User Manual for HLA-PTII federates

For more information about the HLA-PTII co-simulation framework read [1, 2, 3].

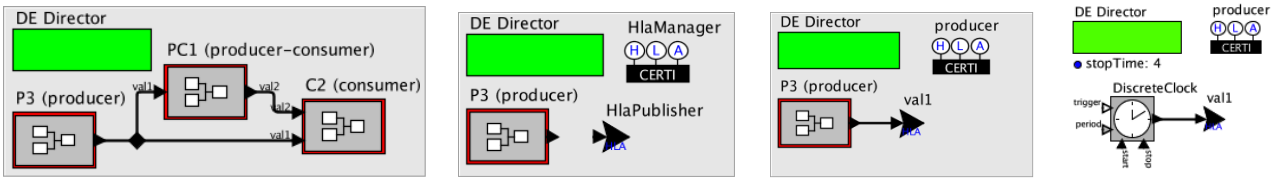
Demos can be found in `$PTII/ptolemy/apps/hlacerti/demo`. The actors can be found in the library under `MoreLibrairies->Co-Simulation->HLA`: HlaManager, HlaPublisher and HlaSubscriber.
The information here are up to revision r71806 (it comes with jcerti.jar and the rtig is launched automatically when running a co-simulation). You need also the HLA compliant RTI called CERTI (tested with versions 3.4.2, 3.4.3 and 3.5) `http://www.nongnu.org/certi/certi_doc/Install/html/index.html`.

Put in your .bash_profile un export for $PTII and $CERTI_HOME. Or at least put aliases:
`alias myPtII=export PTII=your-path-to-ptII''`
`alias cfgCerti=''source your-CERTI_HOME/share/scripts/myCERTI_env.sh.''`
Do not forget to execute these aliases in each terminal you open.

1 Building a model with HLA-PTII co-simulation framework

Let us suppose you have already a (centralized) Ptolemy model as the one of figure 1.a. You want to simulate this model in a distributed way using 3 simulators, one for each composite actor.

Important remark: The top-level model must use a DE director, but you can have a Continuous director in a composite actor inside.



a) Centralized model b) Hla actors c) Naming the actors d) flat federate

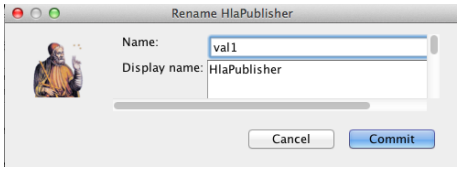
Figure 1: Building Ptolemy federates

Consider that you want to create the `producer` federate. The steps are the following:

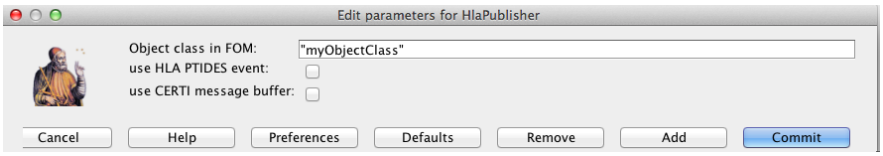
1. Create a FOM file with the classes and attributes used in the simulation as in figure 2.c
2. Duplicate the model of figure 1.a, removing all the blocs except the composite actor "P3 (producer)"
3. Drag the actors HlaManager and HlaPublisher from `MoreLibrairies->Co-Simulation->HLA` (see figure 1.b)
4. Configuring HlaPublisher. Connect the output of the composite actor to the HlaPublisher actor, then: a) right-click on HlaPublisher, Configure/Rename and put `val1` (the name of the class attribute in the fed file as in figure 2.a) in the field `name` ; the name displayed will be the same (figure 1.c); b) double-click HlaPublisher (the window of figure 2.b pops out) and put `MyObjectClass` (the name of the class according to your fed file in figure 2.c) in the parameter `Object class in FOM`
5. Configuring HlaManager. Double-click this actor; the window in figure 2.d appears. Put the name of the federation and the federate as in the fed file (figure 2.c) and browse the fed file. You can change the lookahead value. If you need a synchronization point, tick the field "Require synchronizationPoint?" and choose a same name for all federates of your distributed simulation. Remark: only the last federate to be launched must have the field "Is synchronization point creator?" ticked.

Now your Producer federate is ready and you can create the federates `consumer` and `prodcons`. The steps are similar.

- Do step 2 (duplicate the centralized model removing all composite actors excete the one you want to simulate)



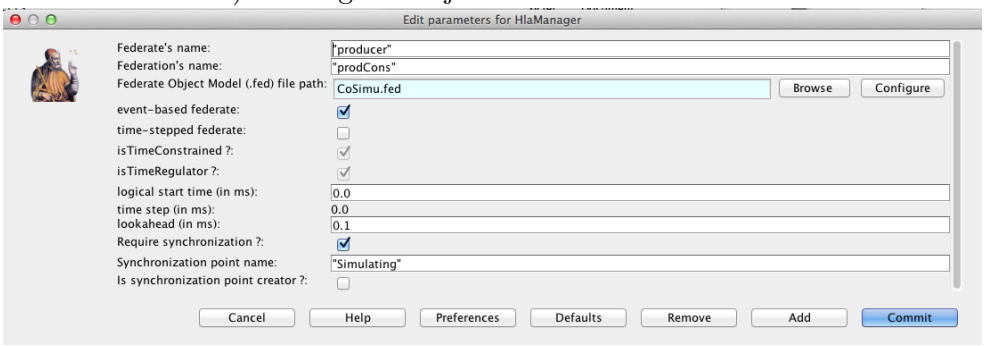
a) Renaming with attribute



b) Naming the object class in HlaPublisher

```
;; CoSimulation
(Fed
  (Federation prodCons)
  (Fedversion v1.3)
  (Federate "prodcons" "Public")
  (Federate "consumer" "Public")
  (Federate "producer" "Public")
  (Spaces)
  (Objects
    (Class ObjectRoot
      (Attribute privilegeToDelete reliable timestamp)
      (Class RTIprivate)
       (Class myObjectClass
              (Attribute val RELIABLE TIMESTAMP)
              (Attribute val1 RELIABLE TIMESTAMP)
              (Attribute val2 RELIABLE TIMESTAMP))))
  (Interactions
    (Class InteractionRoot BEST_EFFORT RECEIVE
      (Class RTIprivate BEST_EFFORT RECEIVE))))
```

c) CoSimu.fed file (FOM)



d) Configuring HlaManager

Figure 2: FOM

- For `prodcons` federate, add a HlaPublisher for `val2` and a HlaSubscriber for `val1`. Follow setp 4 for HlaPublisher actor. For the HlaSubscriber the procedure is the same. Add also a HlaManager (step 5). Pay attention to name the federate as `prodcons` in the configuration window of HlaManager (figure 2.d)
- For `consumer` federate, add two HlaSubscriber and a HlaManager. Proceed as in steps 4 and 5 respectively
- Check if all your federates (but one) have the field "Is synchronization point creator?" unticked. Only one of them must have this field ticked and it must be the last one to be launched.

Remark: if your composite actor has a DE director, you can use directly the composite actor as in figure 1.c or you can have a *flat* model as in figure 1.d. If the composite actor has a Continuous model then you need to use the composite actor.

2 Running the federation

1. Open a terminal and run `cfgCerti` as said in the beginning of this manual or execute the `$CERTI_HOME/share/scripts/myCERTI_env.sh` shell
2. Go to the folder where the 3 federate models are (or give the absolute address) and open the models: `$PTII/bin/vergil consumer.xml producer.xml prod-cons.xml &`
3. Check there is no `rtig` process running (the first model to be run will automatically launch this process). If there is a `rtig` running, kill the process
4. Check there is only one model that has the field "Is synchronization point creator?" ticked. Run the other models in any order but the last to be run is the one that has the cited field ticked.

References

[1] Lasnier, G., Cardoso, J., Siron, P., Pagetti, C. and Derler, P.. *Distributed Simulation of Heterogeneous and Real-time Systems*, 17th IEEE/ACM Inter. Symposium on Distributed Simulation and Real Time Applications - DSRT 2013, 30 Oct. 2013 - 01 Nov. 2013 (Delft, Netherlands). *Best paper award.*

[2] Lasnier, G., Cardoso, J., Siron, P. and Pagetti, C. *Environnement de cooperation de simulation pour la conception de systemes cyber-physiques*, Journal europeen des systemes automatises. Vol. 47 n. 1-2-3, 2013.

[3] TBD