EFDA
EUROPEAN FUSION DEVELOPMENT AGREEMENT
Task Force
INTEGRATED TOKAMAK MODELLING

Training:
The IMP5HCD actor

D. Farina, T. Johnson, G. Vlad and IMP5 contributors

TF Leaders : G. Falchetto, R. Coelho, D. Coster
EFDA CSU Contact Person: D. Kalupin

https://www.efda-itm.eu/ITM/html/imp5_public.html

2011-09-13

Download and installation

- If you do not already have Kepler installed:

```
cd ~
rm -rf kepler .kepler
tar xvf /afs/efda-itm.eu/project/switm/kepler/4.09a/kepler.tar
```

- Make sure you have the line

```
source /afs/efda-itm.eu/project/switm/scripts/ITMv1 kepler test 4.09a
```

  in your ~/.cshrc.

- Download and install imp5hcd

```
cd ~/public
mkdir workflows/
cd workflows
Alt 1:     >  svn co http://gforge.efda-itm.eu/svn/keplerworkflows/trunk/4.09a/imp5/imp5hcd/
Alt 2:     > cp ~tjohnson/public/workflows/4.09a/imp5/imp5hcd/* .
cd imp5hcd
./import_hcd_training_actors
```

  – if you already have some of the actors needed for the IMP5HCD installed, then you will be asked if you wish to reinstall.

Starting IMP5HCD

- Get the input CPOs for shot 5, run 67, machine ”test” from wwwimp5:

```
> cp ~wwwimp5/public/itmdb/itm_trees/test/4.09a/mdsplus/0/euitm_50067.*    ~/public/itmdb/itm_trees/test/4.09a/mdsplus/0/
```

- Start IMP5HCD
  - first time you start kepler, then kepler need to be compiled; start with

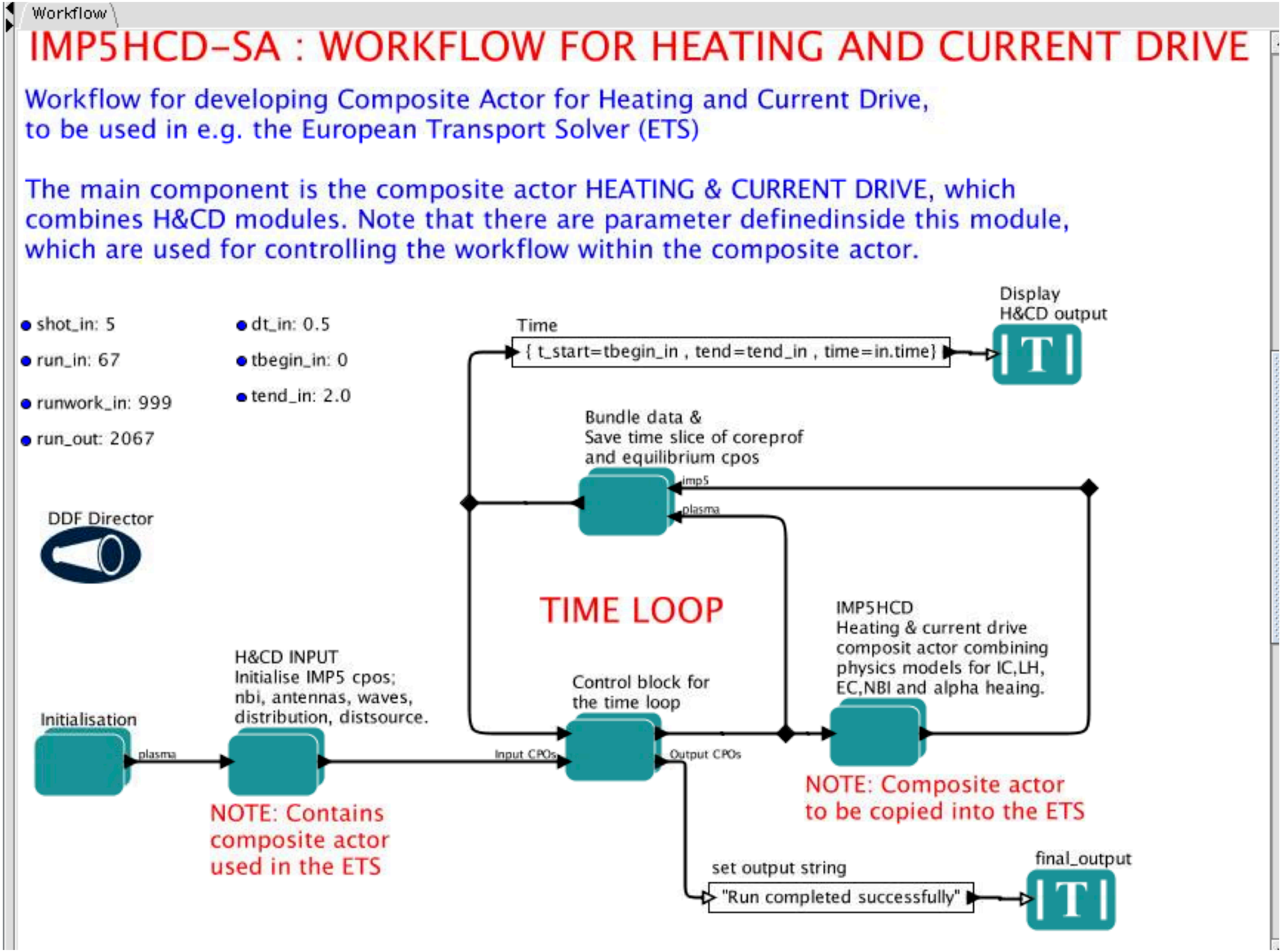
    ```
    > kepler
    ```

    and open the file imp5hcd.xml
  - if you have started kepler before, open imp5hcd directly

    ```
    > kepler.sh  training_sml_imp5hcd.xml
    ```

Since the import script on the last page run compiled Kepler we can run the second option.

NOTE: You must not add the symbols ”&” when calling Kepler, it cannot run in the background!



IMP5HCD-SA : WORKFLOW FOR HEATING AND CURRENT DRIVE

Workflow for developing Composite Actor for Heating and Current Drive,
to be used in e.g. the European Transport Solver (ETS)

The main component is the composite actor HEATING & CURRENT DRIVE, which combines H&CD modules. Note that there are parameter definedinside this module, which are used for controlling the workflow within the composite actor.

● shot_in: 5

● run_in: 67

● runwork_in: 999

● run_out: 2067

● dt_in: 0.5

● tbegin_in: 0

● tend_in: 2.0

DDF Director

Time

{ t_start=tbegin_in , tend=tend_in , time=in.time}

Display
H&CD output

Bundle data &
Save time slice of coreprof
and equilibrium cpos

TIME LOOP

IMP5HCD
Heating & current drive
composit actor combining
physics models for IC,LH,
EC,NBI and alpha heaing.

Control block for
the time loop

H&CD INPUT
Initialise IMP5 cpos;
nbi, antennas, waves,
distribution, distsource.

Initialisation

NOTE: Contains
composite actor
used in the ETS

NOTE: Composite actor
to be copied into the ETS

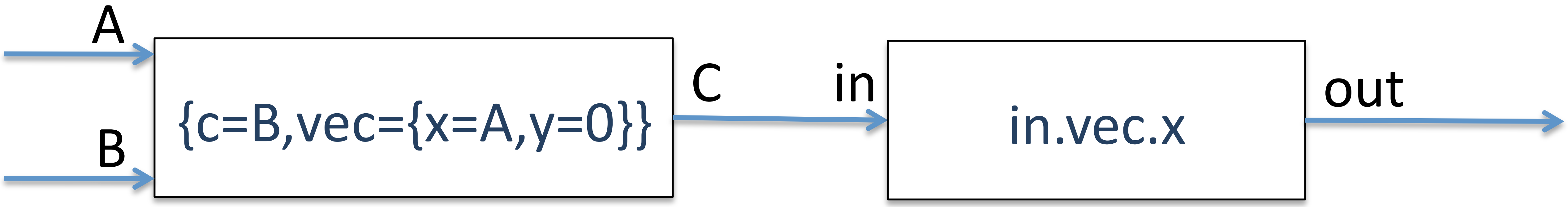
set output string

"Run completed successfully"

final_output

Plasma bundle

- Kepler allow you to bundle data using expressions
  - let A=3.14



then ”out=3.14”

- It is also possible to bundle CPOs; this is done extensively in the IMP5HCD (and in the ETS)

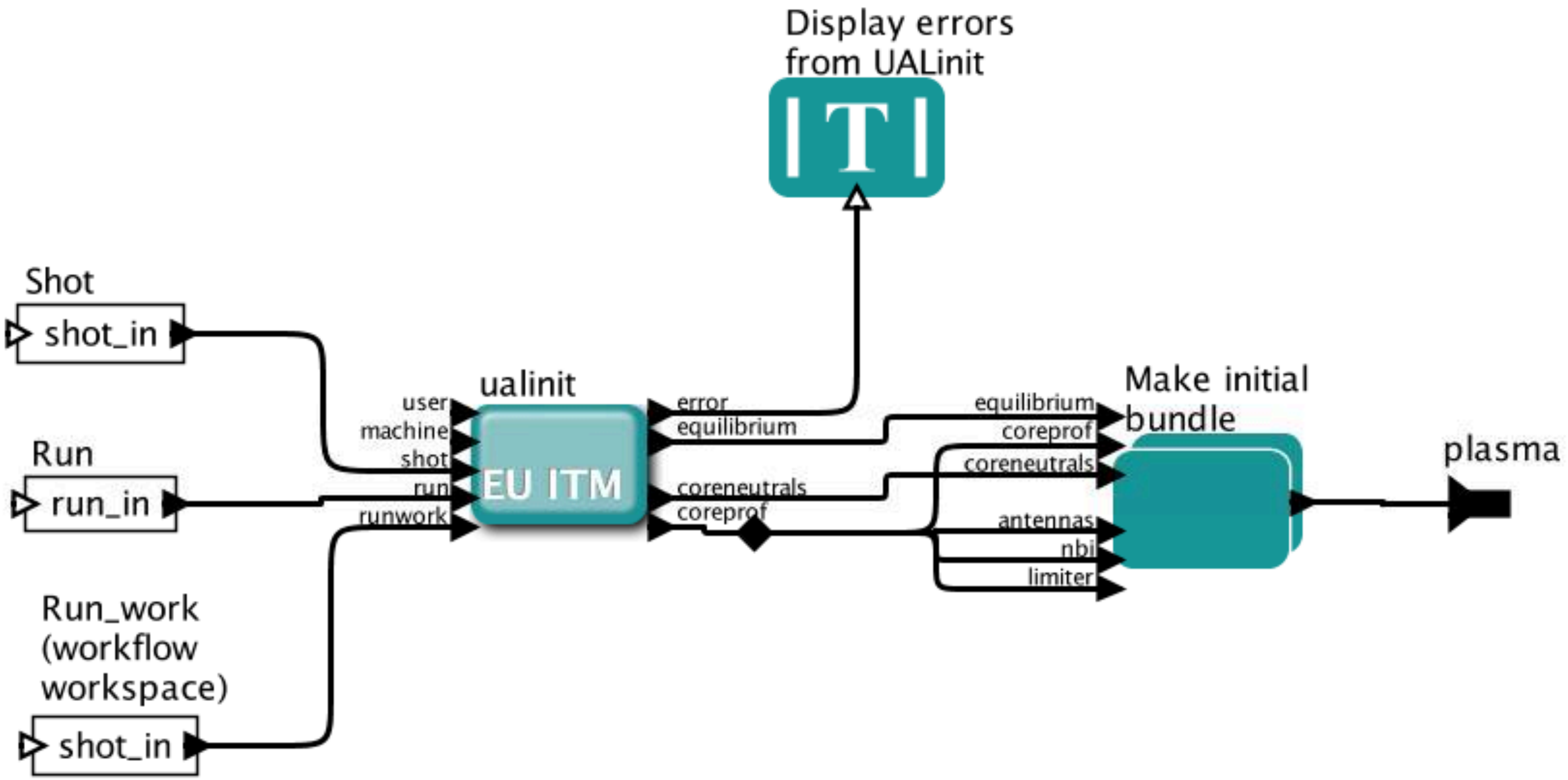
Plasma bundle

- The ITM has deviced a standardised format for bundling CPO and control parameter in the ITM-Plasma bundle

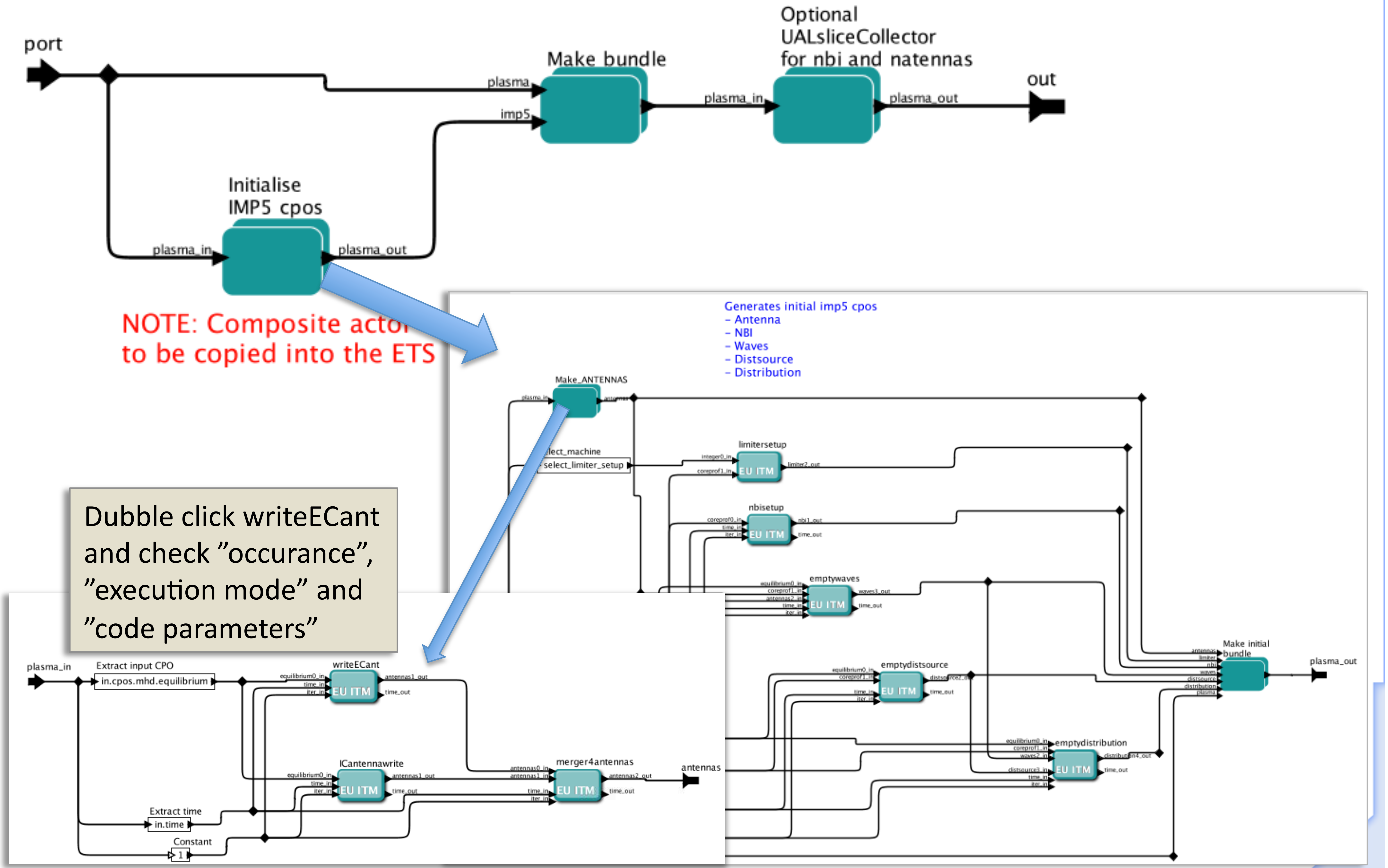
  https://www.efda-itm.eu/ITM/html/itm_conventions.html#itm_conventions_20

- All data transfer between modules in IMP5HCD-SA uses the ITM-Plasma bundle
  - every line in the workflow transports the ITM-Plasma bundle
  - thus we can handle many CPOs in a single line

UAL initialisation



HCD Input



NOTE: Composite actor to be copied into the ETS

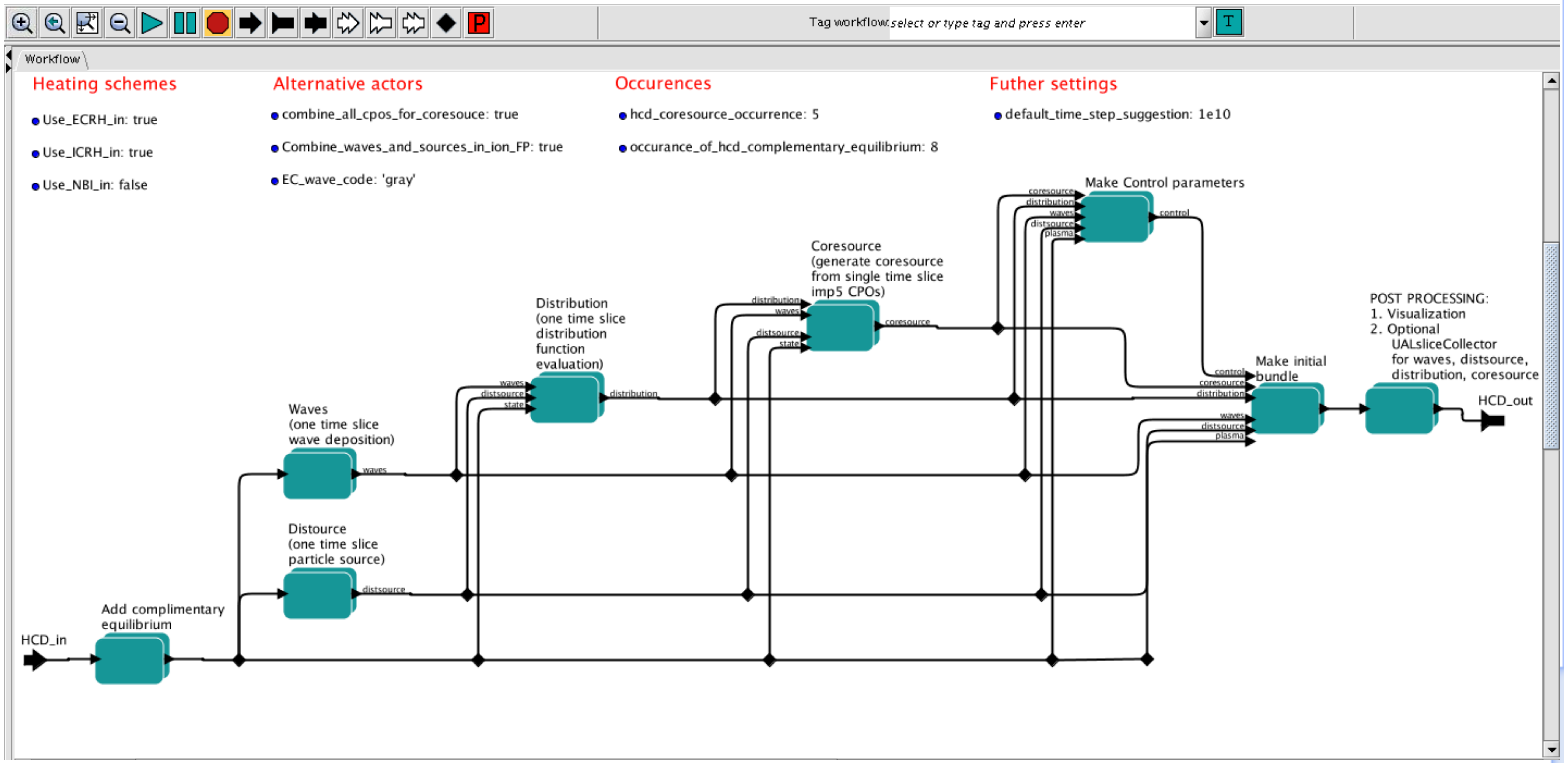
Dubble click writeECant and check ”occurance”, ”execution mode” and ”code parameters”

Occurance and execution mode

- Occurance:
  - For each shot and run number you can store different version of the same CPO. The version number is called the occurance. When writing workflows where the same CPO is generated by more than one code, e.g. here where antennas is generated by both writeECant, ICantennawrite and merger4antennas, each code has to write into a different occurance, or else they will over-write each others results.
- Execution mode:
  - There are different mode in which you can run your actor. The two main ones are
    - jni = Java Native Interface
    - debug = in a totalview debugger

    Note that when you are in debugg mode, then no data is stored in the local memory between calls (as is the case in JNI-mode). The two modes also accesses different codes:
    - in JNI the code is run via a Java wrapper;
    - in debug the code has a fortran-PROGRAM wrapper run in Totalview

Overview of IMP5HCD



Heating schemes

- Use_ECRH_in: true
- Use_ICRH_in: true
- Use_NBI_in: false

Alternative actors

- combine_all_cpos_for_coresouce: true
- Combine_waves_and_sources_in_ion_FP: true
- EC_wave_code: 'gray'

Occurences

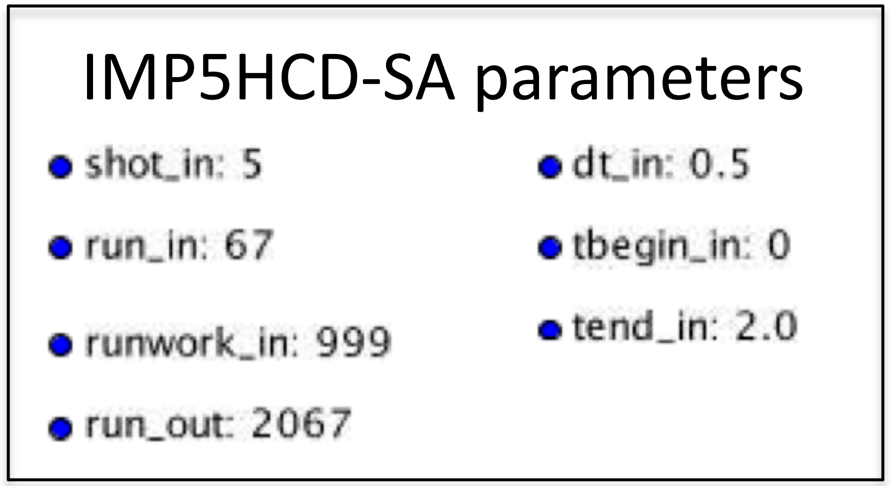
- hcd_coresource_occurrence: 5
- occurance_of_hcd_complementary_equilibrium: 8

Futher settings

- default_time_step_suggestion: 1e10

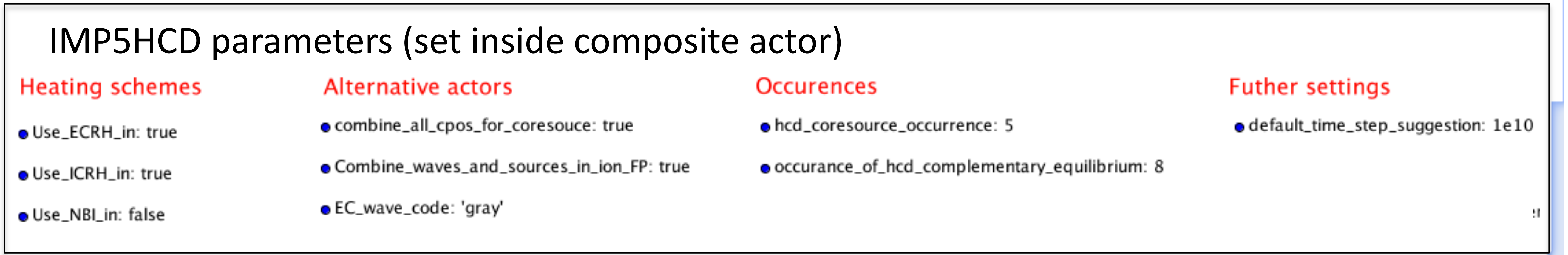
Input parameter

- Three types of input to the workflow:
  - Input CPOs
  - code parameters
  - global workflow parameters
- Global workflow parameters:



IMP5HCD-SA parameters

- shot_in: 5
- run_in: 67
- runwork_in: 999
- run_out: 2067
- dt_in: 0.5
- tbegin_in: 0
- tend_in: 2.0



IMP5HCD parameters (set inside composite actor)

Heating schemes

- Use_ECRH_in: true
- Use_ICRH_in: true
- Use_NBI_in: false

Alternative actors

- combine_all_cpos_for_coresouce: true
- Combine_waves_and_sources_in_ion_FP: true
- EC_wave_code: 'gray'

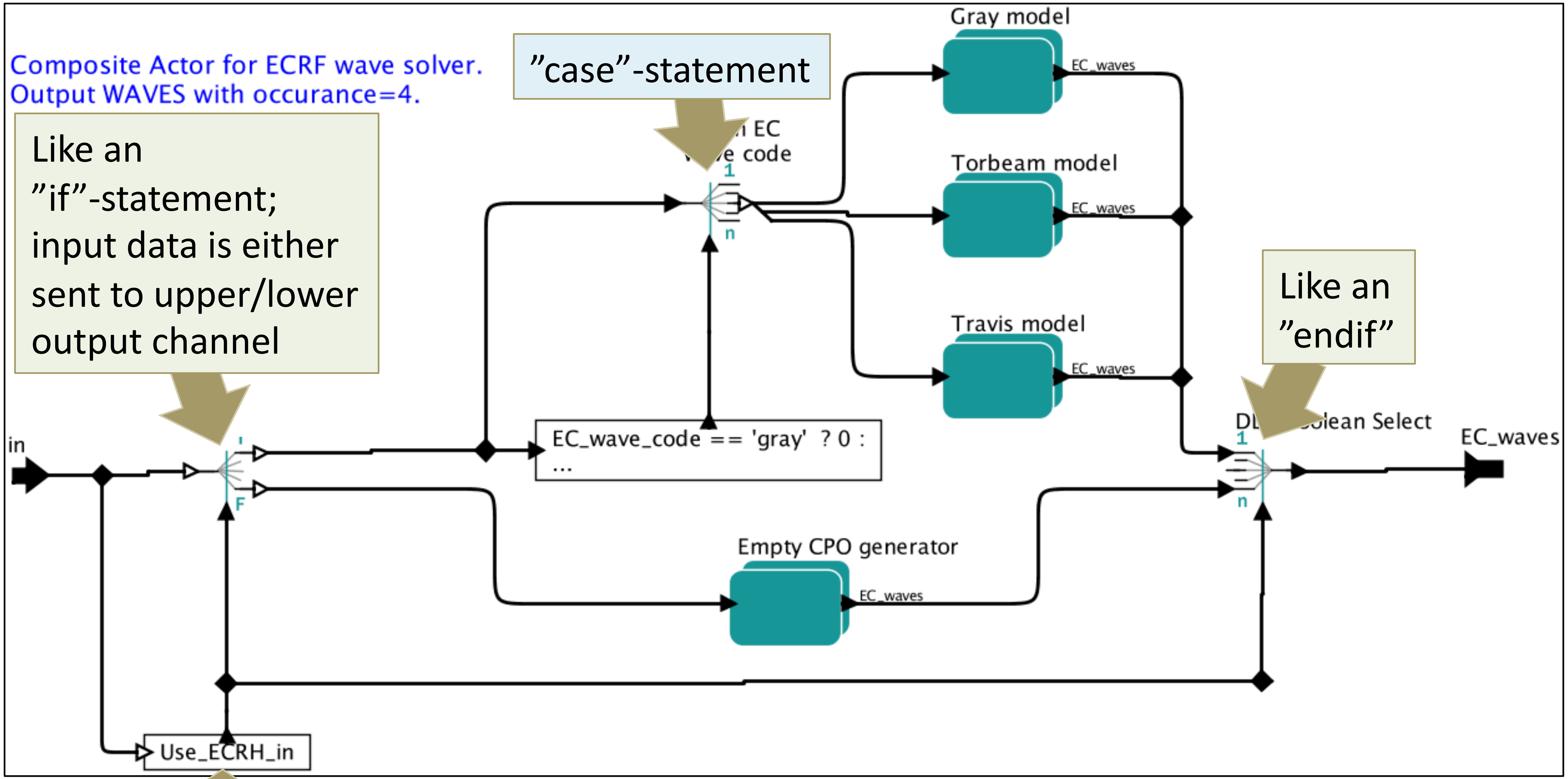
Occurences

- hcd_coresource_occurrence: 5
- occurance_of_hcd_complementary_equilibrium: 8

Futher settings

- default_time_step_suggestion: 1e10

Turning of EC heating



Composite Actor for ECRF wave solver.
Output WAVES with occurance=4.

Like an ”if”-statement; input data is either sent to upper/lower output channel

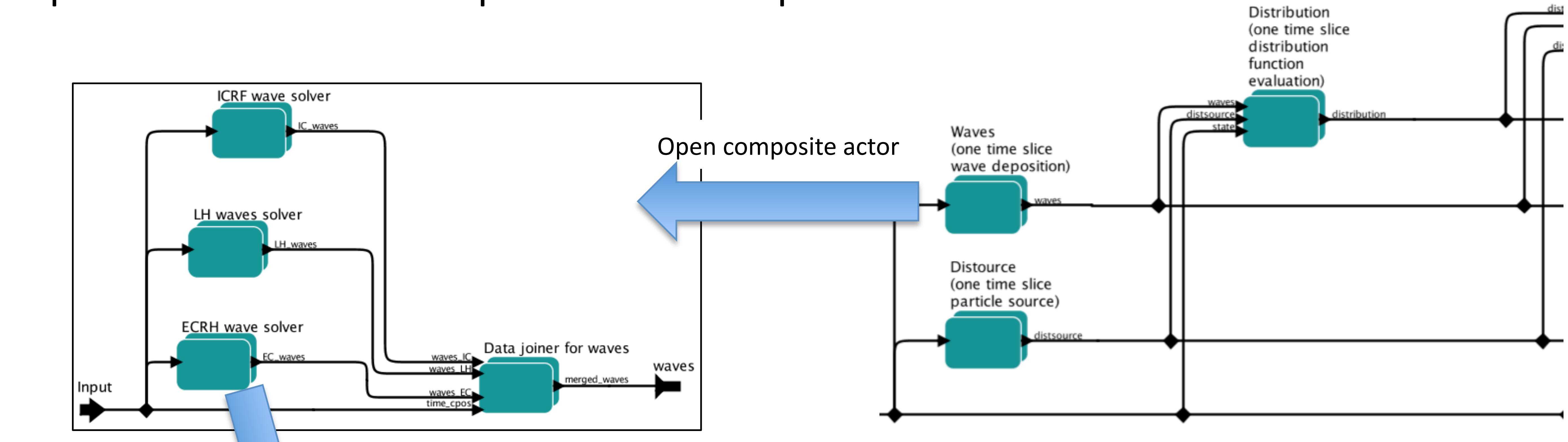
”case”-statement

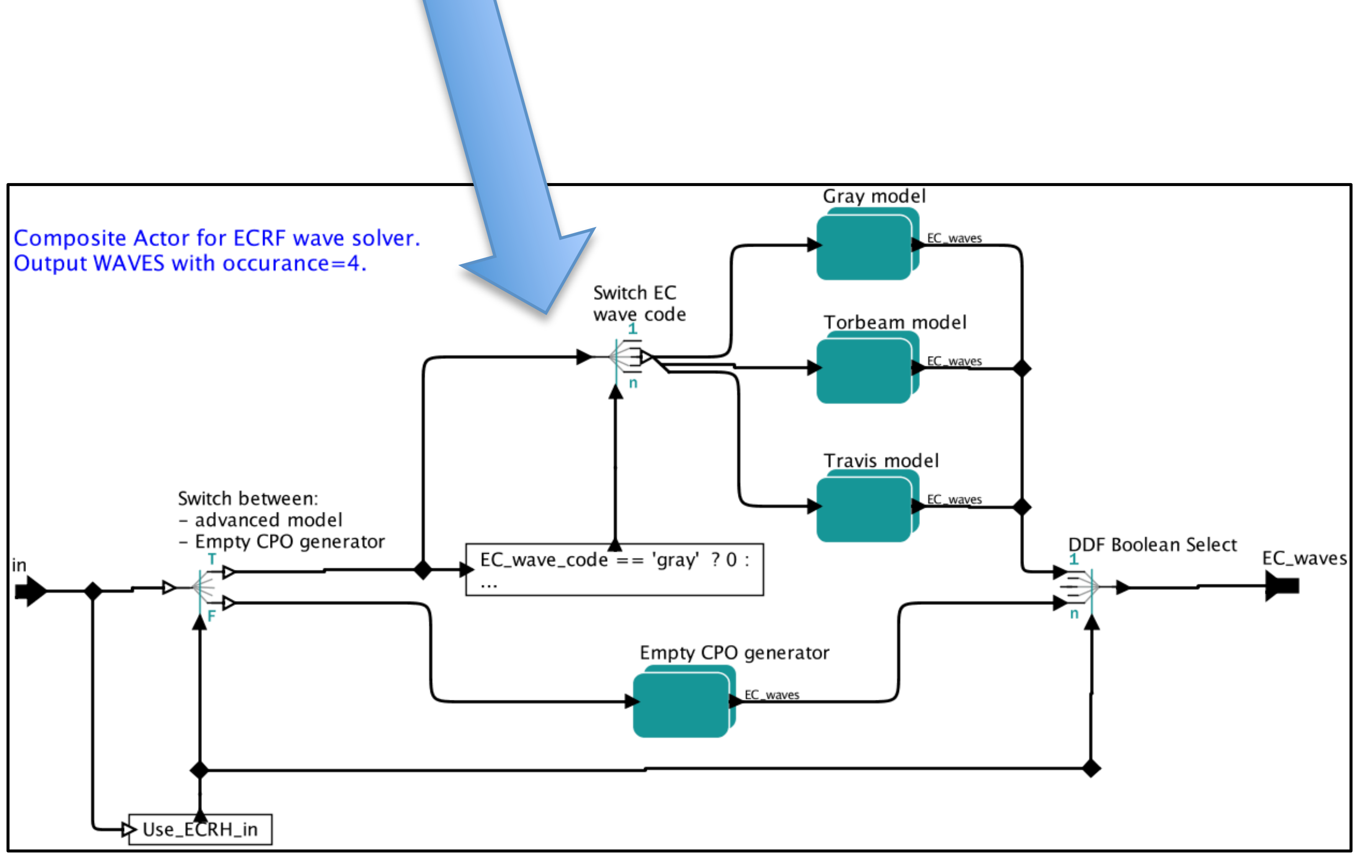
Like an ”endif”

Global parameter (logical) ”Use_ECRH_in” is used as condition for if-statement

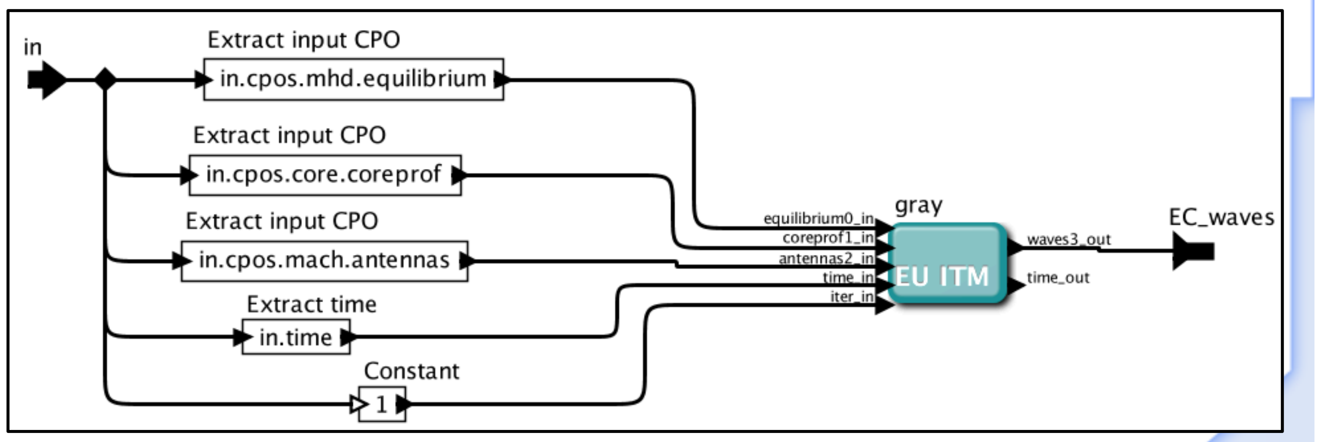
Modifying the IMP5HCD…

Open the ”IMP5HCD”… open ”Waves”… open ”ECRH wave solver”…



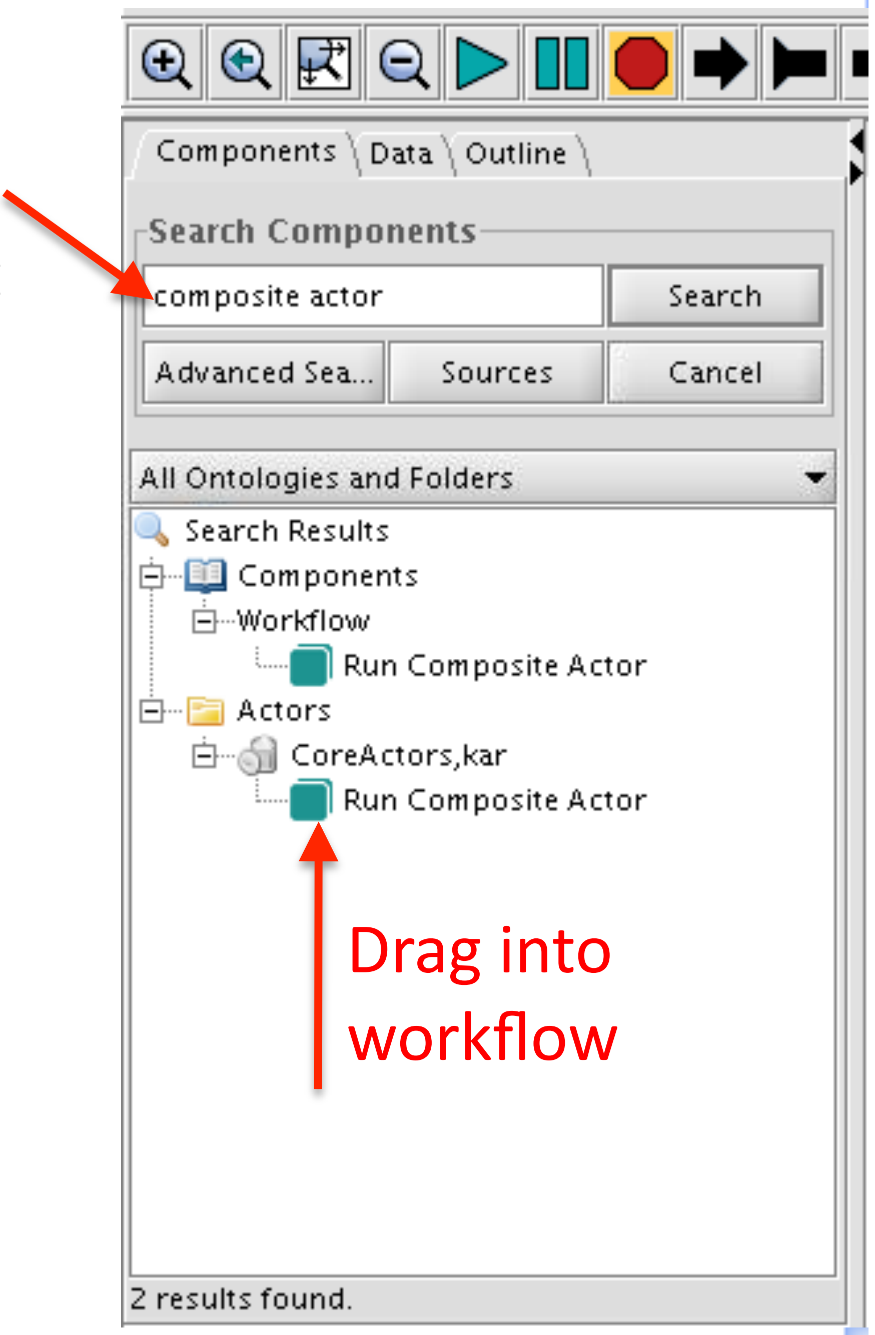


We will now build the following composite actor – a wrapper for GRAY



Create a ”Gray-wrapper” 1(2)

- Start from inside the actor ”ECRH wave solvers” (see previous page)
- Search for: ”Composite actor” and drag into the workflow
  - Left click on the actor and select ”Configure Name”; name it Gray-wrapper
  - Left click and select ”Configure ports”;
    - Introduce two ports (use ”Add port”);
      1. Select: *Input*, Name: *plasma*
      2. Select: *Output*, Name: *waves*
  - Place the the new ”composite actor” above the old Gray-model actor
  - Connect actor in the same way as the old Gray-model actor
- Change selection criteria for ”Select EC wave code”:
  - from ”…EC_wave_code == ’travis’ ? 2 : 0”
  - into ”…EC_wave_code == ’grayw’ ? 3 : 0” , thus new wrapper (port=3) will be access with the EC_wave_code=’grayw’; thus change definition of EC_wave_code in the root-workflow.
- Open the new composite actor…



Drag into workflow

Create a ”Gray-wrapper” 2(2)

- START FROM: inside the ”Gray-wrapper ” composite actor
- Add a Gray actor:
  - Search for GRAY (like search for composite actor; previous graph) and drag actor into workflow
  - Connect GRAY output to the Waves output arrow
- The input to Gray should come from CPOs in the ITM-Plasma bundle, which is entering the Gray-wrapper through the bold arrow named ”Plasma”
- To extract the equilibrium CPO:
  - Search for ”expression” and drag it into workflow, between Gray and the Plasma input arrow
  - Left click and select ”Configure ports”; name the input port ”in”
  - Use the expression to extract the equilibrium CPO from the ITM-Plasma bundle (see https://www.efda-itm.eu/ITM/html/itm_conventions.html#itm_conventions_20)
    - Double click on ”expression” and fill in → in.cpos.mhd.equilibrium
  - Configure name…e.g. ”extract cpo”
  - Connect from the Plasma input arrow to the expression …and on to the Gray/actor port named equilibrium
- Repeat the bullet ”To extract the equilibrium CPO” to extract
  - coreprof (connect to coreprof1_in), antennas (to antennas2_in), time (to time_in) and the integer 1 (to iter_in).

Save KAR file with Gray wrapper

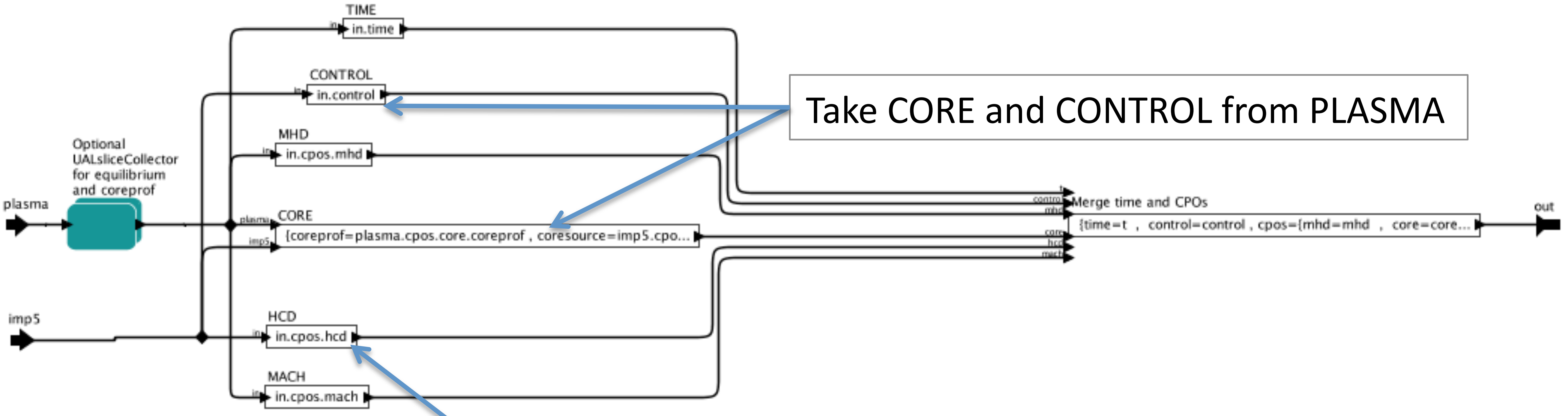
- Close the window with the Gray wrapper
- You should now be inside the ”ECRH wave solver” actor
- To save the Gray-wrapper into a KAR file:
  - left click and select ”save to KAR file”
  - when asked select…to be an IMP5 actor
  - save file as gray_wrapper.KAR
- Next, close all Kepler windows except for imp5hcd.xml

Modify IMP5HCD-SA to run only the Gray code

- You are now in the IMP5HCD-SA window
- Save the file with new name, e.g. gray_training.xml
- Next we wish to replace the IMP5HCD with our new Gray actor
  - remove the IMP5HCD actor
  - Search for gray_wrapper and drag it into the place where the IMP5HCD used to be
  - connect input/output like it was for the IMP5HCD NOTE: since we use the ITM-Plasma bundle the input is the same to the both actors, but the output is different!!

Modify bundling of the Gray-wrapper output

- Open the actor ”Bundle data…”



Take CORE and CONTROL from PLASMA

Note: imp5 is now a WAVES-cpo; take
- WAVES from IMP5 and
- DISTSOURCE and DISTRIBUTION from PLASMA
to generate HCD

Running

- Run normal mode
- Run with Gray in degug mode (double click on gray actor and change from JNI-mode to debug-mode)
- Note:
  - if you double click on a function in Totalview, you entre into this function
  - if you double click on a variable in Totalview, a window appear displaying the value of this variable – this also holds for a CPO!
- Try to step into the gray subroutine and double click on the equilibrium CPO to see its content