

# TP COO : Conception et implémentation d'un simulateur à événements discrets de systèmes hybrides

## 1 Le modèle

### Définition du composant atomique

Un composant atomique est une structure :

$$M = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta \rangle \text{ où} \quad (1)$$

- $X = \{p, v\} \mid p \in IPorts, v \in X_p$ , est l'ensemble des ports d'entrée et leurs valeurs,
- $Y = \{p, v\} \mid p \in OPorts, v \in Y_p$ , est l'ensemble des ports de sortie et leurs valeurs,
- $S$  est l'ensemble des états séquentiels
- $\delta_{ext} : Q \times X \rightarrow S$ , est la fonction de transition externe, où
  - $Q = \{(s, e) \mid s \in S, 0 \leq e \leq ta(s)\}$  est l'état total
  - $e$  est le temps écoulé depuis la dernière transition
- $\delta_{int} : S \rightarrow S$ , est la fonction de transition interne,
- $\lambda : S \rightarrow Y$ , est la fonction de sortie,
- $ta : S \rightarrow \mathbb{R}_{0,\infty}^+$ , est la fonction d'avancement du temps,

### Définition du composant couplé

Un composant couplé est une structure :

$$N = \langle X, Y, D, C \rangle \text{ où}$$

- $X$  est l'ensemble des ports d'entrée et leurs valeurs,
- $Y$  est l'ensemble des ports de sortie et leurs valeurs,
- $D$  est l'ensemble des noms des composants, un élément  $d \in D$  peut soit être un composant atomique ou un composant couplé.
- $C$  est la fonction de couplage. On peut coupler des sorties d'un composant  $d_1$  à des entrées d'un composant  $d_2$ , des entrées du composant couplé à des entrées d'un de ses composants  $d$  ou des sorties d'un de ses composants  $d$  aux sorties du composant couplé. Tout les autres types de connexion sont interdits. On ne peut pas connecter une sortie d'un composant  $d$  à une entrée d'un même composant  $d$ .

## 2 Exemple du gen-buf-proc

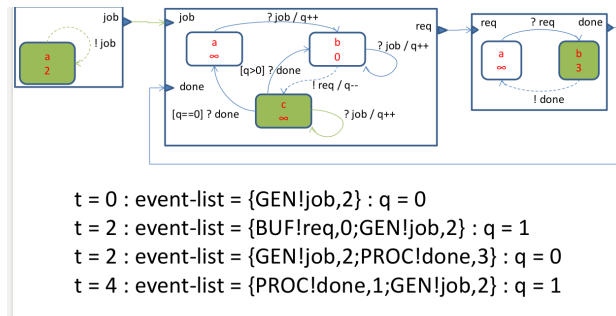


FIGURE 1 – Exemple GBP

### 3 Pseudo code partiel du simulateur

**Algorithm 1**    ordonnanceur

```
Instantier les composants atomiques et les ajouter à une liste C
 $t = 0$ 
 $t_{fin} = 10$ 
for c in C
    récupérer le temps restant avant le prochain événement tr
    construire une liste de composant imminents imms
    stoker le plus petit tr dans  $tr_{min}$ 
for c in imms
    c.lambda()
for c in C
    construire la liste des entrées ins impactées par les sorties produites
for c in C
    if c in imms et c.ins = null
        c.internal
    if c not in imms et c.ins not null
        c.external
    if c in imms et c.ins not null
        c.conflict
    else
        ...
 $t = t + tr_{min}$ 
end
```