ADA7: Reflex Tutorial

- 9:00 – Introduction and trouble shooting
- 9:10 – Overview of Reflex (Wolfram)
- 9:30 – UVES data reduction (Dan)
- 10:30 - Coffee!
- 11:00 – Advanced features of Reflex (Enrique)
- 11:30 – Modifying Reflex workflows (Wolfram/Enrique)

[ 12:10 – XShooter workflow demonstration (Dan) ]

Red = Hands-on session
The ESO Recipe Flexible Workbench

REFLEX

http://www.eso.org/sci/software/pipelines/reflex_workflows

Wolfram Freudling
Science Data Products
Reflex Project

Make pipeline recipes accessible to general users

- New Environment to run ESO VLT pipeline “recipes”
- Gives users enough information how to run recipes
- ESO Reflex Team:
  - Science Guidance by Science Data Products Group: Wolfram Freudling, Daniel Bramich, Sabine Moehler
  - Implementation of Reflex Environment by Data Flow Infrastructure Department: Vincenzo Forchi, Enrique Garcia, Pascal Ballester
  - Implementation of workflows by Pipeline Systems Department: Enrique Garcia, Andrea Modigliani, Pascal Ballester
  - Support via ESO User Support: usd_help@eso.org
Reflex is based on Kepler Workflow engine

- Kepler is freely available under the BSD License.
  - [https://kepler-project.org](https://kepler-project.org)

- Used in life science, ecology, geology

- Kepler provides a graphical user interface (Java)

- run--time engine that can execute workflows either from within the graphical interface

- Current Kepler version 2.3
Workflow Driven Data Reduction

- Manual data reduction (e.g. IRAF):
  - Classify data (e.g. define keywords)
  - Organize data (e.g. decide what biases to use for different data)
  - Run recipe (task), use output as input for next recipe, ...
  - In case of problem:
    - fix problem (e.g. parameters)
    - identify which products are affected
    - re-run recipes

- Workflow driven data reduction:
  - Define rules to classify and organize data
  - Recipes automatically execute when input data are available
  - In case of problem:
    - fix problem (e.g. parameters)
    - Re-run workflow: only affected recipes are executed
Basic Reflex Workflow

- Green boxes represent actors, tokens are sent between actors
- Reflex uses SOFs as tokens
- SOFs are set of files, includes filename, categories + purpose
- Data Organizer interprets OCA rules
- Organizes data in “DataSets”
- A DataSets are SOFs that include everything needed to process one set of science observations + recipe parameters
- DataSets are selected with Data Set Chooser
REFLEX OCA Rules

classifying, organizing and associating astronomical data based on their meta-data (FITS keywords).

- Proprocessing steps imply Data organisation
- Data organisation define in “OCA rules”: text file

- Three types of rules:
  - Classification („This is a Raw Dark“)
  - Organization („This set of darks triggers creation of master darks“)
  - Association (“Creation of master darks needs Raw Darks and Masterbias“)

- DO produces DataSets: A dataset includes ALL files needed to process a set of science data, including calibrations of calibrations.

- Each file in DataSet has a category (e.g. “raw bias”) and a purpose action1/action2/... (e.g. „MasterBias/MasterDark“).
FITS Router

sorting by category

routing by category is explicit.

this is intuitive: each recipe needs well defined input (e.g. category raw biases) and creates well defined output (e.g. category Masterbias).

relations determine work „flow“.
SofSplitter: Sorting by Purpose

- Sorting by Purpose is implicit.
- Data Organiser determines Purpose from information in OCA rules.
- Assumes that each recipe has at least one unique file category.