

Conception, Projet

Cours 9 - Conception détaillée - UML p.2

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

- 1 Modèles détaillés UML
- 2 Cas d'utilisation (Use Case)
- 3 Classe (Class)
- 4 Séquence (Sequence)
- 5 Activité (Activity)

Modèles détaillés UML

Voici les modèles détaillés UML qui seront abordés dans le cadre de ce cours :

- Cas d'utilisation (Use Case)
- Classe (Class)
- Séquence (Sequence)
- Activité (Activity)

Modèles détaillés UML

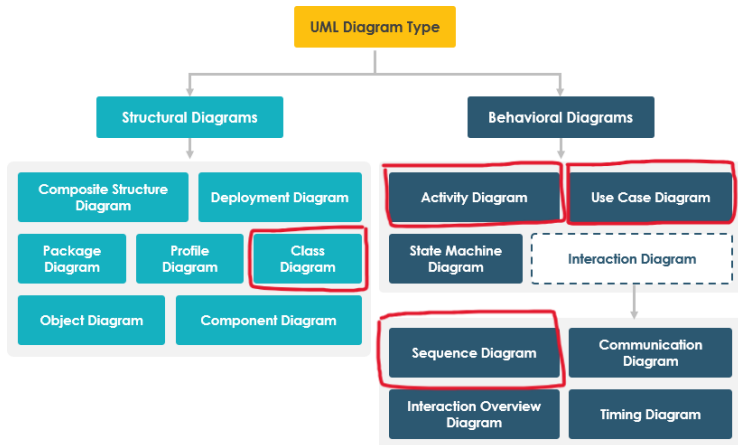


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/overview-of-the-14-uml-diagram-types/>

Modèles détaillés UML - Cas d'utilisation (Use Case)

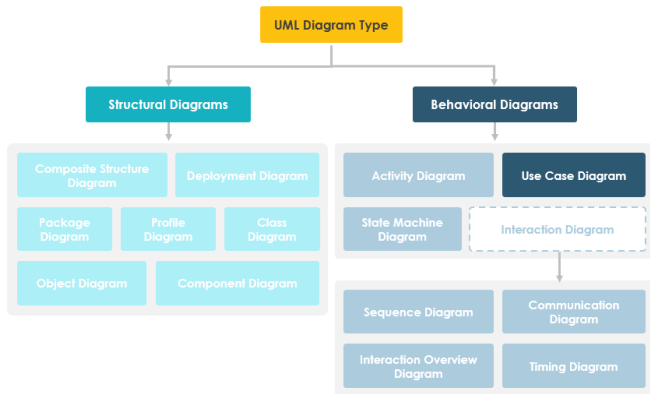


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case)

Les diagrammes de cas d'utilisation sont généralement développés au début du développement et les gens appliquent souvent la modélisation des cas d'utilisation aux fins suivantes :

- Spécifier le contexte d'un système
- Saisir les exigences d'un système
- Valider une architecture de systèmes
- Piloter la mise en œuvre et générer des cas de test
- Développé par des analystes en collaboration avec des experts du domaine

Modèles détaillés UML - Cas d'utilisation (Use Case) - Items

Voici des items utilisés pour un diagramme UML de cas d'utilisation :

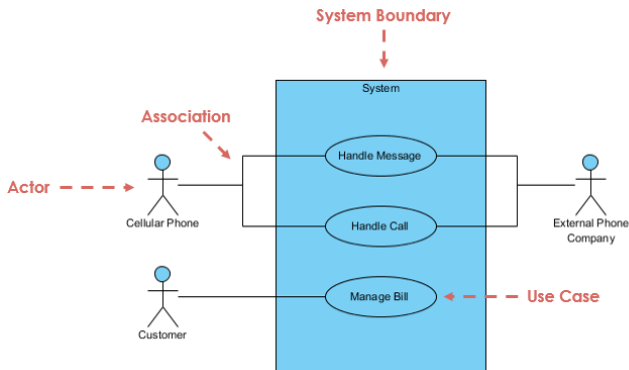


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Items - Acteur (Actor)

- Quelqu'un interagit avec le cas d'utilisation (fonction système).
- Nommé par son nom.
- L'acteur joue un rôle dans l'entreprise.
- Semblable au concept d'utilisateur, mais un utilisateur peut jouer des rôles différents. Par exemple :
 - Un prof. peut être instructeur et aussi chercheur
 - Joue 2 rôles avec deux systèmes
- L'acteur déclenche le ou les cas d'utilisation.
- L'acteur a une responsabilité envers le système (entrées), et l'acteur a des attentes du système (extrants).



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Items - Cas d'utilisation (Use case)

- Fonction du système (processus - automatisé ou manuel).
- Nommé par verbe + nom (ou phrase nominale).
- c'est-à-dire faire quelque chose.
- Chaque acteur doit être lié à un cas d'utilisation, tandis que certains cas d'utilisation peuvent ne pas être liés à des acteurs.



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Items - Lien de communication (Communication link)

- La participation d'un acteur dans un cas d'utilisation est montrée en connectant un acteur à un cas d'utilisation par un lien solide.
- Les acteurs peuvent être connectés à des cas d'utilisation par des associations, indiquant que l'acteur et le cas d'utilisation communiquent entre eux à l'aide de messages.

Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Items - Étendue du système (Boundary of system)

- La limite du système est potentiellement l'ensemble du système tel que défini dans le document sur les exigences.
- Pour les systèmes complexes et de grande taille, chaque module peut être la limite du système.
- Par exemple, pour un système ERP pour une organisation, chacun des modules tels que le personnel, la paie, la comptabilité, etc.
- Peut former une limite système pour les cas d'utilisation spécifiques à chacune de ces fonctions d'entreprise.
- L'ensemble du système peut couvrir tous ces modules illustrant la limite globale du système



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Relations

Les cas d'utilisation partagent différents types de relations.

Définir la relation entre deux cas d'utilisation est la décision des analystes de logiciels du diagramme de cas d'utilisation.

Une relation entre deux cas d'utilisation consiste essentiellement à modéliser la dépendance entre les deux cas d'utilisation.

La réutilisation d'un cas d'utilisation existant en utilisant différents types de relations réduit l'effort global requis dans le développement d'un système.

Les relations de cas d'utilisation sont répertoriées comme suit :

Modèles détaillés UML - Cas d'utilisation (Use Case) - Relations - Extends

- Indique qu'un cas d'utilisation "Mot de passe non valide" peut inclure (sous réserve de spécifié dans l'extension) le comportement spécifié par le cas d'utilisation de base "Compte de connexion".
- Représenter avec une flèche dirigée ayant une ligne pointillée. L'extrémité de la pointe de flèche pointe vers le cas d'utilisation de base et le cas d'utilisation enfant est connecté à la base de la flèche.



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Relations - Include

- Lorsqu'un cas d'utilisation est représenté comme utilisant la fonctionnalité d'un autre cas d'utilisation, la relation entre les cas d'utilisation est nommée comme relation include ou uses.
- Un cas d'utilisation inclut les fonctionnalités décrites dans un autre cas d'utilisation dans le cadre de son flux de processus métier.
- Une relation d'inclusion est représentée avec une flèche dirigée ayant une ligne pointillée. L'extrémité de la pointe de flèche pointe vers le cas d'utilisation enfant et le cas d'utilisation parent connecté à la base de la flèche.



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Relations - Generalization

- Une relation de généralisation est une relation parent-enfant entre les cas d'utilisation.
- Le cas d'utilisation enfant est une amélioration du cas d'utilisation parent.
- La généralisation est représentée sous la forme d'une flèche dirigée avec une pointe de flèche triangulaire. Le cas d'utilisation enfant est connecté à la base de la flèche.
- La pointe de la flèche est connectée au cas d'utilisation parent.



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Exemples - Lien d'association

Un diagramme de cas d'utilisation illustre un ensemble de cas d'utilisation pour un système, c'est-à-dire les acteurs et les relations entre les acteurs et les cas d'utilisation.

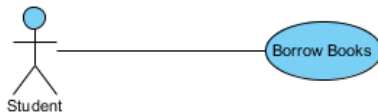


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Exemples - Include

La relation include ajoute des fonctionnalités supplémentaires non spécifiées dans le cas d'utilisation de base. La relation d'inclusion est utilisée pour inclure le comportement commun d'un cas d'utilisation inclus dans un cas d'utilisation de base afin de prendre en charge la réutilisation du comportement commun.

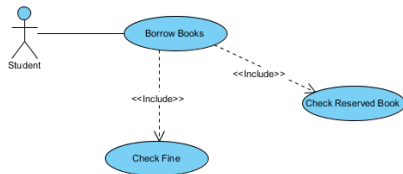


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Exemples - Extends

Les relations d'extension sont importantes car elles montrent des fonctionnalités facultatives ou le comportement du système. La relation d'extension est utilisée pour inclure le comportement facultatif d'un cas d'utilisation d'extension dans un cas d'utilisation étendu. Jetez un oeil à l'exemple de diagramme de cas d'utilisation ci-dessous. Il affiche un connecteur d'extension et un point d'extension "Search".

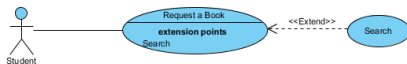


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Exemples - Generalization

Une relation de généralisation signifie qu'un cas d'utilisation enfant hérite du comportement et de la signification du cas d'utilisation parent. L'enfant peut ajouter ou remplacer le comportement du parent. La figure ci-dessous fournit un exemple de cas d'utilisation en montrant deux connecteurs de généralisation qui se connectent entre les trois cas d'utilisation.

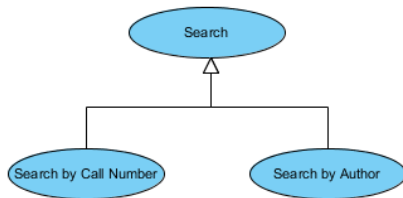


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Exemples - Système de vente de véhicules

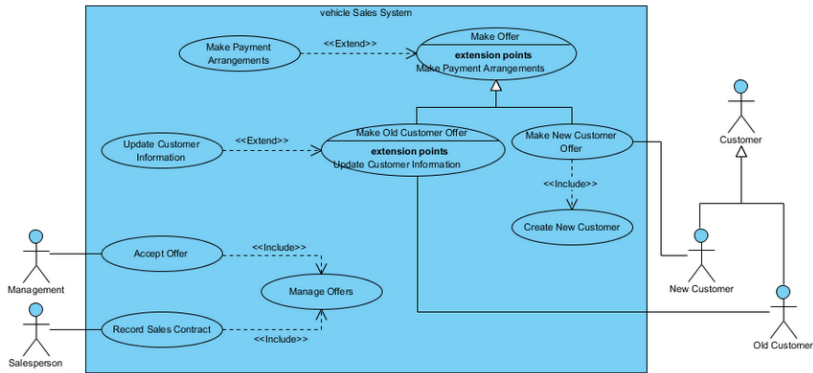


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

Modèles détaillés UML - Cas d'utilisation (Use Case) - Identifier les acteurs

Souvent, les gens trouvent qu'il est plus facile de commencer le processus d'obtention des exigences en identifiant les acteurs. Les questions suivantes peuvent vous aider à identifier les acteurs de votre système (Schneider et Winters - 1998) :

- Qui utilise le système?
- Qui installe le système?
- Qui démarre le système?
- Qui entretient le système?
- Qui arrête le système?
- Quels autres systèmes utilisent ce système?
- Qui obtient de l'information de ce système?
- Qui fournit de l'information au système?
- Y a-t-il quelque chose qui se passe automatiquement à l'heure actuelle?

Modèles détaillés UML - Cas d'utilisation (Use Case) - Identifier les cas d'utilisation

Identifier les cas d'utilisation, puis le processus d'obtention basé sur un scénario se poursuit en demandant quelle valeur observable visible de l'extérieur chaque acteur désire. Les questions suivantes peuvent être posées pour identifier les cas d'utilisation, une fois que vos acteurs ont été identifiés (Schneider et Winters - 1998) :

- Quelles fonctions l'acteur voudra-t-il du système?
- Le système stocke-t-il des informations?
- Quels acteurs vont créer, lire, mettre à jour ou supprimer ces informations?
- Le système doit-il informer un acteur des changements dans l'état interne?
- Y a-t-il des événements externes dont le système doit être au courant?
- Quel acteur informe le système de ces événements?

Modèles détaillés UML - Cas d'utilisation (Use Case) - Conseils

Voici des conseils utiles pour aider dans la création de ce type de diagramme :

- Structurez et organisez toujours le diagramme de cas d'utilisation du point de vue des acteurs.
- Les cas d'utilisation doivent commencer simplement et à la vue la plus élevée possible.
- Ce n'est qu'alors qu'ils pourront être affinés et détaillés davantage. Les diagrammes de cas d'utilisation sont basés sur la fonctionnalité et devraient donc se concentrer sur le "quoi" et non sur le "comment".

Modèles détaillés UML - Classe (Class)

Une forme standard de diagramme de cas d'utilisation est définie dans le langage de modélisation unifié comme indiqué dans l'exemple de diagramme de cas d'utilisation ci-dessous :

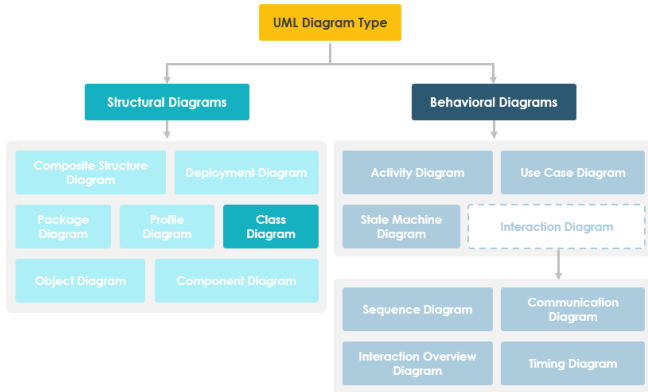


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class)

Un diagramme de classe UML est composé de :

- De classes diverses représentant des entités du domaine d'affaires et/ou des objets
- Un ensemble de relations diverses entre les classes établies

Modèles détaillés UML - Classe (Class)

Une classe consiste d'un **nom**, d'**attributs**, et de **méthodes** :

Le **nom** apparaît dans la première partition.

Les **attributs** dans la troisième partition :

- Le type d'attribut est affiché après les deux-points.
- Les attributs sont mappés sur les variables membres (membres de données) dans le code.

Les **méthodes** dans la deuxième partition :

- Le type de retour d'une méthode est affiché après les deux-points à la fin de la signature de la méthode.
- Le type de retour des paramètres de la méthode est affiché après les deux-points suivant le nom du paramètre.
- Les opérations sont mappées sur les méthodes de classe dans le code

Modèles détaillés UML - Classe (Class)

La représentation graphique de la classe
- MyClass comme indiqué ci-dessus :

- MyClass a 3 attributs et 3 opérations
- Le paramètre p3 d'op2 est de type int
- op2 renvoie un float
- op3 renvoie un pointeur (désigné par un *) vers Class6

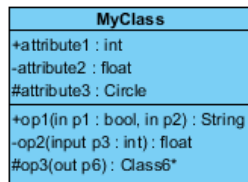


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Relations entre classes

Une classe peut être impliquée dans une ou plusieurs relations avec d'autres classes. Une relation peut être l'un des types suivants :

- Generalisation (Inhéritence)
- Association simple
- Aggrégation
- Composition
- Dépendance

Modèles détaillés UML - Classe (Class) - Relations entre classes - Generalisation (Inhéritence)

- Représente une relation “est une”.
- Un nom de classe abstrait est affiché en italique.
- SubClass1 et SubClass2 sont des spécialisations de Super Class.
- Une ligne pleine avec une pointe de flèche creuse qui pointe de l'enfant à la classe parente

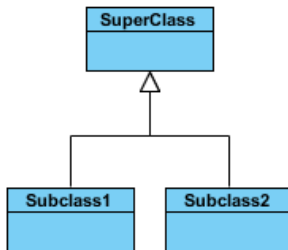


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Relations entre classes - Association simple

- Un lien structurel entre deux classes homologues.
- Il existe une association entre la classe 1 et la classe 2
- Une ligne pleine reliant deux classes

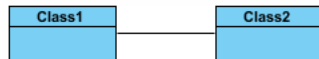


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Relations entre classes - Aggrégation

- La classe 2 fait partie de la classe 1.
- De nombreuses instances (désignées par *) de Class2 peuvent être associées à Class1.
- Les objets de classe 1 et de classe 2 ont des durées de vie distinctes.
- Une ligne pleine avec un diamant non rempli à l'extrémité de l'association reliée à la classe de composite

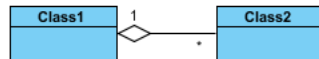


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Relations entre classes - Composition

- Les objets de classe 2 vivent et meurent avec la classe 1.
- La classe 2 ne peut pas se tenir seule.
- Une ligne pleine avec un diamant rempli à l'association reliée à la classe de composite

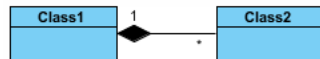


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Relations entre classes - Dépendance

- Existe entre deux classes si les modifications apportées à la définition de l'une peuvent entraîner des changements dans l'autre (mais pas l'inverse).
- La classe 1 dépend de la classe 2
- Une ligne pointillée avec une flèche ouverte

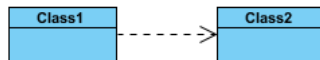


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Visibilité des attributs et opérations/méthodes

Dans la conception orientée objet, il existe une notation de visibilité pour les attributs et les opérations. Le langage UML identifie quatre types de visibilité : public, protégé, privé et package.

Les symboles +, -, # et ~ avant un attribut et un nom d'opération dans une classe indiquent la visibilité de l'attribut et de l'opération :

- + désigne les attributs ou les opérations publiques
- - désigne des attributs ou des opérations privés
- # indique des attributs ou des opérations protégés
- ~ désigne les attributs ou les opérations du package

Modèles détaillés UML - Classe (Class) - Visibilité des attributs et opérations/méthodes

La représentation graphique de la classe “MyClass” comme indiqué ci-dessus :

- attribute1 et op1 de MyClassName sont publics
- attribute3 et op3 sont protégés.
- attribute2 et op2 sont privés.

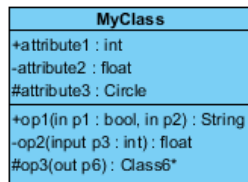


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Visibilité des attributs et opérations/méthodes

Droits d'accès	public (+)	private (-)	protected (#)	Package (~)
Même classe	oui	oui	oui	oui
Classe dérivée	oui	non	oui	oui
Autre classe	oui	non	non	même package

Modèles détaillés UML - Classe (Class) - Multiplicité

Le nombre d'objets de chaque classe qui participent aux relations et la multiplicité peut être exprimé comme suit :

- Exactement un - 1
- Zéro ou un - 0..1
- Beaucoup - 0..* ou *
- Un ou plusieurs - 1..*
- Nombre exact - ex. 3..4 ou 6
- Une relation complexe - par exemple 0..1, 3..4, 6.* signifierait n'importe quel nombre d'objets autres que 2 ou 5

Modèles détaillés UML - Classe (Class) - Exemple - Multiplicité

Exigence : Un étudiant peut suivre de nombreux cours et de nombreux étudiants peuvent être inscrits à un cours.

Dans l'exemple ci-dessous, le diagramme de classe (à gauche), décrit l'énoncé de l'exigence ci-dessus pour le modèle statique tandis que le diagramme d'objet (à droite) montre l'instantané (une instance du diagramme de classe) de l'inscription au cours pour les cours de génie logiciel et de gestion de base de données respectivement)

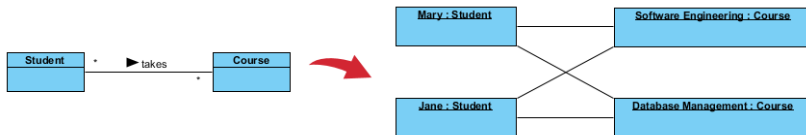


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Exemple - Pièces d'ordinateur

- Une agrégation est un cas particulier d'association désignant une hiérarchie "composée de"
- L'agrégat est la classe parente, les composants sont les classes enfants

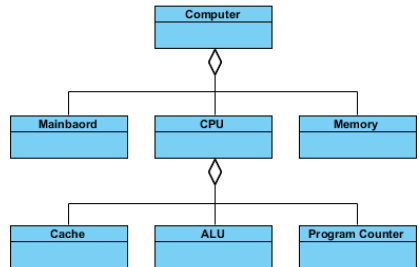


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Exemple - Pièces d'ordinateur

- L'héritage est un autre cas particulier d'une association désignant un "genre de" hiérarchie
- L'héritage simplifie le modèle d'analyse en introduisant une taxonomie
- Les classes enfants héritent des attributs et des opérations de la classe parente.

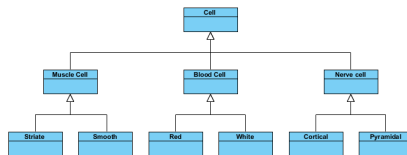


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Exemple - Fenêtre informatique

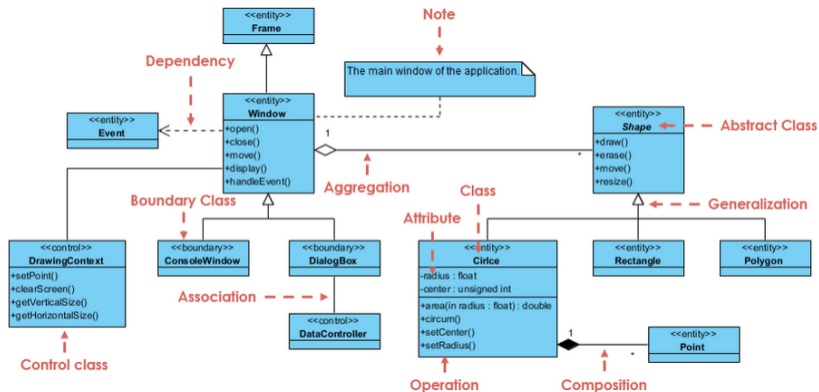


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Modèles détaillés UML - Classe (Class) - Exemple - Fenêtre informatique

Nous pouvons interpréter la signification du diagramme de classe ci-dessus en lisant les points comme suit :

- La forme est une classe abstraite. Il est indiqué en italique.
- La forme est une superclasse. Le cercle, le rectangle et le polygone sont dérivés de la forme. En d'autres termes, un cercle est-une forme. Il s'agit d'une relation généralisation / héritage.
- Il existe une association entre DialogBox et DataController.
- La forme fait partie de La fenêtre. Il s'agit d'une relation d'agrégation. La forme peut exister sans fenêtre.
- Le point fait partie du cercle. Il s'agit d'une relation de composition. Le point ne peut exister sans cercle.

Modèles détaillés UML - Classe (Class) - Exemple - Fenêtre informatique

- La fenêtre dépend de l'événement. Toutefois, Event ne dépend pas de Window.
- Les attributs de Circle sont radius et center. Il s'agit d'une classe d'entités.
- Les noms de méthode de Circle sont area(), circum(), setCenter() et setRadius().
- Le rayon de paramètre dans Circle est un paramètre in de type float.
- La méthode area() de la classe Circle renvoie une valeur de type double.
- Les attributs et les noms de méthode de Rectangle sont masqués. Certaines autres classes du diagramme ont également leurs attributs et leurs noms de méthode masqués.

Modèles détaillés UML - Classe (Class) - Systèmes complexes

Inévitablement, si vous modélisez un grand système ou un grand secteur d'activité, il y aura de nombreuses entités que vous devrez considérer.

Devrions-nous utiliser plusieurs ou un diagramme de classe unique pour modéliser le problème?

La réponse est la suivante :

- Au lieu de modéliser chaque entité et ses relations sur un diagramme de classe unique, il est préférable d'utiliser plusieurs diagrammes de classe.
- La division d'un système en diagrammes à plusieurs classes rend le système plus facile à comprendre, surtout si chaque diagramme est une représentation graphique d'une partie spécifique du système.

Modèles détaillés UML - Séquence (Sequence)

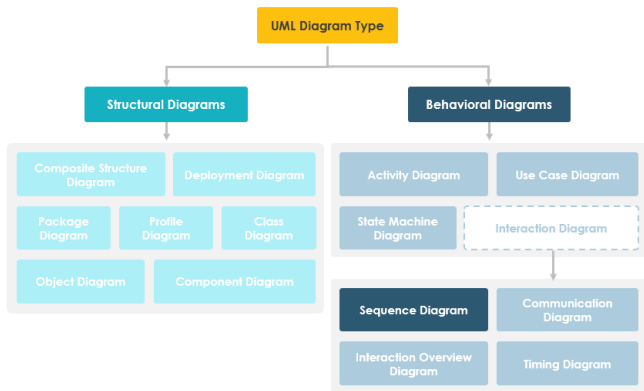


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML- Séquence (Sequence)

Diagrammes de séquence capture :

- l'interaction qui a lieu dans une collaboration qui réalise un cas d'utilisation ou une opération (diagrammes d'instance ou diagrammes génériques)
- l'interactions de haut niveau entre l'utilisateur du système et le système, entre le système et d'autres systèmes, ou entre les sous-systèmes (parfois appelés diagrammes de séquence du système)

Modèles détaillés UML- Séquence (Sequence)

But du diagramme de séquence :

- Modéliser l'interaction de haut niveau entre les objets actifs dans un système
- Modéliser l'interaction entre les instances d'objet au sein d'une collaboration qui réalise un cas d'utilisation
- Modéliser l'interaction entre les objets au sein d'une collaboration qui réalise une opération
- Modéliser les interactions génériques (montrant tous les chemins possibles à travers l'interaction) ou des instances spécifiques d'une interaction (montrant un seul chemin à travers l'interaction)

Modèles détaillés UML- Séquence (Sequence)

Les diagrammes de séquence montrent les éléments au fur et à mesure qu'ils interagissent au fil du temps et ils sont organisés en fonction de l'objet (horizontalement) et du temps (verticalement) :

Dimension d'objet

- L'axe horizontal montre les éléments qui sont impliqués dans l'interaction. De manière conventionnelle, les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils participent à la séquence de messages. Toutefois, les éléments de l'axe horizontal peuvent apparaître dans n'importe quel ordre.

Dimension temporelle

- L'axe vertical représente les procédures temporelles (ou les progrès) vers le bas de la page.

Modèles détaillés UML- Séquence (Sequence)

Le temps dans un diagramme de séquence est tout au sujet de l'ordre, pas de la durée. L'espace vertical dans un diagramme d'interaction n'est pas pertinent pour la durée de l'interaction.

Modèles détaillés UML- Séquence (Sequence) - Exemple - Système pour Hôtel

Le diagramme de Séquence est un diagramme d'interaction qui détaille comment les opérations sont effectuées - quels messages sont envoyés et quand.

Les diagrammes de séquence sont organisés en fonction du temps.

Le temps passe au fur et à mesure que vous descendez la page. Les objets impliqués dans l'opération sont répertoriés de gauche à droite en fonction du moment où ils participent à la séquence de messages.

Modèles détaillés UML - Séquence (Sequence) - Exemple - Système pour Hôtel

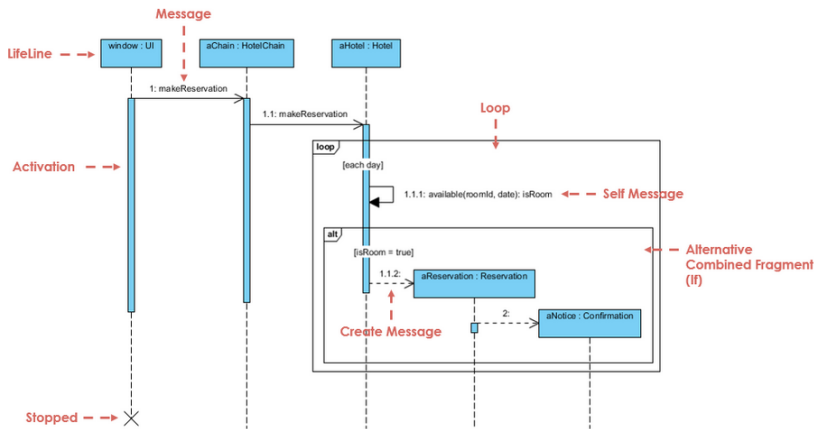


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Acteur

- un type de rôle joué par une entité qui interagit avec le sujet (p. ex., en échangeant des signaux et des données)
- externe au sujet (c.-à-d. dans le sens où une instance d'un acteur ne fait pas partie de l'instance de son sujet correspondant).
- représentent des rôles joués par des utilisateurs humains, du matériel externe ou d'autres sujets.
- Un acteur ne représente pas nécessairement une entité physique spécifique, mais simplement un rôle particulier d'une entité
- Une personne peut jouer le rôle de plusieurs acteurs différents et, inversement, un acteur donné peut être joué par plusieurs personnes différentes.

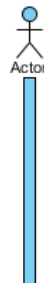


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Ligne de vie (lifeline)

- Une ligne de vie représente un participant individuel à l'Interaction.



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Activation

- Un rectangle mince sur une ligne de vie représente la période pendant laquelle un élément effectue une opération.
- Le haut et le bas du rectangle sont alignés avec l'initiation et le temps d'achèvement respectivement

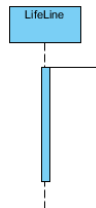


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Message d'appel

- Un message définit une communication particulière entre les lignes de vie d'une interaction.
- Le message d'appel est une sorte de message qui représente un appel du fonctionnement de la ligne de vie cible.

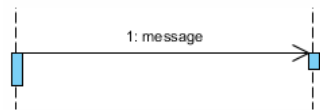


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Message de retour

- Un message définit une communication particulière entre les lignes de vie d'une interaction.
- Le message de retour est une sorte de message qui représente la transmission d'informations à l'appelant d'un ancien message correspondant.

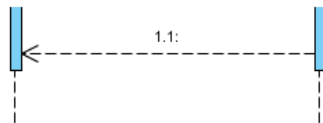


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Message à soi-même (self)

- Un message définit une communication particulière entre les lignes de vie d'une interaction.
- Le message de soi est une sorte de message qui représente l'invocation du message de la même bouée de sauvetage.



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Message récursif

- Un message définit une communication particulière entre les lignes de vie d'une interaction.
- Le message récursif est une sorte de message qui représente l'invocation du message de la même bouée de sauvetage. Sa cible pointe vers une activation au-dessus de l'activation d'où le message a été appelé.

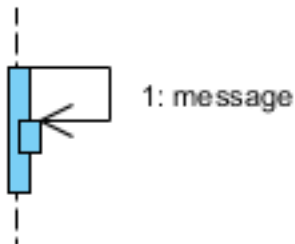


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Message de création

- Un message définit une communication particulière entre les lignes de vie d'une interaction.
- Message de création est une sorte de message qui représente l'instanciation de la bouée de sauvetage (cible).

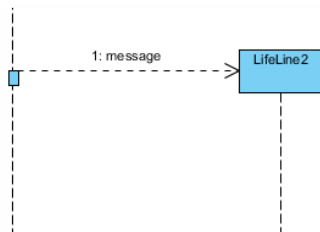


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Message de destruction

- Un message définit une communication particulière entre les lignes de vie d'une interaction.
- Message de destruction est une sorte de message qui représente la demande de destruction du cycle de vie de la cible.

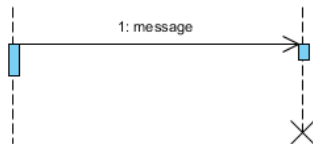


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Message de durée

- Un message définit une communication particulière entre les lignes de vie d'une interaction.
- Le message de durée indique la distance entre deux instants temporels pour un appel de message.

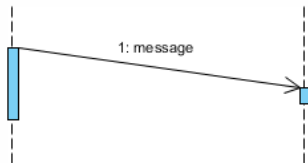


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Classe (Class) - Notations des diagrammes de Séquence - Note

- Une note (commentaire) donne la possibilité de joindre diverses remarques à des éléments.
- Un commentaire n'a aucune force sémantique, mais peut contenir des informations utiles à un modélisateur.



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML- Séquence (Sequence) - Message et orientation du contrôle

- Un événement est n'importe quel moment dans une interaction où quelque chose se produit.
- Objectif du contrôle : aussi appelé occurrence d'exécution, une occurrence d'exécution
- Le message est représenté par le rectangle mince sur la ligne de vie.
- Celui-ci représente la période au cours de laquelle un élément effectue une opération. Le haut et le bas du rectangle sont alignés avec l'initiation et le temps d'achèvement respectivement.

Modèles détaillés UML- Séquence (Sequence) - Message et orientation du contrôle

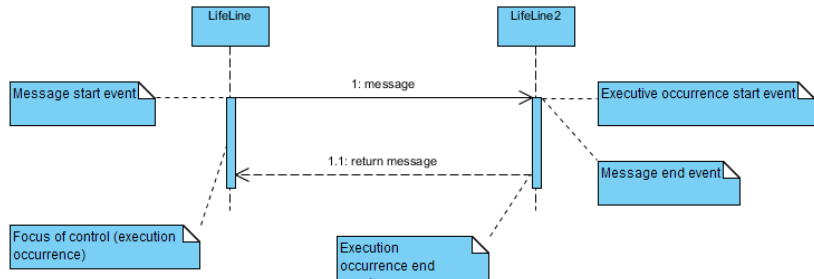


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML- Séquence (Sequence) - Fragments

Un fragment de séquence est représenté sous la forme d'une boîte, appelée fragment combiné, qui entoure une partie des interactions dans un diagramme de séquence.

L'opérateur de fragment (dans le cornet en haut à gauche) indique le type de fragment.

Types de fragments : ref, assert, loop, break, alt, opt, neg.

Modèles détaillés UML- Séquence (Sequence) - Fragements

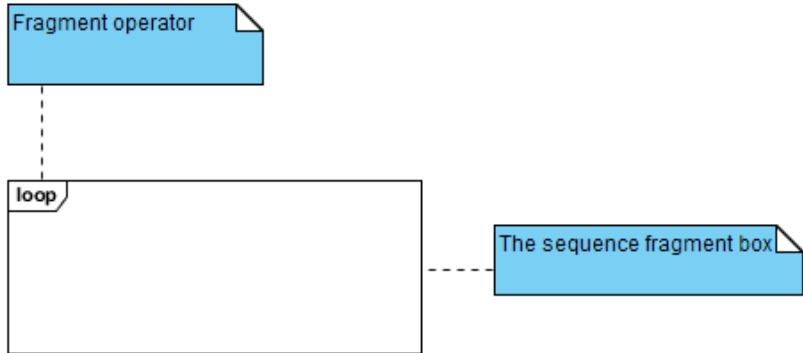


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML- Séquence (Sequence) - Fragments - Types p.1

Opérateur - Type de fragments (partie 1) :

- **alt** - Fragments multiples alternatifs : seul celui dont la condition est vraie s'exécutera.
- **opt** - Facultatif : le fragment s'exécute uniquement si la condition fournie est vraie. Équivalent à un alt seulement avec une trace.
- **par** - Parallèle : chaque fragment est exécuté en parallèle.
- **loop** - Boucle : le fragment peut s'exécuter plusieurs fois, et le garde indique la base de l'itération.

Modèles détaillés UML- Séquence (Sequence) - Fragments - Types p.2

Opérateur - Type de fragments (partie 2) :

- **region** - Région critique : le fragment ne peut avoir qu'un seul thread l'exécutant à la fois.
- **neg** - Négatif : le fragment montre une interaction non valide.
- **ref** - Référence : fait référence à une interaction définie sur un autre diagramme. Le cadre est dessiné pour couvrir les lignes de vie impliquées dans l'interaction. Vous pouvez définir des paramètres et une valeur de retour.
- **sd** - Diagramme de séquence (sequence diagram) : utilisé pour entourer un diagramme de séquence entier.

Modèles détaillés UML- Séquence (Sequence) - Fragments

À noter :

Il est possible de combiner des trames afin de capturer, par exemple, des boucles ou des branches.

Mots-clés de fragment combinés : alt, opt, break, par, seq, strict, neg, critique, ignore, consider, assert et loop.

Les contraintes sont généralement utilisées pour afficher les contraintes de synchronisation sur les messages. Ils peuvent s'appliquer à la synchronisation d'un message ou à des intervalles entre les messages.

Modèles détaillés UML- Séquence (Sequence) - Exemple - Fragments combinés

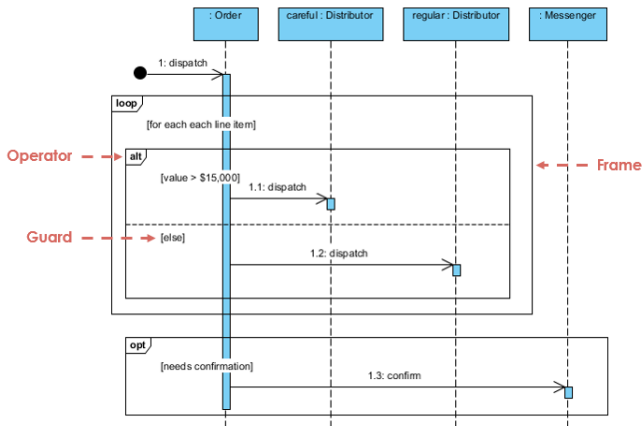


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML- Séquence (Sequence) - Utilisation avec les cas d'utilisations

Les besoins des utilisateurs sont capturés en tant que cas d'utilisation qui sont affinés en scénarios.

Un cas d'utilisation est un ensemble d'interactions entre des acteurs externes et un système. Dans UML, un cas d'utilisation est : la spécification d'une séquence d'actions, y compris des variantes, qu'un système (ou une entité) peut effectuer, en interagissant avec les acteurs du système.

Un scénario est un chemin ou un flux à travers un cas d'utilisation qui décrit une séquence d'événements qui se produit au cours d'une exécution particulière d'un système qui est souvent représenté par un diagramme de séquence.

Modèles détaillés UML- Séquence (Sequence) - Utilisation avec les cas d'utilisations

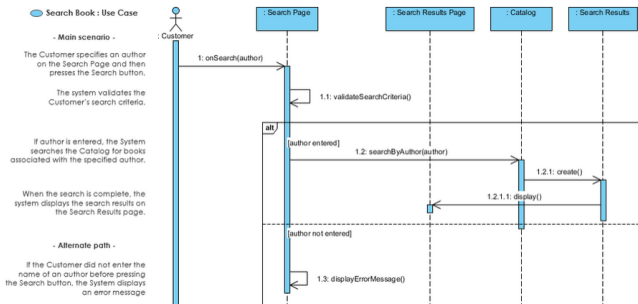


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Activité (Activity)

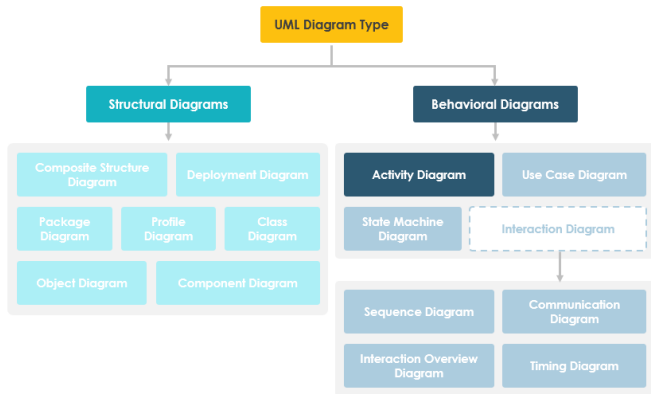


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity)

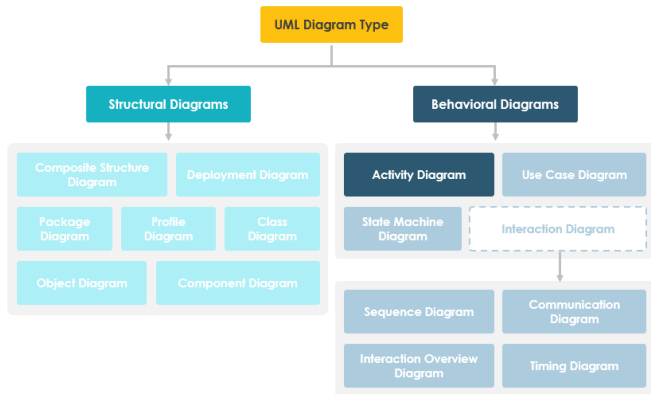


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Activité

- Est utilisé pour représenter un ensemble d'actions



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Action

- Une tâche à accomplir



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Flèche de direction de contrôle

- Affiche la séquence d'exécution



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Flèche de direction objet

- Afficher le flux d'un objet d'une activité (ou d'une action) vers une autre activité (ou action).



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Noeud de départ

- Démontre le début d'un ensemble d'actions ou d'activités



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Noeud final

- À pour but d'arrêter tous les flux de contrôle et les flux d'objets dans une activité (ou une action)



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Noeud objet

- Représente un objet connecté à un ensemble de flux d'objets

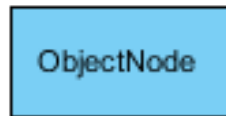


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Noeud de décision

- Représente une condition de test pour s'assurer que le flux de contrôle ou le flux d'objets ne descend que sur un seul chemin



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Noeud de fusion (Décision)

- Rassemble différents chemins de décision qui ont été créés à l'aide d'un nœud de décision.



Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Noeud de séparation (Concurrence)

- Divise le comportement en un ensemble de flux parallèles ou simultanés d'activités (ou d'actions)

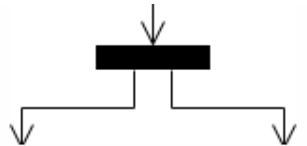


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Noeud de rassemblement (Concurrence)

- Rassemble un ensemble de flux parallèles ou simultanés d'activités (ou d'actions).

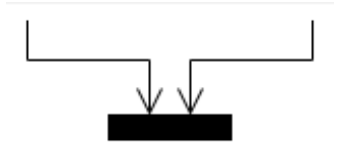


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Items - Partitions (swimlanes)

- Un moyen de regrouper les activités effectuées par le même acteur sur un diagramme d'activité ou de regrouper les activités dans un seul thread.

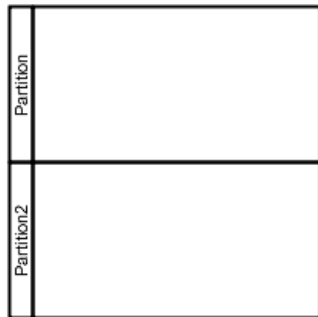


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Modèles détaillés UML - Activité (Activity) - Exemple - Traiteur de commande

Compte tenu de la description du problème liée au flux de travail pour le traitement d'une commande, modélisons la description en représentation visuelle à l'aide d'un diagramme d'activité.
La description est sur la page suivante.

Modèles détaillés UML - Activité (Activity) - Exemple - Traiter l'expédition - Description du problème

Une fois la commande reçue, les activités se divisent en deux ensembles parallèles d'activités.

Un côté remplit et envoie la commande tandis que l'autre s'occupe de la facturation.

Du côté de l'ordre de remplissage, la méthode de livraison est décidée conditionnellement.

Selon la condition, l'activité Livraison de nuit ou l'activité Livraison régulière est effectuée. Enfin, les activités parallèles se combinent pour fermer l'ordre.

Modèles détaillés UML - Activité (Activity) - Résultat graphique

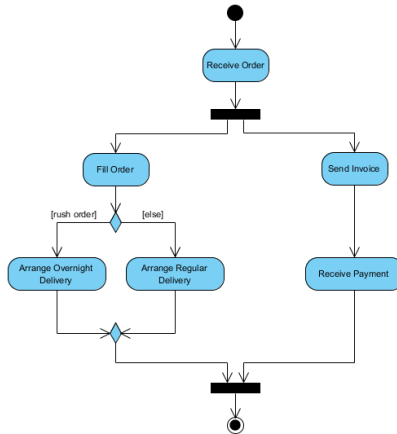


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Activité (Activity) - Exemple - Inscription d'étudiants

Cet exemple de diagramme d'activité UML décrit un processus d'inscription d'étudiants dans une université comme suit :

- Un candidat veut s'inscrire à l'université.
- Le demandeur remet une copie remplie du formulaire d'inscription.
- Le registraire inspecte les formulaires.
- Le registraire détermine que les formulaires ont été remplis correctement.
- Le registraire informe l'étudiant d'assister à la présentation générale de l'université.
- Le registraire aide l'étudiant à s'inscrire à des séminaires
- Le registraire demande à l'étudiant de payer les frais de scolarité initiaux.

Modèles détaillés UML - Activité (Activity) - Exemple - Inscription d'étudiants

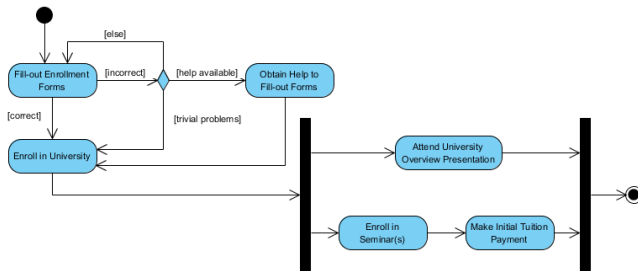


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Activité (Activity) - Partitions (Swimlains)

Une partition est un moyen de regrouper les activités effectuées par le même acteur sur un diagramme d'activité ou un diagramme d'activité ou de regrouper les activités dans un seul fil.

Modèles détaillés UML - Activité (Activity) - Partitions (Swimlains)

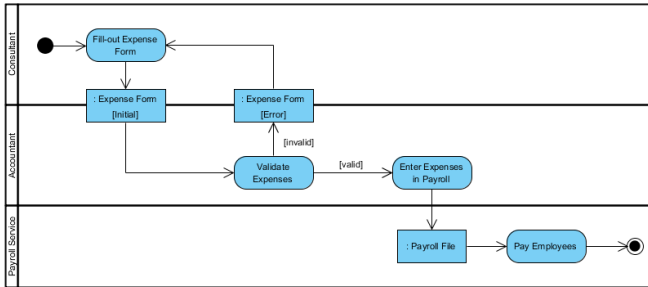


Figure:

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

Modèles détaillés UML - Activité (Activity) - Partitions (Swimlains) - Comparaison normal vs. partition

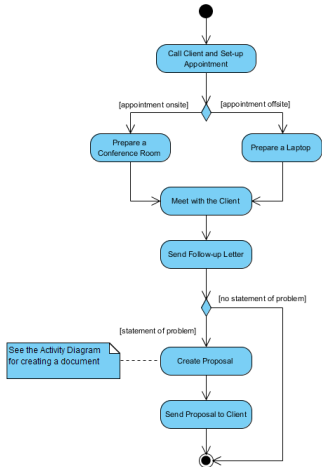


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

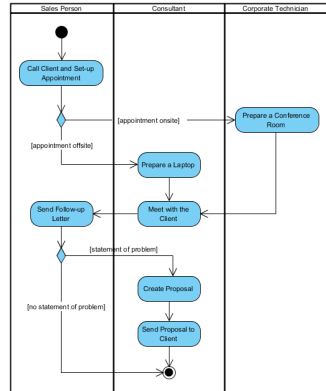


Figure: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>

Bibliographie

- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>
- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>