

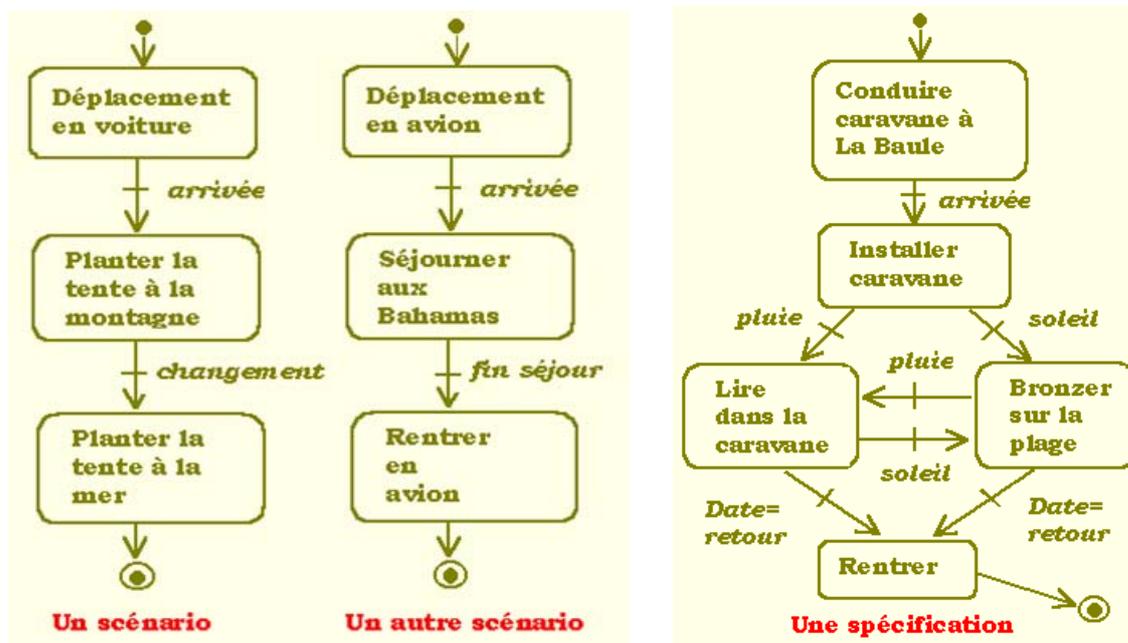
Rappel de cours

Contenu d'un diagramme de comportement

Description des séquences possibles d'états et d'actions par lesquelles un élément peut passer tout au long de sa vie. De telles séquences résultent de sa réaction à des événements discrets tels que des signaux ou des invocations d'opérations.

Deux contextes d'emploi d'un diagramme dynamique

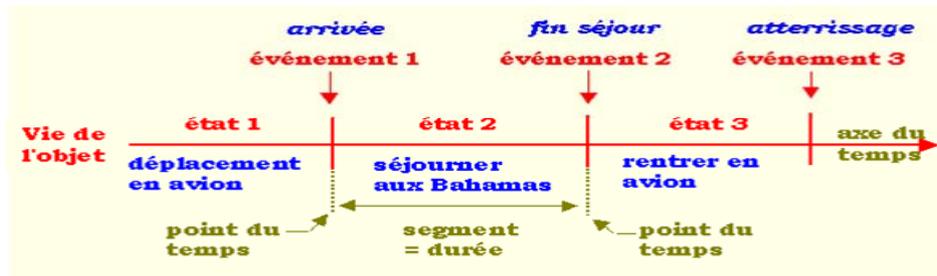
Il peut servir à décrire le comportement concret de la vie d'un objet parmi tous ceux qui sont possibles. On dira alors qu'il s'agit d'un scénario. Il peut servir à fixer le comportement attendu d'un objet tout au long de sa vie, c'est-à-dire spécifier un ensemble de contraintes auxquelles il doit nécessairement se conformer. Ceci revient à limiter délibérément l'explosion combinatoire des scénarios possibles.



Définitions

Un état est une condition dans la vie d'un objet (ou d'une interaction) pendant laquelle il satisfait quelques conditions, exécute quelques actions ou attend quelques événements. Un événement est une occurrence remarquable qui a la capacité de déclencher une transition d'un état vers un autre état.

On constate que état et événement s'entre-définissent mutuellement : un état est atteint depuis un événement (transition), un événement est ce qui déclenche un changement d'état. Mais ce qui permet de les distinguer est un critère de temporalité : un événement est instantané (un point sur une ligne de vie) alors qu'un état possède une durée (un segment sur la même ligne de vie) :



Etats

En relation à son objet, un état peut être considéré de 3 manières différentes :

- **vue externe** : un état est une **réponse** de l'objet qui est sollicité par un événement. Par exemple, quand survient l'événement *fin de séjour*, l'objet *vacances d'été* répond en entrant dans l'état *retour en avion*.
- **vue interne** : un état est un **intervalle** entre 2 événements. Par exemple, l'état *rentrer en avion* est une caractéristique interne à l'objet *vacances d'été* imposée par son extérieur par les événements *fin de séjour* et *atterrissage*.
- **vue normative** : un état est une **condition** portant sur les valeurs liées d'un groupe d'attributs de l'objet. Par exemple, état *rentrer en avion* = attribut *distance* de valeur > 800 km ET attribut *budget* de valeur > 2000 F. Autre exemple : l'état *Liquide* de l'élément Eau peut être défini par la double condition sur son attribut *température* : $0 < T < 100^{\circ}\text{C}$.

Evénements

UML inventorie 5 types d'événements (non nécessairement exclusifs) :

- le **changement de valeur** d'une condition booléenne : une expression passant de la valeur *Faux* à la valeur *Vrai*, quel que soit le moment où elle survient peut constituer un événement. Exemple : *Temps = pluie* dont la valeur est à *Faux*, en passant à la valeur *Vrai* constitue un événement à la suite duquel je rentre dans l'état *mouillé*. Notation par mot clé : **when (expression)**.
- la **réception d'un signal** explicite envoyé par un autre objet. Exemples : *la réception d'un SOS par le bateau de sauvetage*, ou *un message d'erreur*.
- une **demande d'opération** faite par un autre objet. Exemple : *Passe moi le sel*.
- l'épuisement d'un **délai temporel**. Exemple : *la fin du sursis à exécution* est un événement pour le condamné à mort. Notation par mot clé : **after (expression)**.
- la **survenance d'une date** peut être un événement (on l'appellera Timer) : Exemple *le 31/12/1999, minuit* est un événement. Notation par mot clé : **when (expression)**.

Les événements de type 2 et 3 se notent : **nom_d'événement (liste de paramètres)**.

- assimilables à des messages, ils peuvent donc être porteurs d'**attributs**. Exemple : *départ_de_vol (n°vol: Integer, destination: Ville)*.
- ils ont une **portée** qui est la visibilité du package dans lequel se trouve l'objet receveur de l'événement. Un événement n'est pas local à une classe. En cela, il constitue le moyen par lequel des diagrammes de comportement relatifs à des objets différents peuvent être reliés.

Gardes sur les transitions

- **Condition de garde**
Elle est notée entre [], c'est une expression booléenne qui fonctionne comme paramètre de l'événement déclencheur de la transition entre 2 états. Pour que la transition soit franchie il faut :
a) que l'événement survienne
b) que la condition possède la valeur "*Vrai*".
- **Action**
Il s'agit d'une opération **instantanée** associée à un événement, qui provoque le changement d'état de l'objet. Elle est notée par un slash qui précède le nom de l'action : /

Notation complète pour les transitions

nom d'événement (liste de paramètres) [garde] / action

Exercice 1

Un dispositif de contrôle d'accès par carte magnétique à un photocopieur est équipé d'un écran de visualisation qui peut afficher les messages suivants :

- "INSEREZ VOTRE CARTE" lorsque le dispositif est inutilisé.
- "PATIENTER" pendant que le dispositif lit le code d'une carte introduite.
- "CARTE INVALIDE" lorsque le code n'est pas reconnu (illisible) ; la carte est alors automatiquement éjectée.
- "COMPOSEZ VOTRE CODE" lorsque celui-ci a pu être lu.
- "CODE REFUSE" si le code composé n'est pas identique au code lu ; la carte est alors automatiquement éjectée.
- "UTILISATION EN COURS" lorsque le code composé est correct.

L'utilisateur peut à tout moment actionner un bouton qui provoque l'éjection de la carte. Après toute éjection de carte, le dispositif affiche "INSERER CARTE".

Proposer le graphe états-transitions du lecteur de carte.

Exercice 2

Partie 1

Une partie d'échecs peut être décrite comme suit :

- Au départ, le tour de jouer est au joueur possédant les pièces blanches,
- Quand le joueur aux pièces blanches joue, le tour de jouer devient à celui possédant les pièces noires. L'inverse est aussi vrai.
- Quand le joueur possédant les pièces blanches (respectivement, les pièces noires) fait un échec et mat, il gagne la partie,
- S'il y a un pat, la partie est finie par un nul.

Imaginons qu'il existe un contrôleur qui surveille la partie d'échecs. Décrivez, d'un point de vue contrôleur, le digramme Etats-Transitions de la partie d'échecs.

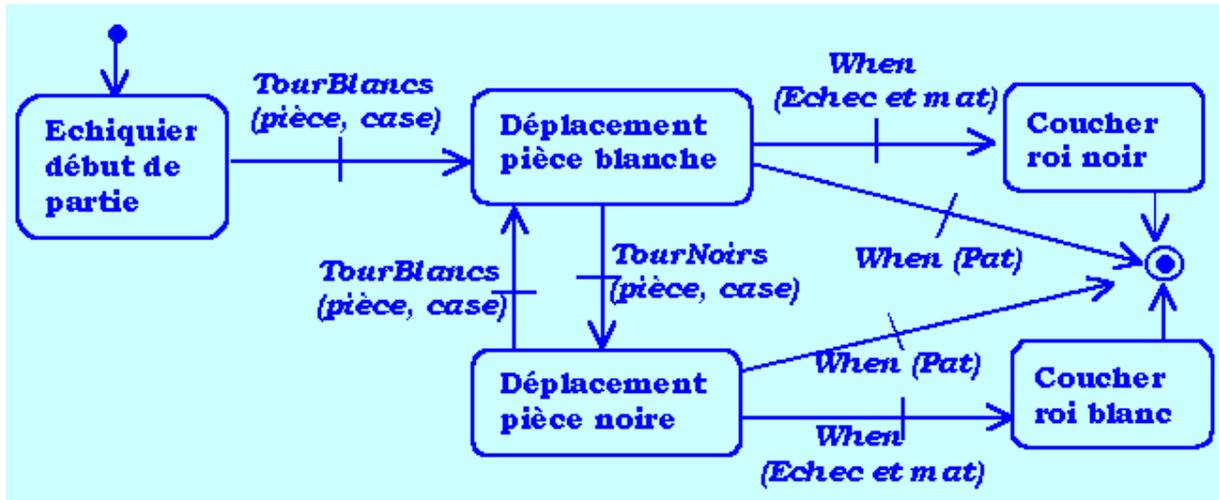
Partie 2

Supposons maintenant que l'on veuille décrire la même partie d'échecs du point de vue du comportement d'un objet **Afficheur** (interface graphique). Son rôle est d'afficher sur un écran l'échiquier et les pièces du jeu. Sur le jeu d'une pièce, il doit effacer celle-ci de sa position initiale, la réafficher dans sa nouvelle position et, pour faire joli, simuler graphiquement la trajectoire du déplacement.

- les événements qui conduisent à la fin de partie sont de type 1 : changement de valeur d'un booléen.

Le diagramme n'exprime pas le cas de l'abandon de partie par un joueur : ce serait un événement adressé par un objet représentatif de joueur au contrôleur, conduisant à un arrêt prématuré de la partie. Ceci nécessiterait donc une interaction joueur(s) - contrôleur.

Partie 2



Commentaires

Ce qui pour le contrôleur était un état (par exemple, *TourBlancs*) devient pour l'afficheur un événement. A l'inverse ce qui était pour le contrôleur un événement devient pour l'afficheur un état : par exemple *Déplacement pièce blanche*. Qu'il s'agisse bien d'un état pour Afficheur est d'ailleurs attesté par cette règle qui s'applique aux humains : le relâchement prématuré d'une pièce au cours de son déplacement vaut pièce jouée. Etonnant non ?

Evidemment, nos deux objets interagissent : **Contrôleur** est, pour **Afficheur**, la source de l'événement *TourBlancs(pièce, case)*. **Afficheur**, de son côté, doit faire un compte rendu à **Contrôleur** pour lui indiquer qu'il a terminé le déplacement d'une pièce blanche, un événement pour **Contrôleur** qui se retrouve de ce fait dans l'état *TourNoirs*. Il va alors demander à l'objet **Echiquier** si par hasard les booléens EchecMat, Pat n'ont pas changé de valeur et, à défaut, demander à l'objet **JoueurNoir** de bien vouloir donner son jeu.