Powering Tests analysis using SWAN notebooks:
Towards a (semi)-automatic coherent analysis of all tests on all circuits

1st Part: The current state

M. Bednarek, Z. Charifoulline, A. Mnich

01/12/2022, MP3 Day
SIGMON (Signal Monitoring) - current state

- **What we have:**
  - ~50 notebooks for the analysis of the Powering Test
  - FPA notebooks for the manual analysis of quench
  - Most of the code is in the `lhcsmapi` library although can’t be considered as a proper api yet - many analysis code still coupled with the Jupyter environment
  - New ‘single-analysis’ packages (`lhcsmapi-qh`, `lhcsmapi-ee`, `lhcsmapi-pic`, …)
  - Library with the tooling around the notebooks - `lhcsmnb`
  - CI/CD (Continuous Integration/Continuous Delivery) pipeline on GitLab for the quality assurance and deployments (to EOS and AccPy repository)
  - Jenkins pipeline for the notebooks’ remote execution

- **How it is used by MP3 (and Proof of Concepts):**
  - Manual execution of the notebooks on SWAN with the ‘copy&paste’ from AccTesting
  - Script execution in the AccPy virtual environment on VM
  - Manual execution of the notebook on VM (within AccPy venv)
  - PyCharm on Windows (with Spark in the local mode)
Current state - Sigmon core and API

- Code coupling reduced - easier to debug, easier to use, resolving metadata, querying and processing can be used independently
- Query API:
  - signals from NxCals
  - signals and scalars from PM dumps
  - timestamps and sources of the dump events
  - tutorial notebook
- Processing API:
  - common transformations
  - tutorial notebook
- Analysis API:
  - generic interface of the Analysis and AnalysisManager
  - common approach to store and access signals and partial results
  - possibility to easily intercept all the warnings occurred
  - evaluation ongoing (new analysis code)
- QPS parameters - tutorial notebook
Current state - Notebooks’ Quality Assurance

- Dedicated job that executes all the notebooks on the top of the current PRO and current DEV and checks the results
- Possibility to manually trigger a job that would test all the notebooks on the top of any branch
- Documentation
- Lessons learned:
  - Sometimes certain lines are not flushed to the output (even though everything was executed)
  - Sometimes empty lines are inserted randomly to the output (parsing may fail)
  - Sometimes the Python kernel dies (race conditions in the kernel)
Current state - Notebooks’ Quality Assurance

\[ lhcdata/lhc-sm-hwc-docker-image \] with acc-py and openjdk11

- \[ \text{pytest} \]
- \[ \text{papermill} \]
- \[ \text{nbconvert} \]
- \[ \text{spark config} \]
- \[ \text{EOS} \]

1. **Preprocesses the notebook:**
   - injects a cell with the spark config
   - changes matplotlib backends
   - ignores tagged cells

2. **Injects the parameters**
3. **Executes the notebook**
4. **Compares with references**
5. **Generates:**
   - Executed notebook
   - HTML report
   - CSV files with analysis results

**Notebook Name Parameters**
Current state - Qualification, notebooks triggering

- A dedicated pipeline on the MPE SW team Jenkins instance
- Can be triggered manually from the UI
- Can be triggered using the Python API - [documentation](#)
- All the Sigmon notebooks that are currently maintained by the SW team can be parametrized and executed without any change

```python
import logging
from lhcsmbn import jenkins
logging.basicConfig(level=logging.INFO)
auth = (USERNAME, USER_TOKEN)

parameters = {
    'notebook': 'ipq/AN_IPQ_FPA.ipynb',
    'nb_type': 'FGC',
    'circuit_name': 'RQ10.L4',
    'timestamp_fgc': 1614319515000000000
}
jenkins.execute_notebook(parameters, auth, './')
```
Current state - Qualification, notebooks triggering
Current state - Deployments

- **AccPy Repository**
  - possibility to install lhcsmapi with `pip`

- **EOS**
  - DEV deployed nightly
  - PRO deployed during the release; all the previous versions available -> quick roll out possible
  - Possibility to deploy the venv linked to a specific branch

```
bash-4.2# echo PYTHONPATH
/eos/project/l/lhcsmapi/venv_SIGMON-176.13kA/cvmfs/projects.cern.ch/cryogenics/hepak/100_nxcals:/cvmfs/sft.cern.ch/lcg/views/LCG_100_nxcals/x86_64-centos7-gcc9-opt/ld/lib/python3.8/site-packages
bash-4.2# ls /eos/project/l/lhcsmapi
venv_copy
venv_dev
venv_gitlab_c1_notebooks_change
venv_json_schema
venv_master
venv
venv_pro_backup
venv_SIGMON-176.13kA
venv_SIGMON-176.13kA_slave_calculation
venv_SIGMON-329_analysismanager
venv_SIGMON-329_resurrect_monitoring_notebooks
venv_SIGMON-347_create_exemplary_notebook
venv_SIGMON-354_qp_tests
venv_SIGMON-397_replace_notebooks
venv_SIGMON-409_quench_detection_1pq
venv_SIGMON-418_check_deployment
venv_SIGMON-429_pro_d3
venv_SIGMON-429_pro_d3
venv_SIGMON-241_document_diffs_betweenApis
venv_SIGMON-241_document_diffs_betweenApis
venv_SIGMON-249_replace_signalTransformationBuilder
venv_SIGMON-290_replace_feature_query
venv_SIGMON-310_use-parameters-resolver
venv_SIGMON-320_add-tutorial-notebook
bash-4.2# ls /eos/project/l/lhcsmapi
analysis api gui __init__.py metadata __pycache__ pyedsl reference.py resources signal Time.py Timer.py utils
bash-4.2#
```
Future plans

● Reflects the roadmap written in March/April 2022
  ○ 6 work packages: Consolidation of the core and API, Automation, Consistency check, Storage, Metadata, Support for data analysts

● Address the problems identified later on:
  ○ Missing analysis code, criteria not implemented
  ○ Limitation of the curve to curve analysis and potential threats
  ○ Analysis criteria not coherent - lack of the analysis maintenance strategy
  ○ Lack of the clear separation (in the code and tooling) between the well defined analysis procedures aimed to be automated and the manual, interactive analysis
Future plans

WP1: Sigmone core and API consolidation
- WP1.1: Designing a data access API for all data sources
- WP1.2: Providing reusable generic analysis utility functions
- WP1.3: Providing reusable analysis blocks
- WP1.4: Separating the analysis logic from the presentation layer
- WP1.5: Coherent data sets ready for analysis

WP2: Automation
- WP2.1: Aggregation of result and communication with user / framework
- WP2.2: Consistency check between the automated and manual mode
- WP2.3: Integration in OP Python framework for manual execution
- WP2.4: Integration in UCAP for online analysis
- WP2.5: Integration in AccTesting for hardware commissioning
- WP2.6: Integration in PMA for event based analysis

WP3: Consistency check
- WP3.1: Analysis Consistency GUI
- WP3.2: Automated pipeline for consistency check

WP4: Storage
- WP4.1: Gathering requirements
- WP4.2: Storage for the preprocessed data aimed to be processed further
- WP4.3: Storage for the analysis results, reports

WP5: Metadata
- WP5.1: Provide circuit components metadata in external sources
- WP5.2: Integrate circuit components metadata from external sources
- WP5.3: Integrate metadata about powering tests from AccTesting
- WP5.4: Integrate PM metadata (signal lists and attributes)
- WP5.5: Enable the use of reference signals from previous tests
- WP5.6: Define and implement strategy to maintain metadata over time

WP6: Support for data analysis
- WP6.1: Energy Extraction analysis notebooks
- WP6.2: Monitoring notebooks
- WP6.3: IT string analysis notebooks
Future plans

● Automation
  ○ Making all the analysis automation ready (overall boolean analysis result is present, the procedure is followed strictly by the analysis code) - input from the experts is essential
    ■ PN0.d3 as an example (+ first attempt to have a combined analysis)
  ○ Automation solution not selected yet - solution might come ATS wide (CTTB task force)
  ○ Integration with AccTesting - work to be done on both AccTesting and Sigmon sides
  ○ Integration with PMA - decision not in our hands (but the direct support of Python by the framework more and more unlikely)

● Storage
  ○ Splice Resistances (Nxcals ?)

● Support for data analysts (actively participating in the development)
  ○ 13kA Energy Extraction for 600A (digital signals analysis)
  ○ Splice Resistance Monitoring (automated on GitLab ?)
  ○ Supporting analysis developers (Zinur, ?)
Summary

- **Current state:**
  - Rather stable environment (many fallbacks possible)
  - Quite usable APIs

- **Current developments:**
  - 13kA EE for RB/RQ ongoing
  - PNO.d3 ongoing
  - Code refactoring, bugfixing, documenting

- **Future plans:**
  - Automation readiness <- absolutely not yet but feasible for Jan 2024
  - Storage, semi-online analysis <- we have strong solution candidates, no development started
  - Analysis maintenance strategies <- discussions and agreements needed
End of part 1

- documentation
  - https://sigmon.docs.cern.ch/
- website
  - https://sigmon.web.cern.ch/
- GitLab
  - https://gitlab.cern.ch/LHCDATA/lhc-sm-api
  - https://gitlab.cern.ch/LHCDATA/lhc-sm-hwc
  - https://gitlab.cern.ch/LHCDATA/lhc-sm-analysis

lhc-signal-monitoring@cern.ch
https://mattermost.web.cern.ch/sigmon