

UalGetCpoBundle Workflow

December 17, 2020

Contents

1 Input to UalGetCpoBundle Actors	1
2 Design of the UalGetCpoBundle-Workflow	1
3 Design of the UalGetCpoBundle Actor	2

Contact: Thomas Johnson, tjohn@kth.se .

Modelling real plasmas requires input from both the relevant experimental database and information about the machine hardware and settings. The purpose of the UalGetCpoBundle is to provide a workflow that reads both input CPOs related to the plasma state and the once that are related to the hardware. The workflow then bundles these CPOs into the CPOS-branch of the [EU-ITM bundle](#)¹. The target applications of this workflow is as an initialisation actor in workflows like the HCD workflow and the ETS.

The actor is still under development and need to be tested. So far only the wall, the nbi and the antennas CPOs are read from the machine description database, but more CPOs will be added (e.g. pfsystems and ironmodel). Note that the UalGetCpoBundle does not fill in the discharge related hardware settings. These have to be set externally with actors like the addECant, addICant and nbifiller.

1 Input to UalGetCpoBundle Actors

The UalGetCpoBundle actor has a number of input ports, through which all input to the actor are passed. The list of input ports reads:

Port name	Datatype	Description
UserName	String	Name of the user whos UAL database we shall read from
MachineName	String	Name of the machine (tokamak) used when accessing the UAL database
ShotNumber	Integer	Shot number to be used when reading from the UAL database
InputRunNumber	Integer	Run number to be used when reading the main CPOs from the UAL database
TemporaryRunNumber_MainCpos	Integer	Run number used when the UALInit temporarily store the main CPOs during a run. Also referred to as <code>runwork</code> .
TemporaryRunNumber_MachineDescription	Integer	Run number used when the UALInit temporarily store the Machine Description CPOs during a run. Also referred to as <code>runwork</code> .
SelectMachineDescriptionCpos	Bundle	Bundle of three boolean fields: {use_MD.wall=True/False, use_MD.nbi=True/False, use_MD.antennas=True/False} The three field are used to select if the wall, the nbi and the antennas CPOs should be taken from the Machine Description Database or from the shot=ShotNumber and run=InputRunNumber .

2 Design of the UalGetCpoBundle-Workflow

The UalGetCpoBundle-Workflow has been developed purely for the testing and development of the UalGetCpoBundle actor. This workflow (1) provides an interface for prescribing the input to be used by the UalGetCpoBundle actor and (2) it prints the output from the same actor.

To fill the input data double click on the composite actor at the top-level of the workflow. A box will then

¹https://www.efda-itm.eu/ITM/html/itm_conventions.html#itm_conventions_plasma_bundle

pop up where you can fill in each of the input parameters described above, as illustrated in figure 1. These parameters are then fed into the actor as illustrated in figure 2.

This is a test workflow for the composite actor:
 -- UalGetCpoBundle--



To adjust the input, double click on the actor below.

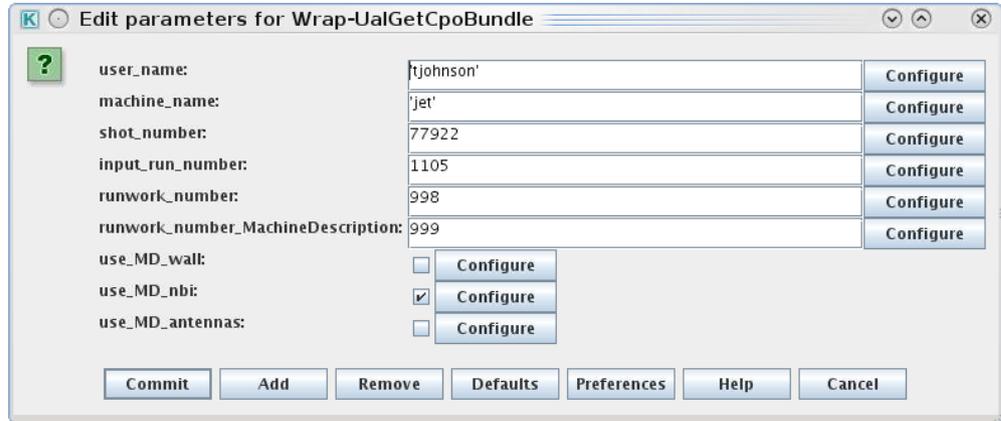


Figure 1. The top level of the UalGetCpoBundle-Workflow provides an interface for describing the input to the UalGetCpoBundle actor.

- user_name: 'tjohnson'
- machine_name: 'jet'
- shot_number: 77922
- input_run_number: 1006
- runwork_number: 998
- runwork_number_MachineDescription: 999
- use_MD_wall: false
- use_MD_nbi: false
- use_MD_antennas: false

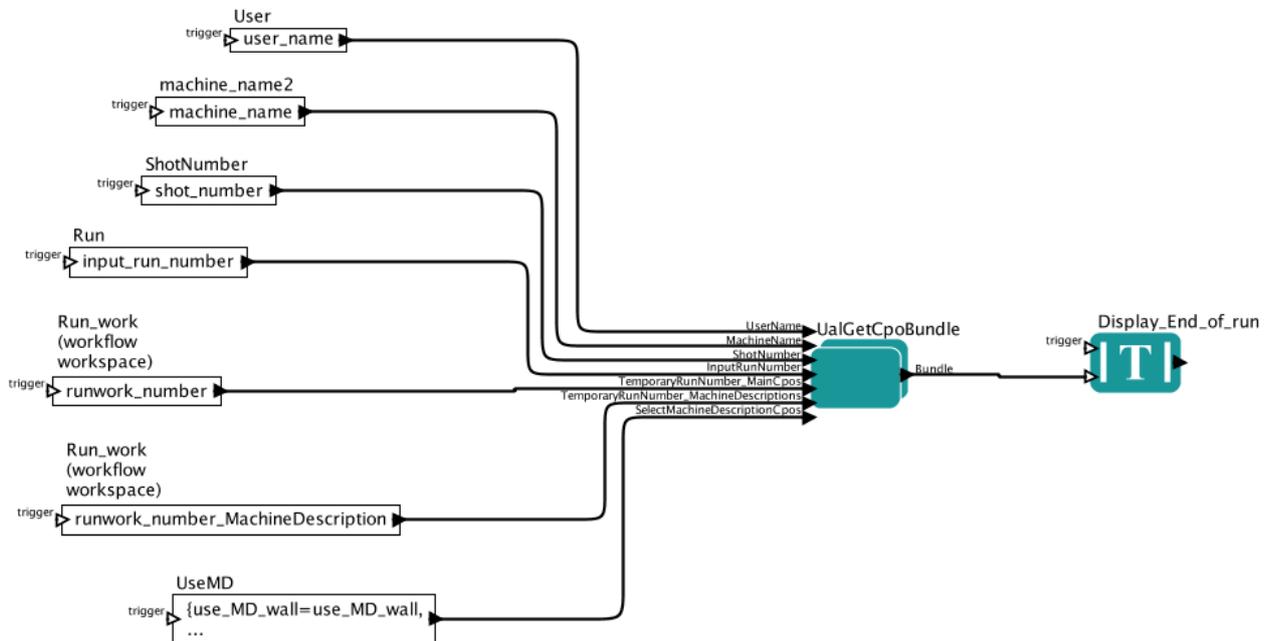


Figure 2. Calling the UalGetCpoBundle actor in the UalGetCpoBundle-Workflow .

3 Design of the UalGetCpoBundle Actor

The UalGetCpoBundle actor, outlined in the figure 3 below, consists of four composite actors and some logic to connect these.

- The first of these actors is `Validate_Input` (see figure 4), which checks the three input run numbers to make sure they are all different.
- The second actor reads all the main CPOs from the input run number `InputRunNumber` and is called `UAL_read_main_CPOs` (see figure 5).

After the third actor there is a switch to determine which Machine Description data is requested and if we need any of the machine description CPO. If we do not, the output from `UAL_read_main_CPOs` will be sent to the output of the `UalGetCpoBundle` actor, otherwise the third actor `UAL_read_machine_description_CPOs` is called.

- This third actor (see figure 6 and 7) first identifies the relevant triplet of `UserId/Shot/Run` to be used when reading data from the machine description database. This procedure uses an xml-document from the `md_and_dm` -repository that related e.g. JET shot 77922 with a certain set `UserId/Shot/Run`. Once this triplet is known the `UALInit` is called and its output is finally bundled.
- The fourth and final composite actor is `Insert_Machine_Descriptions_to_CPO_bundle`, which merges the Machine Description CPOs according to the specifications in the input port `SelectMachineDescriptionCpos`.

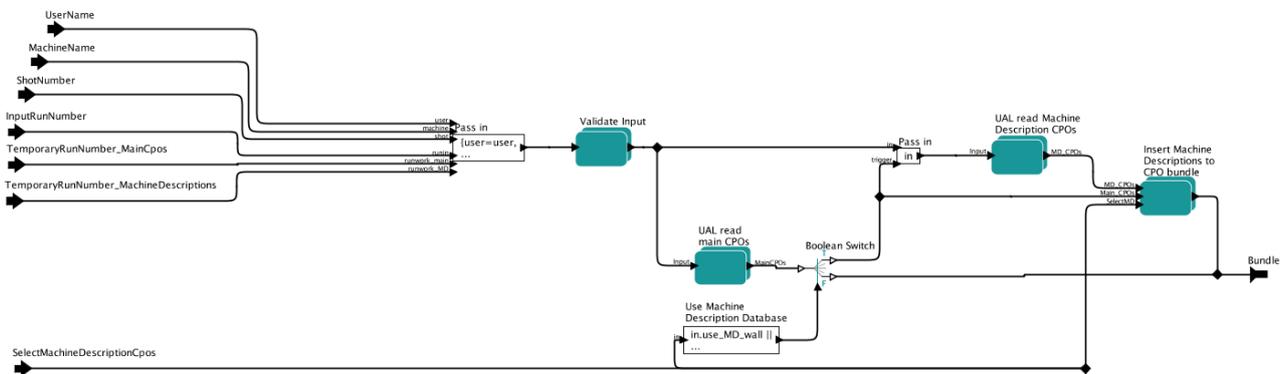


Figure 3. Outline of the `UalGetCpoBundle` actor.

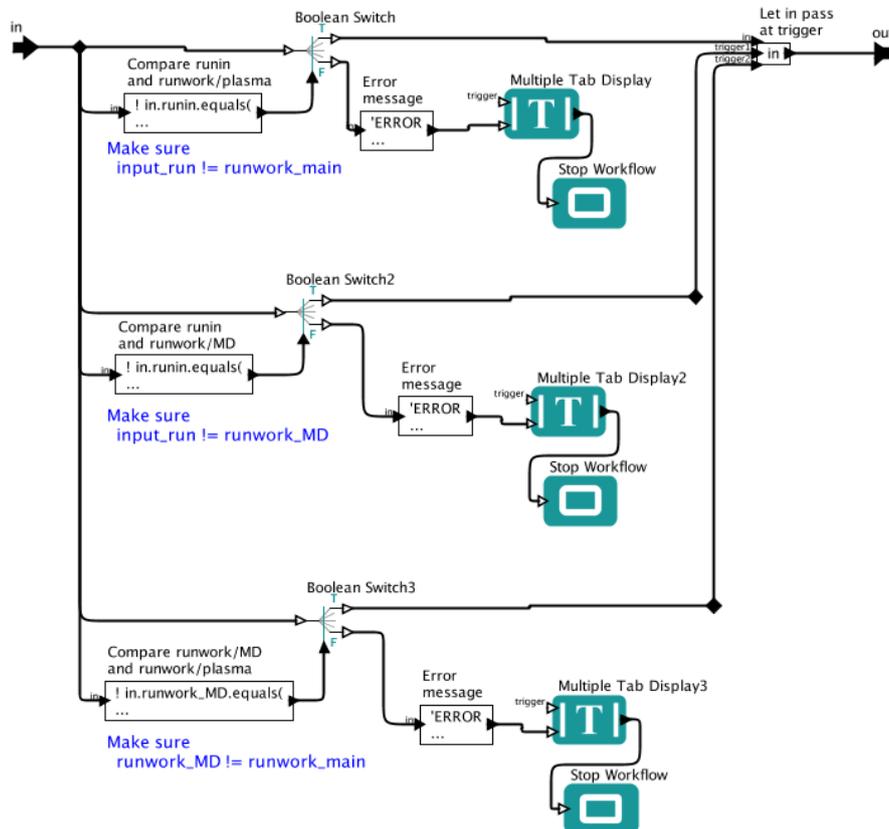


Figure 4. The Validate_Input composite actor, part of UalGetCpoBundle .

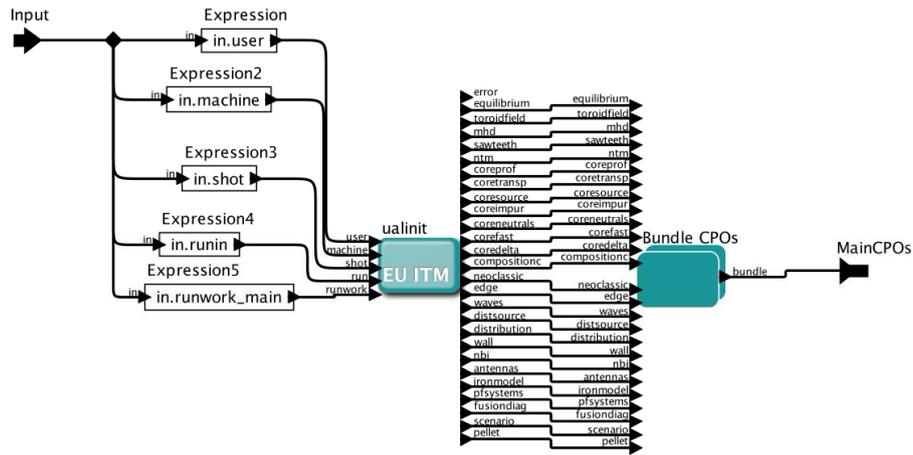


Figure 5. The UAL_read_main_CPOs composite actor, part of UalGetCpoBundle .

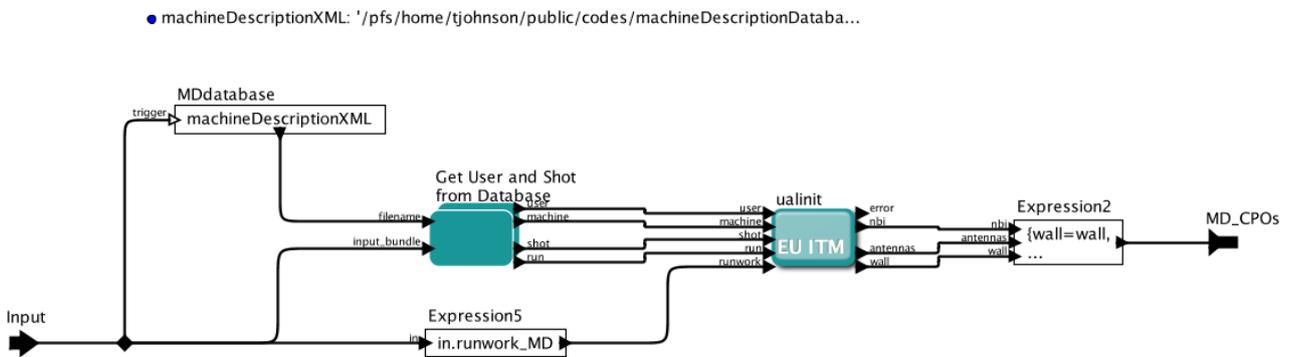


Figure 6. The UAL_read_machine.description_CPOs composite actor, part of UalGetCpoBundle .

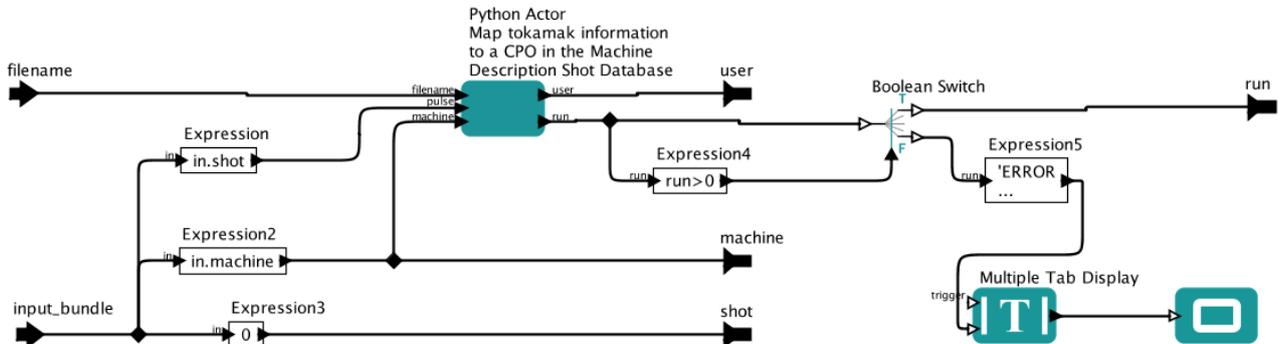


Figure 7. The Get_User_and_Shot_from_Database composite actor, part of UalGetCpoBundle/UAL_read_machine.description_CPOs .