

MECSYCO-DACCOSIM coupling

Benjamin Camus, Mathieu Caujolle, Virginie Galtier

July 23-24-27, 2015

Hybrid Simulation Context

Smart Grid simulation involves both:

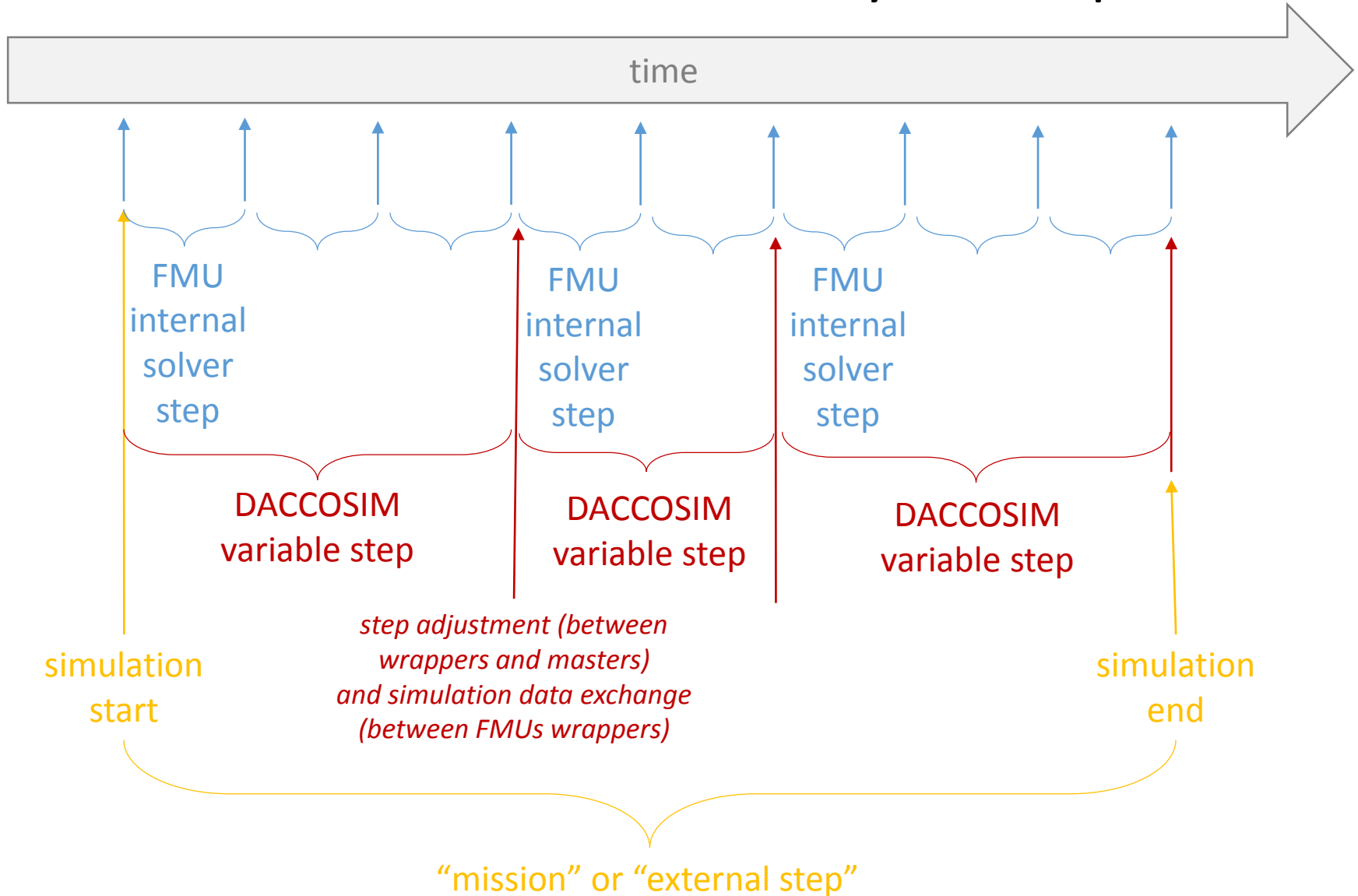
- physics models simulated with equations-based, time stepped simulators and
- telecom and IS models both simulated with event-based simulators

→ “cyber-physical system”

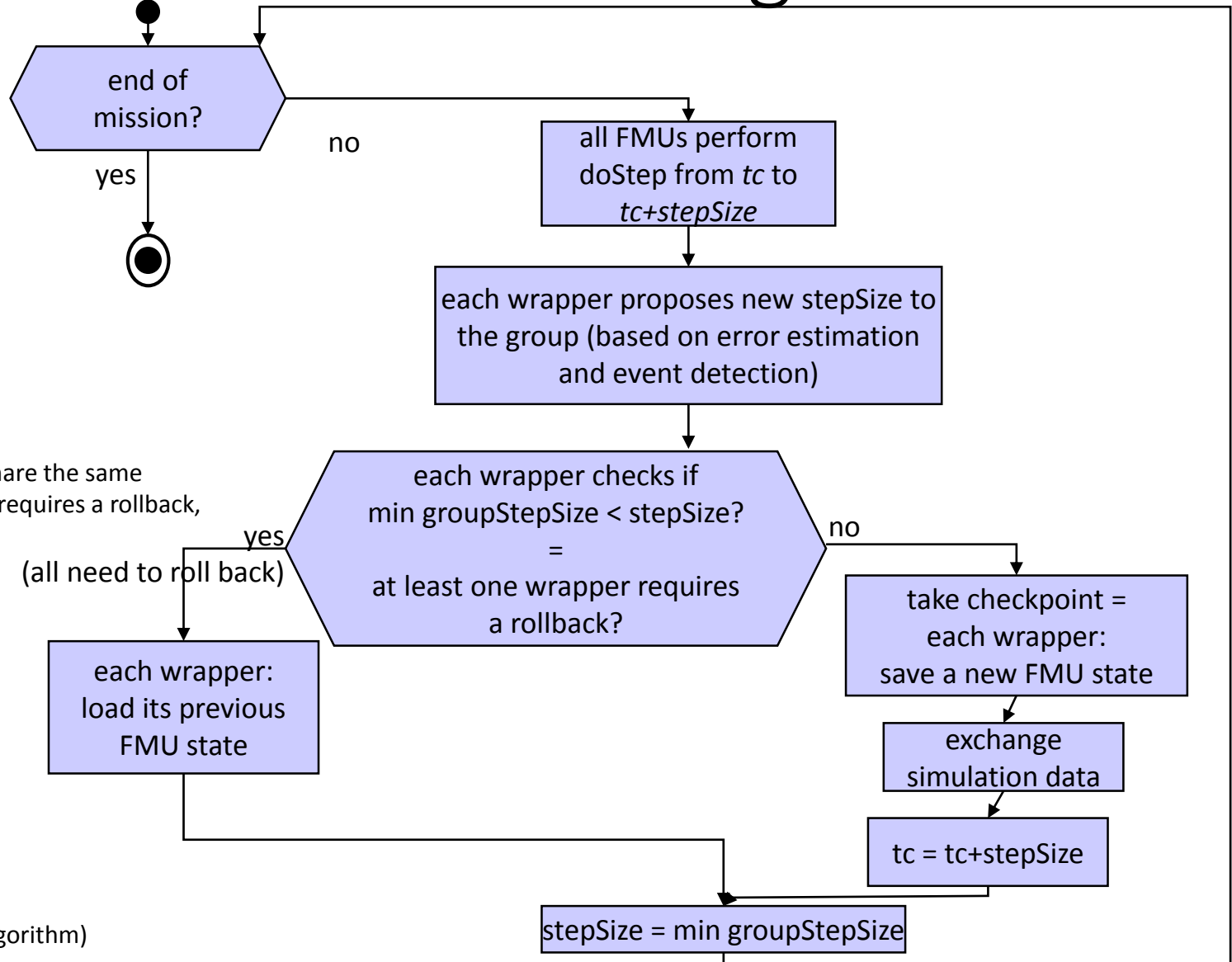
DACCOSIM

- designed to coordinate FMI-compliant time-stepped simulators and to achieve efficient and accurate simulation ; in order to do that, DACCOSIM relies on the rollback capacity of components and manages variable simulation step size
- DACCOSIM is not designed to integrate an event-based telecom simulator such as NS3 for instance (not FMI-compliant and no rollback supported)
- Nevertheless, DACCOSIM is following the progress being done by the FMI standard on the hybrid-cosimulation (version 2.1)

DACCOSIM Vocabulary: “step” ...



Simplified DACCOSIM Algorithm



In DACCOSIM, all FMUs share the same stepSize. When one FMU requires a rollback, all FMUs rollback too.

(“prudent” version of the algorithm)

State-Event Detection and Management in DACCOSIM

- Detection:

At the end of each DACCOSIM step, DACCOSIM is able to detect some state events by checking if Boolean or Integer values have been changed between the beginning of the step and the end of the step.

- Action:

When a state event is detected, DACCOSIM rolls back to the beginning of the step, re-does the step with the minimal step size allowed by the user, and keeps this step size value until the event is detected again in order to locate the event as precisely as possible.

- Notes:

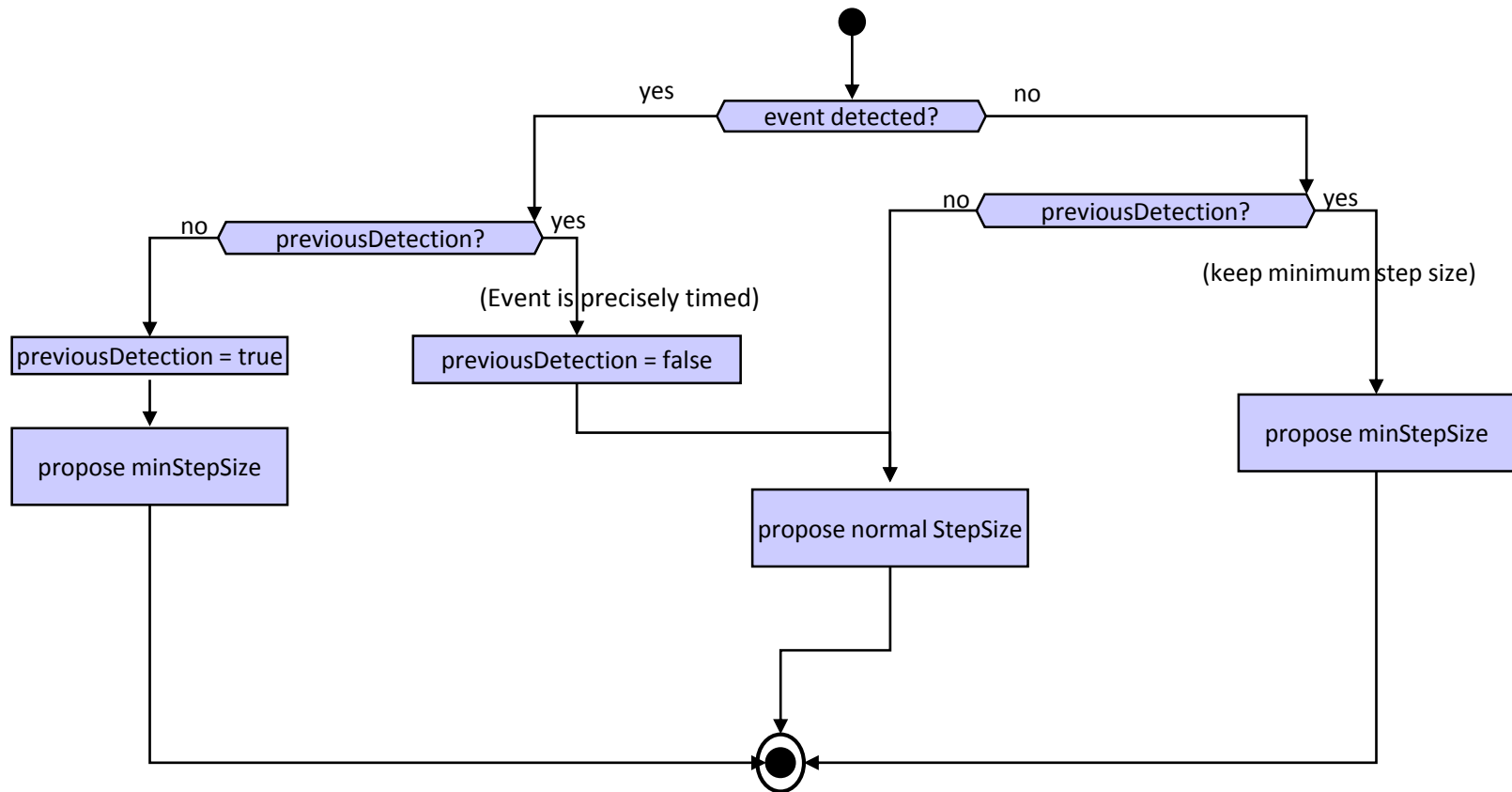
- some events might go undetected if the value changes twice during the step for instance (step too large)

→ this risk can be avoided by the DACCOSIM user when providing minimum and maximum stepSize values carefully chosen based on the time constant of each model

Maybe FMI could evolve to allow the model designer to include this information in the FMU?

- FMI version 2.1 is supposed to handle state-event detection...

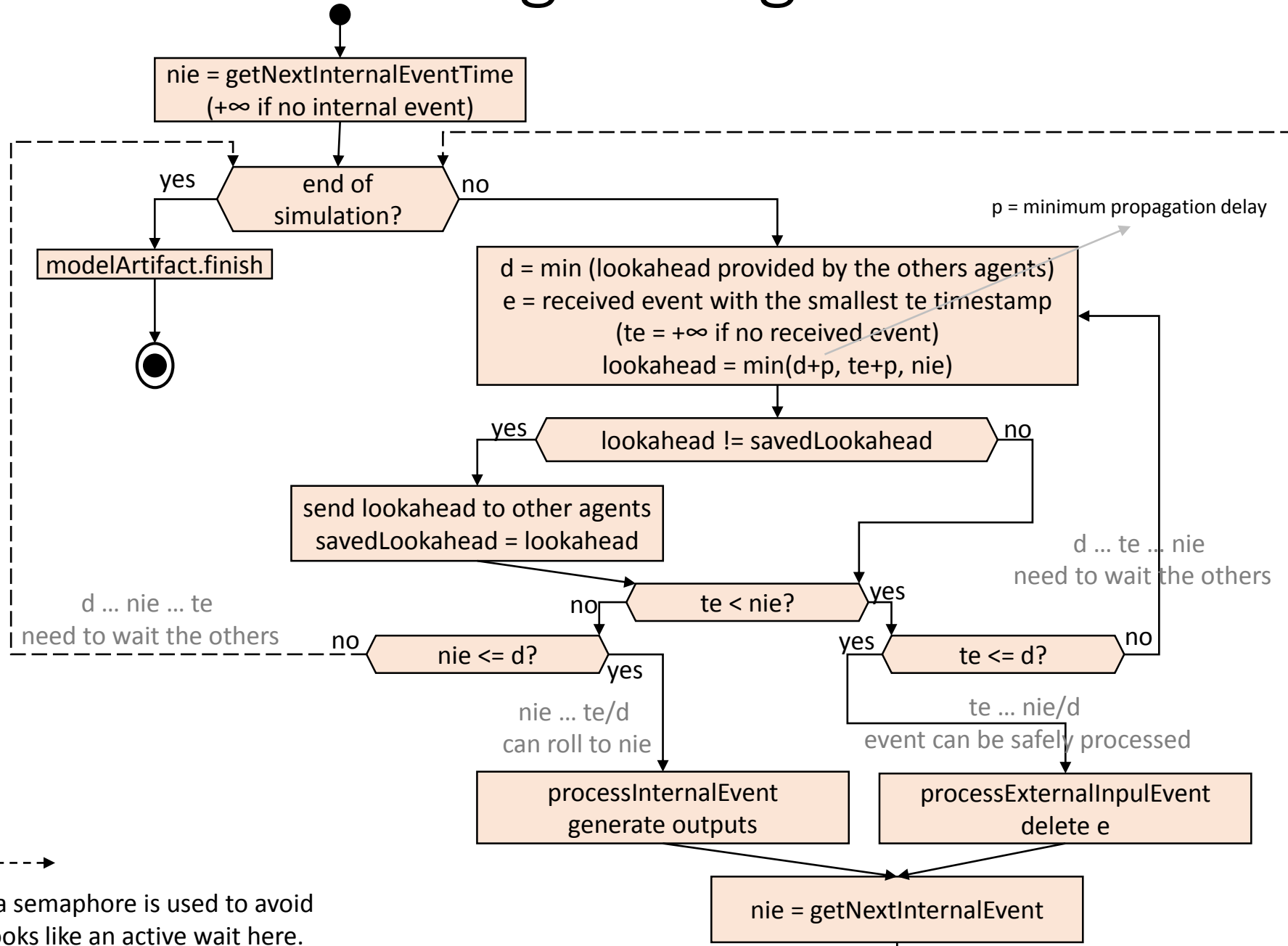
Details for the “new stepSize proposal” to local master



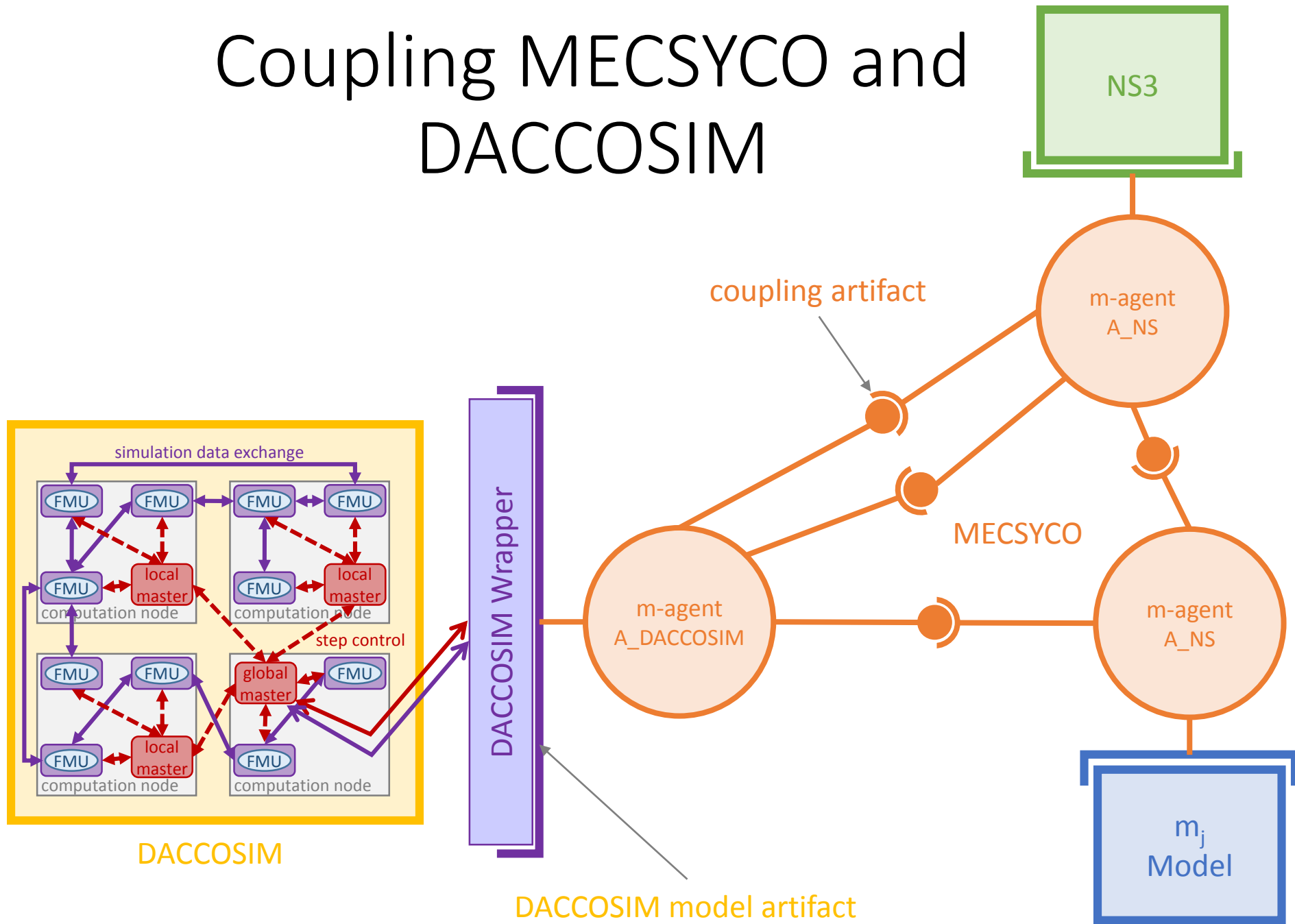
MECSYCO

- designed to coordinate the simulation of DEVS-compliant models (integrating all possible formalisms), guarantees respect of causality constraints for simulators (including event-based), offers basic connection to telecom simulators (such as NS) and to FMUs in compliance with the FMI standard
- MECSYCO is not designed to command components to rollback in order to gain better accuracy

MECSYCO m-agent Algorithm



Coupling MECSYCO and DACCOSIM



Coupling Process to Generate State-Events from DACCOSIM

- output state-events are available in DACCOSIM at the end of a DACCOSIM step
- input state-events are propagated in MECOSYCO after proceeding to the next interval event
 - the end of a DACCOSIM step must be considered as an internal event by DACCOSIM wrapper and model artifact
- but not all DACCOSIM steps result in generation of state-event. Stopping DACCOSIM at the end of each step to coordinate with MECOSYCO might alter the performances
 - the next internal event is the minimum between the date of the end of the DACCOSIM mission and the end of the next DACCOSIM step during which a state-events occurs

Coupling Process to Integrate State-Events into DACCOSIM

- new external values can be injected in DACCOSIM only at the beginning of the mission
- a mission must start at the date of the earliest external event

Coupling Principle

- When asked by its model artifact its next internal event, DACCOSIM engages in an exploration to find its earliest outgoing state-event.
- The validation of the computations performed during exploration only takes place after making sure no external events occur in that period.
- Otherwise, DACCOSIM mission end time is restricted to the date of the earliest external event.

Scenario Example (to be continued)

MECSYCO

- What is your next internal event?
- OK, well I've talk to my colleagues and you're clear: there will be no external event between now and 10. Please, proceed to date 10.

DACCOSIM

- Let me see... I've been configured to stop my simulation at 20. But maybe I will generate a state-event during this mission... Let's call S0 my current state. And let's ride my customized DeLorean... OK, I've been into the future and here is what I've found out: I will generate state-event at 10! And at 10, I'll be in state S10. So if you can assure me my inputs will stay the same, I'd say my next internal event is at 10, this way I'll be able to post my event timely!
- Fine. I'm already in state S10. Let me just get rid of my S0 backup (and maybe log something).

Scenario Example (end)

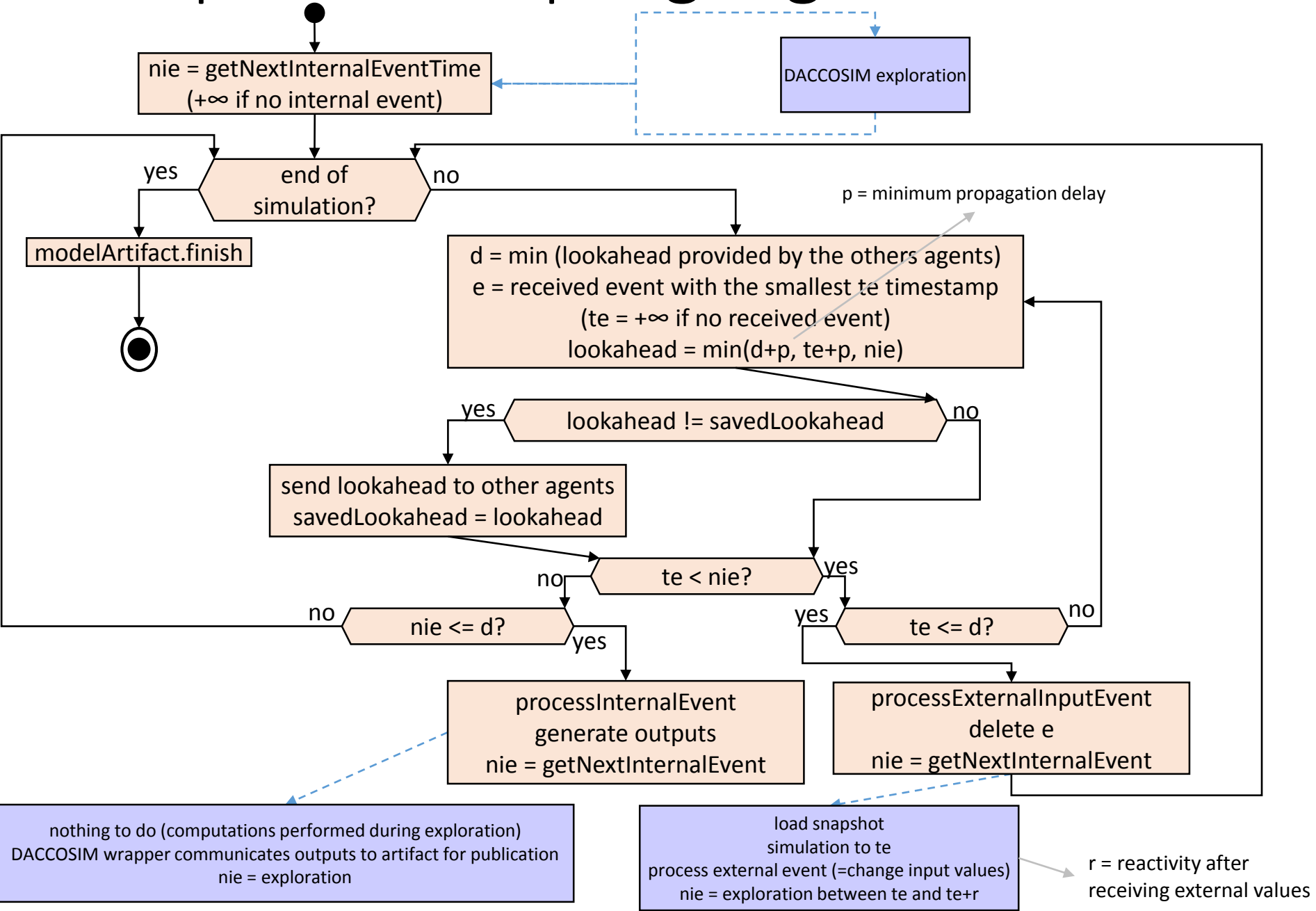
MECSYCO

- What is your next internal event?
- OK. I've talk to my colleagues and another guy sent me this external event for time 15. Please, proceed to date 15 and process this event.
- What is your next internal event?
- OK, my colleagues said there will be no external event between now and 20. Please, proceed to date 20.
- Great! Another successful mission! Good job!

DACCOSIM

- Let me ride my DeLorean from 10 toward 20... Well, nothing to report... I'd say my next internal event is at 20.
- Roger that. I'm loading my backup S10 and rolling from there to 15. OK, done. I can change my model according the external event now. OK, done.
- Let me take a snapshot S15 and ride my DeLorean from 15 toward 20... Well, nothing to report... I'd say my next internal event is at 20.
- OK, done already! Let me just get rid of my S15 backup.

Simplified Coupling Algorithm



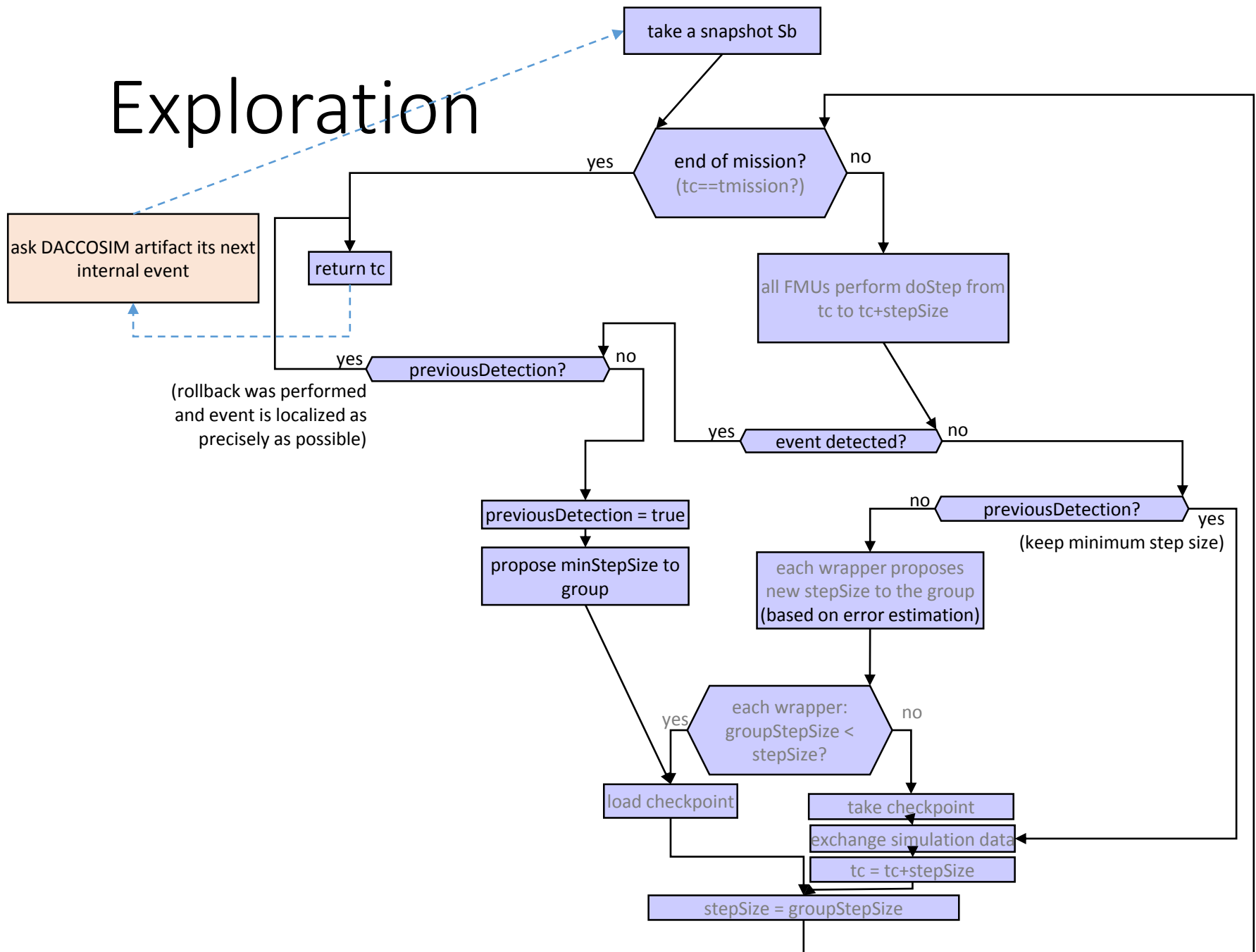
Simultaneous Events

- Currently, if multiple (m) external events are available with the same timestamp, `processExternalInputEvent` is to be called m times.
- This is not efficient with the current coupling algorithm: first event e_1 (@ t_e) is processed, next an exploration is made from t_e but second event e_2 (also @ t_e) forces to cancel the exploration and start again at t_e etc.
 - ✓ It is required to improve MECSYCO so that `processExternalInputEvent` provides a vector EIE of simultaneous events instead of just one event
- The order into which the simultaneous events are provided by the agent is “random”.
 - ✓ To make it determinist, a kind of MECSYCO aggregator can be used.

Exploration

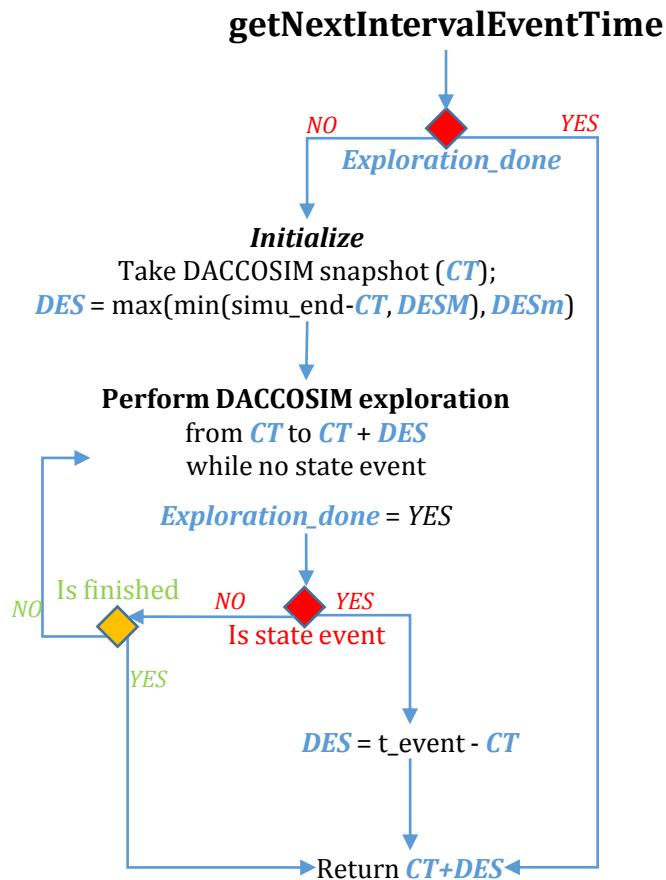
- The exploration phase is based on the standard DACCOSIM algorithm (e.g. the simplified algorithm described on slides 5&7) where the return value for `nie` is:
 - either the end of mission (no internal event detected)
 - or the multiple of the `minStepSize` which brings the simulation as close as possible to the detected event
- Note the “new `stepSize` proposal” algorithm given on slide 7 will be improved depending on the next FMI 2.1 version

Exploration



DACCOSIM Model Artifact getNextIntervalEventTime algo

“Basic” version



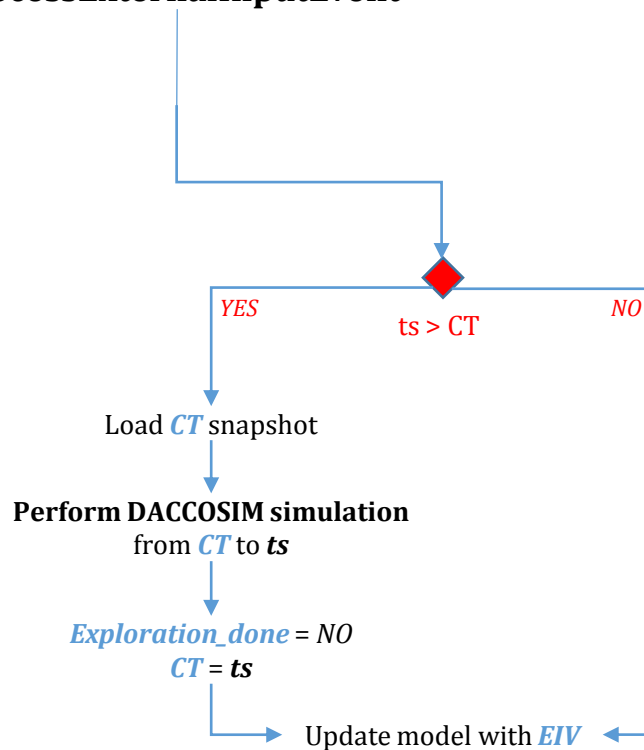
Model Artifact internal variables:

- **CT**: Current Time of the simulation
- **simu_end**: Simulation end time
- **t_event**: estimated next internal event time
- **DES**: DACCOSIM External Step (mission duration)
- **DESM (DESm)**: Maximum (minimum) DACCOSIM External Step (max / min mission duration)
- **EIV**: External Input Value stack
- **t_event**: estimated next internal event time
- **Exploration_done**: Has the exploration been performed?

DACCOSIM Model Artifact processExternalInputEvent algo

“Basic” version

processExternalInputEvent



Model Artifact internal variables:

- **CT**: Current Time of the simulation
- **ts**: timestamp of the ExternalInputEvent
- **EIV**: External Input Value stack
- **Exploration_done**: Has the exploration been performed?

DACCOSIM Model Artifact processInternalEvent algo

“Basic” version

processInternalEvent



$$CT = CT + DES$$

Exploration_done = NO

Model Artifact internal variables:

- **CT**: Current Time of the simulation
 - **DES**: DACCOSIM External Step (mission duration)
- Exploration_done**: Has the exploration been performed?

Minimum Propagation Delay

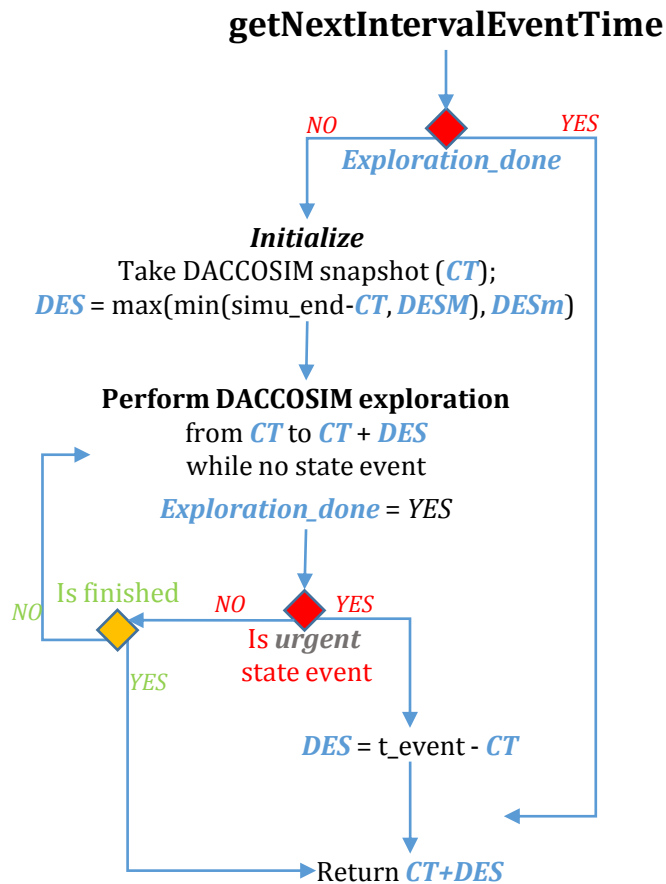
- my Minimum Propagation Delay (MPD) is p
- when I receive an external event at t , I guarantee it will not trigger the generation of an event from me before $t+p$
- Each model has its own $MPD > 0$, fixed at the beginning of the simulation
- For DACCOSIM coupling, $MPD =$ DACCOSIM minimum step size
- The smaller the minimum MPD across the models, the worse the performances because all agents need to communicate with a frequency function of this minimum.
 - ✓ But MECSYCO could easily be modified to keep frequent communications only between implicated agents and not simulation-wide

Urgent vs Non Urgent Events

- Some external events might not require to be taken into account right away
- Such non-urgent events will not trigger the generation of events upon reception (e.g. events received by a log or display component)
- Note: all out-going events are urgent, it's up to the receiver to consider that events received on such port for instance are non-urgent
- “Basic approach” from previous slide needs to be augmented to handle the distinction between urgent and non-urgent external events. See next slides.

DACCOSIM Model Artifact getNextIntervalEventTime algo

“Urgency integrated” version



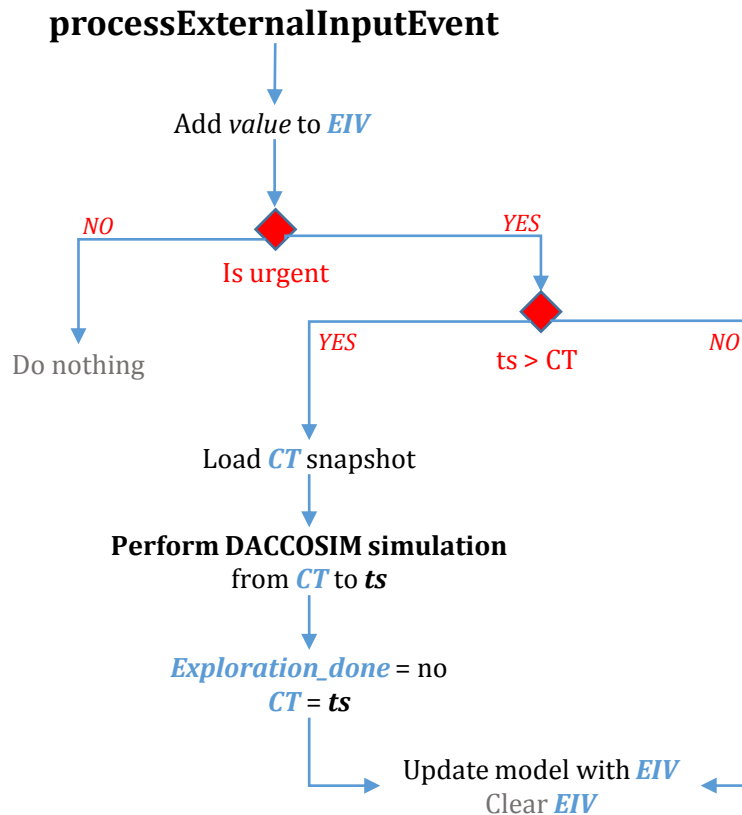
Model Artifact internal variables:

- **CT**: Current Time of the simulation
- **simu_end**: Simulation end time
- **t_event**: estimated next internal event time
- **DES**: DACCOSIM External Step (mission duration)
- **DESM (DESm)**: Maximum (minimum) DACCOSIM External Step (max / min mission duration)
- **EIV**: External Input Value stack
- **t_event**: estimated next internal event time
- **Exploration_done**: Has the exploration been performed?

If the event is non-urgent, the mission is not ended to process it. It is done afterwards

DACCOSIM Model Artifact processExternalInputEvent algo

“Urgency integrated” version



Model Artifact internal variables:

- **CT**: Current Time of the simulation
- **ts**: timestamp of the ExternalInputEvent
- **EIV**: External Input Value stack
- **Exploration_done**: Has the exploration been performed?

Non-urgent events are stored in a DACCOSIM Artifact queue but are not processed in processExternalInputEvent

DACCOSIM Model Artifact processInternalEvent algo

“Urgency integrated” version

processInternalEvent



$CT = CT + DES$

Exploration_done = no



Update model with *EIV*
Clear *EIV*

Model Artifact internal variables:

- **CT**: Current Time of the Simulation
- **DES**: DACCOSIM External Step (mission duration)
- **EIE**: External Input Events stack
- **Exploration_done**: Has the exploration been performed?

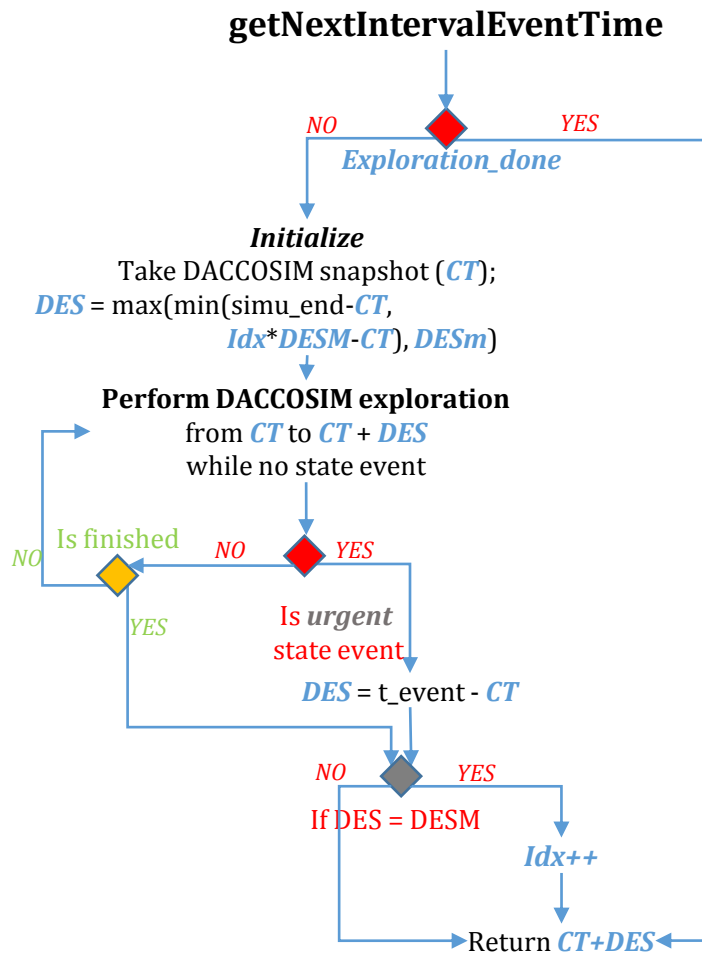
*Non-urgent events are retrieved from
the artifact queue when rolling to the
next internal event*

Events vs Periodic Outputs

- To “program” DACCOSIM to periodically exchange data with other models (e.g. every 100 s), DACCOSIM initial mission length must be set accordingly.
- Example:
 - Models synchronization every 100
 - State events in DACCOSIM at 75, 250 and 430
 - First exploration: mission = 100, prematurely ended because of state event at 75
 - Next mission: DACCOSIM “endSimulation” must be set to 175 or 100?
 - DACCOSIM artifact getNextInternalEvent will return:
 - 75, 175, 250, 350, 430, 530 (slide 24)
 - or
 - 75, 100, 200, 250, 300, 400, 430 (next slide)
 - ✓ A new DACCOSIM user parameter is to be proposed to let the user choose between the 2 approaches

DACCOSIM Model Artifact getNextIntervalEventTime algo

“Urgency & periodic output integrated” version



DESM equivalent to the output periodicity P

Model Artifact internal variables:

- **CT**: Current Time of the simulation
- **simu_end**: Simulation end time
- **t_event**: estimated next internal event time
- **DES**: DACCOSIM External Step (mission duration)
- **DESM (DESm)**: Maximum (minimum) DACCOSIM External Step (max / min mission duration)
- **EIV**: External Input Value stack
- **Idx**: Periodicity Loop index (initial value = 1)
- **Exploration_done**: Has the exploration been performed?

Late Event Propagation Problem

- Boolean value changes at instants 2, 9, and 12
- DACCOSIM initial mission is 20 and initial stepSize is 10
- Exploration :
 - Step 0 to 10: no event detection (odd number of value changes)
 - Step 10 to 20: event detected, proposed mission size = 12
- Another agent propagates external event at 8.
- DACCOSIM is asked to roll from 0 to 8, so it decreases its step size to 8 and this time it detects the event at 2, but this event will only be propagated at 8...

Solution

- Waiting for FMI 2.1 (where it is expected it will offer a fine solution for event detection), it is the user responsibility to define a correct value for maximum stepSize (less than the minimum time interval for two consecutive boolean changes)