

# SysML/UML

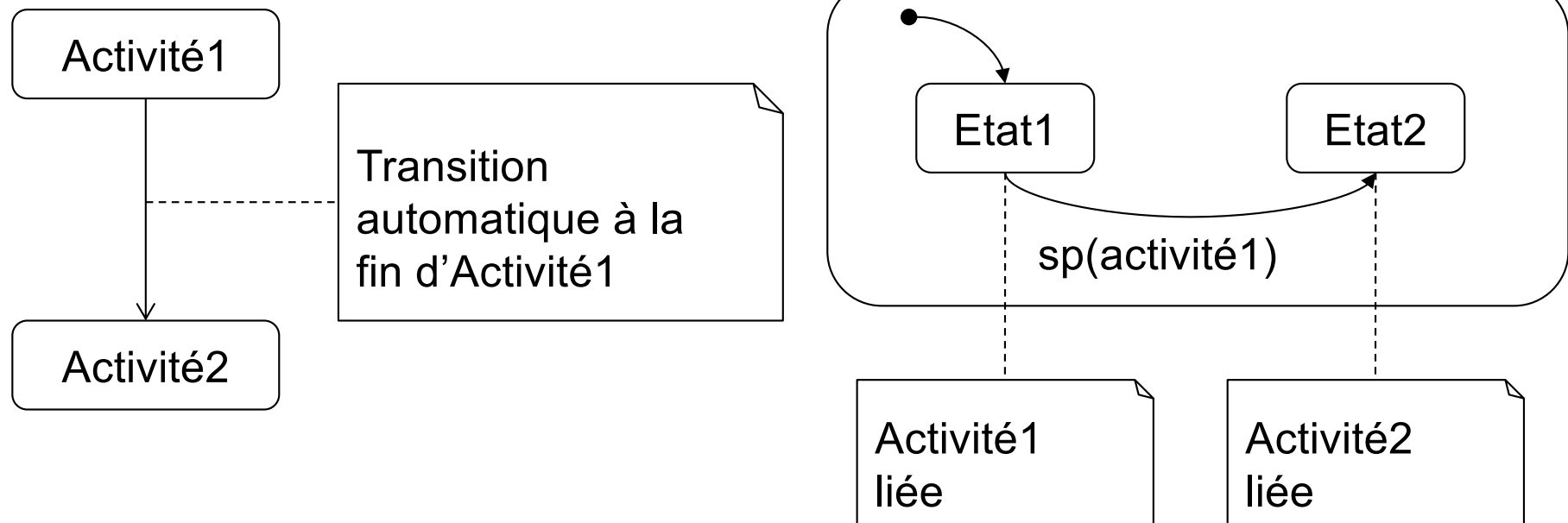
## Diagrammes d'activités

# Diagrammes d'activités

2

- Variantes des diagrammes états/transition en mettant l'accent sur les activités. Peuvent être vus comme des statecharts «à plat» ou chaque état à une activité liée :

Diagramme d'activités :

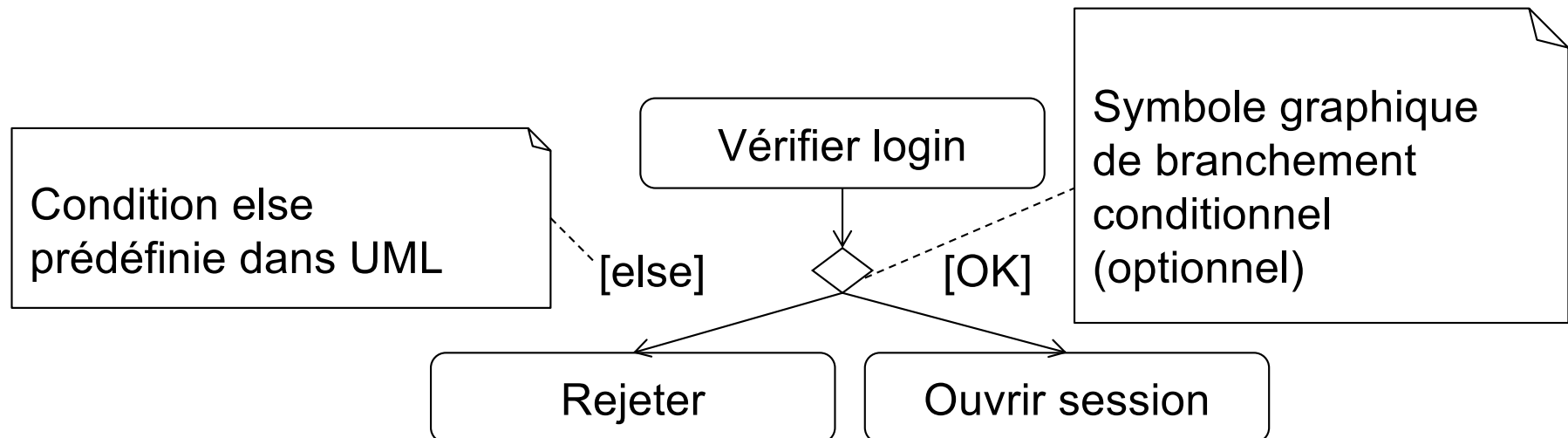


# Diagrammes d'activités : transitions

3

- La plupart des transitions sont automatiques. Il est toutefois possible d'utiliser les mêmes labels que dans les statecharts : Événement[Condition]/Action :

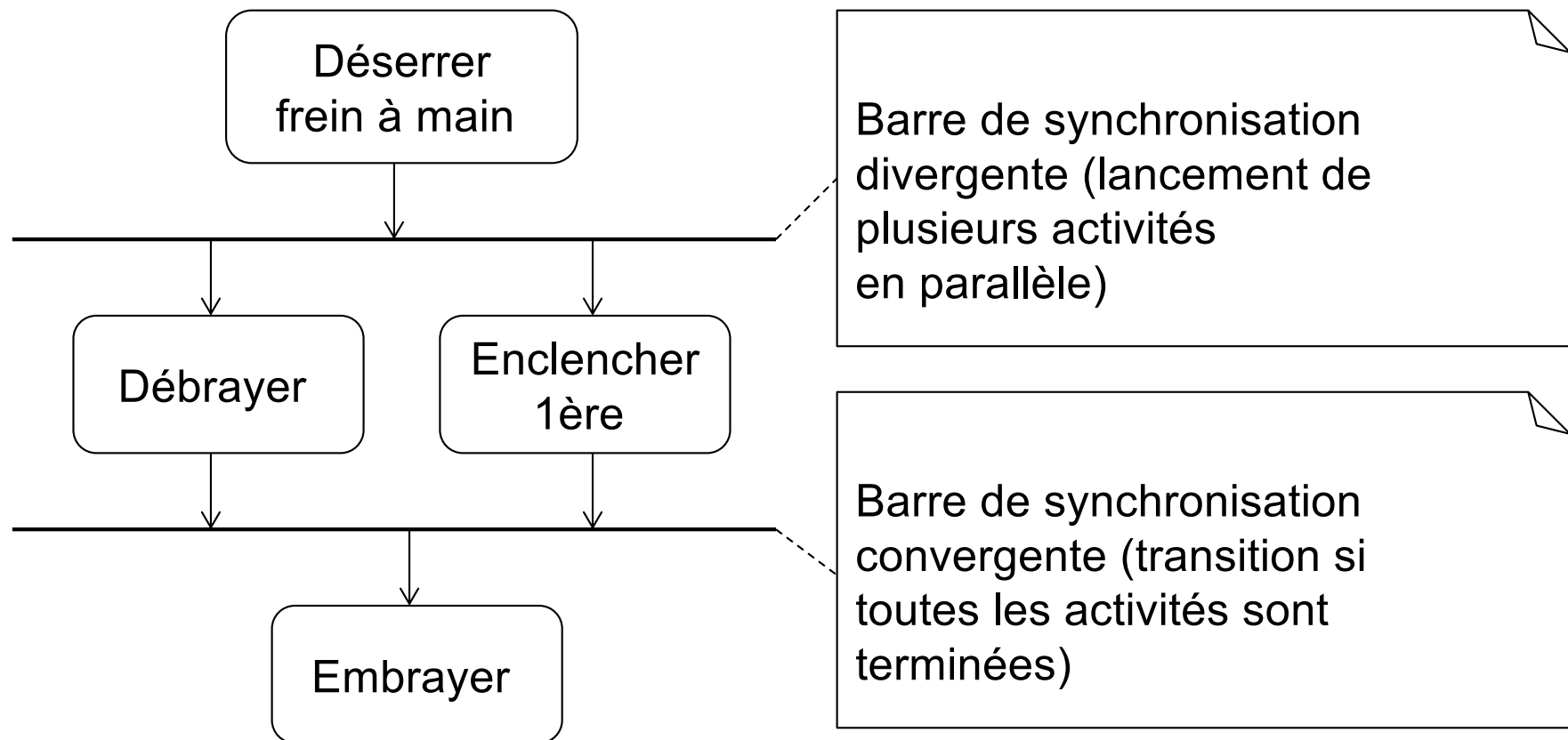
Exemple d'utilisation d'une condition :



# Diagrammes d'activités : synchronisations

4

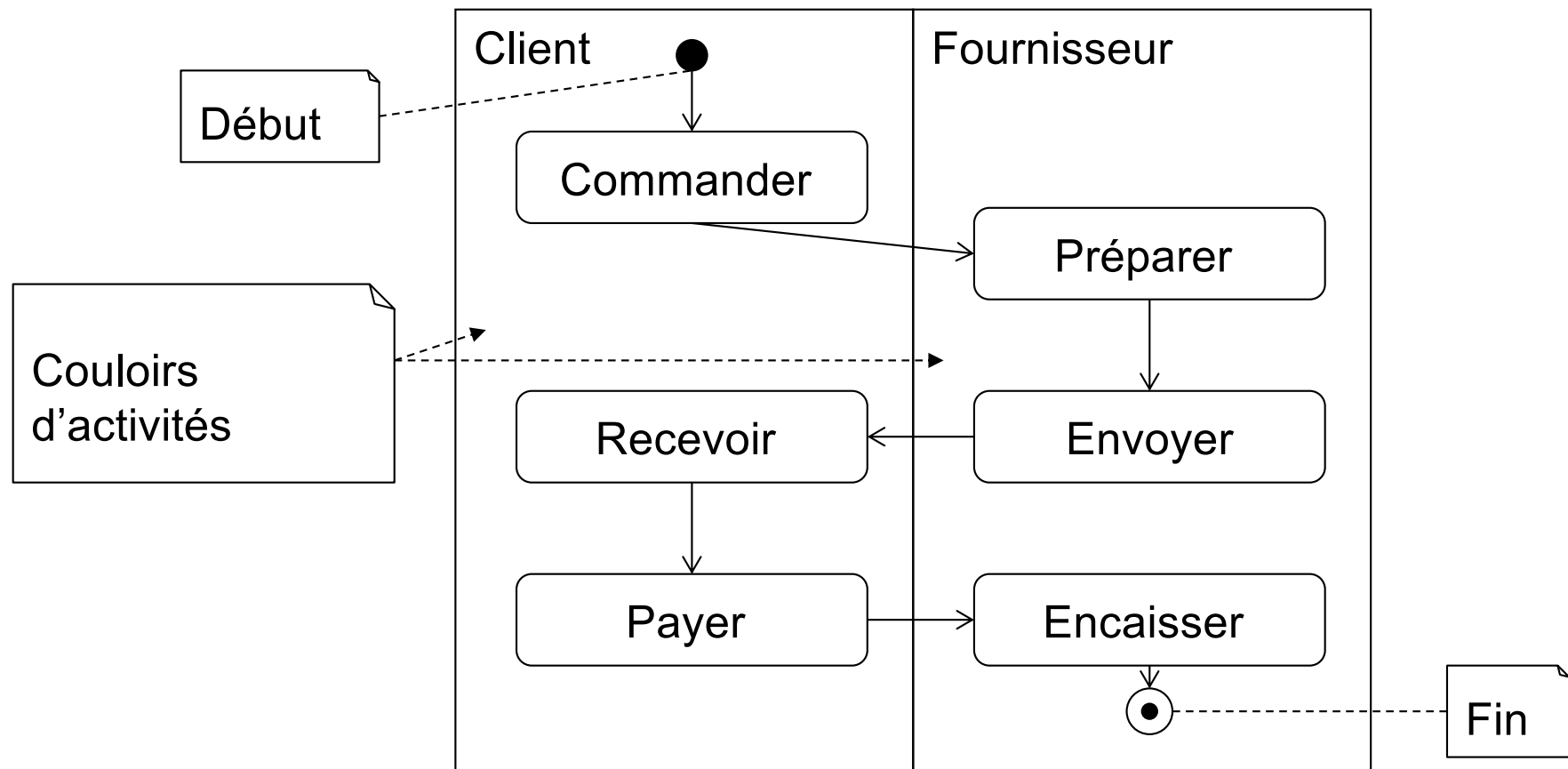
- Il est possible de spécifier l'exécution simultanée et la synchronisation de plusieurs activités :



# Diagrammes d'activités : travées ou couloirs d'activités

5

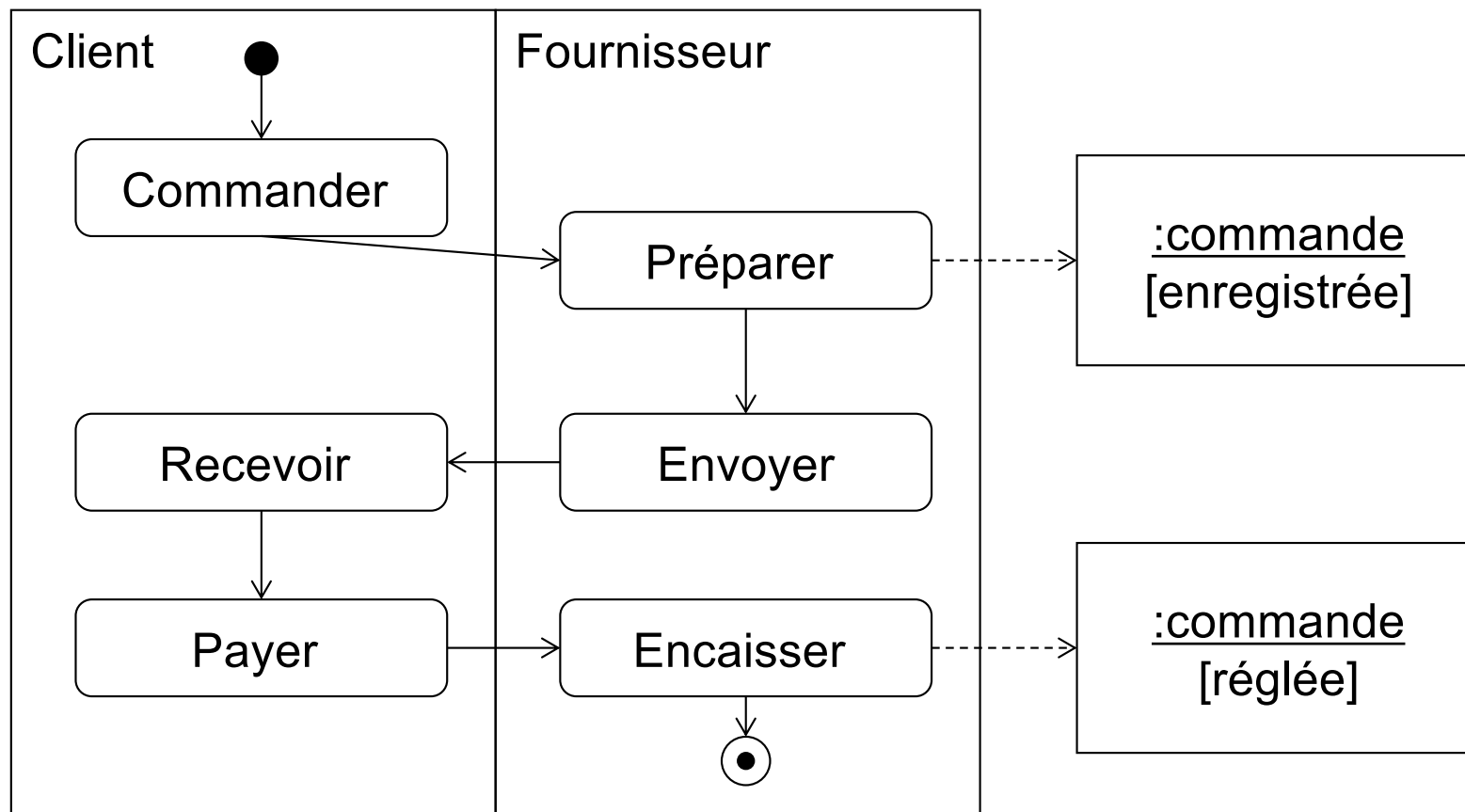
- Il est possible d'organiser un diagramme en travées ou couloirs selon les responsables des diverses activités :



# Diagrammes d'activités : travées ou couloirs d'activités

6

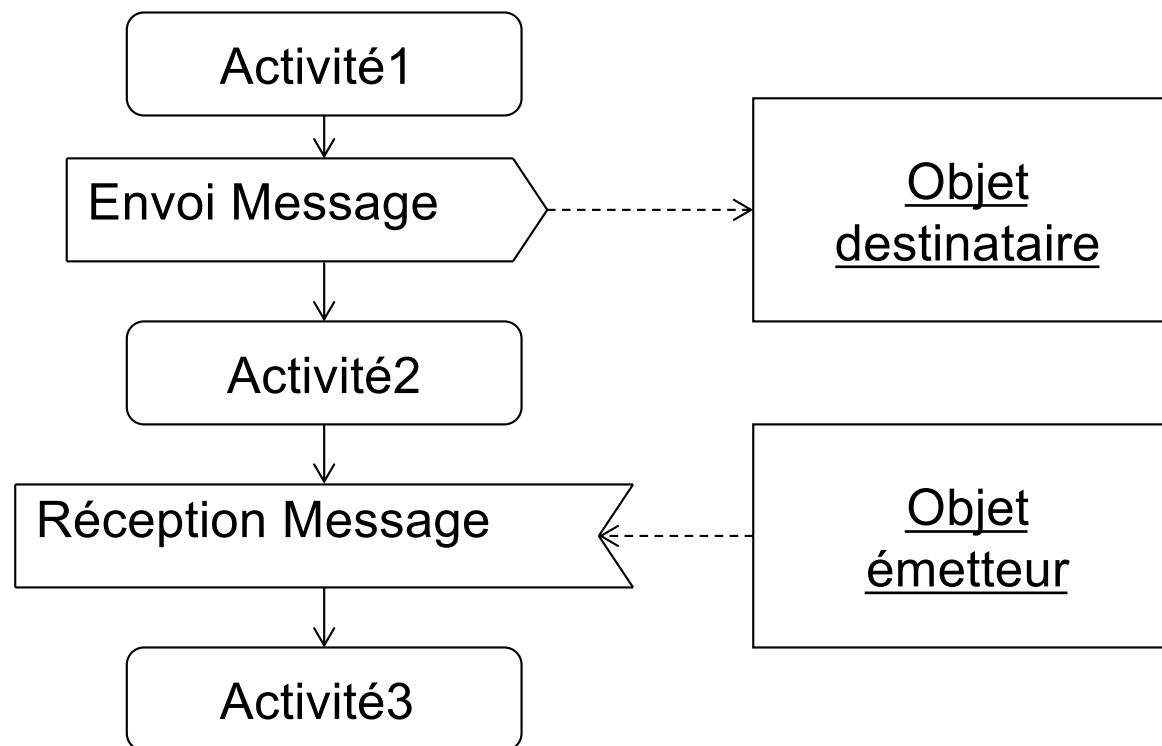
- Il est également possible de faire figurer les objets modifiés par les activités (avec l'état modifié entre [])



# Diagrammes d'activités : envois et réception de messages

7

- On peut aussi représenter les envois et réceptions de message sur les transitions entre activités (en faisant éventuellement figurer les objets destinataires et émetteur) :



# Adaptations SysML

8

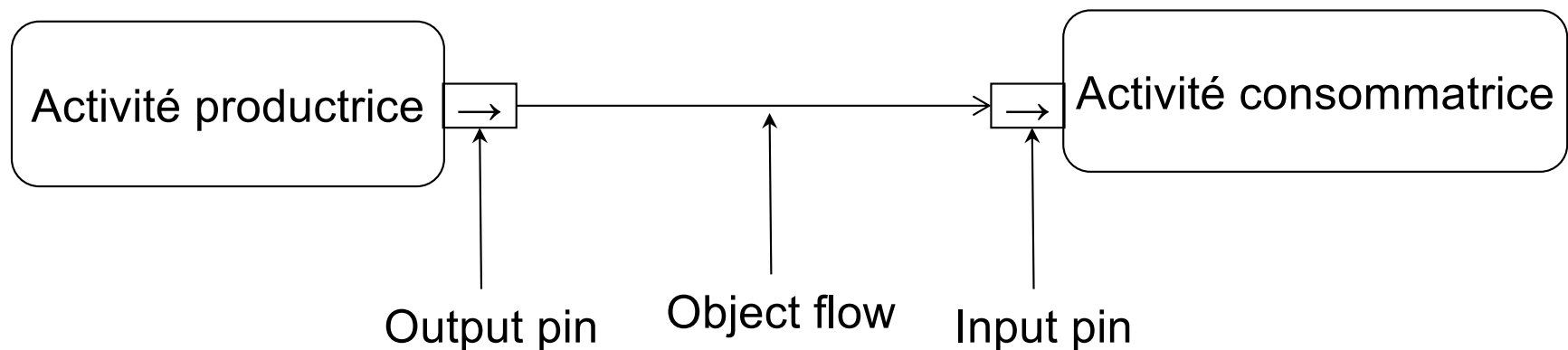
- Dans SysML les arcs orientés reliant les activités sont appelés **flots de contrôle** (control flow)
- Lorsqu'elles sont terminées les activités produisent des **jetons** (tokens) qui circulent sur les flots de contrôle et démarrent l'activité suivante.
- Les jetons se séparent en plusieurs jetons aux **fork** et fusionnent aux **joint**.
- La rigueur sur le nombre de jetons amène à introduire un connecteur **losange convergent** (matérialisant le END IF)



# Adaptations SysML

9

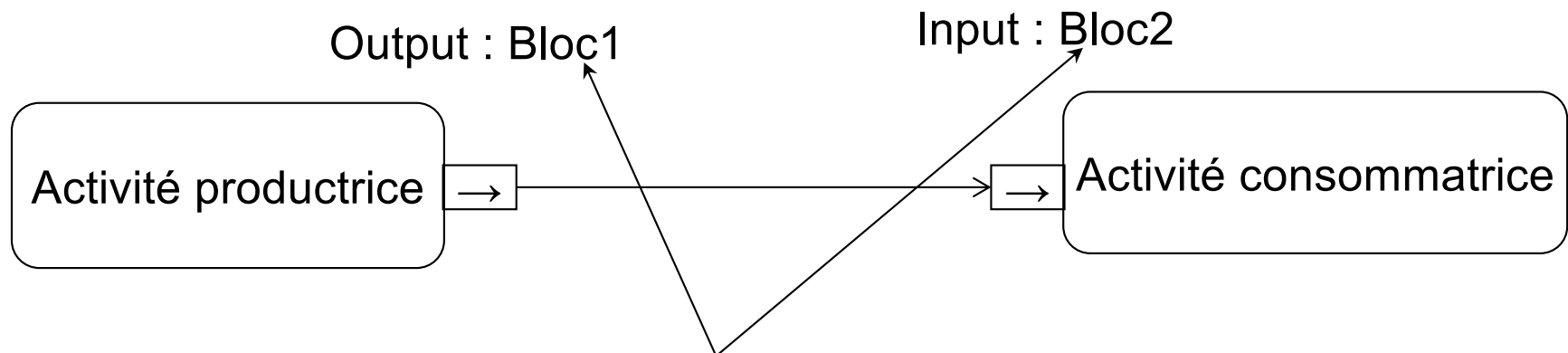
- SysML introduit aussi la notion de **flots d'objets** (object flow) : les activités peuvent produire et consommer des objets et sont dans ce cas reliés par un object flow.
- Ces flots sont connectés au niveau de l'activité productrice à une **broche de sortie** (output pin)
- Ils sont connectés au niveau de l'activité consommatrice par une **broche d'entrée** (input pin)



# Adaptations SysML

10

Les objets produits en sortie ou consommés en entrée sont **typés bloc** (on rappelle qu'une classe donc un bloc est un type abstrait de données). Les types doivent correspondre (ou être liés par une relation de généralisation/spécialisation).

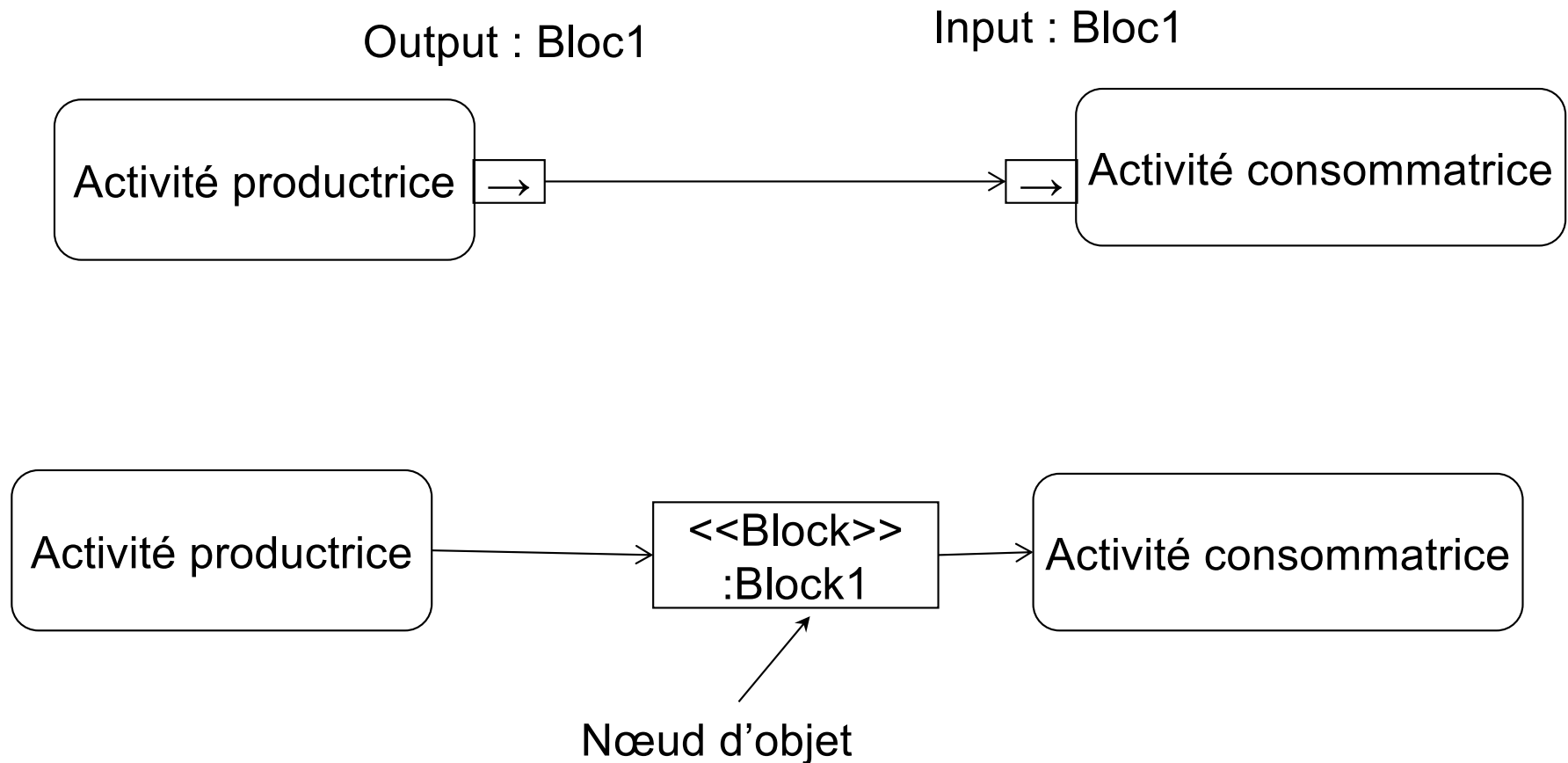


Les types Bloc1 et Bloc2 doivent être compatibles  
(identiques ou liés par généralisation/spécialisation)

# Adaptations SysML

11

Lorsque les types en entrée et sortie sont identiques deux représentations possibles sont admises

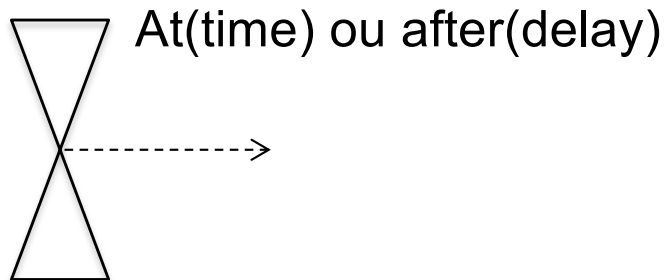


# Adaptations SysML

12

En plus des pentagones convexes « envoi de message » et concave « réception de message », SysML définit un type de nœud « accept time event » basé sur des critères temporels

- **At** : à un temps précisé
- **After** : au bout d'un délai précisé



# Adaptations SysML

13

- SysML offre aussi la possibilité d'indiquer que toute une partie du diagramme est interruptible (**région interruptible**)
- La région en question doit comporter un pentagone accept event, suivie d'un flot d'interruption

