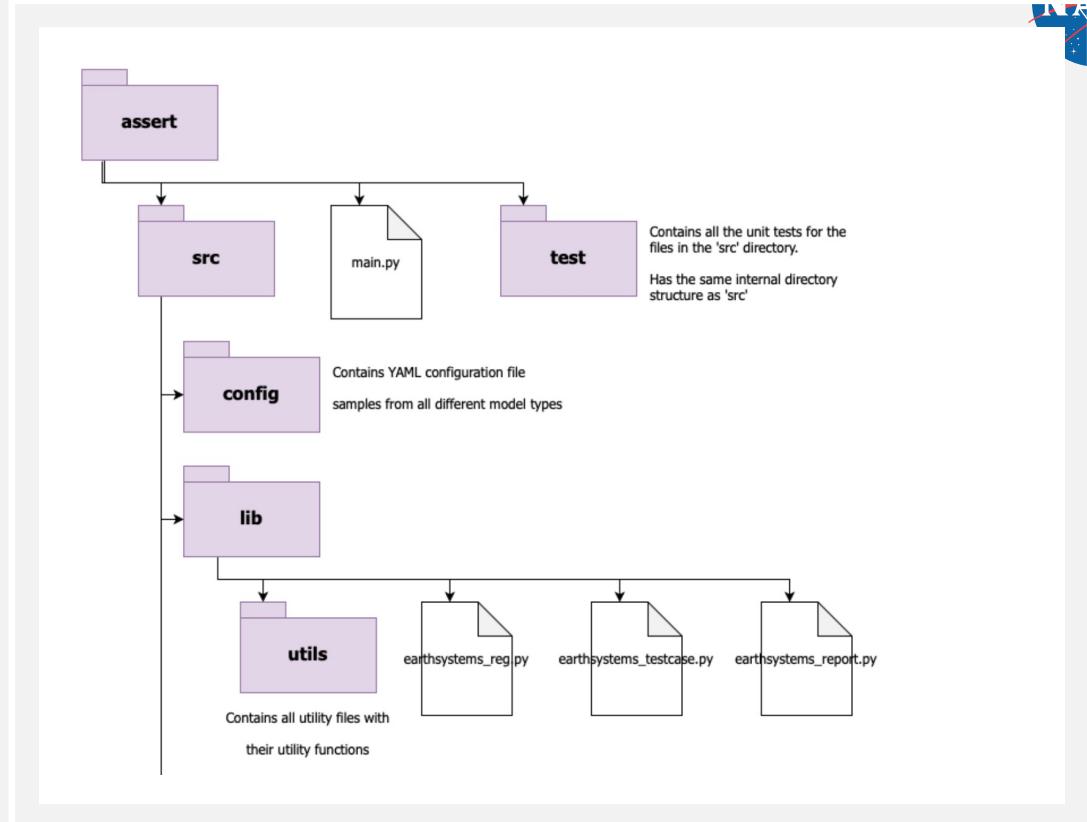
PREPARATION

One point of emphasis for this project was increased rigor and quality of our source code.

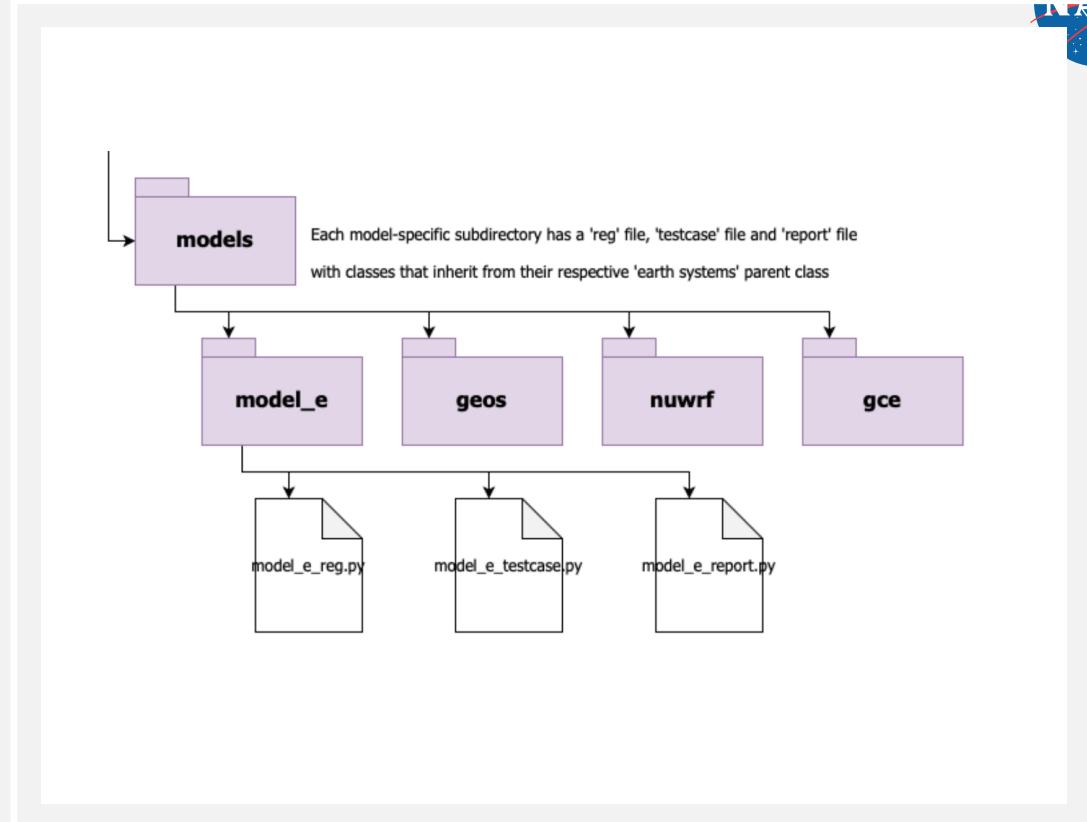
Thus, filesystem organization, Python PEP 8

compliance, logging, and unit testing were the focus of preparation for the main part of the code.

DIRECTORY STRUCTURE I



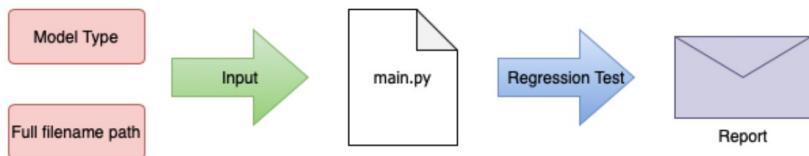
DIRECTORY STRUCTURE II



FUNCTIONALITY

Main Script

This flowchart displays the purpose of the main script in ASSERT: main.py



ASSERT's design is to input a model type and filename, run the appropriate regression test, and report the results.

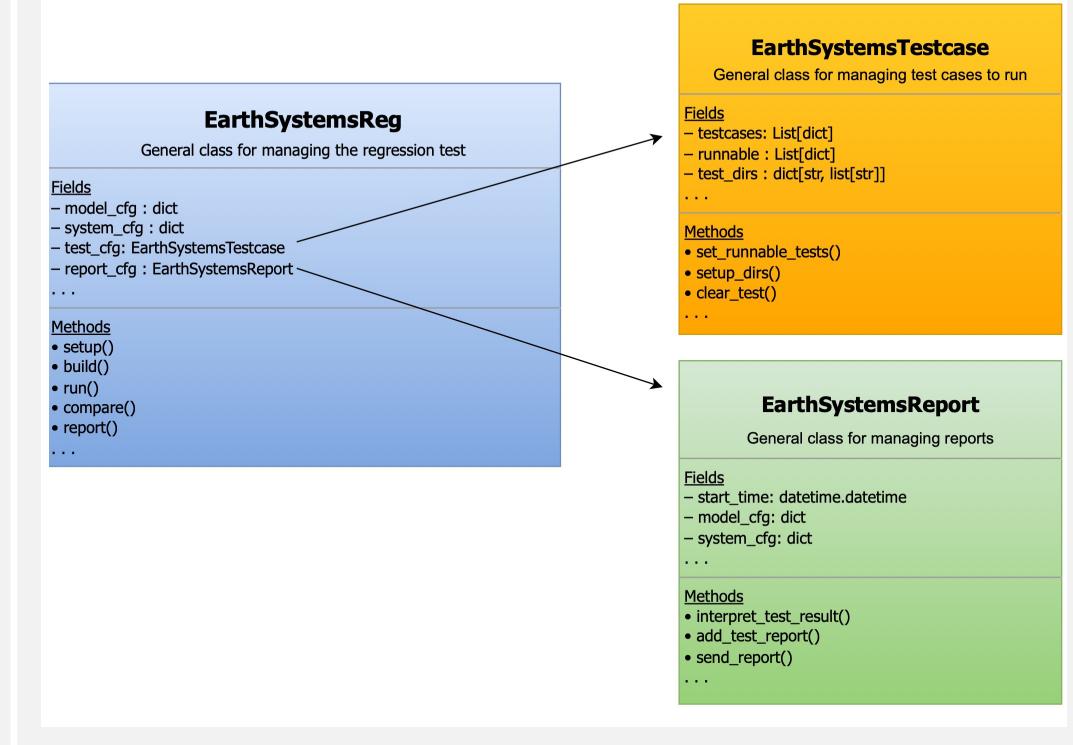
The next few slides overview how I implemented this framework using inheritance and object-oriented programming.



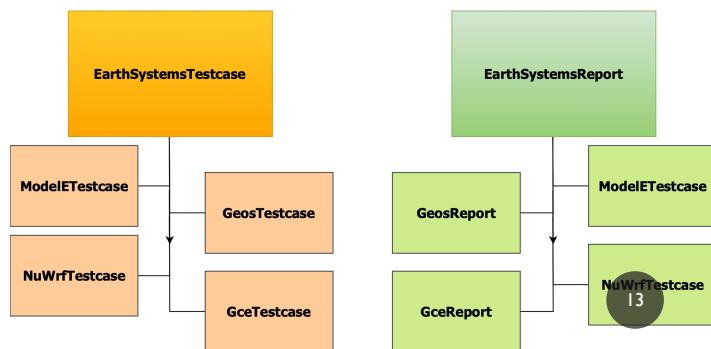
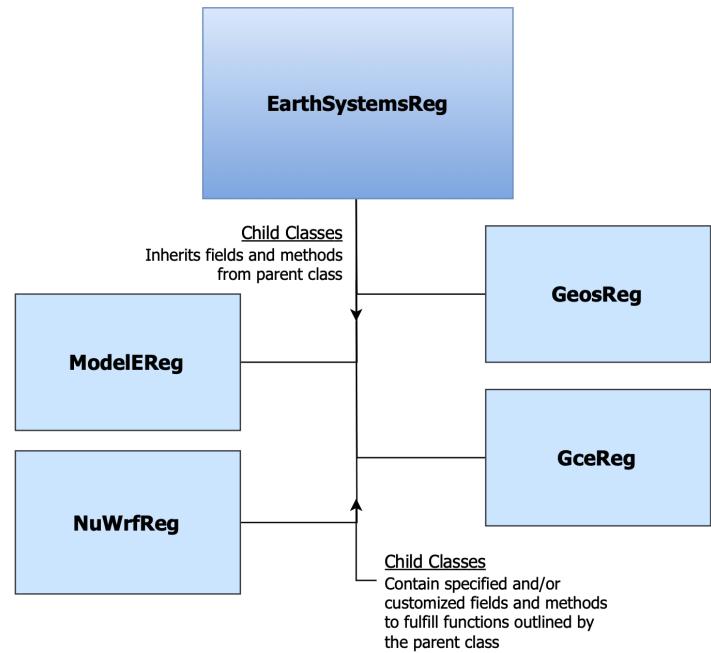
INFRASTRUCTURE II



LEVEL I: GENERAL CLASSES

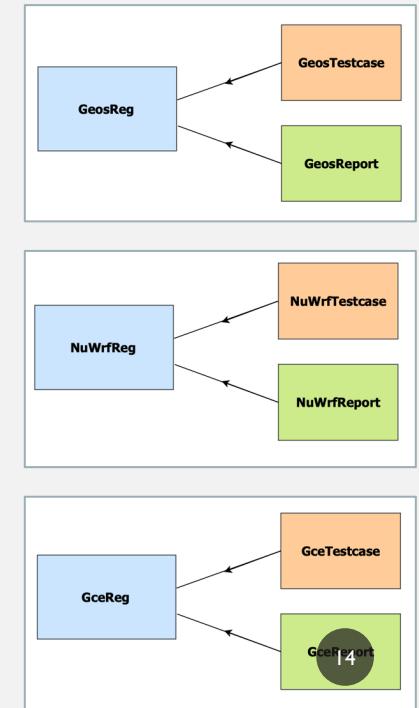
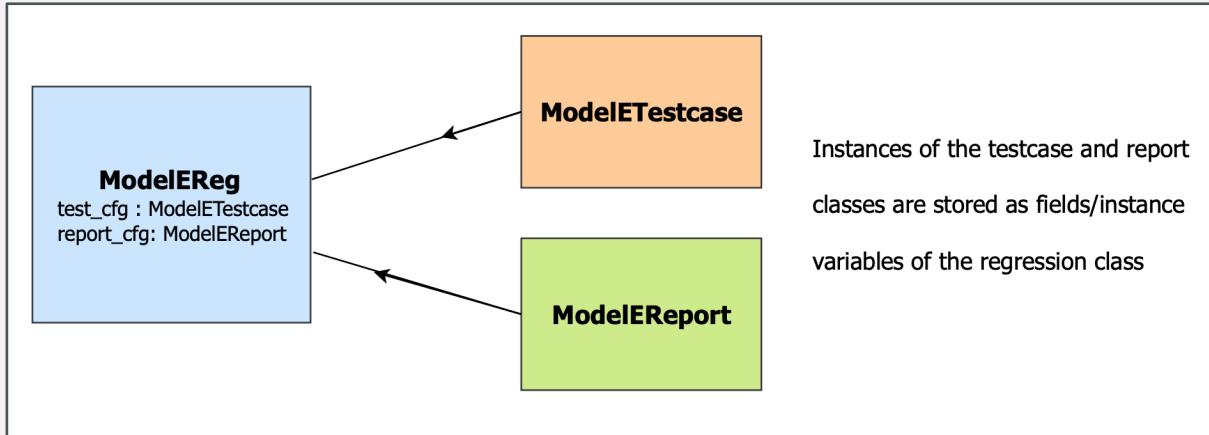


LEVEL 2: CLASS FAMILY TREES





LEVEL 3: MODEL-SPECIFIC CLASSES





DEMONSTRATION

* Running template not models themselves *



CONCLUSION

NEXT STEPS

- Full integration of earth-systems models into ASSERT:

- Writing regression testing code in the current infrastructure



- Writing integration tests for all the added models



- Writing utility functions and files to the infrastructure as needed



- Adjustment of logging as needed



- Formatting of email report



CHALLENGES

- Keeping track of all the inheritance occurring
- Maintaining balance between generality and specificity in the code design
- Creating adequate unit testing for complex utility functions



```
===== FAILURES =====
----- test_reset_scratch -----
>     def test_reset_scratch():
>         reg = ModelEReg(yaml_file=file1.name, start_time=dt.datetime.now())
>
test/models/test_model_e_reg.py:28:
```



REFERENCES AND ACKNOWLEDGEMENTS

Mentors and Colleagues:

- Carlos Cruz
- Jules Kouatchou
- Bruce Van Aartsen



Helpful Websites:

- GeeksforGeeks
- RealPython
- StackOverflow
- Python Documentation





THANK YOU