

# Modélisation des systèmes à événements discrets

## Partie 3

- Règles de franchissement générales
- Étude de cas
- Traitement des objets Smalltalk structurés
- Opérations courantes sur les objets

Denis Giacona

ENSISA

Ecole nationale supérieure d'ingénieurs sud Alsace

12, rue des frères Lumière

68 093 MULHOUSE CEDEX

FRANCE

Tél. 33 (0)3 89 33 69 00



1. Règles de franchissement générales.....	4
1.1. Terminologie.....	4
1.2. Résolution des conflits (cas de transitions non temporisées).....	5
1.3. Conditions d'activation (de validation) .....	6
1.4. Activation d'une transition .....	8
1.5. Effet du tir .....	9
1.6. Exemples d'activation.....	10
1.7. Le simulateur.....	12
1.7.1. La liste d'événements du simulateur .....	12
1.7.2. Algorithme de simulation.....	13
1.7.3. L'histoire du simulateur .....	14
Le simulateur conserve la trace des transitions qui ont été tirées grâce à une liste de type FIFO dont la longueur est définie par l'utilisateur (20 par défaut).....	14
1.7.4. Modes d'affichage du simulateur .....	15
2. Etude de cas (pour des transitions non temporisées).....	16
2.1. Sélection et spécification d'objets avec des attributs littéraux .....	16
2.2. Sélection et spécification d'objets avec des attributs variables .....	18
2.3. Sélection d'objets identiques en provenance de plusieurs sources.....	19
2.4. Blocage d'un traitement avec un arc inhibiteur à attribut.....	20
2.5. Blocage d'un traitement avec une place sortante à capacité.....	21
2.6. Sélection d'objets par un code condition.....	22
3. Traitement des objets structurés.....	25
3.1. Petit aperçu des objets Smalltalk.....	25
3.2. Les tests d'égalité et d'identité .....	26
3.3. Sélection d'objets structurés .....	27
3.4. Modification d'un objet structuré.....	29

- 4. Opérations courantes sur les objets .....33
  - 4.1. Comptage d'opérations ..... 33
  - 4.2. Attributs d'arc sortant fonctions d'attributs d'un arc entrant ..... 34
  - 4.3. Génération d'un nombre fini de jetons .....35

# 1. Règles de franchissement générales

## 1.1. Terminologie

- **franchissement** et **tir** sont synonymes
- une transition est franchissable à condition qu'elle soit **activée**
- activation et **validation** sont synonymes
- l'activation d'une transition dépend de plusieurs **conditions** (présence dans les places amont de jetons **conformes** aux inscriptions d'arc, **code condition = true**)
- franchissement d'une transition activée :
  - immédiat si et seulement si le délai associé est nul
  - retardé si le délai associé est non nul
    - le **code délai** exprime le **retard** du franchissement
    - la **planification** d'un tir peut être annulée si avant l'échéance du tir l'une des ressources nécessaires a disparu
- si plusieurs transitions sont franchissables à la même date et que le franchissement de l'une empêche le franchissement des autres, on dit qu'elles sont en **conflit**

## 1.2. Résolution des conflits (cas de transitions non temporisées)

Soient deux transitions non temporisées ayant une même place d'entrée. Si elles sont validées par la même marque d'entrée, alors une seule transition sera franchie. Le choix de la transition dépendra de la **configuration** de l'outil de simulation :

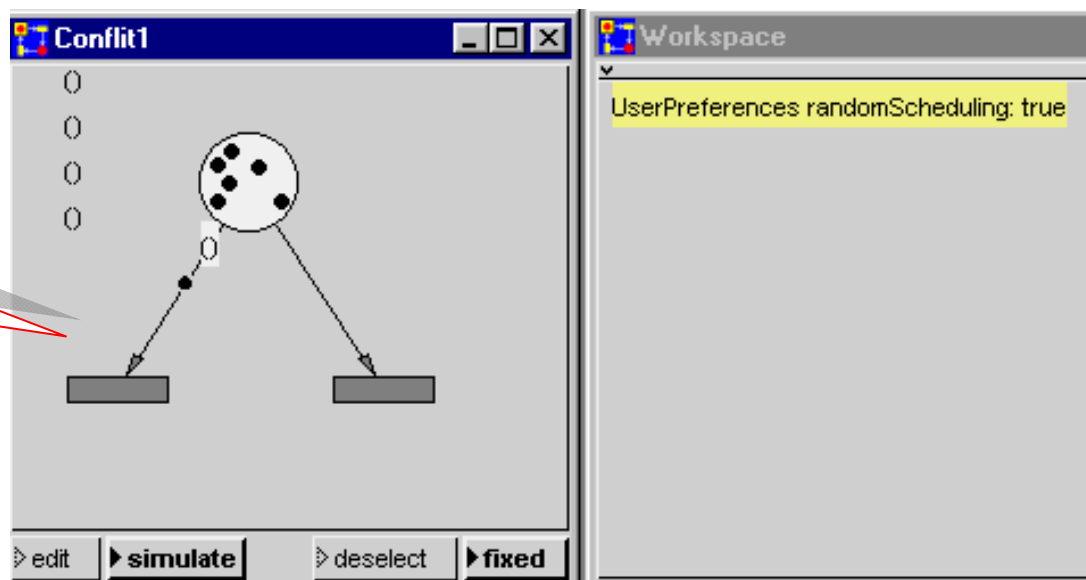
`UserPreferences randomScheduling: false`

→ le choix est fixé par l'implémentation

`UserPreferences randomScheduling: true`

→ le choix est aléatoire

Pour cette configuration, il y a en moyenne autant de franchissements à gauche qu'à droite.



## 1.3. Conditions d'activation (de validation)

- La condition d'activation dépend d'abord des arcs ordinaires :
  - pour un arc entrant sans inscription, chaque place amont doit comporter au moins un jeton ordinaire
  - si un arc entrant comporte  $n$  attributs, la place amont doit comporter au moins un jeton à  $n$  attributs, et dans ce cas :
    - si un attribut d'arc est constant, l'attribut correspondant du jeton doit être identique à cette constante
    - si un attribut d'arc est une variable, la valeur de l'attribut correspondant du jeton est quelconque
    - si plusieurs attributs du même arc contiennent la même variable, les attributs correspondants du jeton doivent être identiques, peu importe leurs valeurs
  - si plusieurs arcs entrants ont des attributs identiques, on doit trouver des jetons à attributs identiques dans l'ensemble des places amont correspondantes

- La condition d'activation dépend ensuite des arcs inhibiteurs :
  - pour un arc inhibiteur sans inscription, la place amont ne doit pas contenir de jeton ordinaire
  - si un arc inhibiteur comporte une inscription à  $n$  attributs, alors pour tous les jetons à  $n$  attributs de la place amont :
    - si un attribut de l'arc est un littéral, l'attribut correspondant de tous ces jetons ne doit pas être identique à ce littéral
    - si un attribut de l'arc est une variable, toute valeur pour l'attribut correspondant de tous ces jetons empêche l'activation de la transition
    - si plusieurs attributs du même arc contiennent la même variable, les attributs correspondants des jetons ne doivent pas être identiques, peu importe leurs valeurs
  
- La condition d'activation dépend enfin du code condition :
  - le code condition de la transition doit rendre la valeur `true` pour les valeurs courantes des variables

## 1.4. Activation d'une transition

### □ Planification du tir

Lorsque les conditions d'activation sont réunies, le tir est d'abord **planifié** :

- si la transition est **non temporisée**, le tir a lieu immédiatement
- si la transition est **temporisée** le tir aura normalement lieu à  $t = \text{date d'arrivée du jeton conforme} + \text{délai évalué par le code délai}$  (sauf retard supplémentaire induit par une place à capacité, ou annulation de la planification)

### □ Prolongation de la planification d'un tir (cas de places de sortie à capacité)

- si le délai prévu a expiré et qu'une place sortante à capacité est saturée, le tir est momentanément retardé ; l'activation est maintenue et le tir aura lieu dès qu'un jeton quitte la place à capacité

### □ Annulation de la planification d'un tir

- soit une place **commune à plusieurs transitions aval**  $t_1, t_2, \dots, t_n$ . Si un jeton a initialement validé  $t_1$ , et que ce jeton a disparu avant la date prévue suite au franchissement de  $t_2$  ou ...  $t_n$ , la planification du tir de  $t_1$  est annulée



## 1.5. Effet du tir

- Le code action est exécuté **chaque fois** qu'un jeton est retiré d'une place amont
- Les marques qui valident la transition sont supprimées des places amont
- Des marques sont introduites dans les places aval conformément aux inscriptions des arcs sortants (constantes ou objets pointés par les variables)

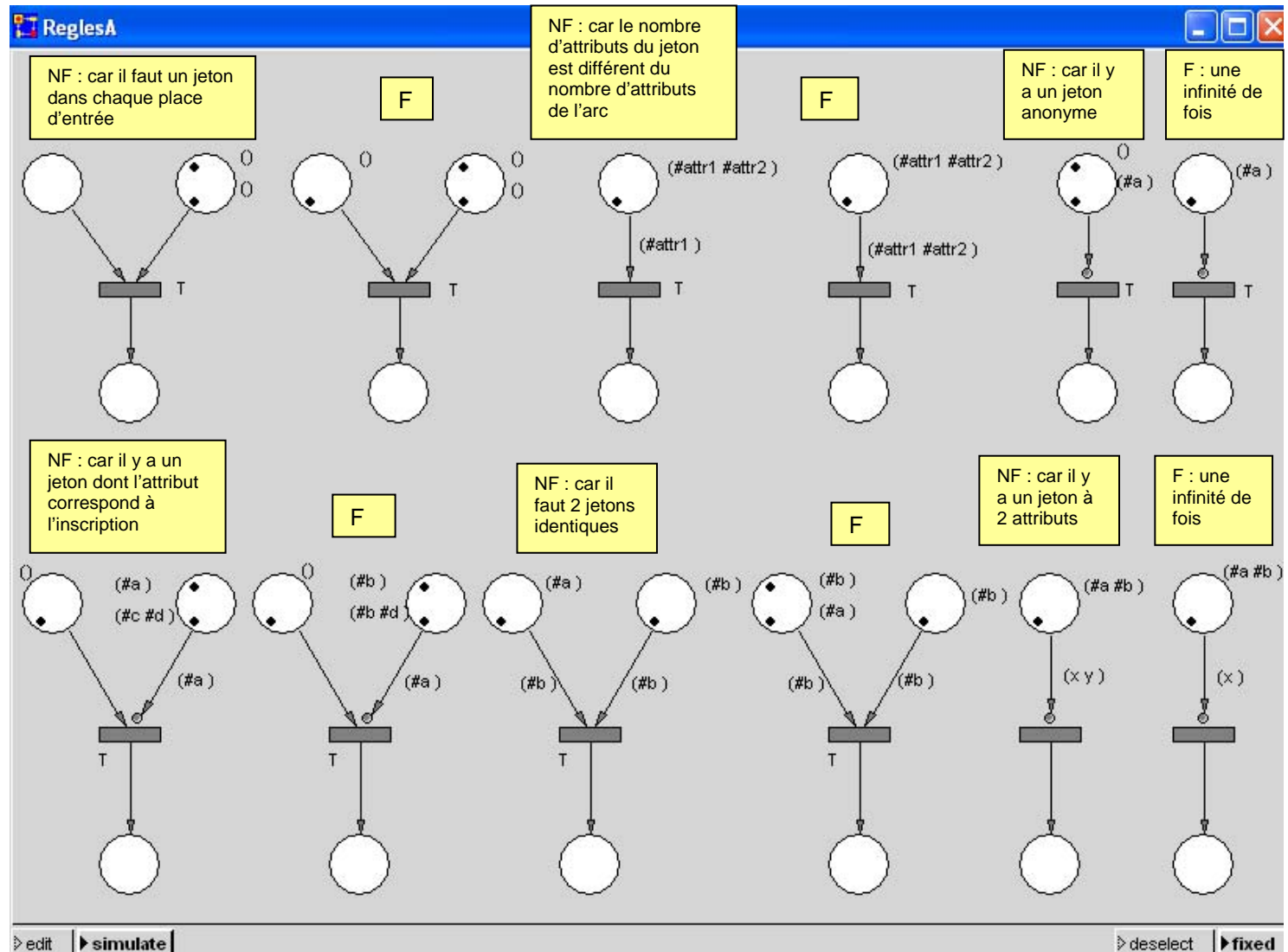
# 1.6. Exemples d'activation

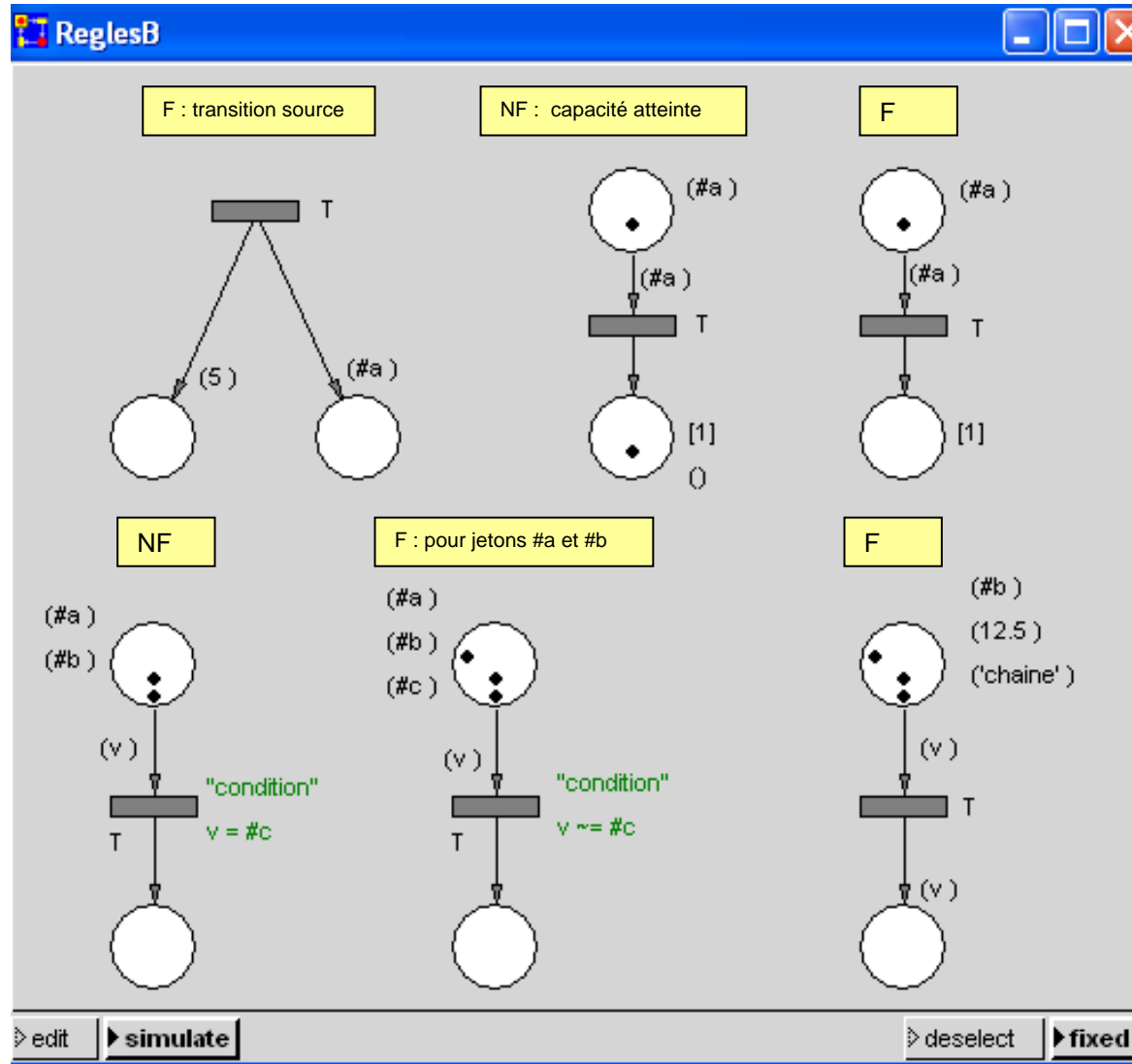
Dans les exemples ci-contre, la transition est non temporisée.

Dès qu'elle est activée (validée), elle devient franchissable (tirable), et est immédiatement franchie (tirée).

NF : Non Franchissable

F : Franchissable





## 1.7. Le simulateur

### 1.7.1. La liste d'événements du simulateur

Le simulateur gère une **liste d'événements** qui ont déjà eu lieu ou qui auront lieu.

#### □ Les événements

Un événement comporte les informations suivantes :

- une transition (à tirer)
- des jetons d'entrée
- des liens avec les variables de transition

#### □ Liste d'événements

Les entrées (éléments) de cette liste sont rangées par **dates croissantes**. Tout élément de la liste comporte :

- une date
- un ensemble d'événements inscrits pour cette date

#### □ Mise à jour de la liste d'événements

Lorsqu'un nouvel événement doit être inscrit pour une date donnée, la liste est parcourue :

- si une entrée (élément) existe déjà pour cette date, l'événement est simplement ajouté à l'ensemble des événements existants
- sinon un nouvel élément est inséré à l'endroit adéquat de la liste.

## 1.7.2. Algorithme de simulation

**L'horloge du simulateur** est un compteur qui représente un nombre entier ou réel dans une unité quelconque à définir en rapport avec le processus réel à simuler.

- partant de l'instant initial 0 (ordre **initialize** du menu simulateur), pour chaque date inscrite, des tirs ont lieu tant qu'il reste des événements dans la liste pour cette date ou des transitions franchissables immédiatement (transitions sans délai)
- lorsque la liste ne contient plus d'événement pour la date de simulation courante et s'il n'y a plus de transitions directes validées, l'élément est retiré de la liste tandis que le temps du simulateur passe à la valeur figurant dans la prochaine entrée de la liste
- la simulation s'arrête lorsque la liste d'événements est vide et qu'il n'y a plus de transition directe franchissable

### 1.7.3. L'histoire du simulateur

Le simulateur conserve la trace des transitions qui ont été tirées grâce à une liste de type FIFO dont la longueur est définie par l'utilisateur (20 par défaut).

### 1.7.4. Modes d'affichage du simulateur

#### □ En mode visualisation dynamique (ordre **run** du menu simulateur)

- les franchissements sont tous affichés séquentiellement sur la fenêtre d'écran (un seul jeton se déplace à la fois)
- l'horloge reste bloquée pour tous les tirs qui sont exécutés à la même date → c'est ainsi que se traduit la simultanéité

#### □ En mode exécution en arrière-plan (ordre **background run** du menu simulateur)

- seul le marquage final est affiché

#### □ En mode exécution arrière (ordre **back** du menu simulateur)

- les jetons reviennent dans les places amont de la transition
- note : les actions de type "effet de bord" ne sont pas annulées

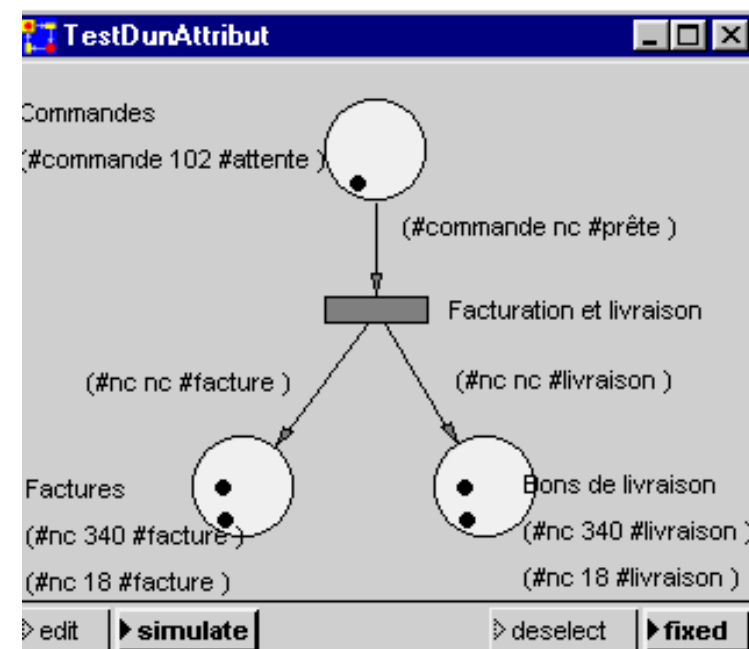
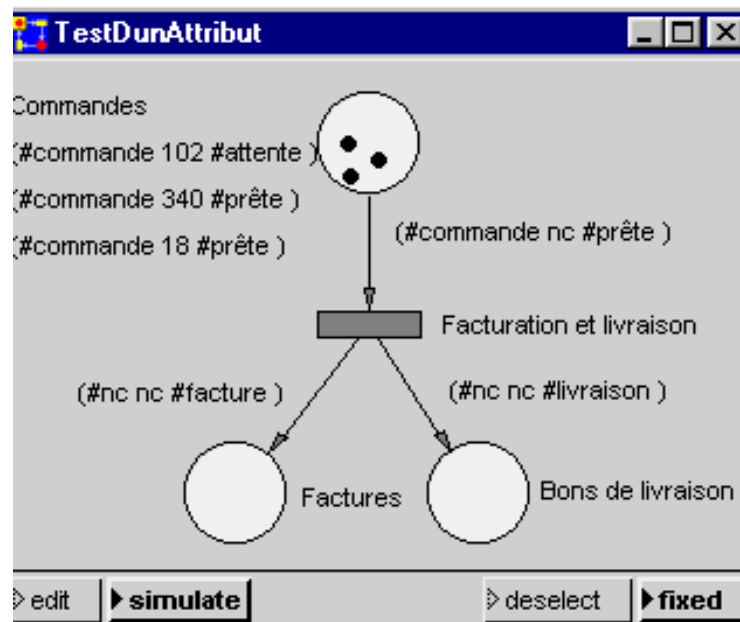
## 2. Etude de cas (pour des transitions non temporisées)

### 2.1. Sélection et spécification d'objets avec des attributs littéraux

- La **sélection** d'objets dans une place est faite par les attributs de l'arc entrant
- La **spécification** des objets déposés dans une place est faite par les attributs de l'arc sortant

#### Exemple 1

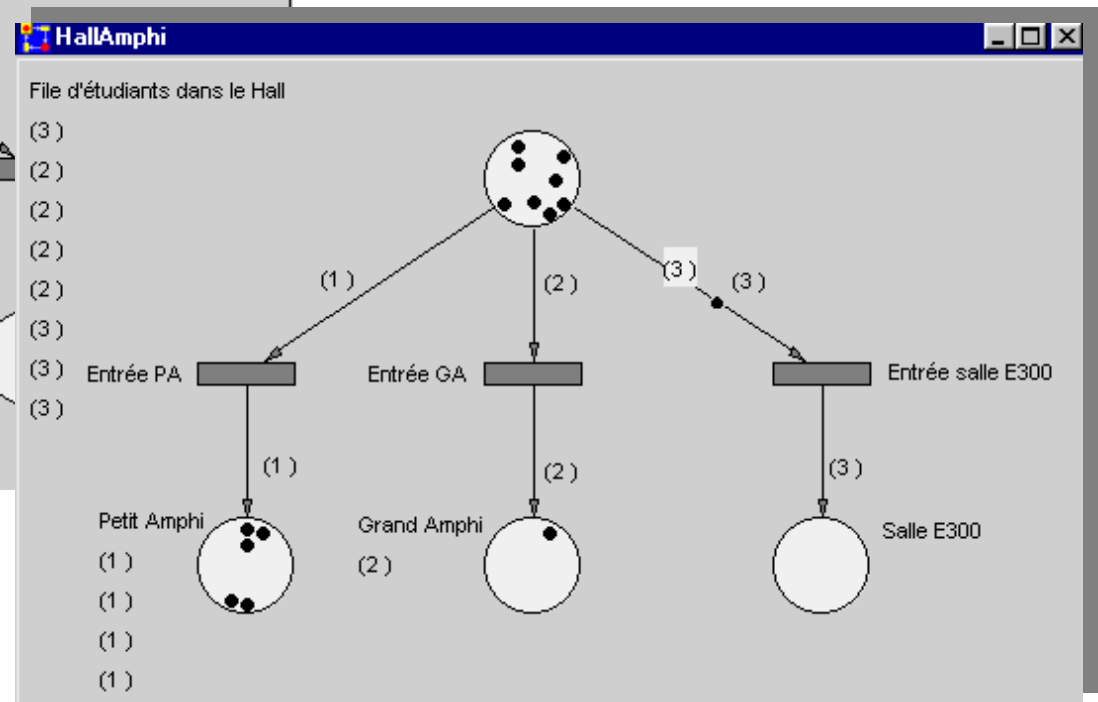
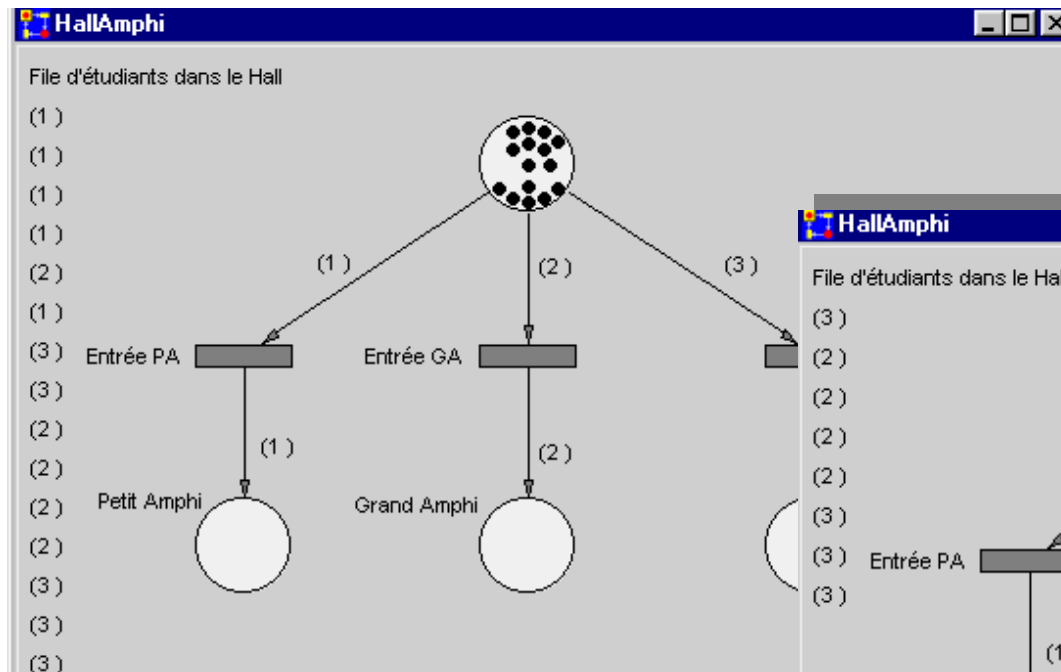
Extraction, à partir d'un lot de commandes, en attente ou prêtes, des commandes prêtes pour une facturation et une livraison





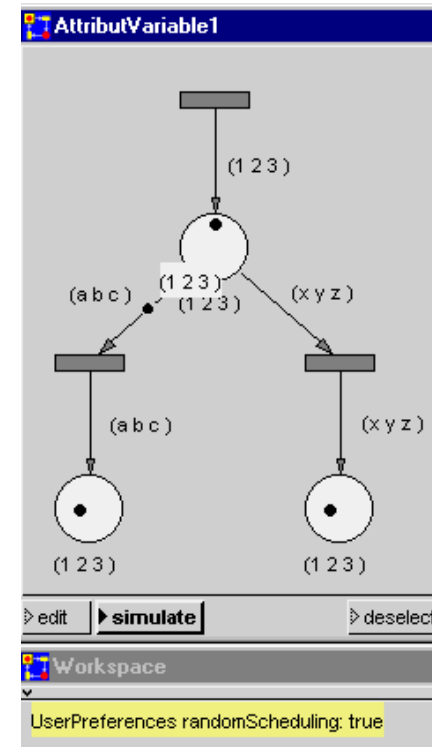
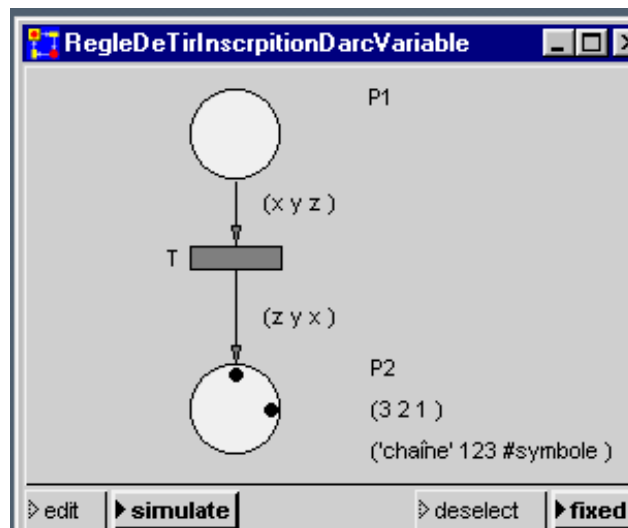
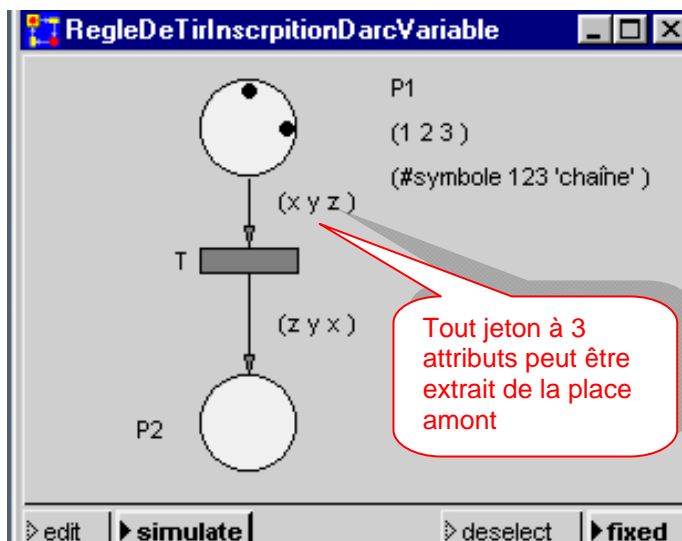
## Exemple 2

- Des étudiants attendent dans le hall de l'école
- Les trois arcs, dont les inscriptions fixent « l'année », sélectionnent et répartissent les étudiants dans les salles
- L'ordre d'entrée des étudiants dans les amphis et la salle E300 est déterminé par le simulateur (le choix est quelconque puisqu'il n'y a pas de délai sur les transitions) ; pour imposer un ordre il faudrait ranger les jetons étudiants dans des collections ordonnées (c.-à-d. des objets structurés Smalltalk)



## 2.2. Sélection et spécification d'objets avec des attributs variables

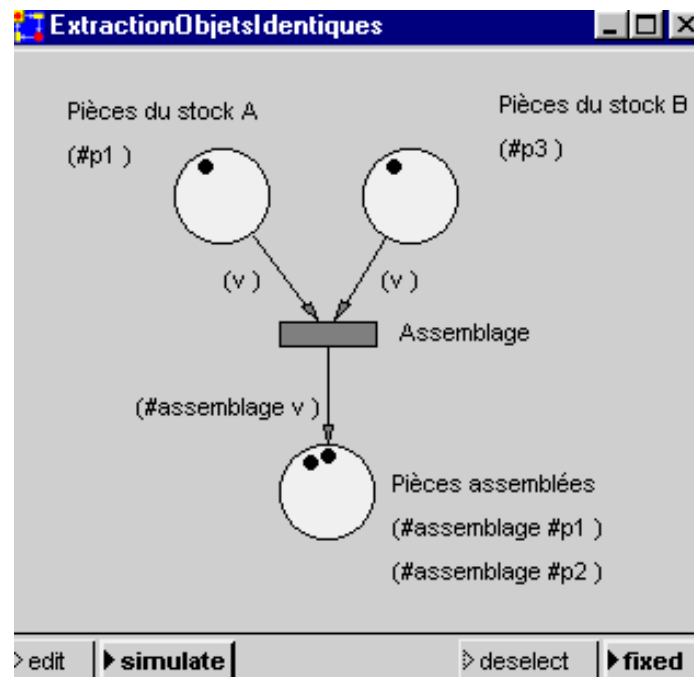
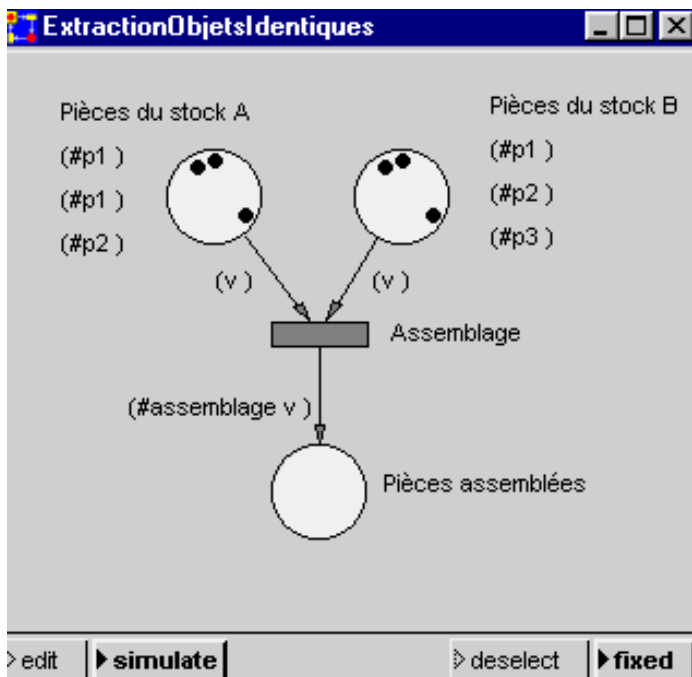
- Une variable est un pointeur d'objet ; elle peut donc référencer des objets de type quelconque
- Si un arc entrant dans une transition sans condition porte une inscription à plusieurs attributs variables, on pourra extraire tous les jetons de la place amont qui ont le même nombre d'attributs que cet arc, quelles que soient leurs valeurs



## 2.3. Sélection d'objets identiques en provenance de plusieurs sources

Si deux arcs d'entrée comportent la même variable d'entrée, seuls les jetons dont le contenu est identique peuvent circuler sur les arcs entrants (👉 ne pas confondre identité et égalité, voir § Objets Smalltalk)

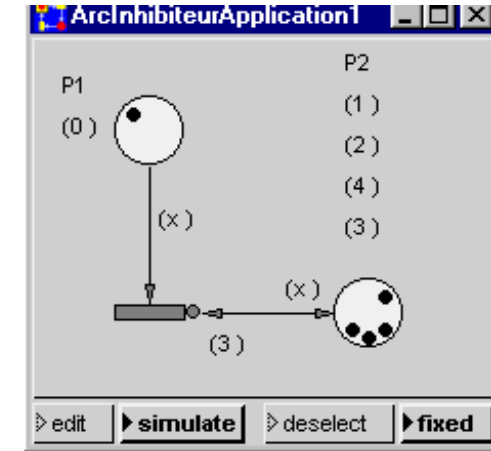
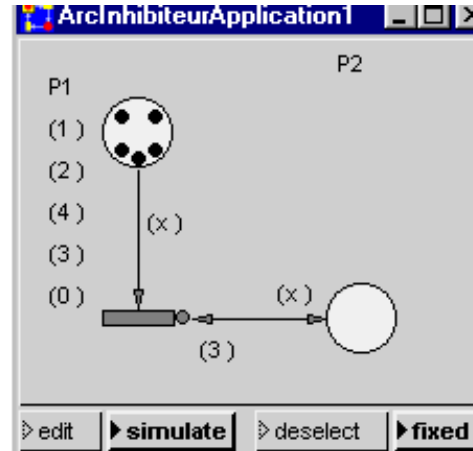
- Deux magasins contiennent des pièces de types différents
- Un assemblage est réalisé sur deux pièces de type identique mais en provenance des deux magasins



## 2.4. Blocage d'un traitement avec un arc inhibiteur à attribut

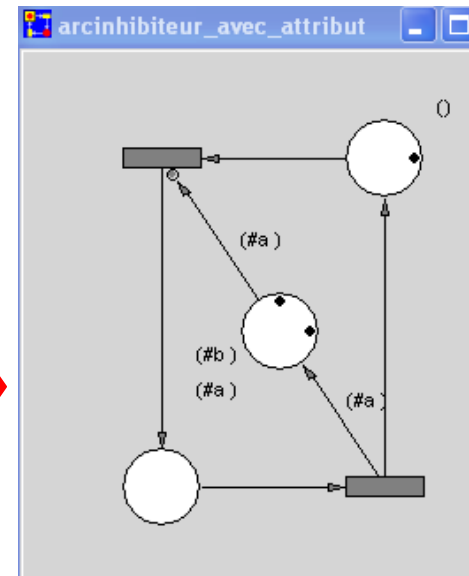
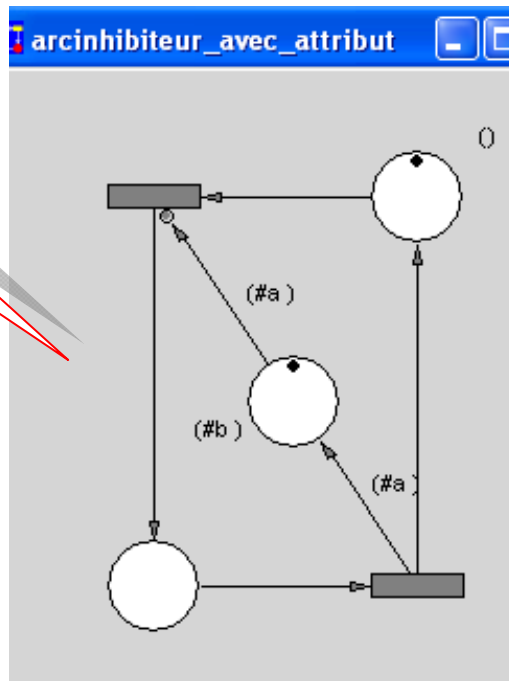
### Exemple 1

Le tir de la transition est impossible dès que (et tant que) la place P2 contient un jeton d'attribut 3



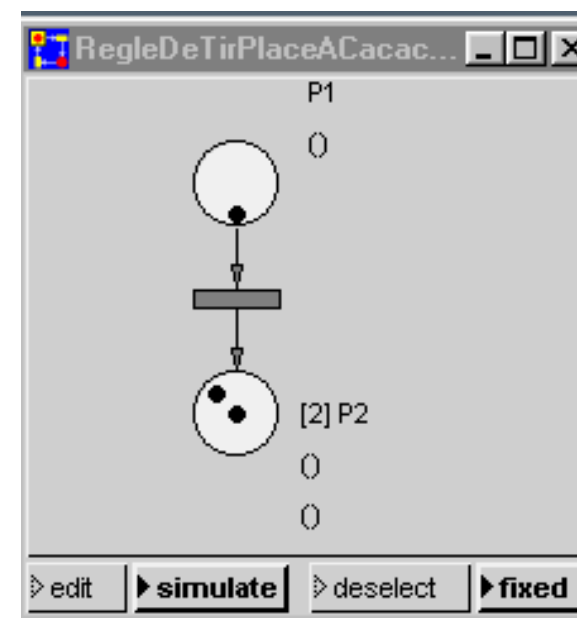
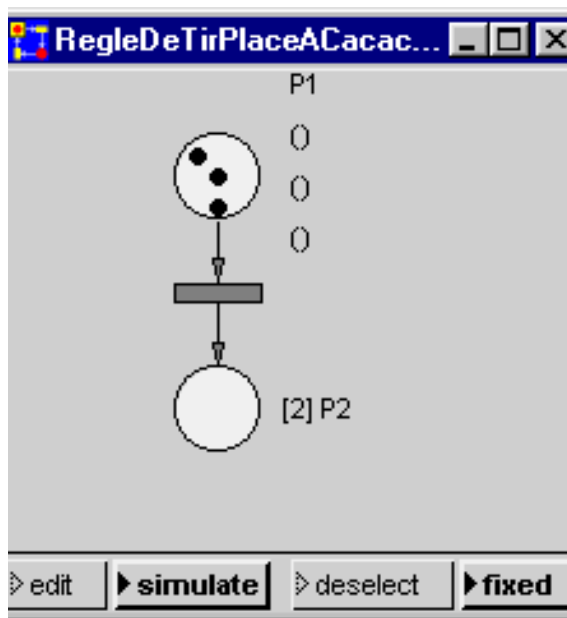
### Exemple 2

À l'instant  $t = 0$



## 2.5. Blocage d'un traitement avec une place sortante à capacité

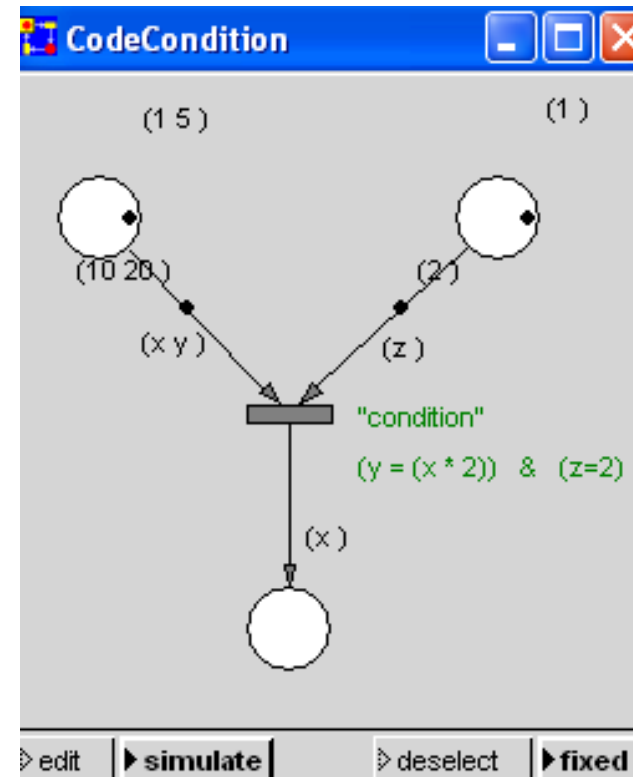
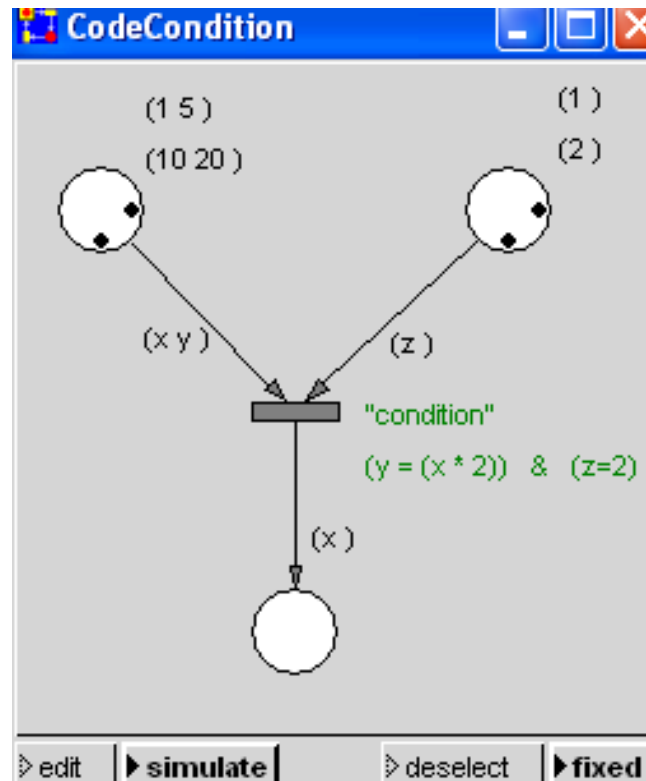
Dès que (et tant que) la place P2, dont la capacité est fixée à 2, contient 2 jetons, le tir de la transition est impossible.



## 2.6. Sélection d'objets par un code condition

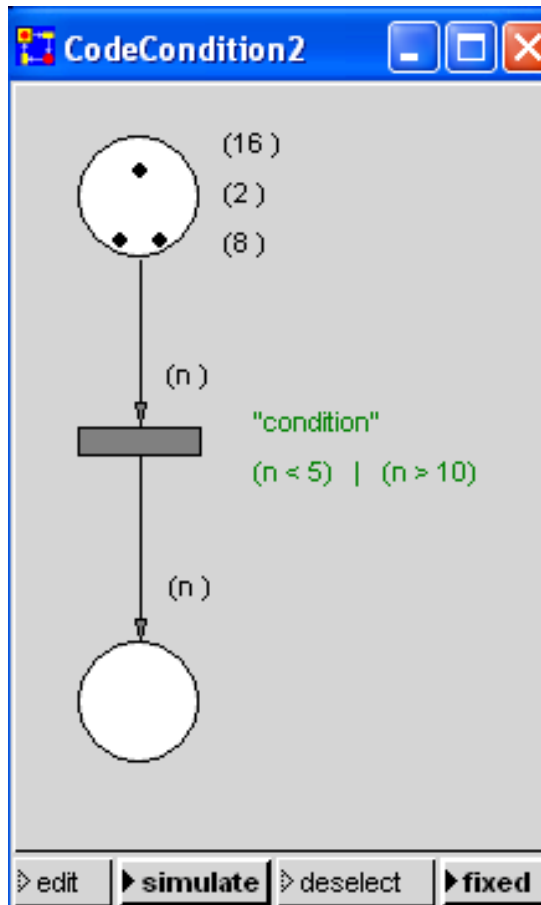
- La sélection est obtenue grâce à un code condition qui teste les valeurs des attributs des jetons via des variables temporaires portant les mêmes noms que les variables d'arc

### Exemple 1

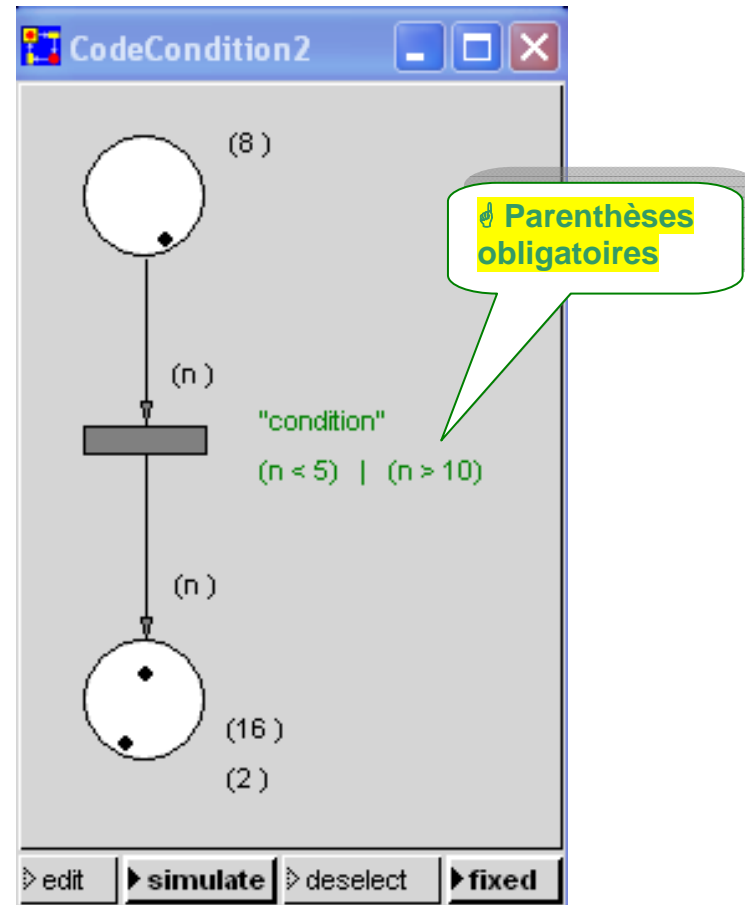


## Exemple 2 : sélection de valeurs appartenant à un intervalle

Initialisation

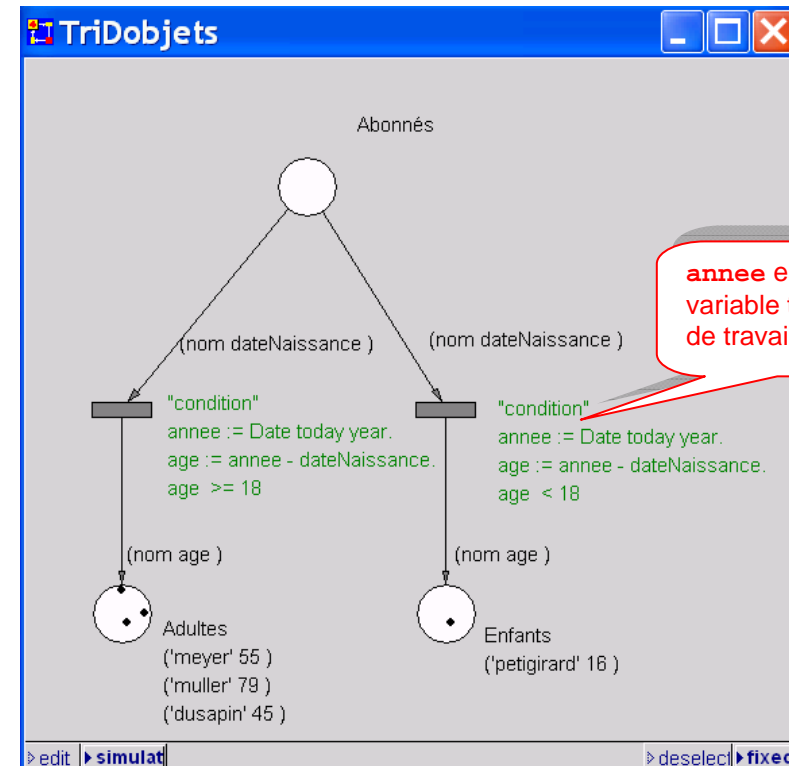
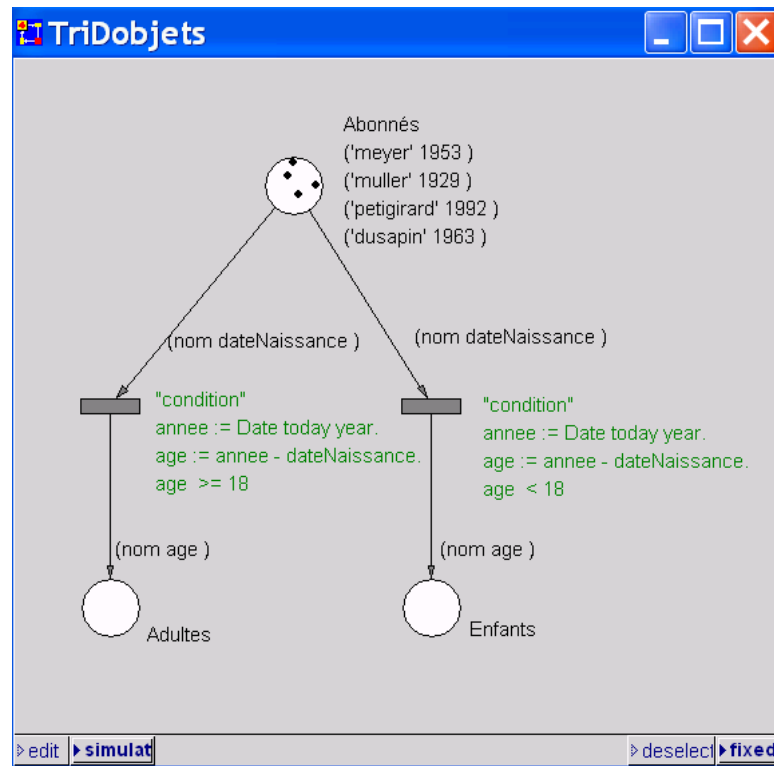


Fin de simulation



### Exemple 3 : tri d'abonnés en fonction de l'âge

- Le code condition peut comporter des variables temporaires ne servant que pour des évaluations intermédiaires





## 3. Traitement des objets structurés

### 3.1. Petit aperçu des objets Smalltalk

#### □ Objets simples

- objets à **valeur constante** (non modifiable), à existence unique
- désignation par des **littéraux**
- nombres (124, 8r432, 16r7F), symboles (\$A, #bulletin)

#### □ Objets structurés

- chaînes de caractères, tableaux, collections, ...
- désignation par des littéraux : chaînes de caractères 'abcd' et tableaux initialisés #(10 'nom' 5.2 (#a #b))
- modifiables : tableaux et collections
- copie possible (pour créer de **nouveaux objets** de même valeur)

## 3.2. Les tests d'égalité et d'identité

**v1 et v2 pointent l'entier 10 (objet non structuré)**

```
| v1 v2 |
v1 := 10.
v2 := 10.
v1 = v2    (test d'égalité, c.-à-d. même valeur)
true
```

```
| v1 v2 |
v1 := 10.
v2 := 10.
v1 == v2   (test d'identité, c.-à-d. même objet)
true
```

→ v1 et v2 pointent le même objet (l'objet de valeur 10 est unique)

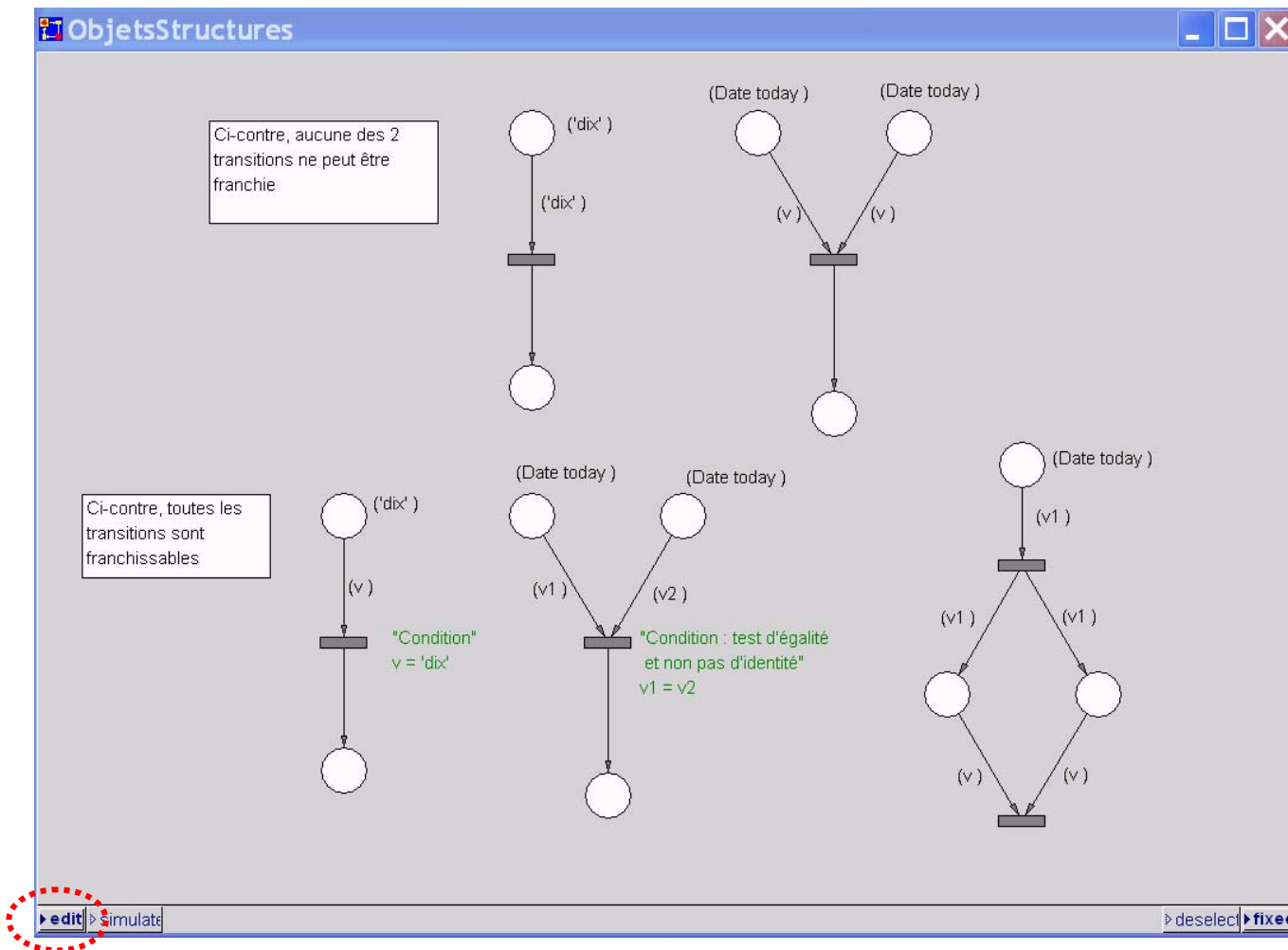
**v1 et v2 pointent la chaîne de caractères 'dix' (objet structuré)**

```
| v1 v2 |
v1 := 'dix'.
v2 := 'dix'.
v1 = v2   (égalité ?)
true
```

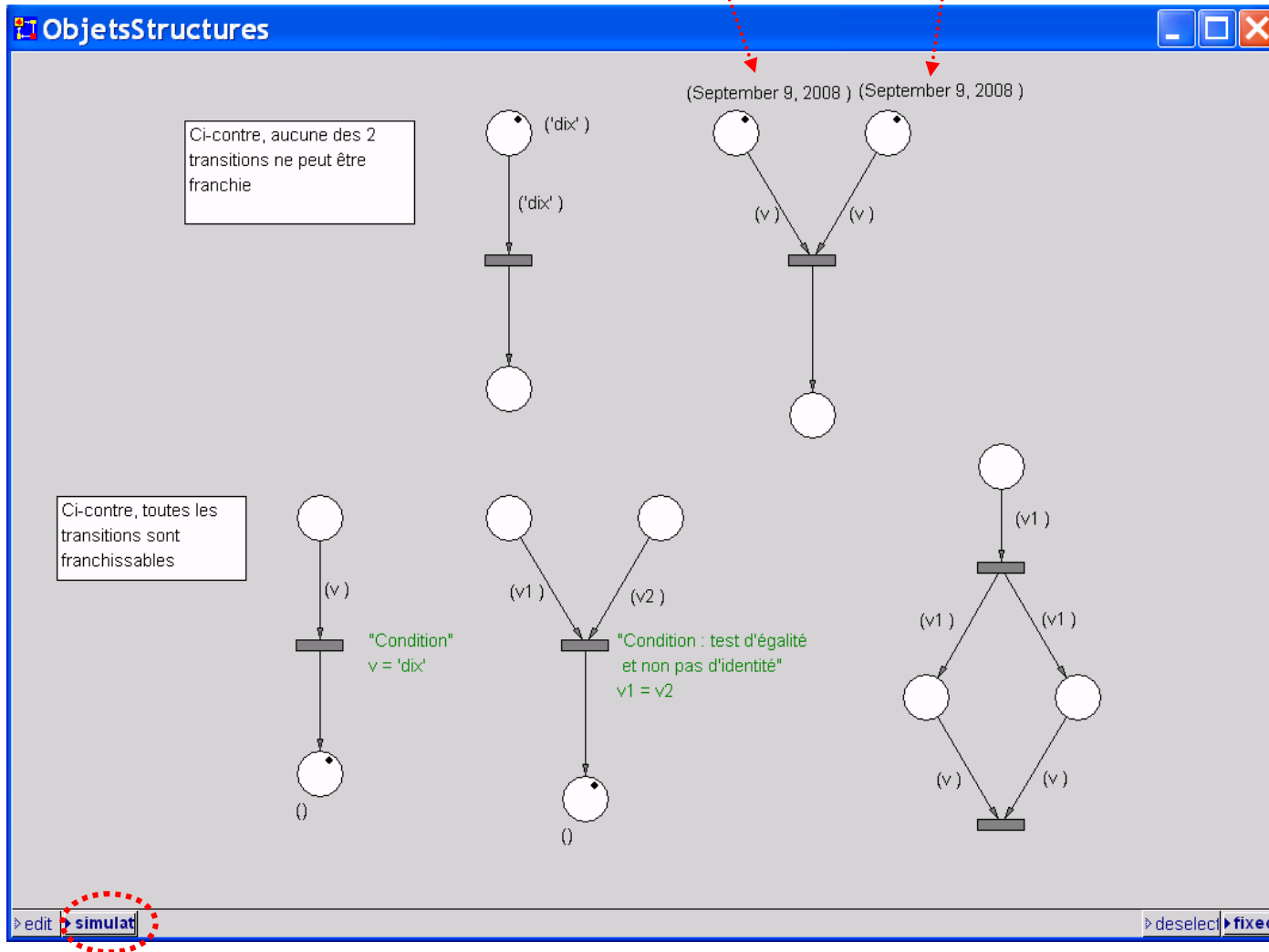
```
| v1 v2 |
v1 := 'dix'.
v2 := 'dix'.
v1 == v2  (identité ?)
false
```

→ v1 et v2 pointent deux objets distincts qui ont la même valeur

### 3.3. Sélection d'objets structurés

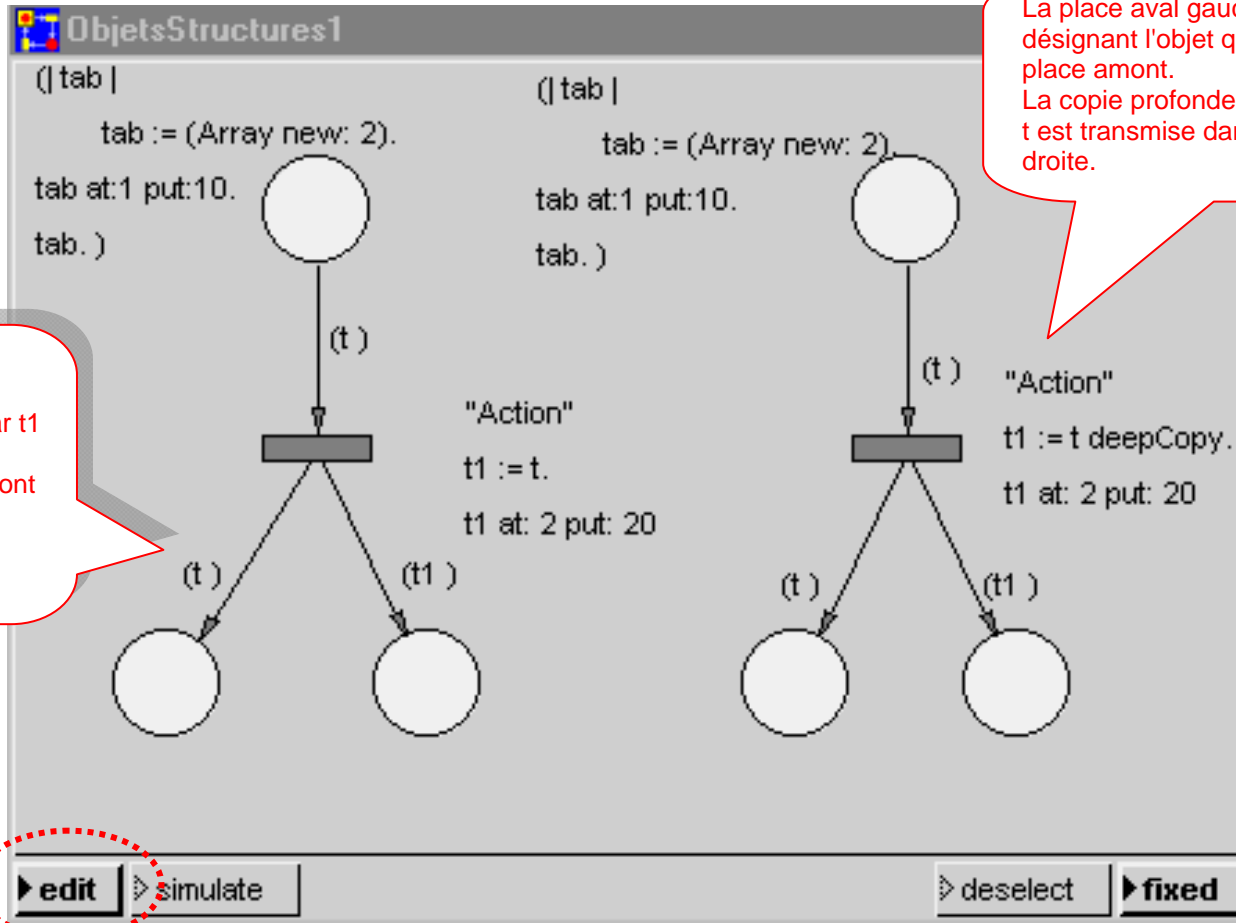


Cette date est représentée par 2 objets égaux en valeur mais non identiques



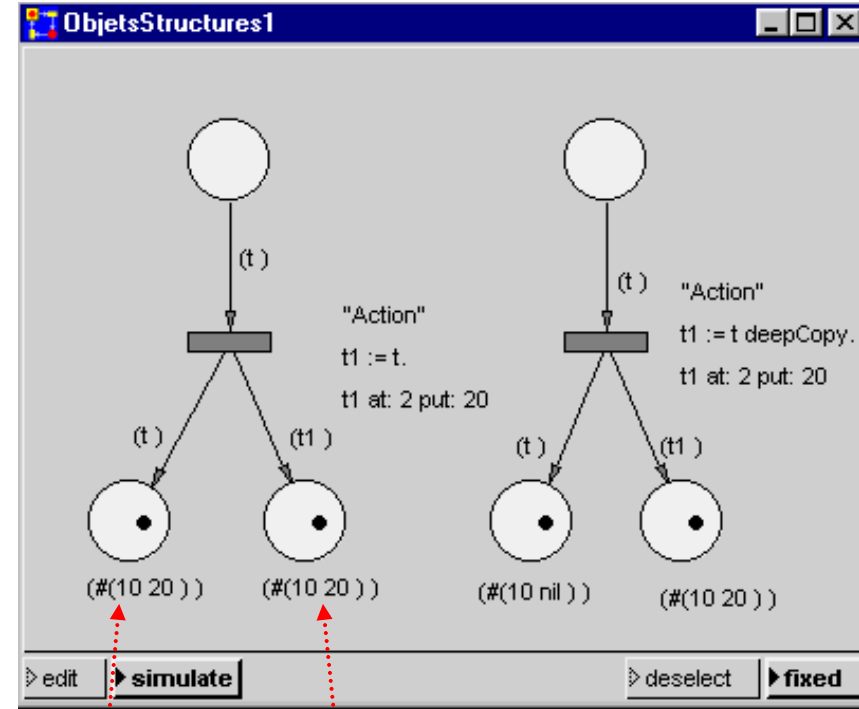
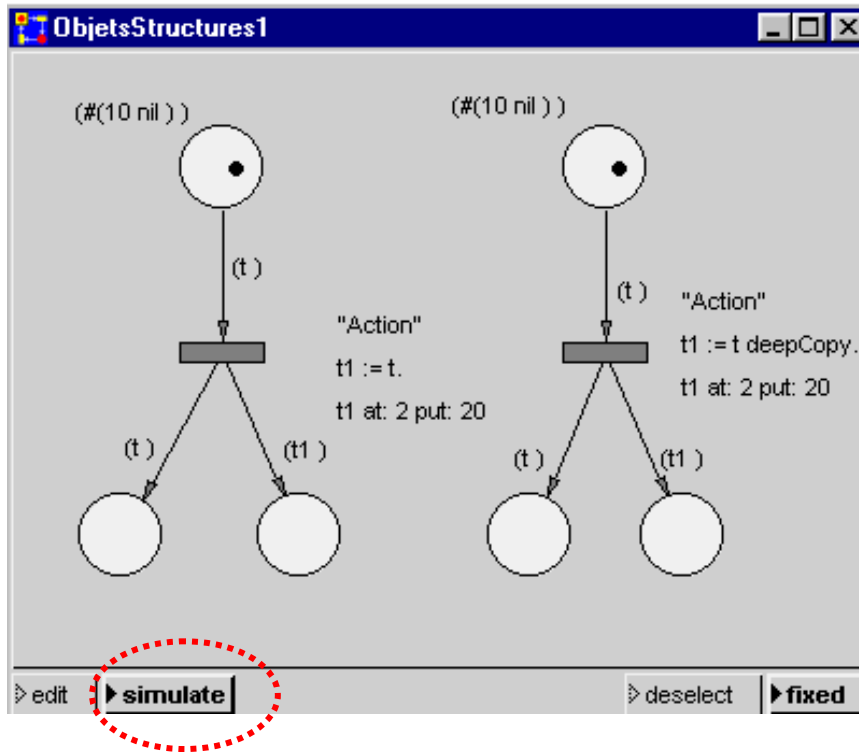
### 3.4. Modification d'un objet structuré

#### Exemple 1



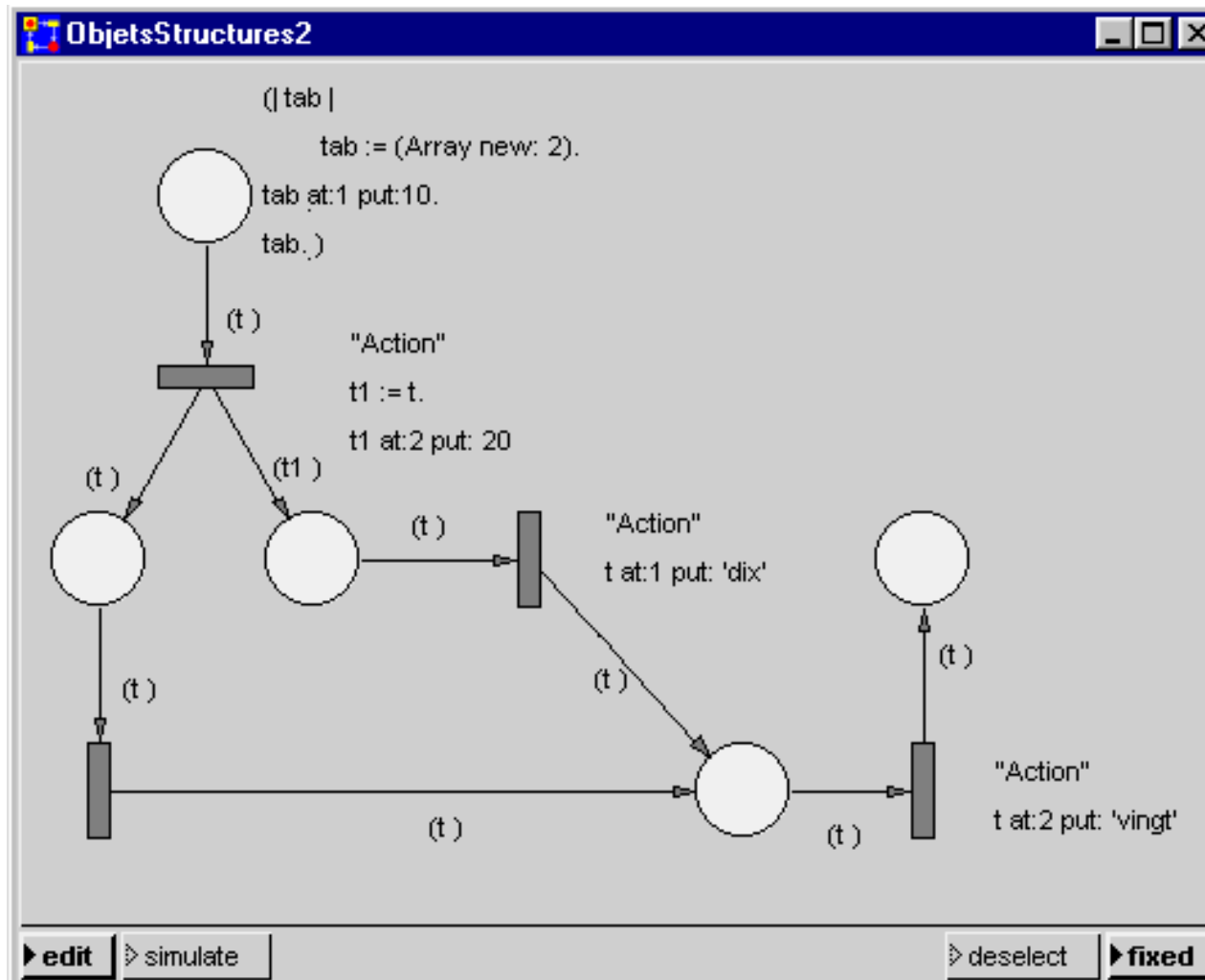
La place aval gauche reçoit un jeton désignant l'objet qui se trouvait dans la place amont.  
La copie profonde de l'objet désigné par t est transmise dans la place aval droite.

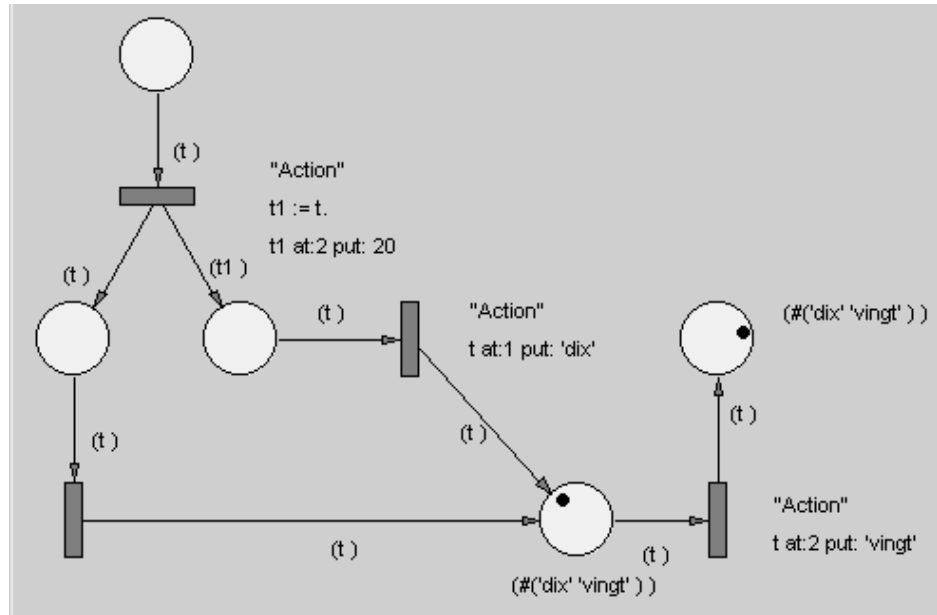
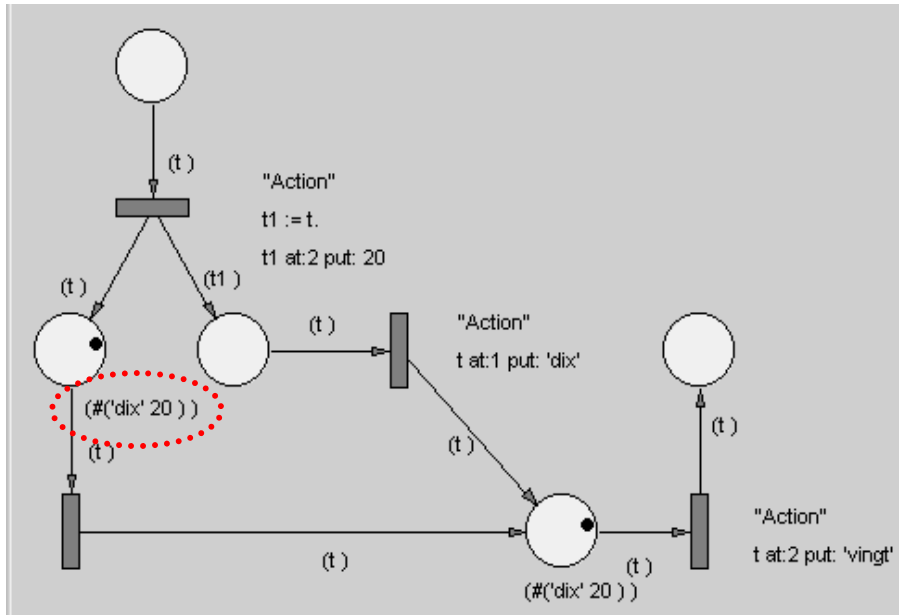
t et t1 pointent le même objet.  
Comme l'objet pointé par t1 est modifié dans la transition, l'arc sortant dont l'inscription est (t) transporte aussi l'objet modifié.



Ces 2 jetons se réfèrent au même objet #(10 20)

□ Exemple 2





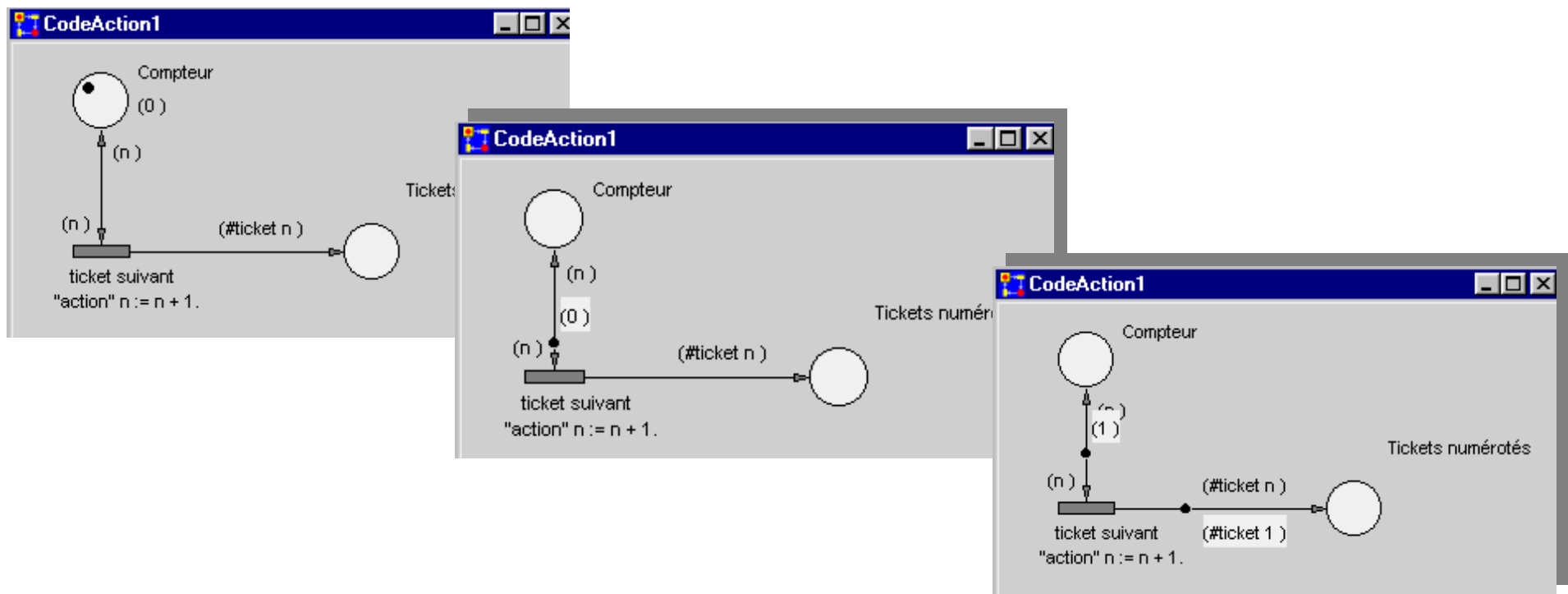


## 4. Opérations courantes sur les objets

### 4.1. Comptage d'opérations

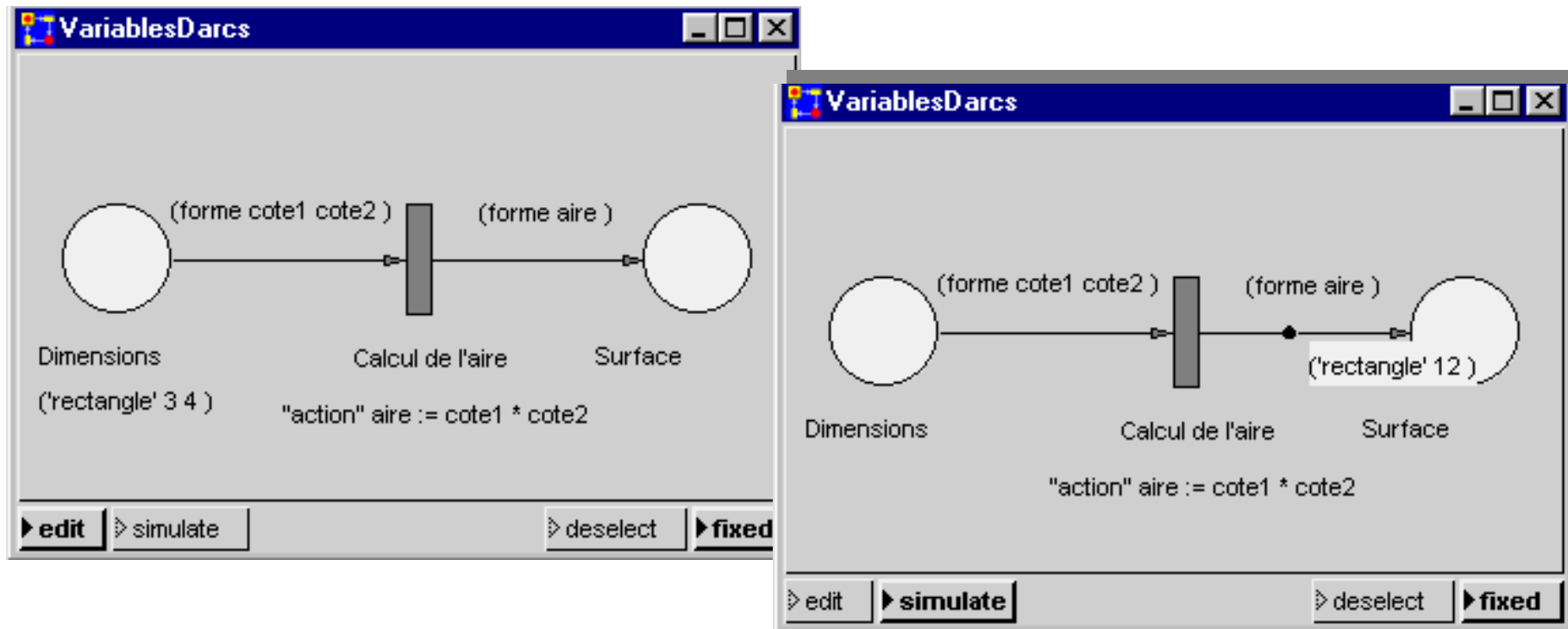
La place `Compteur` est initialisée avec un seul jeton à un seul attribut : la constante 1

- L'incrémement est décrite dans le code action de la transition `ticket` suivant
- À chaque franchissement de cette transition, la valeur courante du compteur est incrémentée et transmise dans la place `Tickets` numérotés
- L'unique jeton de la place `Compteur` est remis à jour après chaque incrémement



## 4.2. Attributs d'arc sortant fonctions d'attributs d'un arc entrant

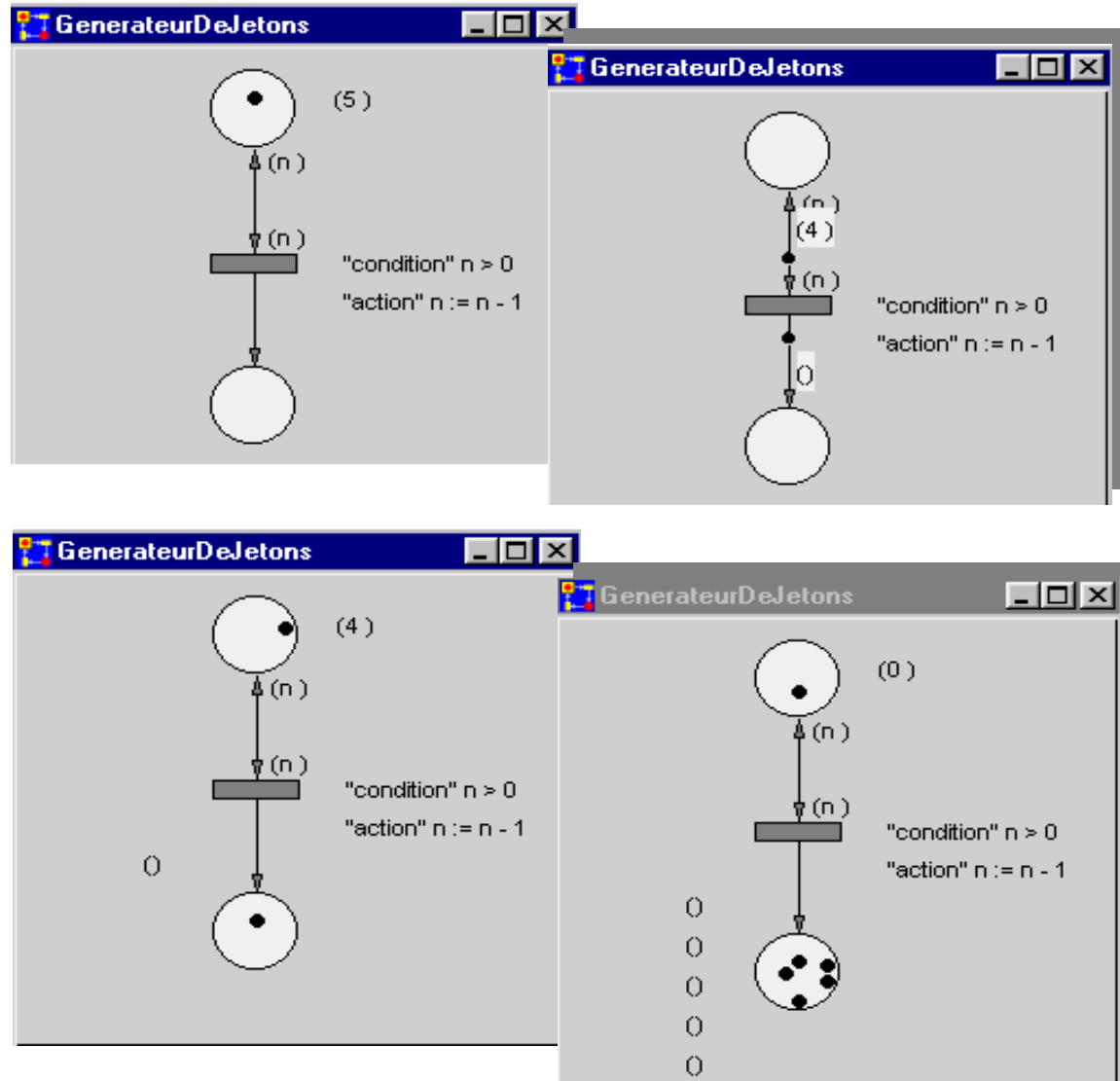
Ici les attributs des arcs sortant sont calculés dans un code action portant sur des variables d'arcs entrants.



## 4.3. Génération d'un nombre fini de jetons

### □ Solution 1 : une place compteur

- La place compteur est initialisée avec un seul jeton à un seul attribut : la constante 5
- La transition est franchie 5 fois; à chaque franchissement le compteur est décrémenté par le code action
- Le code condition fournit la valeur `false` lorsque 5 jetons sont générés, ce qui empêche tout franchissement supplémentaire de transition



❑ Solution 2 : méthode spécifique Smalltalk associée à une transition →

❑ Solution 3 : méthode spécifique Smalltalk écrite dans le code d'initialisation ↓

