Modifying a Reflex Workflow

Public

John Pritchard
**ESO-Reflex and Kepler**

**EsoReflex** is the *ESO Recipe Flexible Execution Workbench*, an environment to run ESO VLT pipelines which employs a workflow engine (*Kepler*) to provide a real-time visual representation of a data reduction cascade, called a workflow, which can be easily understood by most astronomers.
Glossary

**Actors:**
- (mostly) Green folder like things…
- ‘things’ that act on ‘stuff’, in an EsoReflex workflow, the important actors are the **recipe executors** and the graphical display actors.

**Connections:**
- The lines joining things up
- connect ‘things’ allowing the ‘stuff’ coming out of one ‘thing’ to ‘flow’ to another ‘thing’

**Relations:**
- The black diamonds here and there…
- allow ‘stuff’ to be sent to/from multiple ‘things’
The following is being done on a Mac, with fors-5.3.31 and giraf-2.16.2 workflows, installed via MacPorts

If working on RPM based installation replace:

```
/opt/local/ → /usr/
```

If working on a install-script based installation replace:

```
/opt/local/ → <dir_spec>/install/
```
SOFs – The ‘stuff’ that flows…

- SOFs: Set Of Files
  - original concept from esorex, a Filename+Category, e.g. bias.fits MASTER_BIAS
  - Extended for esoReflex by adding “purposes”, checksums, class…

- SOFs are the main objects that flow around an esoReflex workflow
  - they are passed from the output port of one actor to the input port one or more other actors “via” the connectors.
  - The actors act on the input SOF(s), processing the files and/or modifying the SOF that they then output
A number of the standard Kepler actors are useful, e.g. to set variables, or to replicate objects...

But, most of the actors used in the esoReflex workflows are ESO specific and are found in the “Esoreflex” folder

To add an actor to a work flow, simply drag and drop from the side bar onto the workflow…
Step 0: open the WKF and export as…

- Open the standard fors_spec workflow and then export as XML to a new directory.
- Then run the recipe selecting just the 4th dataset (which has a nice set of cosmic rays...)
Step 1: add a ‘simple’ actor

- Select the ‘DataFilter’ actor from the Esoreflex/Workflows.kar folder
- Drag it on to the workflow under the ForsCalib actor
- Disconnect the Connection from the ‘sci’ port of the master FitsRouter and re-connect it to the ‘sof out’ port of the DataFilter actor.
- Create a new connection between the ‘sci’ port of the master FitsRouter and the ‘sof in’ port of the DataFilter actor
FORS Workflow For Spectroscopy Data (v. 5.3.31)

Workflow Instructions

To run this workflow on the data:
- Push on highlighting. Choose "Data" or "Metadata of this data". Each step must have one or two options.
- Press the "Run" button OR click on the workflow.

To run on a different data set:
- Click on DATA, X, Y, and select the appropriate.
- All names of OBS DATA_X, Y will be searched by the workflow.
- Select the workflow CATALOG DATA_X, Y if you do NOT want to use the каталог data associated with the observer.

Input:
- DATA, X, Y will be searched for data.
- There is a chance that the catalog data associated with the observer.
- The data in the catalog table should be a combination of the catalog data and the catalog metadata.

Output:
- DATA, X, Y will be searched for the catalog data associated with the observer.
- There is a chance that the catalog data associated with the observer.

Step 1: Data Organisation and Selection
- Creation of Master Calibration Files
- Response computation
- Science Reduction
- Output Organisation

Step 2: Data Organisation and Selection
The end…

And that’s “How to modify a workflow…”
But seriously…

…Not the end…
Step 2.1: Add a basic Python Actor

- Select the ‘PythonActor’ actor from the Esoreflex/Scripting.kar folder
- Drag it on to the workflow beside and to the left of the DataFilter actor
- Note it (currently) has no ports…
- In a terminal do:
  ```
  cp /opt/local/share/esopipes/giraf-2.16.2/reflex/giraf_in_out.py fors_in_out.py
  ```
- Edit it (double click, or right click and select Edit) and set ”Python script” to fors_in_out.py
- Note how the two ports appear…
Step 2.1: Add a basic Python Actor

- Disconnect the Connection from the ‘sof in’ port of the DataFilter actor and re-connect it to the ‘sof out’ port of the PythonActor actor.

- Create a new connection between the ‘sof out’ port of the PythonActor actor and the ‘sof in’ port of the DataFilter actor.

- Right button Click on the PythonActor actor and select ‘Customize name’.

- Set the name to InOut.

- The InOut actor doesn’t do anything, except write the in and out SOFs to the bookkeeping directory.
Step 2.2: Add a Recipe Executor

- The Esoreflex Recipe Executor is a special case.
- There is no template actor to drag and drop.
- Instead… as per the Reflex Workflow Development Guide (https://www.eso.org/sci/software/reflex):
  
  The RecipeExecutor should be instantiated using the Tools -> Instantiate Component option. The component should be called org.eso.RecipeExecutor. This will show a list of available recipes as seen by the esorex command which is in the current path. If your recipe is not shown there, check your esorex and pipeline installation.

- Add inside a CompositeActor, together with a sofSplitter and a sofAccumulator.
Step 3: Copy paste giraf CRC

- Remove the fors_bias actor.
- Open the giraf workflow
- Select and copy the ‘Cosmic Ray Cleaning’, the SofCombiner and the FitsRouter actors just before and just after the CRC actor, and the connections between these three
- Paste into the fors_spec workflow, and move into position below the DataFilter actor.
Step 3: Copy paste giraf CRC

- Make a new connection from the ‘sof in’ port of the SofCombiner to the relation just after the MasterBias actor

- Reconnect the ‘sof out’ port of the InOut actor to the ‘sof in’ port of the SofCombiner

- Make a new connection from the ‘SCIENCE’ port of the FitsRouter to the ‘sof in’ port of the DataFilter
Ok so it ran past the CRC actor without crashing, but then the DataFilter is empty and continuing the science recipe crashes because there is no science file. So what’s wrong?

In a nutshell, the GIRAF science data have category SCIENCE, while the FORS spec’ data have one of SCIENCE_MXU, SCIENCE_MOS or SCIENCE_LSS. So this needs to be fixed in the workflow and the python script.

Check the categories of the science coming out of the master FitsRouter
Step 4: adapt giraf to fors

- Edit the parameters of the FitsRouter just after the CRC actor

- Add a parameter:
  - Class = ptolemy.data.expr.Parameter
  - Name = SCIENCE_config (the name of the port plus "_config")
  - Default value = "SCIENCE_MXU,SCIENCE_MOS,SCIENCE_LSS"

- Open (not edit) the CRC actor and do the same to the FitsRouter actor just after the Input port
Step 4: adapt giraf to fors

- Edit the parameters of the Astroscrappy actor
- Copy the name of the Python script.
- In a terminal, copy that python script file to the local directory:

  ```
  cp /opt/local/share/esopipes/giraf-2.16.2/reflex/giraf_astroscrappy_multi.py fors_astroscrappy_multi.py
  ```

- Adapt the python script appropriately… 😊

- Set the Python script to the `fors_astroscrappy_multi.py`, make sure that it has the full path to the file.
Workflow to clean Cosmic Rays

Three strategies are used to clean Cosmic Rays (bad pixels). The first one is to median combine all incoming frames. The second and third perform Cosmic Ray detection and interpolation of each individual frame using Python modules PyCosmic and AstroScapppy.

Note that a (different) action is added to the purpose of each output frame so that they can later be separated for separate science processing.
And that really is “How to modify a workflow…”
Exercises for the workshop participants…

- Implement the PyCosmic and Median+Pass-Thru
- Implement the handling of multiple versions of the same RAW file in the ForsScience actor
- Implement the handling of multiple versions of the same RAW file in the ForsScience actor GUI interface

Actually I’ve done all this (except the last step) but it is a bit beyond the scope of today’s presentation… but I hope it will find its way – when finished with the last step – into the public release of the workflow (if not for 2018, then at least for 2019).