

Diagramme d'Etats-Transitions

Objectif du diagramme d'Etats-Transitions (ou Statechart) (1)

- ✎ Décrire le comportement interne d'un objet (pris individuellement) à l'aide d'un automate à états finis : son cycle de vie

Peut également être utilisé pour décrire tout élément à états finis : cas d'utilisation, systèmes, sous-systèmes, méthodes, etc.
- ✎ Cet automate :
 - va décrire les **différentes séquences d'actions** effectuées par l'objet,
 - va décrire les **différentes séquences d'états** que traverse l'objet,
 - pendant le **cycle de vie** de l'objet,
 - en réaction à des **événements** divers : stimulus/signaux, invocations de méthodes.

1

Diagramme d'Etats-Transitions

Objectif du diagramme d'Etats-Transitions (ou Statechart) (2)

- ✎ Le Diagramme d'Etats-Transitions suppose que :
 - les différents états d'un objet sont clairement identifiables,
 - Que le comportement d'un objet est assez complexe,
 - Que l'objet est amené à réagir à beaucoup d'événements.
- ✎ Ce que le diagramme d'Etats-Transitions ne fait pas :
 - Décrire les interactions d'un objet avec son environnement. Il décrit le comportement d'un objet lorsqu'il communique avec son environnement.

2

Diagramme d'Etats-Transitions

Composants

- ✎ Trois composants :
 - Les états
 - Les transitions
 - Les événements
- ✎ Ces 3 composants sont indissociables pour décrire le comportement dynamique d'un objet.
 - Un objet est dans un état donné quelconque (état source)
 - Attend un événement
 - La réception d'un événement déclenche une transition
 - Le déclenchement d'une transition permet à l'objet de passer à un autre état (état destination)



L'objet : un élément **passif** contrôlé par les événements en provenance de son environnement

3

Diagramme d'Etats-Transitions

Etat (1)

- C'est la situation pendant laquelle un objet vérifie une certaine condition ou effectue une certaine activité : l'état est lié à la notion de durée.
- Se traduit par l'ensemble des valeurs des attributs de l'objet.
- Représentation graphique : un rectangle aux coins arrondis

unEtat

- **Exemple 1** : L'état d'une objet à l'arrêt

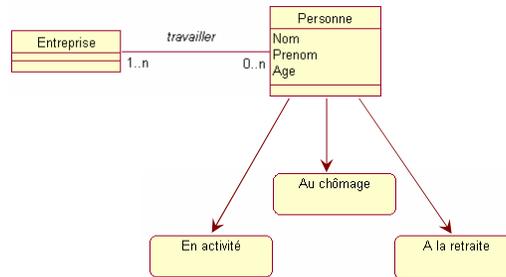
A l'arrêt

4

Diagramme d'Etats-Transitions

Etat (2)

- **Exemple 2** : soit le diagramme de classe suivant modélisant les personnes qui travaillent dans des entreprises :



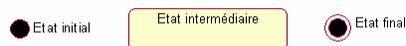
Il y a 3 états qu'une instance de Personne peut prendre:

5

Diagramme d'Etats-Transitions

Etat (3)

- **Etat initial** : il y a toujours un et un seul état initial pour un objet.
- **Etat Final** :
 - Il est possible d'avoir plusieurs état finaux. Chacun correspond à un comportement différent de l' objet.
 - Il est possible de n'avoir aucun état final. Exemple : un système qui ne s'arrête jamais.
- **Etat Final** :
- Représentation graphique :



En règle général l'état initial et l'état final sont facilement identifiable : on a pas besoin de leur donné un nom.

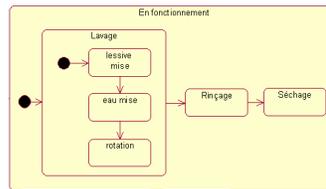
6

Diagramme d'Etats-Transitions

État (4)

■ État composite

- C'est un état qui est composé d'autres sous-états.
- Chaque sous-états peut, à son tour, être composé en d'autres sous-états.
- **Exemple :**
Etat d'une machine à laver en fonctionnement



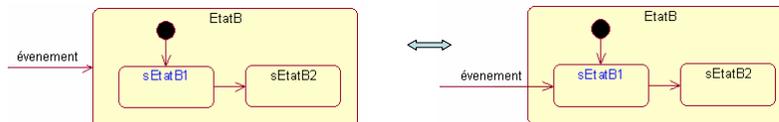
7

Diagramme d'Etats-Transitions

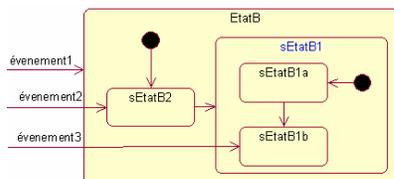
État (5)

■ Transition entre états composites

- Une transition vers un état composite est équivalente à une transition vers le premier sous-état de l'état composite.



- Il est possible pour une transition d'accéder à des sous-états autres que le premier sous-état l'état composite : pouvant appartenir à l'un de ses niveaux hiérarchiques plus profonds .



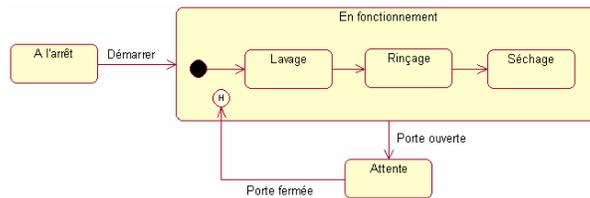
8

Diagramme d'Etats-Transitions

Etat (5)

■ Etat historique

- Par défaut on rentre dans un état composite via le premier sous-état.
- Si l'on souhaite rentrer dans l'état via le dernier sous-état actif avant de le quitter, on utilise un **état historique**, modélisé par un **H encerclé**.
- Exemple : Utilisation d'un état historique pour la réalisation d'un lave-vaisselle.



■ Remarque :

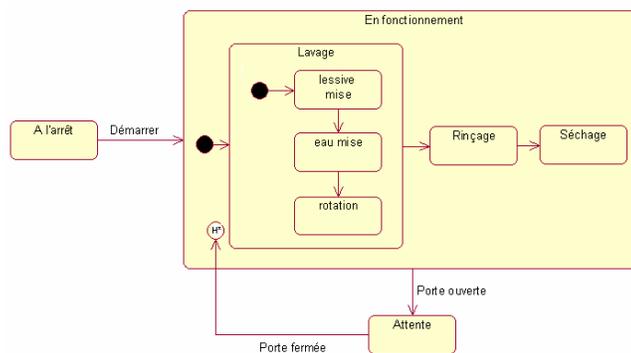
- Peut être placé à n'importe où dans l'état composite
- Plusieurs transitions vers l'état historique peuvent être effectuées

9

Diagramme d'Etats-Transitions

Etat (6)

- Il est possible de mémoriser le dernier sous-état actif, quel que soit son niveau de profondeur en utilisant un **H* encerclé**. Exemple :



10

Diagramme d'Etats-Transitions

Transition (1)

- C'est une relation unidirectionnelle entre deux états indiquant qu'un objet est passé d'un état à l'autre lorsqu'un événement survient.
- Représentation graphique : un arcs orienté



- **Syntaxe :**

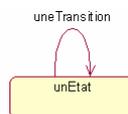
`nom_evenement (args) [garde] [/ action]`

11

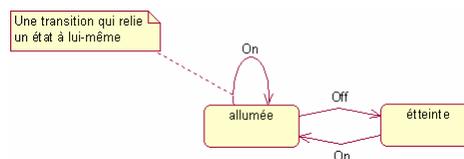
Diagrammes d'Etats-Transitions

Transition (2)

- Le passage de l'objet d'un état à autre peut dépendre de son état source. Par conséquent Une transition ne relie pas nécessairement des états distincts (pas de changement d'état)



Exemple : Une lampe avec 2 boutons-poussoirs (On et Off), si on presse sur On alors qu'elle est déjà allumée, rien ne se produit



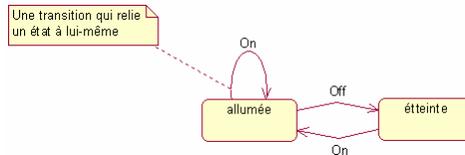
12

Diagrammes d'Etats-Transitions

Transition (2)

- Le passage de l'objet d'un état à autre peut dépendre de son état source.

Exemple : Une lampe avec 2 boutons-poussoirs (On et Off), si on presse sur On alors qu'elle est déjà allumée, rien ne se produit



13

Diagrammes d'Etats-Transitions

Événement (1)

- Appelé également **déclencheur** ou **trigger**.
- C'est une information instantané qui doit être traitée sans délai (un état par contre dure)
- Lorsqu'un événement est reçu par un objet, une transition peut être déclencher pour faire basculer l'objet d'un état vers un nouvel état.
- Un événement peut être paramétrable.

14

Diagrammes d'Etats-Transitions

Événement (2)

- Les types d'événements :

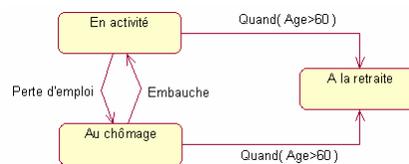
Type d'événement	Description	Syntaxe	Exemple
Appel	Réception d'un message synchrone (pour lequel l'émetteur attend une réponse). Invocation d'une méthode	operationName(p1:type, p2:type, ...)	ClicDroitSouris(p:Point,d:Date)
Signal	Réception d'un message asynchrone.	signalName(p1:type, p2:type, ...)	Imprimer()
modification	Changement de la valeur d'une condition booléenne.	when (exp) ou quand (exp) exp : condition booléenne	Quand(date=01/01/2005)
Temps	Signalement qu'un certain temps s'est écoulé depuis l'entrée dans un état donné.	after (time) ou après (time) time : expression définissent le temps	Après(2 secondes)

15

Diagramme d'Etats-Transitions

Événement (2)

- Exemple 1 :



Chaque transition porte l'événement qui la déclenche

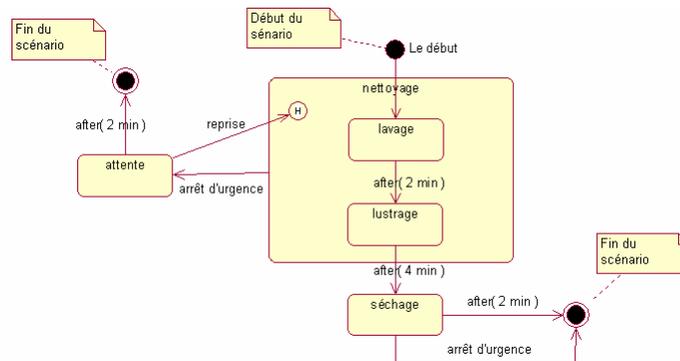
16

Diagramme d'Etats-Transitions

Événement (2)

■ Exemple 2 : Une machine à laver les voiture

La voiture passe par une phase de nettoyage (lavage et lustrage). Le lavage prend 2min. Le lustrage 4 min. Puis le séchage qui prend 2min. Lorsque la machine est en phase de nettoyage, le client peut faire un arrêt d'urgence pour reprendre le nettoyage après. Il dispose de 2 min sinon la machine s'arrête complètement. Le séchage prend 2mn. Si le client fait un arrêt d'urgence la machine s'arrête complètement



17

Diagramme d'Etats-Transitions

Action (1)

- envoi de signaux, invocation d'opération, création ou destruction d'objets.
- Déclenchée par une transition pour faire évoluer l'objet de son état courant vers un autre état.
- Lie les opérations des objets aux événements inscrits dans les diagrammes d'états-transitions.
- Une action a accès aux paramètres de l'événement et aux attributs de l'objet.

18

Diagramme d'Etats-Transitions

Action (2)

- Les types d'actions :

Type d'action	Description	Syntaxe
Affectation	affecte une valeur à une variable.	target:=expression
Appel	Invocation (synchrone) d'une opération d'un objet. Peut retourner une valeur.	opname(arg1, arg2, ...) object.opname(arg1, arg2, ...)
Envoi	Envoi d'un message (signal) asynchrone.	sname(arg1, arg2, ...)
Création	Création d'un nouvel objet.	new Cname(arg 1, arg2, ...)
Destruction	Destruction d'un objet.	object. destroy ()
Séquence	Séquence d'actions	action1; action2; ..

19

Diagramme d'Etats-Transitions

Action (3)

- Certaines actions peuvent être rattachées à un état au lieu d'une transition:
 - Action d'entrée**: action exécutée systématiquement dès l'entrée dans l'état (quelle que soit la transition qui nous y amène...). Notation : **entry**/action (ou **entrée**/action)
 - Action de sortie**: action exécutée systématiquement à la sortie de l'état (quelle que soit la transition qui nous en fait sortir). Notation : **exit**/action (ou **sortie**/action)
 - Action d'exécution** : exécute une activité, càd une opération qui prend un temps significatif lorsque l'objet est dans cet état. Notation : **do**/action (ou **faire**/action)
 - Action sur événement interne** : action exécutée à suite de la génération d'un événement interne (ne provoque pas de changement d'état).
Notation : **event** événement/action

saisie mot de passe

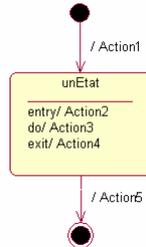
entry/ désactiver l'affichage des caractères
exit/ rendre l'affichage des caractères norm...
do/ saisir un caractère
event saisie d'un caractère/ valider caractère
event after(20s)/ exit

20

Diagrammes d'Etats-Transitions

Action (4)

- En résumé, 5 actions :



- action1 : L'action associée à la transition d'entrée.
- action2 : L'action déclenché à l'entrée dans l'état.
- action3 : L'activité exécutée dans l'état.
- action4 : L'action exécutée à la sortie de l'état (lorsqu'un événement fait passer l'objet dans un autre état).
- Action5 : L'action associée à la transition de sortie de l'état.

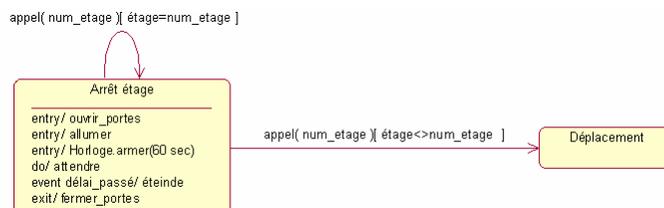
- **Remarque** : L'ordre est sans importance

21

Diagramme d'Etats-Transitions

Action (5)

- **Exemple** : État d'un ascenseur arrêté à l'étage
 - L'appel de l'ascenseur le met dans l'état **Arrêt étage**,
 - les portes s'ouvrent,
 - la lumière s'allume,
 - et l'horloge est armée (60 secondes).
 - En suite on attend : soit le réveil, soit une demande de changement d'étage.
 - Le réveil est un **événement interne** (délai_passé) : engendre l'exécution d'une ou plusieurs actions (éteindre la lumière) sans sortir de l'état.
 - La demande de changement d'étage est un **événement externe** : engendre la sortie de l'état (fermeture des portes) . Si l'étage demandé est l'étage courant, on rentre à nouveau dans l'état Arrêt étage (ouverture des portes, etc.)

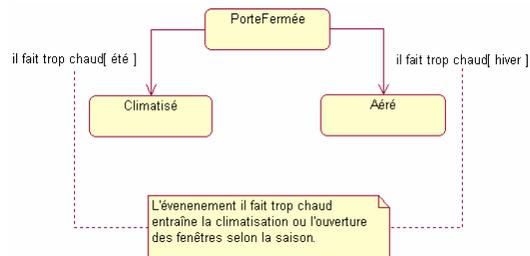


22

Diagramme d'Etats-Transitions

Garde

- Une condition booléenne dont dépend le déclenchement d'une transition à la réception d'un événement,
- évaluée uniquement quand l'événement est déclenché,
- peut contenir des attributs de l'objet ou des paramètres de l'événement associé



23

Diagramme d'Etats-Transitions

Principes de modélisation

- Il s'agit de la modéliser le cycle de vie d'un objet
- Principes de modélisation
 - Établir les états initiaux et finaux.
 - Identifier les évènements auxquels l'objet peut réagir.
 - Identifier les états composites de haut niveau.
 - Détailler les états composites.
 - Identifier les actions d'entrée et de sortie

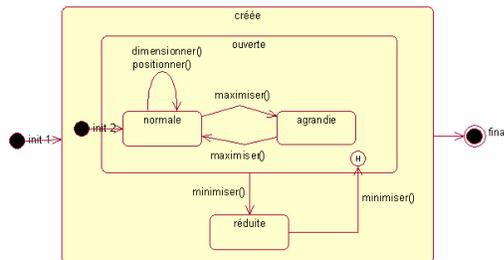
24

Diagramme d'Etats-Transitions

Exemple 1 :

Comportement d'une fenêtre :

- Après sa création, elle répond à stimulus de 3 boutons placés dans l'angle.
- Peut avoir trois états : réduite, normale, agrandie.
- Lorsqu'elle est à l'état normal, elle peut être replacée et redimensionnée.
- Lorsqu'elle est agrandie, elle occupe toute la surface. Il ne peut ni être replacé et redimensionné.
- Lorsqu'on quitte l'état réduite, on retrouve son dernier sous-état lorsqu'elle était à l'état ouverte.



Commentaire

•Init 1 et Init 2 désignent les **points d'entrée** du programme ou du sous-programme. Une nouvelle instance sera initialisé à l'état **créée**, dans le sous-état **ouverte** et dans le sous-état **normal**.

•Le pseudo-état historique permet de retrouver la fenêtre dans son état précédent (normale ou agrandie) quand on quitte l'état réduite.

•L'état final correspond à la destruction de l'instance (la fin de sa vie)

25

Diagramme d'activités

Objectif d'un diagramme d'activité

- Il est similaire un diagrammes d'états-transition : Utilise beaucoup de notations d'états-transitions.
- On ne s'intéresse pas aux états d'un objet, mais plutôt à la réalisation **d'une tâche complexe** (un processus consommant un temps significatif) devant être accomplie séquentiellement.
- Cette tâche est décomposée en **activité**.
- Les transitions sont automatiquement déclenchées par la complétion d'une activité : Le passage d'une activité à une autre se fait **automatiquement lorsqu'une activité est terminée**. Une activité (contrairement à un état) n'attend pas l'occurrence d'un événement pour effectuer une transition. On passe à la prochaine activité dès que la précédente est terminée.
- Un diagramme d'activité peut modéliser :
 - La réalisation d'une méthode,
 - Le déroulement d'un cas d'utilisation

26

Diagramme d'activités

Composants

Trois composants :

- Les activités
- Les transitions
- Les synchronisations
- Les travées

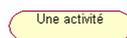
27

Diagramme d'activités

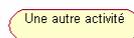
Activité

- ✎ Modélise une étape dans l'exécution d' un algorithme.
- ✎ Consiste en étape dont la terminaison déclenche automatiquement une transition vers une autre étape.
- ✎ **Notation** : un rectangle, avec les côtés de droites et de gauches convexes.

Une activité



Une autre activité



- Le début et la fin (si elle existe) d'un mécanisme sont définis respectivement par un état initial et un état final (il peut y en avoir plusieurs).

● Le début

● Une fin

28

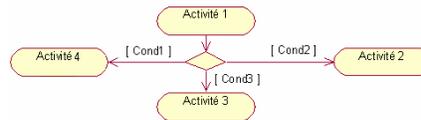
Diagramme d'activités

■ Les transitions

- Un transition relie deux activités
- Une transition ne possède pas signature d'évènement puisque la terminaison de l'activité de départ la déclenche automatiquement.
- **Notation** : une simple flèche.



- Une transition peut faire l'objet d'un débranchement **conditionnel** appelé **décision** : le débranchement de la transition dépend des conditions booléennes et mutuellement exclusive.

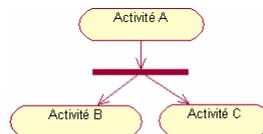


29

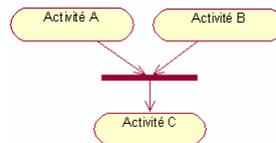
Diagramme d'activités

La synchronisation

- **Synchronisation de flots de contrôle parallèles** : la barre de synchronisation déclenche simultanément les transitions de débranchement.



- **Fusion de flots de contrôle parallèles** : la barre de synchronisation ne peut être franchie que si toutes les transitions en entrée sont déclenchées.

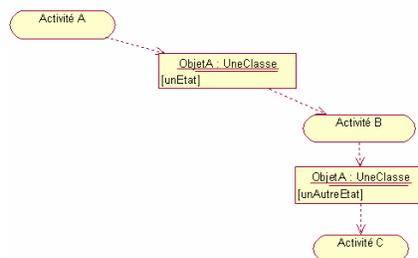


30

Diagramme d'activités

Les Flots entre activités

- Il est possible de représenter les objets s'échangeant entre activités.
- Le même objet peut s'échanger entre plusieurs activités, après modification ou non.
- Si un objet est amené à changer d'état, il faut préciser l'état.
- **Nation :**
 - L'objet : un rectangle indiquant le nom de la classe de l'objet et éventuellement le nom de l'objet et son état.
 - Le lien : une flèche en traits pointillés.

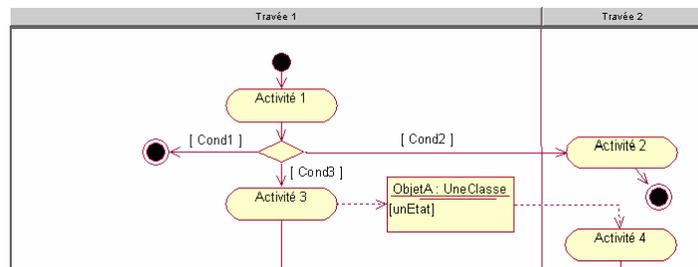


31

Diagramme d'activités

Les travées

- Une travée Modélise une responsabilité.
- Chaque responsabilité peut être assurée par un objet ou plusieurs objets.
- Chaque activité est allouée à une travée donnée.



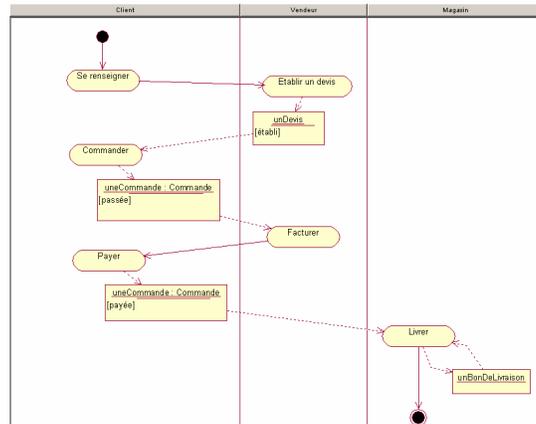
- **Remarque :** La position des travées dans le diagramme d'activités est sans importance.

32

Diagramme d'activités

Exemple

- Modélisation d'une commande via un diagramme d'activité. Le client se renseigne, commande et paye. Le vendeur établit les devis et les factures. La livraison est à la charge du livreur qui établit des bons de livraison.



33

Diagramme de composants

Objectif

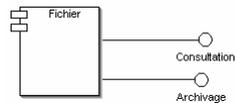
- Décrire les composants et leurs dépendances dans l'environnement de réalisation : vue statique.
- En général, un diagramme de composants n'est utilisé que pour les systèmes complexes.
- Un composant est élément physique. Il peut être un fichier quelconque, un exécutable, une table, une librairie, etc.
- Un composant est éventuellement connecté à d'autres composants, via des dépendances ou des compositions.
- Un composant peut réaliser un ou plusieurs interfaces définissant le comportement offert aux autres composants.
- Un composant peut être déplacé d'un nœud à un autre.
- Un composant peut être remplacé par un autre disposant des mêmes interfaces.

34

Diagramme de composants

Les diagrammes de composants et les classes

- Analogies entre composants et classes
 - Les deux possèdent un nom.
 - Ils peuvent tous deux réaliser un ensemble d'interfaces.
 - Ils peuvent tous deux participer à des relations de dépendances, de généralisation et d'association.
- Différences entre composants et classes:
 - Les classes peuvent avoir des attributs et des opérations.
 - Les composants ne possèdent que des opérations accessibles via des interfaces.
- **Notation** : un rectangle avec deux autres petits rectangles placés sur son côté gauches



Un composant disposant de 2 interfaces

35

Diagramme de composants

Les dépendances entre diagrammes de composants

- Un composant peut dépendre d'un autre lorsque le premier utilise les services offerts par le second.
- **Notation** : une flèche pointillée.

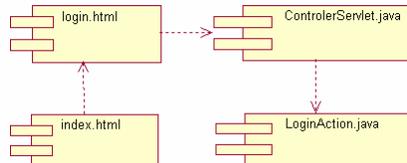


36

Diagramme de composants

Exemple

Un diagramme de composant d'authentification d'un utilisateur



37

Diagramme de déploiement

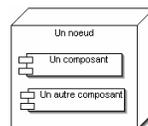
Objectif

- Décrire la disposition physique des différents nœuds (matériels) qui entrent dans la composition du système, ainsi que leur moyens de communication
- Sur ces nœuds seront déployées les différents composants.
- Un nœud dispose obligatoirement d'une mémoire et éventuellement une capacité de calcul.
- Un nœud peut avoir des attributs. Exemple : vitesse du processeur, qté de mémoire, etc.

- **Notation** : Un cube



- Pour montrer qu'un composant réside sur un nœud :



38

Diagramme de déploiement

La nature des nœud

- Il est possible de préciser la nature de l'équipement au moyen d'un stéréotype. UML ne définit aucun stéréotype.
- On propose général les stéréotypes suivants :
 - Les **processeurs** avec puissance de calcul : poste client, serveur, main frame, etc.
 - Les dispositifs (**devices**) sans puissance de calcul : imprimante, scanner, écran, etc.
 - Les mémoires : des dispositifs de stockage : disque dur, etc.
- Exemple de stéréotypes de nœuds

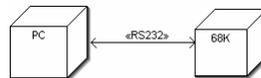


39

Diagramme de déploiement

La communication entre les noeuds

- Les différents nœuds qui apparaissent dans un diagramme de déploiement sont connectés en entre eux par des lignes qui symbolise un support de communication.
- La nature du support de communication précisée au moyen d'un stéréotype

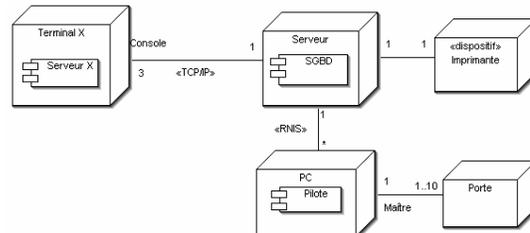


40

Diagramme de déploiement

Exemple

Modélisation d'un système de gestion d'accès à un bâtiment



▪Le diagramme montre que le système est constitué d'un serveur autour duquel gravitent des PC pilotant l'ouverture et la fermeture de portes.

▪On remarque la présence d'informations de **multiplicité** et de rôles comme dans les classes.
–Le nombre de PC n'est pas précisé. En revanche, chaque PC peut être le maître de 10 portes au plus.
– 3 terminaux X jouent le rôle de console pour accéder au système.
–Une imprimante est reliée au serveur.

▪Le diagramme décrit également la **nature des liens de communication** entre les différents noeuds.
–Le serveur et les PC sont reliés par une liaison RNIS.
–Les terminaux X et le serveur communiquent via TCP/IP.
–La nature de la connexion avec l'imprimante n'est pas précisée.

41

Conclusion sur UML

- UML : un langage très riche.
- Ne propose aucune méthodologie : sujet de beaucoup de recherches.
- Les outils d'implémentation d'UML sont en constante évolution.
- Nous avons utilisé la version UML < 2.0 (version actuelle 2).
- UML demande beaucoup de pratique.
- Il faut suivre l'actualité d'UML pour être au courant de son évolution.

42