Introduction to PYROS

VERSION: 28/01/19

This document describes the components of a robotic observatory and set the terminology used in PyROS.

1. What is PyROS

PyROS is a software suite to drive an autonomous robotic observatory. Many features are included in PyROS code:

- Start and stop observations according security criteria.
- Slew a mount and drive instruments to acquire data using commands of hardware controllers.
- Manage several scientific programmes during a same night. Observation time is dynamically shared according quotas and priorities of every programme.
- Interruption of the current observation and start immediately new observations when an alert is received (gamma-ray bursts, gravitational wave, etc.).
- PyROS can be used in platforms Linux, MacOS, Windows.

2. Description of a Robotic Observatory

An observatory is constituted by a building that houses one or more telescopes.

A telescope is constituted by a mechanical support (the mount) that is equipped by an instruments (e.g. one optical tube and a camera).

A robotic observatory, in sense of PyROS, is autonomous. This means that all devices can be driven by software without human operation during observations.

PyROS can be seen as the pilot of the robotic observatory.

In the present context, one PyROS instance pilots only one mount and its associated instruments.

The security of the observatory is a crucial point. The building must be able to put a cover (rolling roof or a dome shutter) over the mount to protect it according weather conditions. To do that there are three kinds of security:

Passive security: The actions of security are ordered by PyROS according the reading of various sensors (rain, wind, etc.).

Active security: The actions of security are done by an independant controller and PyROS can read the status of the controller to decide if observations can be performed.

Semi-active security: The controller is autonomous regarding its security criteria but it can be also set in security mode by PyROS due to additional criteria.

2.1. Hardware

PyROS uses the following terminology:

- **Mount**: A mount is a mechanical support that could be oriented by motors. In PyROS we do not use the word telescope because it is ambiguous if we consider a telescope as only a mount or is a mount and the optical tube + instruments. In PyROS definition, a mount is not the optics (mirrors, lenses) but only mechanics.

- **Channel**: A channel is an instrument (optics, filters, detector) attached to a mount. The detector is often a camera. If we install two optical tubes on a given mount, there are two channels. So a channel includes the optics.

- **Unit**: A unit is a set of one mount + one or more channels. The unit explains which channels are currently operated on a given mount.

- **Node**: A node is composed by a set of mounts and channels.

For example, an observatory have three mounts, three optical tubes and four cameras. All these hardware are described in the Node. Astronomers want to use one of the mounts with one optical tube and two cameras. This association is described as a Unit. We can imagine that after few months one camera will be replaced by another one ever described in the Node. This situation is realistic in telescope networks where hardware is rapidly replaced by a spare when a problem appears.

2.2. Astronomers

PyROS uses the following terminology:

- **PI-Unit**: Principal investigator (PI) of the unit. A human responsable of the unit.

- **Period**: Several months (typically 6 months) during which the hardware remains the same and scientific programmes are defined.

- **Partners**: Organizations that share the observation time in percent. This situation often occur when several organizations funded the observatory. Time is then shared according the funds of each partner. Partner list is defined for a given period.

- **Proposal**: Proposition of a scientific programme. A proposal must explain who is the PI of the scientific programme, the partner and the total duration of asked time to observe over the period. The proposal living time is one period.

- **Scientific Programme**: It is an accepted proposal. Accepted means for example validated by a Time Allocation Committee or by the PI-Unit.

- **PI of Scientific Programme**: There is one PI for one scientific programme. Other users can exists and are named Co-Is. PI of scientific programme is responsible of observations of his/her programme.

When a scientific programme is accepted an amount of seconds is initialized at the beginning of the period. Then, users of a given programme can send observations requests to feed the scheduler of PyROS.

2.3. Data acquisition

PyROS uses the following terminology:

- **Frame**: One read of the detector (can be a matrix of values or a single value according the detector).

- **Plan**: A series of exposures (frames) of same exptime, binning, filter, acquisition mode.

- **Album**: A series of plans for a given channel.

- **Sequence**: One pointing followed by a set of albums.

- **Request**: A group of sequences

An raw image will result of one frame or a combination of frames of the same plan.

A given frame has meta-data coming from the configuration of the unit and from the environment conditions (weather, image quality, etc.).

2.4. Software

PyROS is based on a classical distribution of tasks to several agents. A given agent is designed to perform only one basic task (for example controlling the positions of the mount). The agents communicate commands and status between them.

Agent status and commands are stored in a database. The PyROS database is the central point of PyROS agents. The database contains the configuration of the unit components, the useful data collected by agents (meteo, etc.) and the commands exchanged between agents.

One PyROS session can pilot only one unit. So the PyROS scheduler generate observations plannings for one unit only.

Alerts are received and managed by a dedicated agent.

3. Use PyROS

We distinguish three kind of people that can use PyROS in differents manners:

- Astronomers: They are the end users. They don’t have to understand the machinery behind the web pages they use to send observations requests and to retrieve their acquired data. Astronomers can also check the observatory status to follow the acquisitions of their data via a web page. Anyway, every interaction of astronomers and PyROS are done through web pages.

- Developers: They are coding the PyROS code. PyROS code is open source but it is important to respect some coding rules.

- Maintenance staff: They are technicians, often available in the site of the observatory and able to relaunch PyROS or to do some basic actions when problems occur.

3.1. Configure PyROS

Configure PyROS can be done by developers or by maintenance staff.

A given unit is described by an XML file.

Periods and partners are defined in the section <Mount>.

Here is an example:
TBD.

3.2. Running PyROS

Running PyROS consists to launch agents. The python script pyros.py is used to launch agents. For example:

```
$ python3 pyros.py start majordome -unitfile /myconfig/conf_unit_default.xml
```

Running PyROS can be done by maintenance staff. Normally, PyROS agents must be automatically launched when the computer is booted (i.e. cron for a Linux OS).

3.3. Developing PyROS

PyROS developer must know the Python language, the Django framework and MySQL server management. Please use the two following guides for programmer if you are interested to add/modify codes in PyROS:

- **PyROS Developer guide**
- **PyROS Specifications**

An additional document available on Redmine at IRAP indicates the current status of the development plan.

3.4. Astronomers using PyROS

If you are an astronomer interested by observing a unit driven by PyROS, read the following guide:

- **PyROS Astronomer guide**

3.5. Maintenance of PyROS

If you are involved in the maintenance of a unit driven by PyROS, read the following guide:

- **PyROS Maintenance guide**

3.6. Cite PyROS in papers

A dedicated paper will be written:

Pallier et al. “PyROS a software to drive autonomous robotic observatories” Experimental Astronomy xxx, yyyy, (2021).