Capture of provenance information : example of `logprov` for `gammapy`

Mathieu Servillat & J. Enrique Ruiz

l'Observatoire de Paris — LUTH
Laboratoire Univers et Théories

cnrs dépasser les frontières

PSL RESEARCH UNIVERSITY PARIS

PADC Paris Astronomical Data Centre

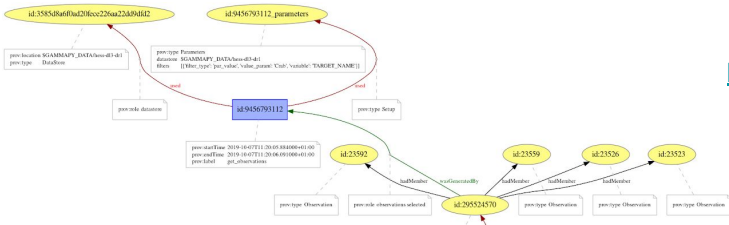
Objectives and requirements

- Capture provenance transparently
  - For Python tools (in particular data analysis on a laptop)
  - Make provenance info compatible between different software (ctapipe, gammapy, …)
  - minimum effort from the developers, reuse existing tools
- Development drivers
  - Use **logging** system
    - known by developers, already integrated (base Python package)
    - log **dictionaries** to structure the information
  - **Non-intrusive** addition of code
    - use **decorators** for classes and methods
  - Fill with useful information
    - **descriptions** in a `definition.yaml` file (activity description, parameters, usage, generation…)
    - log parameters, arguments...

Development within `gammapy`

- `logprov`, now an independent package:
  - https://github.com/mservillat/logprov
- `gammapy`: a community-developed, open-source Python package for gamma-ray astronomy:
  - https://docs.gammapy.org/
- **`gammapy` High Level Interface (HLI)**
  - Analysis class with methods
    - get_observations()
    - set_model()
    - run_fit()
    - …
  - Fixes the https://github.com/mservillat/logprov: 1 gammapy HLI method = 1 activity
- Démo...

Provenance in `gammapy`



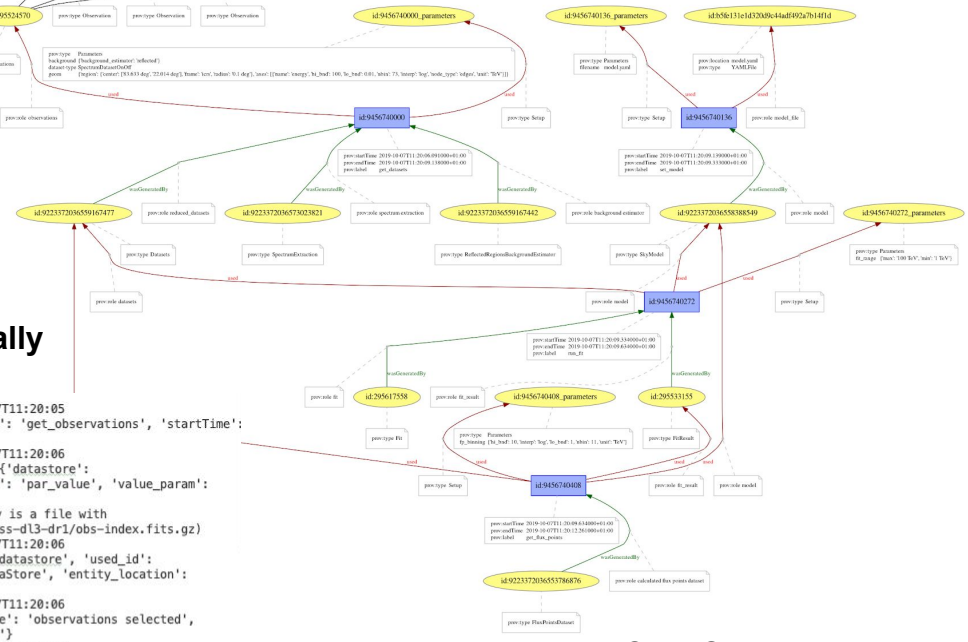
https://openprovenance.org/store/documents/1191.svg

1/ definition.yaml file for description/template
(already **integrated** to the code by the developers)

```
activities:

    get_observations:
        description:
            "Fetch observations from the data store according
            to criteria defined in the configuration"
        parameters:
            - name: datastore
              description: "DataStore path as string"
              value: settings.observations.datastore
            - name: filters
              description: "Filter criteria to select observations"
              value: settings.observations.filters
        usage:
            - role: datastore
              description: "DataStore object file"
              entityType: DataStore
              location: settings.observations.datastore
        generation:
            - role: observations selected
              description: "Observations selected"
              entityType: Observations
              value: observations
              has_members:
                  entityType: Observation
                  list: observations.list
                  id: obs_id
                  namespace: ""

    get_datasets:
        description: "Produce reduced datasets"
        parameters:
```

2/ entries **automatically** stored in the log file

```
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:05
  .884436_PROV_{'activity_id': 9456793112, 'activity_name': 'get_observations', 'startTime':
   '2019-10-07T11:20:05.884419'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
  .091102_PROV_{'activity_id': 9456793112, 'parameters': {'datastore':
  '$GAMMAPY_DATA/hess-dl3-dr1', 'filters': [{'filter_type': 'par_value', 'value_param':
  'Crab', 'variable': 'TARGET_NAME'}]}}
INFO:gammapy.utils.provenance.provenance:_PROV_The entity is a file with
  hash=3585d8a6f0ad20fece226aa22dd9dfd2 ($GAMMAPY_DATA/hess-dl3-dr1/obs-index.fits.gz)
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
  .091413_PROV_{'activity_id': 9456793112, 'used_role': 'datastore', 'used_id':
  '3585d8a6f0ad20fece226aa22dd9dfd2', 'entity_type': 'DataStore', 'entity_location':
  '$GAMMAPY_DATA/hess-dl3-dr1'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
  .091474_PROV_{'activity_id': 9456793112, 'generated_role': 'observations selected',
  'generated_id': 295524570, 'entity_type': 'Observations'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
  .091527_PROV_{'entity_id': 295524570, 'member_id': 23592, 'member_type': 'Observation'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
  .091571_PROV_{'entity_id': 295524570, 'member_id': 23523, 'member_type': 'Observation'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
  .091613_PROV_{'entity_id': 295524570, 'member_id': 23526, 'member_type': 'Observation'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
  .091653_PROV_{'entity_id': 295524570, 'member_id': 23559, 'member_type': 'Observation'}
INFO:gammapy.utils.provenance.provenance:_PROV_2019-10-07T11:20:06
  .091691_PROV_{'activity_id': 9456793112, 'endTime': '2019-10-07T11:20:06.091068'}
```



3/ export to W3C PROV
or search in provenance records

@ M. Servillat & J. E. Ruiz

Code: set up the capture

1/ YAML file following the IVOA
Provenance DM description part

```yaml
activities:

    get_observations:
        description:
            "Fetch observations from the data store according
            to criteria defined in the configuration"
        parameters:
            - name: datastore
              description: "DataStore path as string"
              value: settings.observations.datastore
            - name: filters
              description: "Filter criteria to select observations"
              value: settings.observations.filters
        usage:
            - role: datastore
              description: "DataStore object file"
              entityType: DataStore
              location: settings.observations.datastore
        generation:
            - role: observations selected
              description: "Observations selected"
```

2/ class decorator (or function decorator @trace)

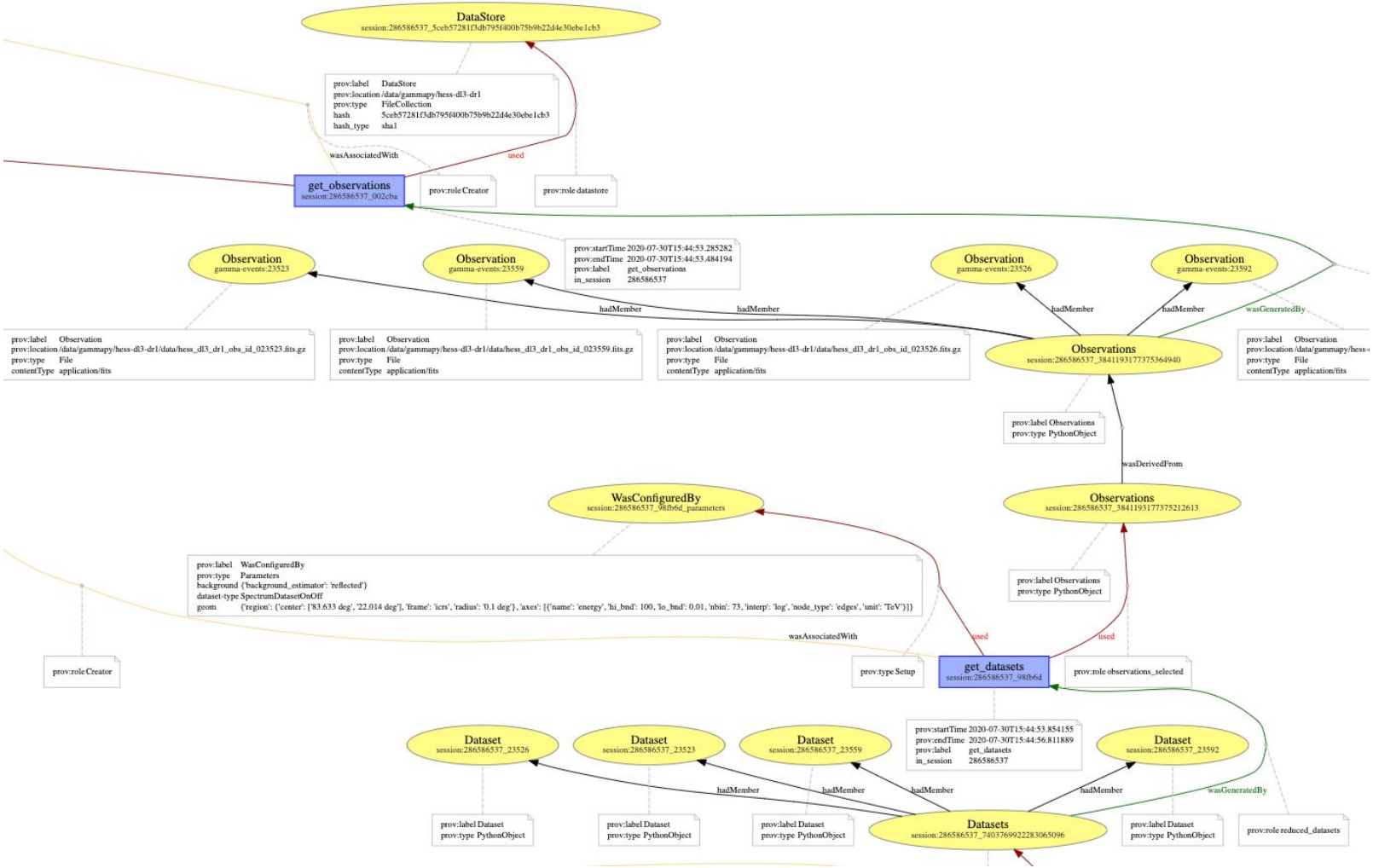
```python
prov_capture = ProvCapture(definitions=definition, config=provconfig)

@prov_capture.trace_methods
class Analysis:
    """Config-driven high-level analysis interface following gammapy structure
```

Code: what the user runs

```
# config = AnalysisConfig.from_template("3d")
config = AnalysisConfig.from_template("1d")
config.settings["general"]["logging"]["level"] = "INFO"
os.environ['GAMMAPY_DATA'] = '/data/gammapy'
# config.settings["general"]["logging"]["filename"] = logname
# config.settings["general"]["logging"]["filemode"] = 'w'
start = datetime.datetime.now().isoformat()
analysis = Analysis(config)
analysis.get_observations()
analysis.observations = copy.deepcopy(analysis.observations)
analysis.get_datasets()
with open("/data/gammapy/model.yaml") as f:
    model = yaml.safe_load(f)
analysis.set_model(model=model)
# analysis.set_model(filename="/data/gammapy/model.yaml")
# analysis.datasets["stacked"].background_model.tilt.frozen = False
analysis.run_fit()
# analysis.datasets = copy.deepcopy(analysis.datasets)
analysis.get_flux_points()
analysis.flux_points.write("flux_points.fits")
```

this transformation is not captured

model set with a filename or a dictionary


DataStore
session:286586537_5ceb57281f3db795f400b75b9b22d4e30ebe1cb3
prov:label DataStore
prov:location /data/gammapy/hess-dl3-dr1
prov:type FileCollection
hash 5ceb57281f3db795f400b75b9b22d4e30ebe1cb3
hash_type sha1
wasAssociatedWith
used
get_observations
session:286586537_002cba
prov:role Creator
prov:role datastore
prov:startTime 2020-07-30T15:44:53.285282
prov:endTime 2020-07-30T15:44:53.484194
prov:label get_observations
in_session 286586537
Observation
gamma-events:23523
Observation
gamma-events:23559
Observation
gamma-events:23526
Observation
gamma-events:23592
hadMember
hadMember
hadMember
hadMember
wasGeneratedBy
prov:label Observation
prov:location /data/gammapy/hess-dl3-dr1/data/hess_dl3_dr1_obs_id_023523.fits.gz
prov:type File
contentType application/fits
prov:label Observation
prov:location /data/gammapy/hess-dl3-dr1/data/hess_dl3_dr1_obs_id_023559.fits.gz
prov:type File
contentType application/fits
prov:label Observation
prov:location /data/gammapy/hess-dl3-dr1/data/hess_dl3_dr1_obs_id_023526.fits.gz
prov:type File
contentType application/fits
Observations
session:286586537_3841193177375364940
prov:label Observation
prov:location /data/gammapy/hess-
prov:type File
contentType application/fits
prov:label Observations
prov:type PythonObject
wasDerivedFrom
WasConfiguredBy
session:286586537_98fb6d_parameters
Observations
session:286586537_3841193177375212613
prov:label WasConfiguredBy
prov:type Parameters
background {'background_estimator': 'reflected'}
dataset-type SpectrumDatasetOnOff
geom {'region': {'center': ['83.633 deg', '22.014 deg'], 'frame': 'icrs', 'radius': '0.1 deg'}, 'axes': [{'name': 'energy', 'hi_bnd': 100, 'lo_bnd': 0.01, 'nbin': 73, 'interp': 'log', 'node_type': 'edges', 'unit': 'TeV'}]}
prov:label Observations
prov:type PythonObject
wasAssociatedWith
used
used
prov:role Creator
prov:type Setup
get_datasets
session:286586537_98fb6d
prov:role observations_selected
prov:startTime 2020-07-30T15:44:53.854155
prov:endTime 2020-07-30T15:44:56.811889
prov:label get_datasets
in_session 286586537
Dataset
session:286586537_23526
Dataset
session:286586537_23523
Dataset
session:286586537_23559
Dataset
session:286586537_23592
hadMember
hadMember
hadMember
hadMember
wasGeneratedBy
prov:label Dataset
prov:type PythonObject
prov:label Dataset
prov:type PythonObject
prov:label Dataset
prov:type PythonObject
Datasets
session:286586537_740376992283065096
prov:label Dataset
prov:type PythonObject
prov:role reduced_datasets

Implementation details

- Identifiers and unicity...
  - the identifier is recomputed from the entity each time it is used
  - id(), hash(), str(), file hash (md5, sha1…)
- Parameter values directly attached to the activity
- Check for entity modifications before usage
  - indicate a derivation if entity has changed (but no more information about what happened)
- session identifier
  - when the Analysis class is instanciated
  - place to keep the system information, the global configuration, ...
- limits
  - only wrapped methods are traced
  - need definition for usage/generation

OPUS + `gammapy`

- 1 OPUS job
  - Runs several `gammapy` functions
  - Stores result entities
- Internal provenance
  - Store objects
  - **Link to OPUS job**
    - Sub-activities ?
    - W3C PROV Bundle ?
  - **link to result**
    - Stored in OPUS archive
    - Derivation ?
    - Copy ?