

Automates et modélisation des SI

Modélisation

Modélisation avec des automates

State charts

Modéliser

Créer une représentation abstraite (simplifiée)

Exemple: le plan d'une ville

rues, noms des rues, monuments, parcs

pas les feux rouges, ni les trottoirs, ni les poubelles, ...

Exemple: en physique

Force = Masse x Accélération

simplification qui ne tient pas compte du frottement

Pourquoi modéliser ?

- Pour mieux comprendre
- Pour prendre des décisions
- Pour faire des simulations
- Pour construire de nouveaux objets (applications, bases de données, ...) qui améliorent le fonctionnement du système.

Le modèle dépend des objectifs, du point de vue, etc.

Modéliser avec les automates

Pour représenter les aspects dynamiques d'un système

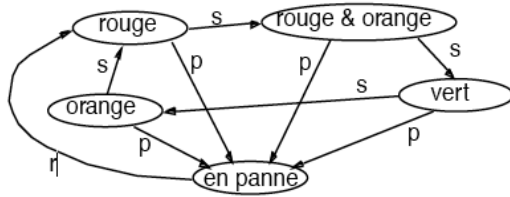
- Evolution de l'état du système
- Evolution de l'état de sous-systèmes pris individuellement (objets)

Construction de l'automate

Etats de l'automate \longleftrightarrow Etats du système (ou de l'objet)

Alphabet (symboles) \longleftrightarrow Evènements qui agissent sur le système.

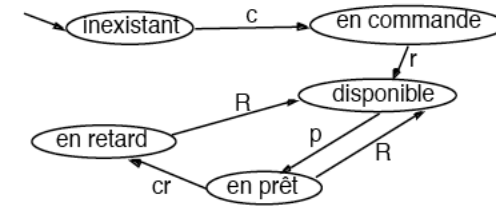
Exemple - feu tricolore



Alphabet : { s(ignal de contrôle), p(anne), r(éparation) }

Exemple - livre

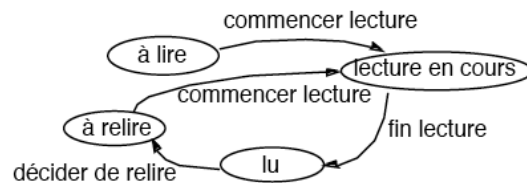
Point de vue de la bibliothèque



alphabet: {c(ommander), r(ecevoir), p(rêter), R(endre), cr(onstater retard)}

Exemple - livre

Point de vue du lecteur



Automates et conception

Faire un modèle du comportement du système à développer.

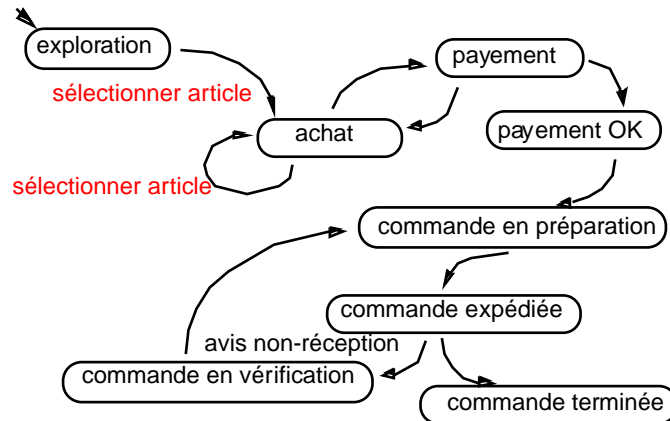
Basé sur l'analyse du s.i. / du domaine (lien fort).

Produire une spécification du comportement sous forme d'un ou plusieurs automates.

Utiliser la spécification pour développer et vérifier le système

!! Ne sert pas à spécifier les calculs et autres transformations de données !!

Exemple: achat sur un site de vente par internet



Utilisation de cette spécification :

- la mémorisation d'une commande doit comprendre la mémorisation de son état
- les programmes de gestion des commandes doivent s'assurer que la spécification est respectée
- l'interface du système doit aider les utilisateurs (clients et marchands) à respecter la spécification (montrer les actions possibles et impossibles)

State Charts

Problèmes posés par la modélisation avec un AF

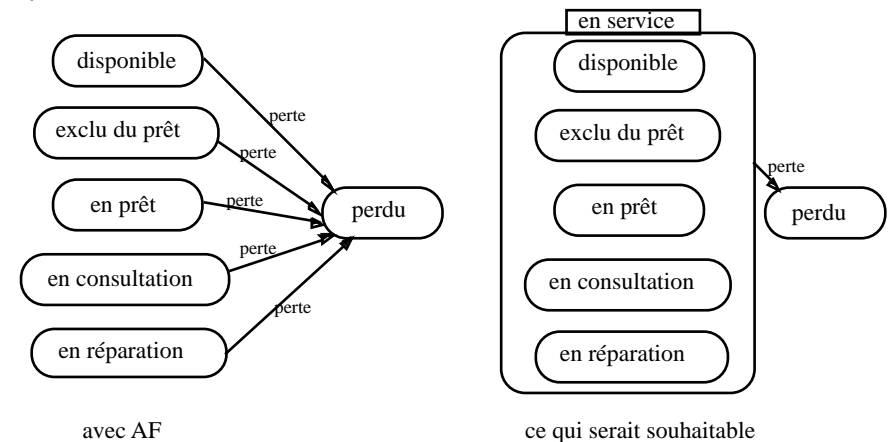
- Diagrammes «plats», pas de hiérarchie, de modularité
- Grand nombre de transitions correspondant à un même événement
- Explosion du nombre d'états dès que le système croît
- Impose la séquentialité alors qu'il peut y avoir parallélisme

Définition et utilisation des state charts

Diagrammes "plats"

Tous les états sont au même niveau, pas de sous-états

- > il en découle un grand nombre de transitions pour un même événement
- > problèmes de lisibilité



Explosion du nombre d'états

Exemple : modéliser le cycle de vie d'un livre dans une bibliothèque

Deux sortes d'état orthogonaux (indépendants):

1) Disponibilité : disponible - en prêt - exclu du prêt

2) Lisibilité : neuf - bon état - usagé - hors d'usage

Il y a 12 états = 12 combinaisons possibles

(disponible & neuf), (disponible & bon état), ..., (en prêt & neuf), (en prêt & bon état), ...

Si on ajoute un nouvel état de disponibilité en réparation, il faut ajouter 4 états au système : (en réparation & neuf) , ..., (en réparation & hors d'usage)

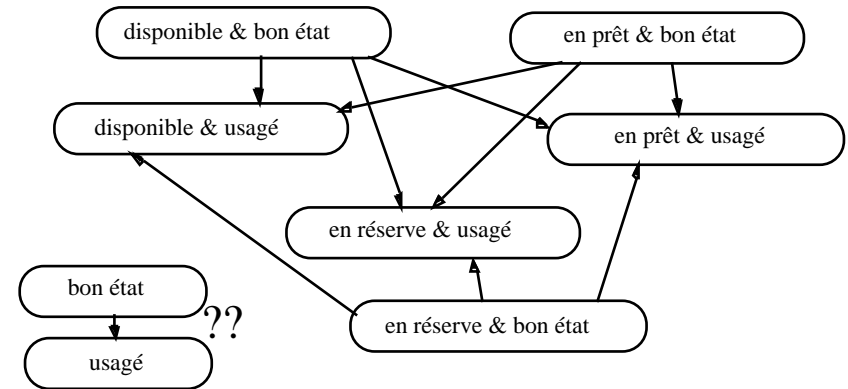
Si l'on veut ajouter le fait qu'un livre peut être réservé ou non (troisième sorte d'états), on devra doubler le nombre d'états. (réservé & en prêt & neuf) , ..., (non réservé & exclu du prêt & usagé)

Sequentialité

• Impose la séquentialité alors qu'il peut y avoir parallélisme

Possible, mais extrêmement difficile de voir où réside le parallélisme

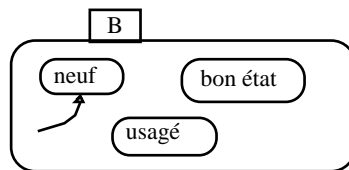
Exemple



State charts

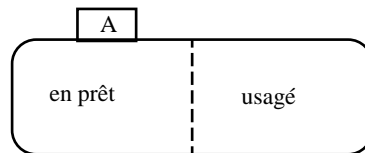
Etats atomiques

Etats composés OU (exclusif)



La flèche indique le sous-état d'entrée

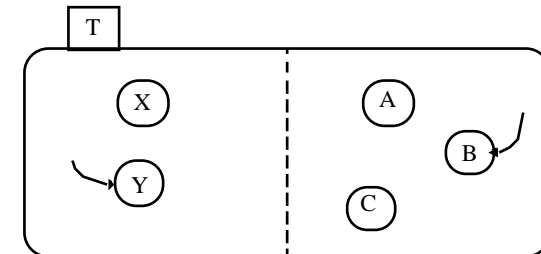
Etat composés ET



Etre dans l'état A signifie être à la fois (en prêt) et (usagé).

Etat ET composé de OU

Un état ET représente le produit cartésien de ses sous-états

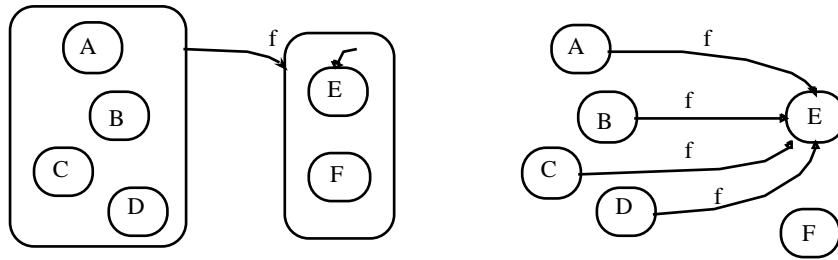


Représente $\{X, Y\} \times \{A, B, C\} = \{ (X, A), (X, B), (X, C), \dots, (Y, B), (Y, C) \}$

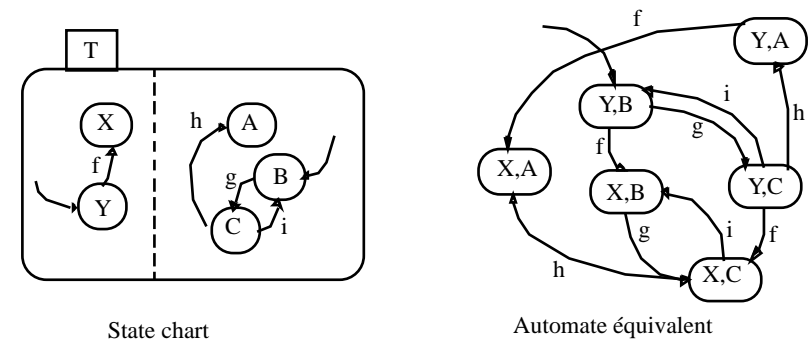
L'état d'entrée est (Y, B)

Transitions

Une transition d'un état composé à un autre représente une transition depuis chacun des sous-états.



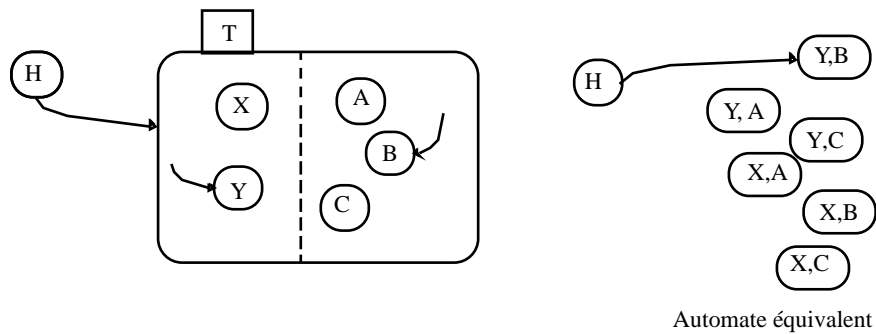
Transition à l'intérieur des composantes OU d'un ET



Entrée dans un état composé

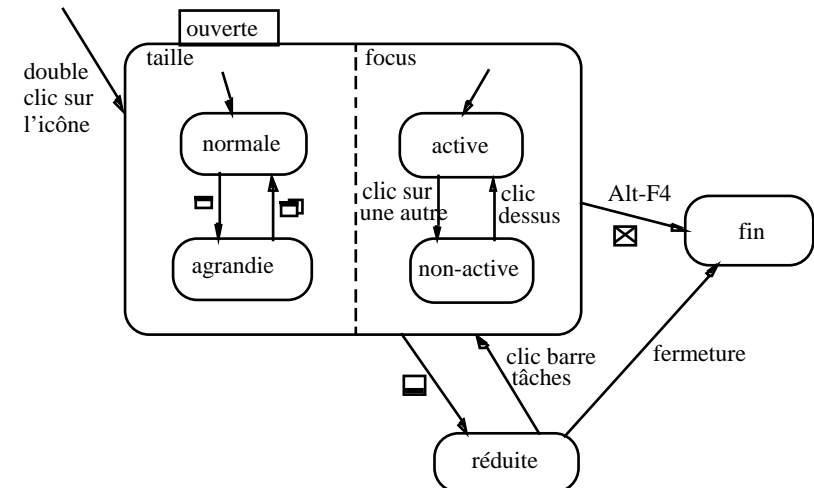
Un transition vers un état composé arrive sur
l'état d'entrée (composition OU)

l'état d'entrée de chaque composante (composition ET)



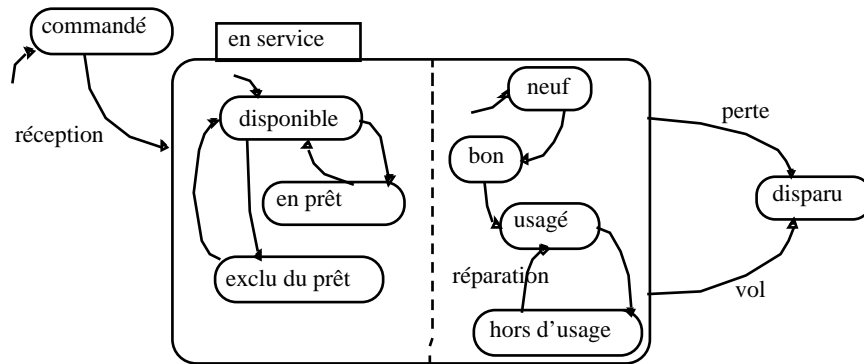
Exemple

Fenêtre dans Windows

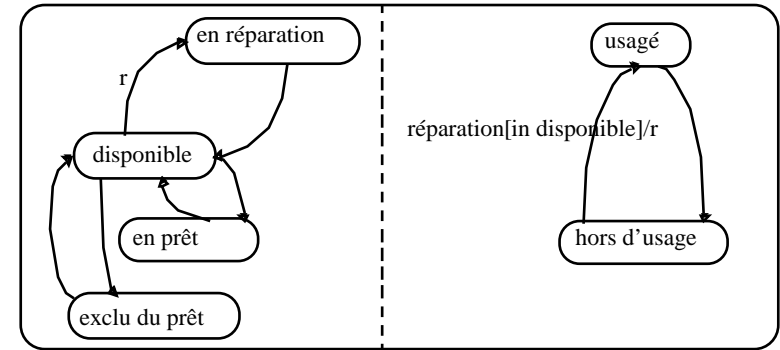


Exemple

Cycle de vie d'un exemplaire d'ouvrage dans une bibliothèque :



Interaction entre états orthogonaux

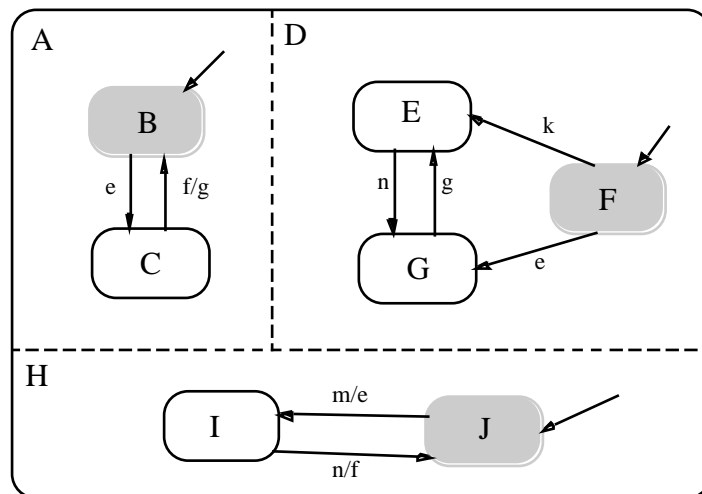


La transition réparation ne peut se faire que si le livre est disponible

Elle déclenche la transition r

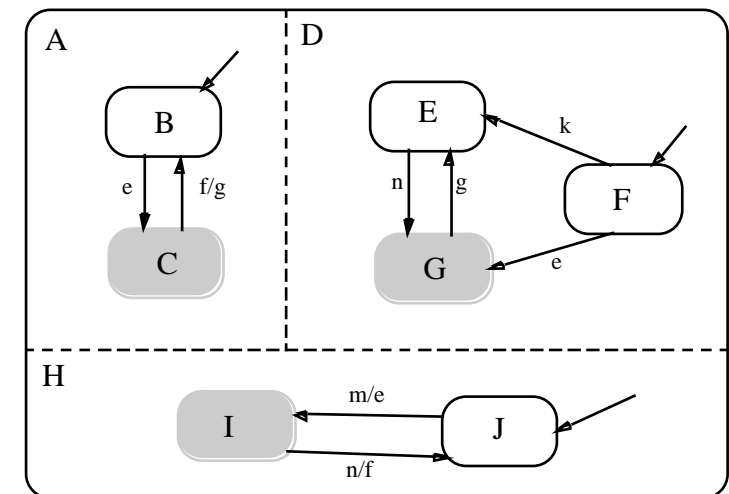
En terme d'automate: {disponible, hors d'usage}—réparation—> {en réparation, usagé}

Réaction en chaîne



Réaction en chaîne

événement m



Réaction en chaîne

événement n

