

# Présentation de la formation

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Présentation du formateur
- Qu'est ce que UML ?
- Le plan de formation
- Les références des ressources logicielles
- Les liens utiles





# Présentation du formateur

---

- Fabien Brissonneau
- Email : fabien.brissonneau@gmail.com
- Consultant Concepteur et Formateur
- Missions d'architecture, de conception , de réalisation logicielles
- Fondateur de eiXa6
- Actuellement en mission sur un projet de gestion
- Mes références :
  - Mon profil Viadeo : <http://fr.viadeo.com/fr/profile/fabien.brissonneau>
  - Mon profil LinkedIn : <http://fr.linkedin.com/pub/fabien-brissonneau/65/902/92a/>



# Qu'est ce que UML?

---

- Langage de modélisation « Unified Modeling Language »
- UML est une notation graphique utilisable pour :
  - Décrire les besoins des utilisateurs sur une application
  - Spécifier un système
  - Concevoir un système informatique
  - Réaliser une documentation
  - Améliorer la communication entre les parties prenantes d'un projet
- UML peut être utilisé de manière plus ou moins formelle, tout à fait compatible avec les démarches agiles. UML est un outil.





# Le plan de formation

---

- **Module 1 : Généralités**

- Chapitre 1 : Les diagrammes UML
  - Chapitre 2 : Mettre en œuvre UML dans le projet

- **Module 2 : UML pour la maîtrise d'ouvrage**

- Chapitre 1 : Les processus métier
  - Chapitre 2 : Les objets métier
  - Chapitre 3 : L'expression de besoins

- **Module 3 : UML en analyse**

- Chapitre 1 : Découvrir les classes
  - Chapitre 2 : Les états des objets

- **Module 4 : UML en conception**

- Chapitre 1 : Construire le modèle de conception
  - Chapitre 2 : Passer de l'analyse à la conception





# Les références des ressources logicielles

---

- Site de VP
  - <http://www.visual-paradigm.com/>
- Site de Astah
  - <http://astah.net>
- Site de Modelio
  - <http://www.modeliosoft.com/fr.html>



# Les liens utiles

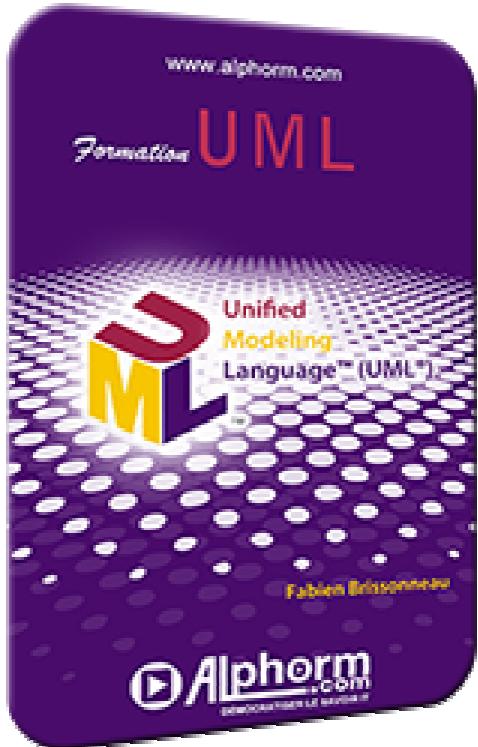
---

- Le site de l'OMG
  - <http://www.uml.org/>
- Les experts
  - Pascal Roques
    - <http://www.dotnetguru2.org/proques/index.php>

 Il n'y a plus qu'à ...

---

GO



Généralités

## Les diagrammes UML

# Les diagrammes UML

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau

Consultant, concepteur et formateur

Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Origines et généralités sur UML
- Les différents diagrammes UML





# Origine de UML et objectifs

---

- Unified Modeling Language
- Langage graphique de modélisation
- Fusion des méthodologies de
  - Booch
  - Rumbaugh
  - Jacobson
- Société Rational, supporté par Rose



# Généralités sur l'utilisation de UML

---

- Langage, non lié à un processus
- 14 diagrammes qui sont des vues partielles du modèle à décrire
- Objectif de lisibilité
- Applicable du métier jusqu'au code
- Certains diagrammes utilisent l'approche orientée objets



# Les diagrammes

---

- Diagramme de cas d'utilisation
- Diagramme de séquence
- Diagramme de communication
- Diagramme d'états
- Diagramme d'activités
- Diagramme de vue d'ensemble des interactions
- Diagramme de temps
- Diagramme de classes
- Diagramme de packages
- Diagramme de composants
- Diagramme de déploiement
- Diagramme d'objets
- Diagramme de structures composites
- Diagramme de profils



# Le diagramme de classes

---

- Montre les types d'objets à utiliser
- Proche du MCD
- Diagramme très courant
- Diagramme de structure, et statique, il ne peut rien dire sur la dynamique du système



# Le diagramme d'objets

---

- Instance du diagramme de classes
- Sert à montrer un exemple de la structure, en particulier lorsque le diagramme de classe est interprétable
- Peu courant, car rapidement complexe et difficile à rendre exhaustif



# Le diagramme de packages

---

- Un package est un regroupement de n'importe quel élément UML
- Ce diagramme montre les relations entre packages du système étudié
- A noter que les dépendances entre packages peuvent être étudiées par des indicateurs



# Le diagramme de composants

---

- Un composant est un élément offrant des services et nécessitant d'autres composants
- Il fournit les services via des interfaces et requiert d'autres interfaces
- Peut être décomposé
- Utile lors d'une rétro-conception



# Le diagramme de déploiement

---

- Montre les nœuds physiques d'un déploiement : machines, réseaux, ....
- Représentation graphique qui profite des stéréotypes
- Description des besoins ou expression d'un existant



# Le diagramme de structures composites

---

- Représente des compositions complexes de façon non ambiguë
- Introduit les concepts de classes structurées, de *parts*, de *connectors*



# Le diagramme de profils

---

- Représente les types et stéréotypes disponibles pour la réalisation d'un modèle



# Le diagramme de cas d'utilisation

---

- Représentation du comportement du système sous forme de cas d'utilisation
- Diagramme statique, sert de point d'entrée à l'étude de besoins



# Le diagramme d'activités

---

- Représente les enchaînements nécessaires à la réalisation d'un processus
- Utile dans la description des processus métier
- Utile dans la description des cas d'utilisation
- Facile à introduire, simple et efficace



# Le diagramme d'états

---

- Représente le cycle de vie d'un objet
- Utile pour la description d'un objet métier
- Trouve une traduction aussi dans le code
- Le contexte peut être complexe



# Le diagramme de séquence

---

- Représente les interactions entre entités au cours du temps
- Simple et facile à introduire
- Utile à quasiment tous les niveaux du métier au code



# Le diagramme de communication

---

- Représente ce que le diagramme de séquence sait représenter
- Disposition spatiale des entités
- Assez riche, donc rapidement touffu
- Souvent utilisé sous le nom de diagramme de contexte dynamique



# Le diagramme de temps

---

- Représente des contraintes de temps dans les interactions entre objets
- Diagramme complexe, qui mélange les états des objets et les messages entre ces mêmes objets



# La vue d'ensemble des interactions

---

- Représente les séquences entre les différentes interactions de l'étude
- Plutôt utilisé comme un outil de synthèse de plusieurs interactions
- Possibilités de présentations expressives

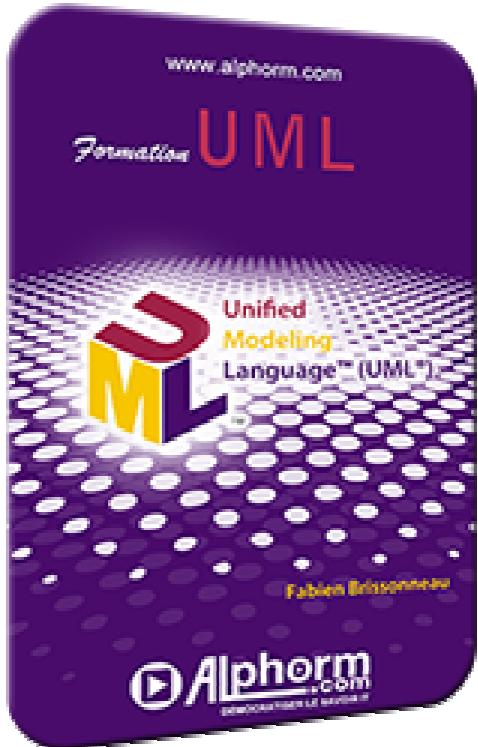


# Ce qu'on a couvert

---

- UML est un langage graphique, mais non une démarche
- Il y a 14 diagrammes différents dans UML
  - 7 diagrammes de structure
  - 7 diagrammes de comportement
- Certains diagrammes utilisent l'approche orientée objets





Généralités

## Les diagrammes UML

# L'approche orientée objets

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau

Consultant, concepteur et formateur

Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Rappels sur l'approche orientée objets
- Pourquoi ?
- Avantages attendus ?
- Vocabulaire





# Pourquoi utiliser cette approche ?

---

- Définition naturelle de composants.
- Meilleure modularité de l'application.
- Cohérence du code (données/traitements).
- Mécanismes d'extension (héritage).
- Séparation des interfaces de l'implémentation



# Analyse/Conception/Codage

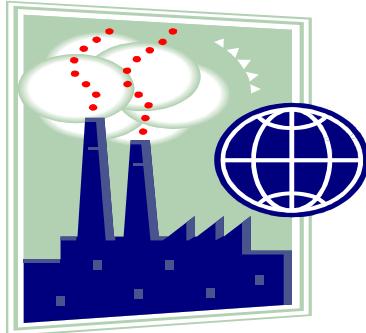
---

- Programmation
  - Utiliser un langage orienté objets
- Conception
  - Mettre en place une solution logicielle utilisant les concepts et suivant les principes
- Analyse
  - Décrire le problème avec des classes
- En utilisant le paradigme « orienté objets », problème et solution sont liés



# Objet et classe

---



- **Classe**
  - Type d'objets
  - Descripteur (usine) d'objets

- **Objet**
  - Vivant
  - Instance de classe
  - Connait sa classe





# Avantages attendus

---

- Lisibilité de chaque classe
- Sécurité du code
- Réutilisabilité du code de chaque classe
- Réutilisabilité de groupes de classes



# Lisibilité de chaque classe

---

- Le code est cohérent car les traitements et les données sont liés par la proximité.
- L'intention est plus claire.
- L'objectif de la classe est mieux compris grâce à la définition des services.



# Sécurité du code

---

- Les données sont encapsulées et visibles uniquement des traitements autorisés.
- L'état de l'objet est mieux maîtrisé.



# Réutilisabilité d'une classe

---

- La classe est définie indépendamment du contexte.
- Replacer la classe dans un contexte différent est donc simple.



# Réutilisabilité d'un groupe de classes

---

- Un groupe de classes est réutilisable s'il est bien construit, avec une forte cohérence interne.
- La réutilisation passe soit par l'héritage, soit par l'implémentation d'interfaces.



# Quelques concepts simples

---

- Classification
- Encapsulation
- Abstraction
- Méthodes/Surcharge
- Messages
- Héritage/Redéfinition
- Polymorphisme

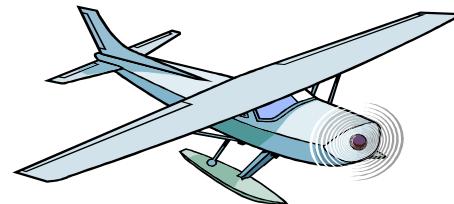


# La classification

---



<b>Voiture</b>
- vitesse: int
+ demarrer() : void + freiner() : void

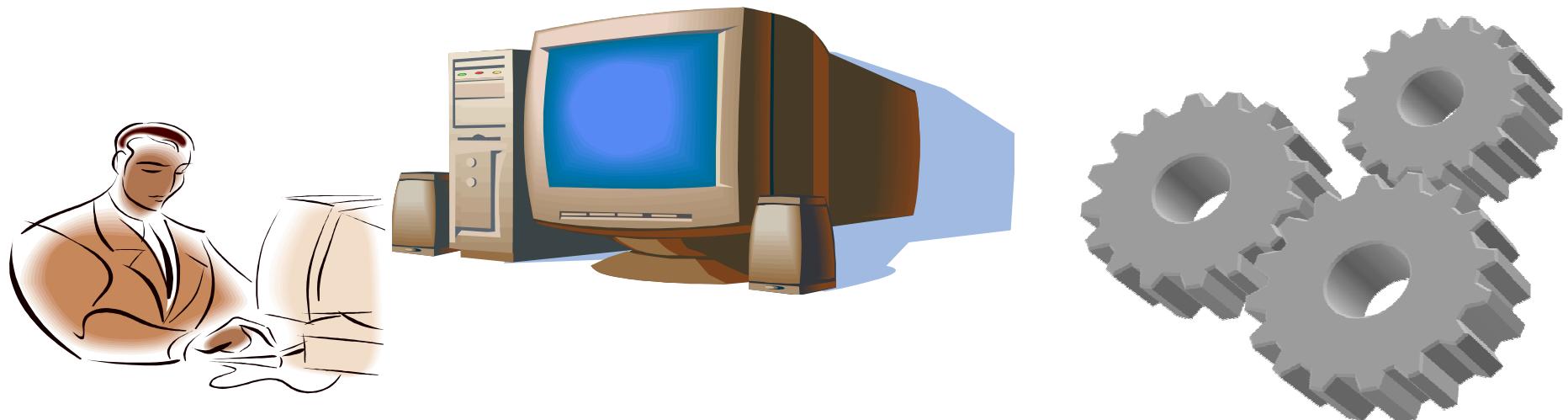


<b>Avion</b>
- vitesse: int - altitude: int
+ decoller() : void + atterrir() : void



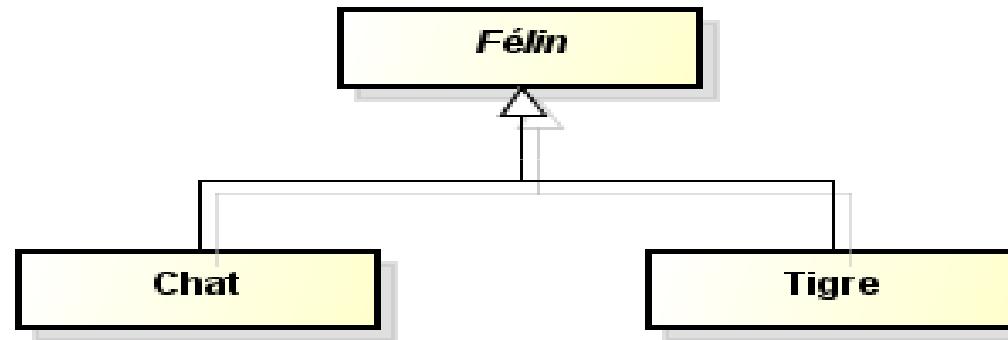
# L'encapsulation

---



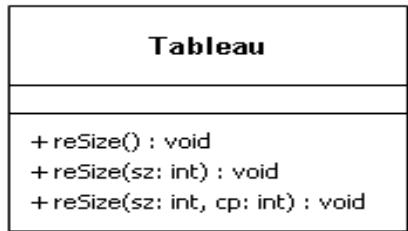


# L'abstraction





# Les méthodes et la surcharge



**Tableau t = new Tableau();**

**t.reSize();**

**t.reSize(5);**



# Les messages 1/2

---

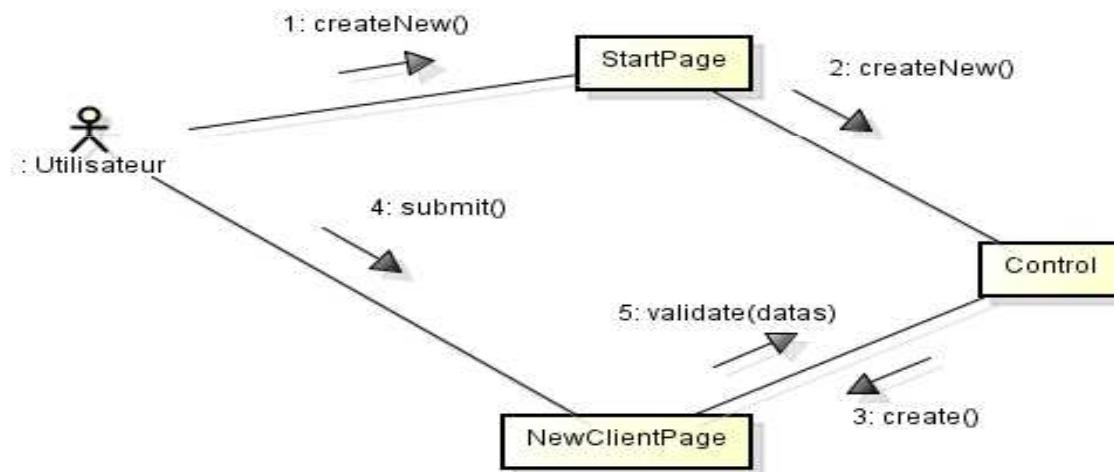
- Un objet émetteur envoi des messages à un récepteur
- Traduit la dynamique du système





# Les messages 2/2

- Les objets s'échangent des messages





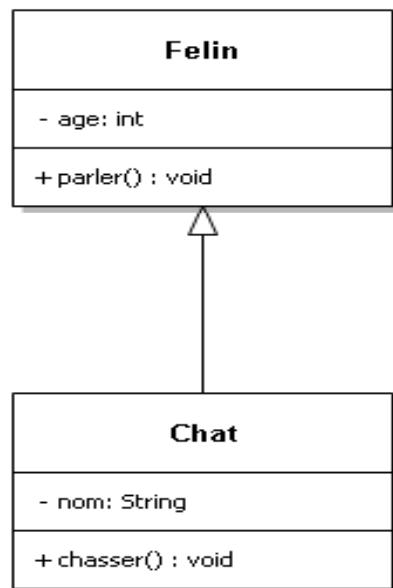
# Objectifs

---

- Rappels sur la programmation orientée objets
- Usage et limites de l'héritage
- Rappels sur UML
- Utiliser UML



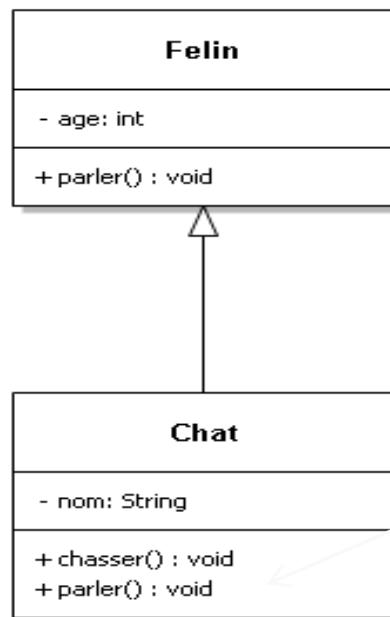
# L'héritage



```
Felin minou = new Chat();
minou.parler();
minou.chasser();
```



# La redéfinition des méthodes



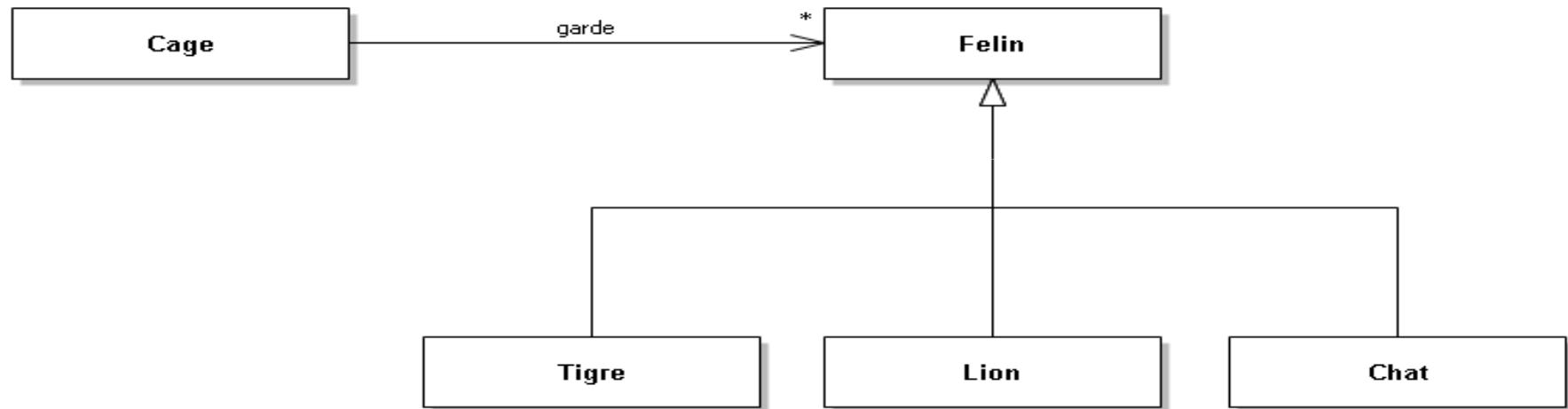
**La méthode parler est redéfinie.**

**Le prototype est le même.**

**La définition est différente.**

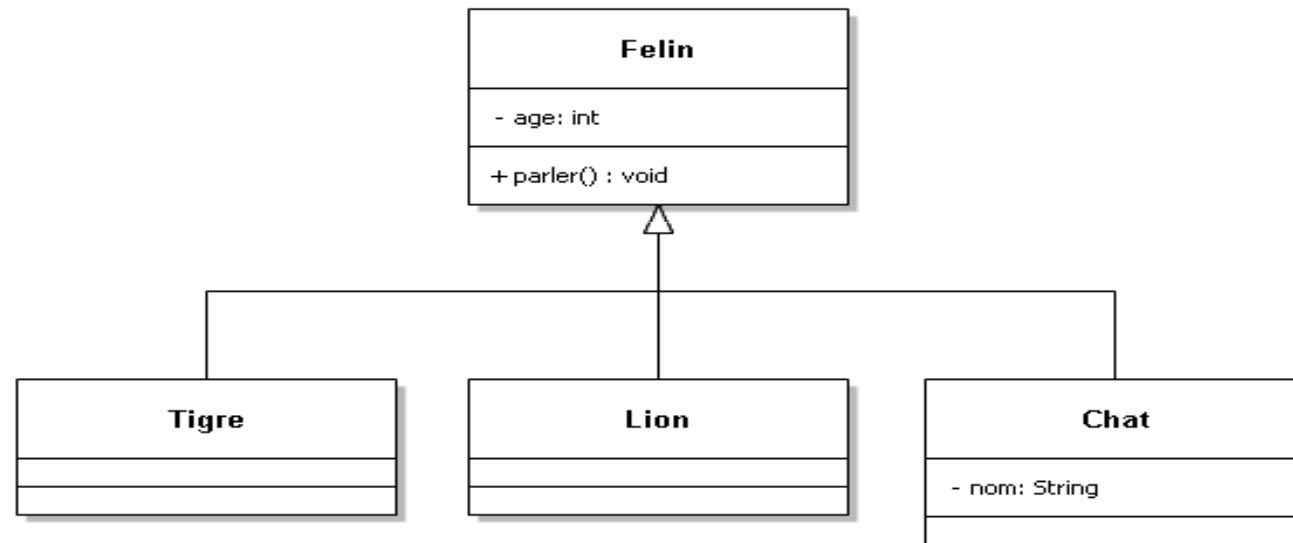


# objets



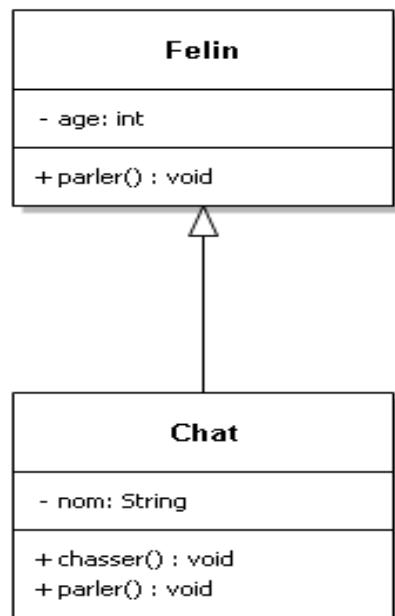


# double





# comportement.



**Héritage + redéfinition**

**Le comportement du Chat est spécialisé**



# La validité d'un héritage

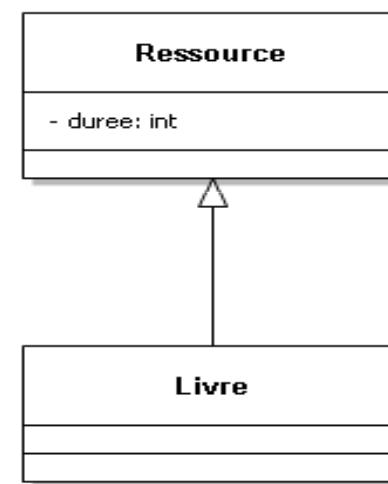
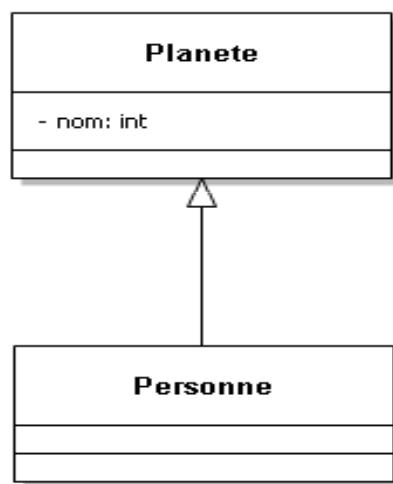
---

Un héritage est une relation EST-UN

Un héritage est une relation à 100%



# EST-UN et 100%



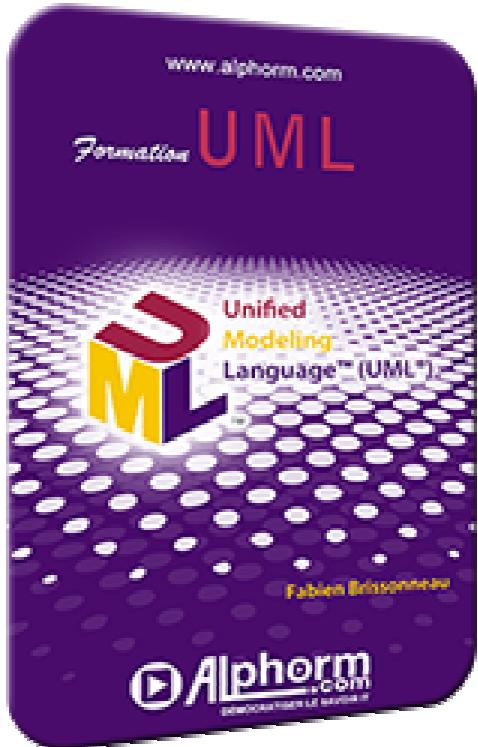


# Ce qu'on a couvert

---

- UML est un langage graphique, mais non une démarche
- Il y a 14 diagrammes différents dans UML
- Certains diagrammes utilisent l'approche orientée objets





Généralités

## Les diagrammes UML

# Le diagramme de classes

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau

Consultant, concepteur et formateur

Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Usages du diagramme de classes
- Syntaxe de base
- Syntaxe avancée





# Usages du diagramme de classes

---

- Diagramme de structure, statique
- Représente les types d'objets utilisés
- Appliqué aux :
  - Objets métiers
  - Objets d'analyse
  - Objets de conception, logiciels



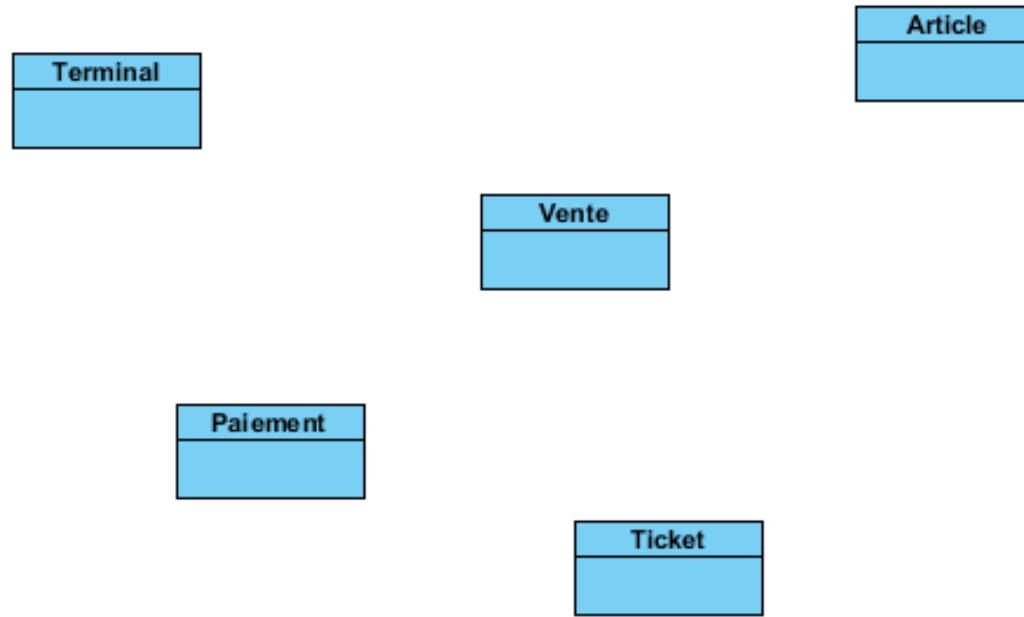
# La syntaxe de base

---

- Classe, attribut, méthode
- Association, multiplicité
- Généralisation-spécialisation
- Classe abstraite
- Interface et implémentation

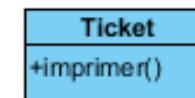
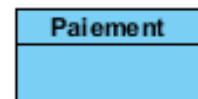
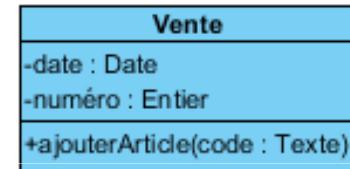
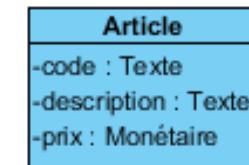
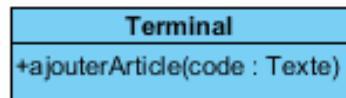


# Classes



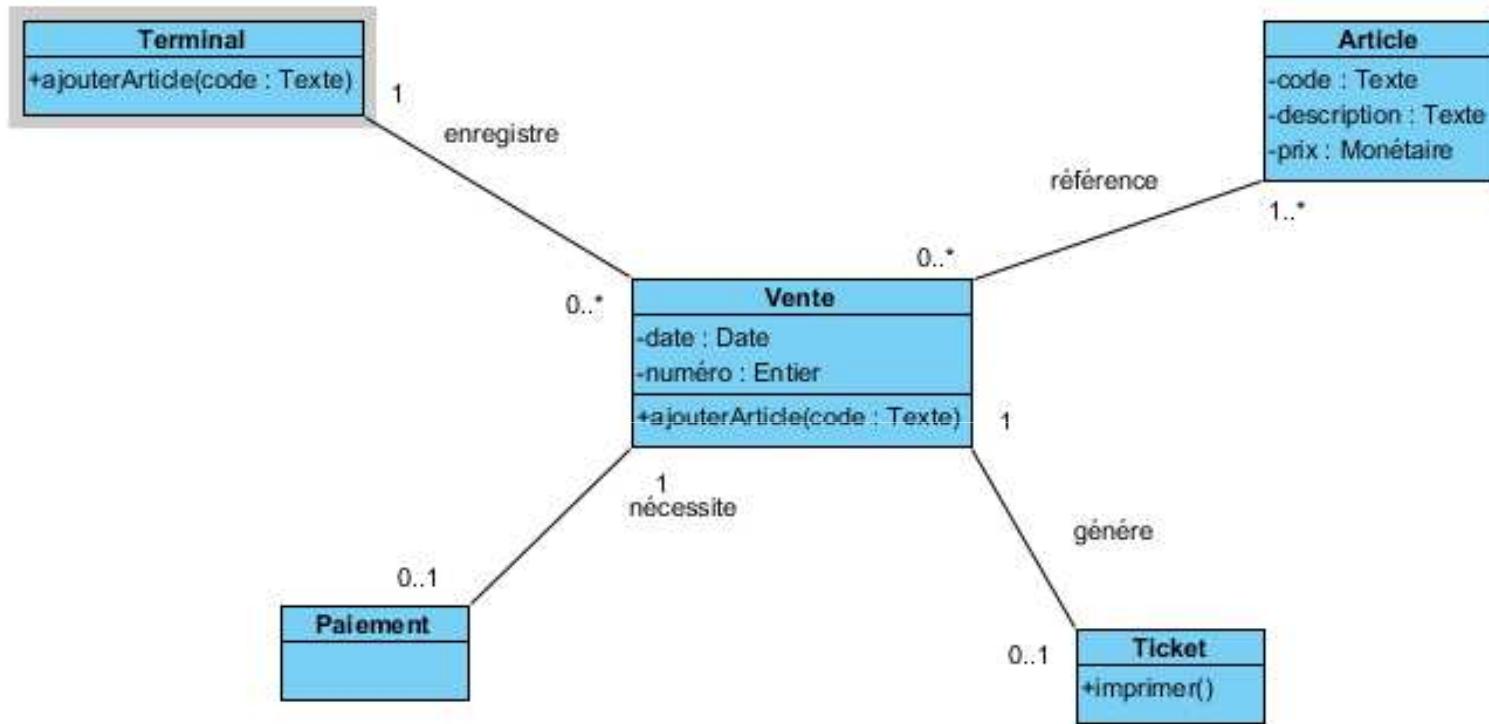


# Classes, attributs, méthodes, visibilités



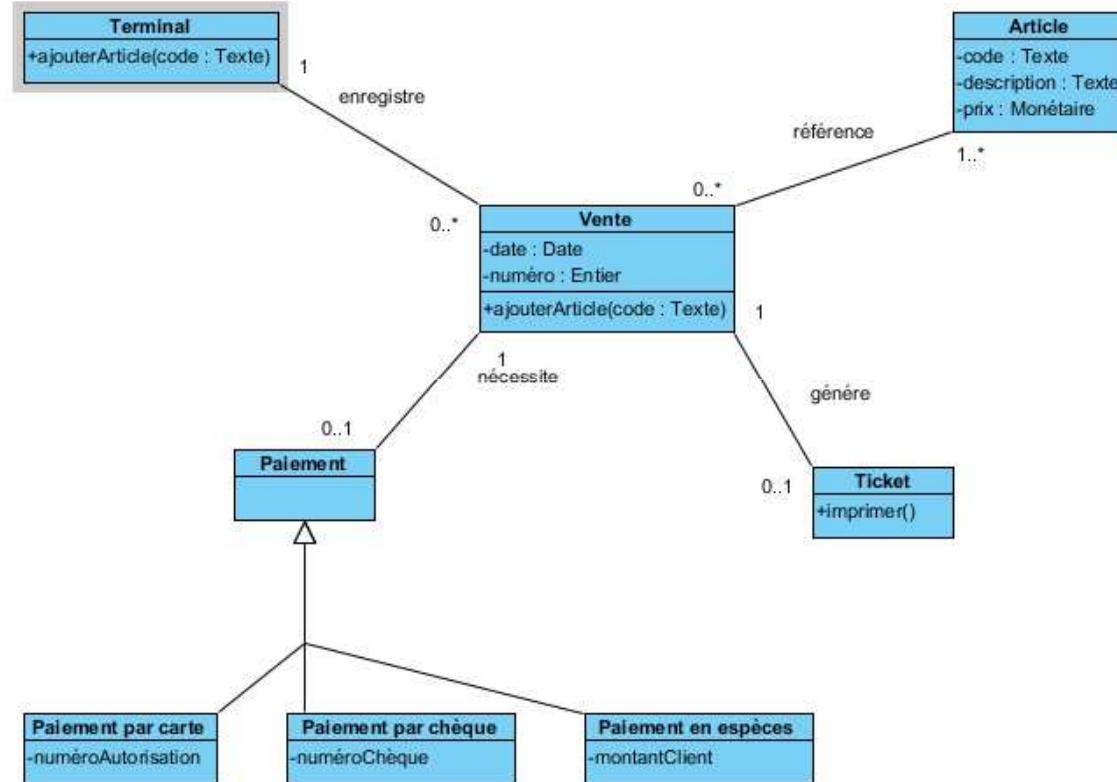


# Association, multiplicités





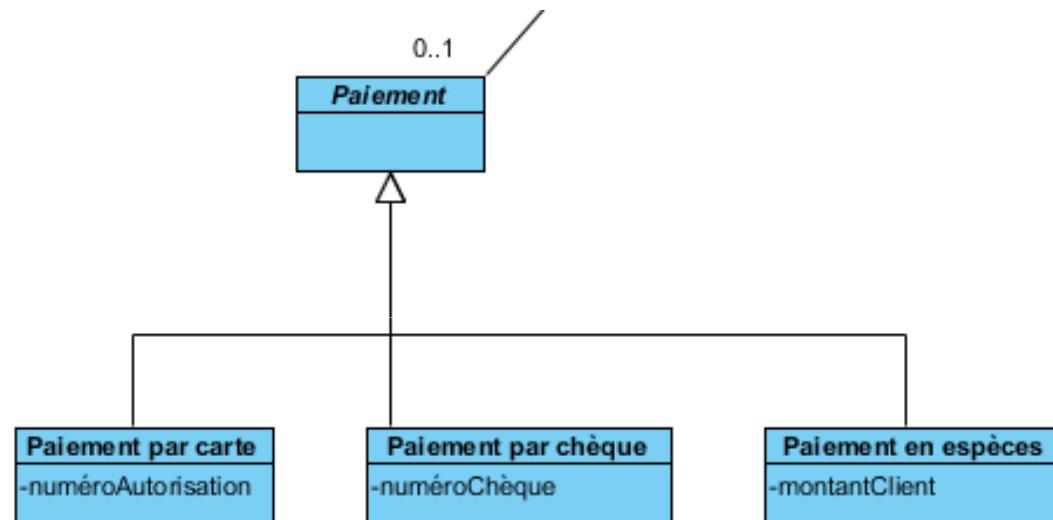
# Généralisation-Spécialisation





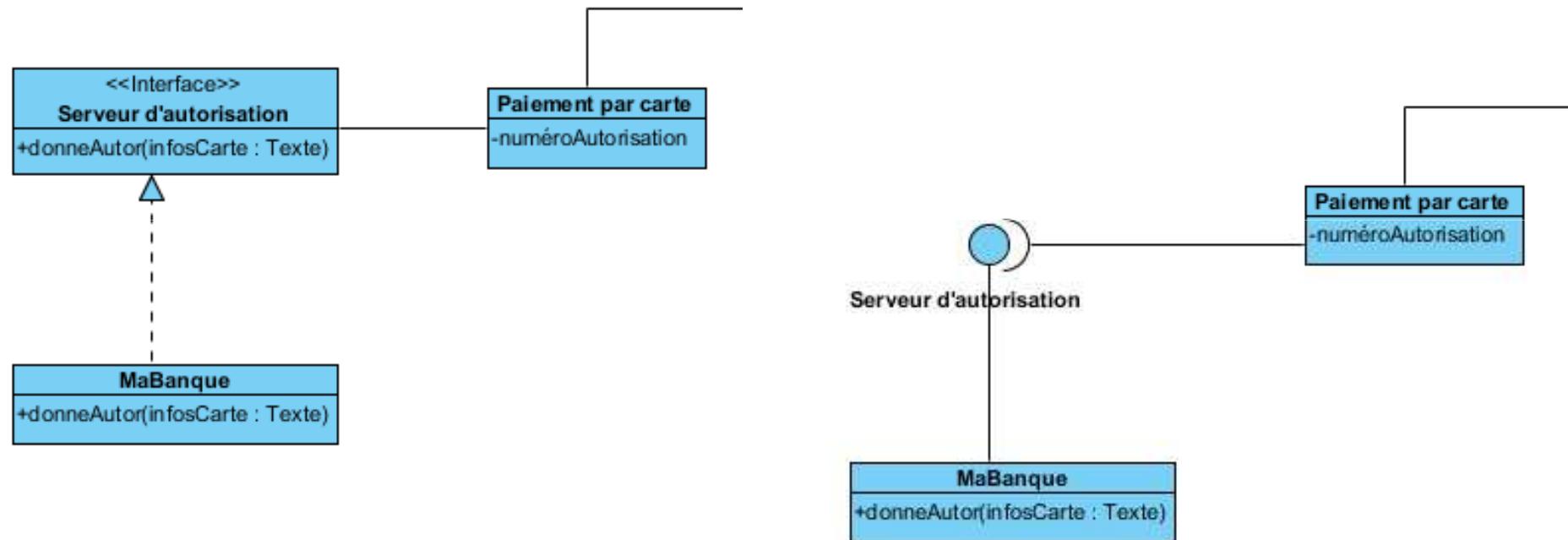
# Classe abstraite

---





# Interface et implémentation





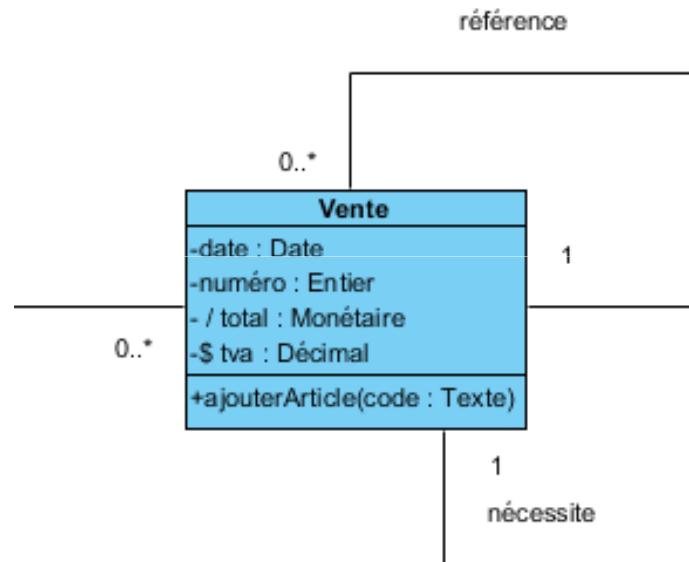
# Syntaxe avancée

---

- Attributs dérivés et constants
- Attributs et méthodes statiques
- Restriction de navigabilité et noms de rôles
- Dépendances contextuelles
- Classe d'association
- Compositions et agrégations
- Notes et contraintes

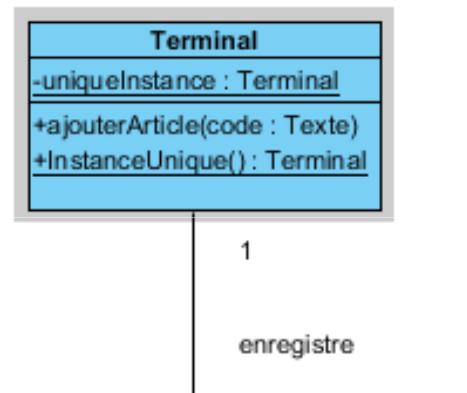


# Attributs dérivés et constants



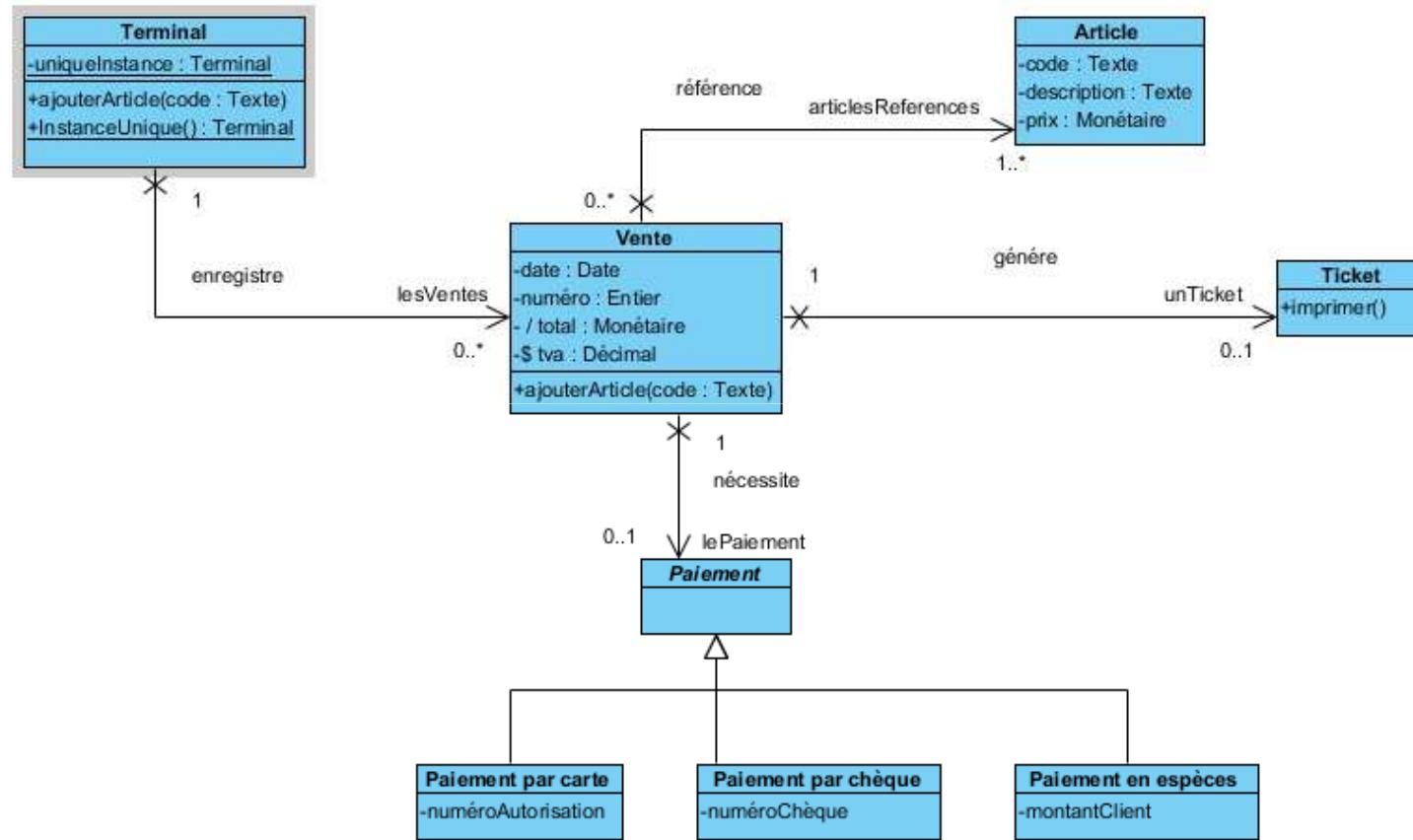


# Attributs et méthodes statiques



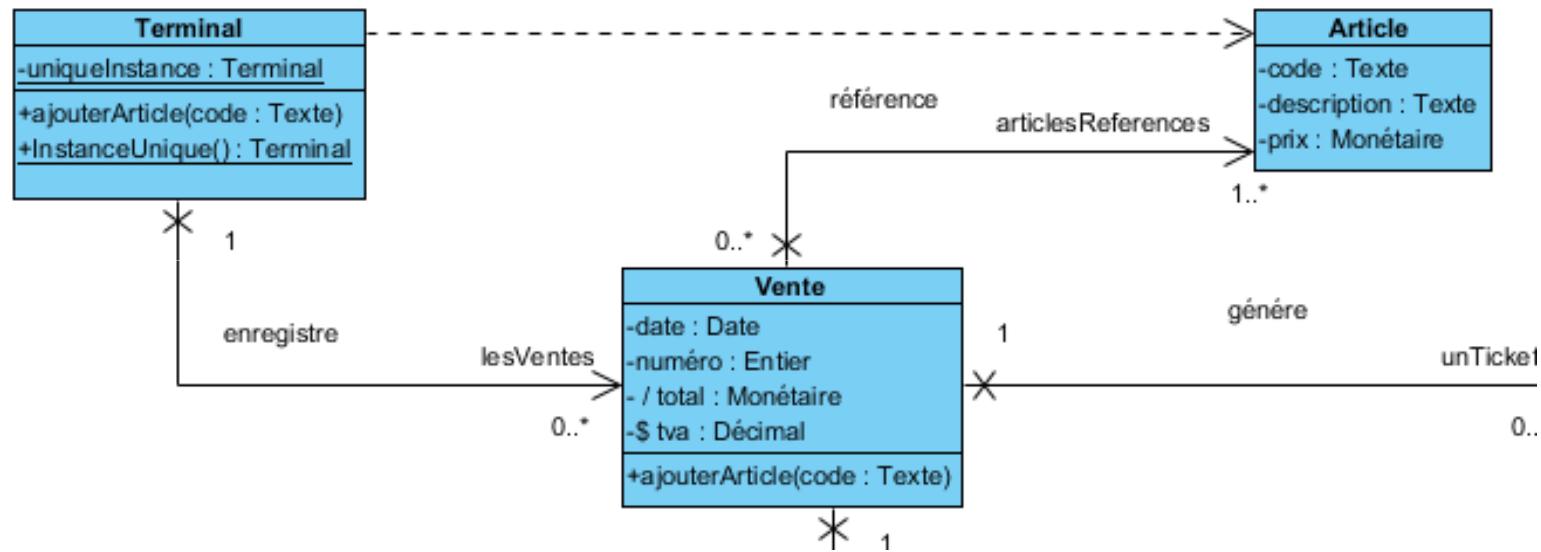


# Restriction de navigabilité et noms de rôles



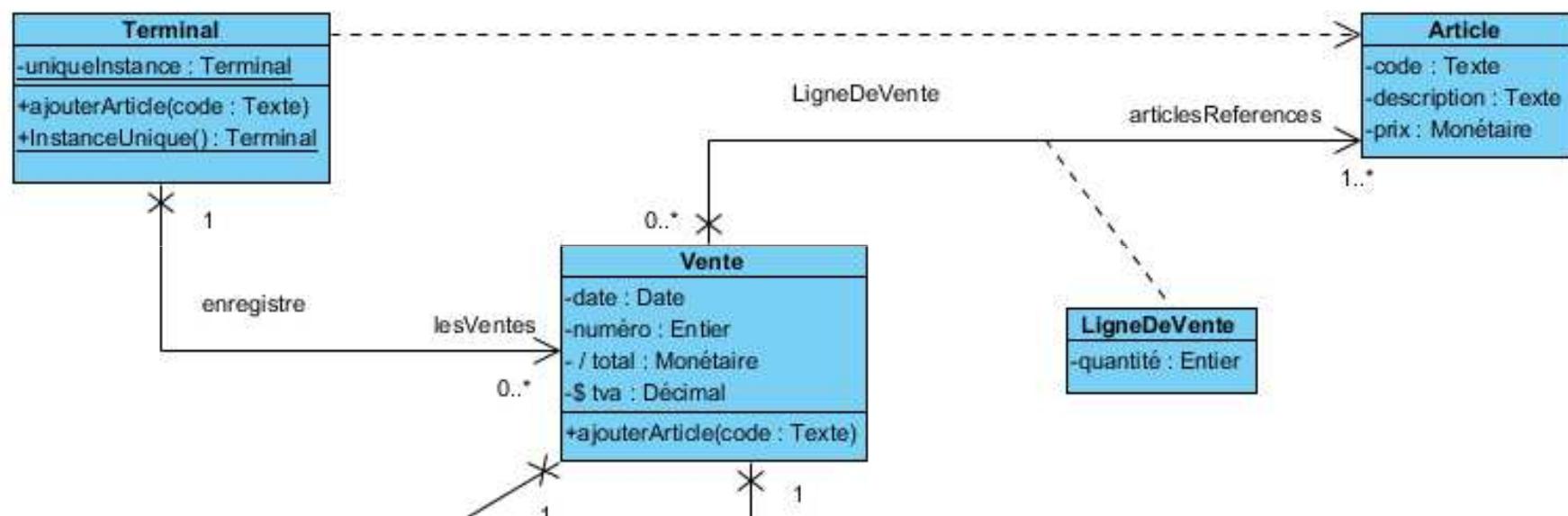


# Dépendances contextuelles



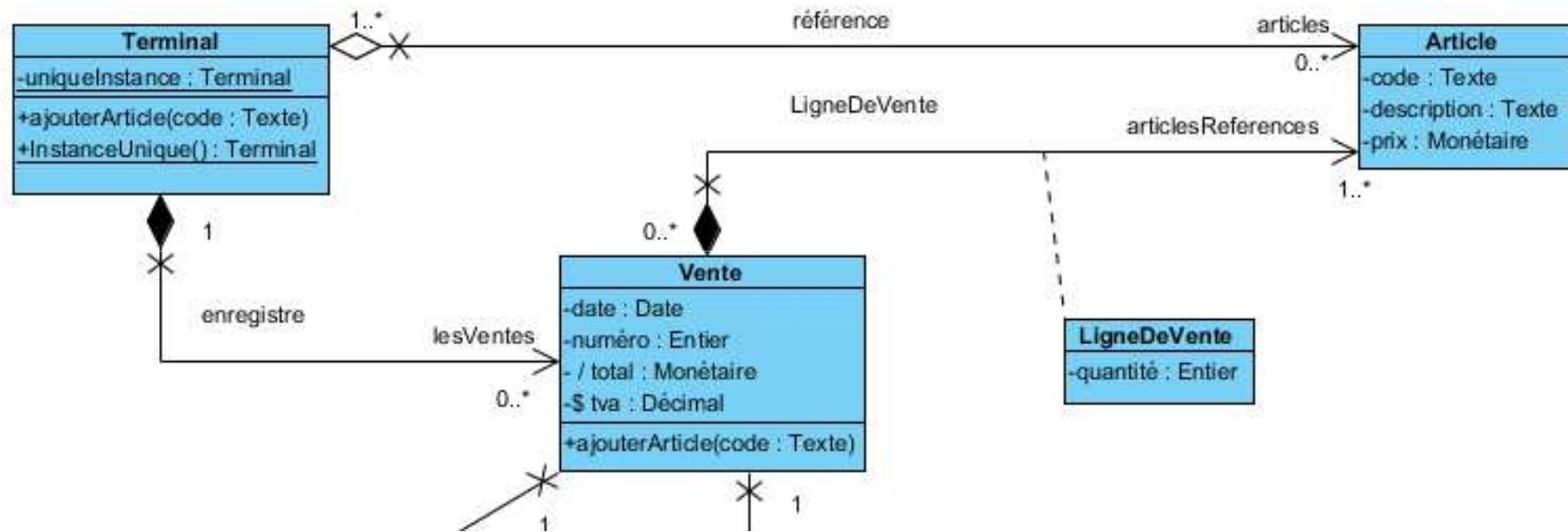


# Classe d'association

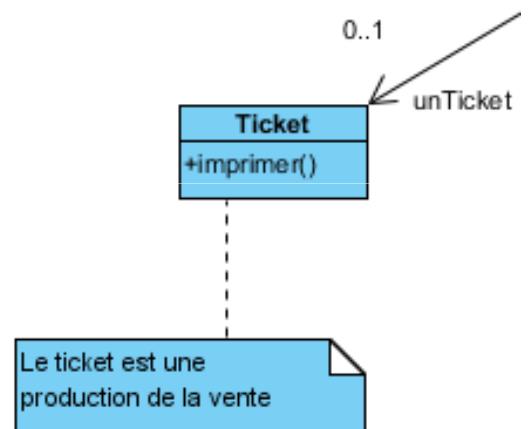




# Composition et agrégation



# Notes et contraintes

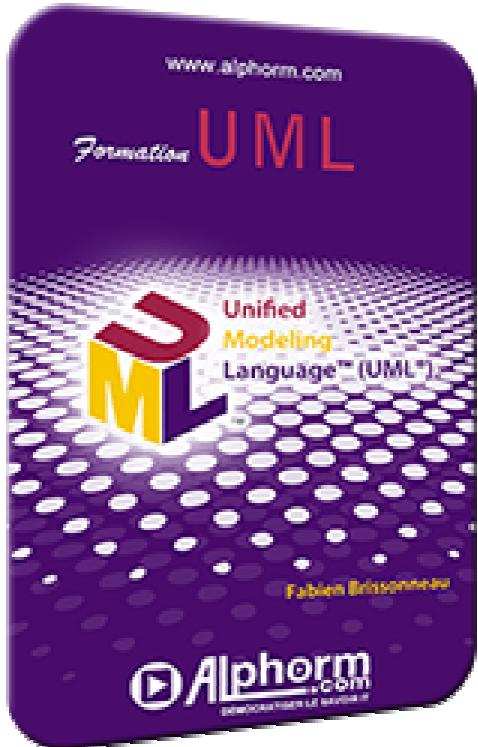


# Ce qu'on a couvert

---

- Les usages du diagramme de classes
- La syntaxe de base : classe, attribut, méthode, association, multiplicité, généralisation-spécialisation, classe abstraite, interface et implémentation
- La syntaxe plus avancée : attributs dérivés et constants, attributs et méthodes statiques, restriction de navigabilité et noms de rôles, dépendances contextuelles, classe d'association, compositions et agrégations, notes et contraintes





Généralités

## Les diagrammes UML

# Le diagramme de séquences

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau

Consultant, concepteur et formateur

Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Usages du diagramme de séquences
- Syntaxe de base
- Syntaxe avancée





# Usages du diagramme de séquences

---

- Diagramme de comportement, dynamique
- Un des diagrammes d'interaction entre les objets, il met en valeur les échanges de messages dans le temps
- Il peut montrer :
  - Les interactions entre acteurs métiers
  - Les interactions entre acteurs du système et le système
  - Les interactions entre objets d'analyse
  - Les interactions entre objets logiciels



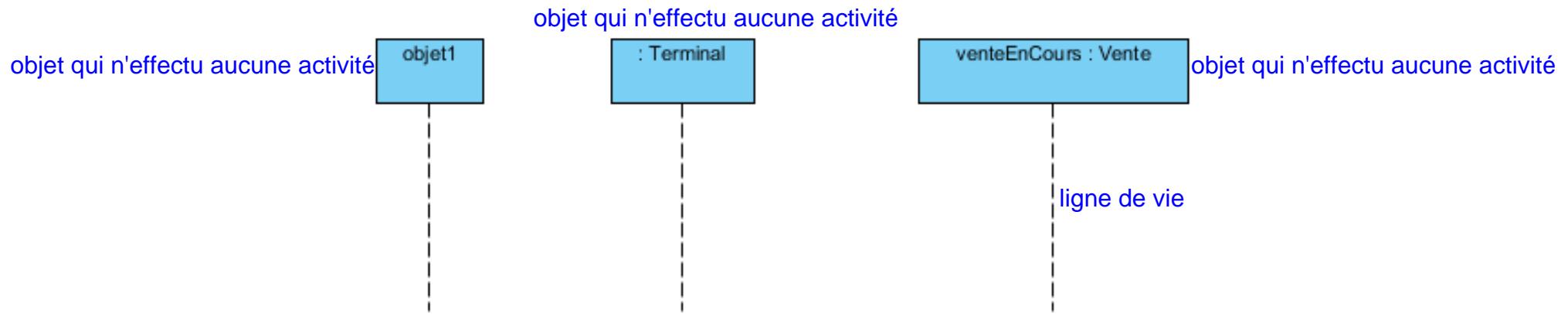
# La syntaxe de base

---

- Objet, ligne de vie, focus de contrôle
- Messages de synchrones, asynchrones
- Messages de création, de destruction

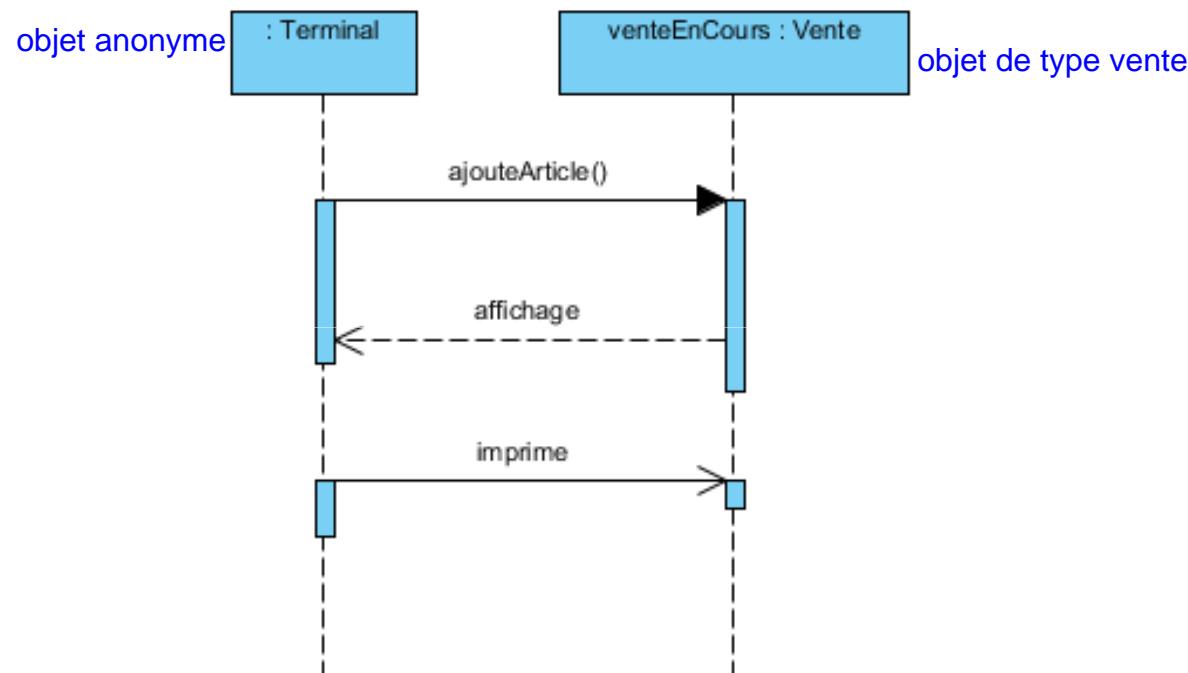


# Lignes de vie



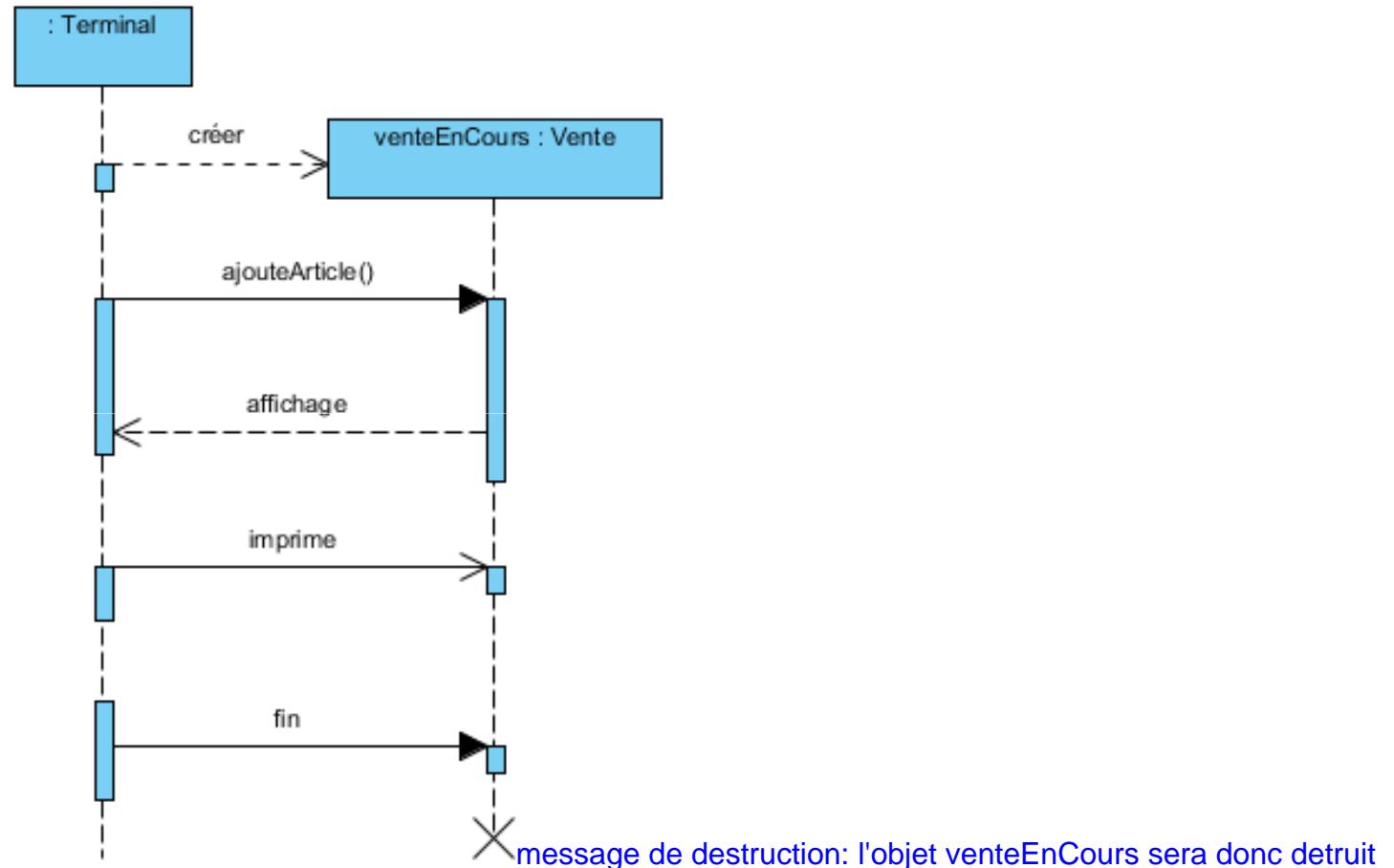


# Messages





# Messages de création, de destruction





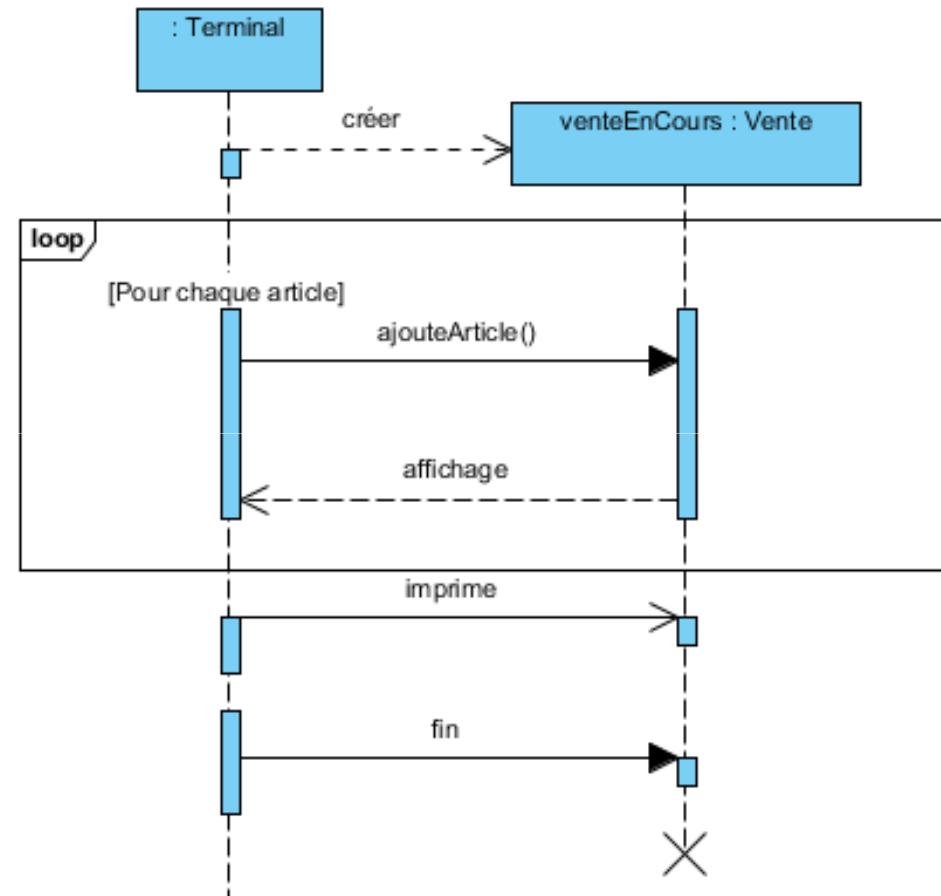
# Syntaxe avancée

---

- Les cadres LOOP, ALT, REF, OPT
- Auto-messages, messages récursifs

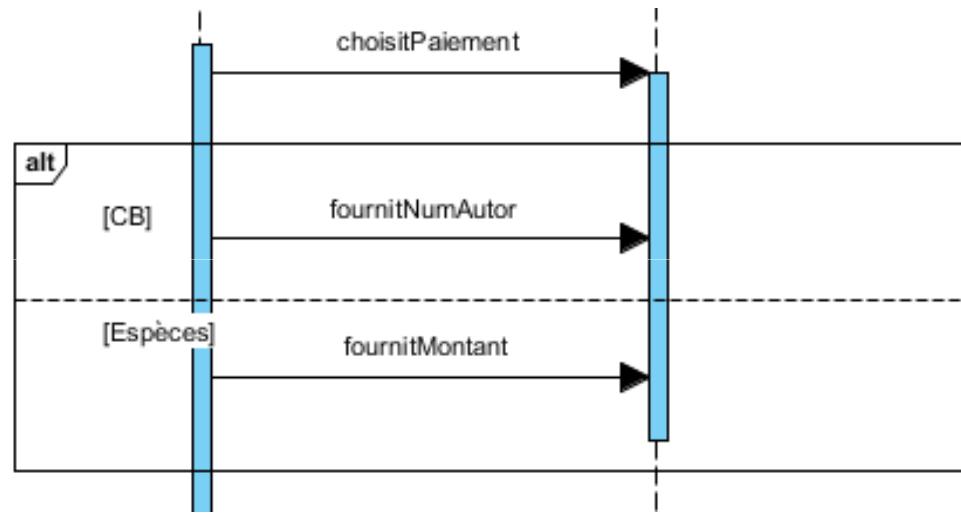


# Cadre LOOP



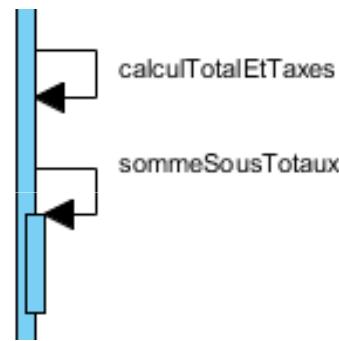


# Cadre ALT





# Auto-messages et messages récursifs



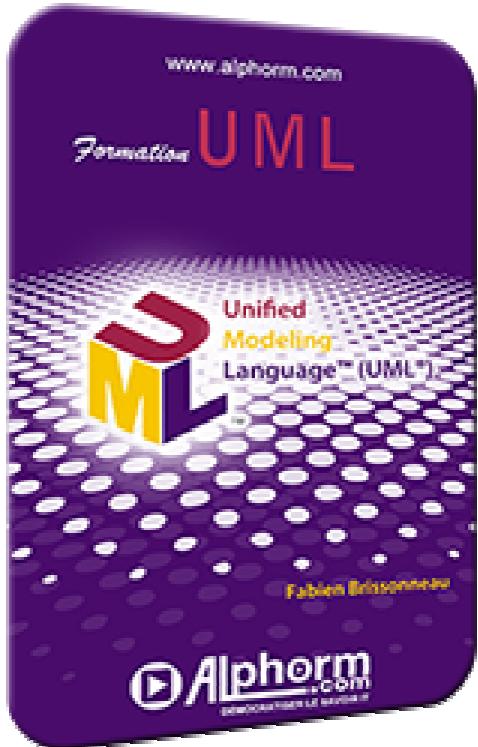


# Ce qu'on a couvert

---

- Les usages du diagramme de séquences
- La syntaxe de base : ligne de vie, focus de contrôle, les retours , messages, synchrones, asynchrones, création et destruction
- La syntaxe plus avancée : les cadres (LOOP et ALT), les auto-messages et messages récursifs





Généralités

## Les diagrammes UML

Le diagramme de cas  
d'utilisation

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

---

- Usages du diagramme de cas d'utilisation
- Syntaxe de base
- Syntaxe avancée





# Usages du diagramme de cas d'utilisation

---

- Diagramme de comportement, statique
- Le diagramme dédié à la description statique des processus
- Ne montre aucune dynamique, liste les fonctionnalités du système



# La syntaxe de base

---

- Acteur, humain ou machine
- Cas d'utilisation
- Association
- Frontière du système



# Définition des cas d'utilisation

---

- Un cas d'utilisation est une fonctionnalité du système, indépendante a priori des autres cas d'utilisation
- UML n'a pas inventé les cas d'utilisation
- Un acteur est un rôle joué par une entité externe au système, en interaction directe avec le système
- Le diagramme des cas d'utilisation est une vue de haut niveau de l'étude de besoins



# Acteur

---

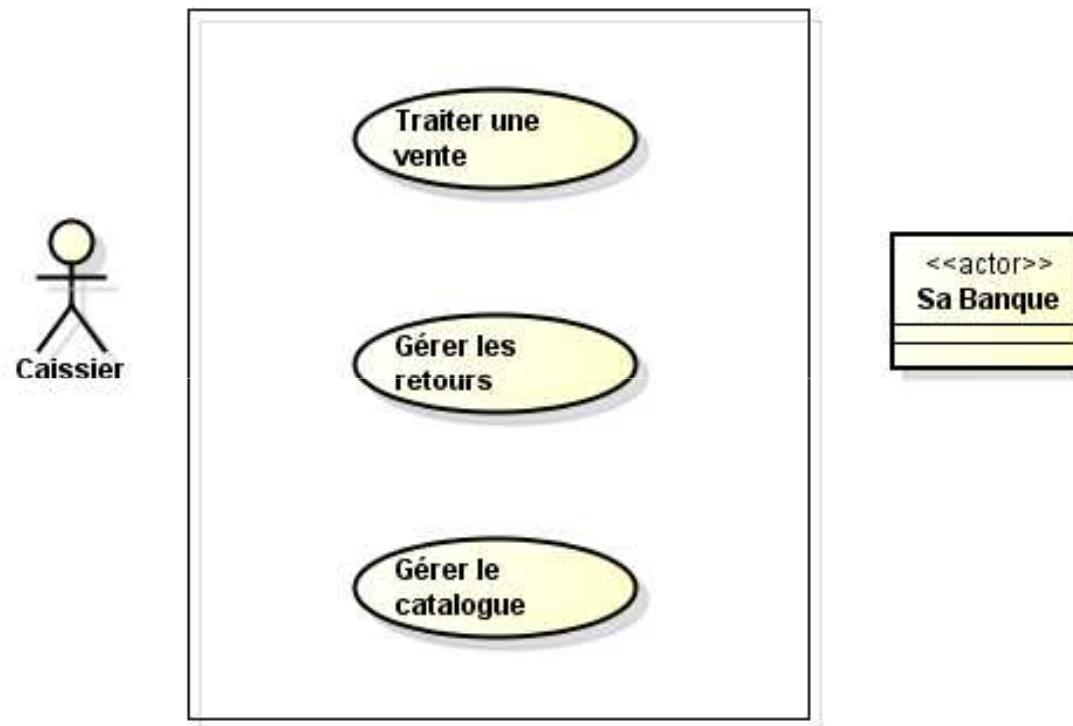


Caissier



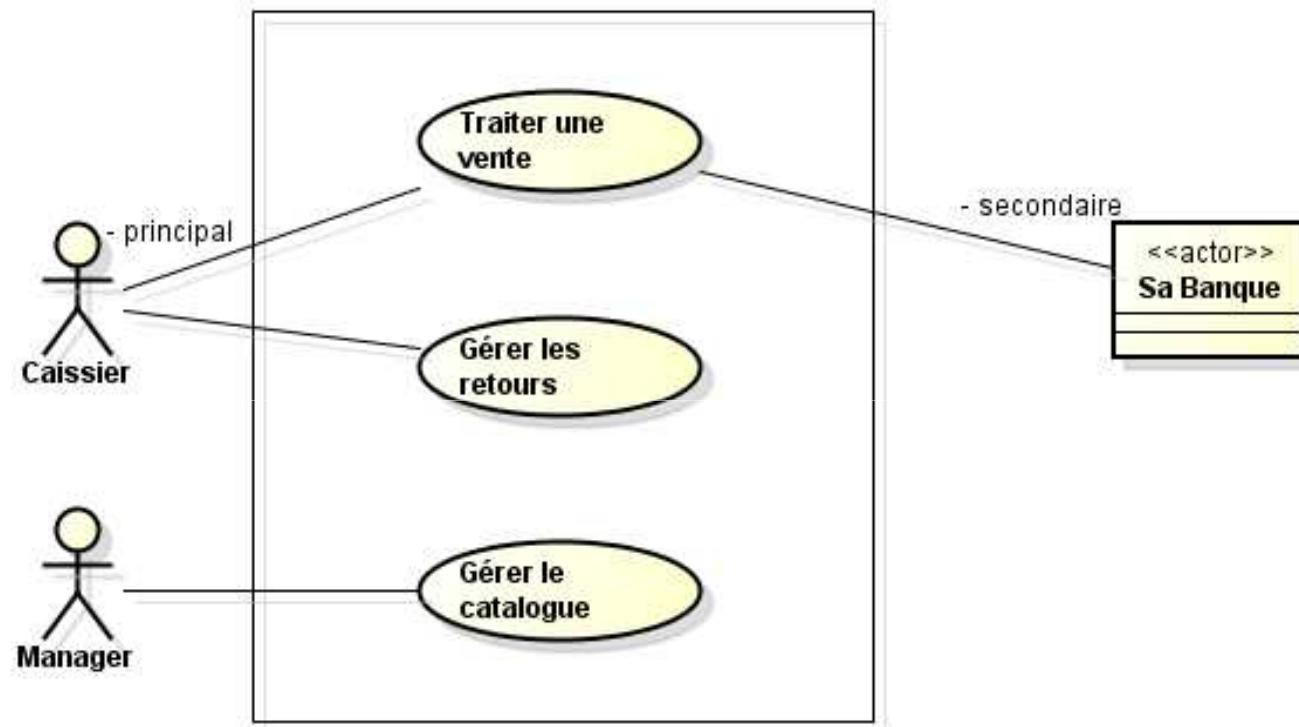


# Cas d'utilisation





# Association





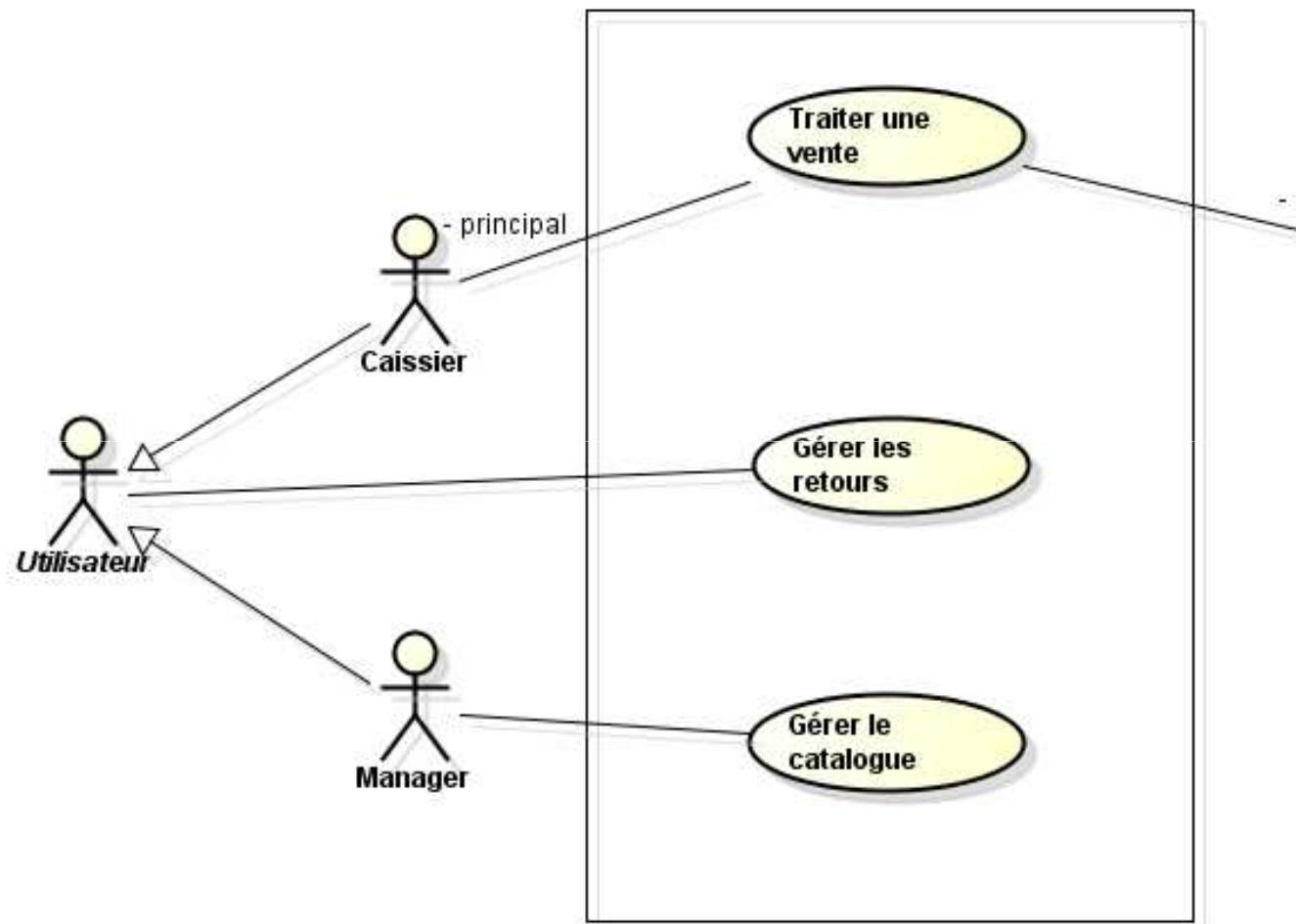
# La syntaxe avancée

---

- Acteur abstrait
- Cas d'utilisation abstrait, généralisation
- Inclusion, extension



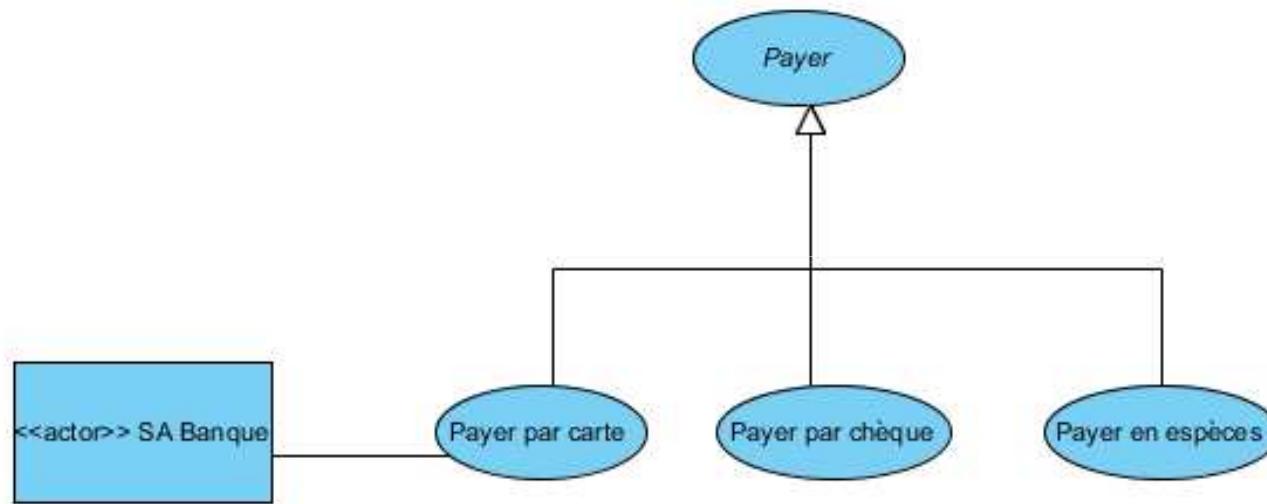
# Acteur abstrait, généralisation



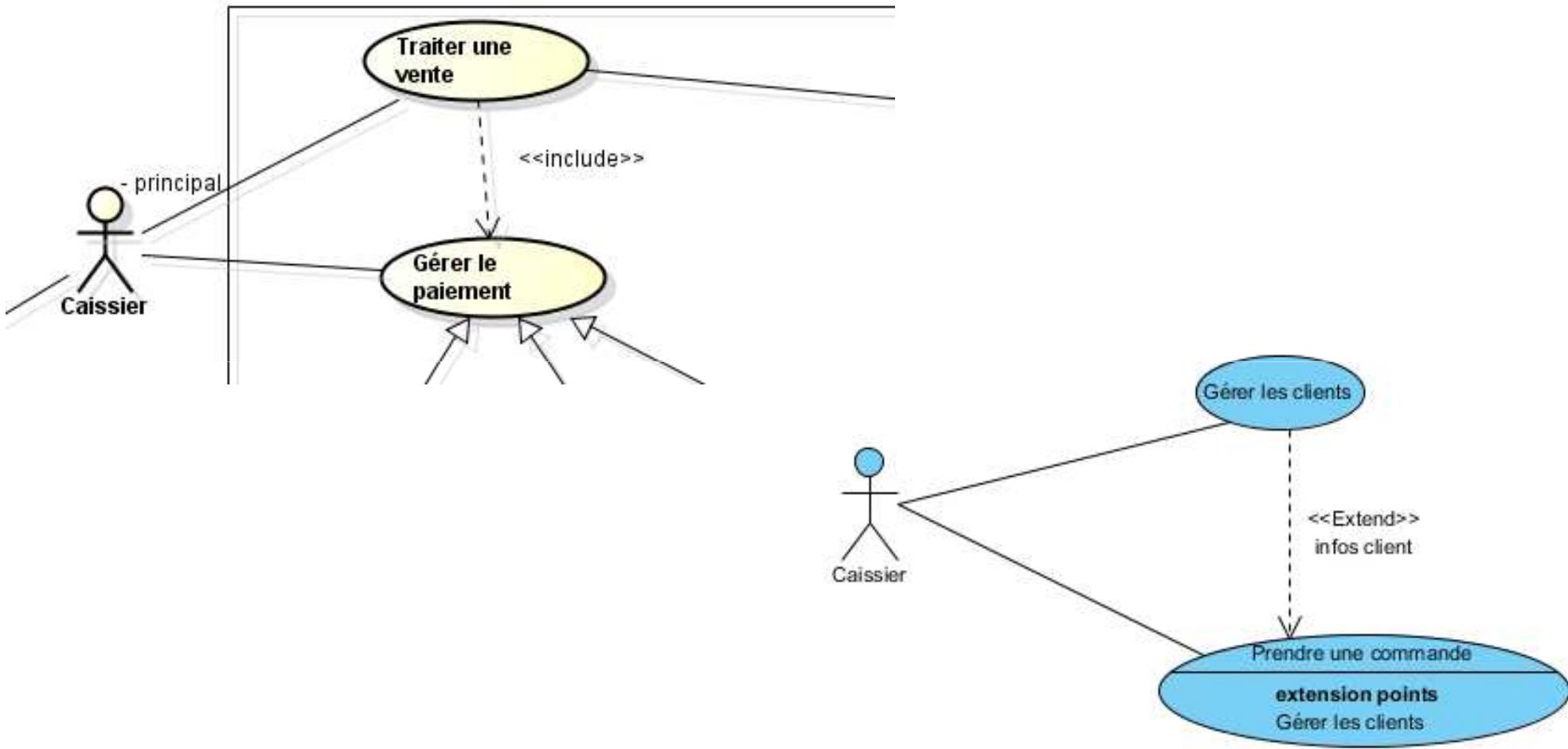


# Cas d'utilisation abstrait, généralisation

---



# Inclusion, extension

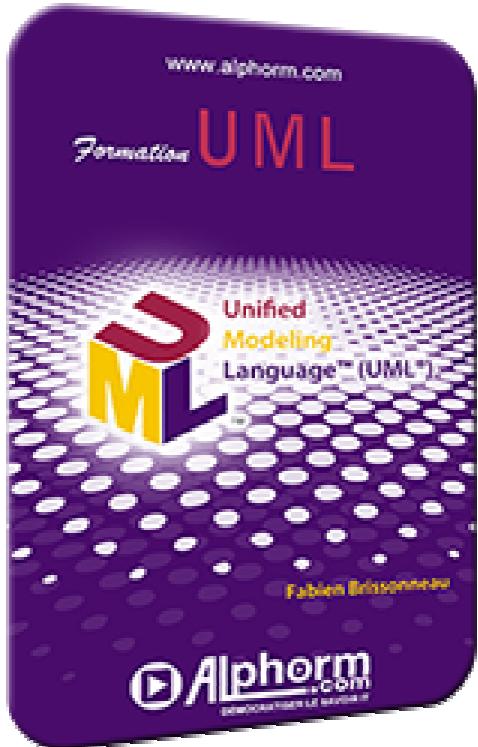


# Ce qu'on a couvert

---

- Les usages du diagramme de cas d'utilisation
- La syntaxe de base : acteur, cas d'utilisation, association
- La syntaxe plus avancée : les inclusion, extension et généralisation entre cas d'utilisation, la généralisation entre acteurs





Généralités

## Les diagrammes UML

# Le diagramme d'activités

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

---

- Usages du diagramme d'activités
- Syntaxe de base
- Syntaxe avancée





# Usages du diagramme d'activités

---

- Diagramme de comportement, dynamique
- Le diagramme est dédié à la description d'un processus, processus métier ou cas d'utilisation
- Un diagramme qui permet de vérifier la complétude d'un processus



# La syntaxe de base

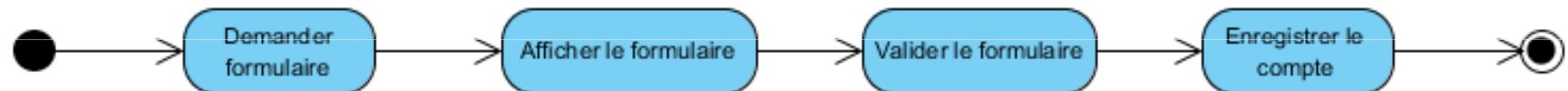
---

- Action, début et fin du processus
- Flux de contrôle, décision
- Activité



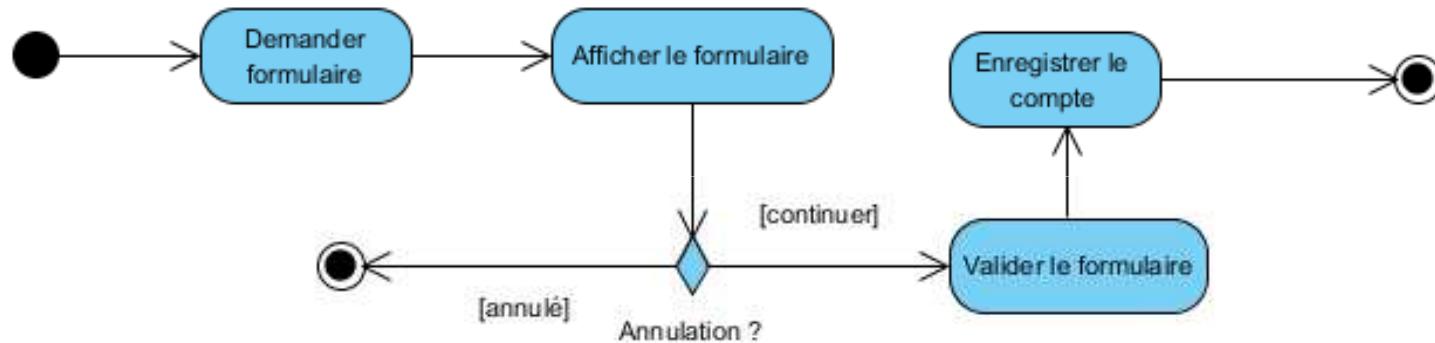
# Action, début et fin de processus

---



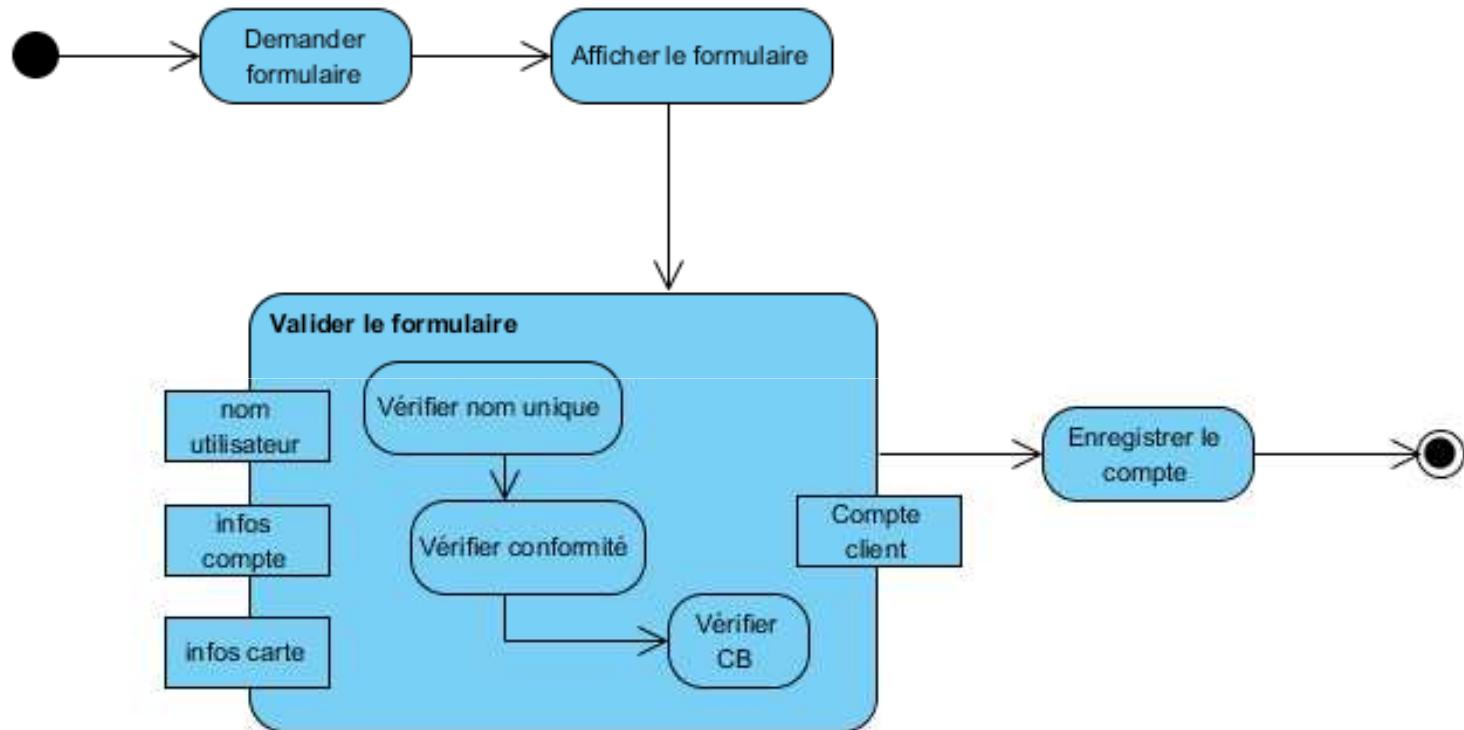


# Décisions





# Activité





# La syntaxe avancée

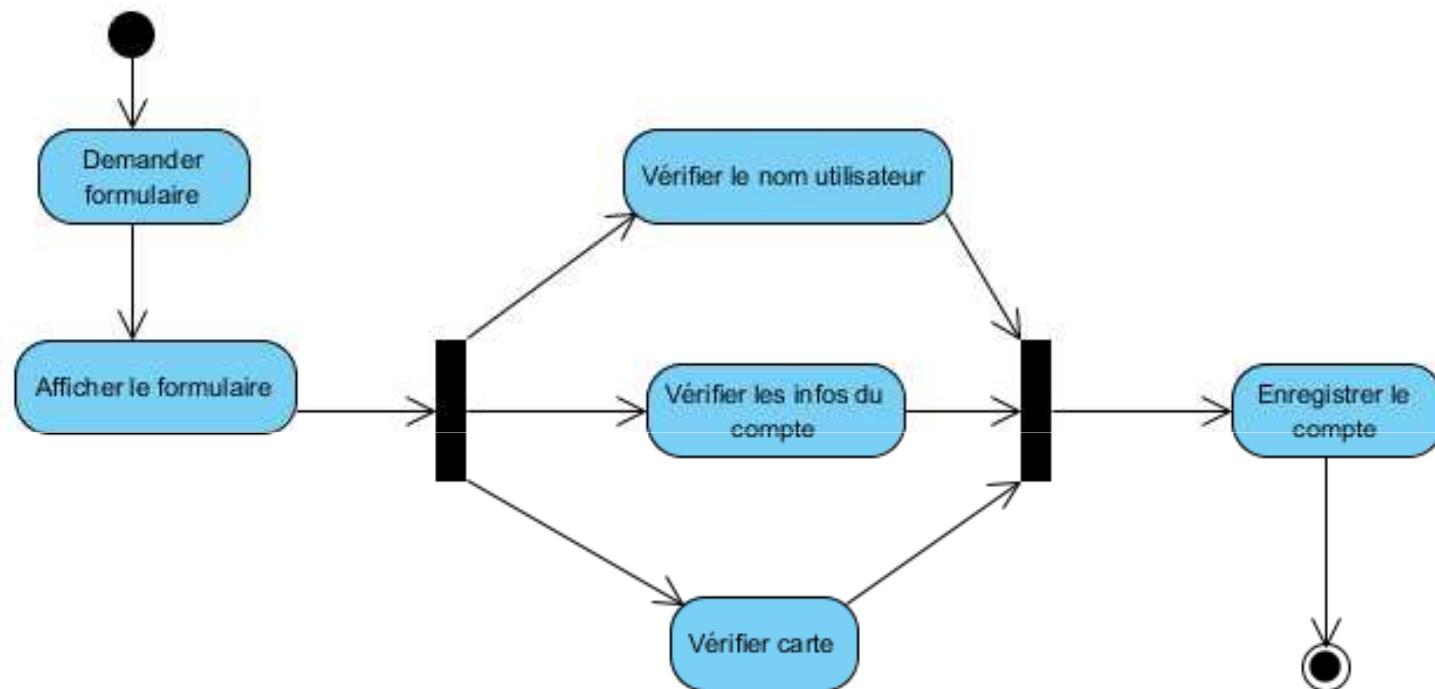
---

- Fork et join
- Fin locale
- Région interruptible
- Object node, flux d'objets
- Partition, swimlane
- Evénements



# Fork et Join

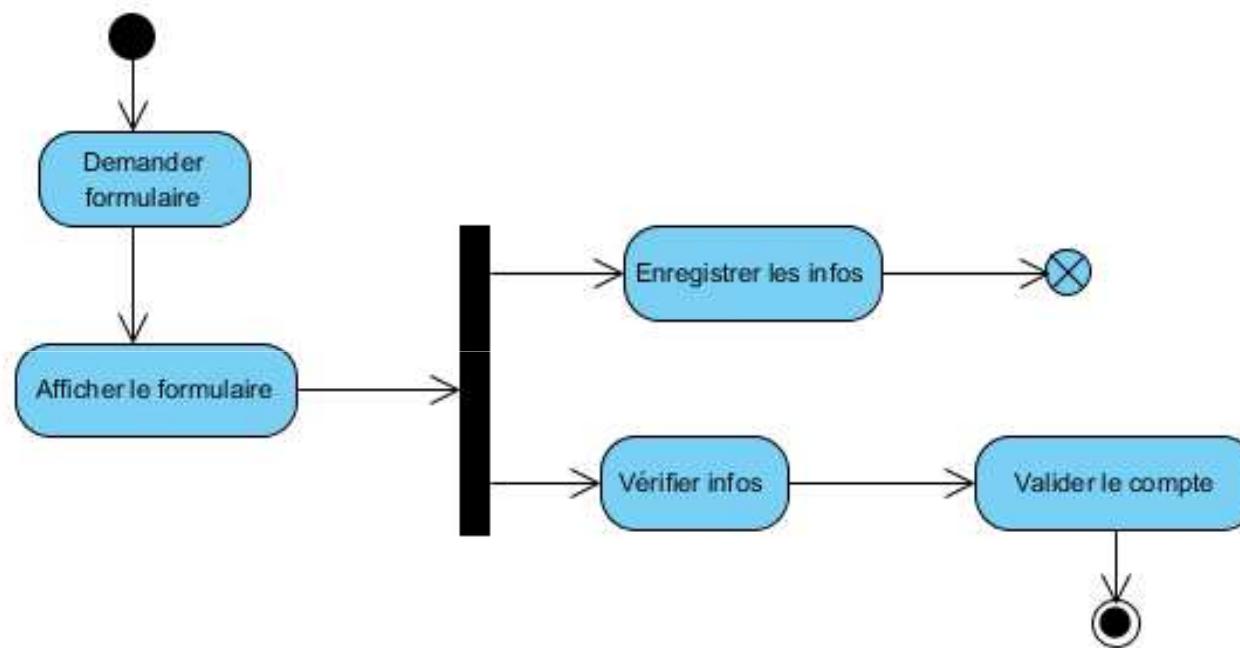
---





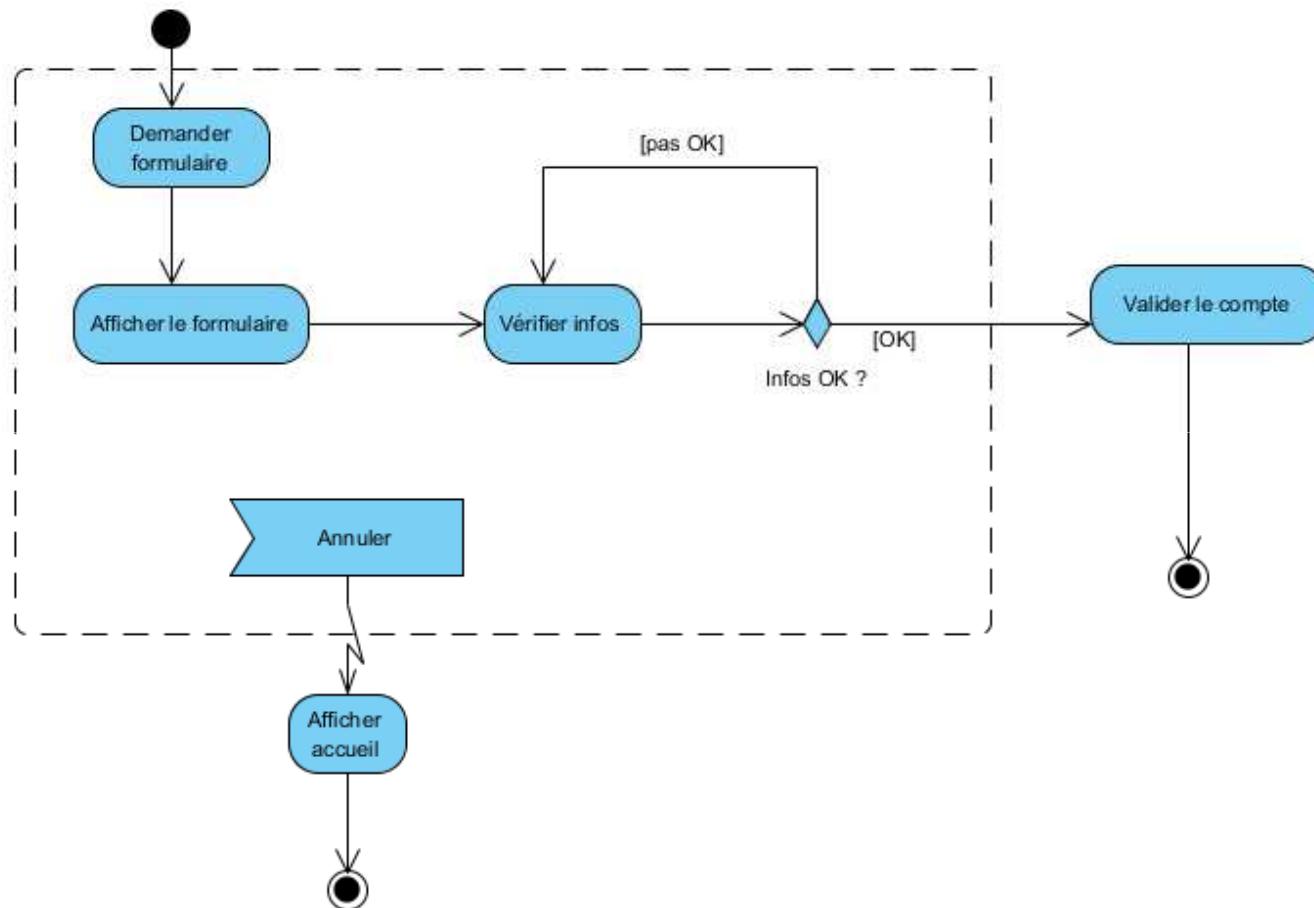
# Fin locale de flux

---



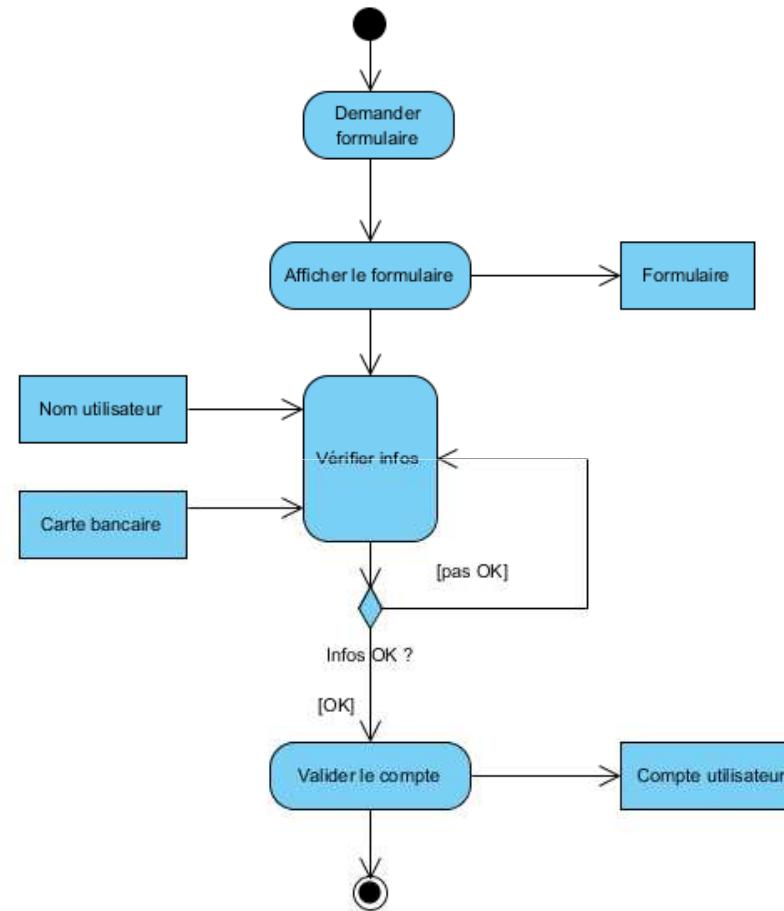


# Région interruptible



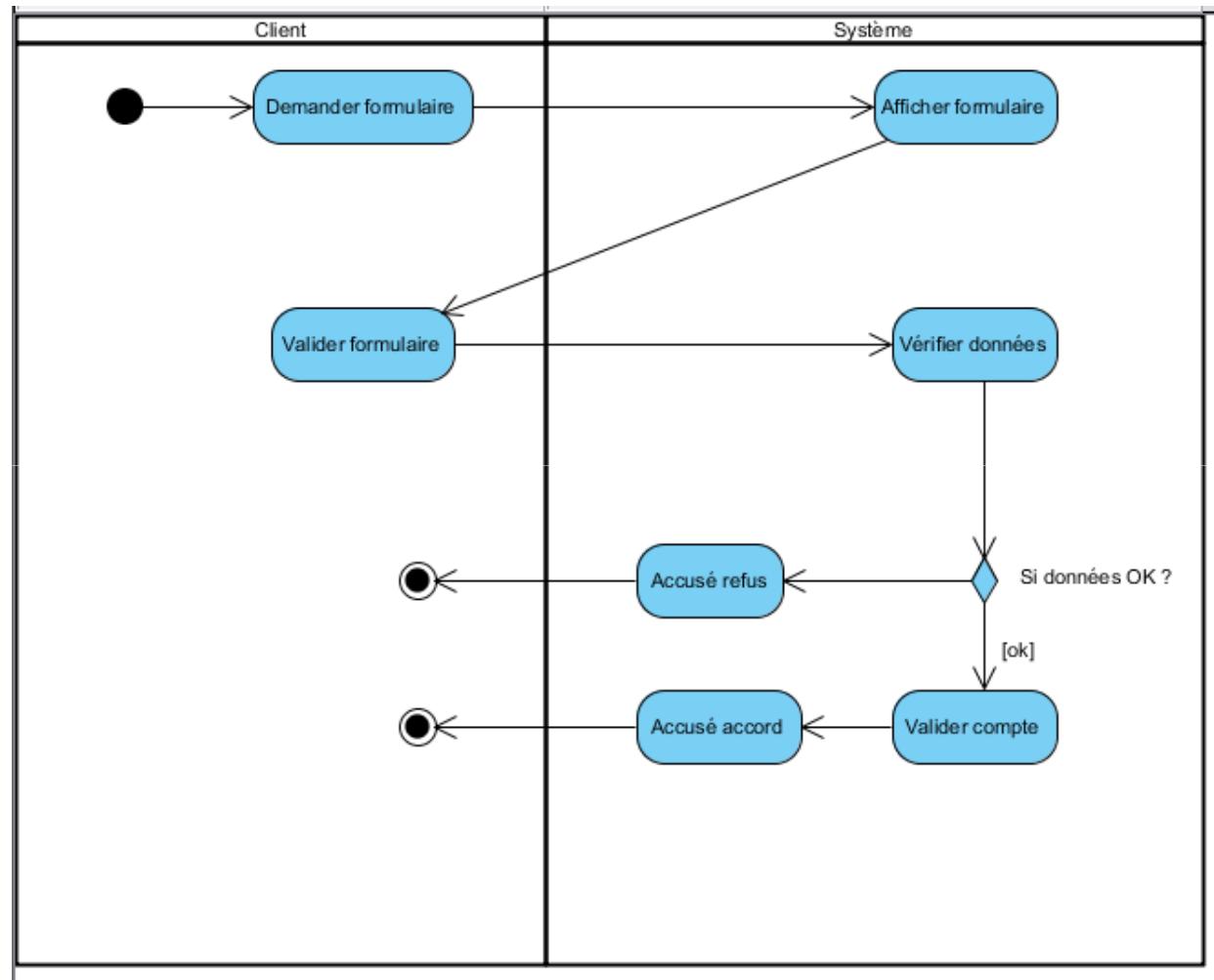


# Object node, flow



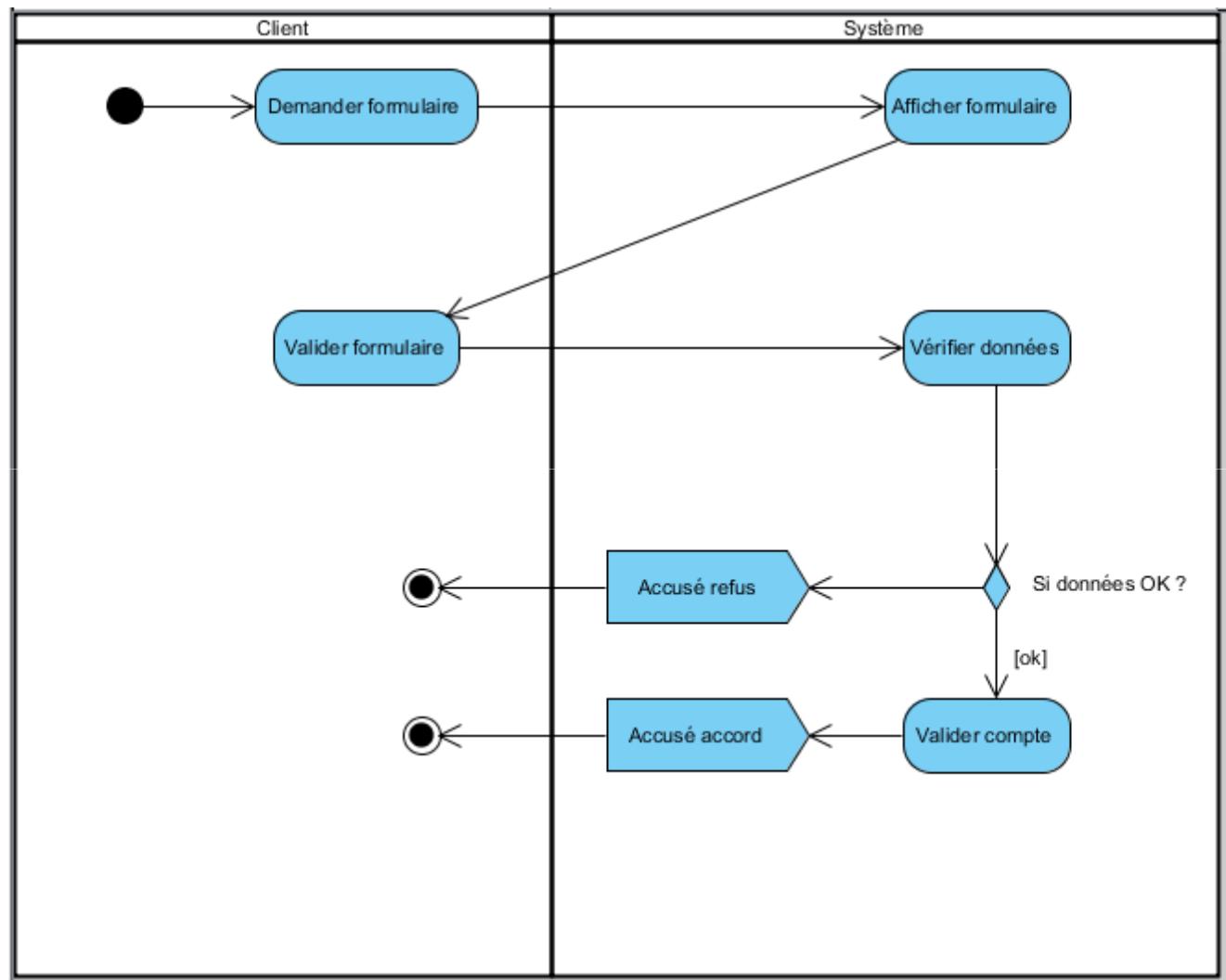


# Partition





# Evénements



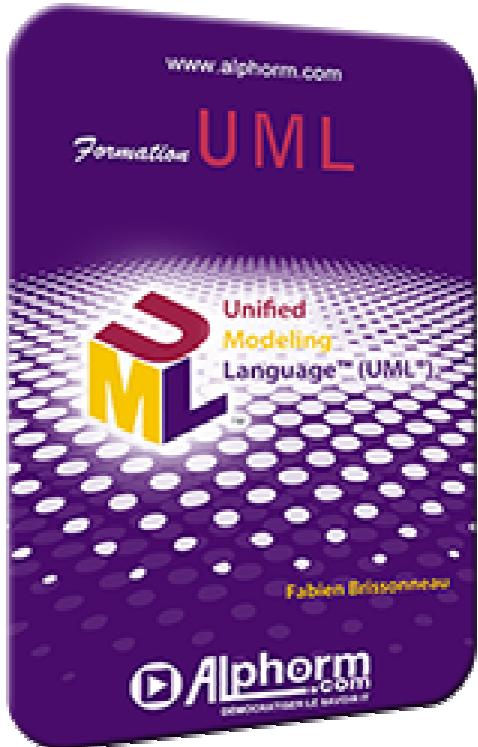


# Ce qu'on a couvert

---

- Les usages du diagramme d'activités
- La syntaxe de base : action, début et fin du processus, flux de contrôle, décision, activité
- La syntaxe plus avancée : fork et join, fin locale, région interruptible, object node, flux d'objets, partition, swimlane, événements





Généralités

## Les diagrammes UML

Le diagramme de  
composants

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau

Consultant, concepteur et formateur

Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Usages du diagramme de composants
- Syntaxe de base





# Usages du diagramme de composants

---

- Diagramme de structure, statique
- Le diagramme permet la représentation de parties de l'application qui sont connectés les unes aux autres, sans pour autant parler de classes.
- Utilisé pour éviter de parler de classes, ou de packages
- Utile lors des efforts de rétro-conception



# La syntaxe de base

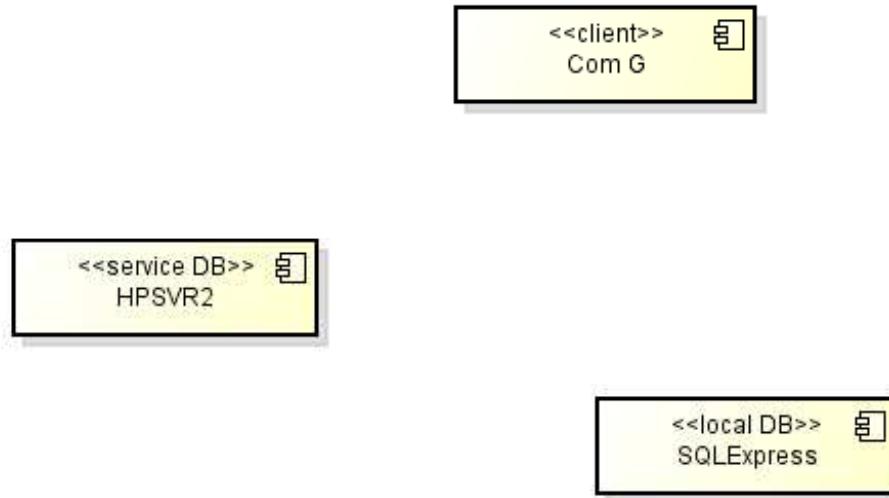
---

- Composant
- Interfaces
- Ports
- Dépendances



# Le composant

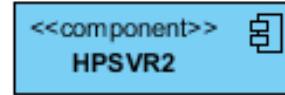
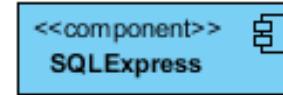
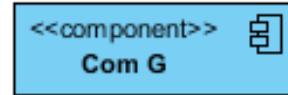
---





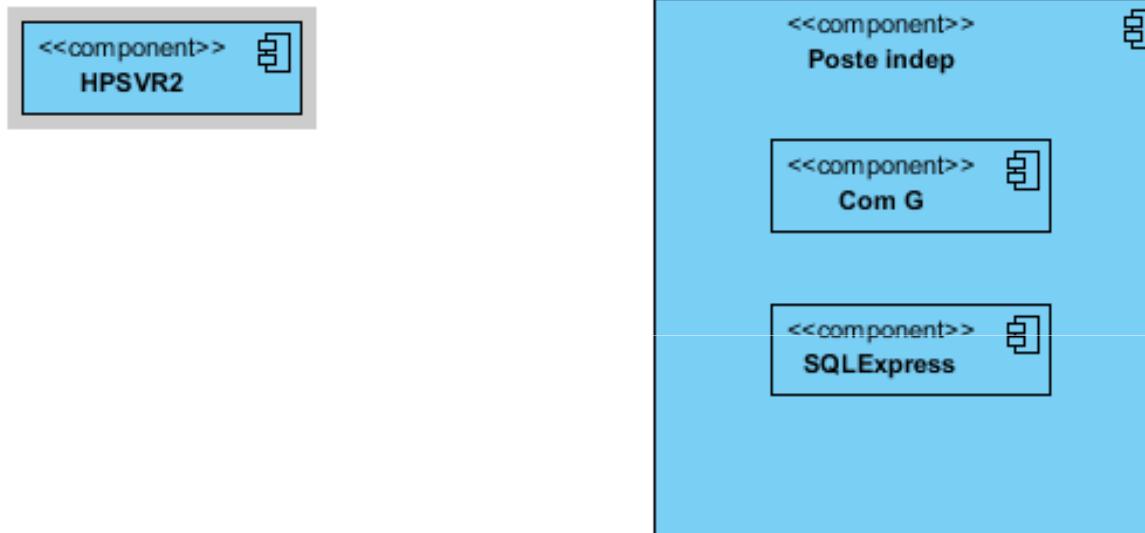
# Le composant

---



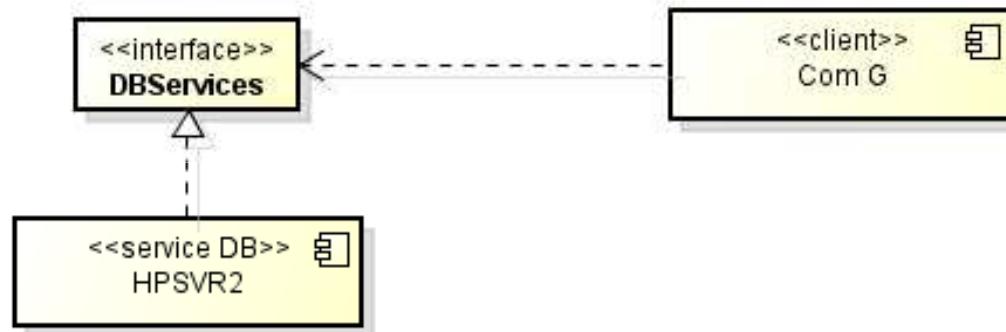


# Composant... décomposé



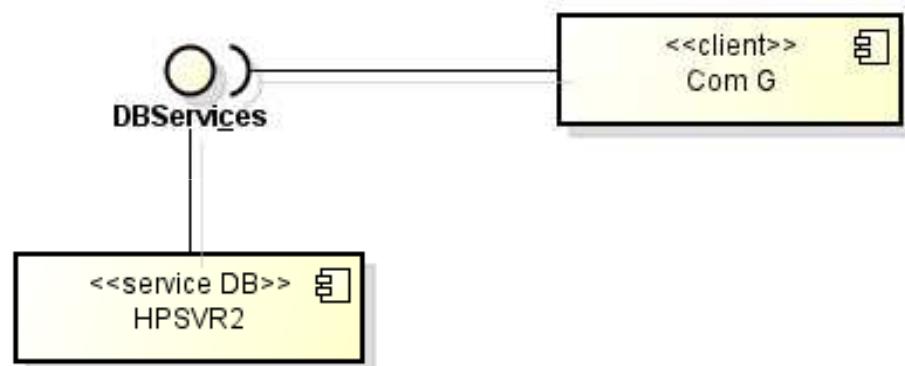


# Interfaces



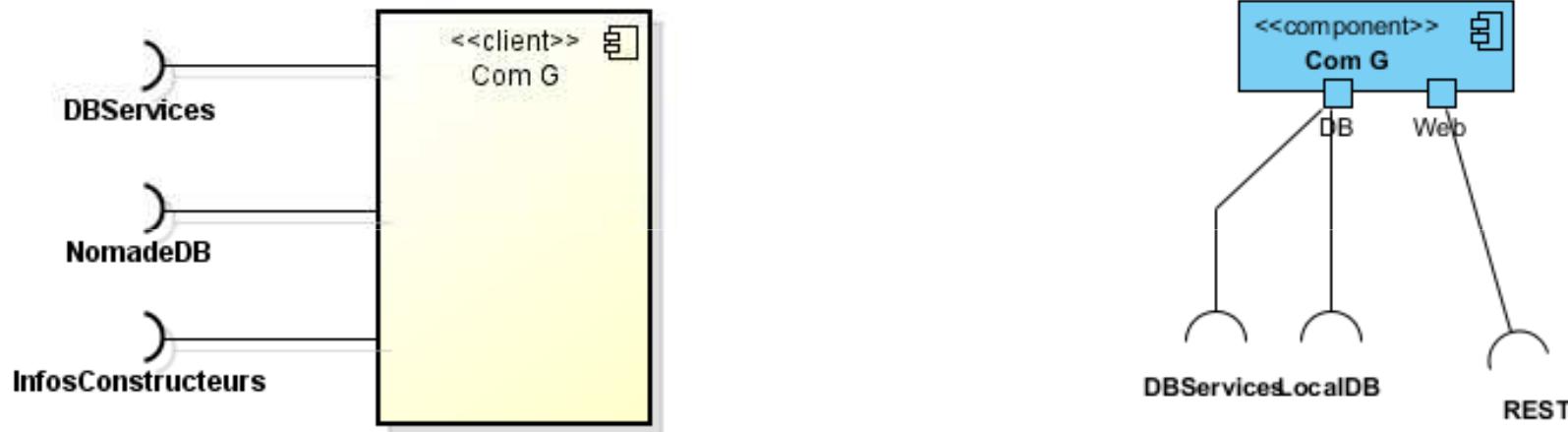


# Interfaces



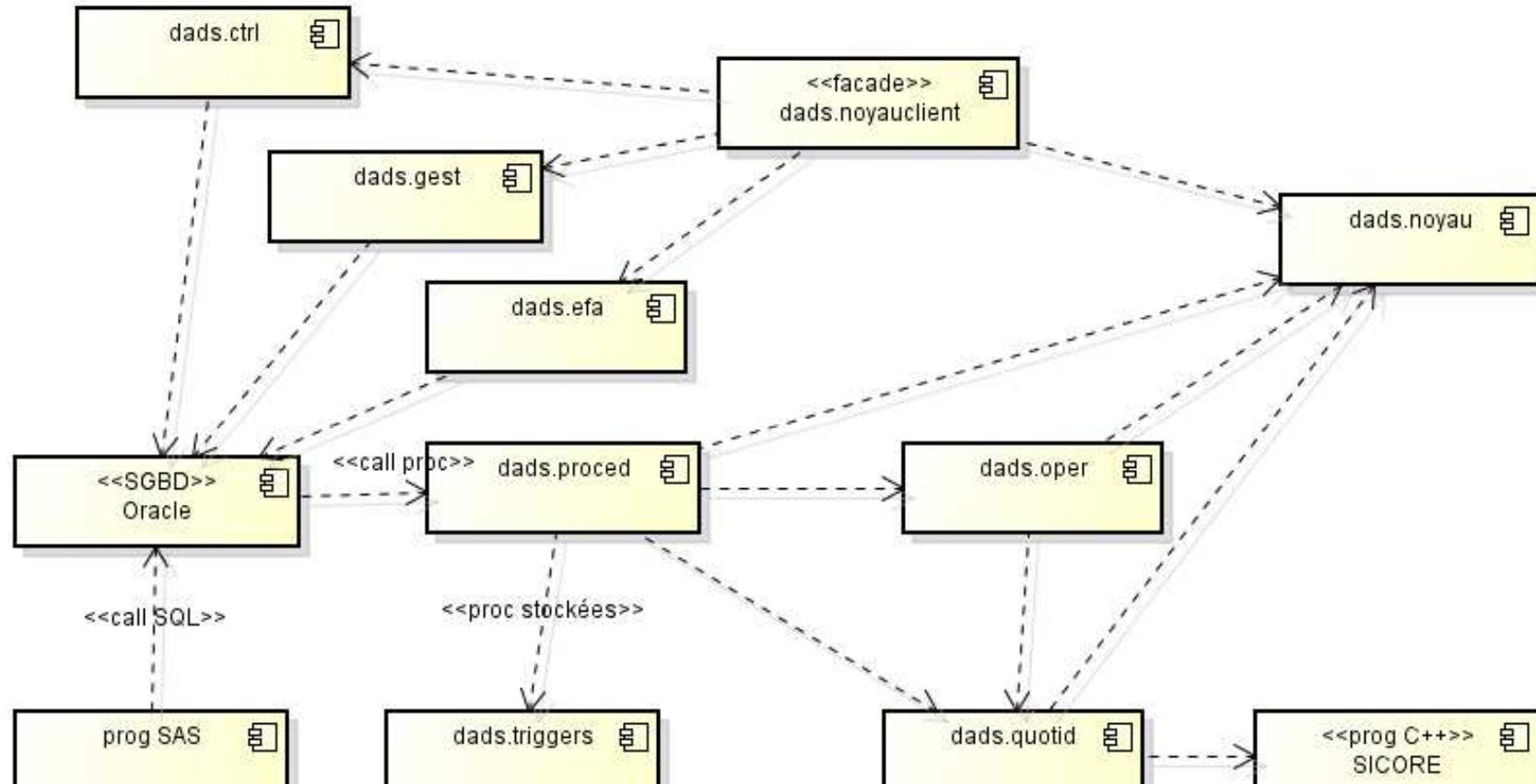


# Composants et interfaces





# Composants et dépendances



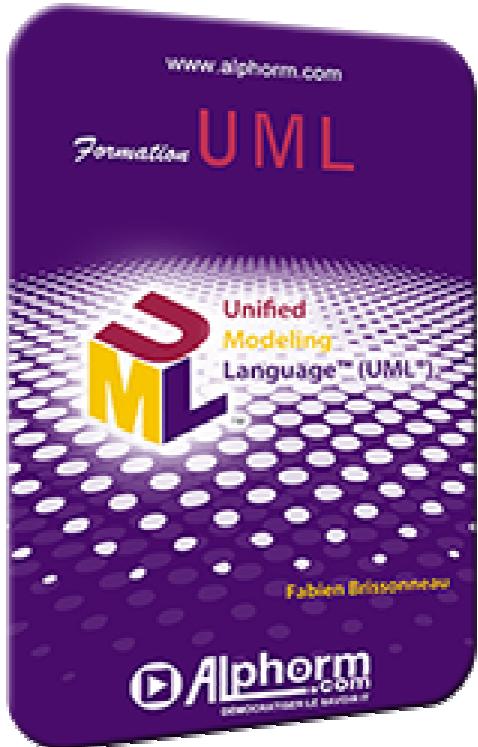


# Ce qu'on a couvert

---

- Les usages du diagramme de composants
- La syntaxe de base : les composants et les interfaces, les dépendances entre les composants





Généralités

## Les diagrammes UML

# Le diagramme de déploiement

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

---

- Usages du diagramme de déploiement
- Syntaxe de base





# Usages du diagramme de déploiement

---

- Diagramme de structure, statique
- Il sert à décrire une architecture physique, les nœuds d'un réseau sur lequel sera déployé l'application



# La syntaxe de base

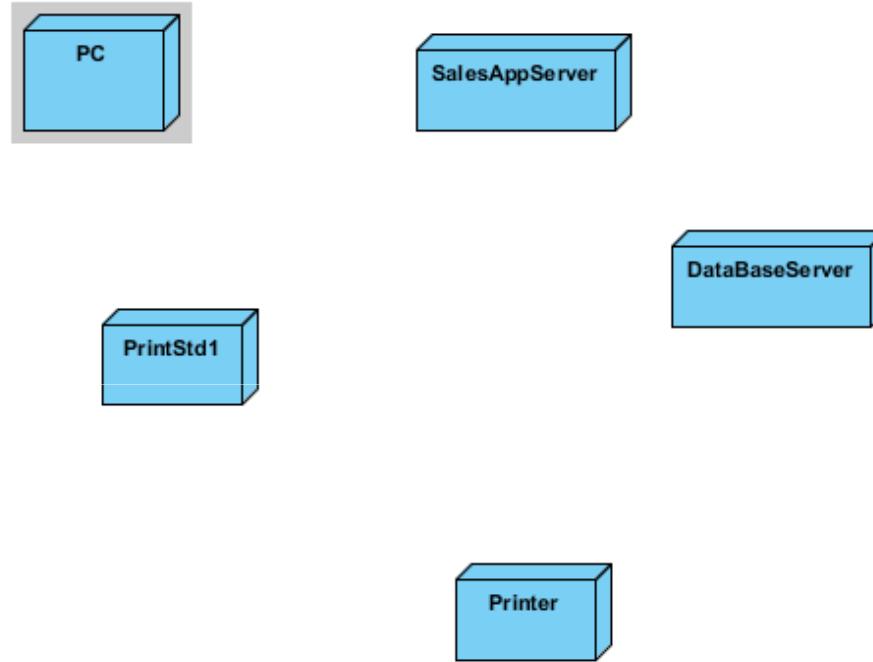
---

- Noeud
- Lien
- Stéréotypes
- Artefacts



# Les noeuds

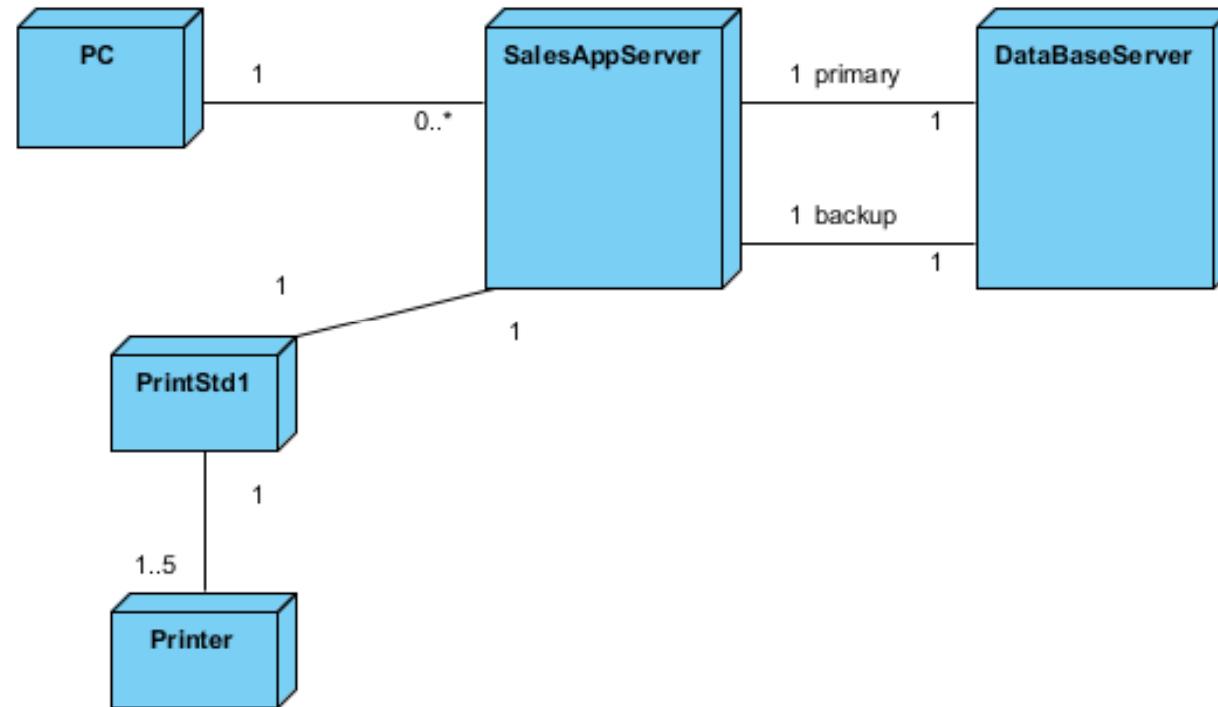
---





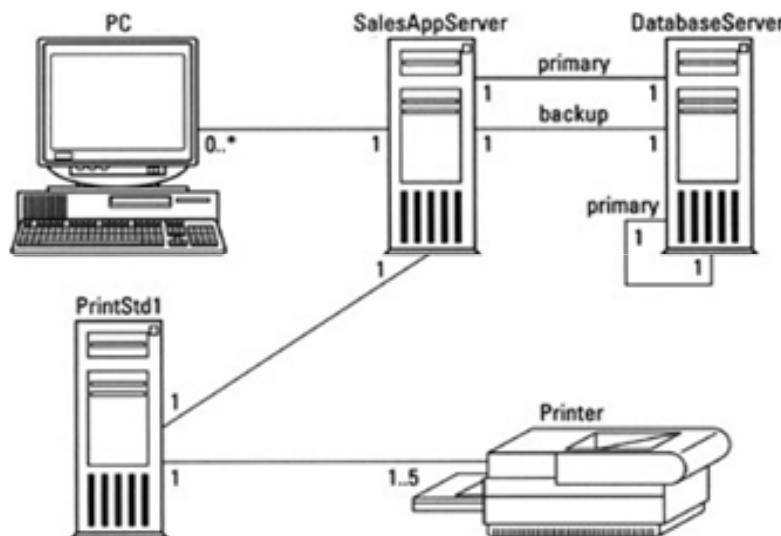
# Noeuds et liens

---

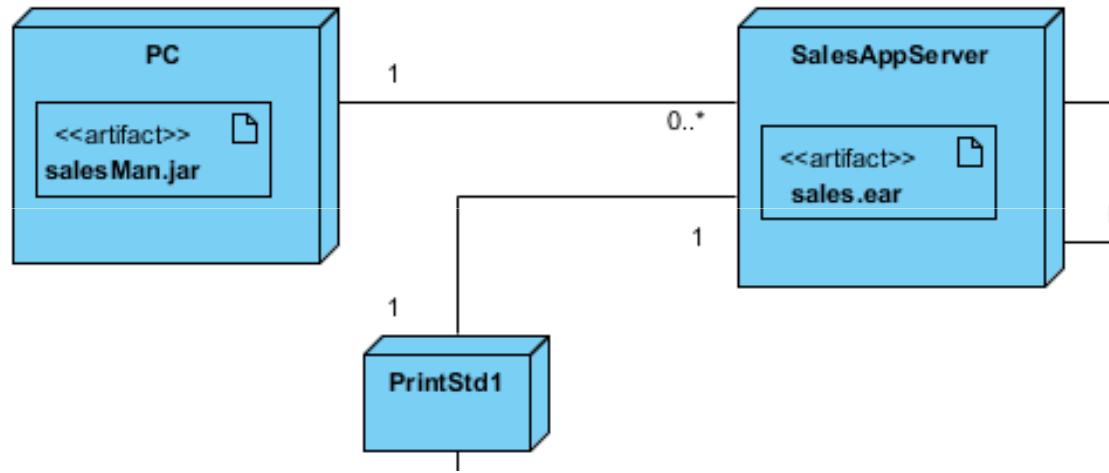




# Noeud et stéréotypes



# Noeud et artefacts



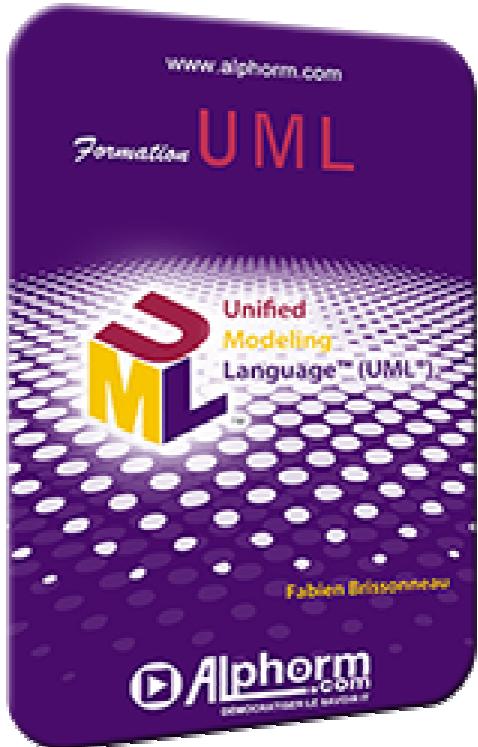


# Ce qu'on a couvert

---

- Les usages du diagramme de déploiement
- La syntaxe de base : nœud, lien, artefacts





Généralités

## Les diagrammes UML

# Le diagramme de packages

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

---

- Usages du diagramme de packages
- Syntaxe de base





# Usages du diagramme de packages

---

- Diagramme de structure, statique
- Un package regroupe n'importe quel type d'élément UML
- Très utile pour regrouper des classes
- On étudie préférentiellement les dépendances entre ces classes
- Permet d'avoir une vue de haut niveau sur l'architecture de l'application



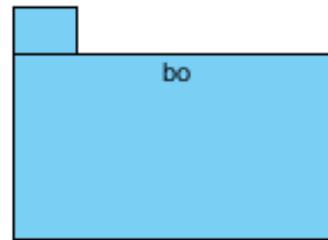
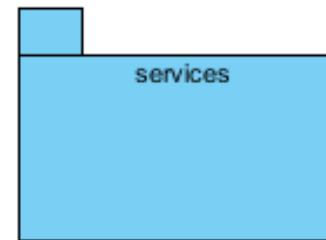
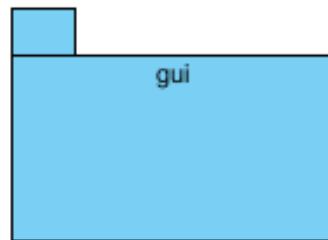
# La syntaxe de base

---

- Packages, stéréotypes
- Dépendances
- Contenance



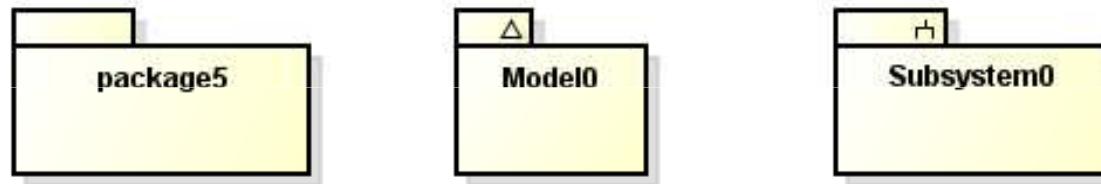
# Package





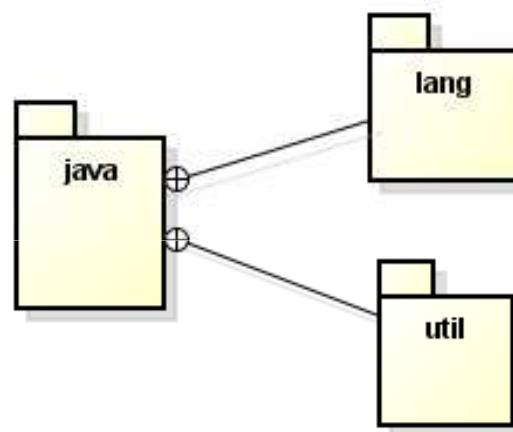
# Stéréotypes courants

---



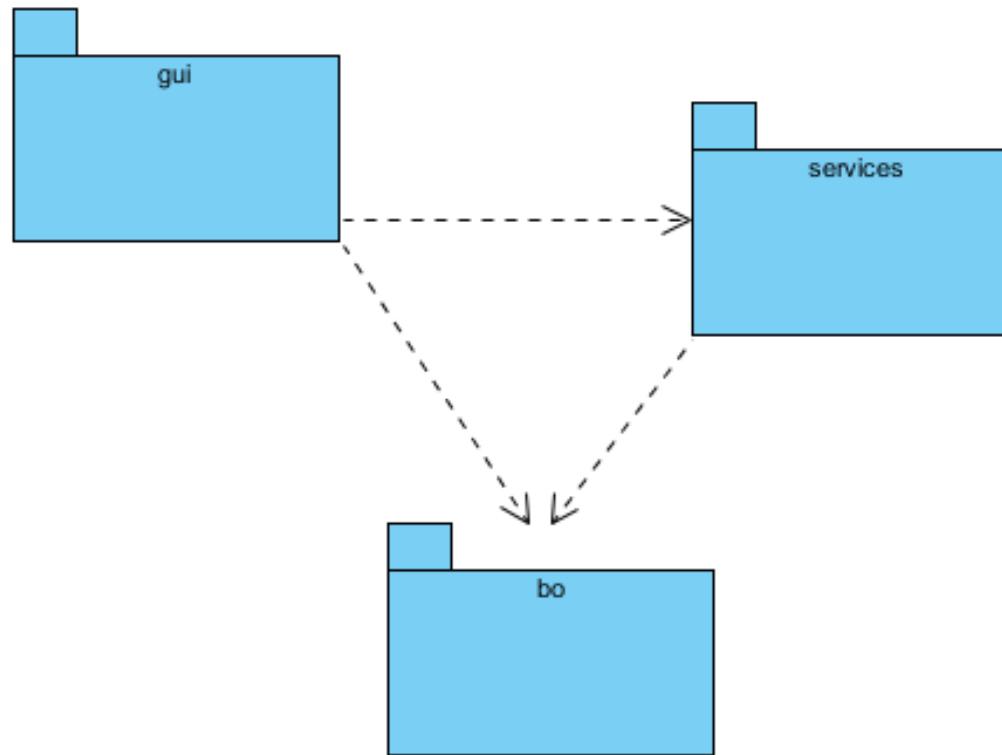


# Contenance



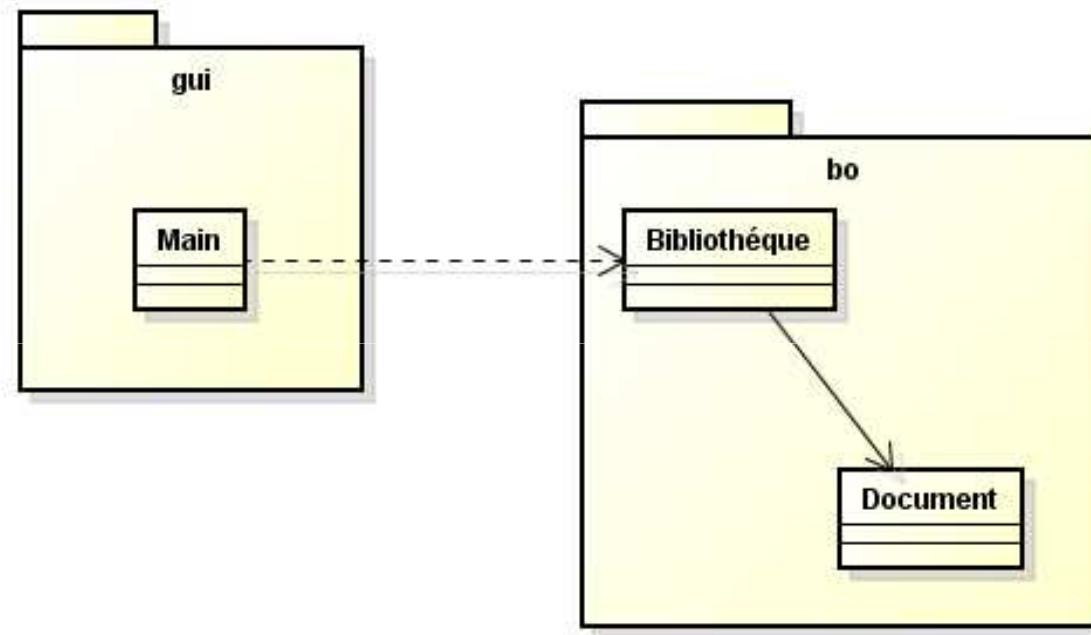


# Dépendances entre packages



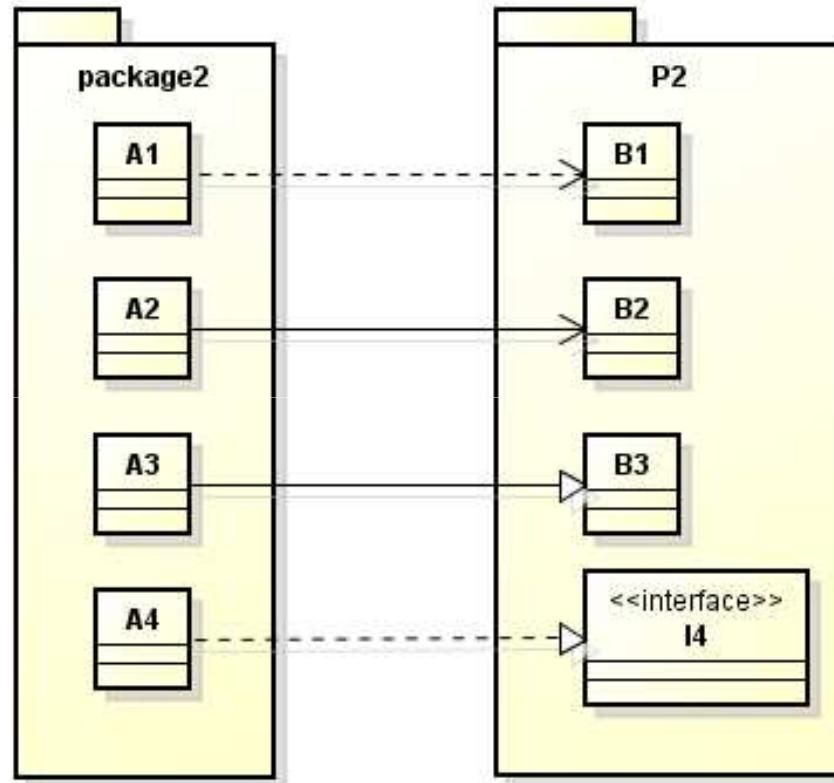


# Origines des dépendances entre packages



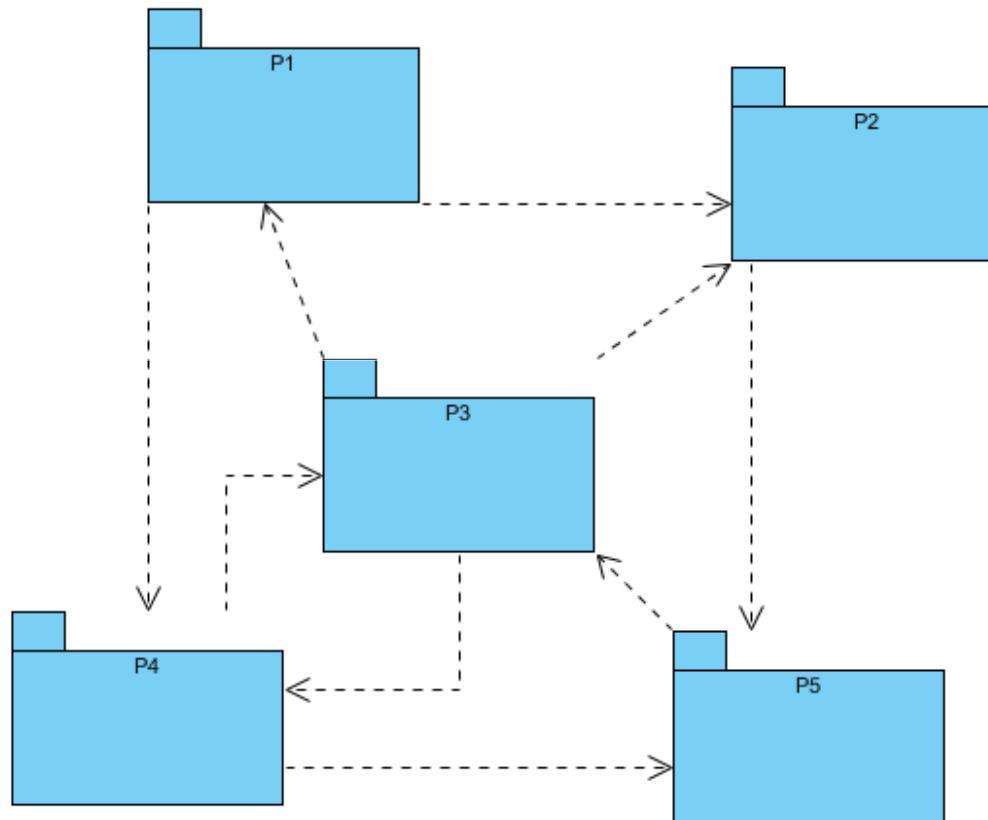


# Origines des dépendances





# Acyclic Dependency Principle



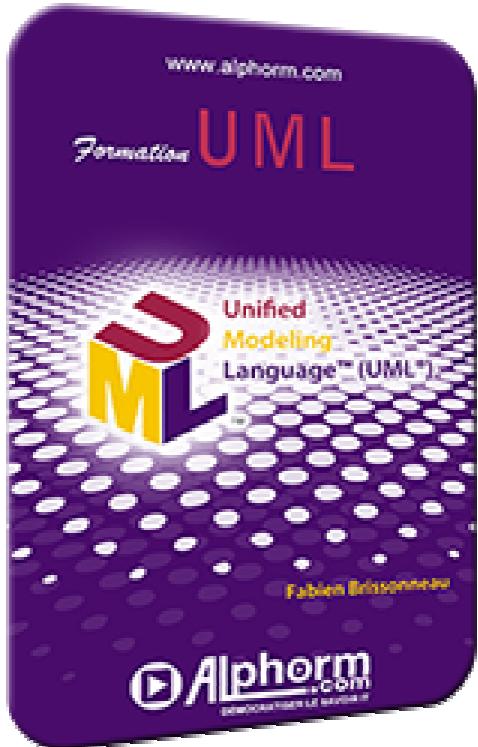


# Ce qu'on a couvert

---

- Les usages du diagramme de packages
- La syntaxe de base : package, stéréotypes, contenance, dépendances
- L'organisation en packages, les bons principes





Généralités

## Les diagrammes UML

# Le diagramme d'états

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

---

- Usages du diagramme d'états
- Syntaxe de base
- Syntaxe avancée





# Usages du diagramme d'états

---

- Diagramme de comportement, dynamique
- Un état représente un temps dans la vie d'un objet
- Le diagramme d'état représente le cycle de vie d'un objet
- Très intéressant pour décrire les états possibles et les transitions autorisées
- Utilisé sur des objets dont le cycle de vie est significatif, ou dont les réponses aux stimuli extérieurs varient selon les états



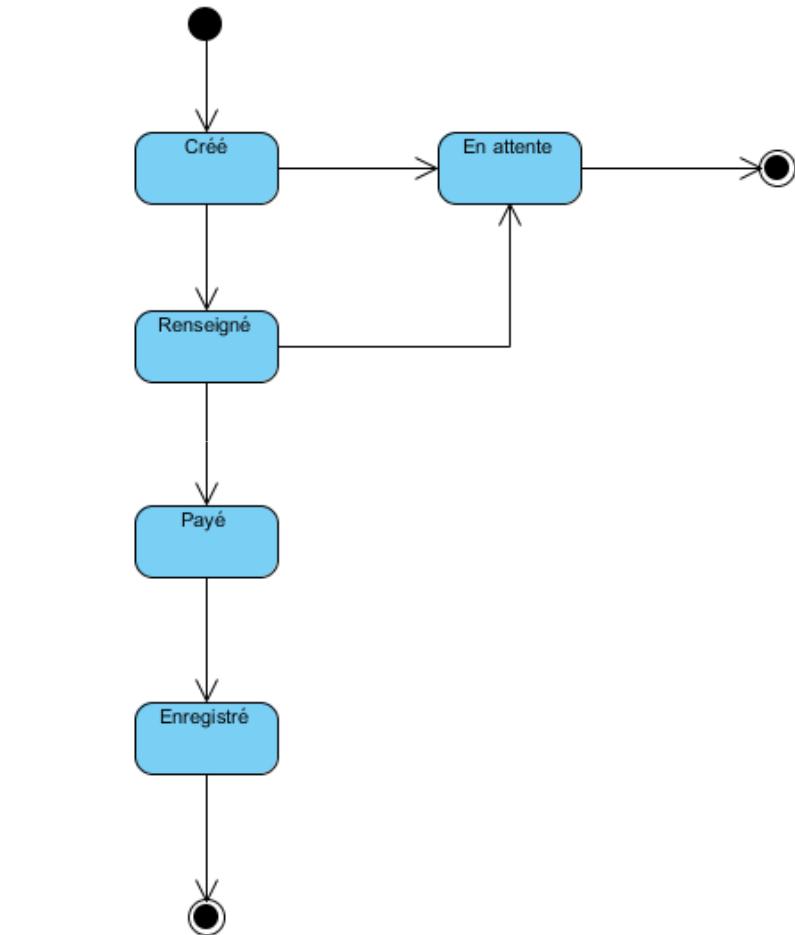
# La syntaxe de base

---

- Etats, transitions, initial, final
- Evénements, conditions
- Effets, activités

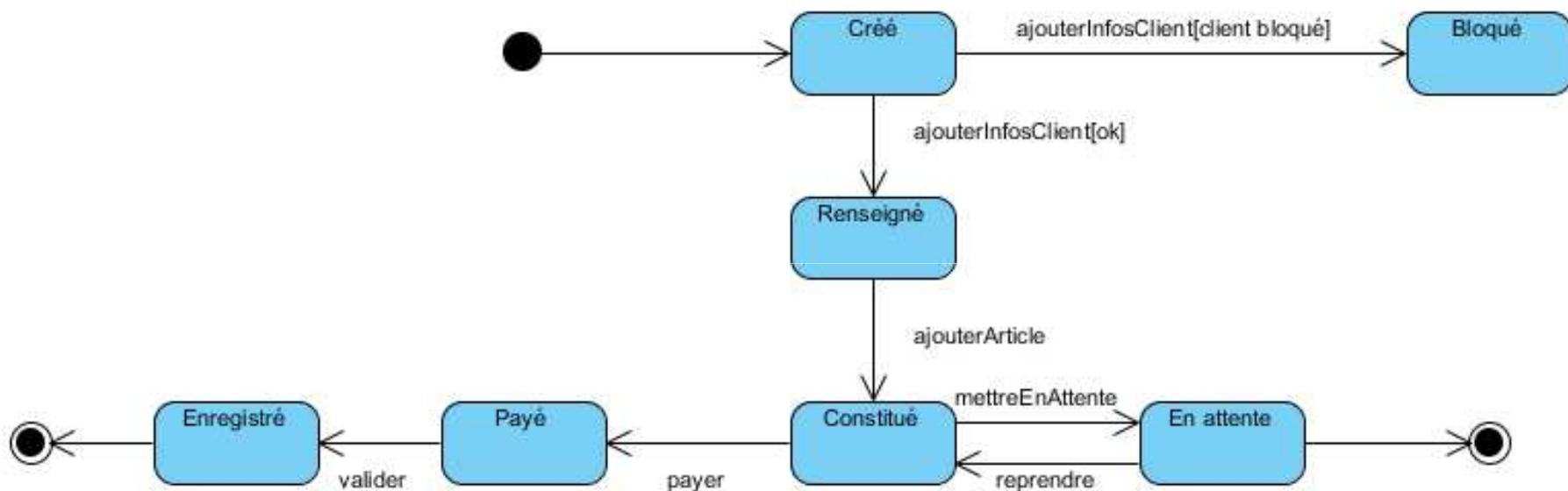


# Etats



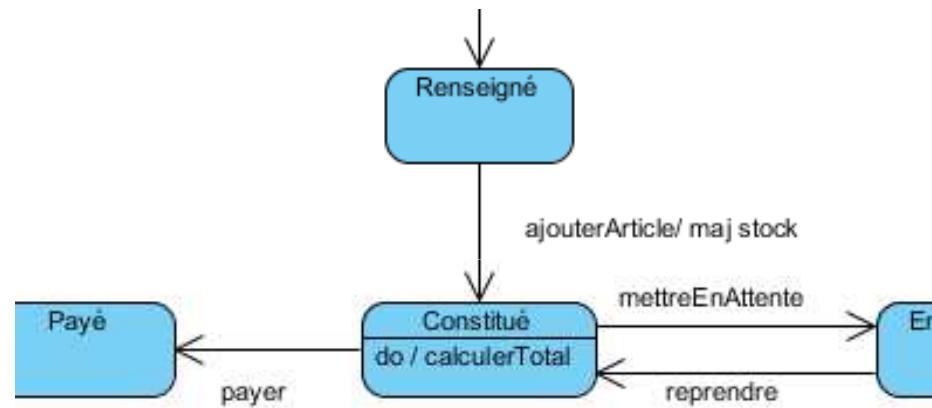


# Evénements et conditions





# Activités et effets





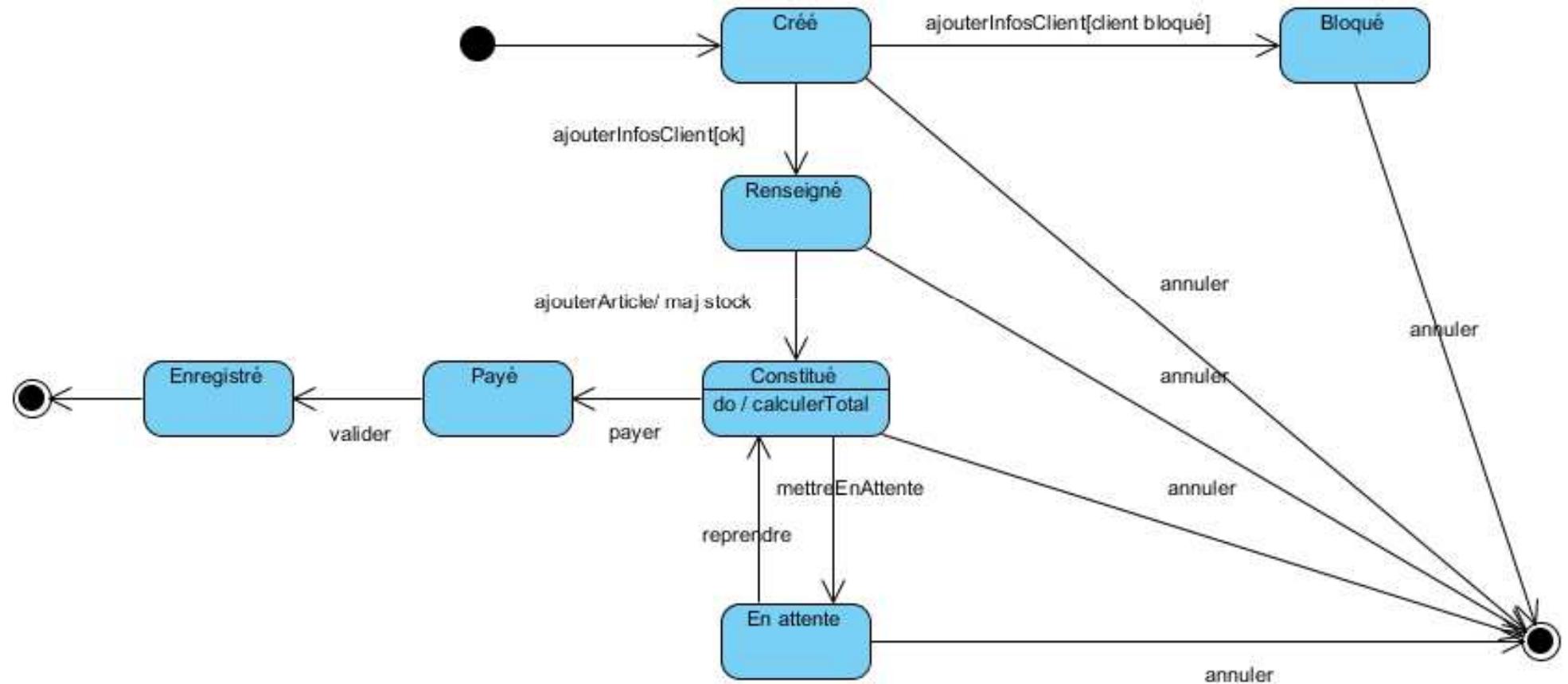
# La syntaxe avancée

---

- Evénements prédéfinis, états composites
- Transition propre ou interne
- Effets en entrée, en sortie

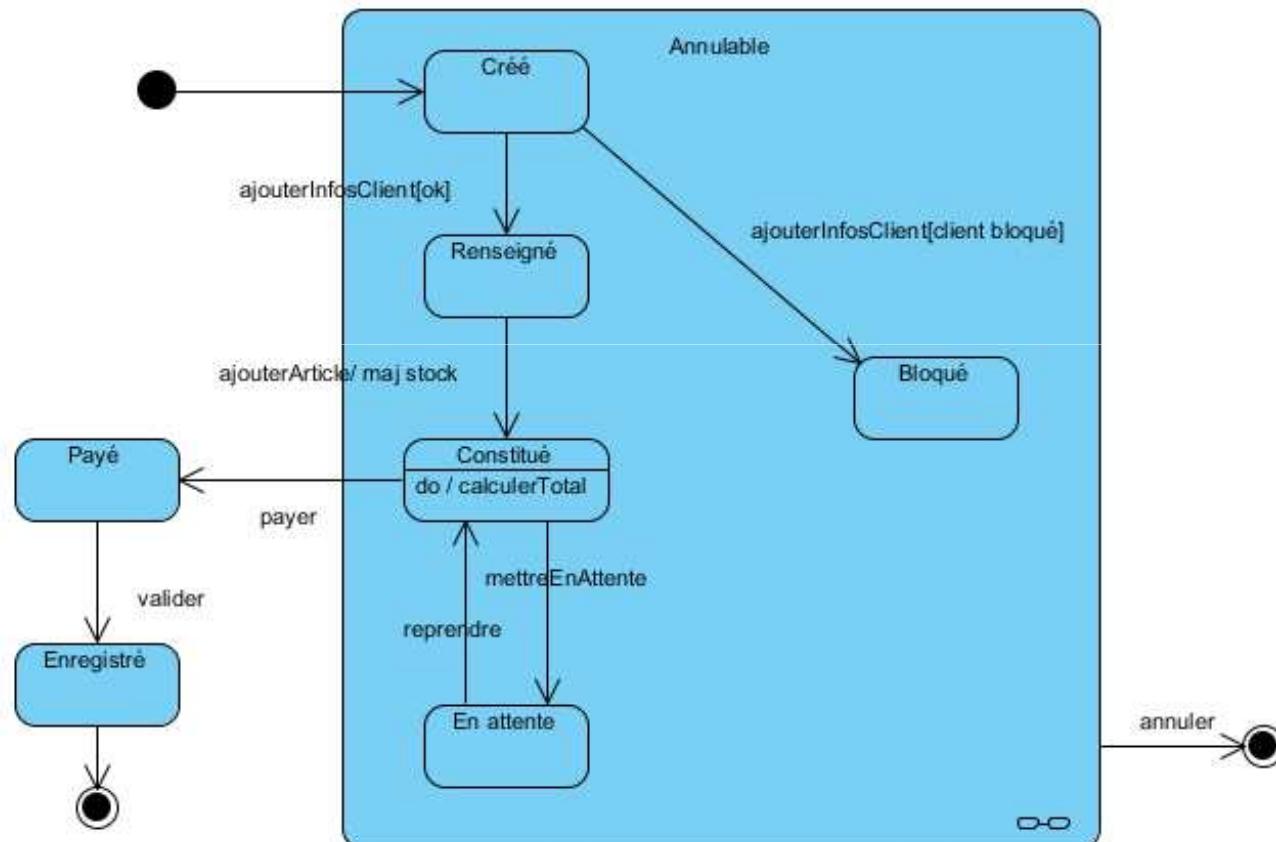


# Evénements prédéfinis, états composites



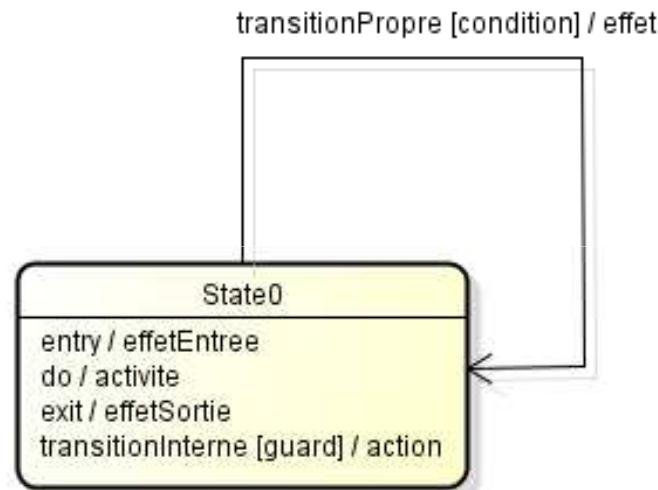


# Etats composites



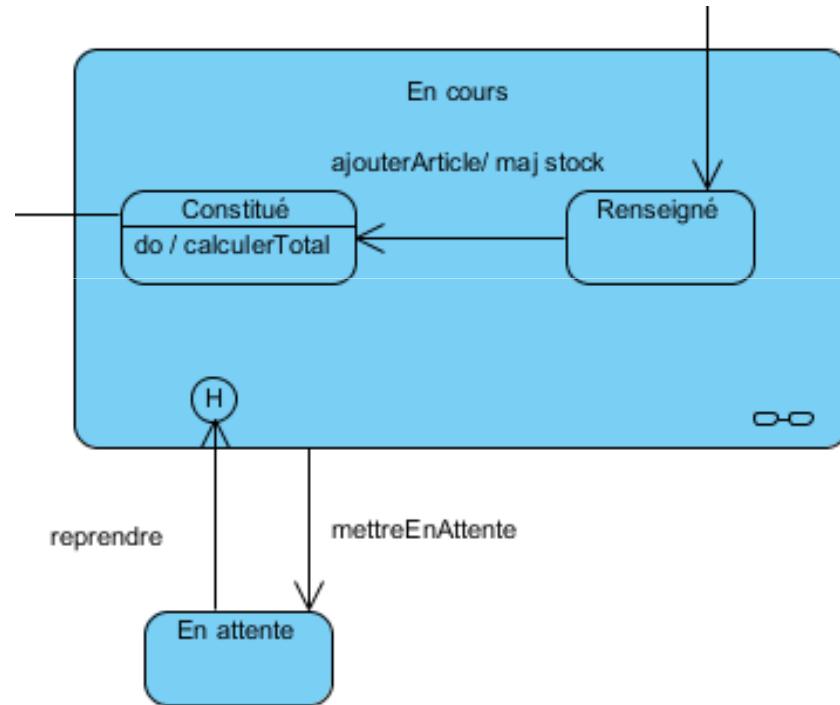
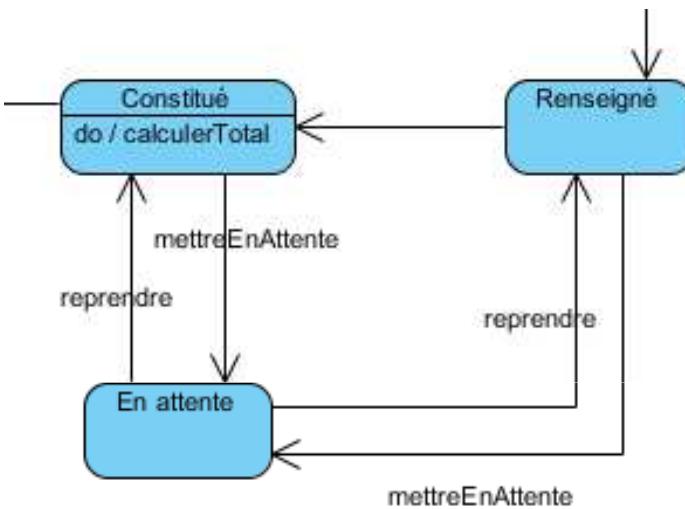


# Transition propre, interne, effet entrée, sortie





# Pseudo état historique



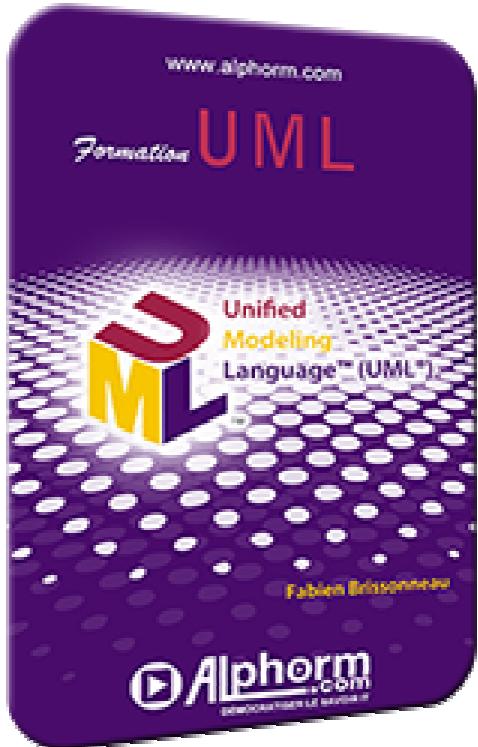


# Ce qu'on a couvert

---

- Les usages du diagramme d'états
- La syntaxe de base : états, transitions, initial, final, événements, conditions, effets, activités
- La syntaxe avancée : événements prédéfinis, états composites, transition propre ou interne, effets en entrée, en sortie, pseudo-état historique





Généralités

## Les diagrammes UML

Le diagramme de communication

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

---

- Usages du diagramme de communication
- Syntaxe de base
- Syntaxe avancée





# Usages du diagramme de communication

---

- Diagramme de comportement, dynamique
- Sert à montrer les interactions entre des objets. Les objets sont disposés dans l'espace, liés par des liens, et les messages circulent sur ces liens
- Destiné aux interactions logicielles, il peut être utilisé à plus haut niveau
- Parfois utilisé dans l'analyse de besoins sous le nom de diagramme de contexte dynamique



# La syntaxe de base

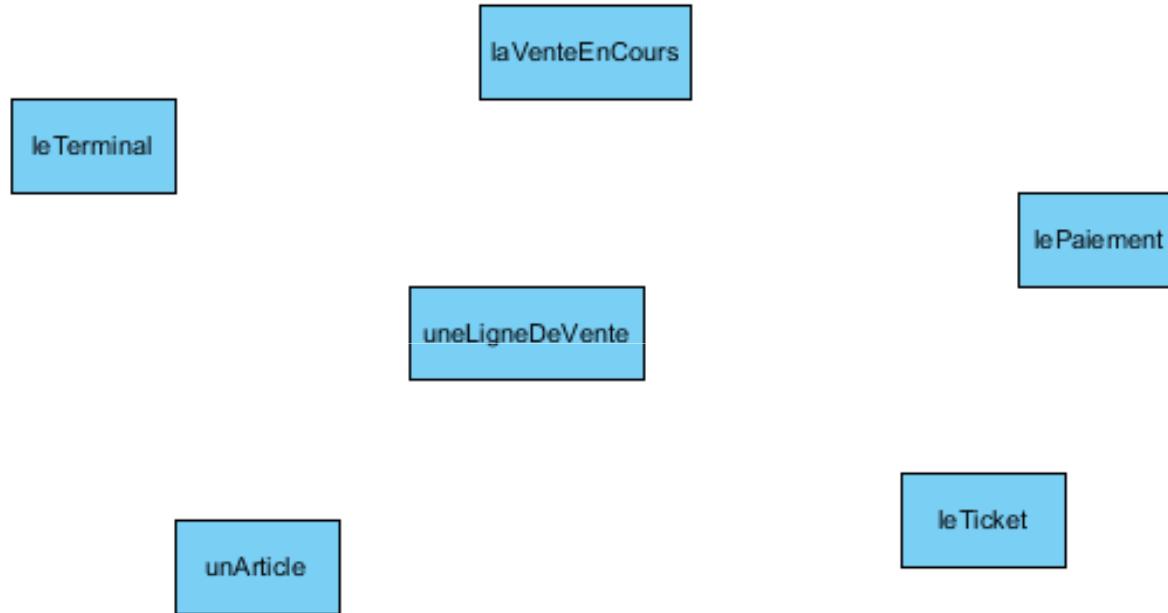
---

- Les objets
- Les liens
- Les messages



# Les objets

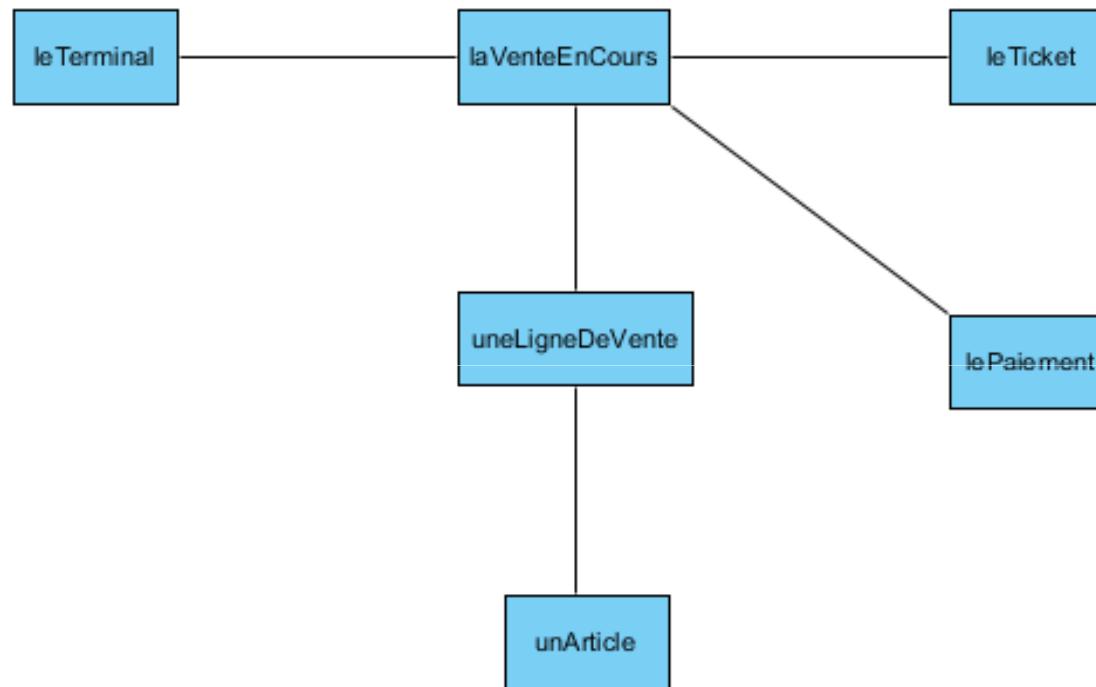
---





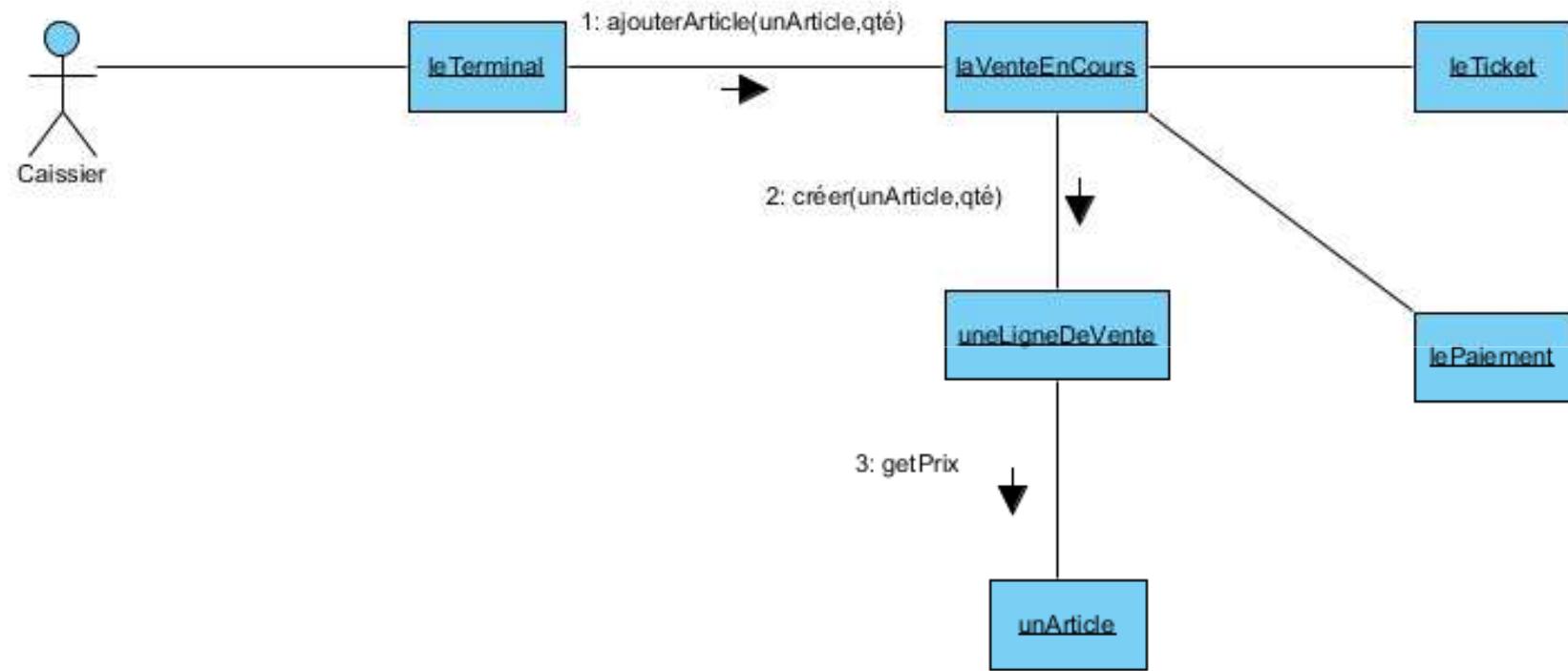
# Les liens

---





# Les messages





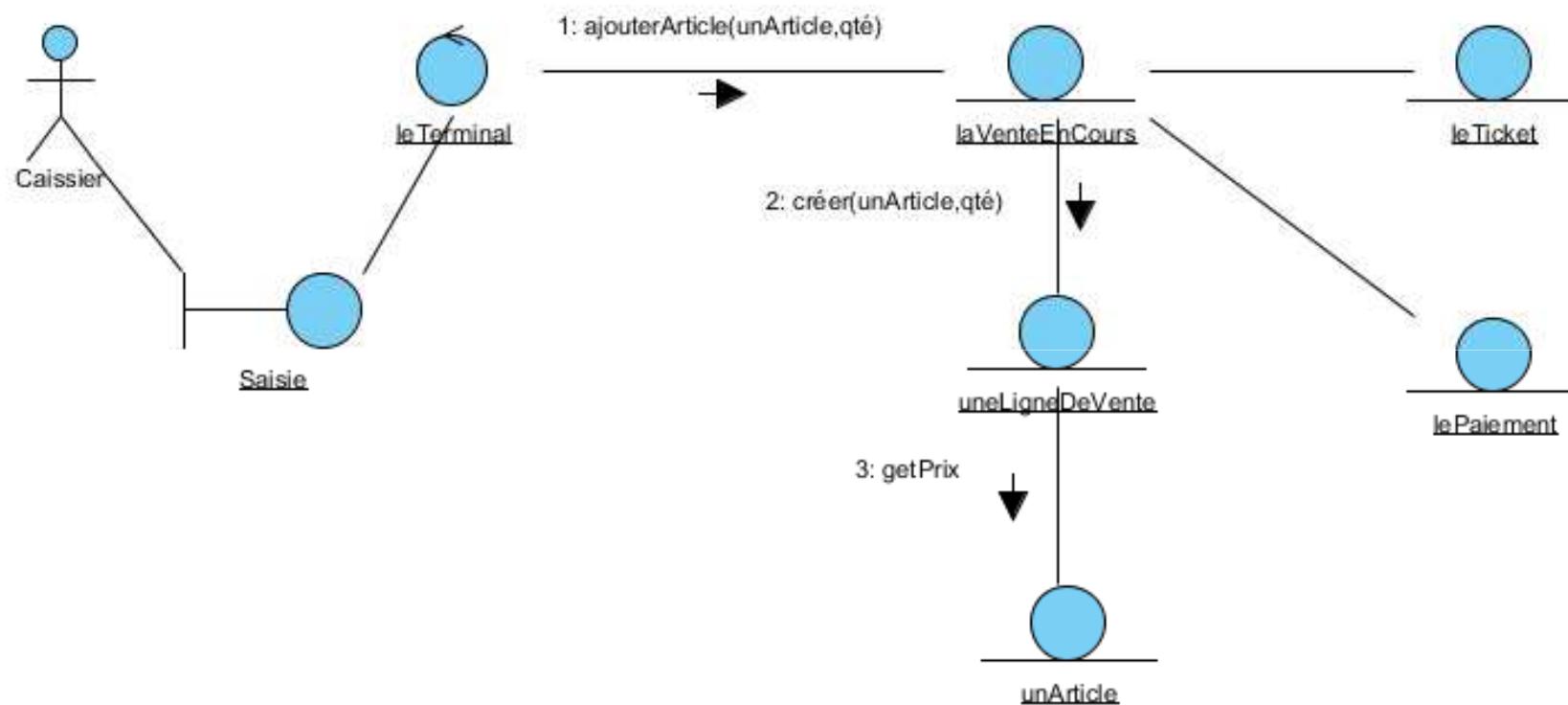
# La syntaxe avancée

---

- Boundary, control, entity
- Numérotation
- Itération

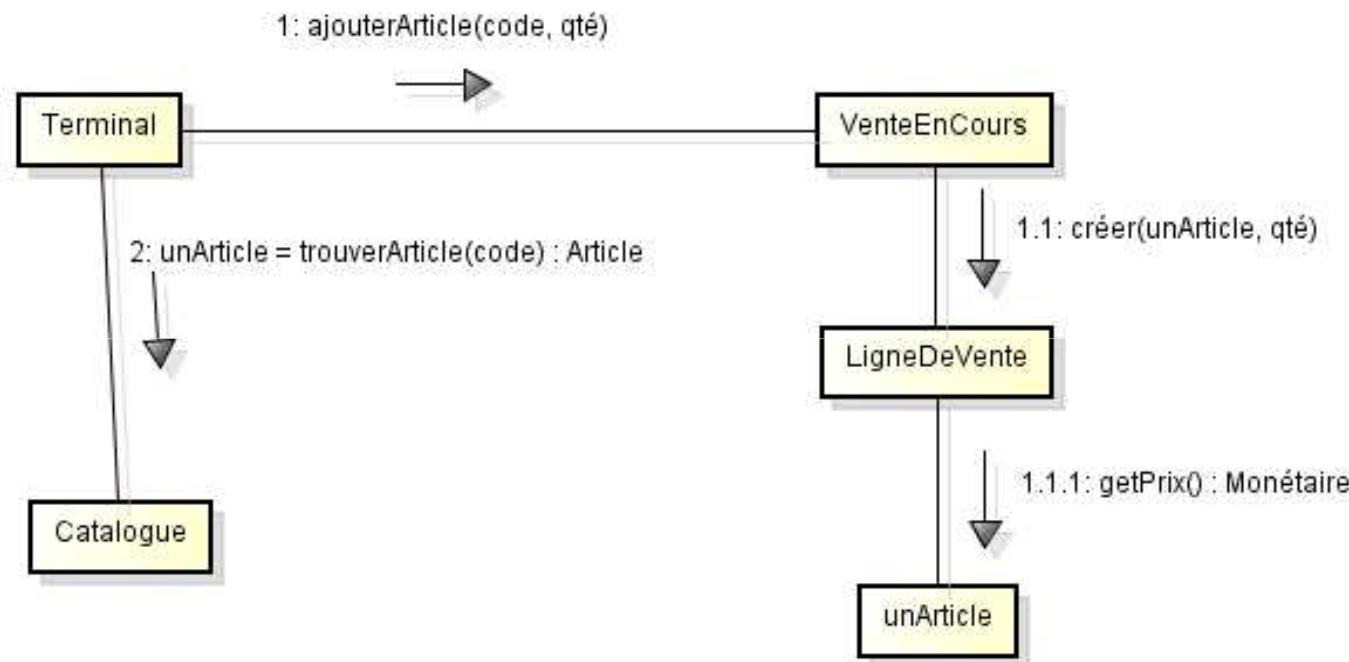


# Boundary, control, entity



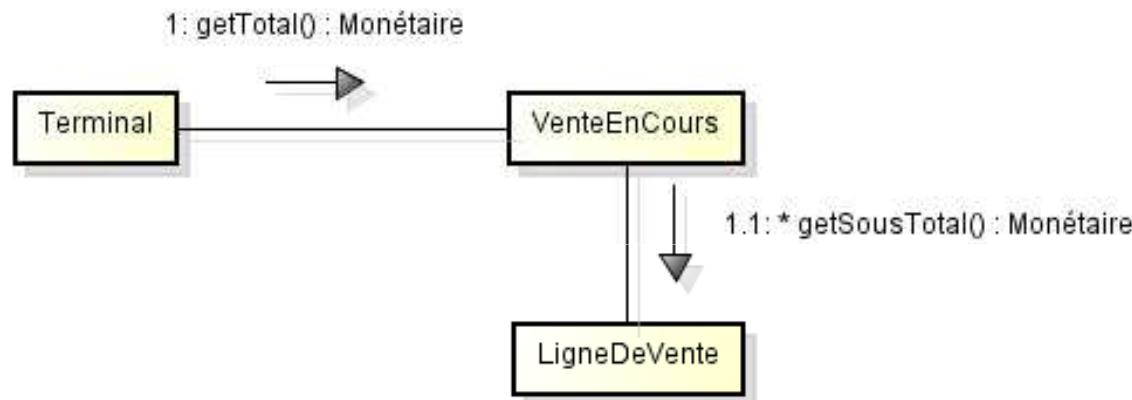


# Numérotation avec numéros mineurs



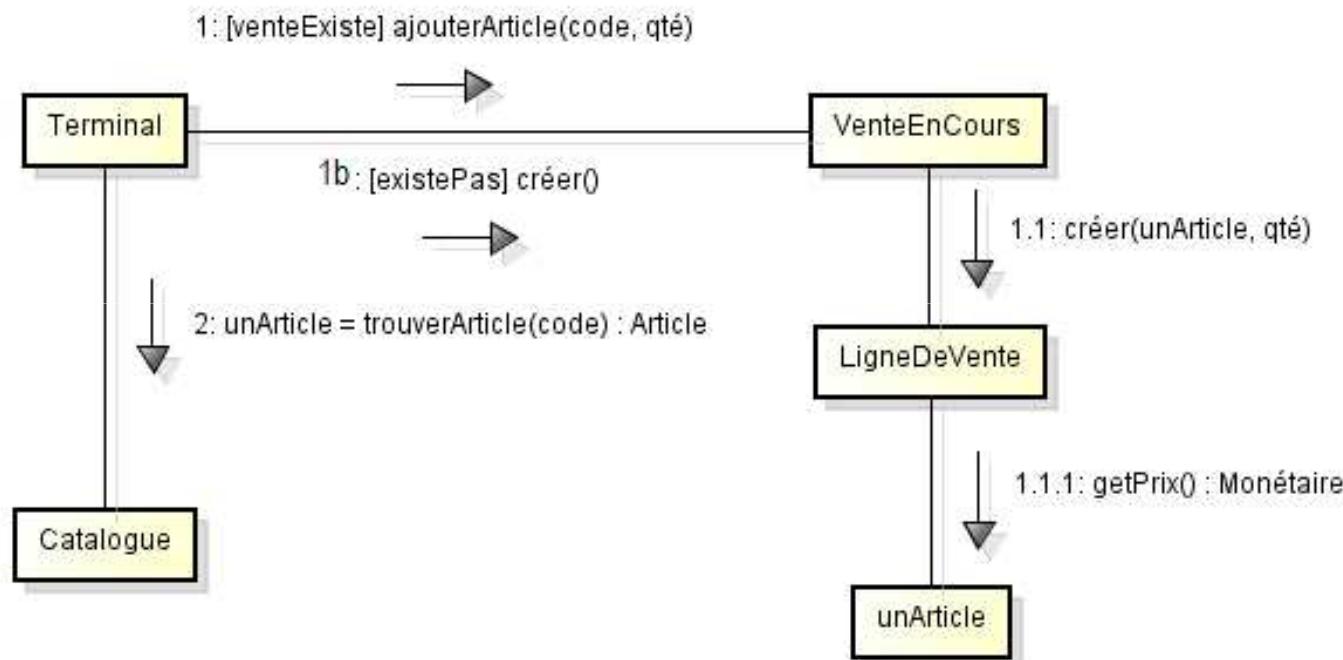


# Itération





# Conditions

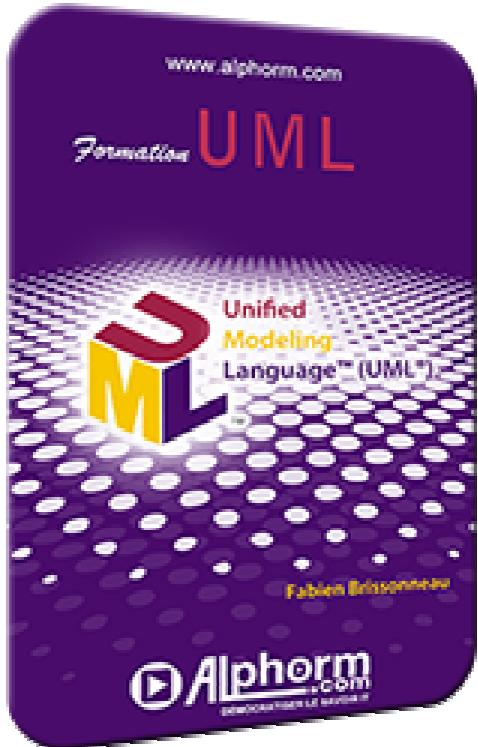


# Ce qu'on a couvert

---

- Les usages du diagramme de communication
- La syntaxe de base : objets, liens, messages
- La syntaxe avancée : boundary, control, entity, numérotation, itération, conditions





Généralités

## Les diagrammes UML

# Le diagramme de temps

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

---

- Usages du diagramme de temps
- Syntaxe de base





# Usages du diagramme de temps

---

- Diagramme de comportement, dynamique
- Ce diagramme est utilisé pour décrire les états des objets, ainsi que les contraintes temporelles entre les états de ces objets



# La syntaxe de base

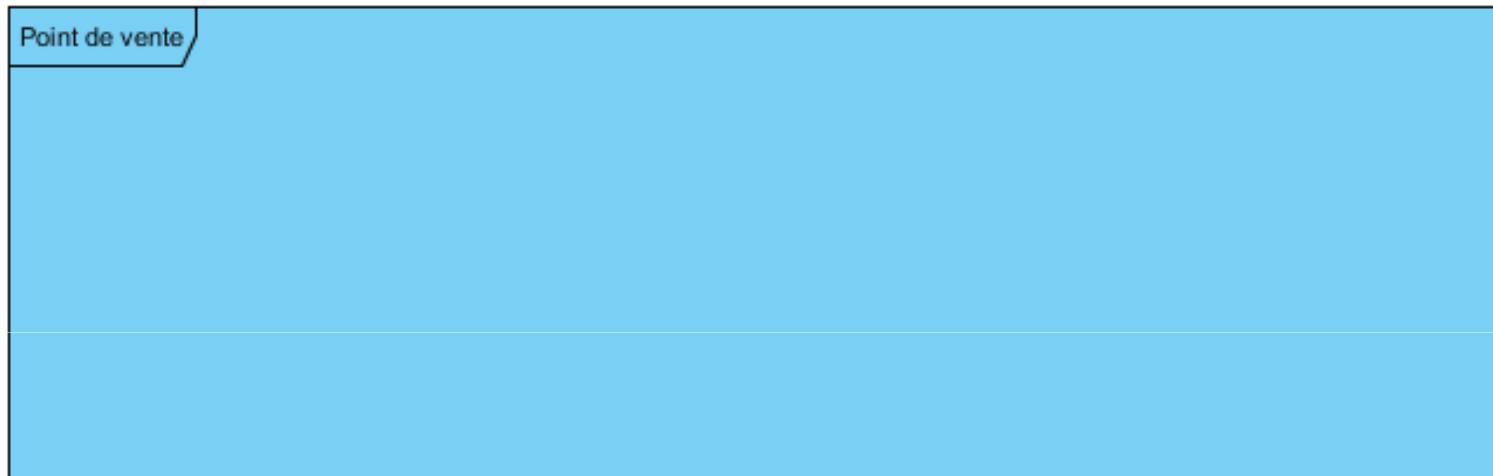
---

- Le cadre
- Les objets
- Les états
- L'échelle de temps
- Contraintes de temps et de durée



# Le cadre

---





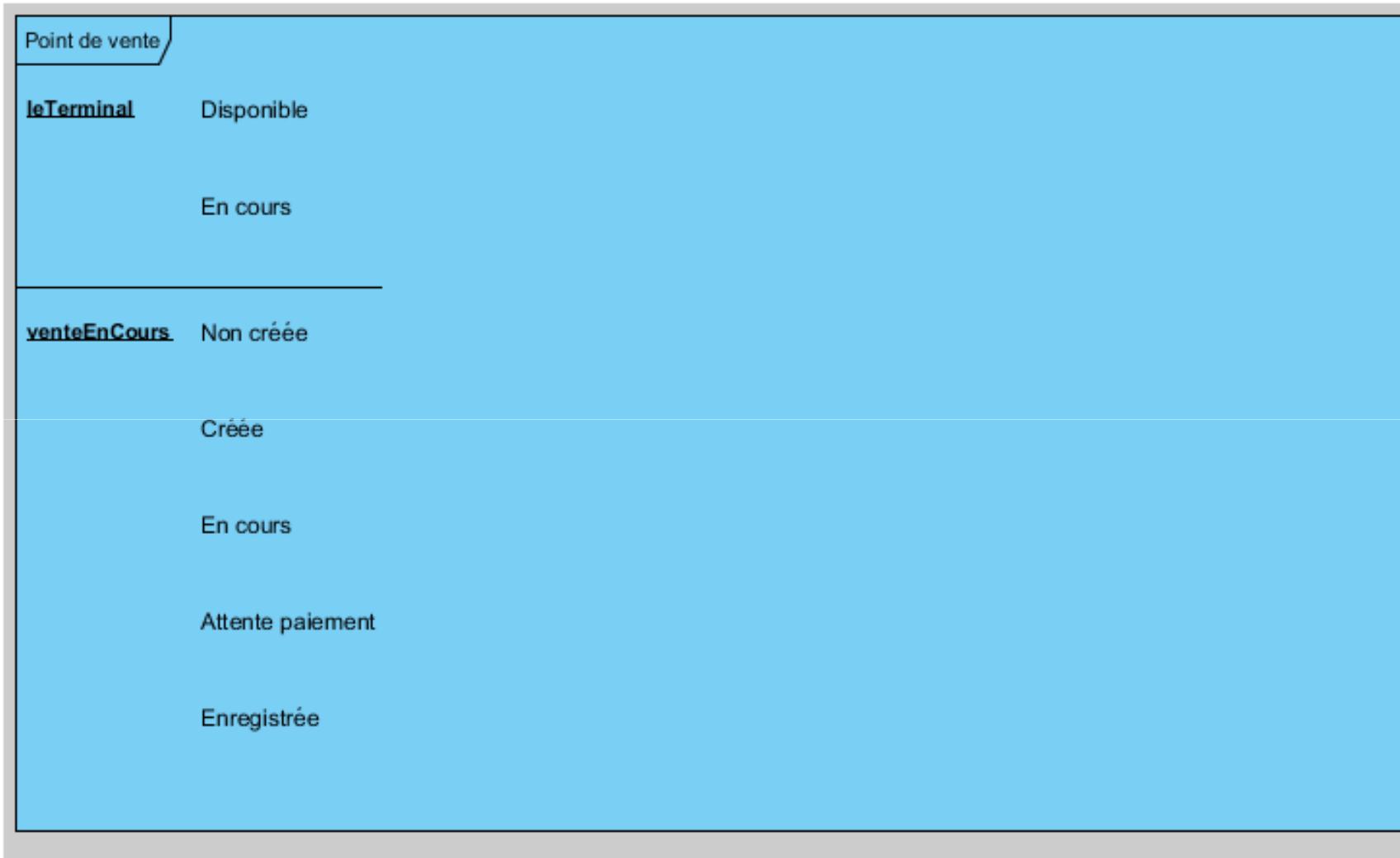
# Les objets

---

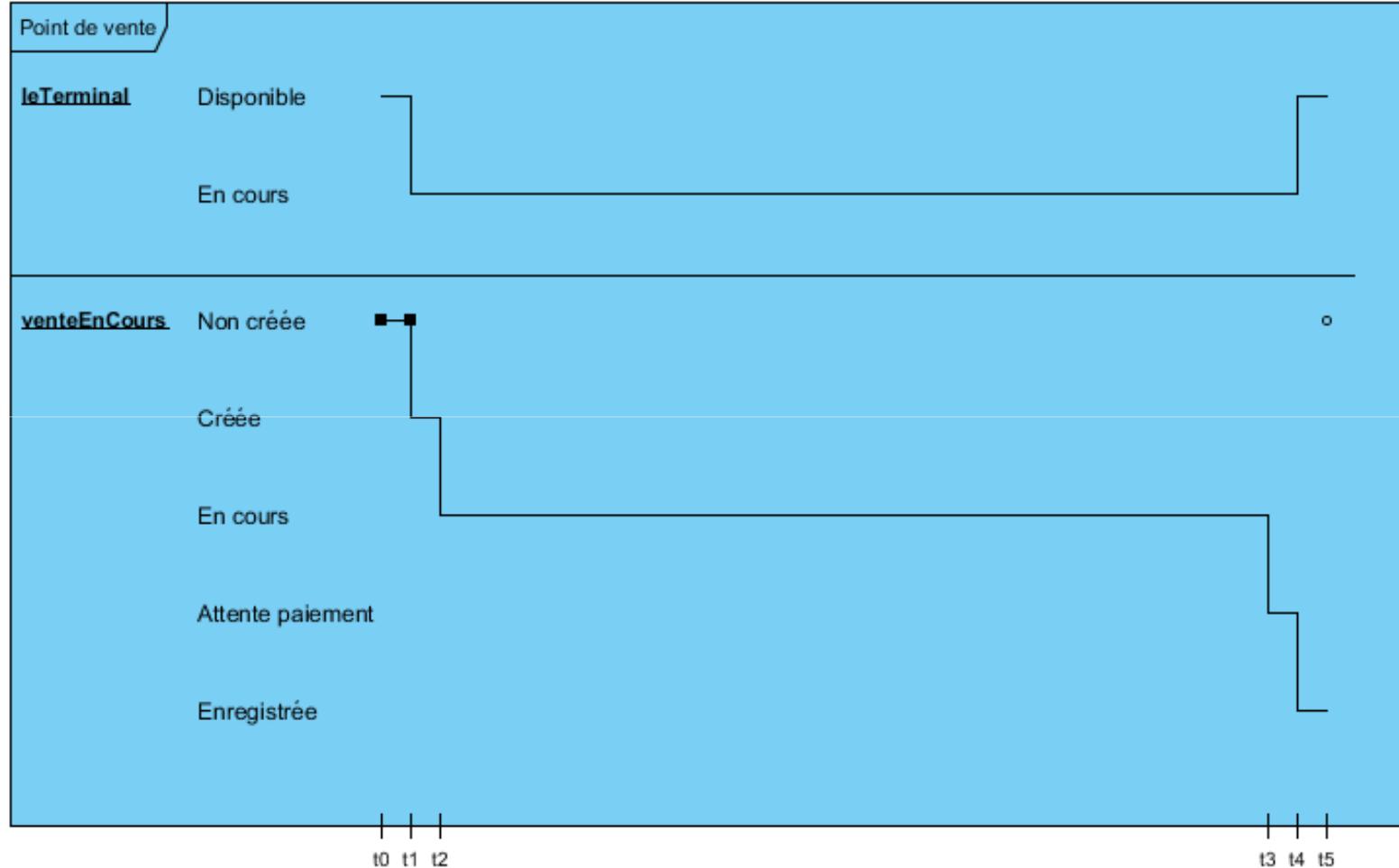




# Les états

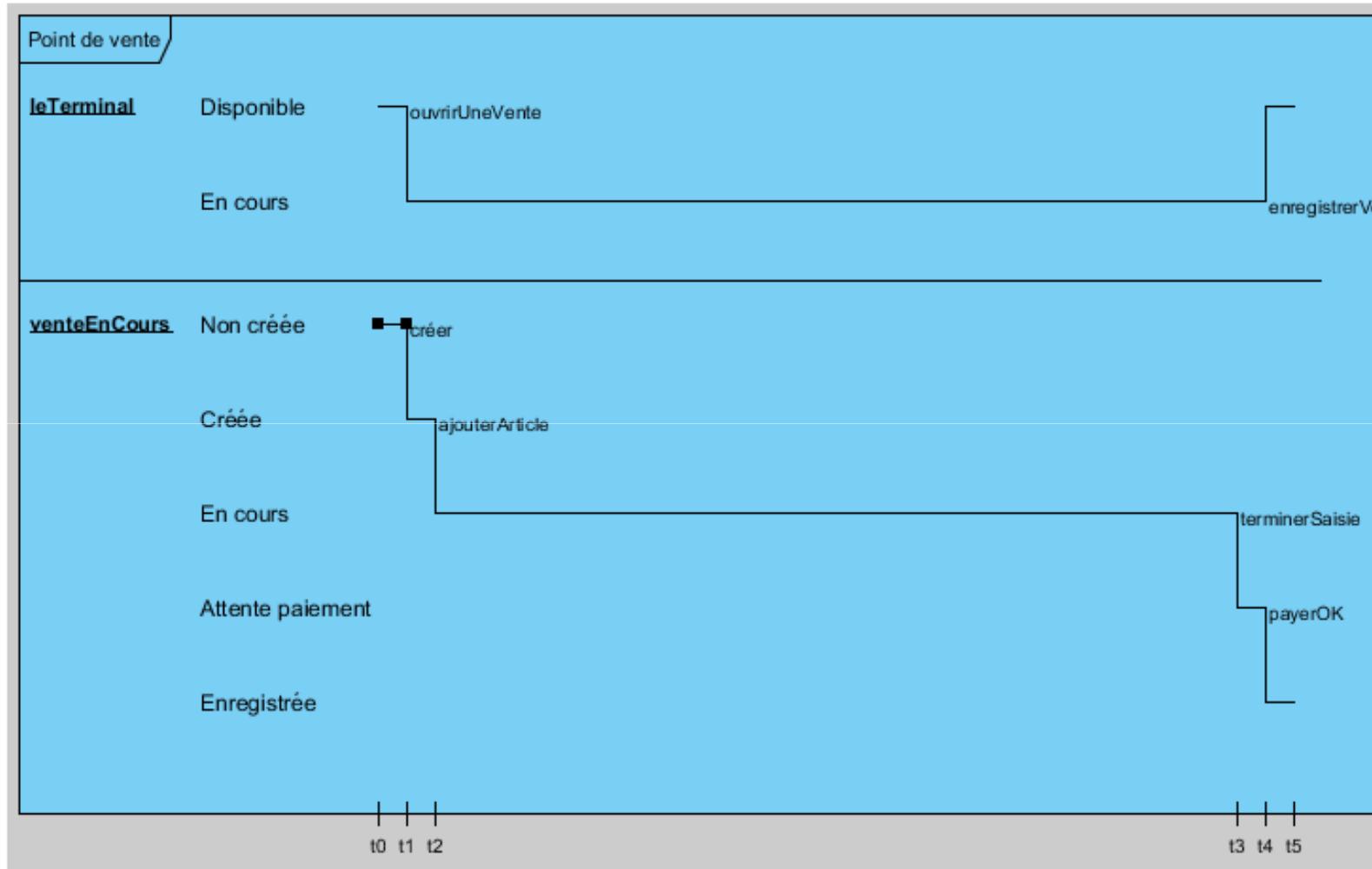


# L'échelle de temps

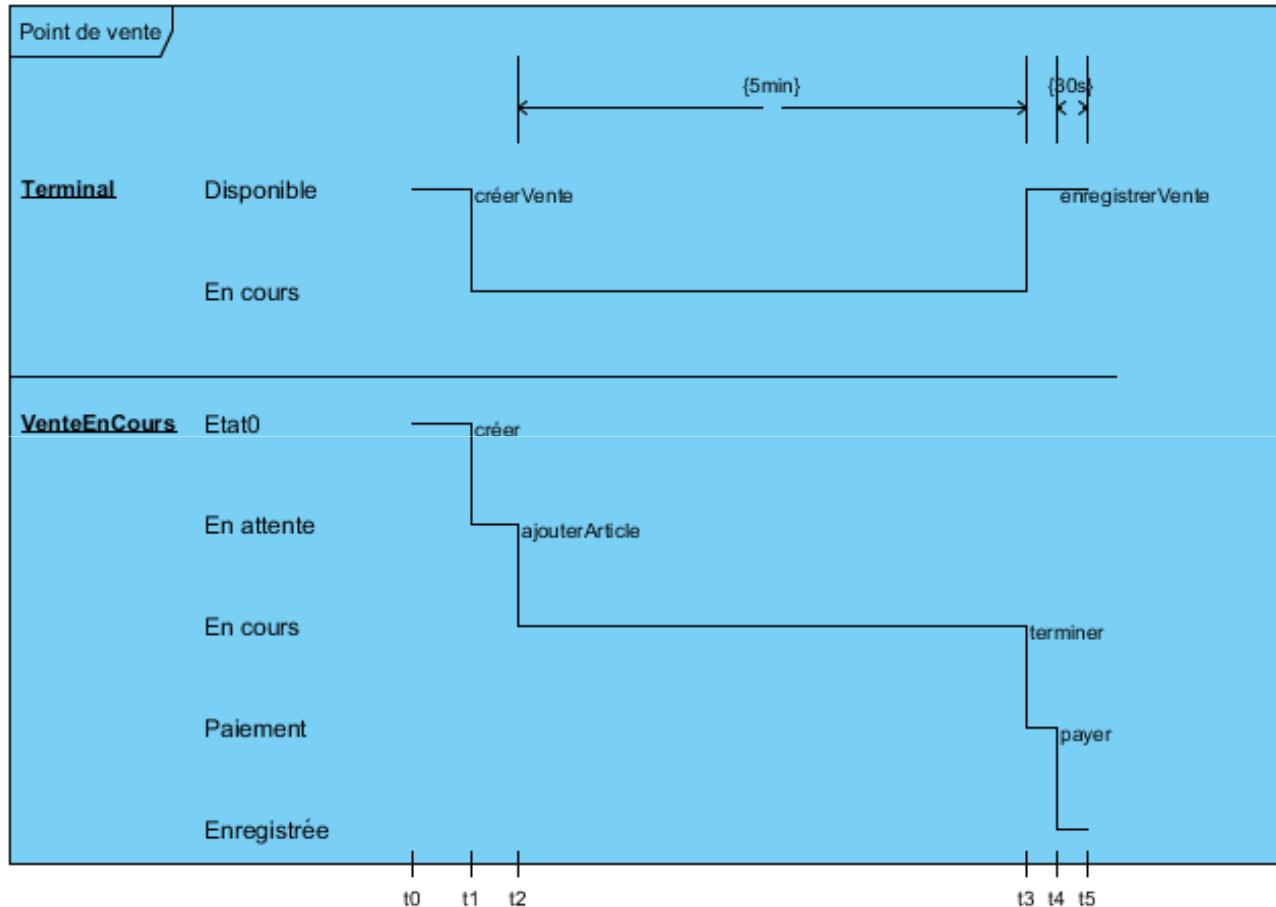




# Stimuli

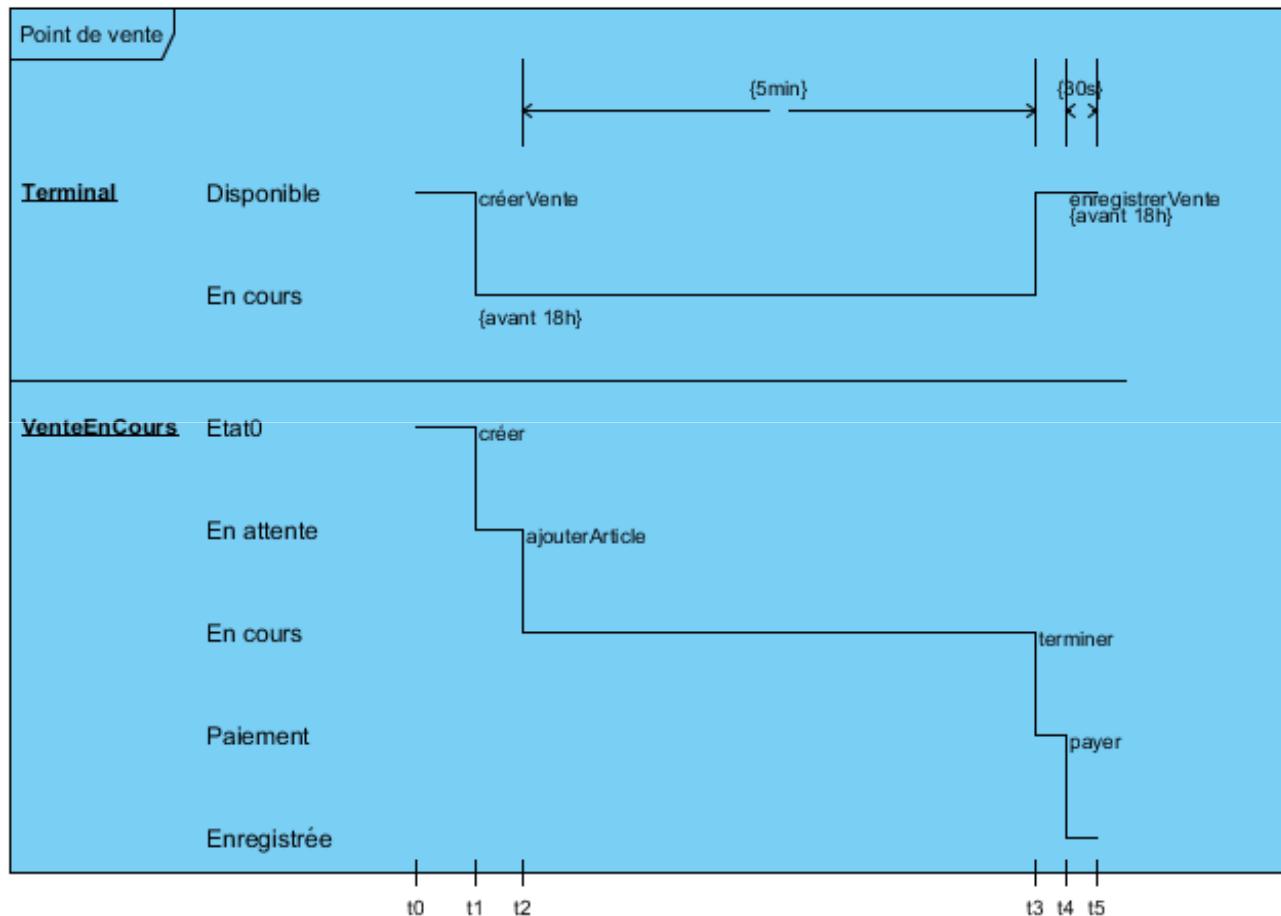


# ML Constraintes de durée





# Contraintes de temps



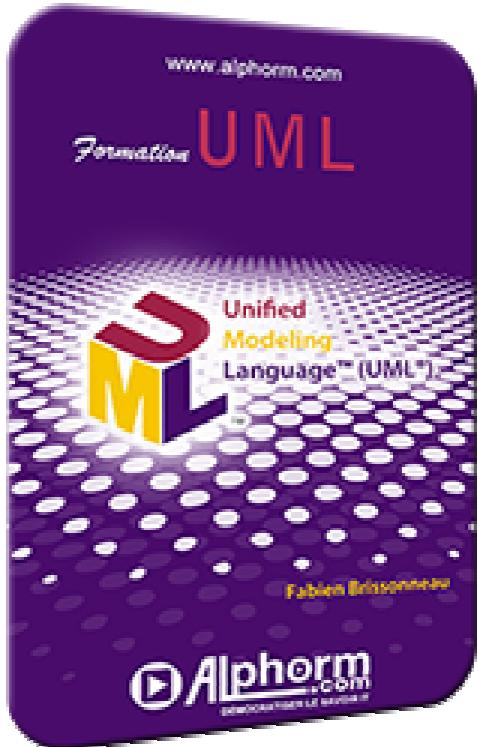


# Ce qu'on a couvert

---

- Les usages du diagramme de temps
- La syntaxe de base : le cadre, les objets, les états, l'échelle de temps, les contraintes de temps et de durée





Généralités

## Les diagrammes UML

Les diagrammes d'objets,  
de vue d'ensemble et de  
profil

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

---

- Usages du diagramme d'objets
- Syntaxe de base
- Usages du diagramme de vue d'ensemble des interactions
- Syntaxe de base
- Usages du diagramme de profil
- Syntaxe de base





# Usages du diagramme d'objets

---

- Diagramme de structure
- Ce diagramme est une instantiation du diagramme de classes
- Sert à donner un exemple de la structure, en particulier lorsque celle-ci est ambiguë
- Finalement peu utilisé, car non exhaustif



# La syntaxe de base

---

- Les objets
- Les liens



# Les objets

---

`leTerminal :`  
`Terminal`

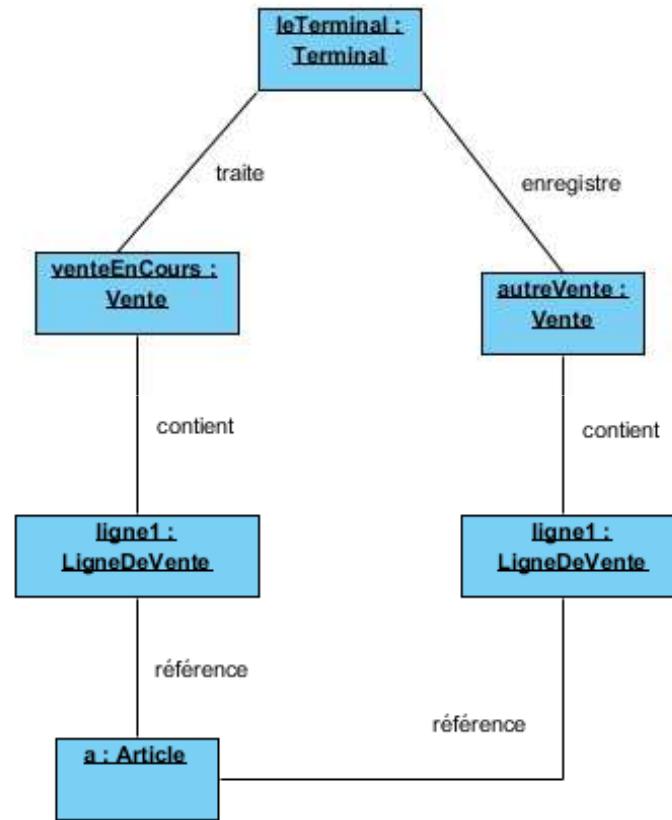
`venteEnCours :`  
`Vente`

`ligne1 :`  
`LigneDeVente`

`a : Article`



# Les liens





# Usages de la vue d'ensemble d'interactions

---

- Diagramme de comportement, dynamique
- Ce diagramme est utilisé pour donner un synoptique des interactions
- Utilise la syntaxe des diagrammes d'activités



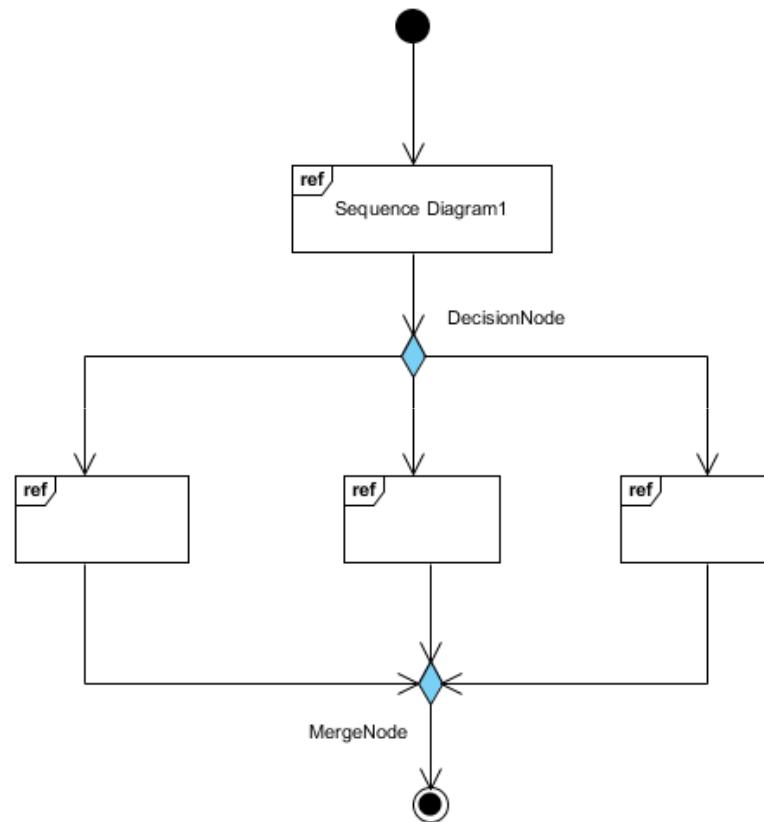
# La syntaxe de base

---

- Syntaxe des diagrammes d'activités, cadres références
- Cadres expansés

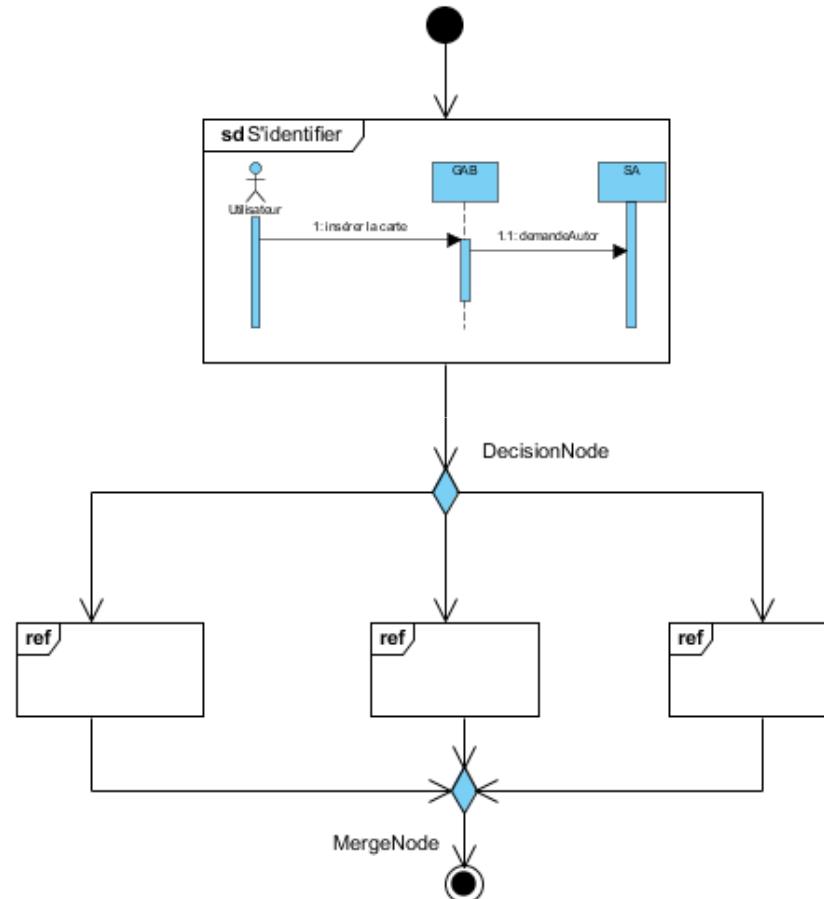


# Cadre REF, syntaxe des activités





# Cadres expasés





# Usage du diagramme de profils

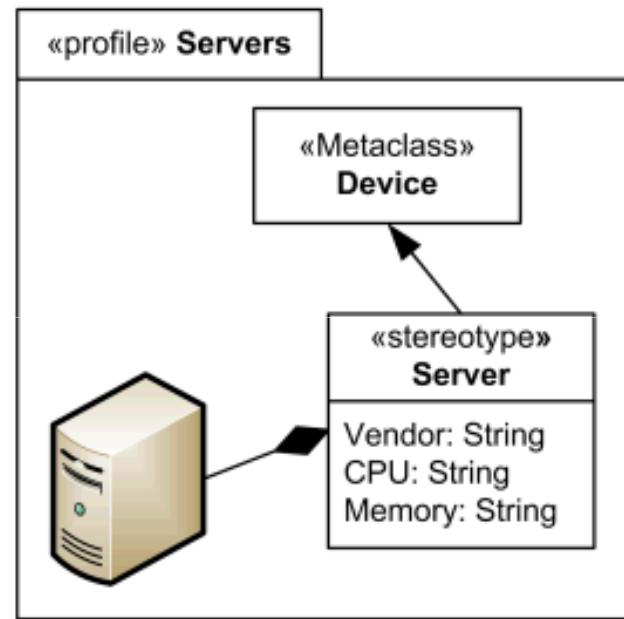
---

- Mécanisme d'extension du métamodèle UML permettant de définir ses propres stéréotypes, taggedvalues et contraintes
- Adapte UML aux plateformes (J2EE, ...) ou aux domaines (temps-réel...)
- Par exemple, dans un profil Java, il faut restreindre l'héritage à une seule branche
- Un profil est un package qui contient des classes, des stéréotypes, des types de données, des types primitifs, des énumérations



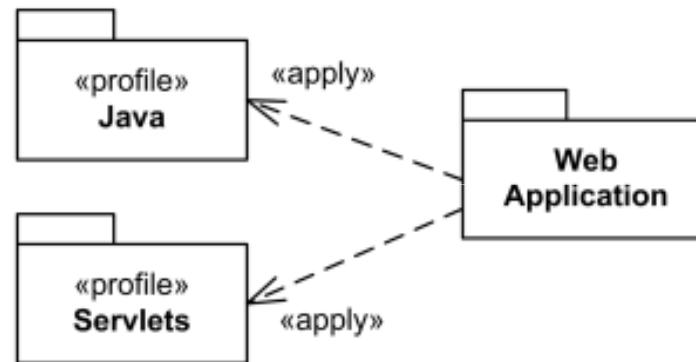
# Syntaxe de base

---





# Syntaxe de base

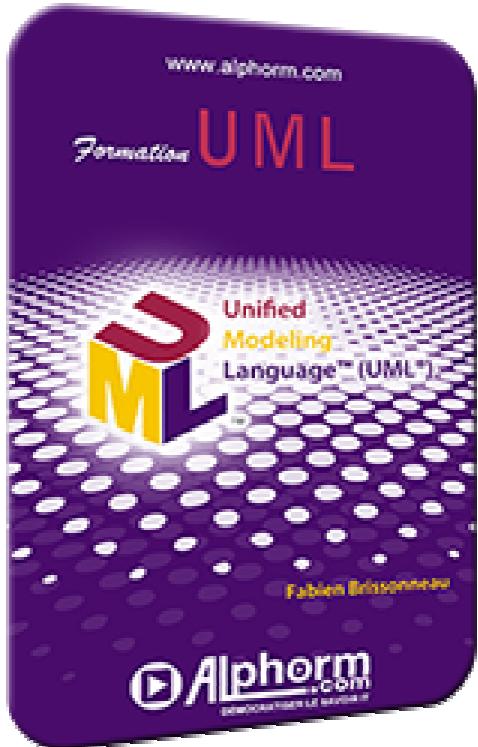


# Ce qu'on a couvert

---

- Les usages du diagramme d'objets
- La syntaxe de base : objets, liens
- Les usages du diagramme de vue d'ensemble des interactions
- La syntaxe de base : cadres REF, cadres expansés, syntaxe des diagrammes d'activités
- Usages du diagramme de profil
- Syntaxe de base





Généralités

## Les diagrammes UML

Le diagramme de  
structures composites

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

Contact : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

---

- Usages du diagramme de structures composites
- Syntaxe de base





# Usages de la vue de structures composites

---

- Diagramme de structure, statique
- Essentiellement un diagramme de classes dédié aux structures composites



# La syntaxe de base

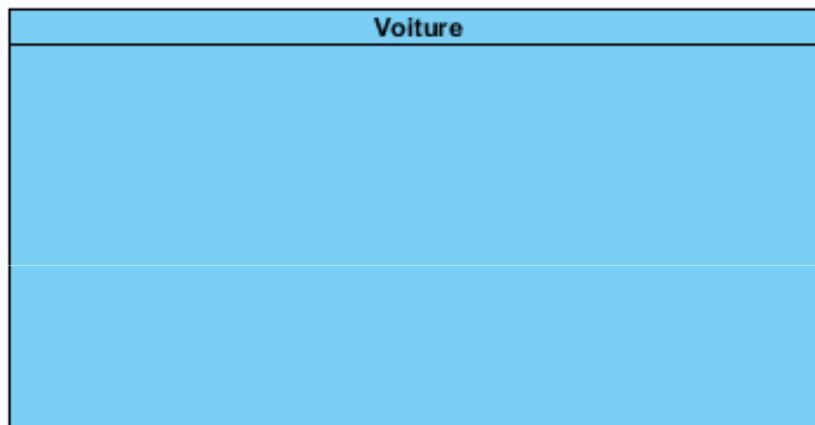
---

- Classe structurée
- Partie
- Port
- Connecteur



# Classe structurée

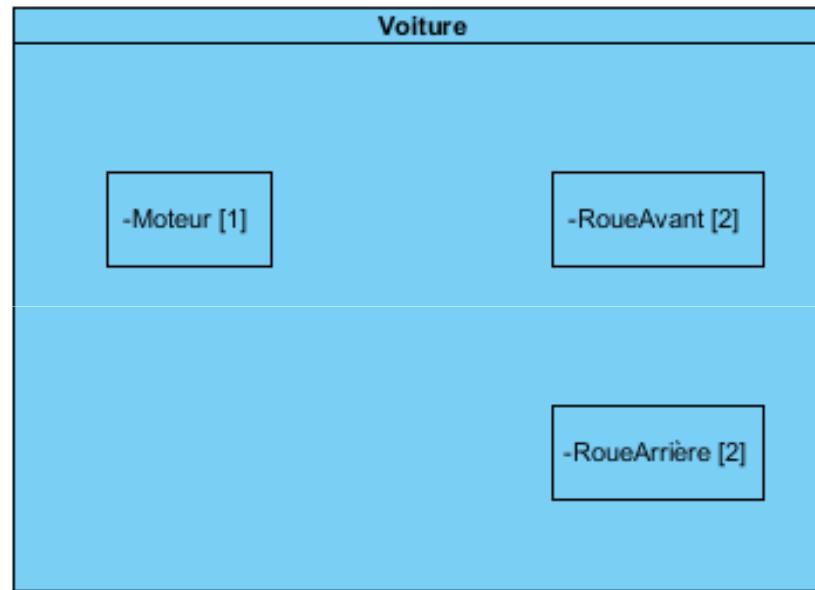
---





# Partie

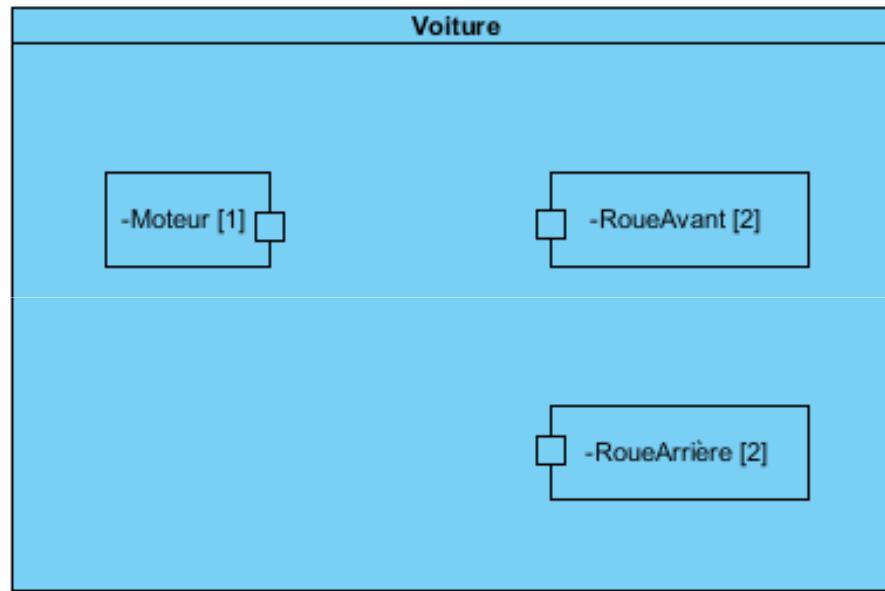
---





# Ports

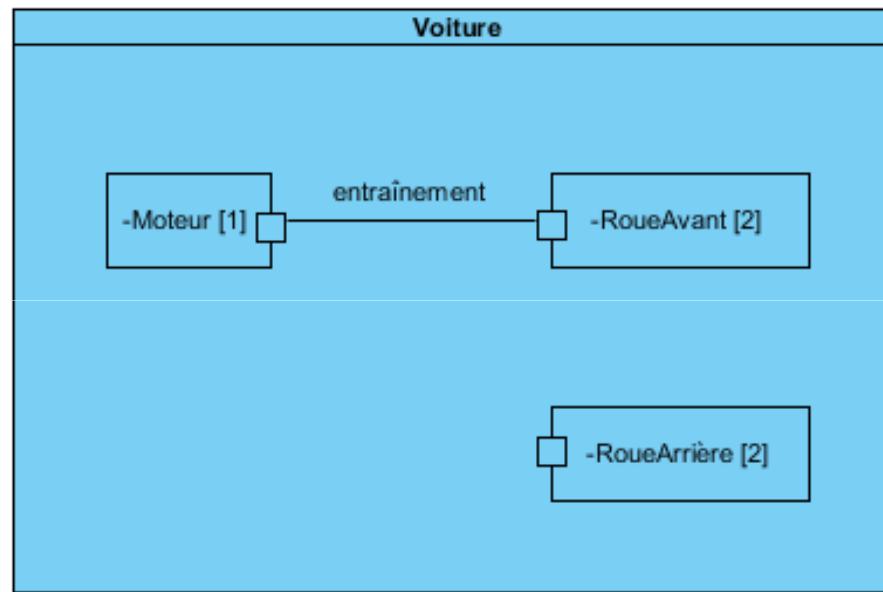
---





# Connecteurs

---



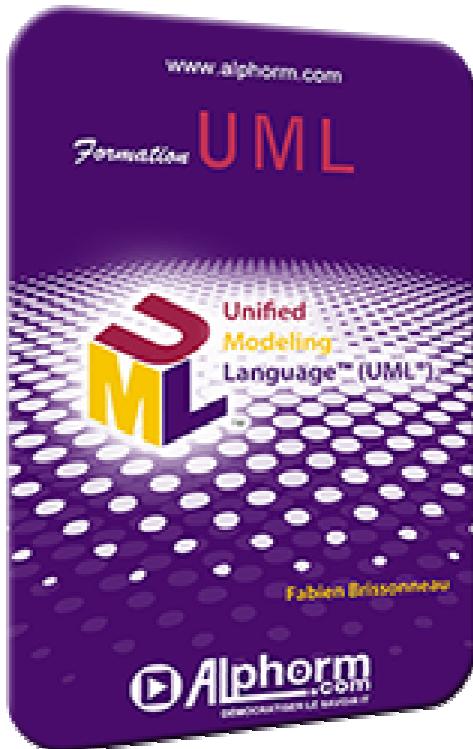


# Ce qu'on a couvert

---

- Les usages du diagramme de structures composites
- La syntaxe de base : classe structurée, partie, port, connecteur





*UML*

UML dans un développement

Utiliser les diagrammes

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

*Formation UML*

Fabien Brissonneau

Consultant, concepteur et formateur

Objets Logiciels

*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Pourquoi faire un diagramme ?
- Qu'est-ce qu'un bon diagramme ?
- Que faire des diagrammes ?
- Organiser un projet UML, utiliser les packages





# Pourquoi faire un diagramme ?

---

- Réaliser une modélisation métier, processus ou objets
- Préciser une expression de besoin
- Décrire un modèle d'analyse
- Décrire un modèle de conception



# Qu'est-ce qu'un bon diagramme ?

---

- Un bon diagramme est un diagramme lisible,
- Adapté aux lecteurs,
  - -> éviter la syntaxe non comprise par les lecteurs
- Conforme à la réalité
  - -> éviter les idées géniales



# Que faire des diagrammes ?

---

- Les diagrammes, quels problèmes, quels outils
- Relation avec le texte
- Ne pas utiliser tous les diagrammes
- Les diagrammes doivent rester cohérents avec le système



# Organiser un projet UML

---

- Les packages permettent de structurer le projet
- Les packages « modèles » sont les étapes du processus
- Les packages de classes regroupent les classes

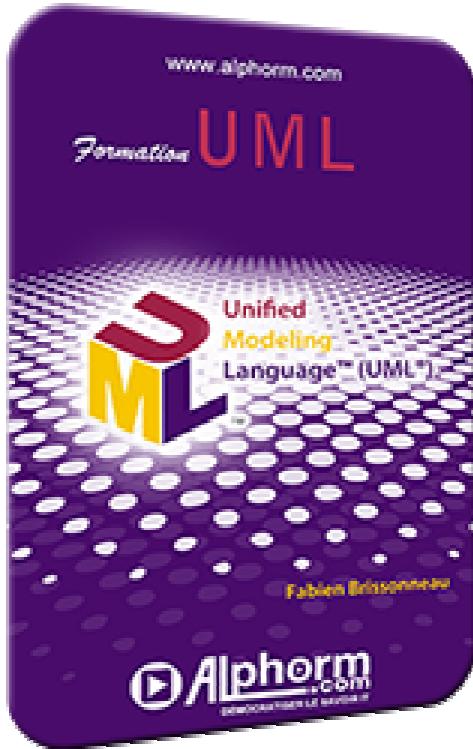


# Ce qu'on a couvert

---

- Pourquoi faire un diagramme ?
- Qu'est-ce qu'un bon diagramme ?
- Que faire des diagrammes ?
- Organiser un projet UML, utiliser les packages





*UML*

UML dans un développement

UML et le processus

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

---

- Définition du processus
- Place de la conception
- Quels diagrammes pour quelles étapes ?
- UML dans les outils





# Processus

---

- Ensemble ordonné d'étapes visant la réalisation d'un objectif.
- Réussite en terme de budget, délais, fonctionnalités... traçabilité ...



# Mettre UML en œuvre

**BM**

**BPM : Activités, Séquences**

**BOM : Classes**

**Besoins**

**Cas d'utilisation**

**Activités**

**Analyse**

**Classes**

**Séquence, Etats**

**Conception**

**Classes**

**Communication**



# Le manifeste de l'Agilité

---

**Manifesto for Agile Software Development**

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions** over processes and tools
- Working software** over comprehensive documentation
- Customer collaboration** over contract negotiation
- Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.



# Scrum : les rôles

---

- Product Owner : représentant du client et des utilisateurs. Définit l'ordre d'implémentation des fonctionnalités.
- Équipe : auto-gérée. Pas de hiérarchie.
- Scrum Master : protège l'équipe et gère les problèmes non-techniques.
- Stakeholders : intervenants non techniques souhaitant avoir une vue sur le projet, sans s'investir.



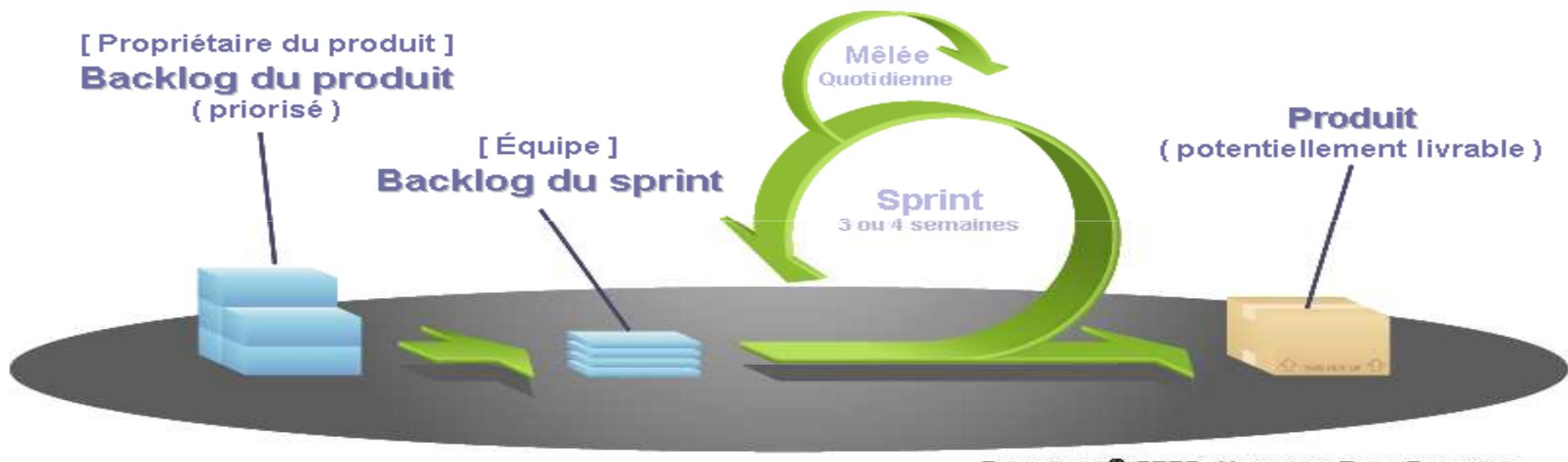
# Planification

---

- Release/projet, Sprint, Quotidien
  - Sprint : itération
  - Release : produit livrable
  - Quotidien : réunion « Scrum Meeting »
- 
- Backlog de produit : fonctionnalités
  - Baklog de sprint



# Scrum





# eXtreme Programming

---

- Planning
  - User Stories, petites releases fréquentes, itérations démarrant par planning
- Management
  - Equipe confortable, standup meeting chaque jour, mesure de l'avancement du projet
- Conception
  - Favoriser la simplicité, refactoriser souvent
- Codage
  - Standard, tests first, par paire, intégrations fréquentes
- Tests
  - Tests unitaires, bug=>test créé



# UML dans les outils

---

- Structurer un projet UML
- Que produire avec un outil ?
- Partager un référentiel

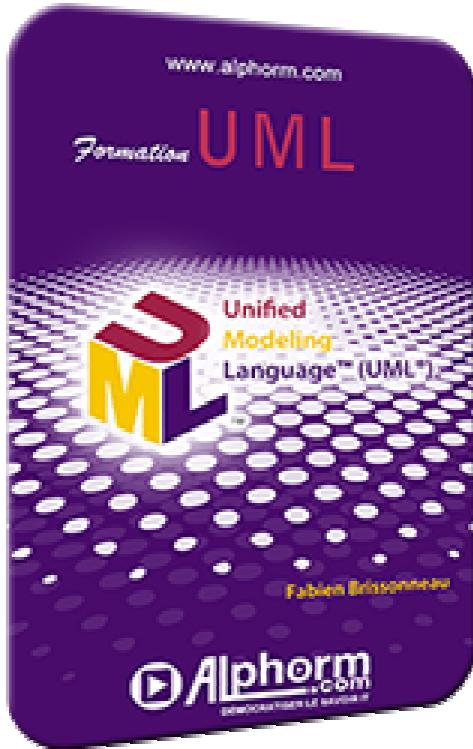


# Ce qu'on a couvert

---

- Définition du processus
- Place de la conception
- Quels diagrammes pour quelles étapes ?
- UML dans les outils





*UML*

## UML dans un développement

# L'enchaînement entre les diagrammes UML

Fabien Brissonneau

Consultant, concepteur et formateur  
Objets Logiciels

*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML



# Plan

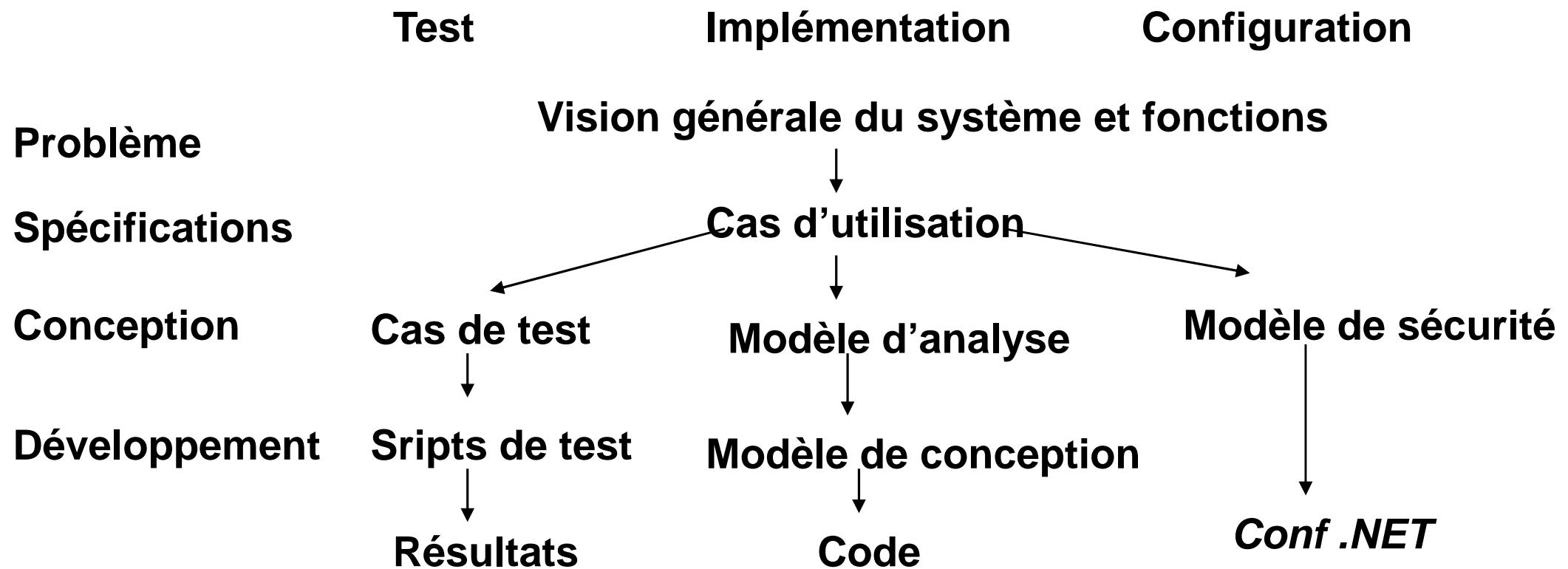
---

- Description du métier
  - Processus
  - Objets
- Description des besoins
- Description de l'analyse
- Description de la conception
- Vers le code et la base de données



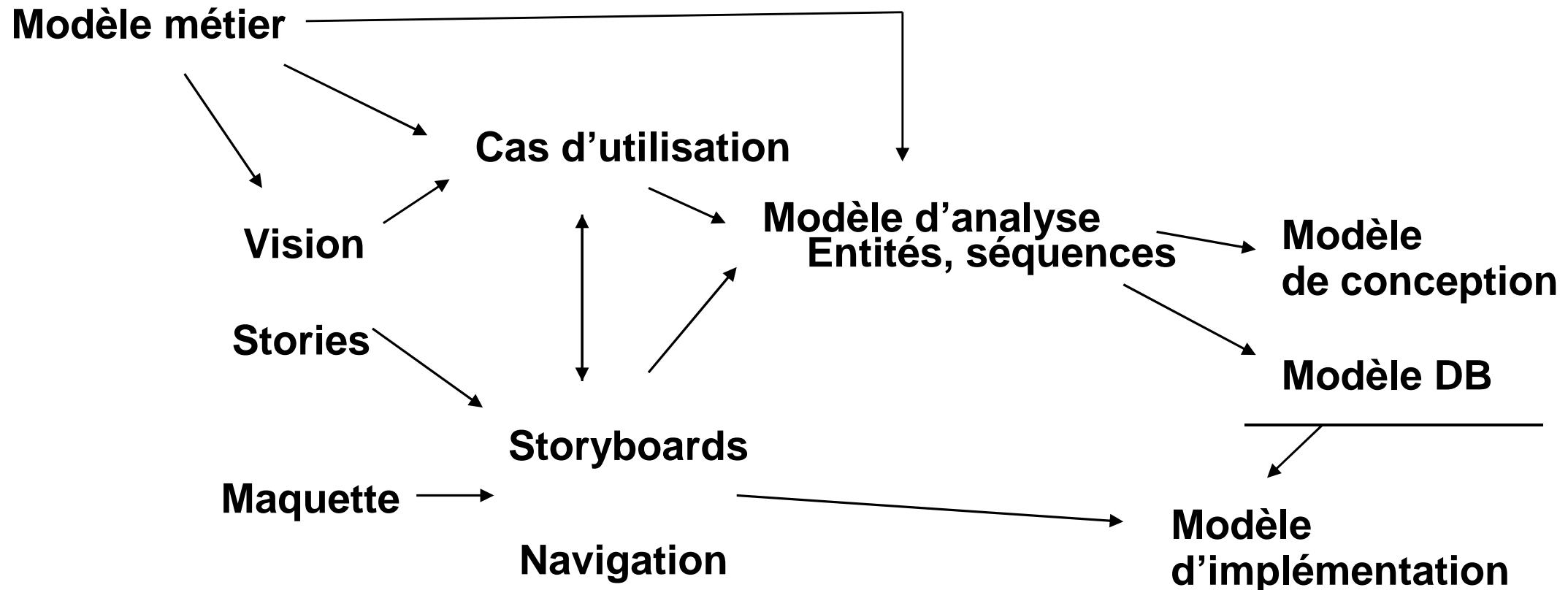


# Traçabilité



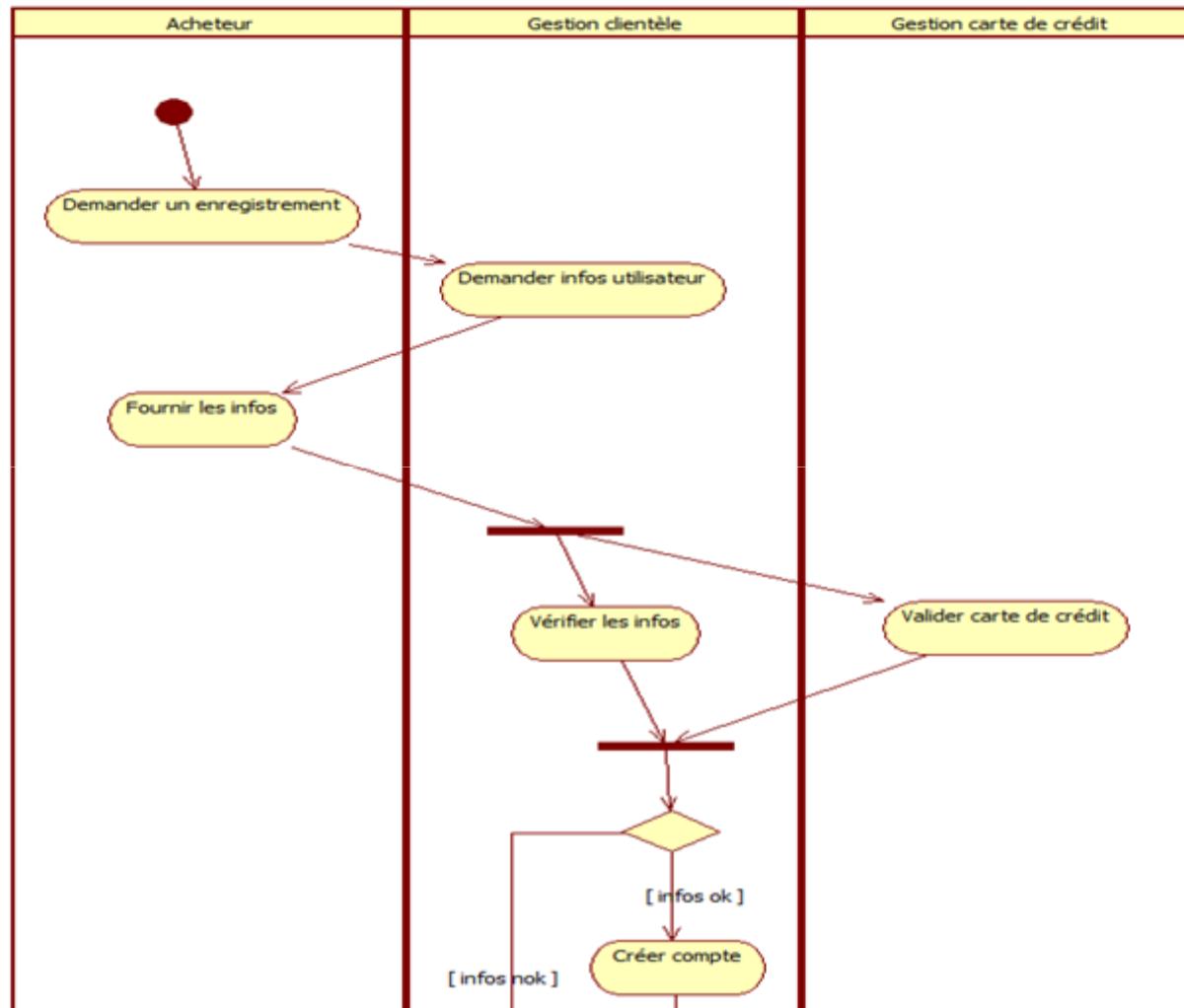


# Les processus



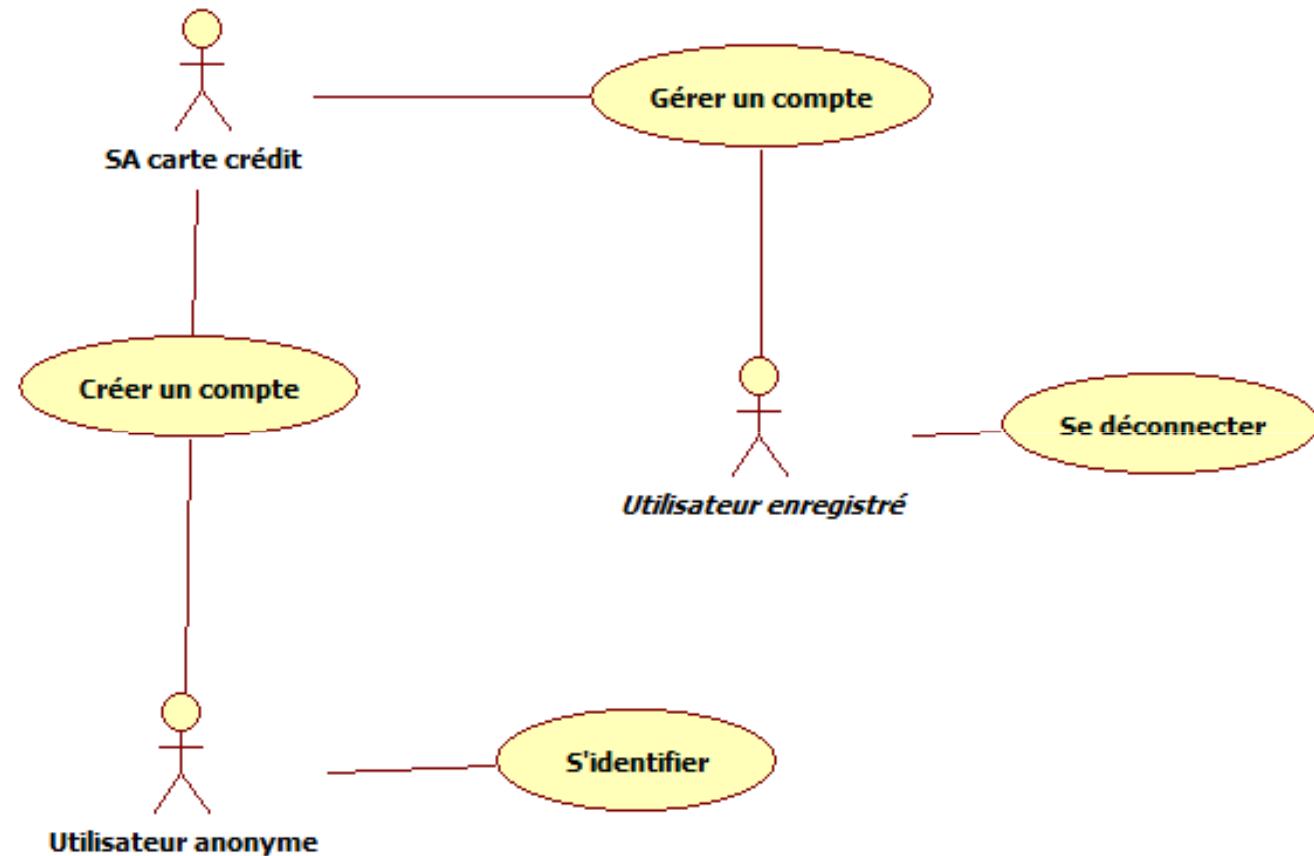


# Description du métier



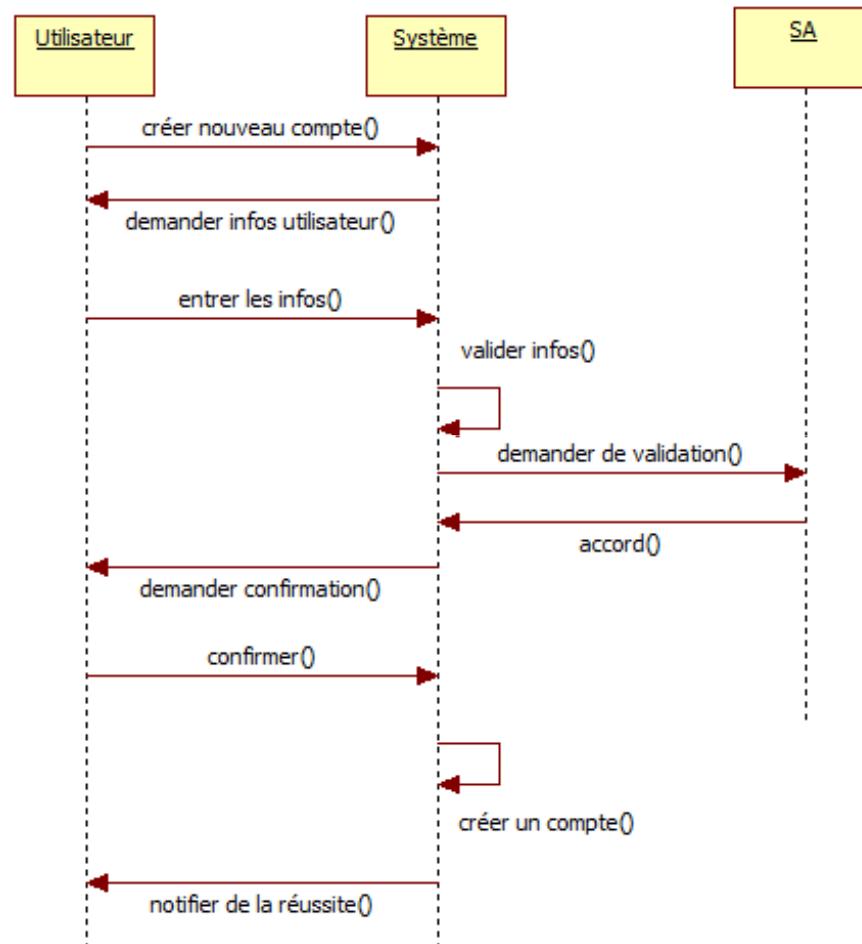


# Description des besoins



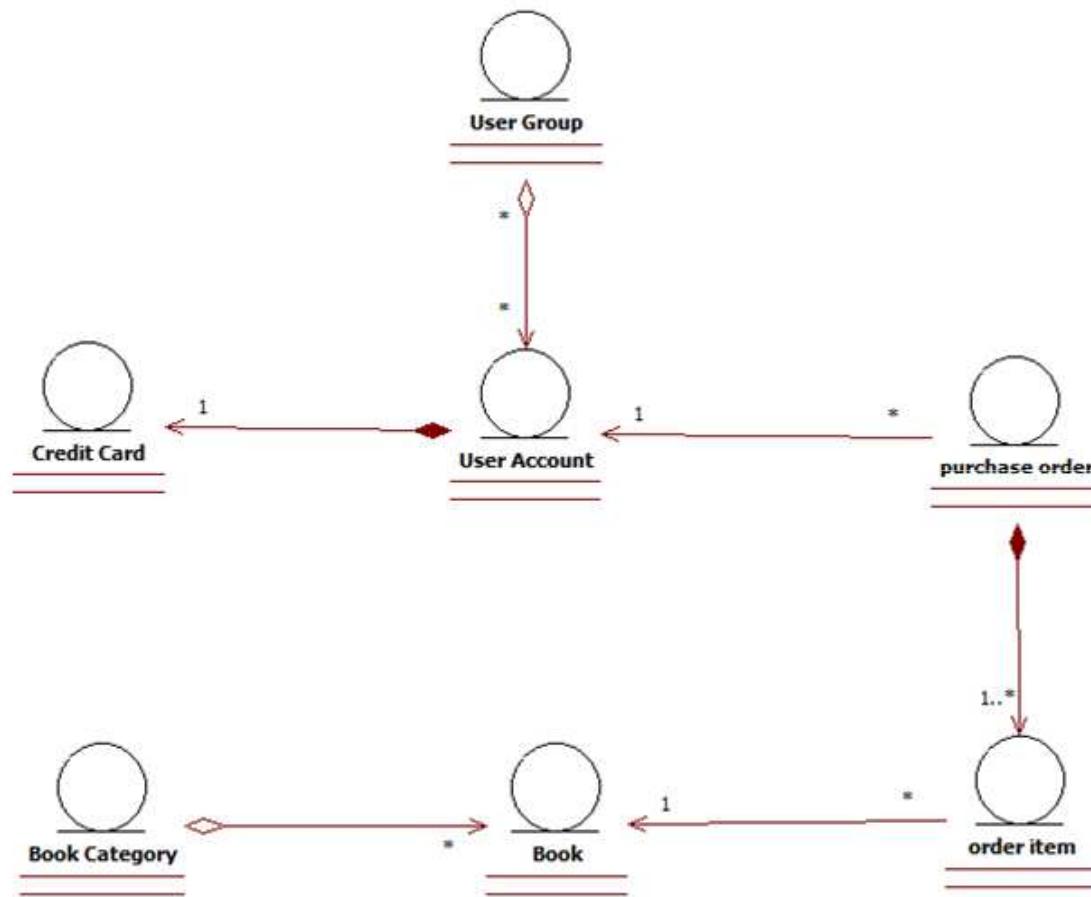


# Description des besoins



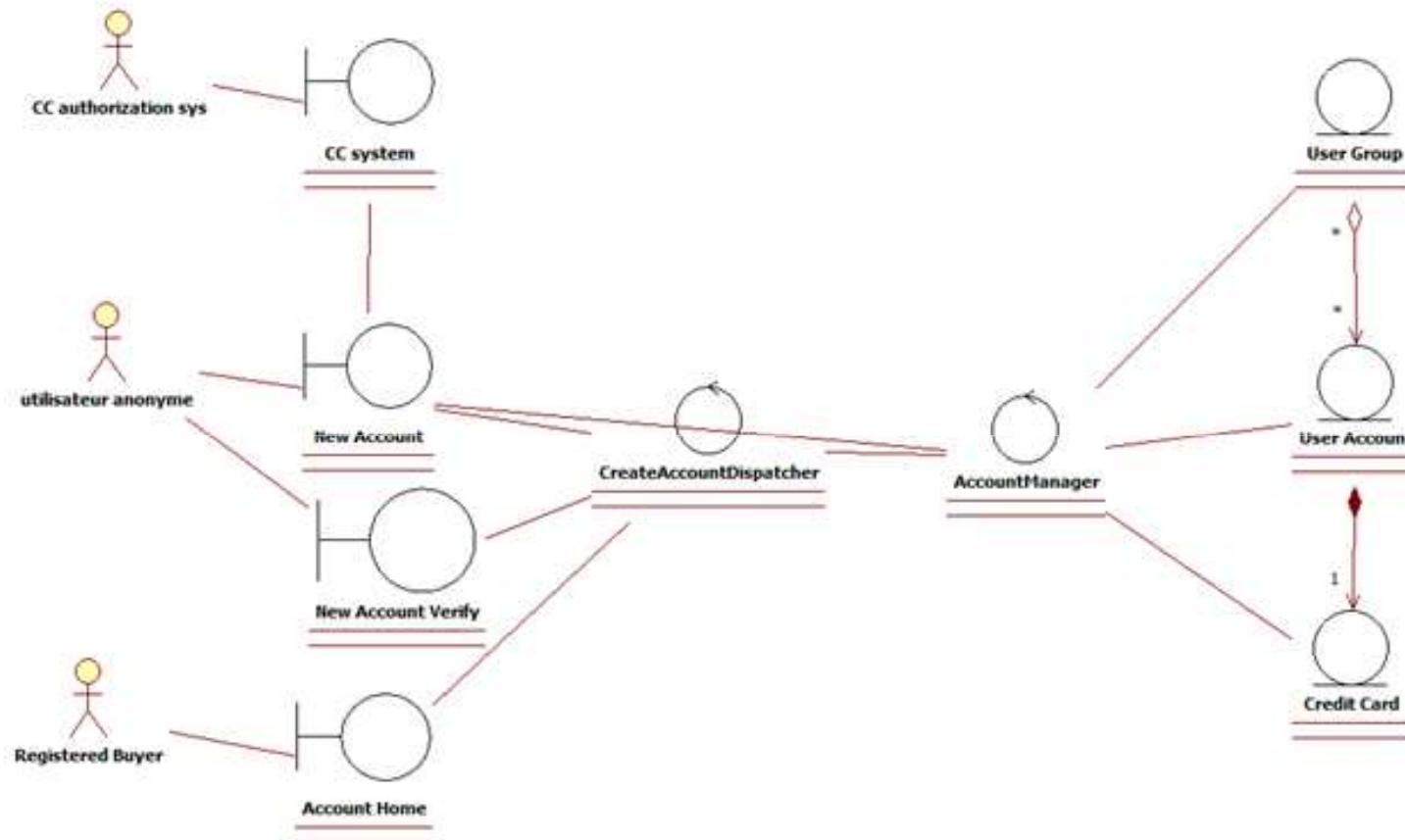


# Description de l'analyse



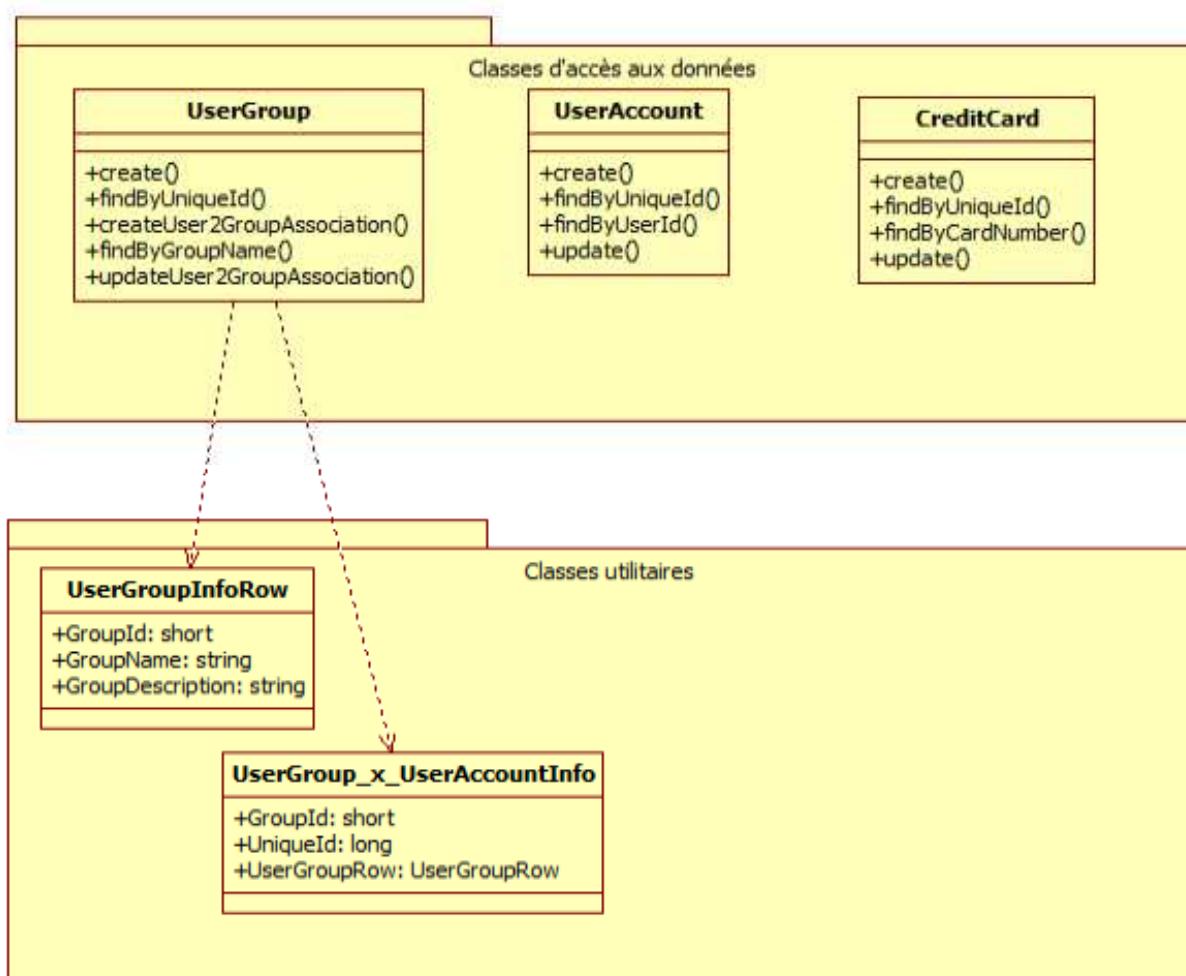


# Description de la conception



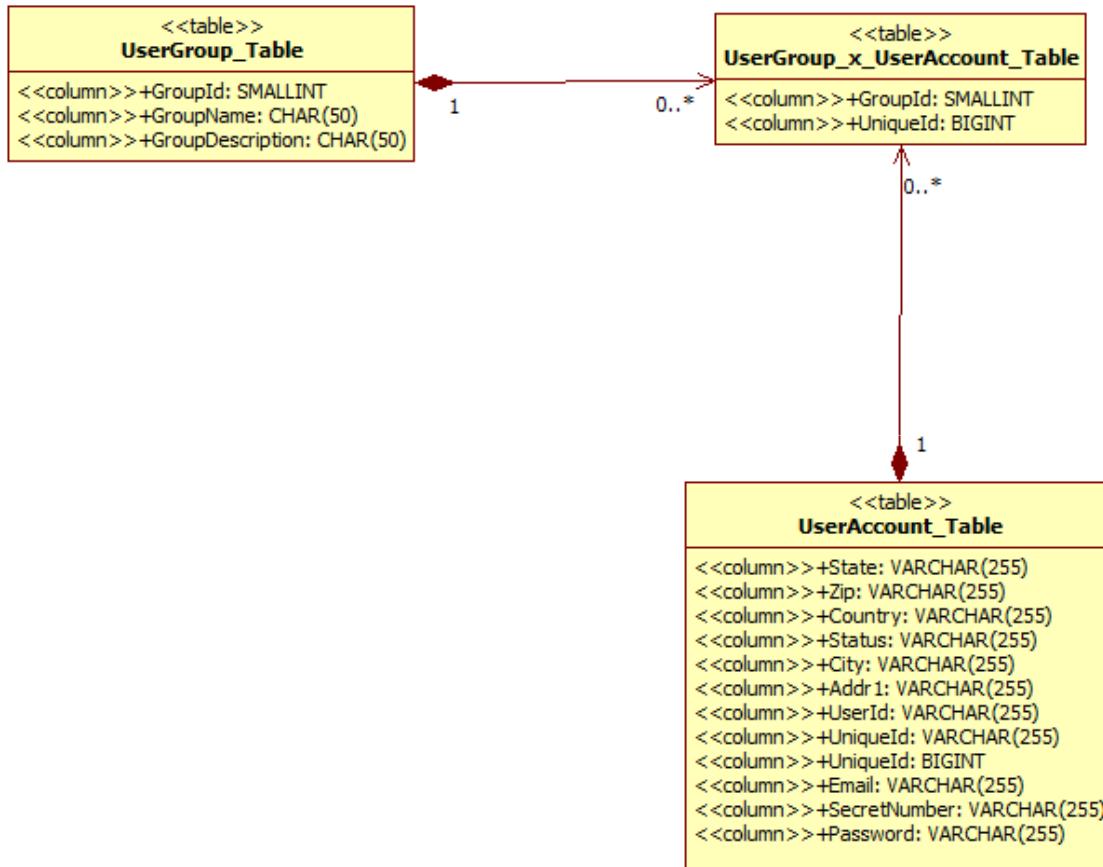


# Vers le code et la base ...





# Vers le code et la base ...



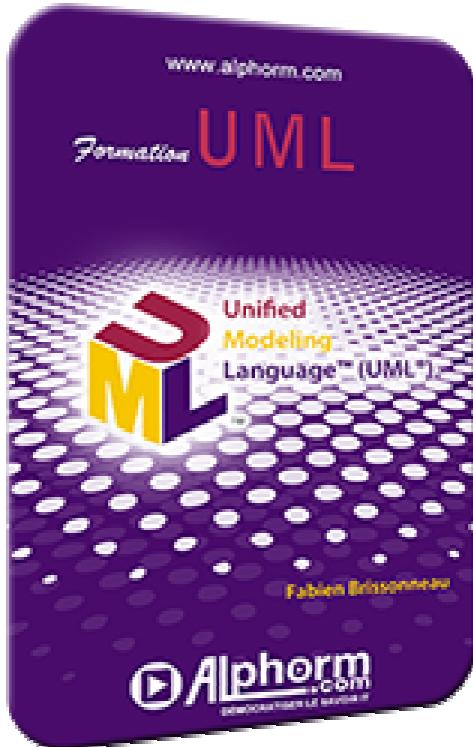


# Ce qu'on a couvert

---

- Description du métier
- Description des besoins
  - Processus
  - Objets
- Description de l'analyse
- Description de la conception
- Vers le code et la base de données





*UML pour la maîtrise d'ouvrage*  
Description des processus « métier »  
**Processus « métier »**

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

*Formation UML*

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Choix d'un outil
- Organisation du projet
- Présentation de l'étude de cas
- Recherche et schématisation des processus métier





# Choix d'un outil

---

- Astah
  - Version communautaire
  - Bon support de la norme UML
  - Présence des stéréotypes métiers



# Organisation du projet

---

- Projet UML
  - Utiliser le package stéréotypé « model »
  - Créer un modèle « métier »



# Présentation de l'étude de cas

---

- Thème : la gestion d'une librairie
  - Chercher les acteurs métier
    - Client, Bibliothécaire, Editeur, Banque, ...
  - Chercher les processus métier
    - Vendre des livres, Acheter des livres d'occasion, Mettre à jour le catalogue, Mettre à jour le stock, Proposer des événements, ...



# Schématisation des processus métier

---

- Le diagramme des processus
  - Diagramme de cas d'utilisation
  - Stéréotype Business UseCase

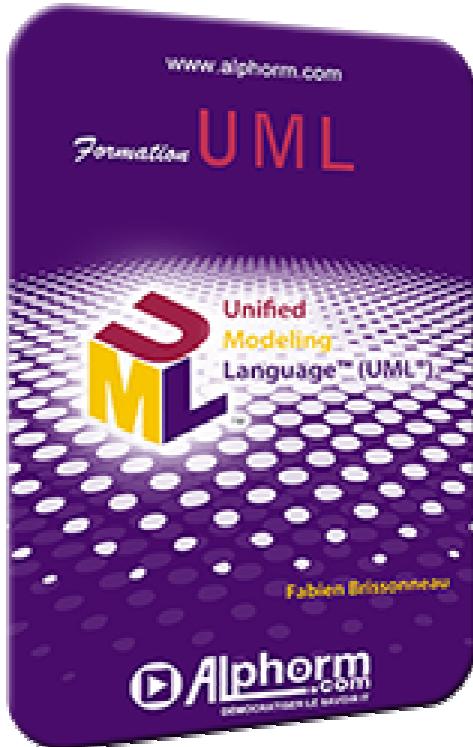


# Ce qu'on a couvert

---

- Définition des processus métier
- Présentation d'un outil
- Représentation des processus métier sur un cas





*UML pour la maîtrise d'ouvrage*  
Description des processus « métier »

# Décomposer un processus « métier »

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Choix d'un processus « métier »
- Le diagramme d'activité
- La syntaxe du diagramme d'activité





# Choix d'un processus « métier »

---

- Le modèle « métier »
  - Dans l'outil, un package
  - Organisation des processus métier dans le modèle



# Le diagramme d'activité

---

- Création du diagramme
  - Sous le processus
  - Au moins un diagramme par processus



# La syntaxe du diagramme d'activité

---

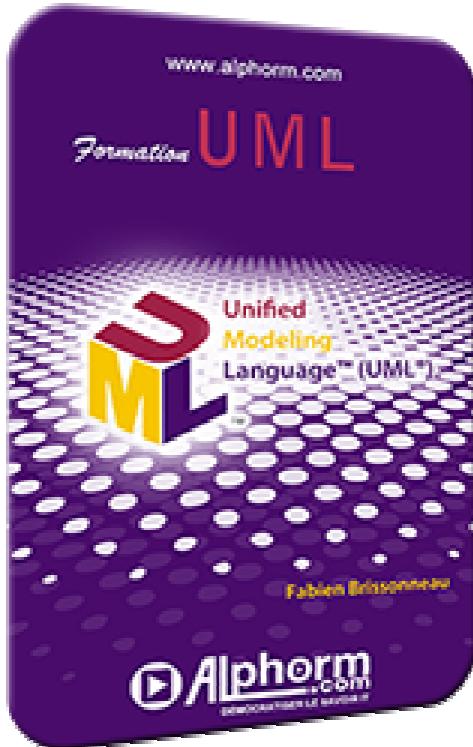
- Le diagramme d'activité
  - Actions, activités
  - Initial, final
  - Décisions
  - Fork

# Ce qu'on a couvert

---

- La description d'un processus se fait par un diagramme d'activité
- La syntaxe de base est très simple
- Ce diagramme permet de vérifier la complétude du processus et de se poser les bonnes questions





*UML pour la maîtrise d'ouvrage*  
Description des processus « métier »

Utiliser un diagramme  
de séquences

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Choix d'un processus « métier »
- Le diagramme de séquences
- La syntaxe du diagramme de séquences





# Choix d'un processus « métier »

---

- Le modèle « métier »
  - Dans l'outil, un package
  - Les interactions entre acteurs



# Le diagramme de séquences

---

- Création du diagramme
  - Sous le diagramme d'activité
  - Un diagramme de séquence par scénario du processus



# La syntaxe du diagramme de séquences

---

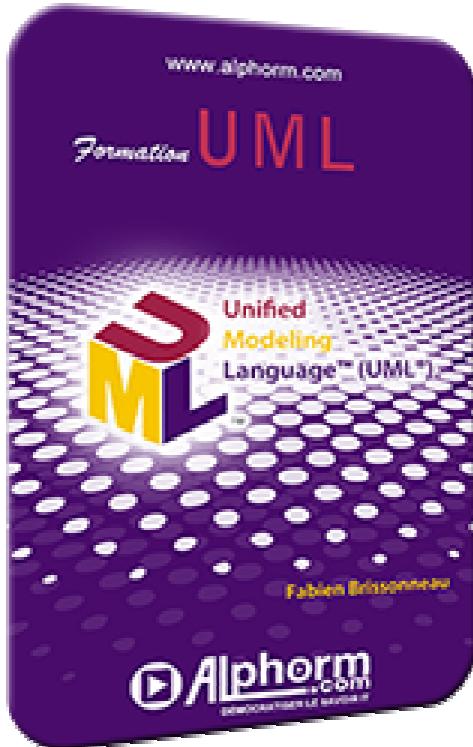
- Le diagramme de séquences
  - Ligne de vie
  - Message
  - Cadres LOOP

# Ce qu'on a couvert

---

- Un processus peut être représenté par plusieurs scénarios
- Chaque scénario peut être décrit pas un diagramme de séquences
- Un diagramme de séquences est particulièrement adapté lorsque les interactions sont importantes





*UML pour la maîtrise d'ouvrage*  
Description des objets « métier »

Découvrir les objets  
« métier »

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Qu'est-ce qu'un objet métier ?
- Les « object flow »
- Mise en œuvre dans l'exemple





# Qu'est-ce qu'un objet « métier »

---

- Objet utilisé par le métier, matériel ou pas
- Produit ou consommé dans un processus métier
- A des caractéristiques stables, en termes de données, ou de comportements



# Les « object flow »

---

- Notation des flux d'objets dans les diagrammes d'activité
- Organisation dans le diagramme



# Mise en œuvre dans l'exemple

---

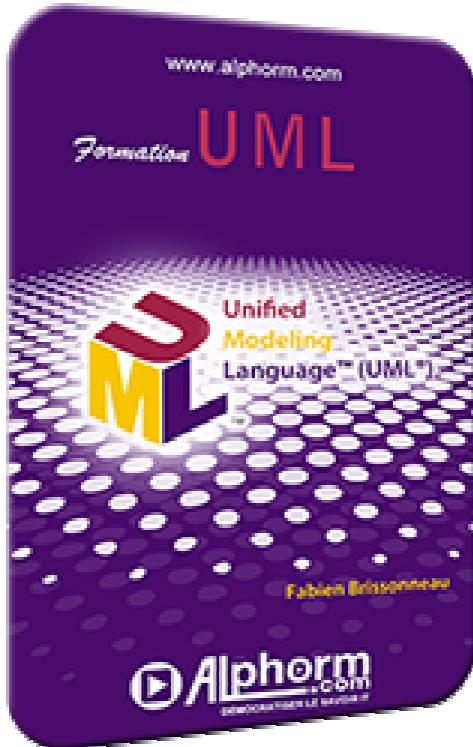
- Les objets produits ou consommés la vente de livres
  - Compte client, Livre, CB, Facture,...

# Ce qu'on a couvert

---

- Qu'est-ce qu'un objet métier ?
- La représentation des « object flow »
- Mise en œuvre dans l'exemple





*UML pour la maîtrise d'ouvrage*  
Description des objets « métier »

Le diagramme de classes  
pour les objets « métier »

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Classes et objets
- Les composantes du diagramme de classes
- Etude de cas





# Classes et objets

---

- Objets
  - Entités ayant une identité
  - Des états
  - Des comportements
- Classes
  - Type des objets
  - Descripteur des objets



# Les composantes du diagramme de classes

---

- Classes
  - Nom
- Attributs
  - Nom, Type
- Associations
  - Nom
  - Multiplicités



# Etude de cas

---

- Classes à représenter
  - Facture
  - CB
  - Compte client
  - Livre
  - Vente

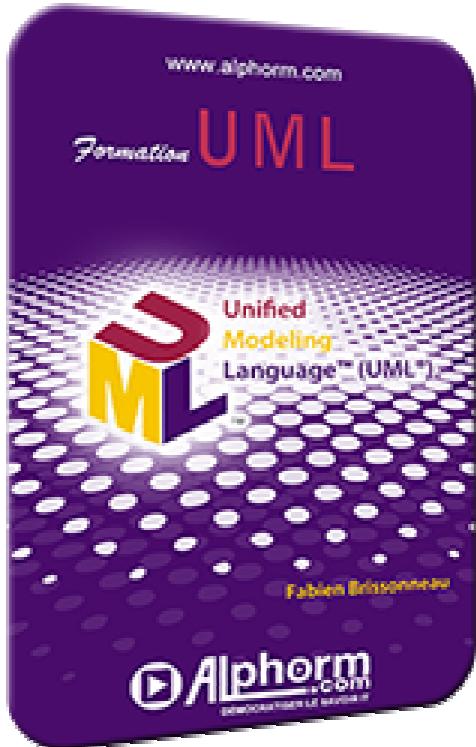


# Ce qu'on a couvert

---

- Rappel des différences entre classe et objet
- Que contient un diagramme de classes
- Exemple de mise en œuvre





*UML pour la maîtrise d'ouvrage*  
L'expression de besoins

# Le diagramme de cas d'utilisation

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Le modèle des besoins
- Organisation du projet
- Présentation de l'étude de cas
- Le diagramme de cas d'utilisation





# Le modèle de besoins

---

- Ne pas se comprendre peut coûter cher
- Améliorer le dialogue métier-développeur
- UML car langage neutre et graphique
- Décrit les besoins, fonctionnels ou non, de l'utilisateur



# Organisation du projet

---

- Nouveau niveau de modélisation : package « model »
- Basé sur la modélisation « métier » si elle existe
- Sera la base du modèle d'analyse



# Présentation de l'étude de cas

---

Thème : construction d'une application de vente à distance (Web)

## Créer un compte

L'utilisateur est n'importe quel individu qui a accès au site web et qui n'a pas déjà un compte sur le système. Un tel utilisateur peut parcourir le catalogue et sélectionner un livre à commander, mais ne peut faire la commande. Les individus non enregistrés veulent s'enregistrer car ils veulent être capables de faire une commande, ou bien lorsqu'ils font leur première commande, on leur demande de s'enregistrer d'abord.

## Gérer un compte

L'utilisateur est une personne qui a un compte et qui veut modifier quelque chose sur son compte.

## S'identifier sur le site

L'utilisateur est une personne qui a un compte sur le système et qui veut être identifié pour avoir accès aux fonctionnalités du système.

## Sortir du site

L'utilisateur est une personne qui a un compte, et qui est actuellement identifiée. L'utilisateur veut retourner dans l'état non-identifié, afin de quitter le site en étant sûr que son compte ne sera pas utilisé par quelqu'un d'autre.

## Parcourir le catalogue

L'utilisateur est une personne qui veut acheter des livres sur le site. L'utilisateur veut parcourir le catalogue des livres disponibles dans le catalogue.



# Présentation de l'étude de cas

---

## Voir le panier

L'utilisateur est une personne qui veut acheter des livres sur le site. L'utilisateur a sélectionné quelques livres dans le panier. L'utilisateur veut revoir son panier pour modifier les quantités ou enlever des lignes.

## Finir la commande (payer)

L'utilisateur est une personne qui veut acheter des livres sur le site. L'utilisateur a sélectionné quelques livres dans le panier. L'utilisateur veut lancer la commande pour le contenu du panier.

## Gérer les utilisateurs

L'utilisateur est une personne qui a des droits d'administrateur. L'utilisateur peut désactiver un compte utilisateur.



# Le diagramme de cas d'utilisation

---

- Cas d'utilisation
- Acteur
- Frontières du système
- Association
- Généralisation
- Extension
- Inclusion

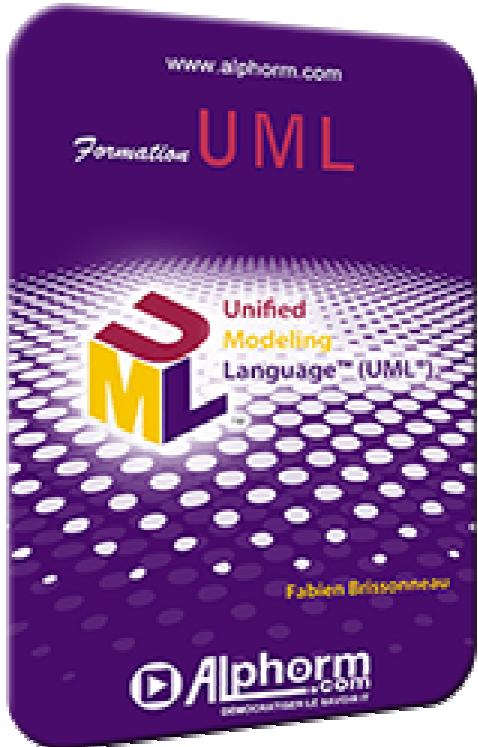


# Ce qu'on a couvert

---

- Définition du modèle des besoins
- Organisation du projet dans un outil
- Le diagramme de cas d'utilisation de l'étude de cas





*UML pour la maîtrise d'ouvrage*  
L'expression de besoins

## Le diagramme d'activité

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Choisir un cas d'utilisation
- Organiser le projet
- Le diagramme d'activité





# Choisir le cas d'utilisation

---

- Chaque cas d'utilisation peut être décrit par :
  - Un diagramme d'activité
  - N diagrammes de séquences
  - Un diagramme d'état
  - Un diagramme de communication



# Organiser le projet

---

- Créer le diagramme d'activité sous le cas d'utilisation
- Prévoir plusieurs diagrammes d'activité si le processus est trop complexe



# Le diagramme d'activité

---

- Action, activité, transition
- Décision, merge
- Fork, Join
- Partition
- Evénement
- Région interruptible

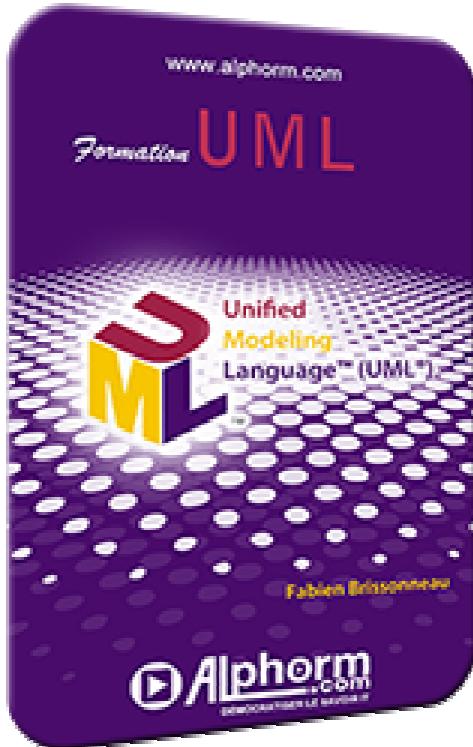


# Ce qu'on a couvert

---

- Où créer le diagramme d'activité
  - Un diagramme d'activité par cas d'utilisation
- La syntaxe de base





*UML pour la maîtrise d'ouvrage*  
L'expression de besoins

# Le diagramme de séquences

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Choisir un cas d'utilisation
- Organiser le projet
- Le diagramme de séquences





# Choisir le cas d'utilisation

---

- Chaque cas d'utilisation peut être décrit par :
  - Un diagramme d'activité
  - N diagrammes de séquences
    - Ne faire que les diagrammes qui apportent à la compréhension
  - Un diagramme d'état
  - Un diagramme de communication



# Organiser le projet

---

- Créer le diagramme de séquences sous le cas d'utilisation
  - Scénario nominal
  - Scénario qui posent un problème
  - Un enchaînement de scénarios



# Le diagramme de séquences

---

- Ligne de vie
- Focus de contrôle
- Message
- Auto-message
- Note, lien de note

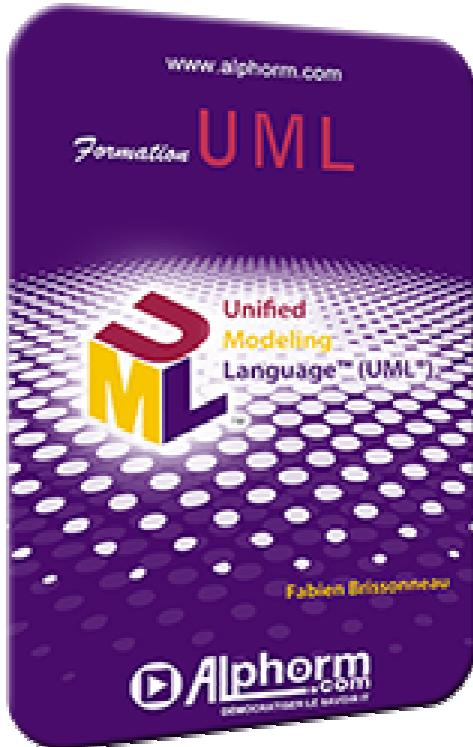


# Ce qu'on a couvert

---

- Pourquoi créer un diagramme de séquences
- La syntaxe de base





*UML pour l'analyse*

Découverte des classes

Le modèle du domaine

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau

Consultant, concepteur et formateur

Objets Logiciels

*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Le diagramme de classe
- Les étapes du processus dans un outil
- La création du modèle du domaine





# Le diagramme de classe

---

- Classe
- Attribut
- Association



# Les étapes du processus

---

- *Métier*
- Besoins
- Analyse
- Conception
- *Codage*



# La création du modèle du domaine

---

- 1.Le caissier ouvre une vente
- 2.Le terminal prépare la saisie en affichant la date
- 3.Le caissier saisit le code de chaque article, éventuellement une quantité
- 4.A chaque fois, le terminal affiche la description de l'article, ainsi que son prix unitaire et le sous-total
- 5.A la fin, le caissier indique la fin de la saisie
- 6.Le terminal affiche le total de la vente
- 7.Le caissier choisit alors le type de paiement en espèces
- 8.Le terminal demande le montant donné par le client
- 9.Le caissier donne le montant
- 10.Le terminal affiche le montant à rendre
- 11.Le caissier indique qu'il a rendu la monnaie
- 12.Le terminal enregistre la vente et fournit un ticket

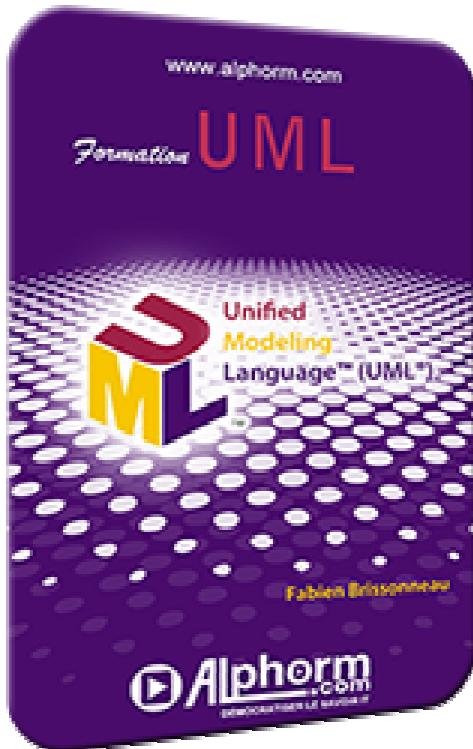


# Ce qu'on a couvert

---

- Définition du modèle du domaine
- La syntaxe essentielle du diagramme de classe





*UML pour l'analyse*  
Découverte des classes

Partir du cas d'utilisation

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Lier cas d'utilisation et classes
- Exemple avec l'étude de cas





# Lier cas d'utilisation et classes

---

- Justifier les classes par rapport aux besoins
- Eviter les idées géniales de l'analyste ou du développeur
- Maîtriser l'évolution des modèles
- Parler de « réalisation des cas d'utilisation »



# Etude de cas

---

- Pour chaque cas d'utilisation
  - Créer les classes participantes
- Regrouper les classes
  - Eviter doublons et synonymes

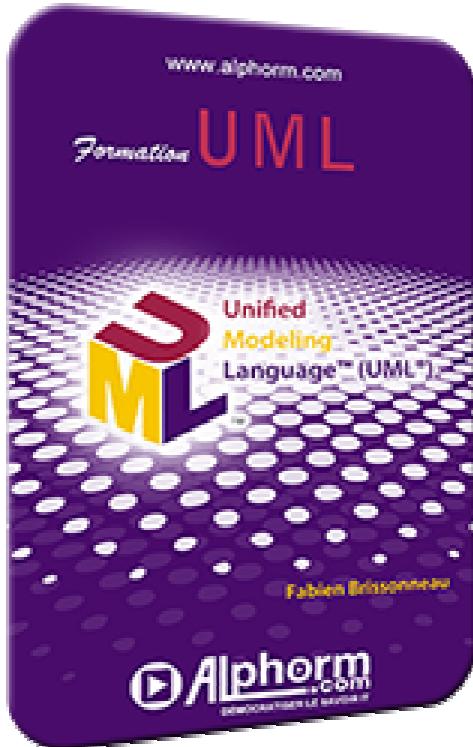


# Ce qu'on a couvert

---

- La liaison entre fonctions et orienté objet
- Constituer les classes d'analyse du projet





*UML pour l'analyse*  
Découverte des classes

# Diagramme de séquence d'analyse

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Créer les classes par cas d'utilisation
- La syntaxe du diagramme de séquence
- Etude de cas : formaliser l'utilisation des classes





# Créer les classes par cas d'utilisation

---

- Les classes sont explicables par cas d'utilisation
- Le modèle de structure est indissociable du modèle dynamique



# La syntaxe du diagramme de séquence

---

- Ligne de vie
- Messages typés : création, destruction,...
- Paramètres et retours sur les messages
- Les cadres



# Etude de cas

---

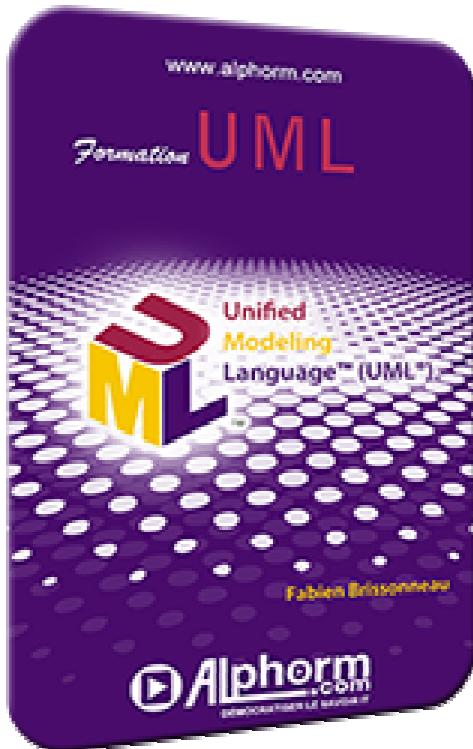
- Partant d'un diagramme de séquences système
- Remplacer le système par des objets issus des classes du modèle
- Créer le diagramme de séquence d'analyse

# Ce qu'on a couvert

---

- Créer les classes par cas d'utilisation ne suffit pas
- La syntaxe du diagramme de séquence peut être relativement complexe
- Montrer que les classes du diagramme de classes sont réellement utilisées





*UML pour l'analyse*  
Les état des objets

La notation de base du  
diagramme d'états

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Quand utiliser un diagramme d'état ?
- Notation de base
- Etude de cas





# Quand utiliser un diagramme d'état

---

- Objet ayant un nombre d'états significatifs
- Objet ayant une réponse variable selon son état
- Découvrir le cycle de vie d'un objet



# Notation de base

---

- Initial, final
- Etat, transition
- Evénement, condition, effet



# Etude de cas

---

- Choisir un objet dont les états sont importants
- Créer le diagramme d'état dans le projet

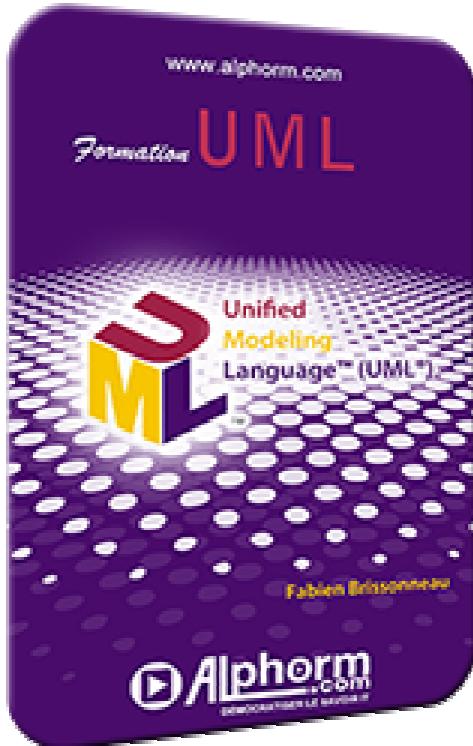


# Ce qu'on a couvert

---

- Nous avons montré sur quel genre d'objet créer un diagramme d'état
- Nous avons présenté la notation de base
- Et application à l'étude de cas





*UML pour l'analyse*  
Les état des objets

La notation avancée du  
diagramme d'états

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- La notation avancée du diagramme
- Etude de cas





# La notation avancée

---

- Etats composites
- Transition interne
- Activités, effets entrée et sortie
- Evénements prédéfinis



# Précautions d'usage ...

---

- La notation complète des diagrammes d'états peut être très complexe
- Eviter de trop vouloir en mettre
- Garder le diagramme lisible



# Etude de cas

---

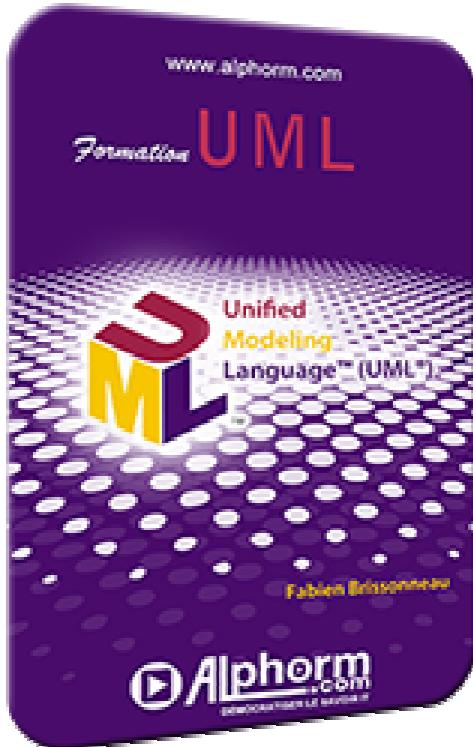
- Améliorer le diagramme précédent
- Evénements prédéfinis

# Ce qu'on a couvert

---

- Nous avons proposé une notation avancée des diagrammes d'état
  - Etat composite
  - Transition interne
  - Activité
  - Effet en entrée, en sortie
  - Evénements prédéfinis





*UML pour la conception*

## Construire le modèle de conception

# La notation pour la conception

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

*Formation UML*

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Pourquoi et quand mettre des détails dans la notation ?
- Notations supplémentaires
- Etude de cas





# Pourquoi et quand ?

---

- Le modèle de conception est une abstraction pour le code
- Le lien avec le langage de programmation
- Génération de code



# Notation supplémentaire

---

- Nom de rôle, navigabilité, restriction
- Dépendance contextuelle
- Visibilité des membres, paramètres et types, retour et type
- Package, organisation



# Etude de cas

---

- Reprendre un modèle pour le compléter
  - Visibilités
  - Types utilisés
  - Méthodes
  - ...

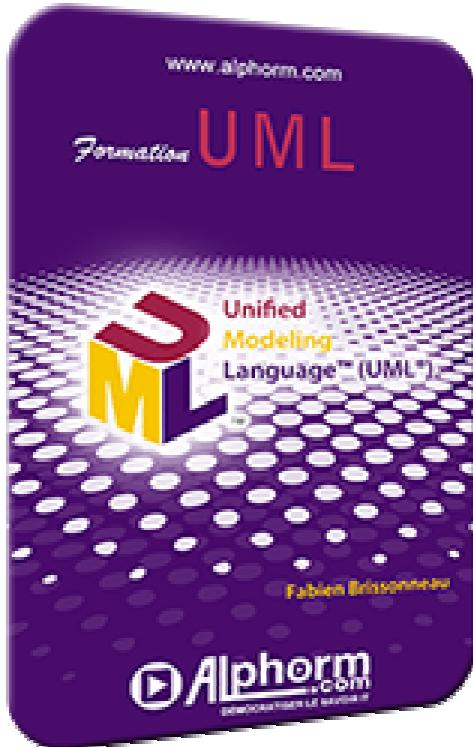


# Ce qu'on a couvert

---

- Pourquoi et quand mettre des détails dans la notation ?
- Notations supplémentaires
- Et application à l'étude de cas





*UML pour la conception*

## Construire le modèle de conception

# Passer de l'analyse à la conception

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

*Formation UML*

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- D'où partir ?
- La traçabilité entre les modèles
- Etude de cas





# D'où partir ?

---

- Modèle d'analyse
- Architecture technique
- Influence du langage de programmation



# La traçabilité entre les modèles

---

- Classe d'analyse ou bien classe de conception
- Création des diagrammes en fonction du lecteur
- Les transformations possibles



# Etude de cas

---

- Exemple site web
- Base de données
- Java

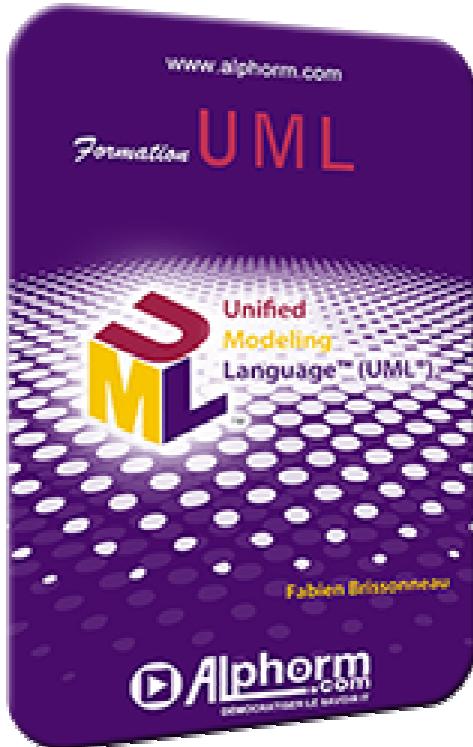


# Ce qu'on a couvert

---

- D'où partir ?
- La traçabilité entre les modèles
- Etude de cas





*UML pour la conception*  
Concevoir et coder

Traduire le diagramme  
de classe en code

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- Traduire classe, attribut et méthode
- Traduire les associations
- Etude de cas





# Traduire classe, attribut et association

---

- Classe => class
- Attribut => déclaration de données membres
- Méthodes => déclaration de fonctions membres



# Traduire les associations

---

- Associations multiplicité 1 max
- Associations multiplicité supérieure à 1



# Etude de cas

---

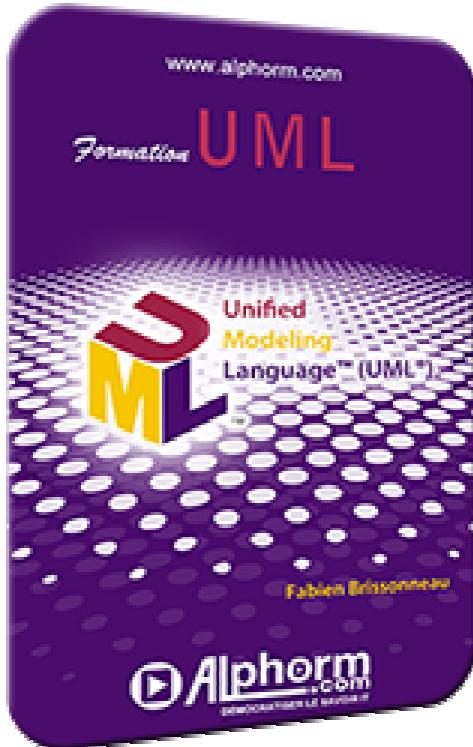
- La commande
  - Numéro de commande
  - Créer la commande
  - Générer la facture de la commande

# Ce qu'on a couvert

---

- Traduire une classe, les attributs et les méthodes
- Traduire les associations
- Et application à l'étude de cas





*UML pour la conception*  
Concevoir et coder

Traduire le diagramme  
de séquences en code

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

alphorm.com™ ©



# Plan

---

- Messages / méthodes
- Traduire les envois de messages
- Etude de cas





# Messages / méthodes

---

- Messages et méthodes sont très proches
- Le message est l'information passant d'un émetteur vers un récepteur
- La méthode est le code déclenché par ce message



# Traduire les envois de message

---

- Le modèle dynamique correspond aux corps des méthodes
- Le diagramme de séquence représente en une seule fois l'ensemble des appels sur plusieurs niveaux



# Etude de cas

---

- La séquence « passer commande »
- La méthode de la classe commande
- Le corps de la méthode

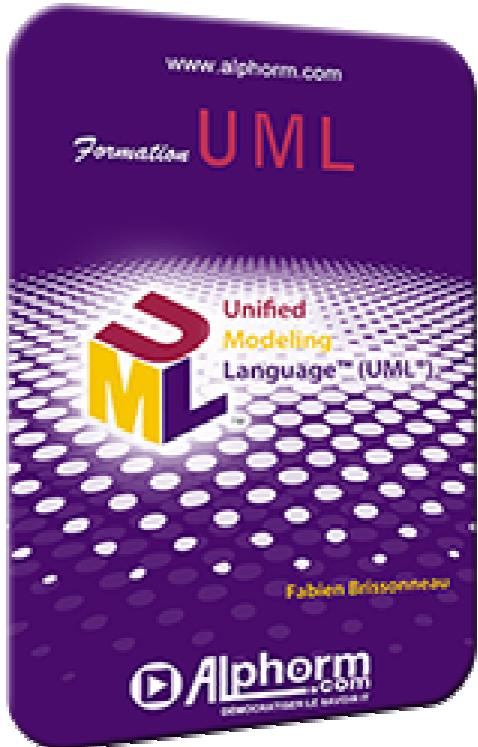


# Ce qu'on a couvert

---

- Messages / méthodes
- Traduire les envois de messages
- Et application à l'étude de cas





*UML pour la conception*  
Concevoir et coder

# Reporter la complexité sur le diagramme de classes

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

Site : <http://www.alphorm.com>  
Blog : <http://www.alphorm.com/blog>  
Forum : <http://www.alphorm.com/forum>

Formation UML

alphorm.com™ ©



# Plan

---

- Pourquoi éviter la complexité dynamique
- Un mécanisme de base de l'orienté objet : le polymorphisme
- Etude de cas





# Pourquoi éviter la complexité dynamique ?

---

- La complexité
  - Logique conditionnelle
  - Corps de méthode trop long
  - ...
- Fragilité, rigidité, immobilité



# Le polymorphisme

---

- Mécanisme qui permet d'envoyer un message à un objet sans connaître le type réel de cet objet, mais la méthode qui va être déclenchée dépend elle du type réel
- Remplace la logique conditionnelle avec test du type de l'objet



# Exemple

---

- Cas du système de fichier
- Fichier, Répertoire, Lien
- Imaginer un traitement sur le répertoire ...

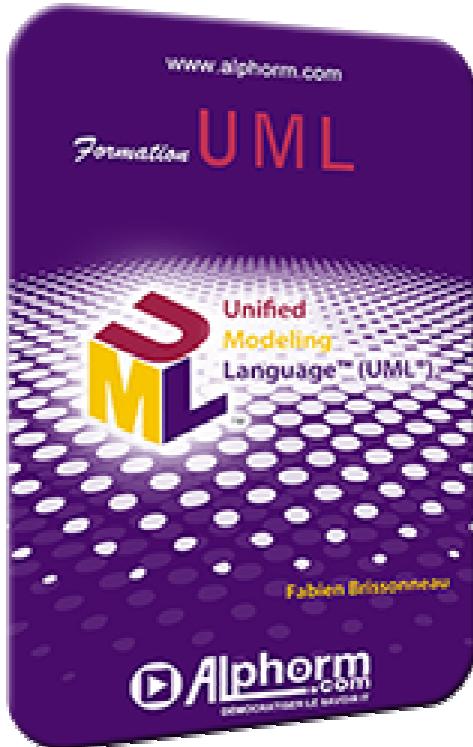


# Ce qu'on a couvert

---

- Pourquoi éviter la complexité dynamique
- Un mécanisme de base de l'orienté objet : le polymorphisme
- Etude de cas





# Conclusion

Site : <http://www.alphorm.com>

Blog : <http://www.alphorm.com/blog>

Forum : <http://www.alphorm.com/forum>

Formation UML

Fabien Brissonneau  
Consultant, concepteur et formateur  
Objets Logiciels  
*Contact* : fabien.brissonneau@gmail.com

alphorm.com™ ©



# Plan

---

- UML pour quoi faire
- Ce que l'on a couvert
- Pour aller plus loin ...



# Qu'est ce que UML?

---

- Langage de modélisation « Unified Modeling Language »
- UML est une notation graphique utilisable pour :
  - Décrire les besoins des utilisateurs sur une application
  - Spécifier un système
  - Concevoir un système informatique
  - Réaliser une documentation
  - Améliorer la communication entre les parties prenantes d'un projet
- UML peut être utilisé de manière plus ou moins formelle, tout à fait compatible avec les démarches agiles. UML est un outil.





# Ce que l'on a couvert

---

- **Module 1 : Généralités**

- Chapitre 1 : Les diagrammes UML
  - Chapitre 2 : Mettre en œuvre UML dans le projet

- **Module 2 : UML pour la maîtrise d'ouvrage**

- Chapitre 1 : Les processus métier
  - Chapitre 2 : Les objets métier
  - Chapitre 3 : L'expression de besoins

- **Module 3 : UML en analyse**

- Chapitre 1 : Découvrir les classes
  - Chapitre 2 : Les états des objets

- **Module 4 : UML en conception**

- Chapitre 1 : Construire le modèle de conception
  - Chapitre 2 : Passer de l'analyse à la conception





# Pour aller plus loin

---

- Le choix d'un outil ...
- Le temps de l'apprentissage ...
- La conception objet ...

 Il n'y a plus qu'à ...

---

Merci !