



Modélisation UML

Diagrammes Structurels



Diagramme des Classes

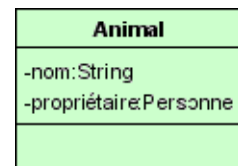
Le diagramme des classes identifie la structure des classe d'un système, y compris les propriétés et les méthodes de chaque classe. Les diverses relations, telles que la relation d'héritage par exemple, qui peuvent exister entre les classes y sont également représentées. Le diagramme des classes est le diagramme le plus largement répandu dans les spécifications d'UML. Une partie de la popularité du diagramme des classes provient du fait qu'il existe des outils tels que Rational XDE, ClassBuilder, Omodo for Elipse ou Poseïdon permettant de produire directement du code source dans les principaux langages informatiques (Java, C++, et de C #, Phyton...) à partir de ces modèles (*forward engennering*). Ces outils peuvent synchroniser les modèles et le code, réduisant votre charge de travail. Ils peuvent également produire des diagrammes de classes à partir du code source orienté objet. (*reverse engennering*)

Représentation

Les éléments d'un diagramme des Classes sont les classes et les relations qui les lient.

Classes

les classes sont les modules de base de la programmation orientée objet. Une classe est représentée en utilisant un rectangle divisé en trois sections. La section supérieure est le nom de la classe. La section centrale définit les propriétés de la classe. La section du bas énumère les méthodes de la classe.



Association

une association est une relation générique entre deux classes. Elle est modélisée par une ligne reliant les deux classes. Cette ligne peut être qualifiée avec le type de relation, et peut également comporter des règles de multiplicité (par exemple un à un, un à plusieurs, plusieurs à plusieurs) pour la relation.



Composition

si une classe ne peut pas exister par elle-même, mais doit être un membre d'une autre classe, alors elle possède une relation de composition avec la classe contenante. Une relation de composition est indiquée par une ligne avec un "diamant" rempli.



Dépendance

quand une classe utilise une autre classe, par exemple comme membre ou comme paramètre d'une de ces fonctions, elle "dépend" ainsi de cette classe. Une



relation de dépendance est représentée par une flèche pointillée.

Agrégation

les agrégations indiquent une relation de contenant-contenu. Elle décrite par une relation "possède". Une relation d'agrégation est représentée par une ligne avec un "diamant" creux.

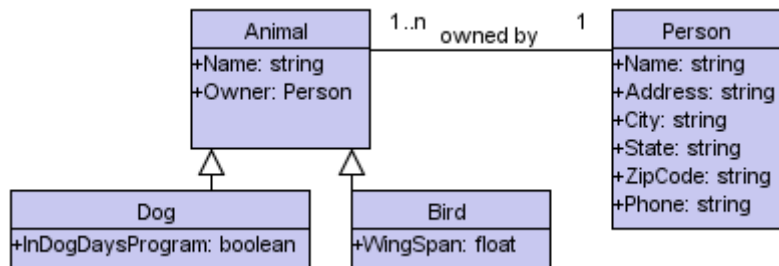


Généralisation

une relation de généralisation est l'équivalent d'une relation d'héritage en terme orientés objet (relation "est-un "). Une relation de généralisation est indiquée par une flèche creuse se dirigeant vers la classe "parent ".



Considérons l'exemple d'un système vétérinaire. Des animaux de compagnie, comme des chiens ou des oiseaux, sont suivis par leurs propriétaires. Le diagramme suivant modélise une solution potentielle. Considérant que les chiens comme les oiseaux sont "un genre" d'animal, nous utilisons un relation de généralisation.



Pour valider votre modèle, vous pouvez affecter des données réelles dans chacune de ces classes. Il y a un diagramme dédié à cette tâche, **le diagramme des objets**.

Diagramme des Objets

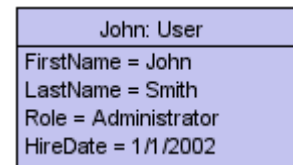
les diagrammes des objets modélisent des exemples de classes. Ce type de diagramme est employé pour décrire le système à un instant particulier. En utilisant cette technique, vous pouvez valider le diagramme des classes et ses règles de multiplicité avec des données et des scénarios réels. D'un point de vue de la notation, les diagrammes des objets empruntent des éléments du diagrammes des classes

Représentation

Souvent, le diagramme des objets utilise une notation plus simple que le diagramme des classes correspondant, se focalisant sur les instances des objets et non sur les relations entre leurs classes (héritage compris). Beaucoup de diagrammes des objets représentent seulement les objets et les associations.

Objet

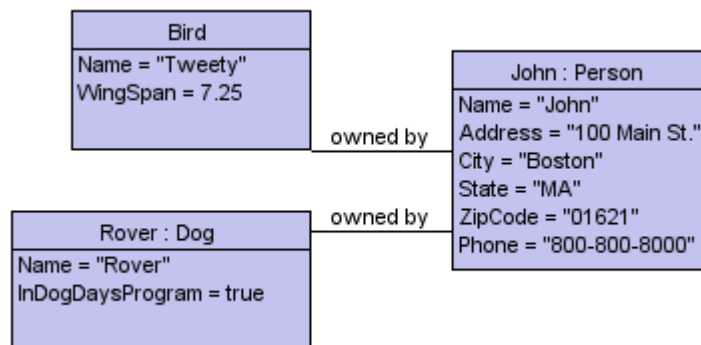
des objets sont identifiés en plaçant le nom d'instance suivi des deux points (:) devant le nom de la classe. Les valeurs de propriété sont écrites comme des paires " nom=valeur ". L'icône représentant un objet est un rectangle divisé en sections.



Association

le diagramme des objets peut contenir également des associations. Souvent, les contraintes, le détail des relations et les règles de multiplicité trouvées dans le diagramme de classe ne sont pas représentés pour ne se concentrer que sur les objets et leurs propriétés. Les associations entre les objets sont représentées simplement en utilisant une ligne les joignant.

Dans la section précédente sur le diagramme des classes, nous avons considéré les classes d'un système vétérinaire. Ici, nous utiliserons des instances de ces classes dans un diagramme des objets. Nous considérerons le cas de John, un amoureux des animaux de compagnie de Boston, du MA, client de l'hôpital vétérinaire. Il a deux animaux de compagnie, Rover, un chien, et Tweety, un oiseau.



Notez comme dans cet exemple, un propriétaire avec plusieurs animaux de compagnie peut être utilisé pour vérifier la relation de multiplicité définie dans le diagramme des classes.

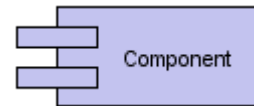
Diagramme des composants

les diagrammes des composants tombent sous la catégorie des diagrammes d'implémentation, un genre de diagramme qui modélise l'implémentation et le déploiement du système. Le diagramme des composants est principalement employé pour décrire les dépendances entre les divers composants logiciels tels que la dépendance entre les fichiers exécutables et les fichiers source. Cette information est semblable à celle contenue dans les fichiers *makefile*, qui décrivent des dépendances des codes sources qui doivent être employés pour compiler correctement une application.

Représentation

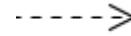
Composant

un composant représente une entité logicielle d'un système. (fichier de code source, programmes, documents, fichiers de ressource .etc.). Un composant est représenté par une boîte rectangulaire, avec deux rectangles dépassant du côté gauche.

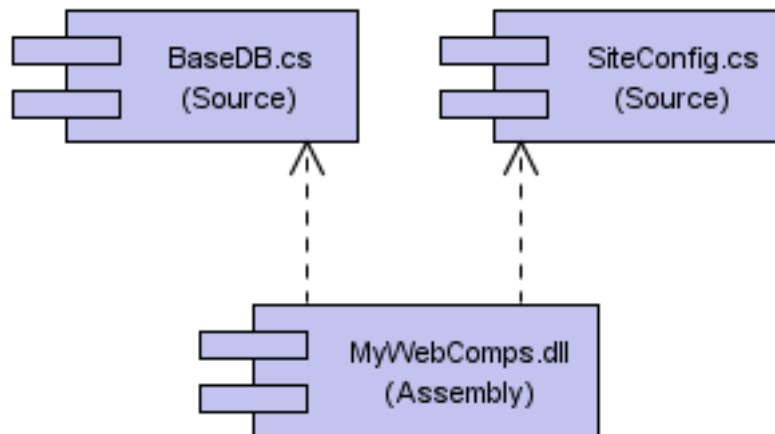


Dépendance

une dépendance est utilisée pour modéliser la relation entre deux composants. La notation utilisée pour cette relation de dépendance est une flèche pointillée, se dirigeant d'un composant donné au composant dont il dépend



Par exemple, le diagramme des composants ci-dessous identifie *myWebComps.dll* comme dépendante des fichiers sources *BaseDB.cs* et *SiteConfig.cs*.



Vous pouvez personnaliser l'affectation des icônes, en utilisant les icônes stéréotypées pour certains composants à la place de l'icône standard. Par exemple, vous pouvez modéliser une application Web, où vous différenciez graphiquement les pages ASP, les fichiers Javascript, et les images. Voici un exemple de diagramme des composants utilisant les icônes stéréotypées pour modéliser les dépendances dans un formulaire ASP:

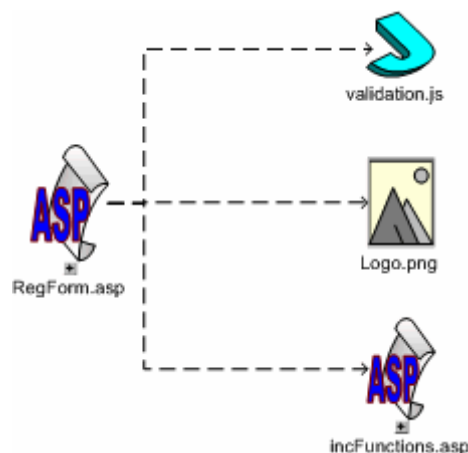

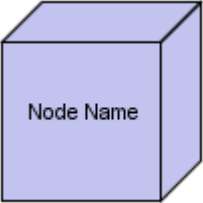



Diagramme de déploiement

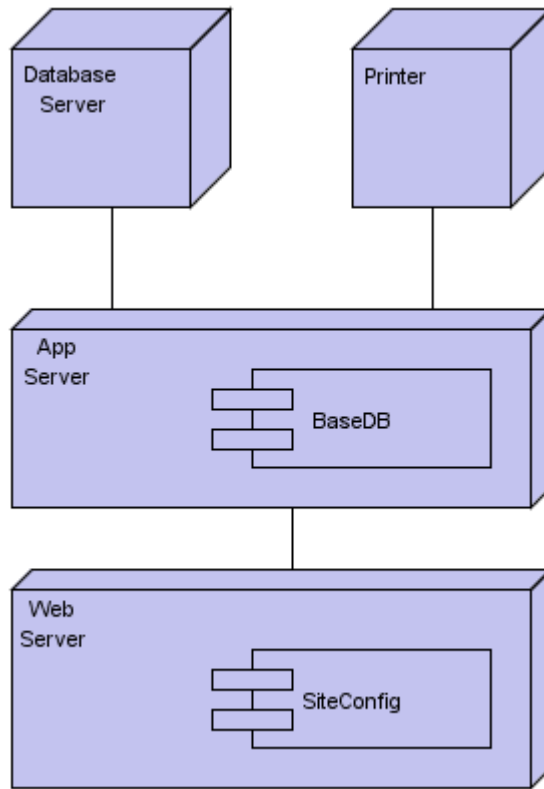
les diagrammes de déploiement sont un autre modèle de la catégorie des diagrammes d'exécution. Le diagramme de déploiement modélise les composants matériels utilisés pour implémenter un système et l'association entre ces composants. Des composants peuvent apparaître également sur un diagramme de déploiement pour montrer le lieu géographique leur déploiement. Des diagrammes de déploiement peuvent être mise en œuvre dès la phase de conception pour documenter l'architecture physique du système.

Représentation

les éléments utilisés dans des diagrammes de déploiement sont des composants, comme dans les diagrammes des composants, et des nœuds, qui représentent les ressources physiques de traitement du système, et leurs associations.

Composant	Un composant représente une entité logicielle du système. (fichier de code source, programmes, documents, fichiers de ressource .etc.). Sur un diagramme de déploiement, les composants sont placés dans des nœuds pour identifier l'endroit de leur déploiement.	
Nœud	Un nœud représente un ensemble d'éléments matériels du système. Cette entité est représentée par un cube tridimensionnel.	
Association	une association, représentée par une ligne pleine entre deux nœuds, indique une ligne de communication entre les éléments matériels.	

Par exemple, le diagramme de déploiement ci-dessous montre que le composant SiteConfig est déployé sur le serveur Web et le composant BaseDB est déployé sur le serveur APP. Nous pouvons également déterminer que le serveur Web communique avec le serveur APP, et le serveur APP communique avec le serveur de base de données et une imprimante.





Modélisation UML

Diagrammes

Comportementaux



Diagramme des cas d'utilisation

les diagrammes des cas d'utilisation identifient les fonctionnalités fournies par le système (cas d'utilisation), les utilisateurs qui interagissent avec le système (acteurs), et les interactions entre ces derniers. Les cas d'utilisation sont utilisés dans la phase d'analyse pour définir les besoins de "haut niveau" du système. Les objectifs principaux des diagrammes des cas d'utilisation sont:

- fournir une vue de haut-niveau de ce que fait le système
- Identifier les utilisateurs ("acteurs") du système
- Déterminer des secteurs nécessitant des interfaces homme-machine. (IHM)

Les cas d'utilisation se prolongent au delà des diagrammes imagés. En fait, des descriptions textuelles des cas d'utilisation sont souvent employées pour compléter ces derniers et représentent leurs fonctionnalités plus en détail.

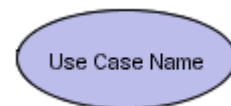
Représentation Graphique

Les composants de base des diagrammes des cas d'utilisation sont l'acteur, le cas d'utilisation, et l'association.

Acteur un acteur est un utilisateur du système, et est représenté par une figure filaire. Le rôle de l'utilisateur est écrit sous l'icône. Les acteurs ne sont pas limités aux humains. Si le système communique avec une autre application, et effectue des entrées/sorties avec elle, alors cette application peut également être considérée comme un acteur.



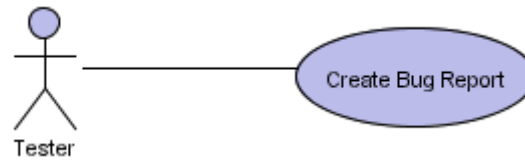
Cas d'utilisation un cas d'utilisation représente une fonctionnalité fournie par le système, typiquement décrite sous la forme *Verbe+objet* (par exemple **immatriculer voiture**, **effacer utilisateur**). Les cas d'utilisation sont représentés par une ellipse contenant leurs nom.



Association les associations sont utilisées pour lier des acteurs avec des cas d'utilisation. Elles indiquent qu'un acteur participe au cas d'utilisation sous une forme quelconque. Les associations sont représentées par une ligne reliant l'acteur et le cas d'utilisation.



L'image suivante montre comment ces trois éléments de base collaborent pour former un diagramme de cas d'utilisation: *le testeur crée un registre des bugs*



Représentation textuelle

Chaque cas d'utilisation, est associé à une série d'actions représentant la fonctionnalité voulue, ainsi que les stratégies à utiliser dans l'alternative où la validation échoue, ou des erreurs se produisent. Ces actions peuvent être également définies dans la description de cas d'utilisation. Rien n'étant prévu à ce sujet dans UML 1.4, il n'y a donc aucune norme pour représenter ces cas textuellement. Cependant, il y a quelques règles communes que vous pouvez suivre:

Règles communes de description textuelle des cas d'utilisation:

- Ecrire un paragraphe décrivant l'ordre des actions
- Lister deux colonnes comprenant d'une part les actions de l'acteur et d'autre part les réponses du système
- Utiliser un patron identifiant les acteurs, les conditions préalables, les postconditions, les scénarios principaux de réussite du processus, .etc.

Rappelez-vous, le but du processus de modélisation est de pouvoir illustrer le mieux possible les besoins du système, aussi n'hésitez pas à employer toutes les méthodes qui permettront une meilleure compréhension des membres du projet.

Voici un exemple de représentation textuelle:

Créer le Registre des Bogues (paragraphe de description)

Le Testeur ouvre un registre de Bogues. Le Testeur indique la source du Bogue, une description du problème, et la personne à qui le Bogue devrait être assigné. Le système enregistre le Bogue et informe la personne affectée qu'un nouveau bogue lui a été soumis.

Créer le Registre des Bogues (patron)

<i>Acteur primaire:</i>	Le Testeur
<i>But :</i>	Le testeur examine une application et découvre un nouveau bogue. Il, ou elle, veut le rapporter de sorte qu'il puisse être traité.
<i>Portée:</i>	système - le système de garantie de la qualité pour l'application XYZ
<i>Niveau:</i>	Utilisateur
<i>Destinataires et intérêts:</i>	Le testeur: veut enregistrer un nouveau bogue L'assigné: veut être avisé de tous les nouveaux bogues Directeur de la qualité: veut connaître tous les bogues enregistrés
<i>Condition préalable:</i>	aucune
<i>Déclenchement:</i>	Le testeur découvre un bogue en examinant une application
<i>Scénario Principal de Succès:</i>	1. Le testeur contrôle lance un nouveau rapport de bogue. 2. Le système enregistre le bogue avec la date de la soumission.

3. Le système informe l'utilisateur affecté.

Extension:




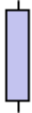
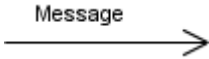
1a. Le testeur ne sait pas à qui assigner le bogue. Le système assigne le bogue au directeur de la qualité.

Diagramme des séquences

les diagrammes des séquences documentent les interactions à mettre en œuvre entre les classes pour réaliser un résultat, tel qu'un cas d'utilisation. UML étant conçu pour la programmation orientée objet, ces communications entre les classes sont reconnues comme des messages. Le diagramme des séquences énumère des objets horizontalement, et le temps verticalement. Il modélise l'exécution des différents messages en fonction du temps.

Représentation

Dans un diagramme des séquences, les classes et les acteurs sont énumérés en colonnes, avec leurs lignes de vie verticales indiquant la durée de vie de l'objet.

Objet	les objets sont des instances des classes, et sont rangés horizontalement. La représentation graphique pour un objet est similaire à une classe (un rectangle) précédée du nom d'objet (facultatif) et d'un point-virgule (:).	
Acteur	les acteurs peuvent également communiquer avec des objets, ainsi ils peuvent eux aussi être énumérés en colonne. Un acteur est modélisé en utilisant le symbole habituel: <i>Stickman</i> .	
Ligne de vie	les lignes de vie, <i>LifeLine</i> , identifient l'existence de l'objet par rapport au temps. La notation utilisée pour une ligne de vie est une ligne pointillée verticale partant de l'objet.	
Activation	les activations, sont modélisées par des boîtes rectangulaires sur la ligne de vie. Elles indiquent quand l'objet effectue une action.	
Message	les messages, modélisés par des flèches horizontales entre les activations, indiquent les communications entre les objets.	

L'exemple ci-dessous représente un diagramme des séquences qui utilise des objets par défaut (aucun nom n'est spécifié). Vous pouvez imaginer beaucoup d'exemples où un utilisateur effectue une action dans l'interface utilisateur, et le système appelle alternativement un autre objet pour le traitement.

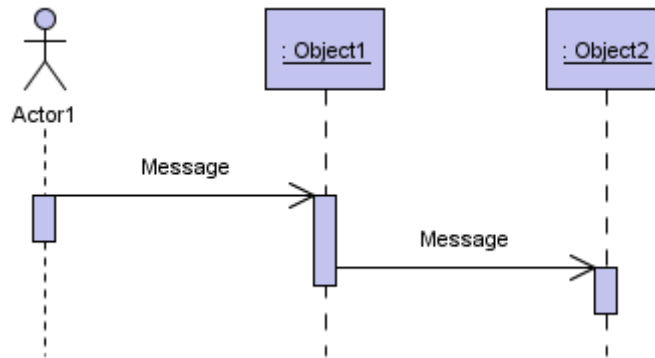


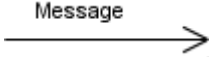


Diagramme des collaborations

Comme les autres diagrammes comportementaux, les diagrammes de collaboration modélisent les interactions entre les objets. Ce type de diagramme est un croisement entre un diagramme des objets et un diagramme des séquences. À la différence du diagramme des séquences qui modélise l'interaction dans un format de type ligne-colonne, le diagramme des collaborations emploie une disposition libre des objets tels qu'on les trouve dans un diagramme des objets. Ceci facilite la vision de toutes les interactions impliquant un objet particulier.

Afin de maintenir l'ordonnancement des messages dans un tel diagramme de formes libres, ces derniers sont notés avec un numéro chronologique. La lecture d'un diagramme des collaborations implique de commencer au message 1.0, et de suivre ainsi la séquence des messages d'objet en objet.

Représentation

Objet	les objets, instances des classes, représentent une des entités impliquées dans les communications. La représentation graphique pour un objet est similaire à une classe (un rectangle) précédée du nom d'objet (facultatif) et d'un point-virgule (;) .	
Acteur	les acteurs peuvent également communiquer avec des objets, aussi peuvent ils être présents sur des diagrammes de collaborations. Un acteur est modélisé en utilisant le symbole habituel: <i>Stickman</i> .	
Message	les messages, modélisés par des flèches entre les objets, sont affectés d'un numéro et indiquent les communications entre les objets.	

Voici l'exemple d'un administrateur utilisant une application Web d'enchaînement pour contrôler un compte d'utilisateur. Notez que vous pouvez suivre le processus d'un objet à l'autre, selon le séquençement ci-dessous:

1. Recherche utilisateur
 - 1.1. Récupération utilisateur
2. Mise à jour utilisateur
 - 2.1. Validation utilisateur
 - 2.2. mise à jour utilisateur

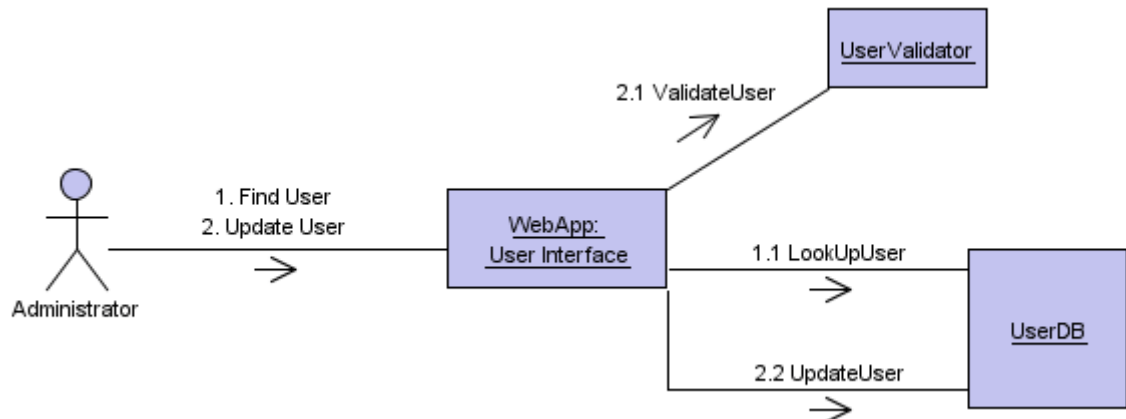
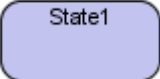
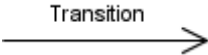




Diagramme d'état

Les diagrammes d'état sont utilisés pour documenter les divers modes ("état") qu'une classe peut prendre, et les événements qui causent une transition d'état. Par exemple, votre téléviseur peut être éteint (dans l'état "éteint"), et l'action sur le bouton d'alimentation le fait s'allumer (état "allumé"). Une nouvelle action sur le bouton d'alimentation provoque une transition d'état "allumé" vers l'état "éteint". En comparaison avec les autres diagrammes comportementaux qui modélisent les interactions entre des classes multiples, les diagrammes d'état modélisent eux typiquement les transitions d'une seule classe.

Représentation

Etat	la notation de l'état décrit le mode de l'entité. Elle est représentée par rectangle avec les coins arrondis, contenant le nom de l'état.	
Transition	une transition décrit le changement de l'état d'un objet, provoqué par un événement. La notation utilisée pour représenter une transition est une flèche, avec le nom d'événement écrit au-dessus, au-dessous, ou à côté.	
Etat initial	l'état initial est l'état d'un objet avant toute transitions. Pour des objets, ceci pourrait être l'état lors de leur instanciation. L'état initial est représenté par un cercle plein. Un seul état initial est autorisé sur un diagramme.	
Etat final	l'état final représente la destruction de l'objet que nous modélisons. Ces états sont représentés par un cercle plein entouré d'un cercle.	

Voici un exemple de diagramme d'état qui modélise l'état du compte d'un utilisateur:

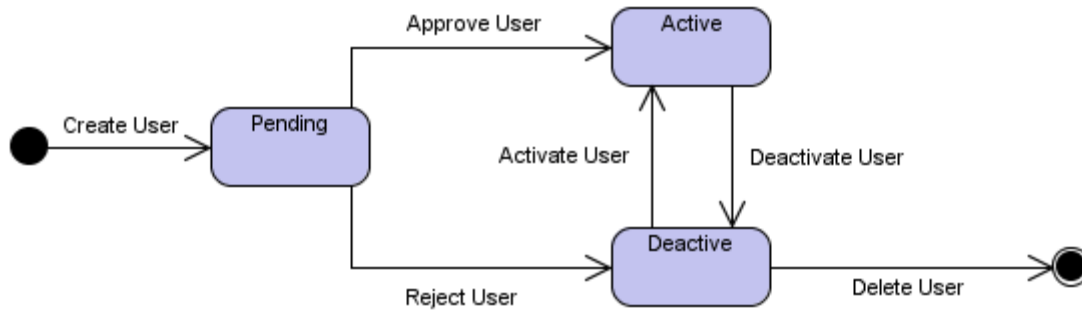

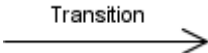
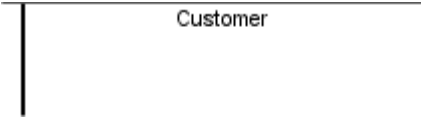




Diagramme d'activité

les diagrammes d'activité sont utilisés pour documenter le déroulement des opérations dans un système, du niveau commercial au niveau opérationnel (de haut en bas). En regardant un diagramme d'activité, vous trouverez des éléments des diagrammes d'état. En fait, le diagramme d'activité est une variante du diagramme d'état où les "états" représentent des opérations, et les transitions représentent les activités qui se produisent quand l'opération est terminée. L'usage général des diagrammes d'activité permet de faire apparaître les flots de traitements induits par les processus internes par rapport aux évènements externes.

Représentation

Etat d'activité	l'état d'activité marque une action faite par un objet. Il est représenté par un rectangle arrondi.	
Transition	quand un état d'activité est accompli, le traitement passe à un autre état d'activité. Les transitions sont utilisées pour marquer ce passage. Les transitions sont modélisées par des flèches.	
Couloir (Swimlane)	Dans un diagramme d'activité, on peut placer les activités dans des couloirs (<i>Swimlanes</i>) qui représentent des systèmes. Les objets sont énumérés au dessus de la colonne, et les barres verticales séparent les colonnes pour former les swimlanes.	
Etat initial	l'état initial marque le point d'entrée la première activité. Il est représenté, comme dans le diagramme d'état, par un cercle plein. Il ne peut y avoir qu'un seul état initial sur un diagramme.	
Etat final	L'état final marque la fin du déroulement des opérations	

modélisées. Il peut y avoir des états finaux multiples sur un diagramme. Ils sont représentés par un cercle plein entouré d'un autre cercle.

Barre de Synchronisation

Souvent, certaines activités peuvent être faites en parallèle. Pour dédoubler le traitement "Fork", ou le reprendre quand des activités multiples ont été accomplies ("join"), des barres de synchronisation sont utilisées. Celles-ci sont modélisées par des rectangles pleins, avec des transitions multiples entrantes ou sortantes.



Exemple

Considérons l'exemple de diagramme suivant, respectant les spécifications d'UML 1.4. Le diagramme d'activité commence par un appel à l'activité "Request Service". A l'accomplissement de cette opération, une barre de synchronisation est utilisée pour indiquer un traitement en parallèle, où le paiement du client, et la prise en charges de la commande par le service des ventes et le magasin sont effectués simultanément. Une autre barre de synchronisation est utilisée pour indiquer que quand le client a bien payé et que la commande à bien été procédée. La commande est alors prête pour la livraison. Après la livraison le client peut recevoir sa commande et le processus est complet.

