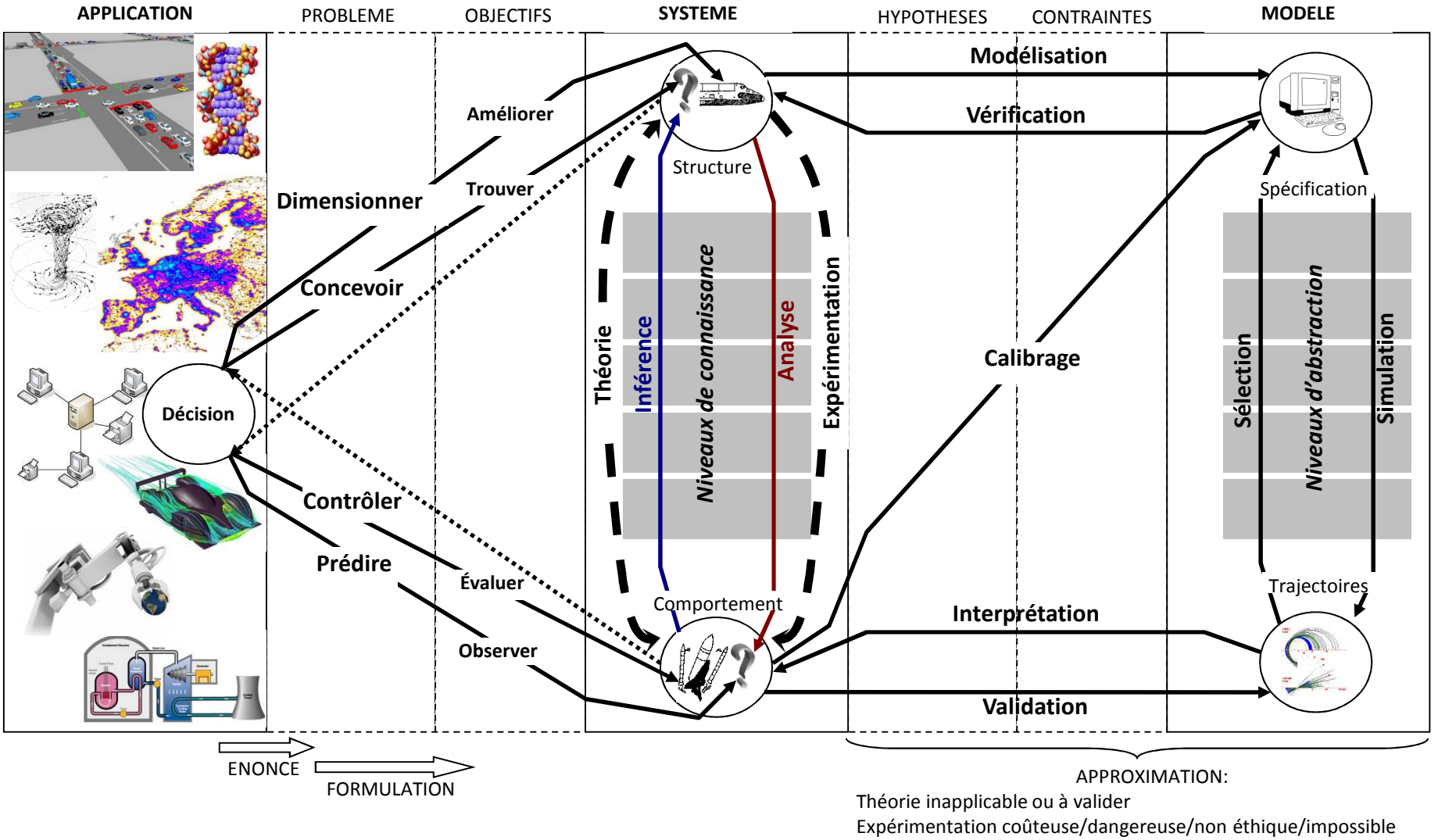
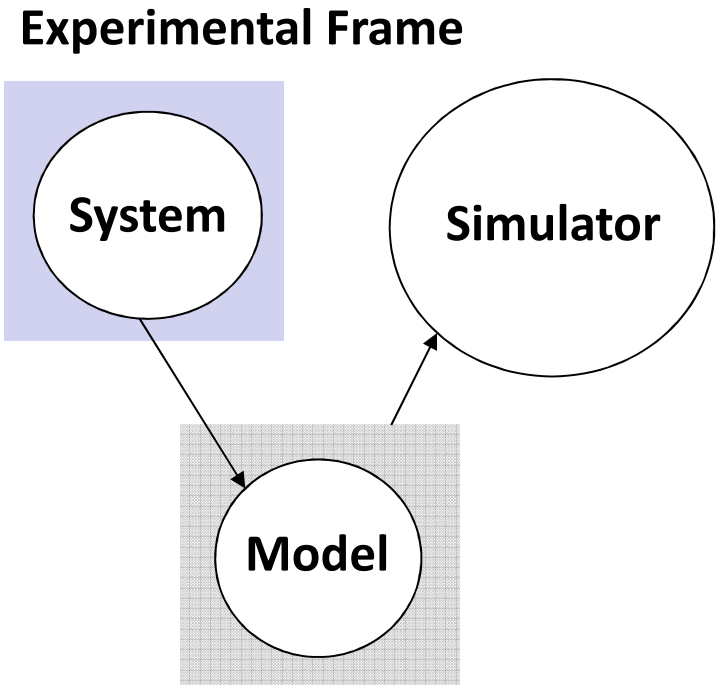
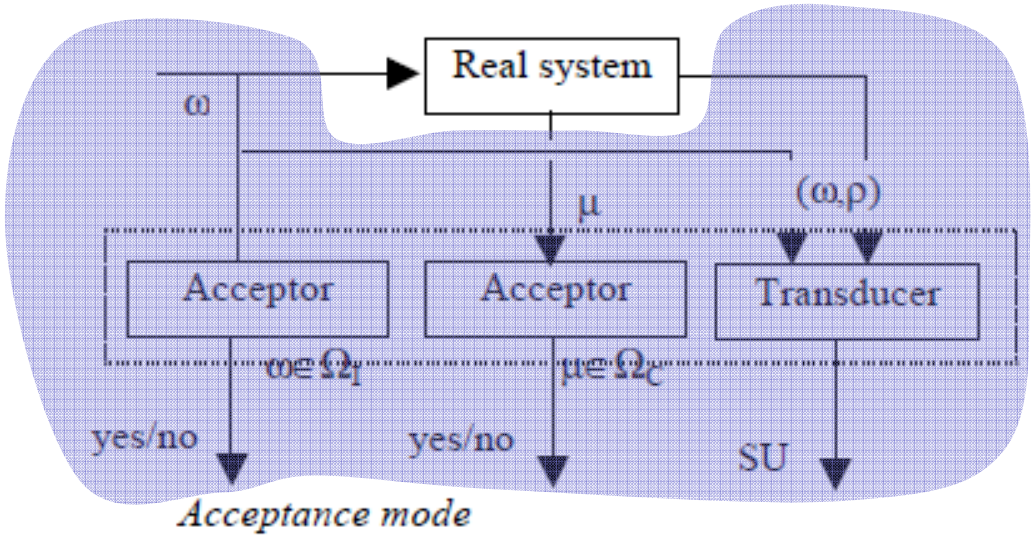
DEVS

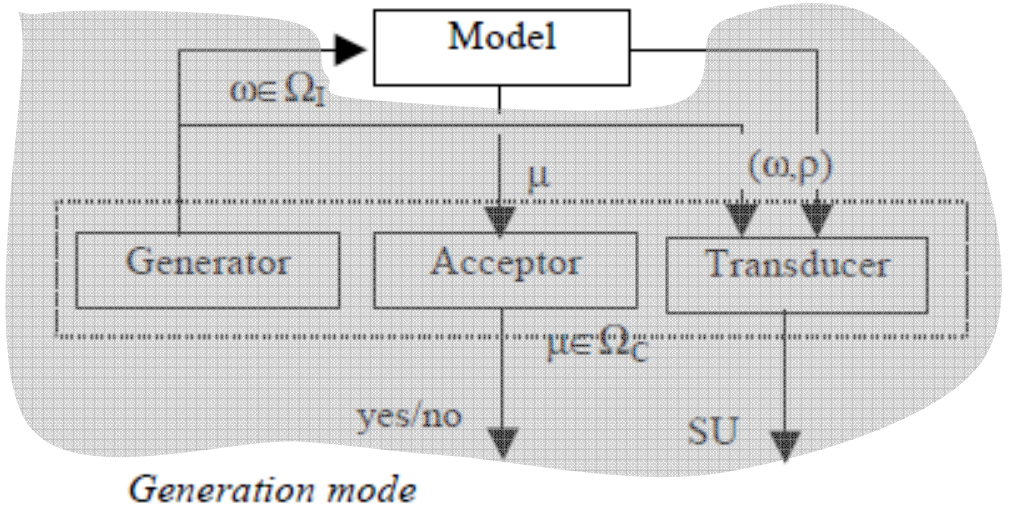
Understanding simulation models in their life-cycle



DEVS paradigm



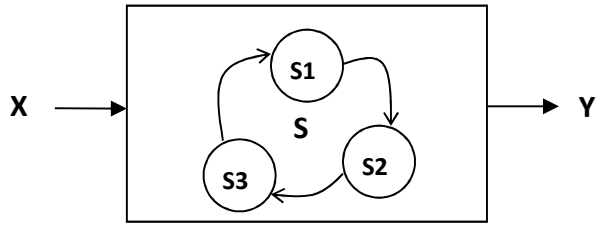


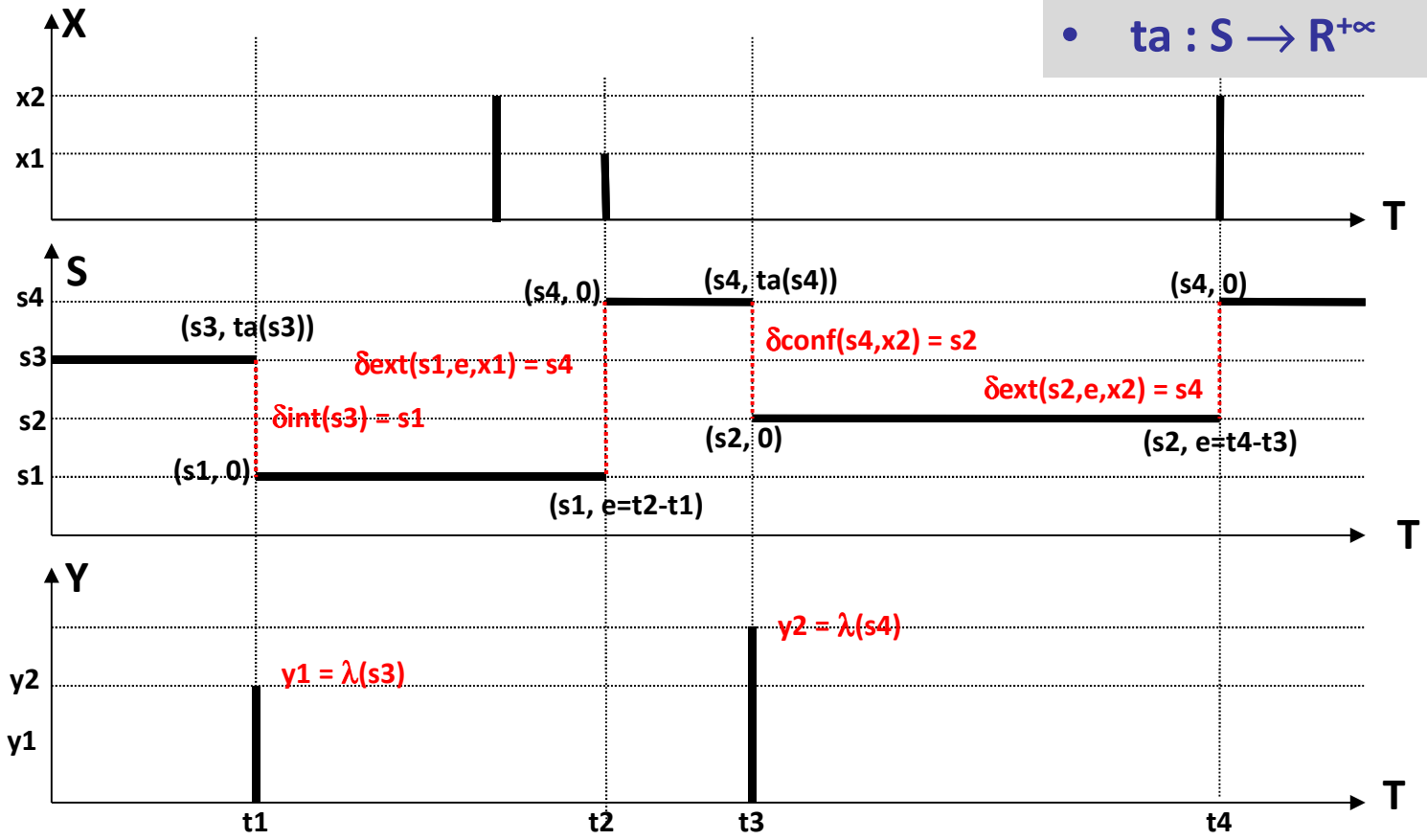


Atomic model

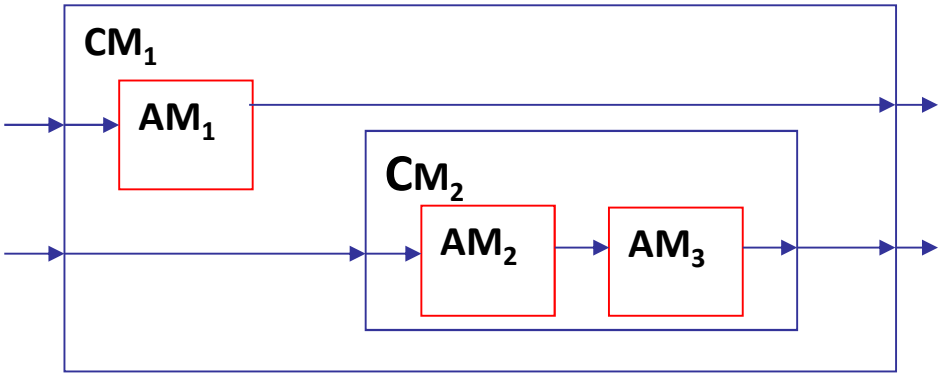
$M = \langle X, Y, S, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$

- $\delta_{int} : S \rightarrow S$
- $\delta_{ext} : Q \times X^b \rightarrow S$
- $Q = \{(s,e) / s \in S, 0 \leq e < ta(s)\}$
- $\delta_{conf} : S \times X^b \rightarrow S$
- $\lambda : S \rightarrow Y^b$
- $ta : S \rightarrow R^{+\infty}$





Coupled model

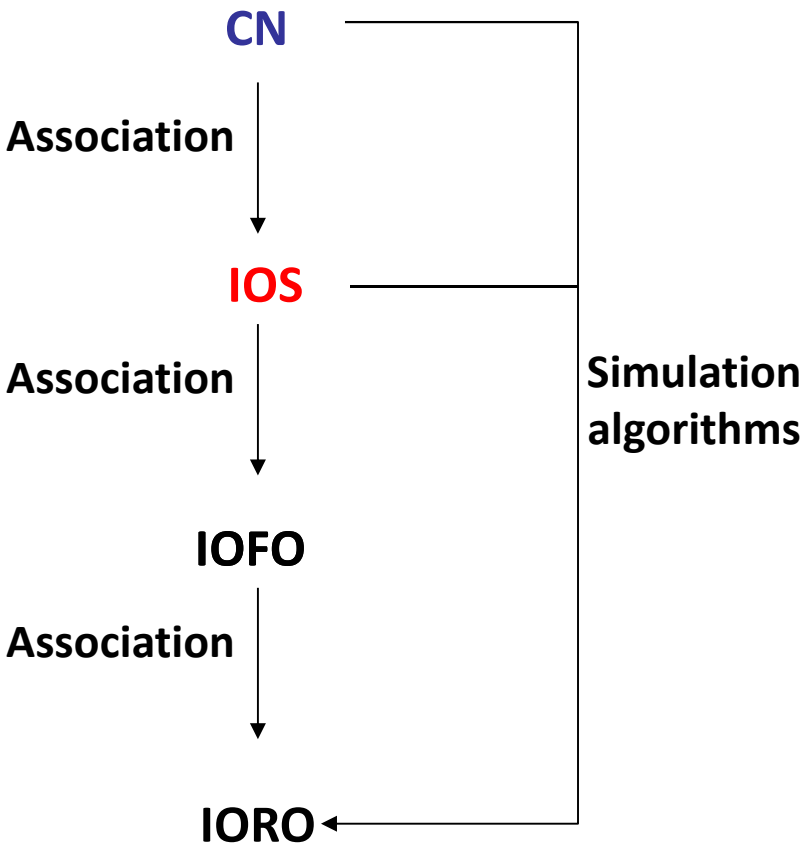


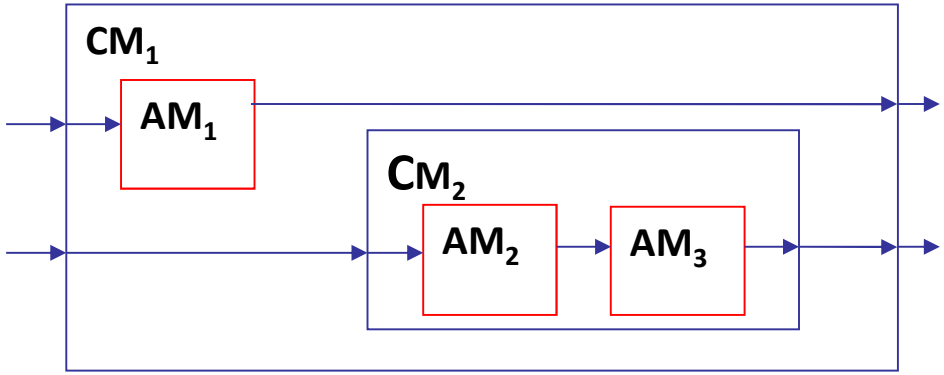
$M = \langle X, Y, D, \{M_d, d \in D\}, EIC, EOC, IC \rangle$

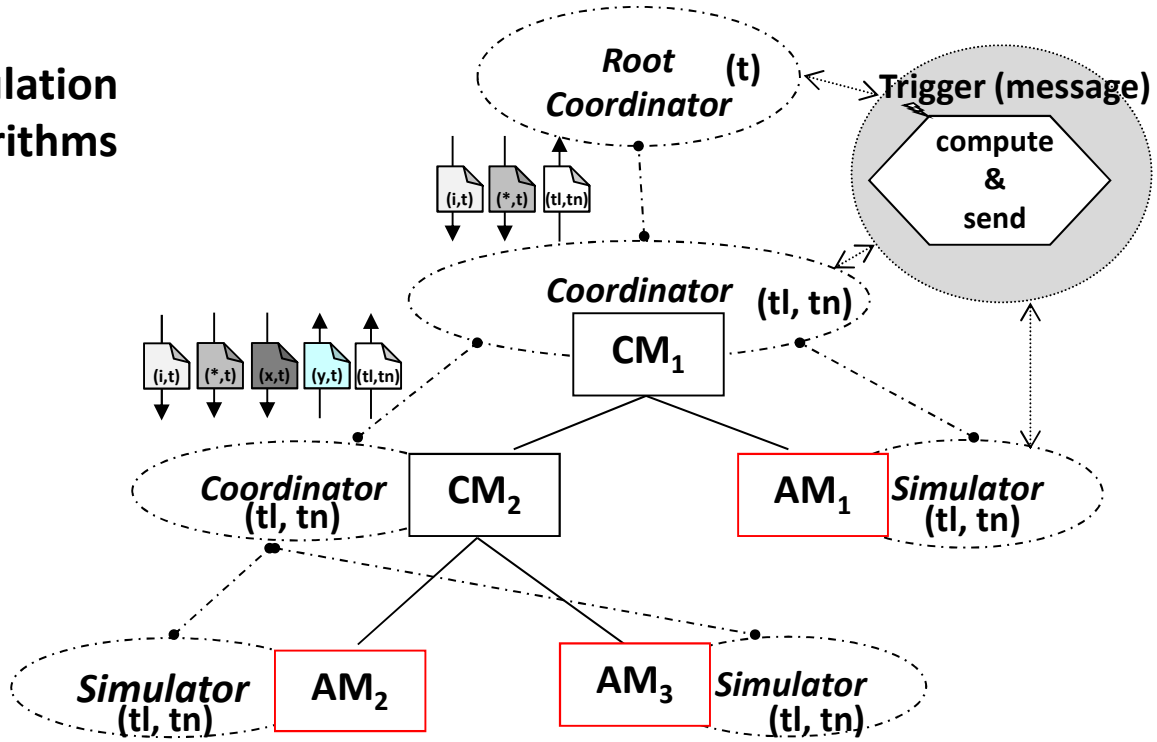
$M = \langle X_{self}, Y_{self}, D, \{M_d\}_D, \{I_d\}_D, \{Z_{i,j}\}_{D \cup \{self\}} \rangle$

- $\forall i \in D \cup \{self\} : I_i \subseteq D \cup \{self\}$
- $\forall i \in D \cup \{self\} : i \notin I_i$
- $Z_{self,j} : X_{self} \rightarrow X_j \ \forall j \in D$
- $Z_{i,self} : Y_i \rightarrow Y_{self} \ \forall i \in D$
- $Z_{ij} : Y_i \rightarrow X_j \ \forall i,j \in D$
- $Select : 2^D \rightarrow D$

Simulation protocol







Simulator

(@,t)	if t = tN then y := λ(s) send (y, t) to the parent coordinator send (done, t) to the parent coordinator else raise error
(q, t)	lock the bag, add event q to the bag and unlock the bag send (done, t) to the parent coordinator
(*, t)	case tL ≤ t < tN and bag is not empty e := t -tL s := δ_{ext}(s,e,bag) empty bag tL := t tN := tL + ta(s) case t = tN and bag is empty s := δ_{int}(s) tL := t tN := tL + ta(s) case t = tN and bag is not empty s := δ_{con}(s,bag) empty bag tL := t tN := tL + ta(s) case t > tN or t < tL raise error send (done, tN) to parent coordinator

Coordinator

(@,t)	if t = tN then tL := t for all imminent child processors i with minimum tN send (@, t) to child i cache i in the synchronize set wait until (done, t) are received from all imminent processors send (done, t) to the parent coordinator else raise an error
(q, t)	lock the bag, add event q to the bag and unlock the bag
(*, t)	if tL <= t < tN then for all receivers, j $\in$ I_{self} and all q $\in$ bag q := $Z_{self,j}$(q) send (q, t) to j cache j in the synchronize set empty bag wait until all (done, t) are received for all i in the synchronize set send (*, t) to i wait until all (done, tN) are received tL := t and tN := minimum of components' tN's clear the synchronize set send (done, t) to parent coordinator else raise an error
(y, t)	for all influencees j of child i q := $Z_{i,j}$(y) send (q, t) to child j cache j in the synchronize set wait until all (done, t) are received from j's if self $\in$ I_i (y is to be transmitted upward) then y := $Z_{i,self}$(y) send (y, t) to the parent coordinator

Root coordinator

t := tN of the topmost coordinator
while t ≠ ∞
send (@, t) to the topmost coordinator
wait until (done, t) is received from it
send (*, t) to the topmost coordinator
wait until (done, tN) is received from it
raise simulation completed