

UED Pilotage et maîtrise de la chaîne Logistique

UML

Unified Modeling Language

Laure FREBOURG – laure.frebourg@gmail.com

Objectifs du cours

- Ce qu'est UML
- L'utilisation d'UML dans un projet informatique
- Concept informatique d'objet
- Etre capable de lire les diagrammes UML
- En comprendre les principes d'élaboration
- Contrôle de connaissances : QCM au test

Plan du cours

1. Introduction
2. Modélisation des objets
3. Modélisation du comportement
4. Modélisation de l'installation
5. Etudes de cas

1. Introduction

Historique

- Développement des langages de programmation objet (notamment C++ dans les années 80, JAVA années 90)
 - Besoin d'une modélisation objet
- Nombreuses méthodes ont émergé
- Fusion en 1997 avec UML, langage standardisé par OMG (Object Management Group)

Webographie

The screenshot shows the homepage of the Unified Modeling Language (UML) website, located at www.uml.org. The page features the OMG logo and the text "WE SET THE STANDARD™". The "UNIFIED MODELING LANGUAGE™" logo is prominently displayed. A navigation menu at the top includes links for About Us, Press Room, Calendar, Documents, Members Only, Technology, and Industries. Below the menu, a large section titled "UML® Resource Page" contains links to Introduction to UML, UML Success Stories, UML Certification Program, and Vendor Directory. There are also sections for a new whitepaper and a UML Use Case Narratives Survey. On the left, there's a "Getting Started With UML:" section and a paragraph about UML's role in modeling. On the right, there's a "Free UML Tool" section featuring a Use Case Diagram.

www.uml.org

OMG[®]
WE SET THE STANDARD™

UNIFIED MODELING LANGUAGE™

About Us | Press Room | Calendar | Documents | Members Only | Technology | Industries

UML® Resource Page

[[Introduction to UML](#)] [[UML Success Stories](#)] [[UML Certification Program](#)] [[Vendor Directory](#)]

NEW OMG WHITEPAPER:
Visual Modeling: Past, Present and Future
by Andrew Watson, Object Management Group

UML Use Case Narratives Survey

Getting Started With UML:

The Unified Modeling Language™ - UML - is OMG's most-used specification, and the way the world models not only application structure, behavior, and architecture, but also business process and data structure.

UML, along with the [Meta Object Facility \(MOF™\)](#), also provides a key foundation for OMG's [Model-Driven Architecture®](#), which unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution.

OMG is a [not-for-profit computer industry specifications consortium](#); our members define and maintain the UML specification which we publish in the series of documents linked on this page for your free download. Software providers of every kind build tools that conform to these specifications. To model in UML, you'll

Free UML Tool

Use Case Diagram

Webographie

Créer un compte ou se connecter

article discussion modifier historique

Vos dons permettent à Wikipédia de continuer à exister ! Merci de votre soutien.

Unified Modeling Language

Pour les articles *homonymes*, voir [UML](#).

UML (en anglais *Unified Modeling Language*, « langage de modélisation unifié ») est un langage graphique de modélisation des données et des traitements. C'est une formalisation très aboutie et non-propriétaire de la modélisation objet utilisée en génie logiciel. L'OMG travaille actuellement sur la version UML 2.1.

Il est l'accomplissement de la fusion des précédents langages de modélisation objet **Bloch**, **OMT**, **OOSE**. Principalement issu des travaux de **Grady Booch**, **James Rumbaugh** et **Ivar Jacobson**. UML est un standard défini par l'**OMG** (Object Management Group).

Sommaire [masquer]

1 Le formalisme d'UML

- 1.1 Les vues
- 1.2 Les diagrammes
- 1.3 Les modèles d'éléments
 - 1.3.1 Les modèles d'éléments de type commun
 - 1.3.2 Les modèles d'éléments de type relation

2 Standardisation UML



Logo d'UML

Navigation

- Accueil
- Portails thématiques
- Index alphabétique
- Une page au hasard

Contribuer

- Aide
- Communauté
- Modifications récentes
- Accueil des nouveaux arrivants
- Faire un don

Recherche

Boîte à outils

Consulter Chercher

Webographie

The screenshot shows a web browser window displaying the IBM Rational UML Resource Center. The URL in the address bar is <http://www-01.ibm.com/software/rational/uml/>. The page title is "Unified Modeling Language". The left sidebar contains a navigation menu with links to Home, Solutions, Services, Products, Support & downloads, and My IBM. The main content area features a large image of binary code forming a 3D perspective grid. Below the image, a text block explains the historical significance and benefits of UML. A section titled "IBM Rational can help you apply the UML" lists various resources and products. To the right, there's a sidebar titled "We're here to help" with links for chat, quotes, email, and phone support, along with priority codes. Another sidebar titled "Highlights" lists white papers and a link to Adobe Reader. The bottom right sidebar is titled "The Rational Edge" and lists several white papers.

Software > Rational > UML Resource Center >

Unified Modeling Language

Overview | Resources | Products | MDA | Documentation

Rational software

We're here to help

Easy ways to get the answers you need.

Chat now | Request a quote | E-mail IBM

Or call us at:
877-426-3774
Priority code:
104CBW61

Highlights

White paper: What's new in UML 2.0? (346KB)

White paper: The value of modeling (156KB)

Get Adobe® Reader®

The Rational Edge

Adopting use cases Part I: Understanding types of use cases and artifacts (360KB)

The role of the service-oriented architect (230KB)

developerWorks UML articles and tutorials

Related Products

- Rational Developer for v7.5
- Rational Developer for SOA Construction v7.5
- Rational Business Developer v7.5 and v7.5.1
- Rational Team Concert v1.0
- Rational Host Access Transformation Services v7.5
- Rational Host Integration Solution v7.5
- Databorough X-Analysis
- Warranties and licenses

Modeling has been an essential part of engineering, art and construction for centuries. Complex software designs that would be difficult for you to describe textually can readily be conveyed through diagrams. Modeling provides three key benefits: visualization, complexity management and clear communication.

UML™, the Unified Modeling Language™, is a visual language for specifying, constructing, and documenting the artifacts of systems. You can use UML with all processes, throughout the development lifecycle, and across different implementation technologies. UML was approved by the OMG™ as a standard in 1997. Over the past few years there have been minor modifications made to the language. UML 2 is the first major revision to the language.

IBM Rational can help you apply the UML

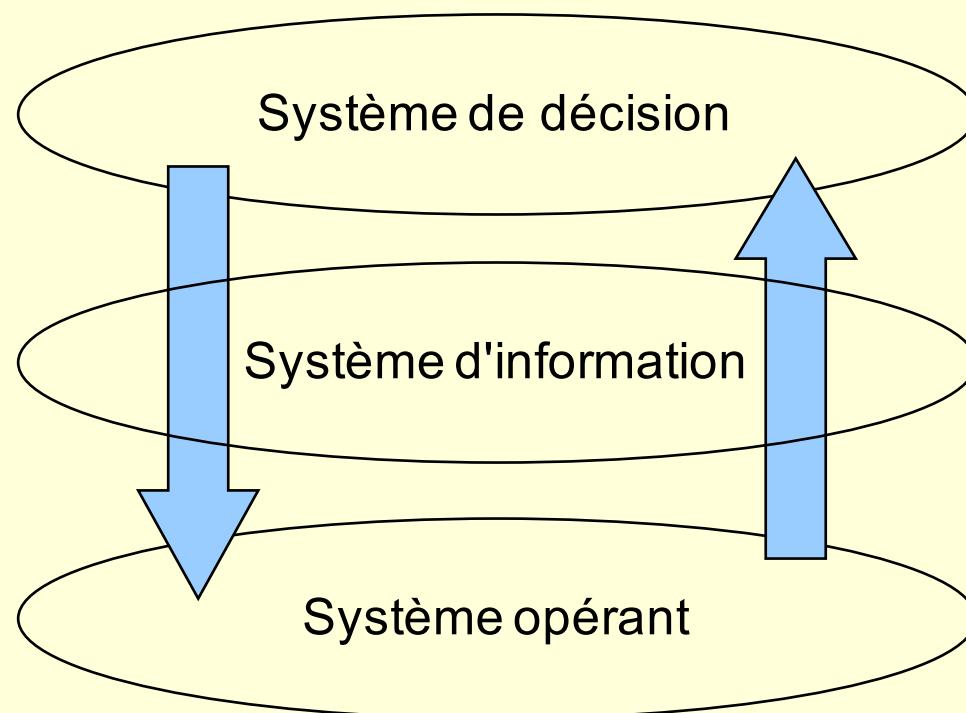
- [Resources](#) – Articles, white papers and tutorials to help you get started with UML or implement UML techniques.
- [Products](#) -- IBM Rational products that automate the practice of applying the Unified Modeling Language (UML)
- [Model-Driven Architecture \(MDA\)](#) – Information and resources on Model Driven Architecture, a philosophy of how models should be used in the software development process.
- [Documentation](#) – A directory of links into the UML standards documentation and resources on the Object Management Group (OMG) site.

Information

- Information : interprétée par un agent
- Types d'information dans l'organisation :
 - Données réelles
 - Données générées à partir de traitements
 - Agrégation de données opérationnelles
 - Objectifs
 - Décisions
- Qualités de l'information : exactitude, pertinence, complétude, disponibilité

Systèmes d'information

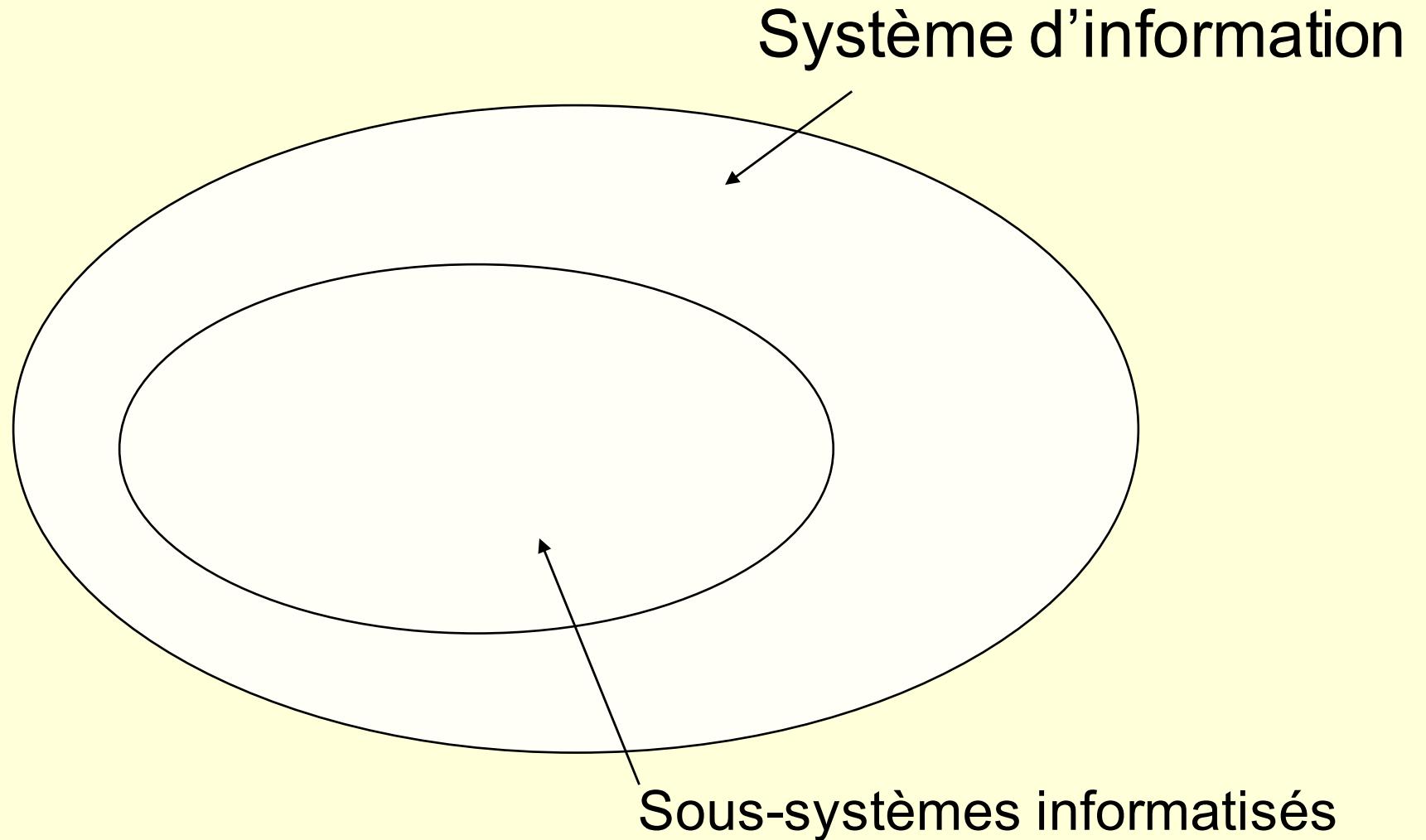
- Théorie systémique : 3 sous-systèmes



Systèmes d'information

- Définition : Ensemble des moyens matériels, logiciels et humains qui permettent la gestion de l'information au sein de l'organisation :
 - Acquisition de l'information
 - Stockage de l'information
 - Traitement de l'information
 - Diffusion de l'information

Systèmes d'information



UML : orienté processus

- Processus : ensemble d'activités ayant pour objet l'obtention d'un résultat final défini
- Correspondent aux activités de base de l'entreprise
- Réalisées par des ressources (humaines, informatiques, techniques, etc.) organisées

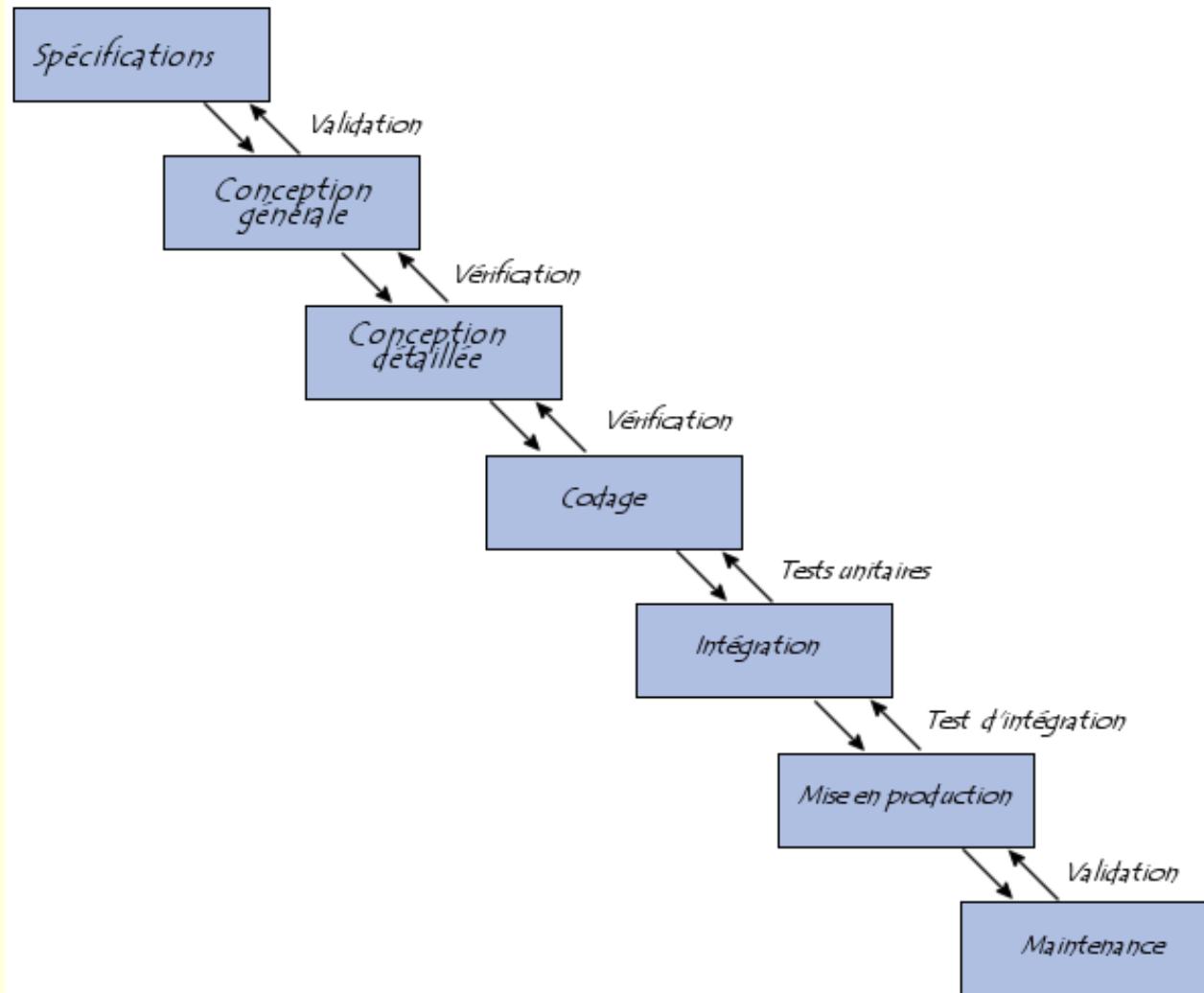
Processus : ensemble d'activités

- Une activité est caractérisée par :
 - Sa place chronologique dans le processus
 - Données d'entrée
 - Données de sortie
 - Ressources mobilisées
 - Moyens de pilotage

Cycles de vie d'un projet informatique

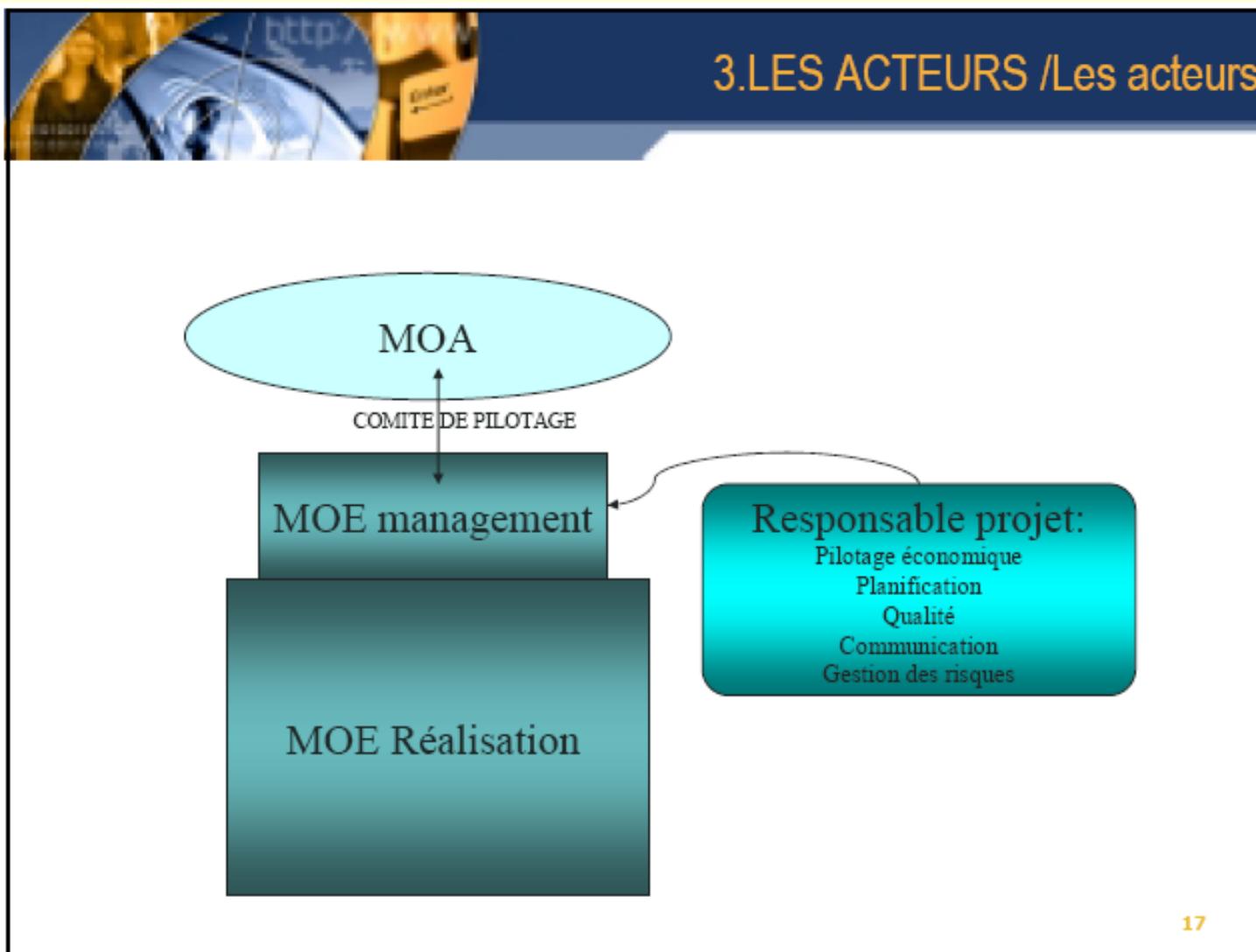
- Définition des objectifs
- Analyse des besoins et faisabilité
- Conception architecturale
- Conception détaillée
- Implémentation
- Tests
- Intégration
- Documentation, formation
- Mise en production
- Maintenance

Cycle en cascade



Tiré de commentcamarche.net/genie-logiciel/cycle-de-vie.php3

Utilisateurs et informaticiens



17

De la nécessité de se comprendre...

La vie d'un projet informatique 'bien' mené



Comment le client a exprimé son besoin



Comment le chef de projet l'a compris



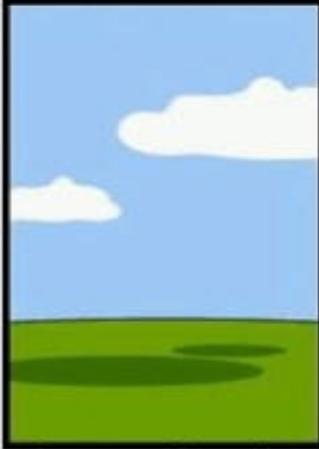
Comment l'ingénieur l'a conçu



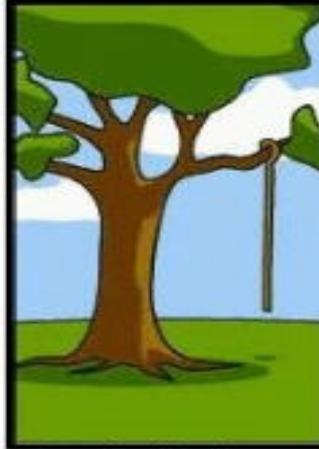
Comment le programmeur l'a écrit



Comment le responsable des ventes l'a décrit



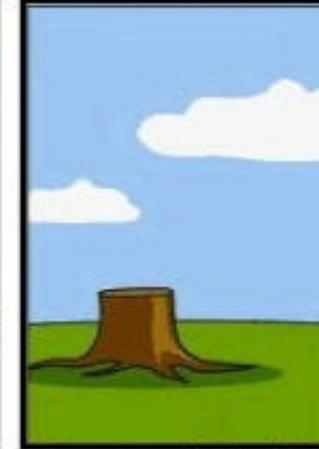
Comment le projet a été documenté



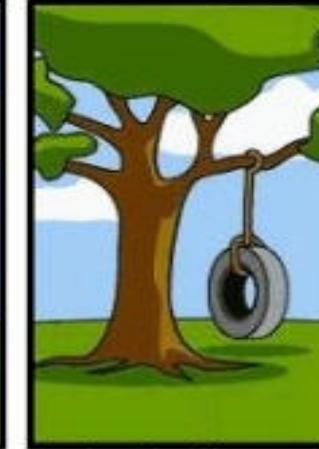
Ce qui a finalement été installé



Comment le client a été facturé

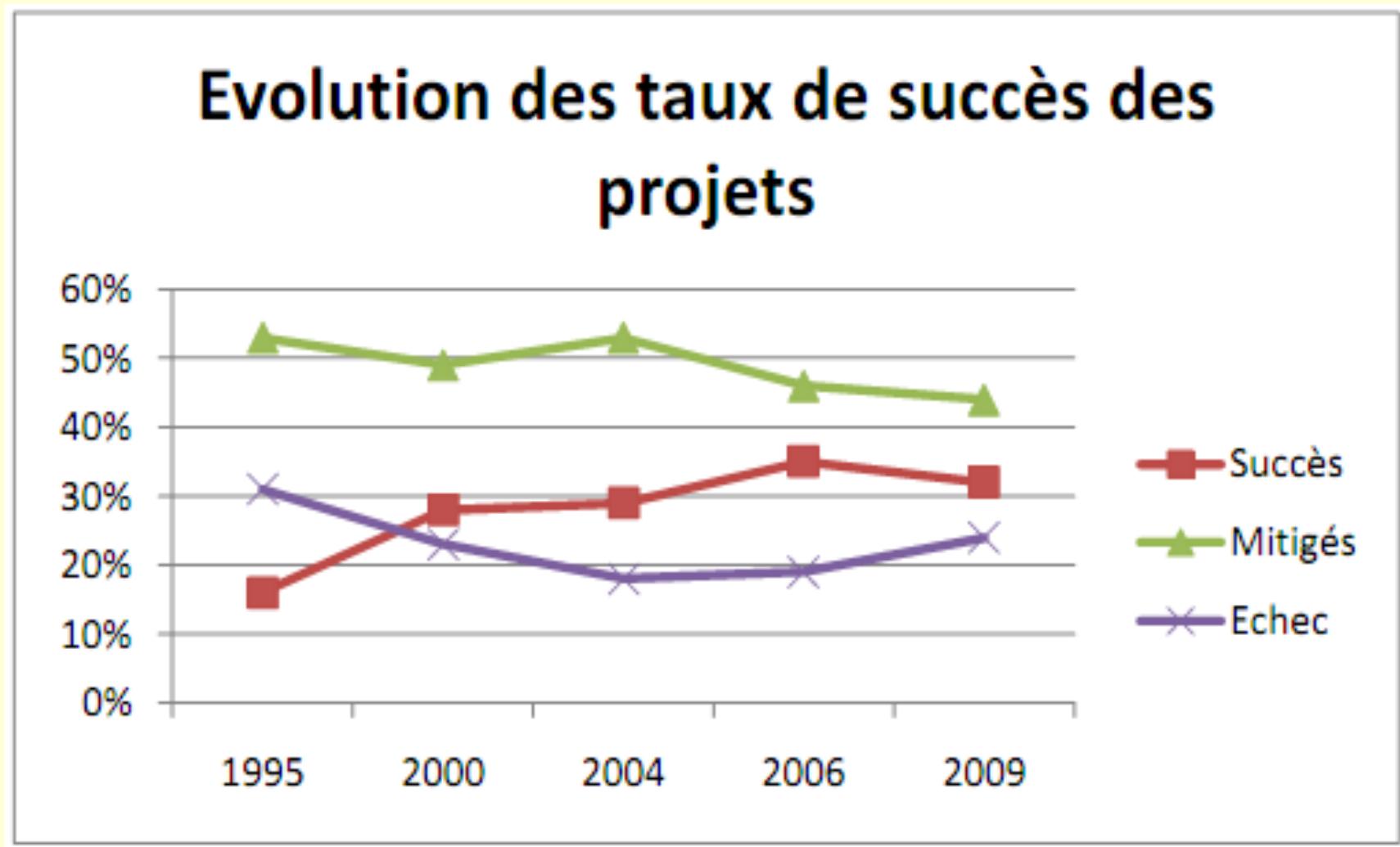


Comment la hotline répond aux demandes

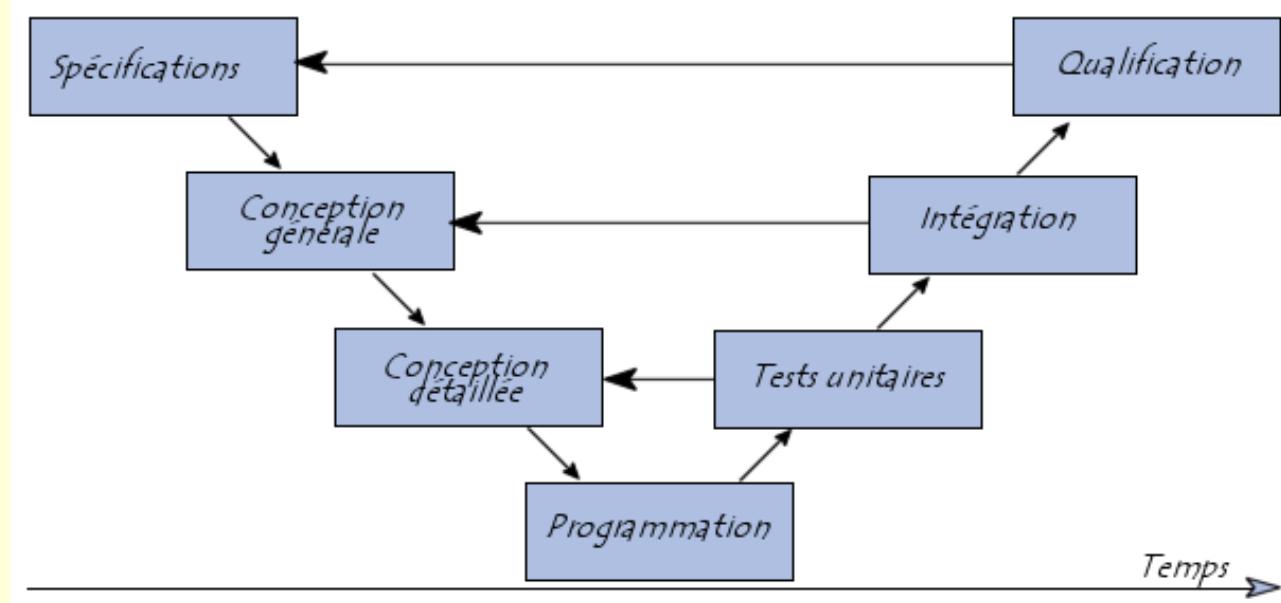


Ce dont le client avait réellement besoin

Réussite ou échec des projets



Cycle en V



Méthodes agiles ou itératives

- Conception initiale minimale
- Intégration progressive de fonctionnalités avec tests et validation par le client à chaque itération
 - pas d'effet « tunnel »
- Incapacité à l'exhaustivité du client en début de projet
 - collaboration avec le client plutôt que négociation contractuelle
 - ouverture au changement plutôt que plan rigide

Méthodes agiles ou itératives

- RAD : Rapid Application Development
- XP : Extreme Programming
- RUP : Rational Unified Process (propriété IBM),
basé sur UML
 - collaboration permanente avec le client → nécessité d'un langage commun
 - U.M.L.

Axes méthodologiques

- Modélisation des processus métiers
- Modélisation du système d'information
 - *Construire des représentations*
- Dialogue entre les acteurs du projet
- Réutilisation et production de composants logiciels
- Maîtrise des évolutions

Urbanisation

LES TROIS VISIONS SUR
LE SYSTEME D'INFORMATION

Vision Métier

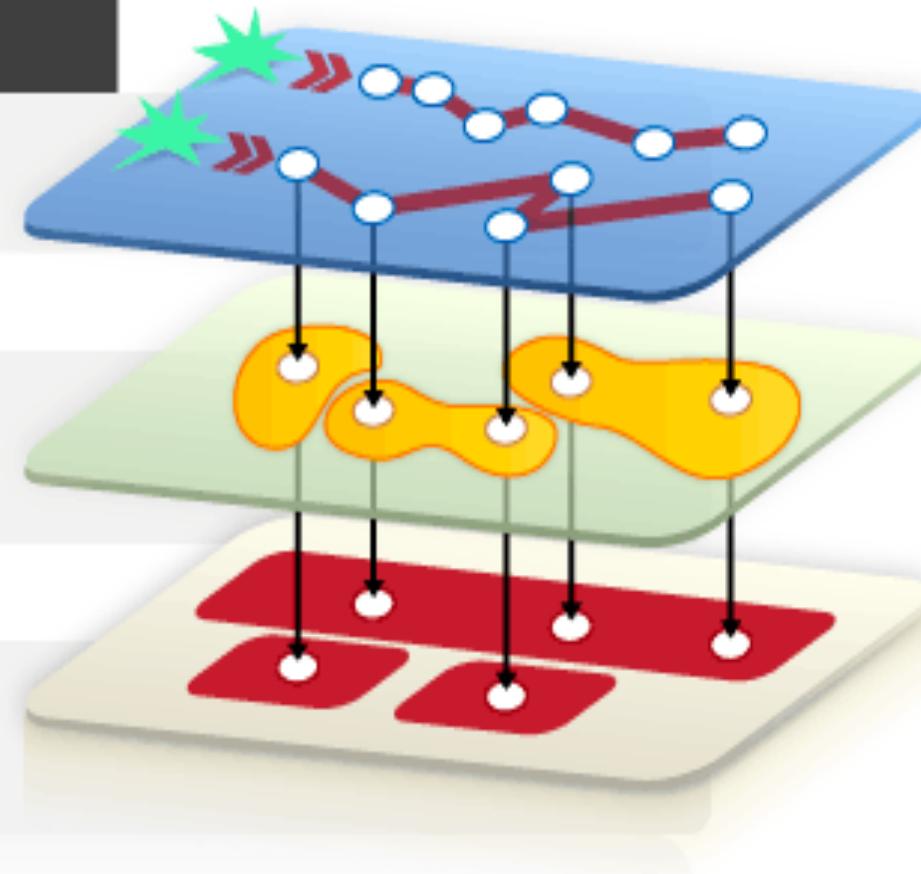
Événements - Processus

Vision fonctionnelle du SI

Plan d'Occupation des Sols
(P.O.S) fonctionnel du SI

Vision Informatique

Applications, progiciels, BD
Architecture technique



Les grands principes de la modélisation

3 principes sont retenus pour la modélisation d'un système d'information :

- **Le découpage en domaines**
- **L'approche par niveaux**
- **La diversité des points de vue**

Le découpage en domaines

Si le système à modéliser est vaste, un découpage permet de mieux comprendre, de maîtriser son développement et d'assurer des évolutions partielles.

Principe 1 : Le système d'information d'une entreprise est découpé en sous-systèmes d'information que l'on appelle **domaines**. Un **domaine** représente un métier, c'est-à-dire une mission, un savoir-faire, des compétences.

L'approche par niveaux

Le modèle ANSI/SPARC a proposé 3 niveaux de description d'une base de données, cette idée a été étendue à la modélisation des systèmes d'information.

Principe 2 : La représentation d'un système d'information se fait d'abord à un niveau conceptuel-organisationnel, puis à un niveau logique et enfin à un niveau physique.

La diversité des points de vue

Un système d'information est souvent complexe. Une des caractéristiques de cette complexité est qu'on ne peut en rendre compte par une représentation unique.

Principe 3 : La modélisation d'un système d'information s'appuie sur plusieurs types de diagrammes, qui rendent chacun compte d'un point de vue particulier.

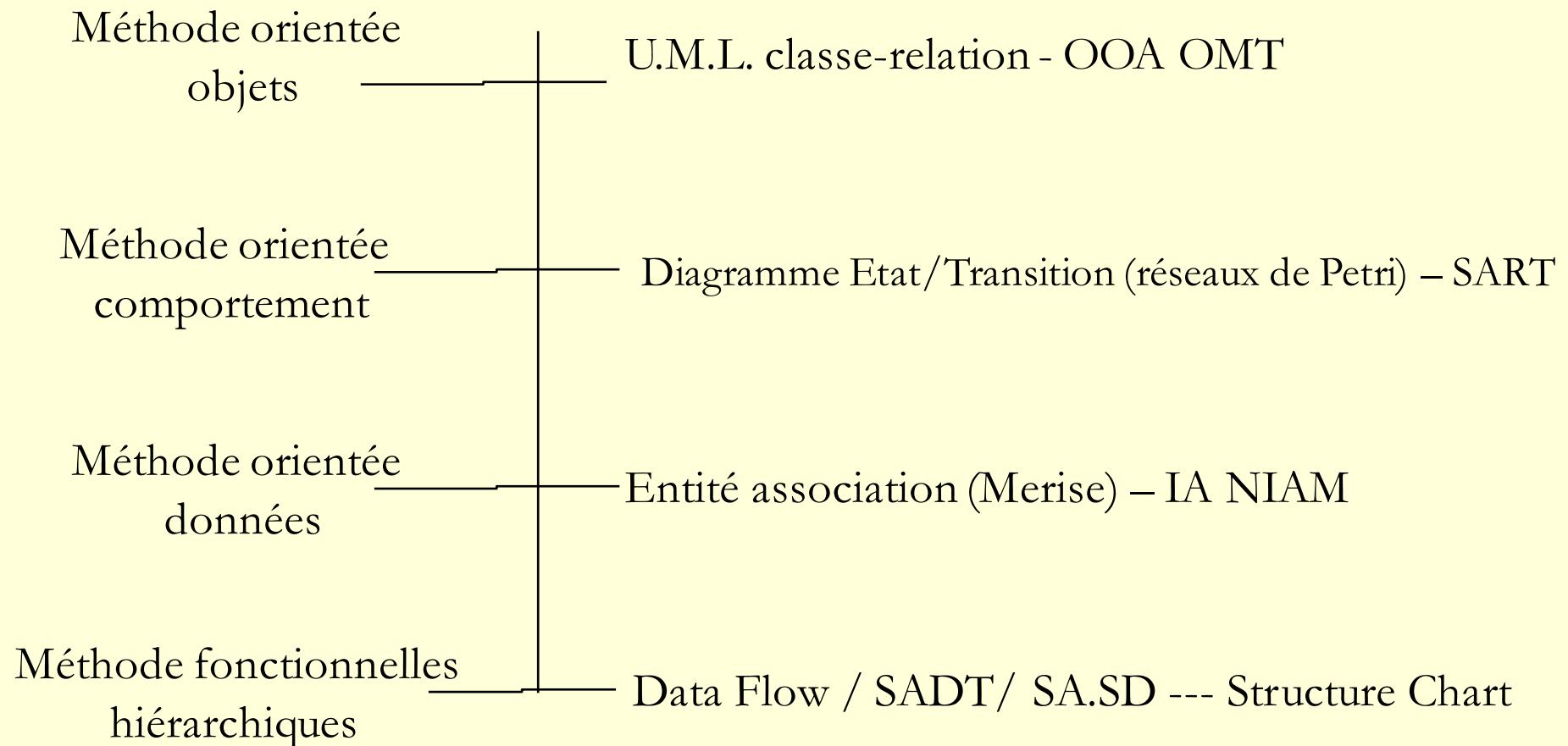
Qualité des modèles

- **Cohérence**
- **Complétude**
- **Capacité de vérification**
- **Précision**
- **Capacité de génération de code**

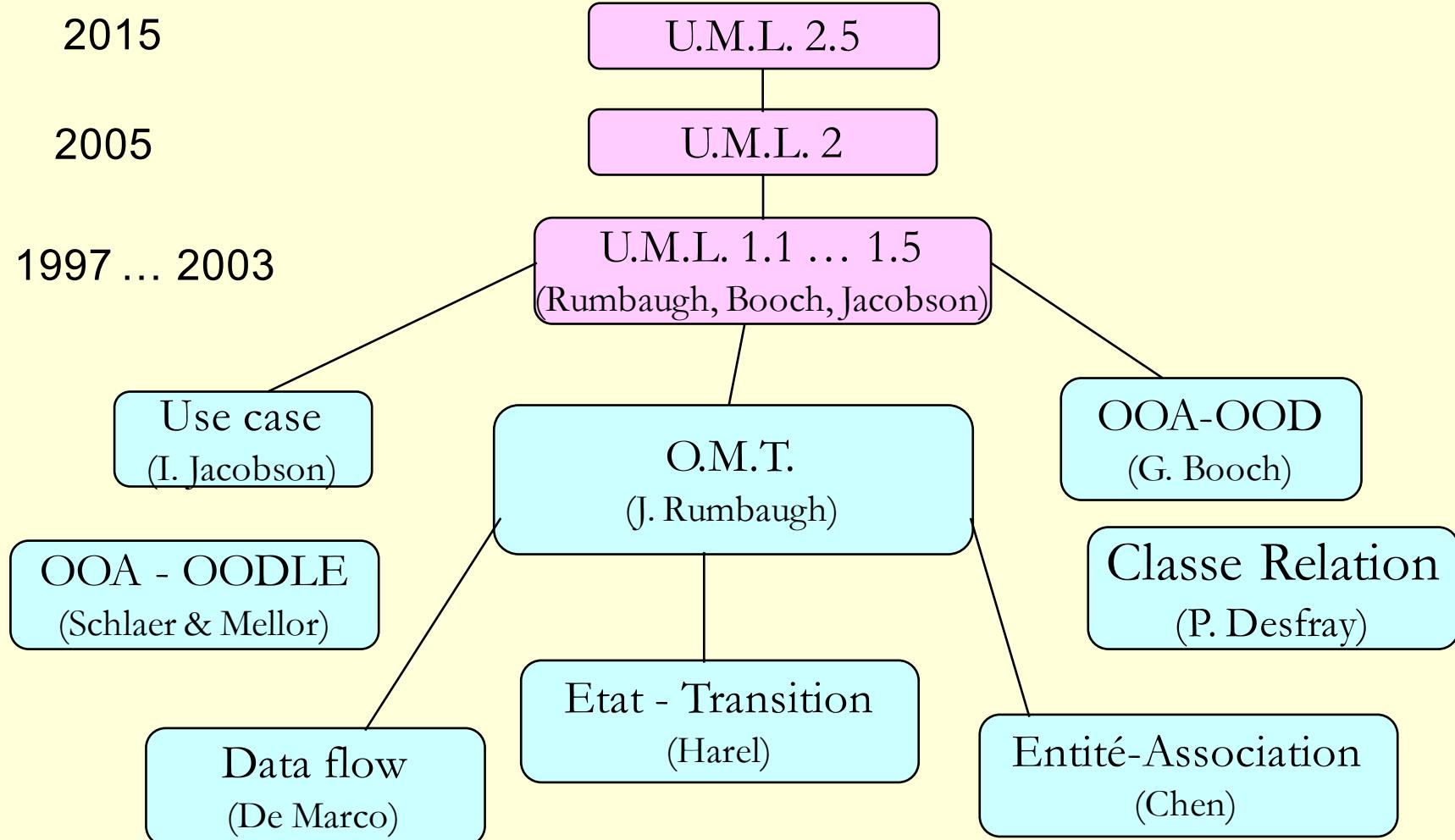
Conception SI Assistée par Ordinateur (Atelier Génie Logiciel)

Classification des méthodes

Les méthodes de spécification et de conception



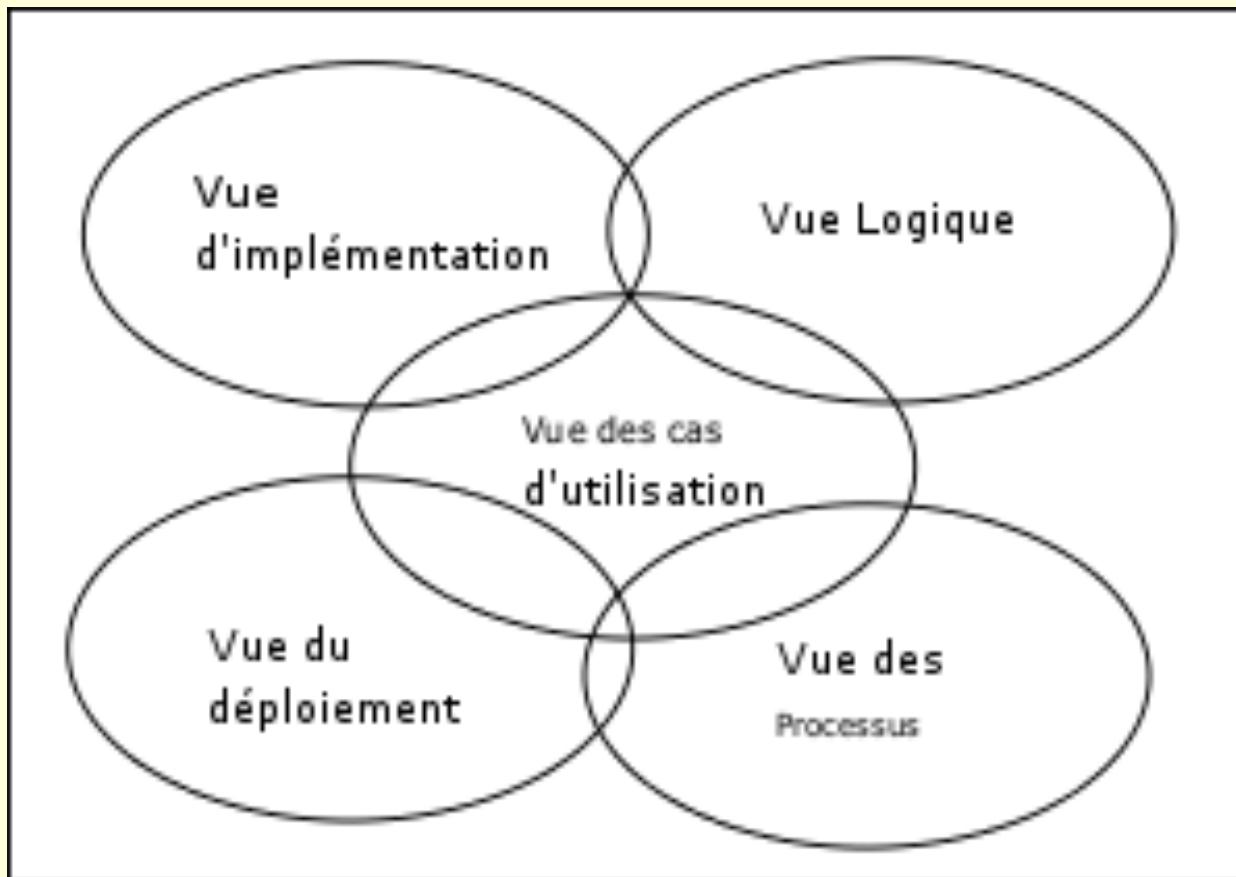
Généalogie d'UML



Formalisme d'UML

- Vues : description du système orientée vers une certaine fonction
- Diagrammes : graphes représentant les interactions entre les éléments
- Modèles d'éléments : composants de base de la description du système
- Langage OCL : Object Constraint Language

Les vues



Les vues

- Logique : modélisation des principaux éléments d'architecture et mécanismes logiciels. Vue des équipes de conception et développement.
- Implémentation : modules (composants logiciels) qui implémentent les éléments définis dans la vue logique. Vue des équipes d'intégration.

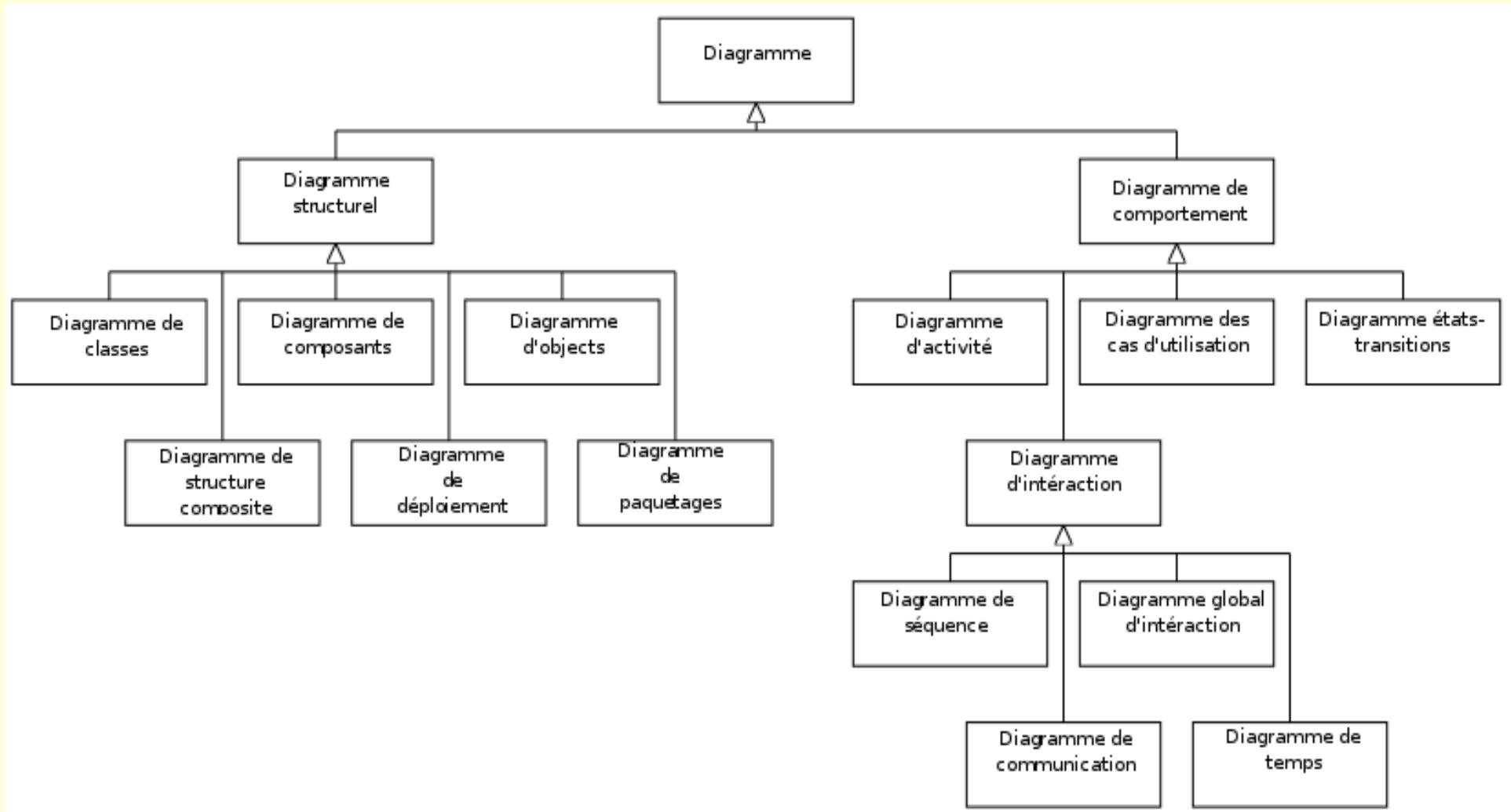
Les vues

- Processus : synchronisation des processus, vue utile dans le cas d'architectures complexes multi-tâches.
- Déploiement : elle précise l'architecture informatique de production

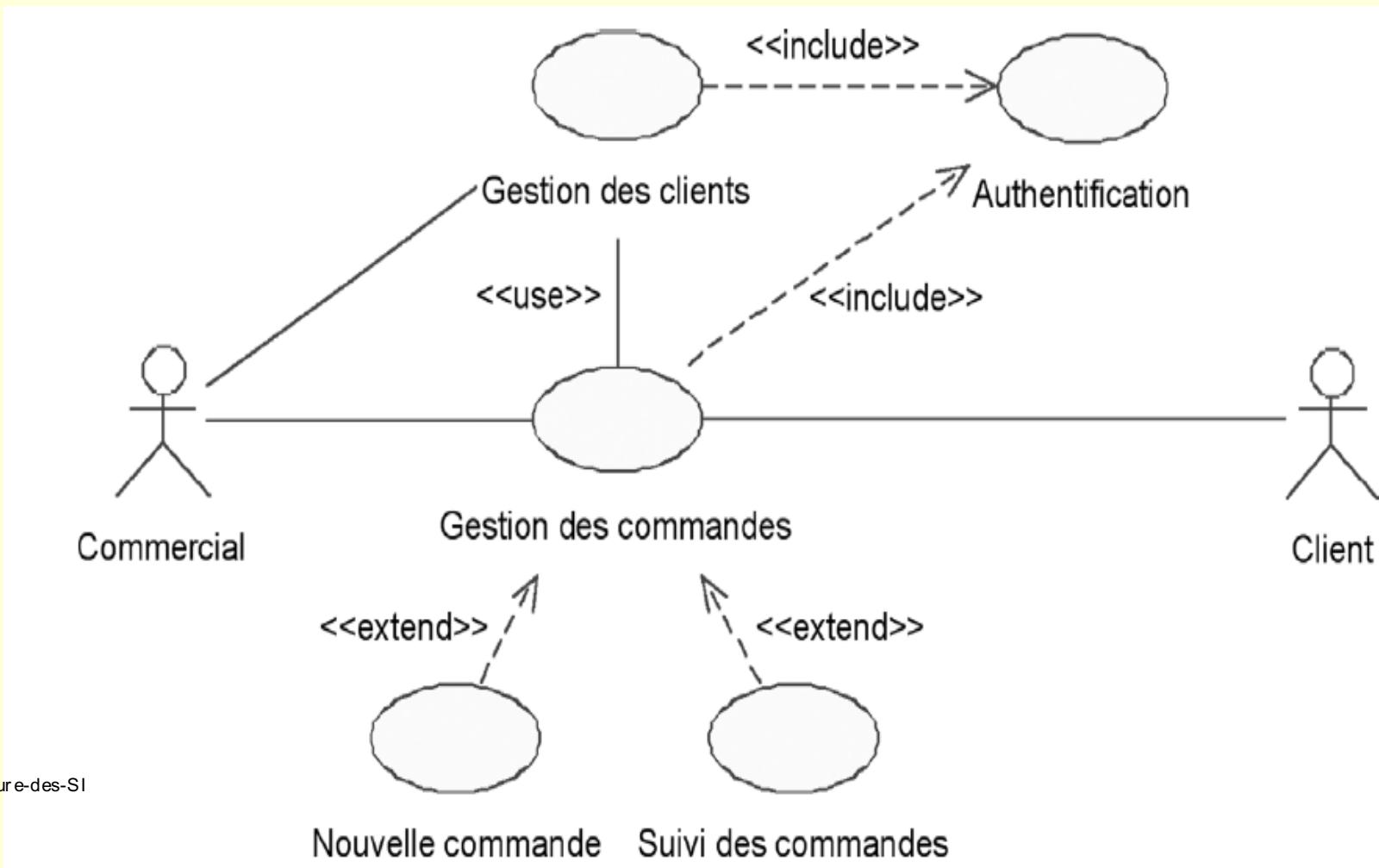
La vue « cas d'utilisation »

- Description centrale
- Système "utilisé" par les acteurs
- Correspond aux besoins attendus par chaque acteur (QUOI et QUI).

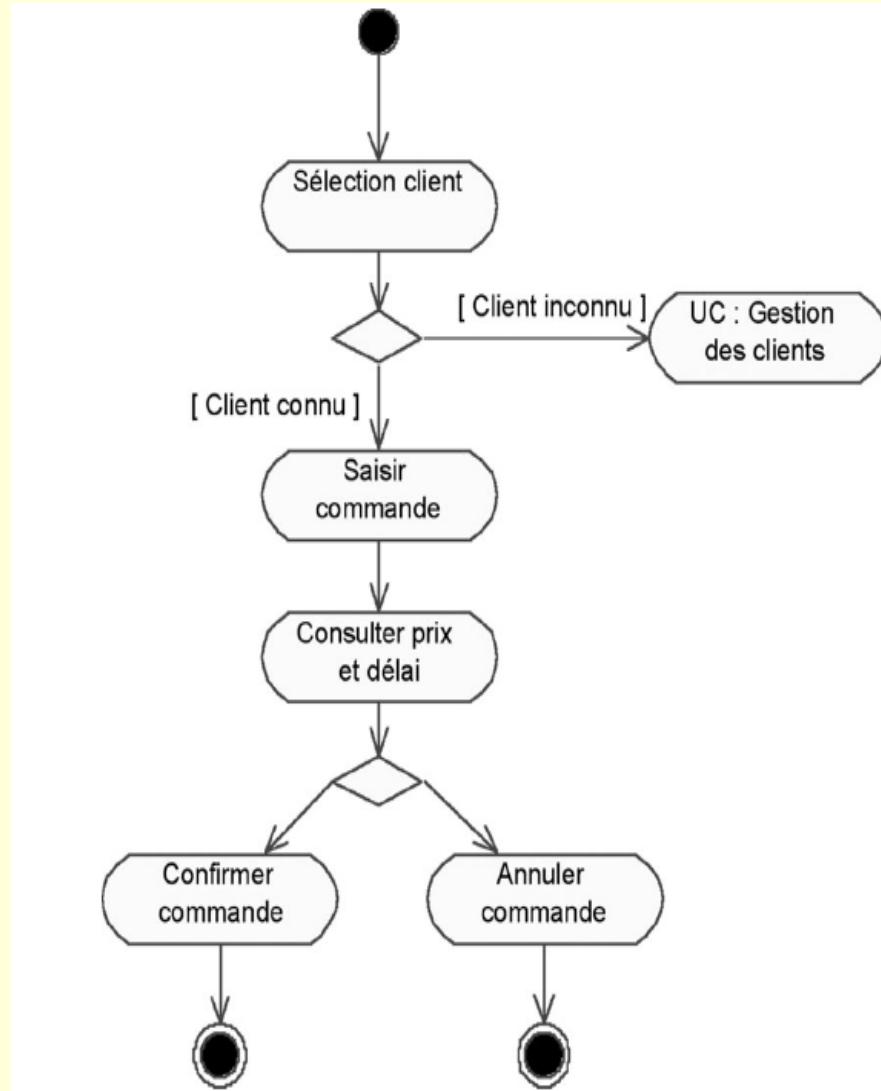
Les diagrammes



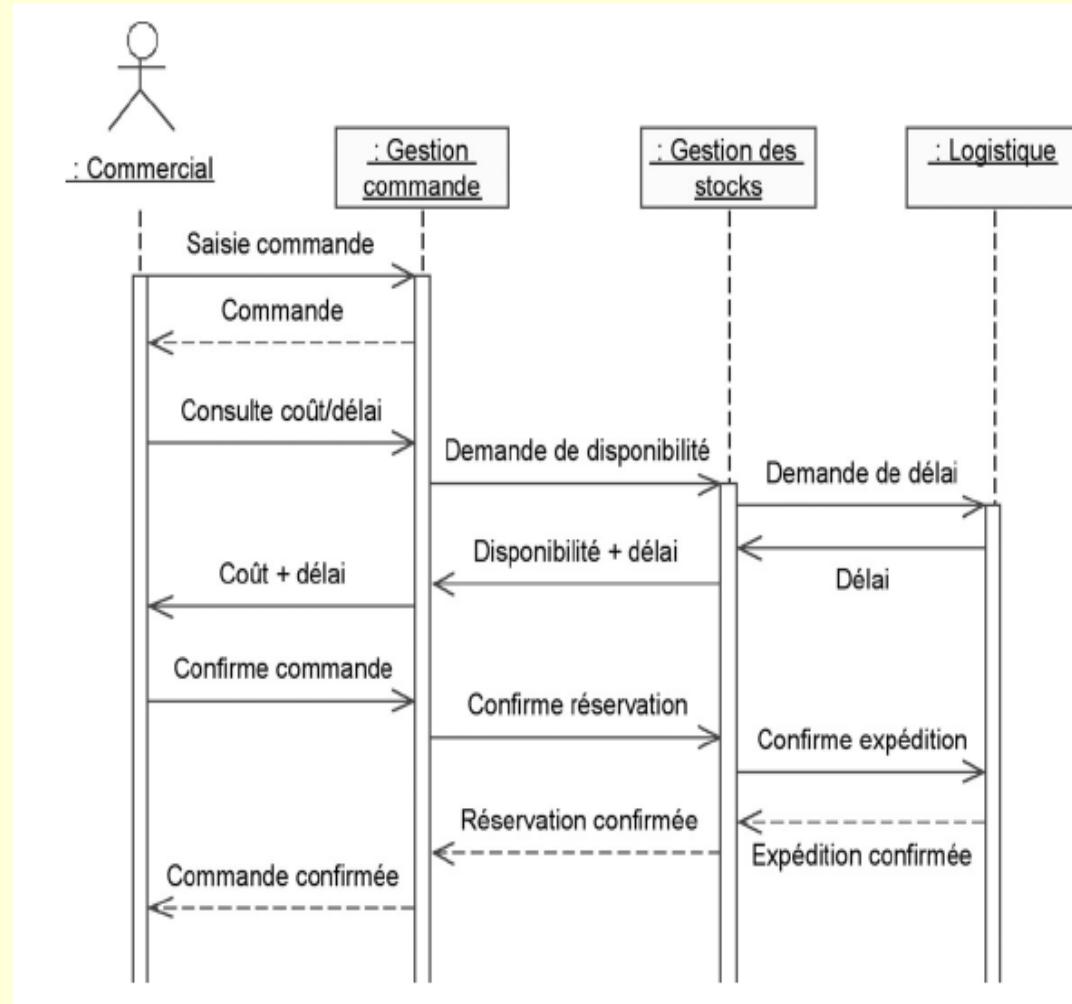
Exemple de diagramme de cas d'utilisation



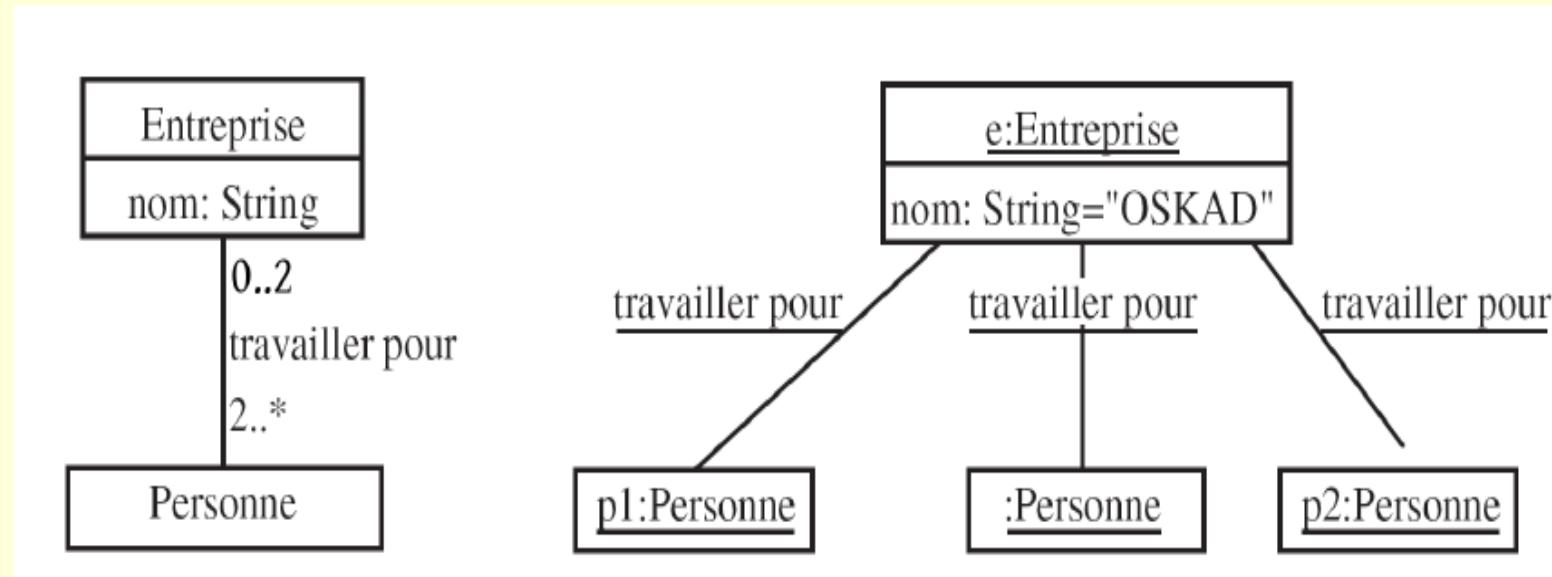
Exemple de diagramme d'activité



Exemple de diagramme de séquences



Exemple de diagramme de classes et d'objets



Tiré de <http://laurent-audibert.developpez.com/Cours-UML/>

Des outils logiciels

Article Discussion Lire Modifier Afficher l'historique Rechercher

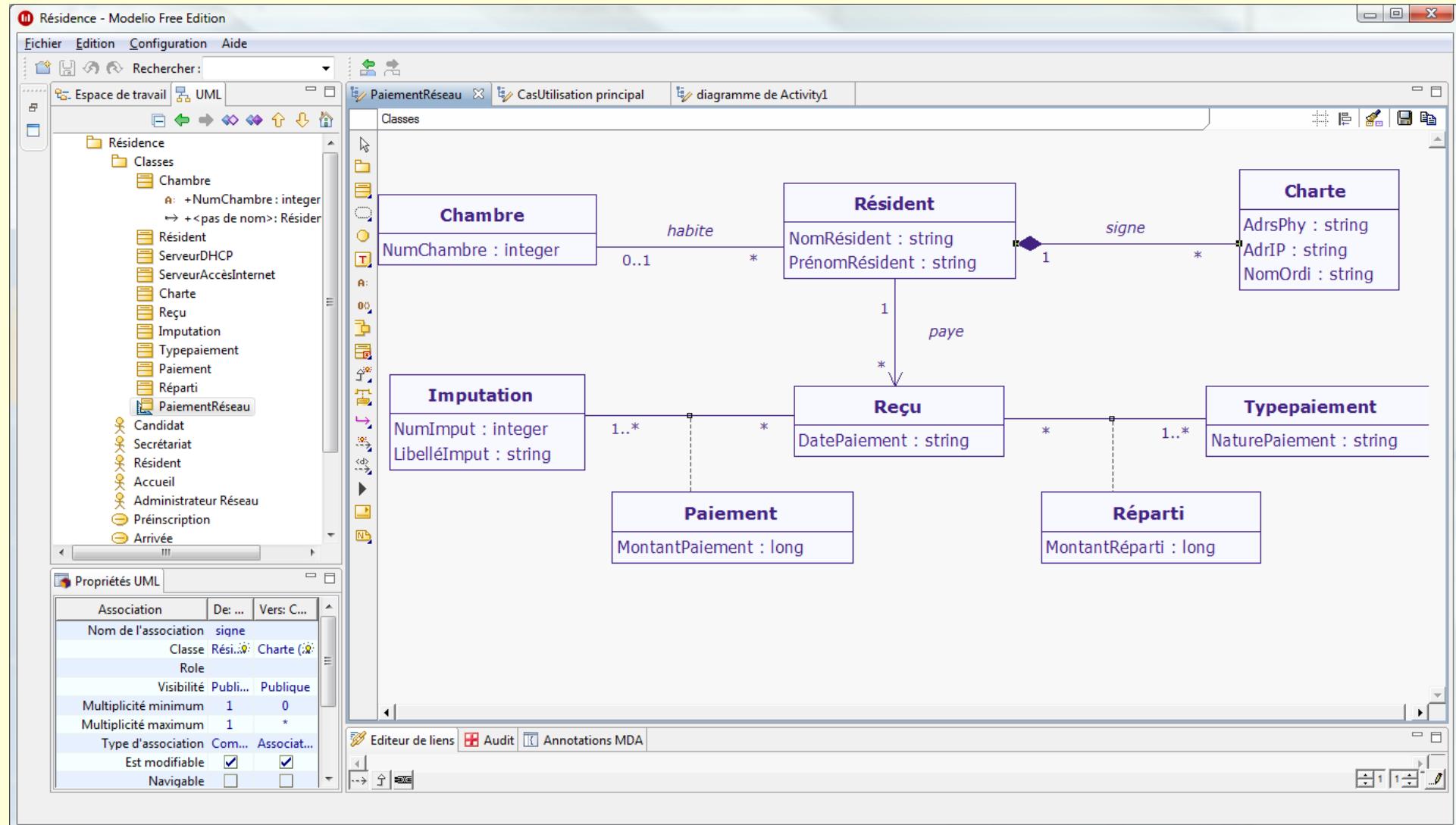
Comparaison des logiciels d'UML

Cet article compare les logiciels d'UML.

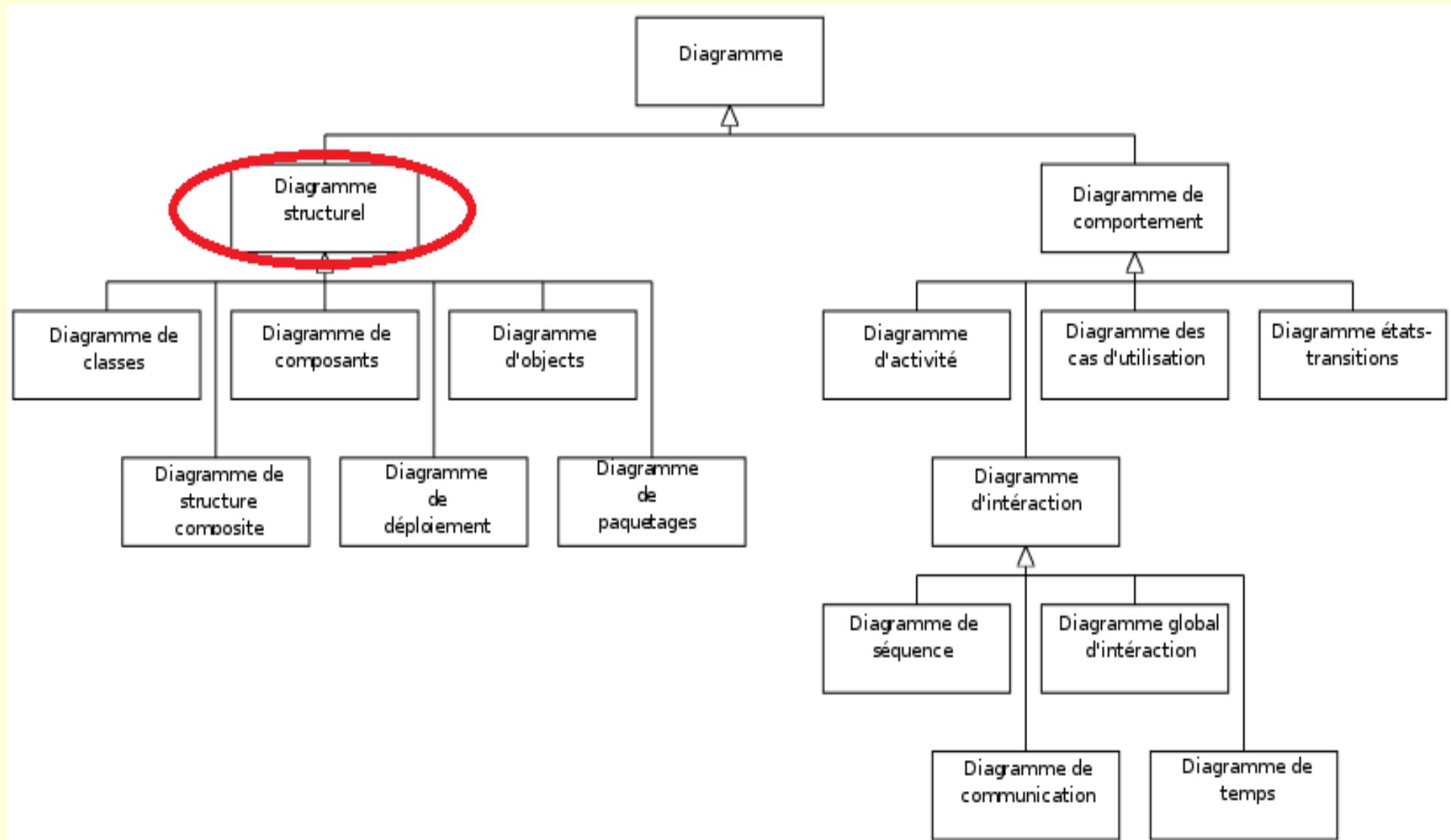
Général [modifier]

Nom	Créateur	Plateforme / OS	Première diffusion publique	Dernière diffusion stable	Open source ?	Licence logiciel	Langage de programmation utilisé
eUML2 Studio	Soyatec	Multiplate-forme (Java / Eclipse)	2003	2011	✗ Non	Free, Commercial	Java
Acceleo	Obeo	Multiplate-forme (Java / Eclipse)	2006-03	2011-02-14	✓ Oui	EPL	Java
AmaterasUML		Multiplate-forme (Java)	2005-12-31	2009-04-19	✓ Oui	EPL v1.0	Java
ArgoUML	Tigris.org	Multiplate-forme (Java)	1998-04	2011-01-30	✓ Oui	EPL v1.0	Java
Astade		Multiplate-forme		2010-03-11	✓ Oui	GPL	Basé sur wxWidgets
ATL	OBEO, INRIA la communauté du logiciel libre	Multiplate-forme (Java)		2009-06-22	✓ Oui	EPL	Java
BOUML	Bruno Pagès	Multiplate-forme	2005-02-26	2011-02-06	✓ Oui	GPL	C++/Qt
Dia	Alexander Larsson/GNOME Office	Multiplate-forme (GTK+)	2004?	2009-05-03	✓ Oui	GPL	C
Eclipse UML2 Tools	Eclipse Foundation	Multiplate-forme (Java)	En développement	À venir	✓ Oui?	GPL?	Java

Exemple du produit Modelio



Les diagrammes



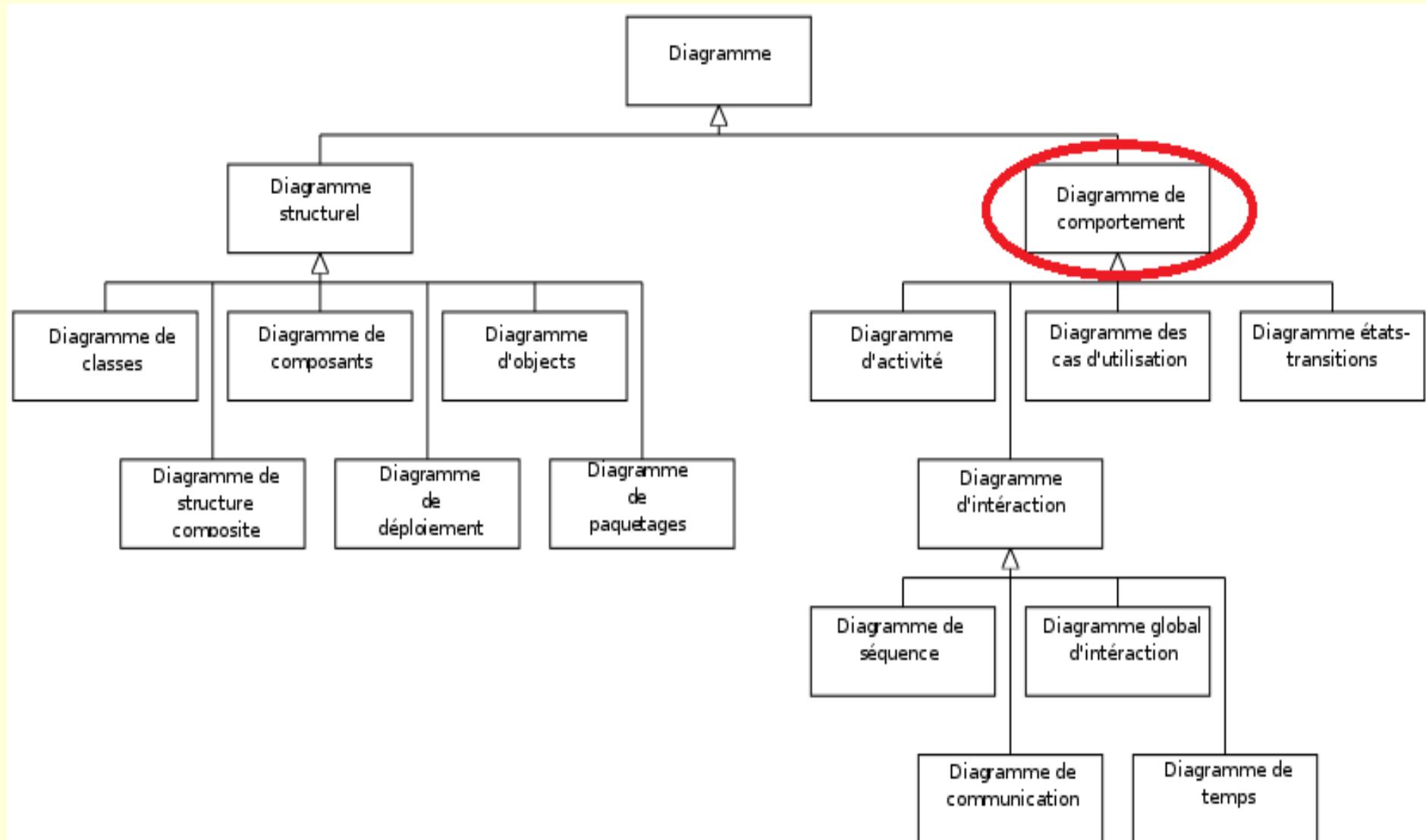
Diagrammes Structurels ou Diagrammes statiques

- Diagramme de classes : classes intervenant dans le système.
- Diagramme d'objets : instances de classes (objets) utilisées dans le système.
- Diagramme de composants : permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...)

Diagrammes Structurels ou Diagrammes statiques (suite)

- Diagramme de déploiement : représente les éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent avec eux.
- Diagramme des paquetages : relations entre des ensembles de classes regroupées en paquetages (UML 2)
- Diagramme de structures composites : relations entre sous-systèmes (UML 2)

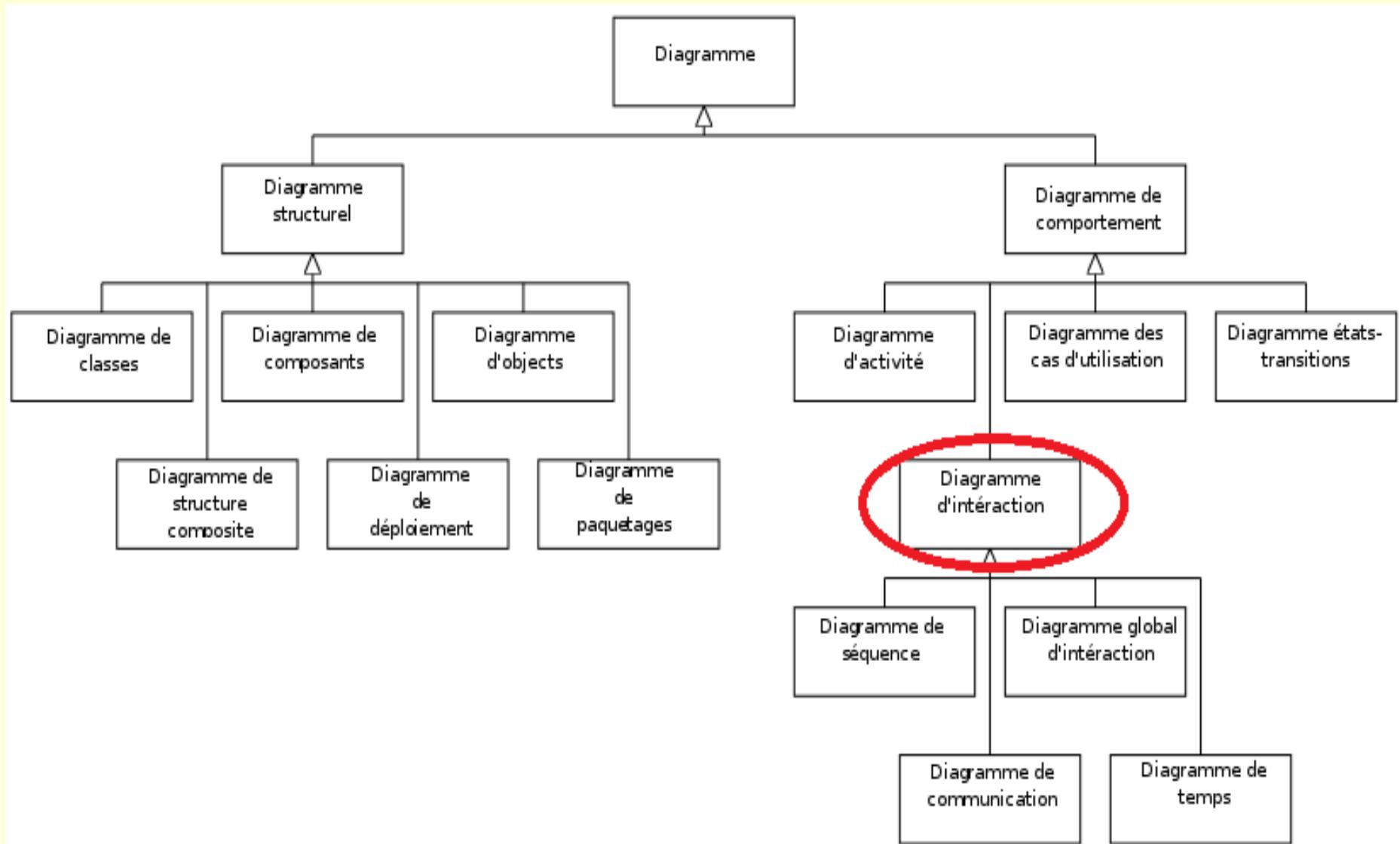
Les diagrammes



Diagrammes Comportementaux ou Diagrammes dynamiques

- Diagramme des cas d'utilisation : décrit les possibilités d'interaction entre le système et les acteurs, c'est-à-dire toutes les fonctionnalités que doit fournir le système.
- Diagramme États-Transitions : montre la manière dont l'état du système (ou de sous-parties) est modifié en fonction des événements du système.
- Diagramme d'activité : variante du diagramme d'états-transitions, permet de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables (multi-threads ou multi-processus).

Les diagrammes



Diagrammes d'interactions

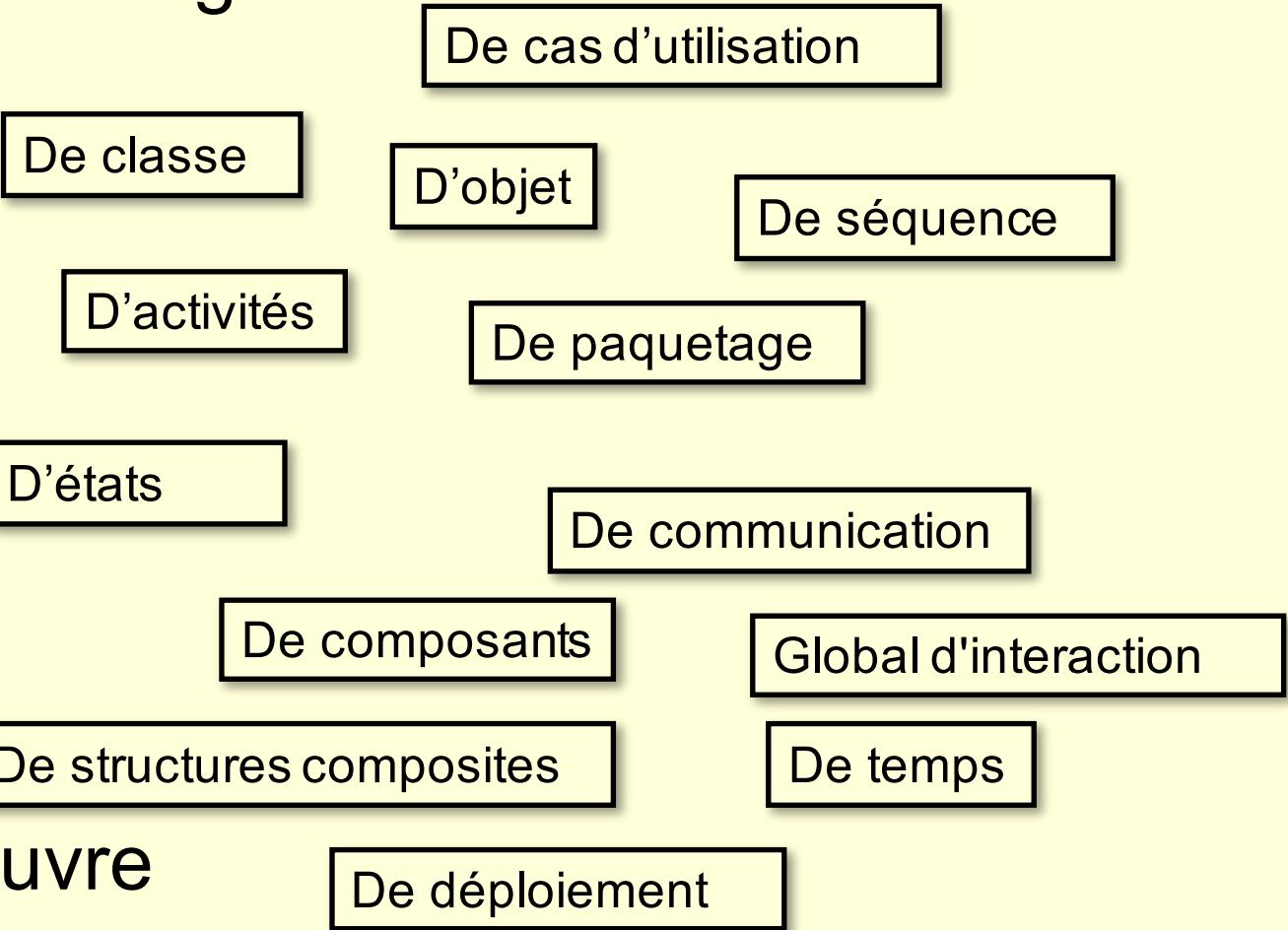
- Diagramme de séquence : représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou des acteurs.

Diagrammes d'interactions (suite)

- Diagramme de communication : représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
- Diagramme global d'interaction : variante du diagramme d'activité où les nœuds sont des interactions.
- Diagramme de temps : représentation des interactions où l'aspect temporel est mis en valeur.

Utilisation des diagrammes

De la maîtrise d'ouvrage



2. Modélisation des objets

Un monde d'objets

- Les objets du monde réel nous entourent, ils naissent, vivent et meurent
- Les objets informatiques définissent une représentation simplifiée des entités du monde réel
- Les objets représentent des entités matérielles ou abstraites (concept)
- Un objet encapsule le comportement et la connaissance d'une abstraction

Les objets sont des abstractions

- **Abstraction** : « faculté des humains de se concentrer sur l'essentiel et d'oublier les détails » pour découvrir des concepts.
 - Une abstraction est un résumé, un condensé
 - Mise en avant des caractéristiques essentielles
 - Dissimulation des détails
 - Une abstraction se définit par rapport à un point de vue
 - Un objet n'existe pas de façon isolée : fournit un ensemble de services, ou demande un ensemble de services

abstractions

Exemple de concepts concrets (créés à partir d'exemples réels) :

milou, rintintin, flipper, lassy, ...

Concept chien (poils, dents, 4 pattes,), Concept dauphin

Exemple de concepts abstraits (créés à partir d'un raisonnement) :

Concept de mammifère

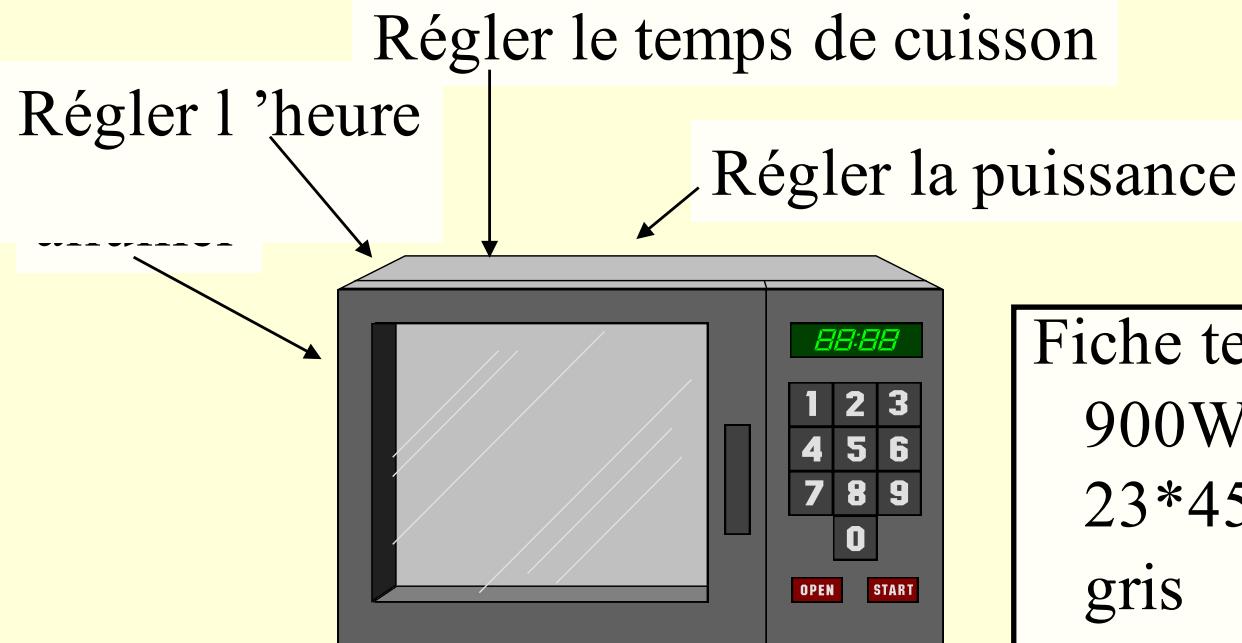
Les concepts dépendent du contexte (point de vue) :

exemple sur le concept de voiture

trois personnes : vendeur de voiture, assureur, mécanicien

trois points de vue : marchandise, contrat assurance, système mécanique

Caractéristiques fondamentales des objets



- L'état
- Le comportement
- L'identité

9:35

L'état d'un objet

- L'état regroupe les valeurs instantanées de tous les attributs d'un objet
- L'état évolue au cours du temps
- L'état d'un objet à un instant donné est la conséquence de ses comportements passés

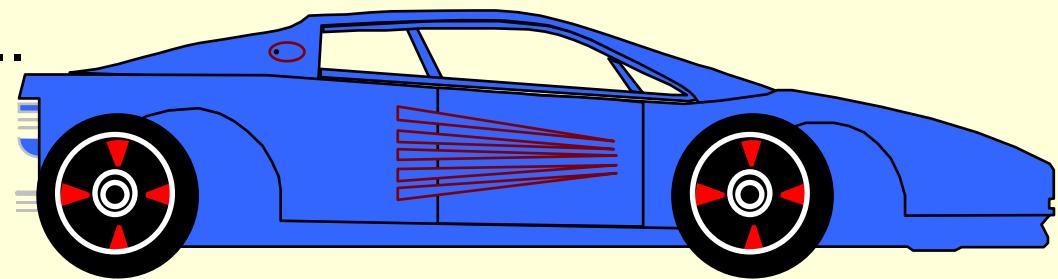
Exemples

- Pour un signal électrique : l'amplitude, la fréquence, la phase...
- Pour une voiture : la marque, la puissance, la couleur, le nombre de places assises ...

Une voiture
30 litres

Après un parcours
de 100 Km

Une voiture
20 litres



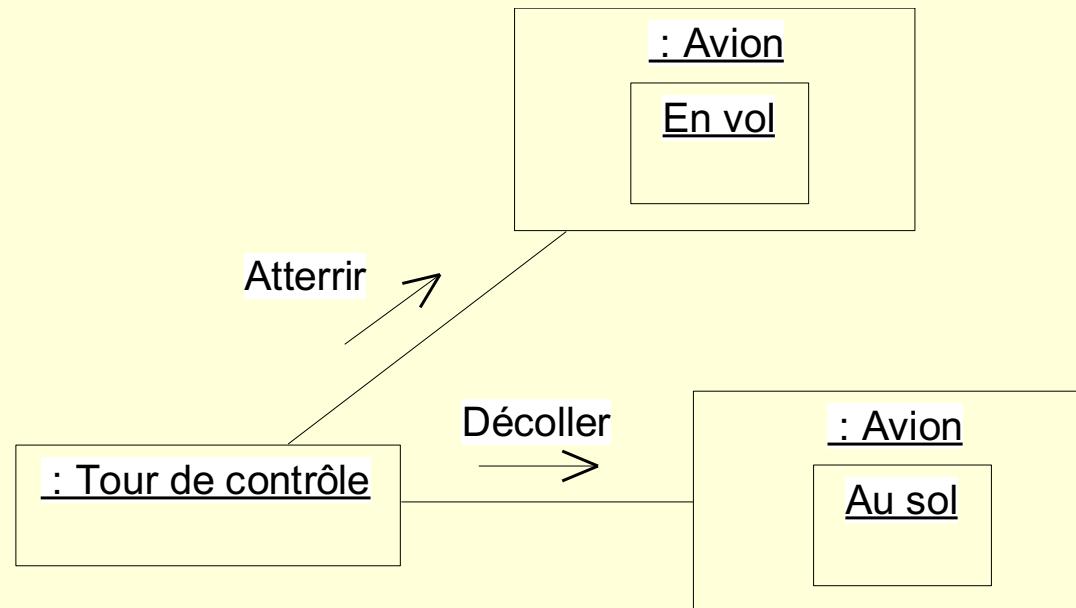
Une voiture
Bleu
979 kg
12 CV

Le comportement

- Le comportement décrit les actions et les réactions d'un objet
- Le comportement regroupe toutes les compétences d'un objet
- Le comportement peut se représenter sous la forme d'opérations ou de fonctions appelées **méthodes**

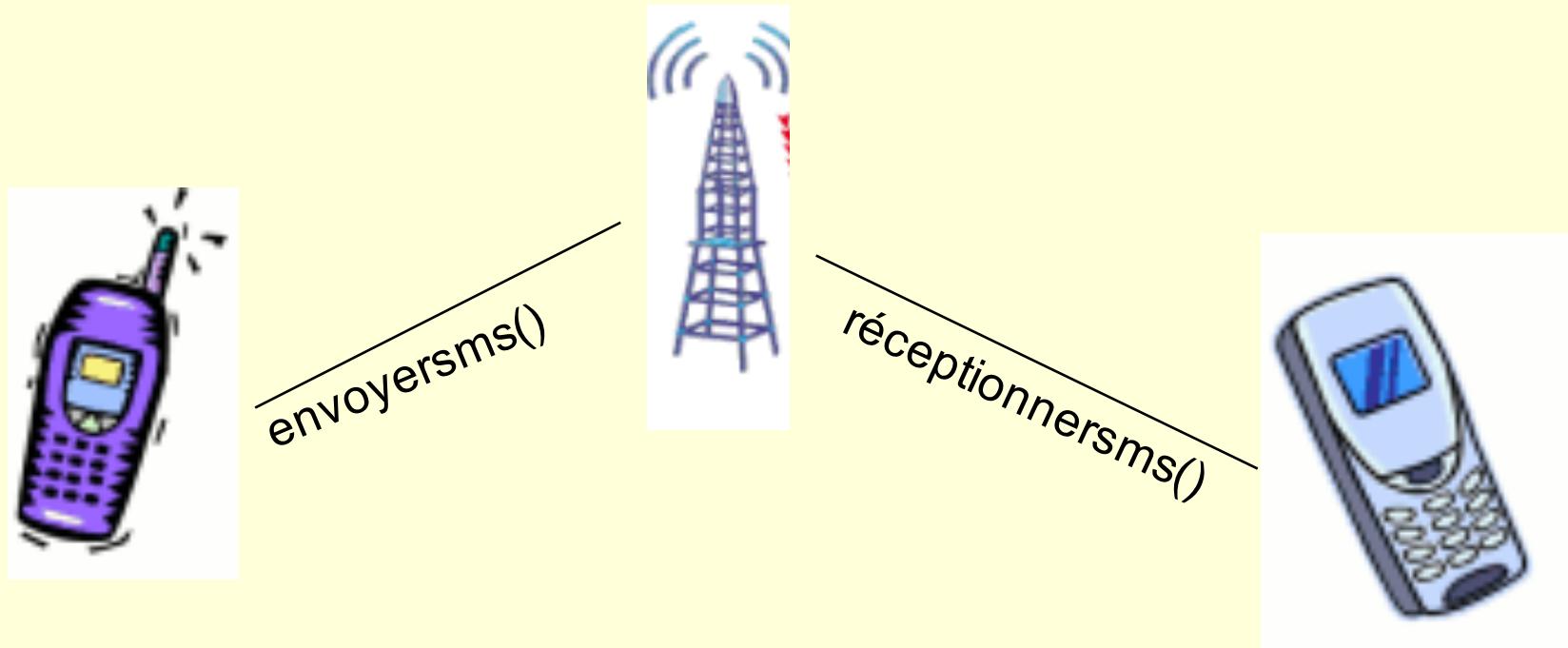
Comportement (suite)

- Etat et comportement sont liés :
 - Le comportement dépend de l'état
 - L'état est modifié par le comportement

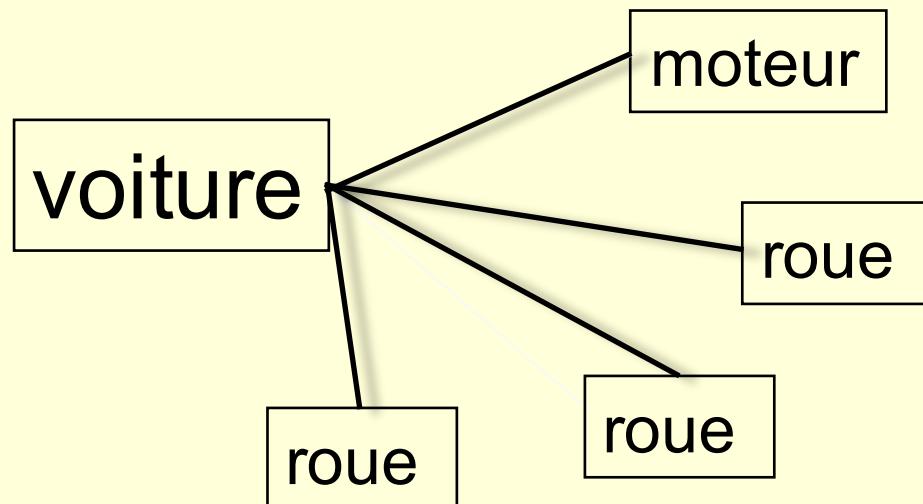


Les objets interagissent

Les objets émettent et reçoivent des messages



Les objets sont associés

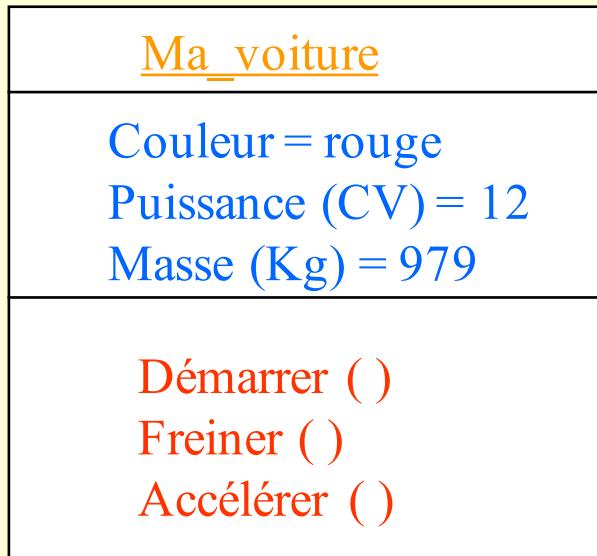


Définition de l'objet

Objet : élément matériel ou immatériel, dans la réalité étudiée, qui satisfait aux principes de *distinction*, de *permanence* et d'*activité*.

Un objet possède une *identité*, un *état* et un *comportement*.

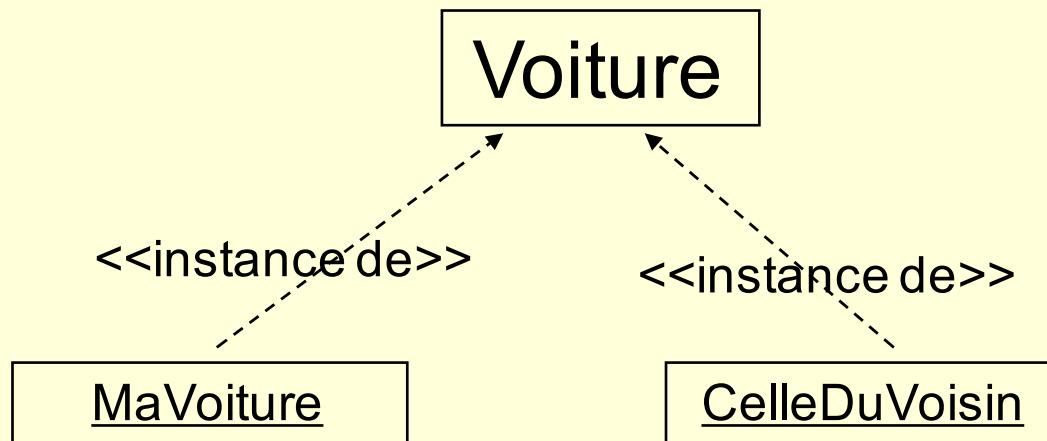
Représentation UML d'un objet



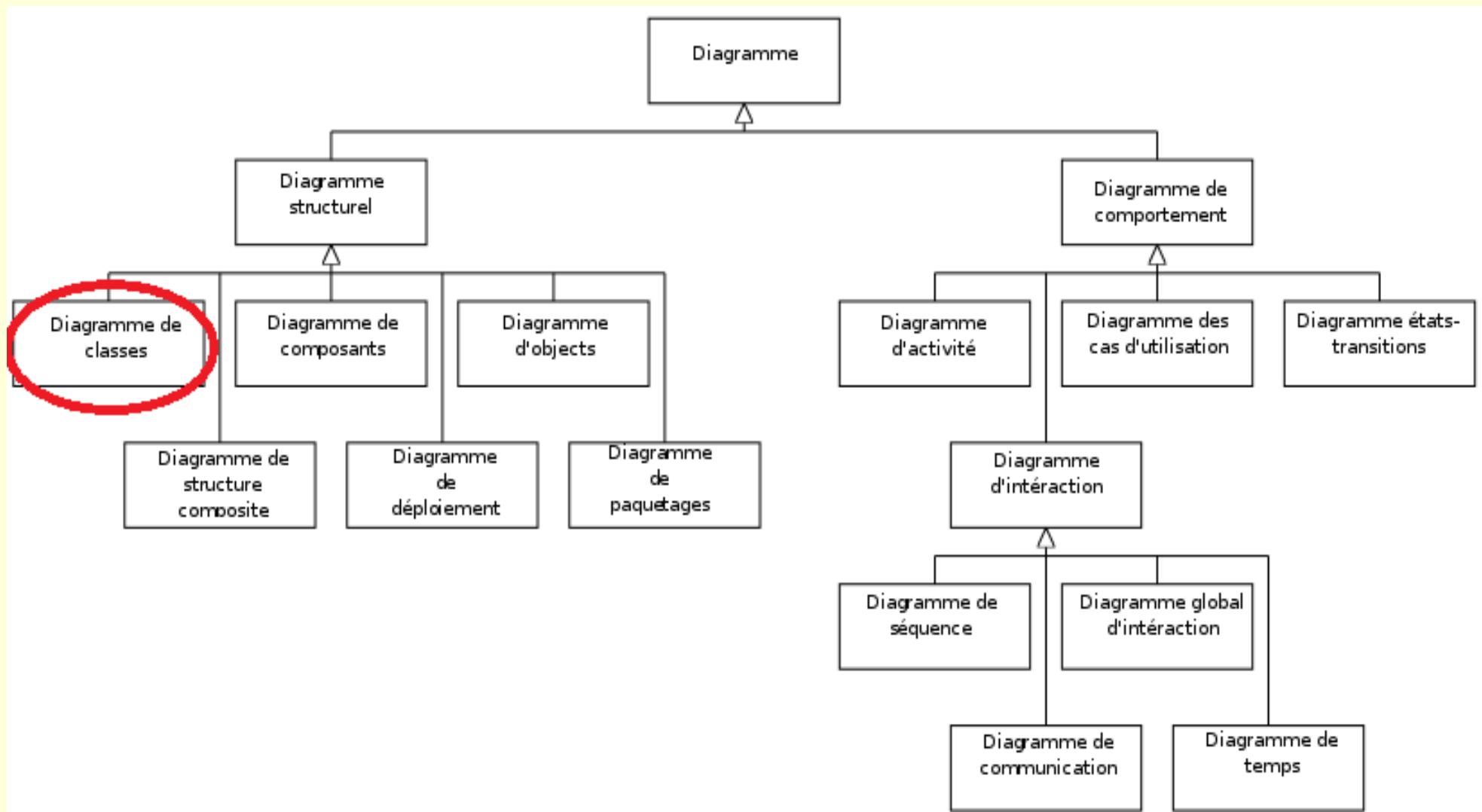
- L'**identité** permet de distinguer tout objet de façon non ambiguë, indépendamment de l'état
- L'**état** regroupe les valeurs instantanées de tous les attributs d'un objet
- Le **comportement** décrit les actions et les réactions d'un objet (méthodes)

Objets et classes

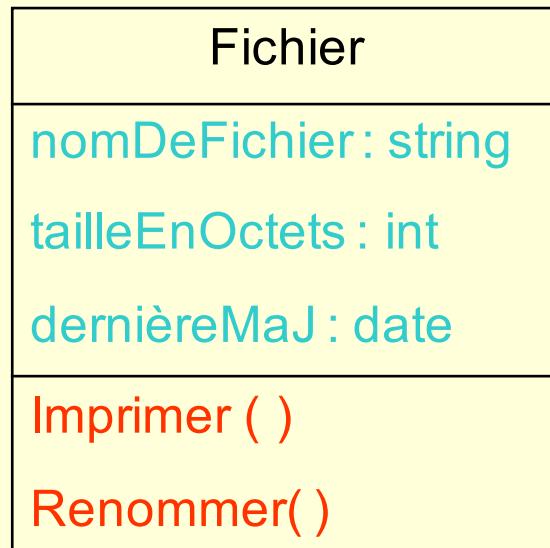
- Abstraction des objets définit une classe
- Objet est une **instance** d'une classe
- Une classe est un « moule » à objet



Les diagrammes



Représentation UML d'une classe



- Nom de la classe
- Liste des **attributs** avec le **type**
- Opérations ou **méthodes** spécifiques à la classe

Notation d'instance

<u>42161:Fichier</u>
nomDeFichier : poly.doc
tailleEnOctets : 4500
dernièreMaj : 8/12/2011
Imprimer ()
Renommer()

<u>608237:Fichier</u>
nomDeFichier : prés.ppt
tailleEnOctets : 13800
dernièreMaj : 11/12/2011
Imprimer ()
Renommer()

Notation anonyme

<u>:Fichier</u>
nomDeFichier : prés.pdf
tailleEnOctets : 3800
dernièreMaj : 11/01/2012
Imprimer ()
Renommer()

Opérations et méthodes

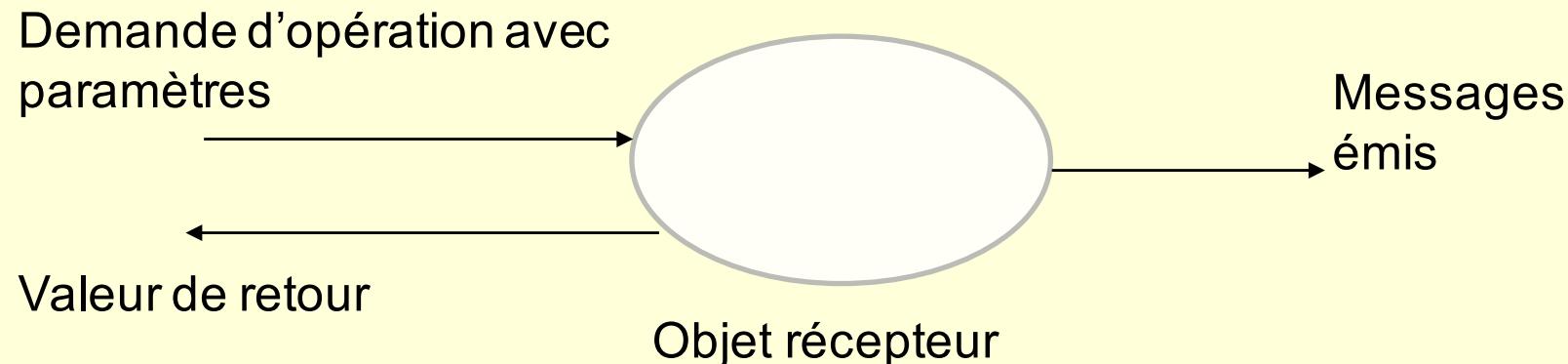
- Nom
- Paramètres (optionnel)
 - Nom
 - Type
 - Entrées/Sorties (In/Out)
- Valeur de retour

Opération : signature = nom + paramètres

*Méthode : comportement, traitement (=
implémentation d'une opération)*

Messages

- Demande faite par l'objet émetteur à l'objet récepteur d'exécuter l'une de ses opérations



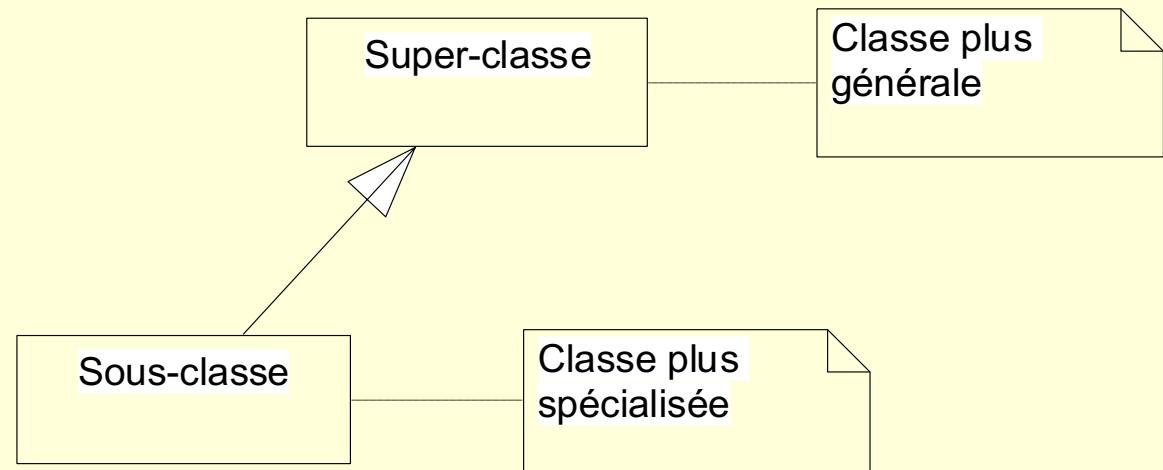
Constructeur, destructeur

- Constructeur : opération définie pour la création (instanciation) d'un objet.
- Destructeur : opération définie pour la fin d'un objet.

Correspondent à des messages particuliers et n'ont pas de valeur de retour

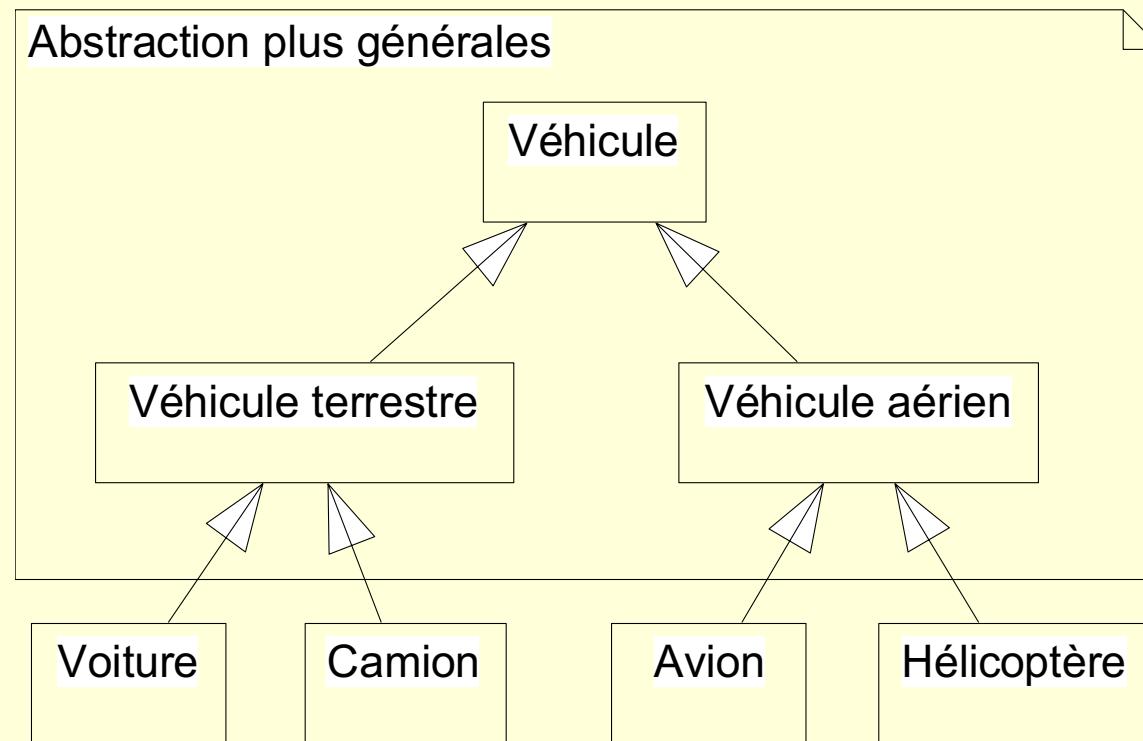
Hiérarchies de classes

- Gérer la complexité
 - Arborescences de classes d'abstraction croissante
- Généralisation
 - Super-classes
- Spécialisation
 - Sous-classes
- Héritage



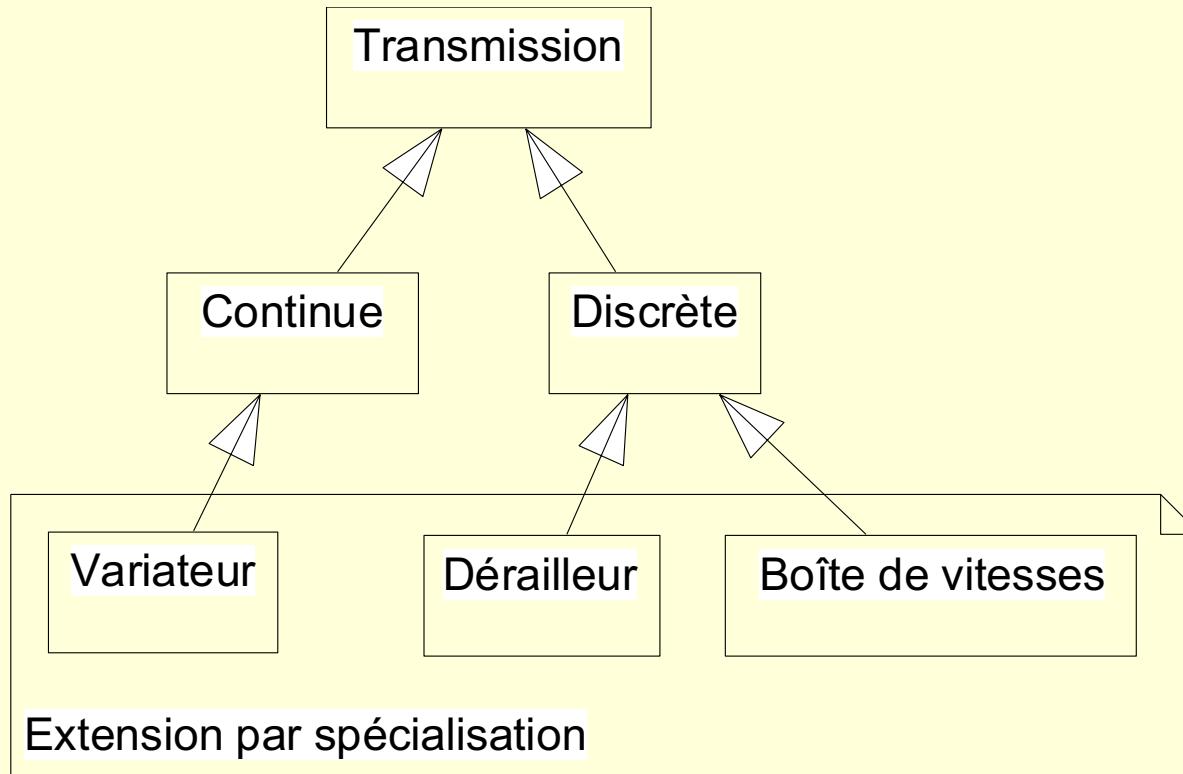
Généralisation

- Factoriser les éléments communs
 - attributs, opérations et contraintes



Spécialisation

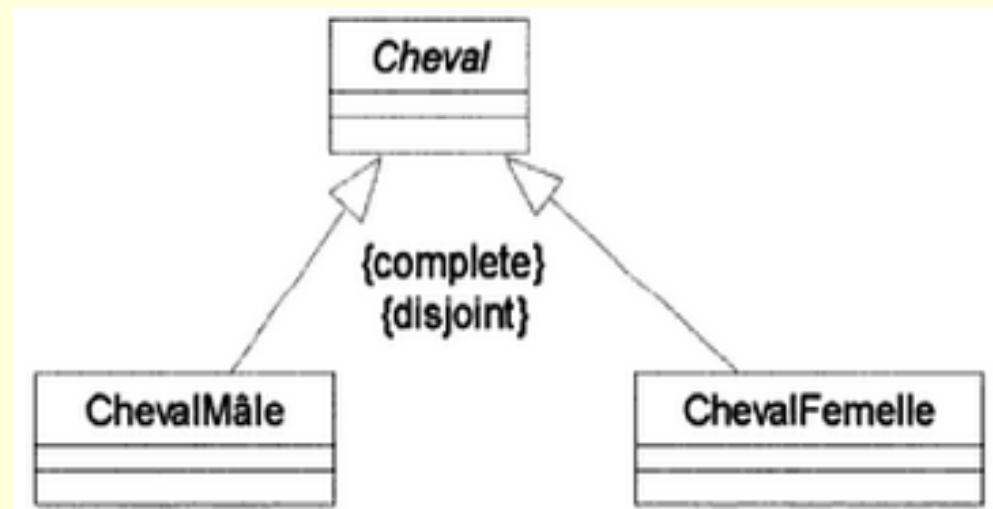
- Extension cohérente d'un ensemble de classes
 - réutilisation du code déjà écrit
 - ajout de nouvelles fonctionnalités



Contrainte sur héritage

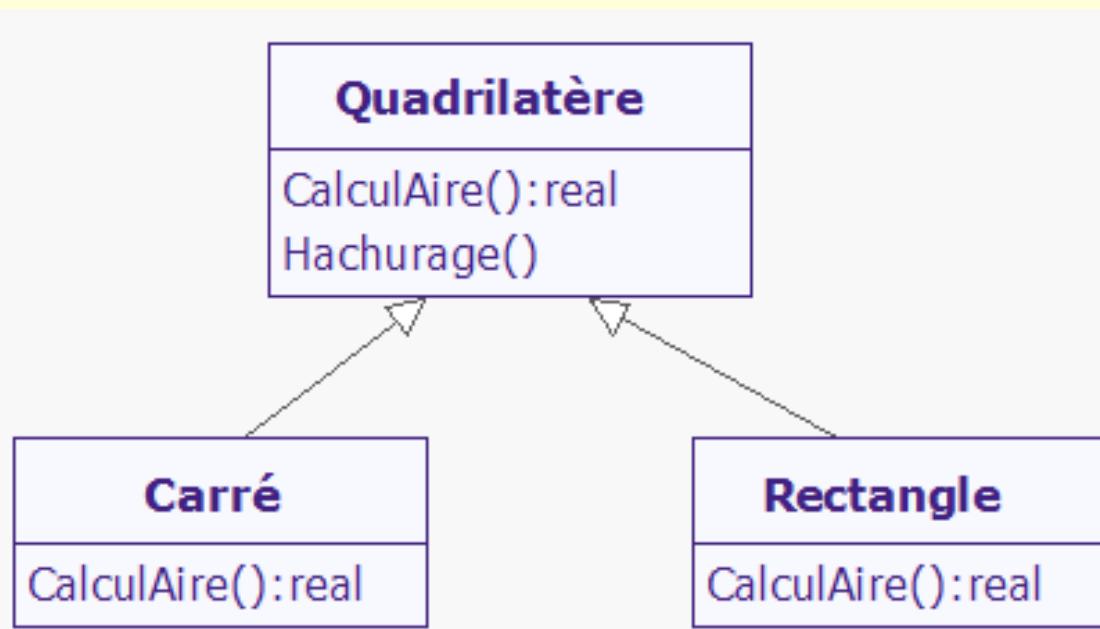
4 types de contraintes sur la relation entre sous-classes et sur-classe :

- La contrainte {incomplete} : ensemble des sous-classes ne couvre pas la surclasse
- La contrainte {complete} : l'ensemble des sous-classes est complet et couvre la surclasse
- La contrainte {disjoint} : aucune instance en commun
- La contrainte {overlapping} : possibilité d'avoir des instances en commun



Polymorphisme : redéfinition

- Une méthode peut-être redéfinie dans une sous-classe.



Un même message va déclencher des traitements différents selon la classe de l'objet qui le reçoit

Polymorphisme : surcharge

- Une classe peut avoir plusieurs méthodes avec le même nom mais des paramètres différents
- Ex : surcharge de constructeur lorsque tous les attributs ne sont pas nécessairement renseignés

Classes abstraites et « Design Pattern »

- But : définir des traitements génériques indépendants des types des attributs et des paramètres des opérations.
- Notation UML : italique



- Nécessité de définir des sous-classes pour avoir des objets.

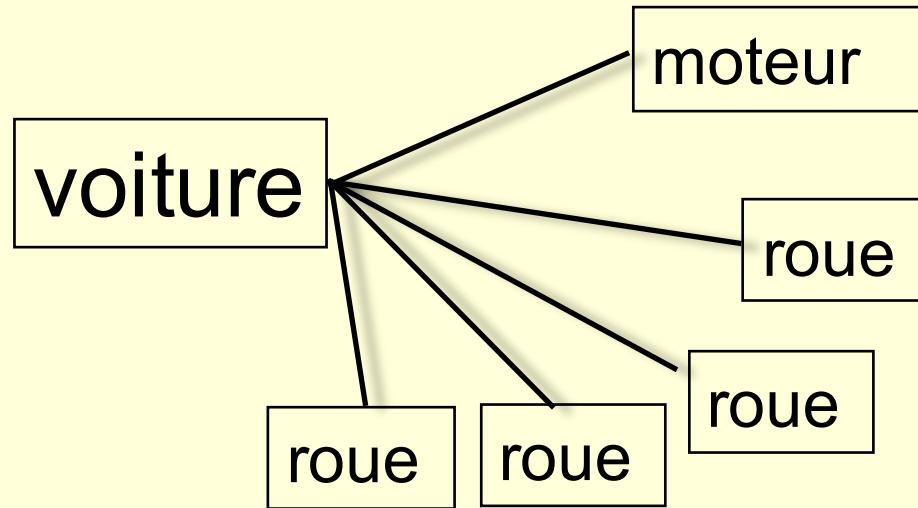
Analyse / Programmation

Analyse et conception « objet »

n’impliquent pas

Langage de programmation objet

Modélisation des associations



Modélisation des données et bases de données

Base de données : ensemble de données évolutives organisé logiquement et physiquement pour être utilisé par des programmes multiples.

Système de gestion de bases de données

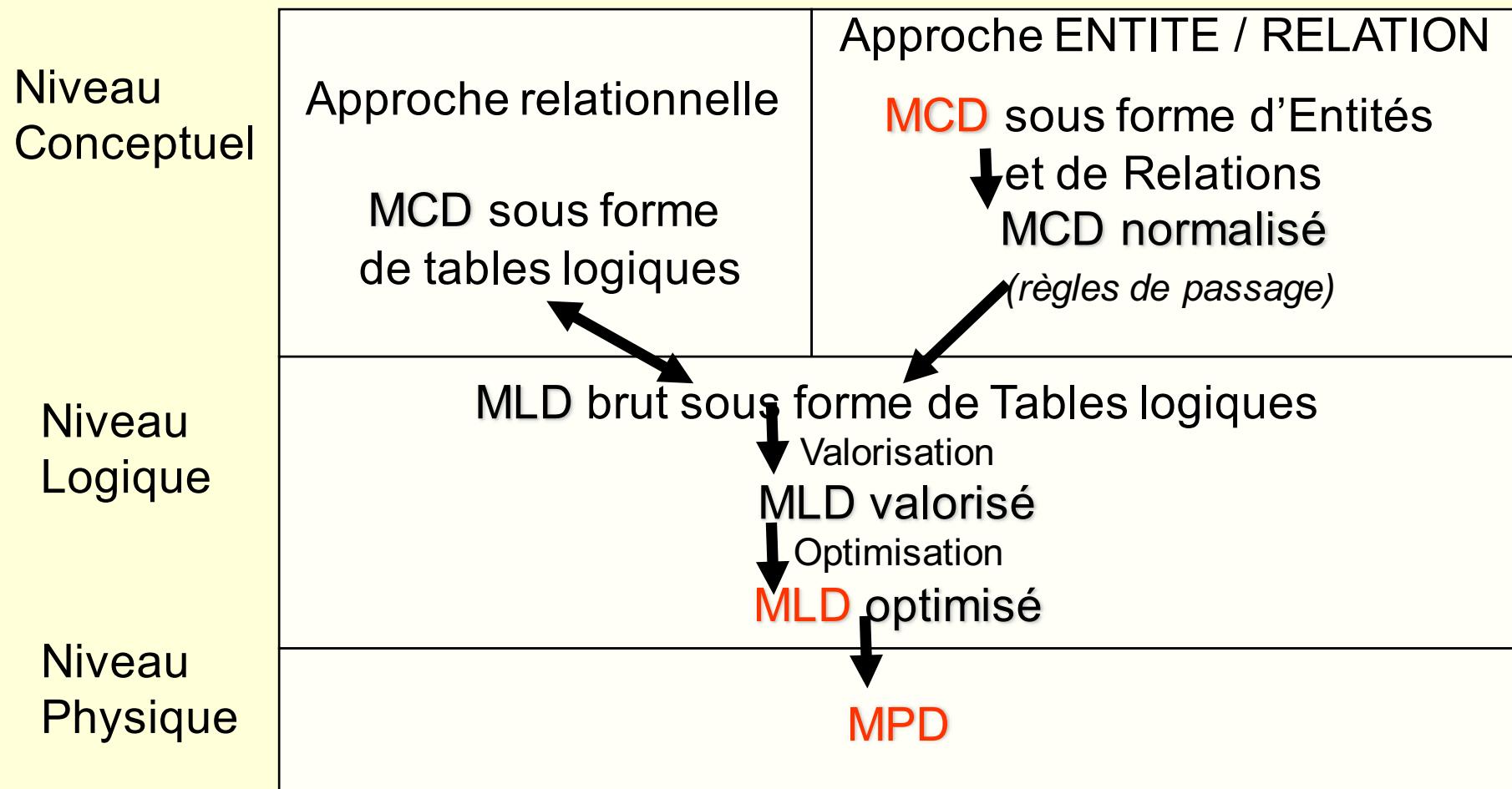
- Composants (programmes) élémentaires d'accès aux données.
- Centraliser et partager l'information.
- Assurer l'indépendance données/traitements.
- Préserver la cohérence et la persistance dans le temps des données.
 - Transactions
 - Sauvegardes

Modèles des SGBD

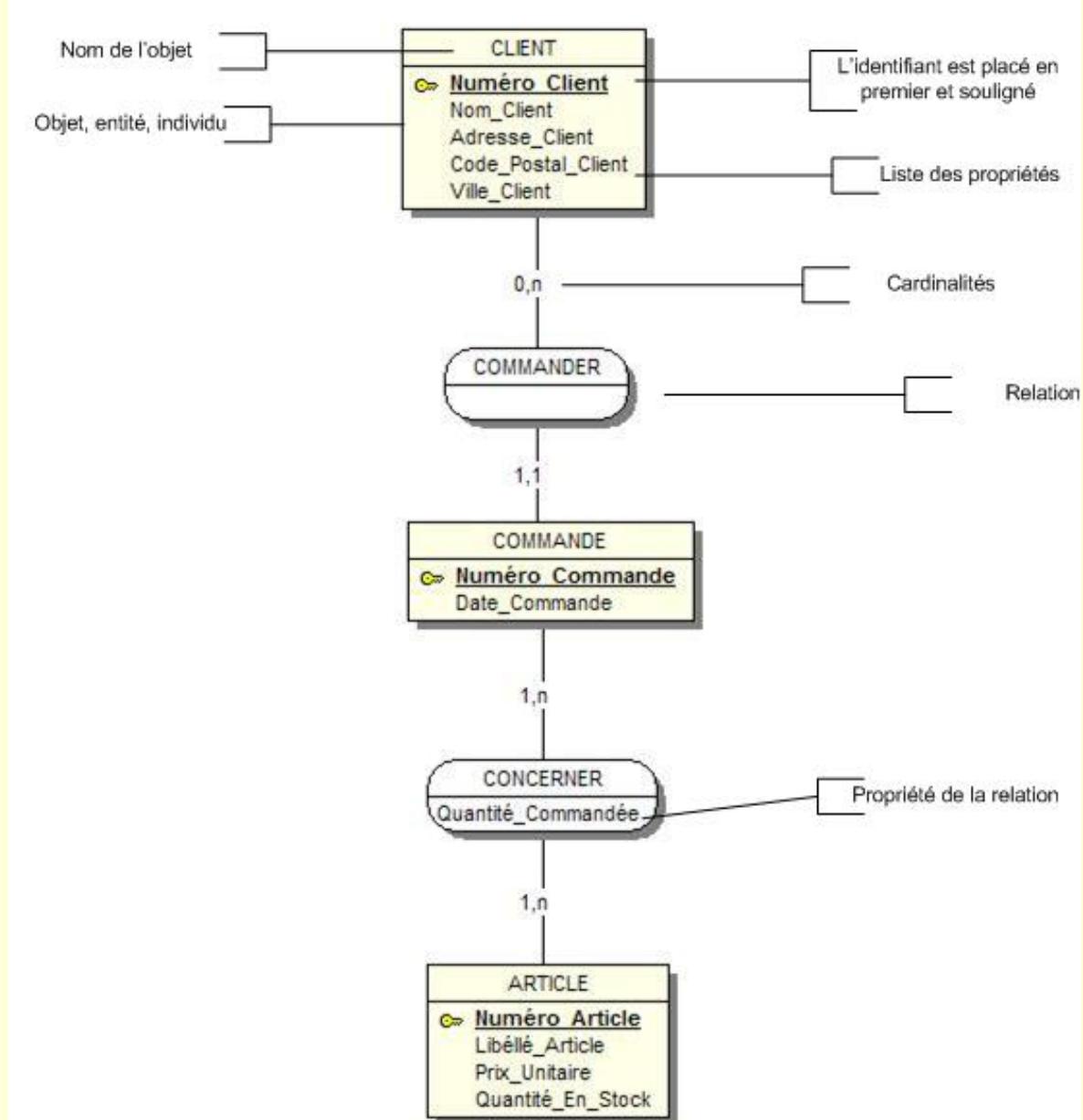
- SGBD relationnels : oracle, db2, SQL server, MySQL, Access,
- SGBD objets (souvent adaptation des grands éditeurs)
- Langage SQL : langage de requête non procédural.

Vous avez aimé Merise

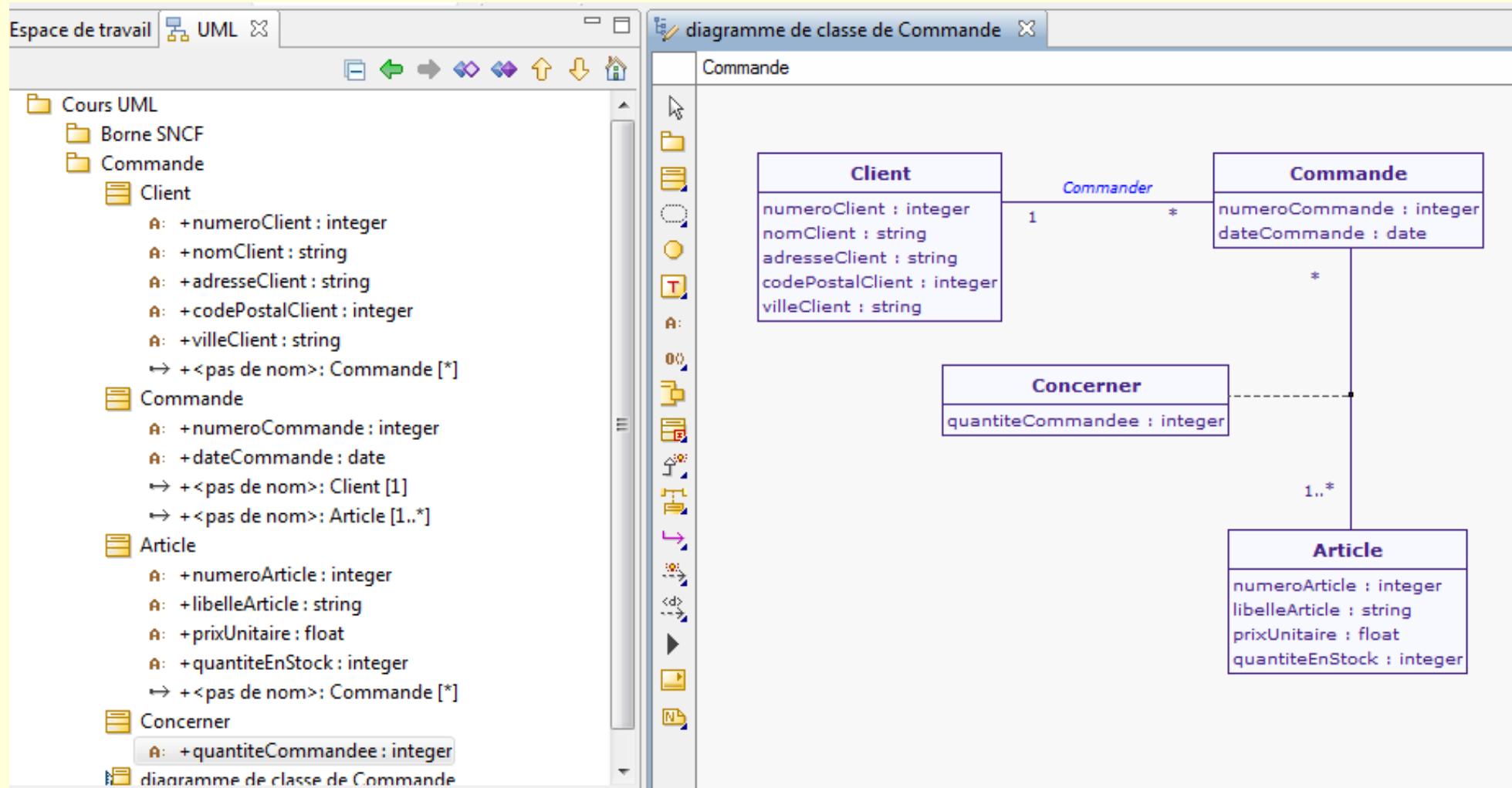
Enchaînement des modèles de Données



Modèle Conceptuel de Données



Vous aimerez le diagramme de classes

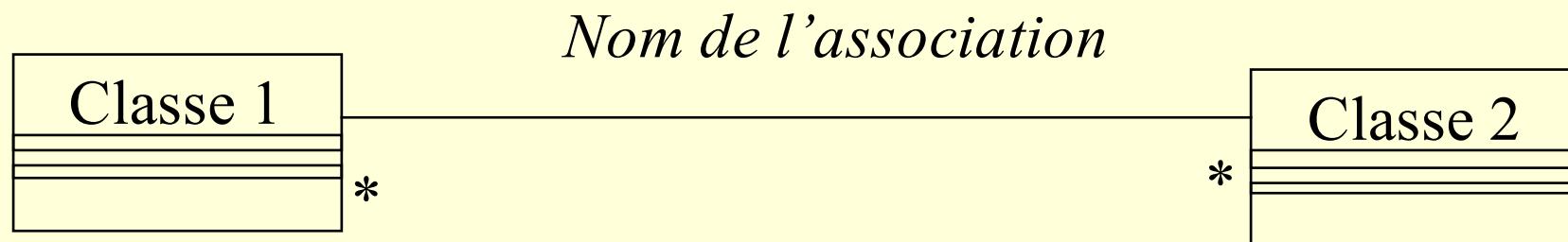


Vérification du modèle : règles de normalisation

- **Règle 1** : toutes les propriétés sont élémentaires (non décomposables).
- **Règle 2** : chaque objet doit posséder un identifiant et un seul.
- **Règle 3** : les propriétés d'un objet autres que l'identifiant doivent être en dépendance fonctionnelle monovaluée de cet identifiant (Ex. il ne peut pas y avoir plusieurs prix unitaires pour un article).

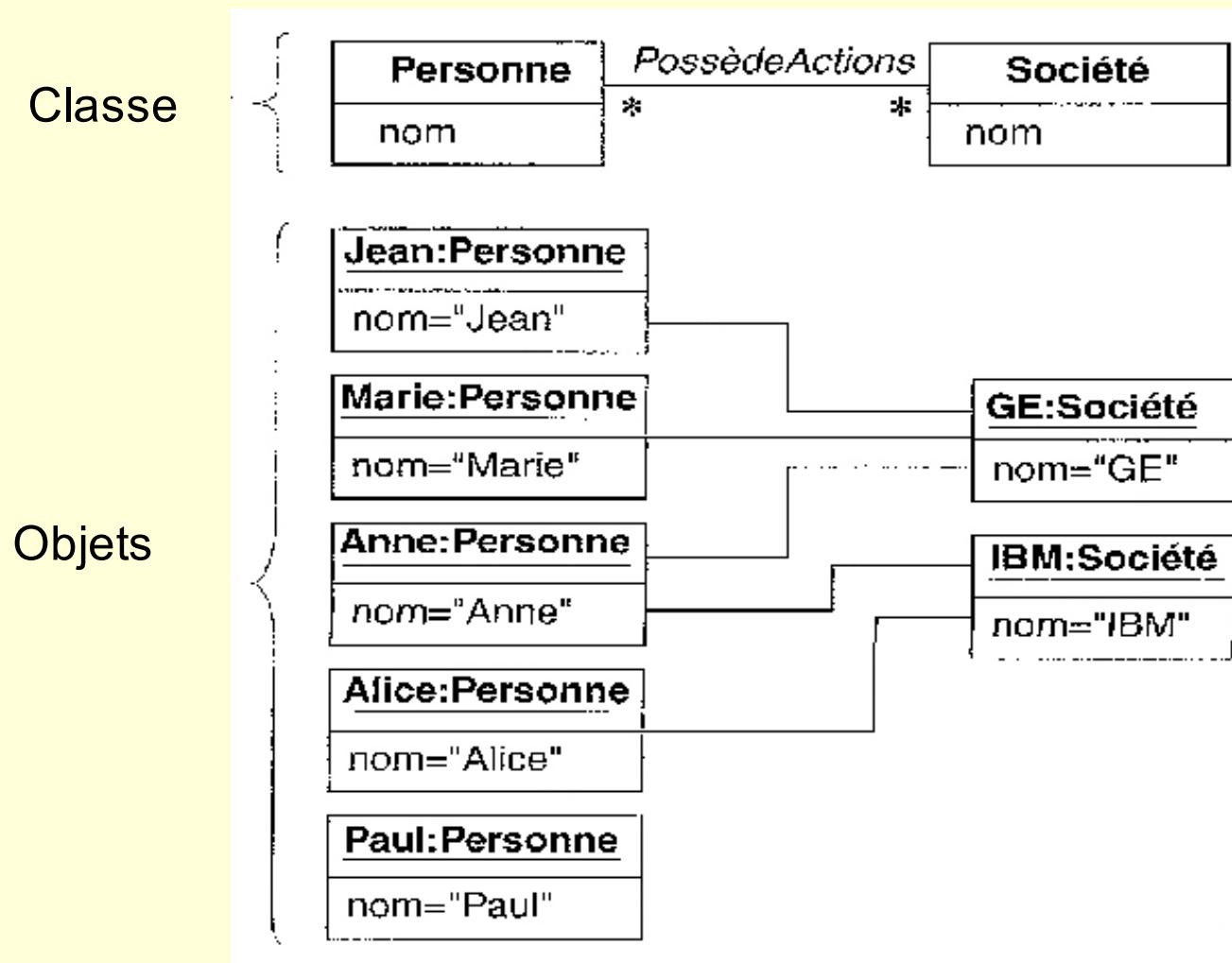
Association entre classes

- Pour décrire les attributs d'une classe on fait souvent référence à des attributs d'autres classes. **Une association** permet cette référence tout en respectant la règle de non redondance.
 - Notation :



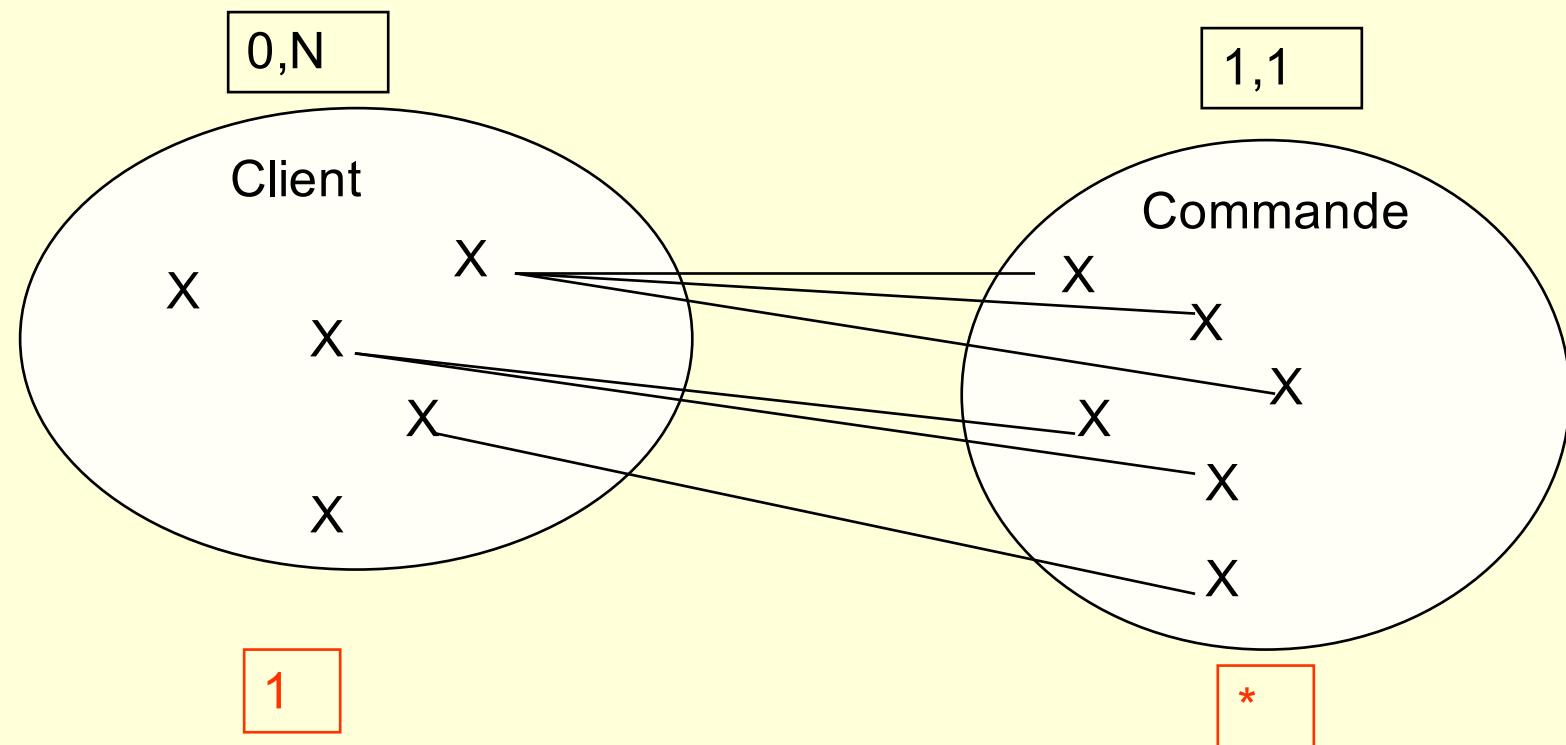
Liens entre objet

- Instance d'une association



Cardinalité / Multiplicité

Cardinalité : nombres minimum et maximum d'occurrences de la relation pouvant exister pour une occurrence de l'objet



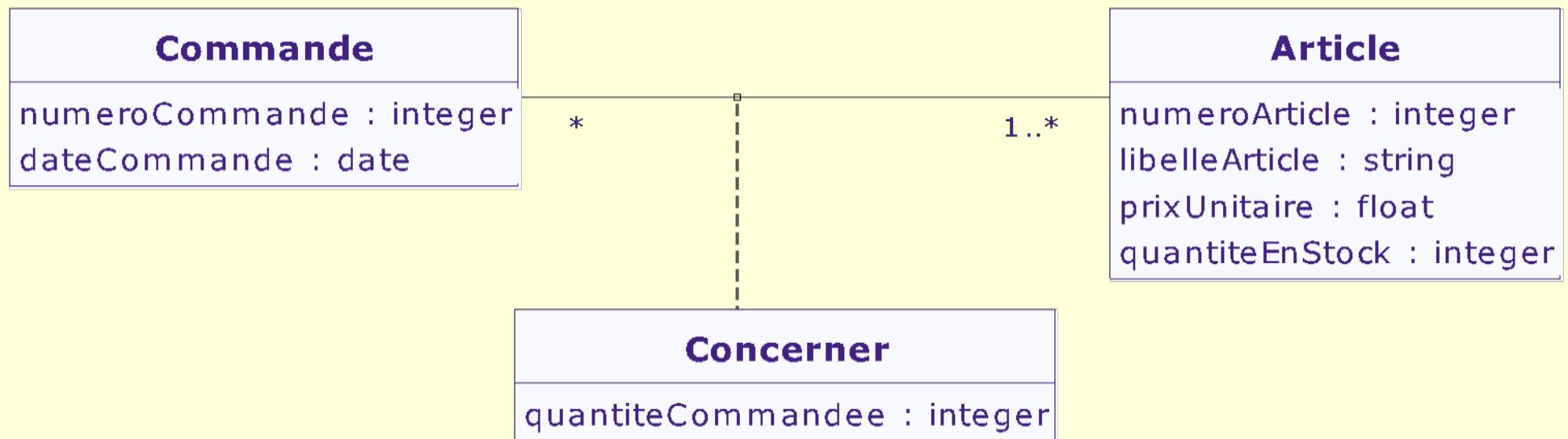
Multiplicité : nombre d'objets de la classe pouvant être associés à un seul et unique objet de la classe à l'autre extrémité

Cardinalité / Multiplicité

	MERISE	UML
Exactement une	1,1	1
Plusieurs (0 ou plus)	0,N	0..* ou *
Optionnelle	0,1	0..1
1 ou plus	1,N	1..*
Plage spécifiée	N_1, N_2	$N_1..N_2$ ⁹²

Les classes-associations

Association + attributs



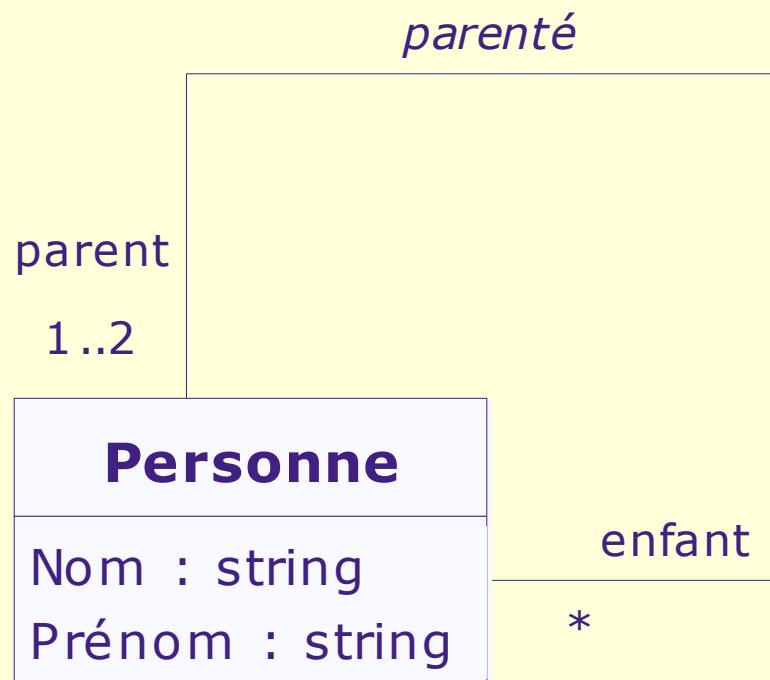
Association et rôles

Les extrémités d'une associations peuvent être nommée



Association réflexive

Association d'une classe avec elle même



Les classes au-delà des limites du modèle relationnel

- Ajout de méthodes :

Personne
Nom : string
Prénom : string
DateNaissance : Date
Age()

- Attribut multivalué :

Roue
Jante : undefined
Pneu : undefined
PressionActuelle()

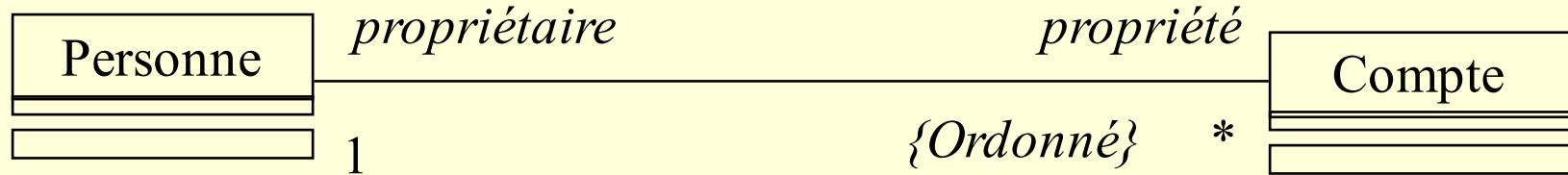
Camion
Numéro : integer
ChargeUtile : real
RouesCamion[4..10] : Roue

Visibilité des attributs et des méthodes

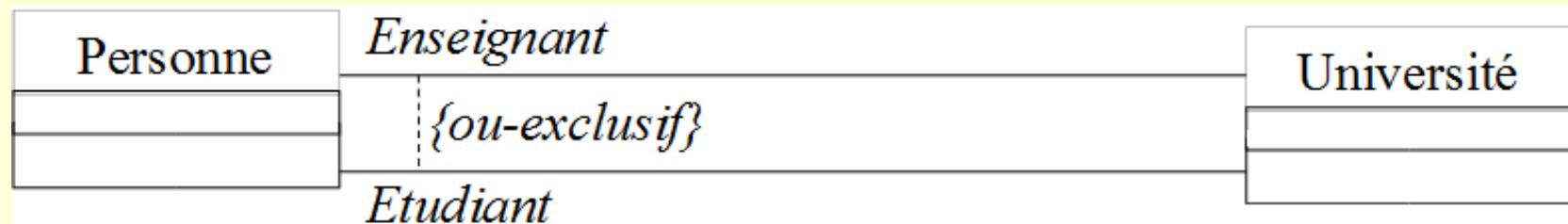
- Public : accessible pour tout utilisateur d'une classe, y compris la classe elle-même. « + »
- Protégé : accessible par la classe elle-même et par ses héritiers. « # »
- Privé : accessible seulement par la classe elle-même (tout accès autre se fait par le biais d'accesseur généralement nommés get et set). « - »

Contraintes sur les associations

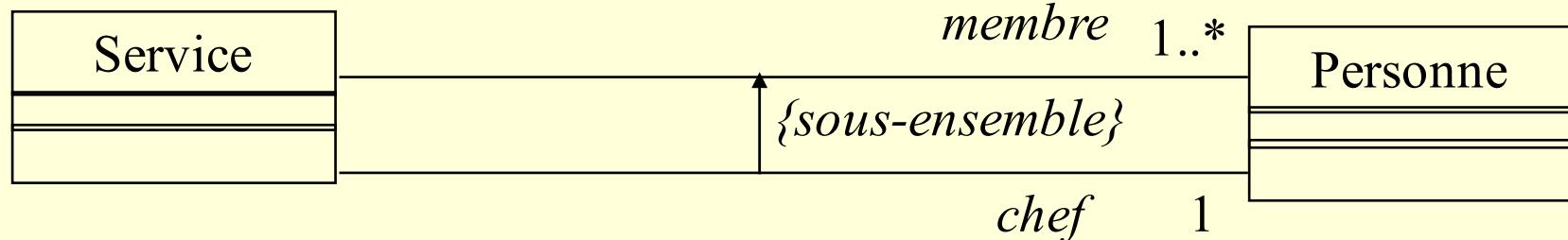
- {Ordonné}



- {Ou-exclusif}



- {Sous-ensemble}



L'agrégation

Une association de type agrégation traduit la volonté de renforcer la dépendance entre les classes.

L'agrégation correspondant à une classification, la granularité de l'agrégé étant plus importante que celle de l'agrégat.



L'agrégation se représente en ajoutant un petit losange du côté de l'agrégé

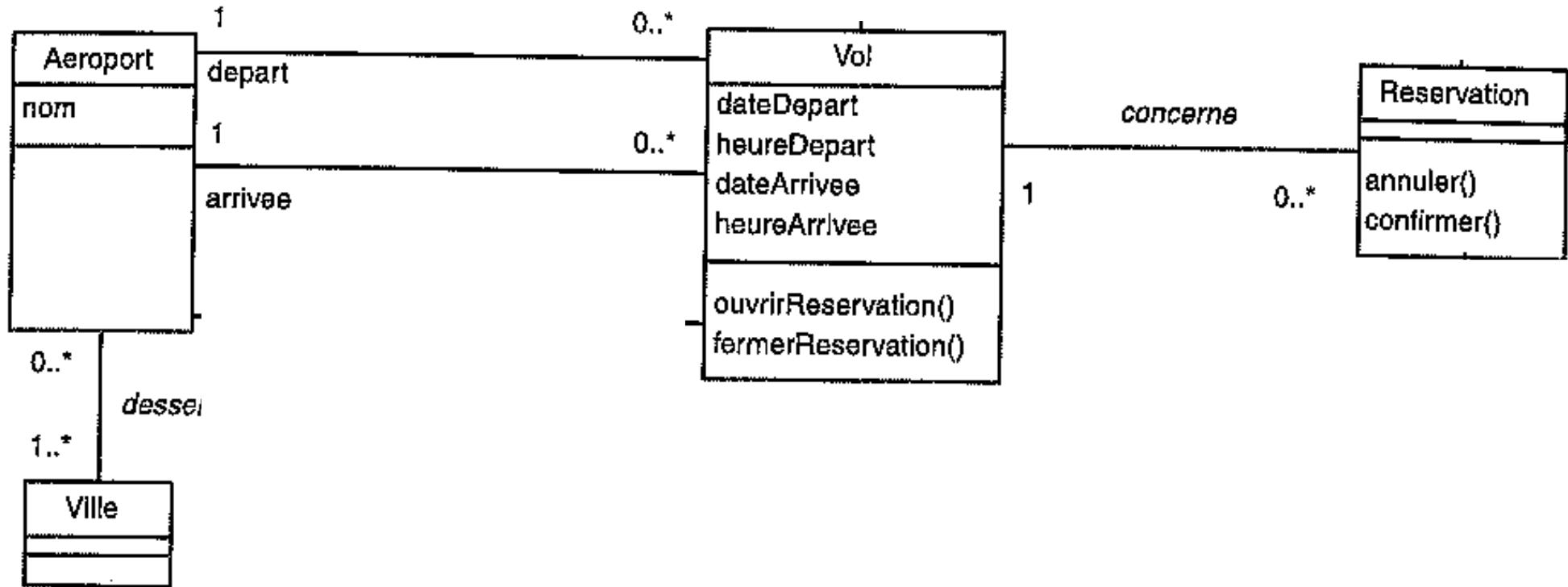
Composition

- La composition est un cas particulier d'agrégation.
Elle implique une coïncidence des durées de vie
du composant et du composite.



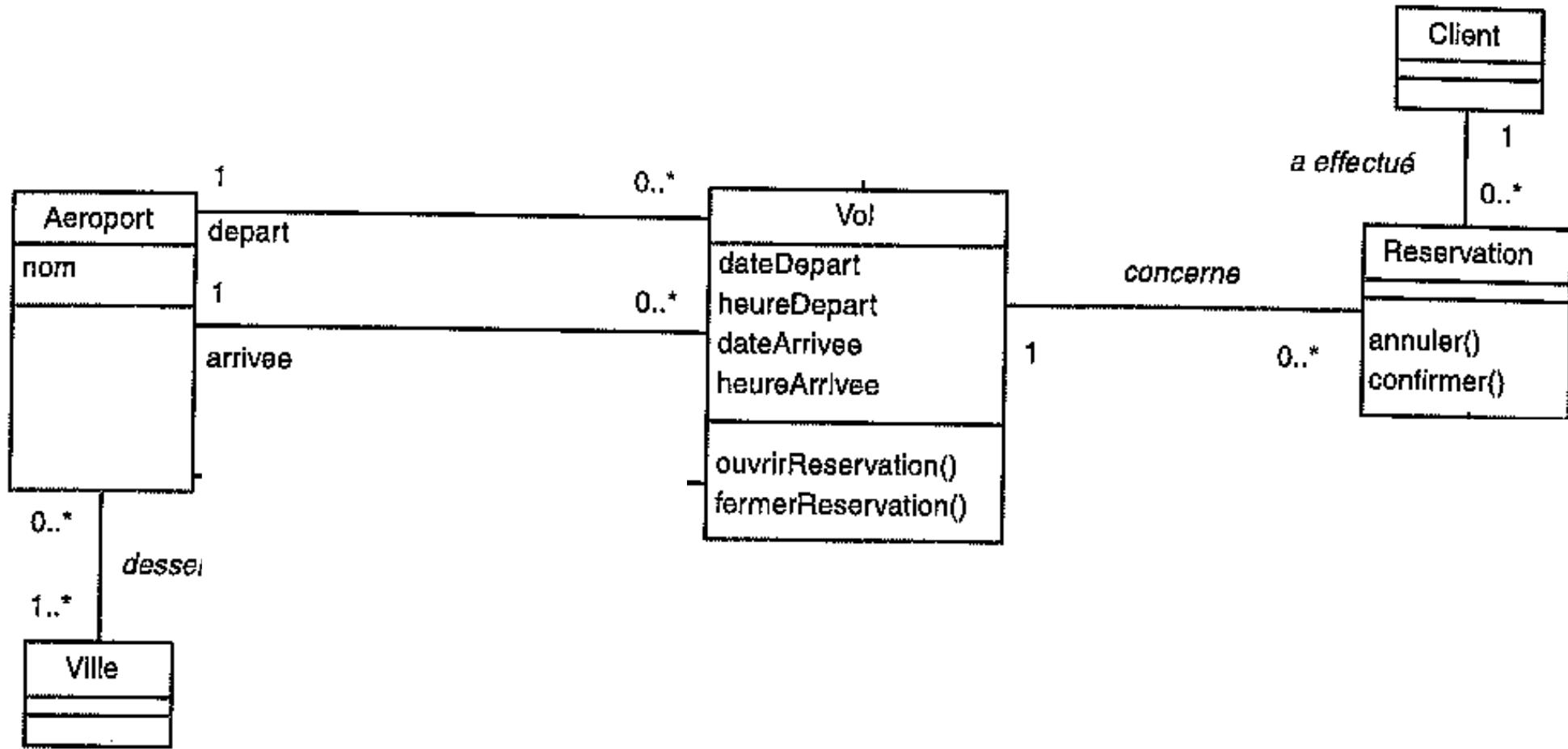
La destruction d'une instance de dossier entraîne la destruction des instances des document associées

Application



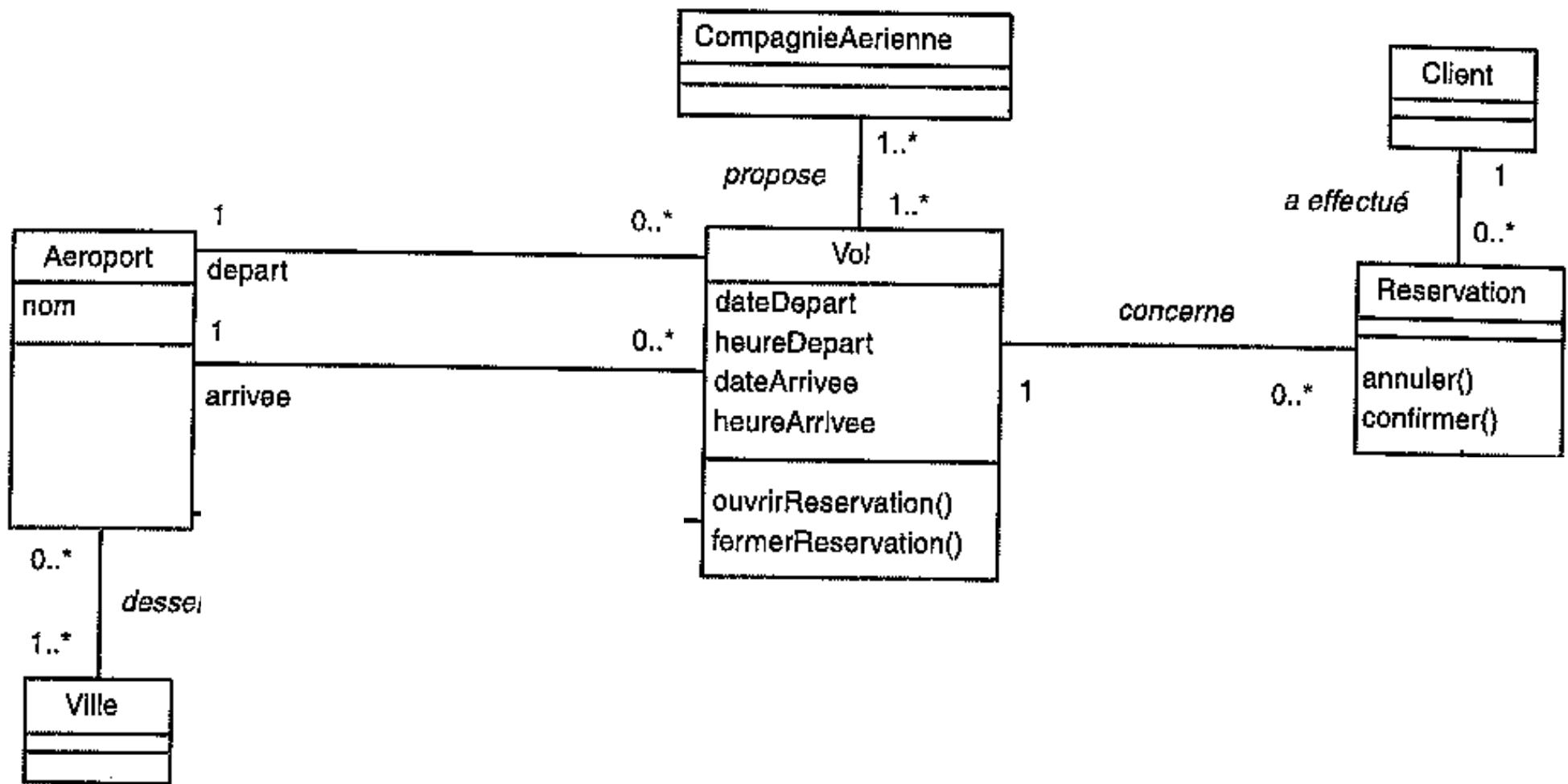
1 – Modéliser le fait qu'une réservation soit effectuée par un client.

Application



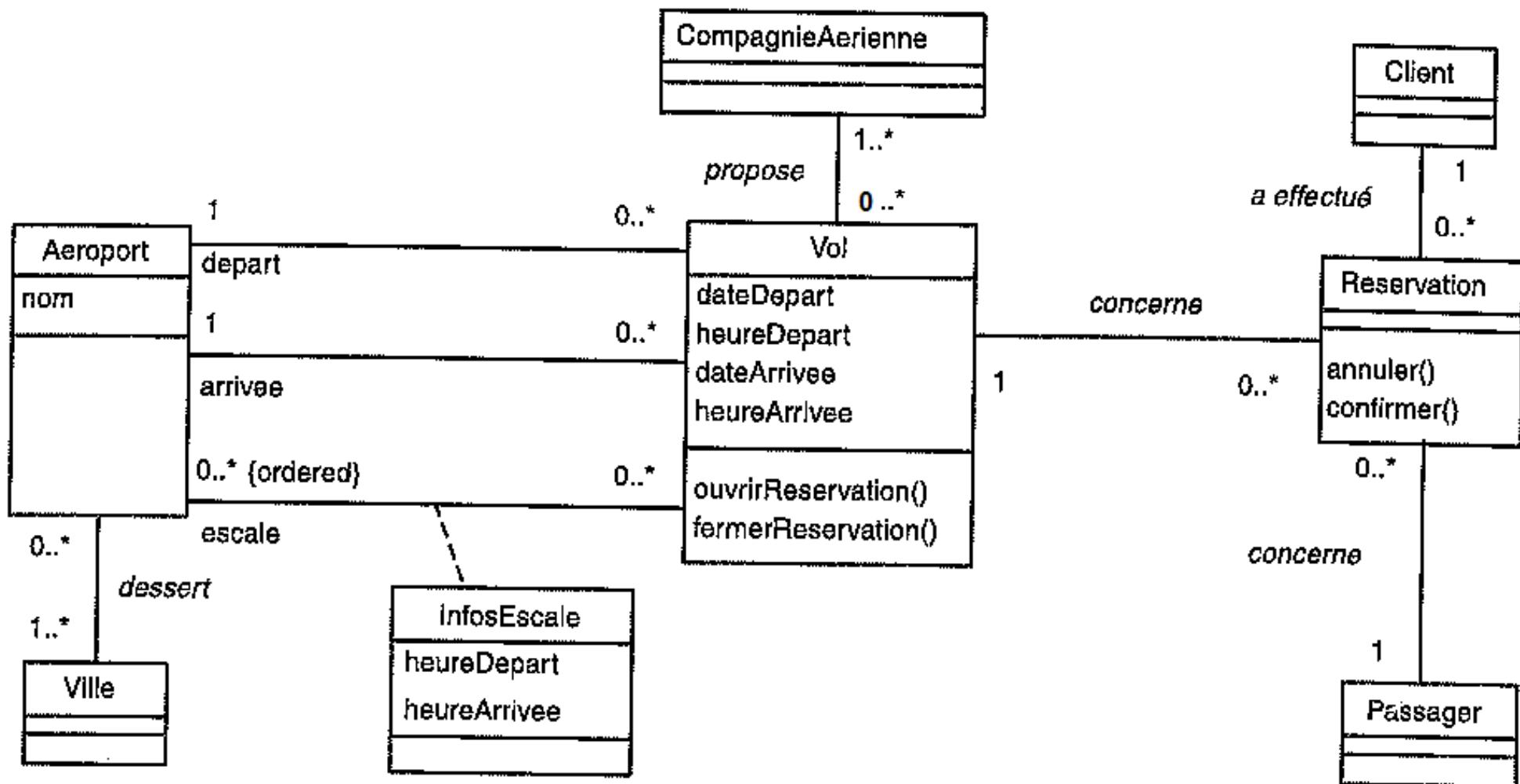
2 – Modéliser le fait qu'un vol soit proposé par une compagnie aérienne.
Dans le cadre des alliances comme Skyteam un vol peut être proposé par plusieurs compagnies.

Application



3 – Un vol peut faire des escales dans des aéroports (on doit connaître les heures d'arrivée et de départ lors de l'escale)

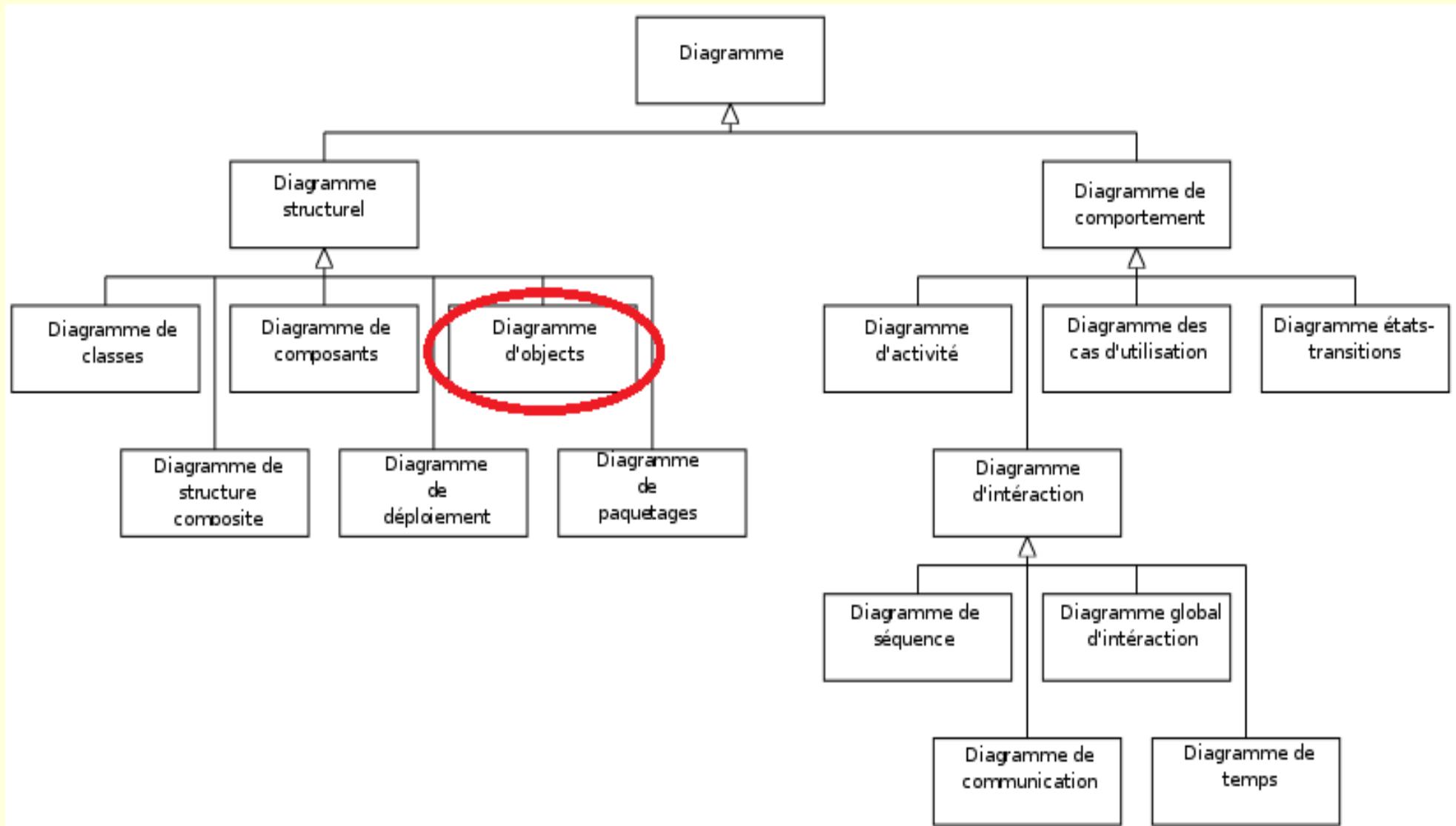
Application



Source : UML en action, Roques, Eyrolles

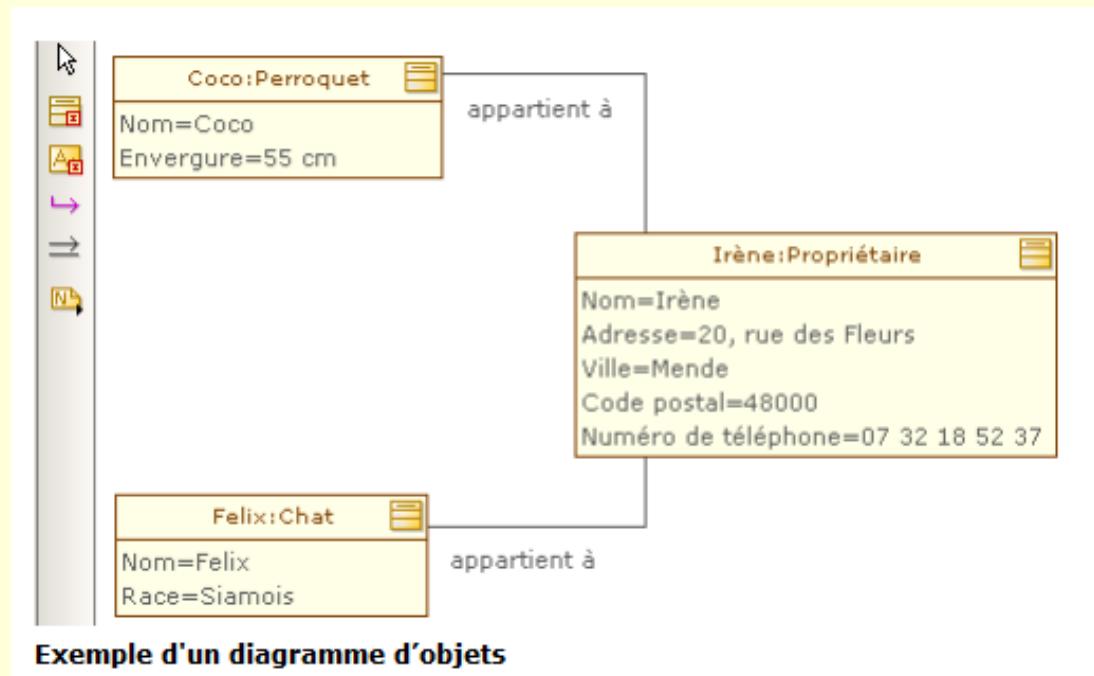
Autre possibilité : établir une classe Escale associée à un vol et à un aéroport.

Les diagrammes



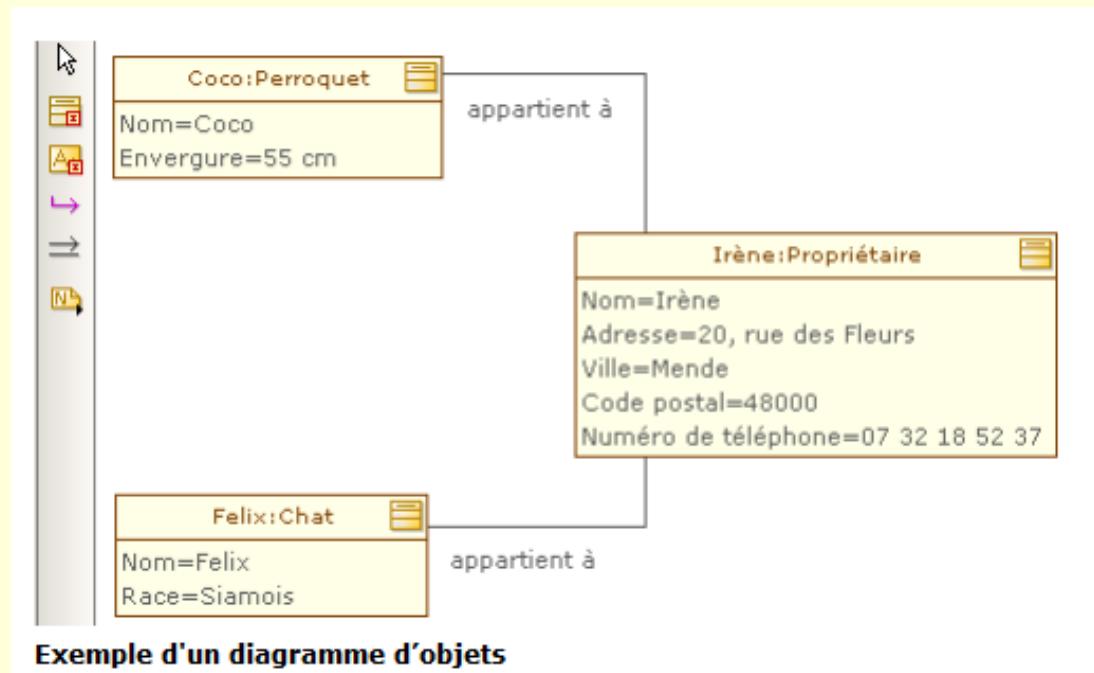
Le diagramme d'objet

- Exemples d'instances de classes
- Fournit des données pour les tests

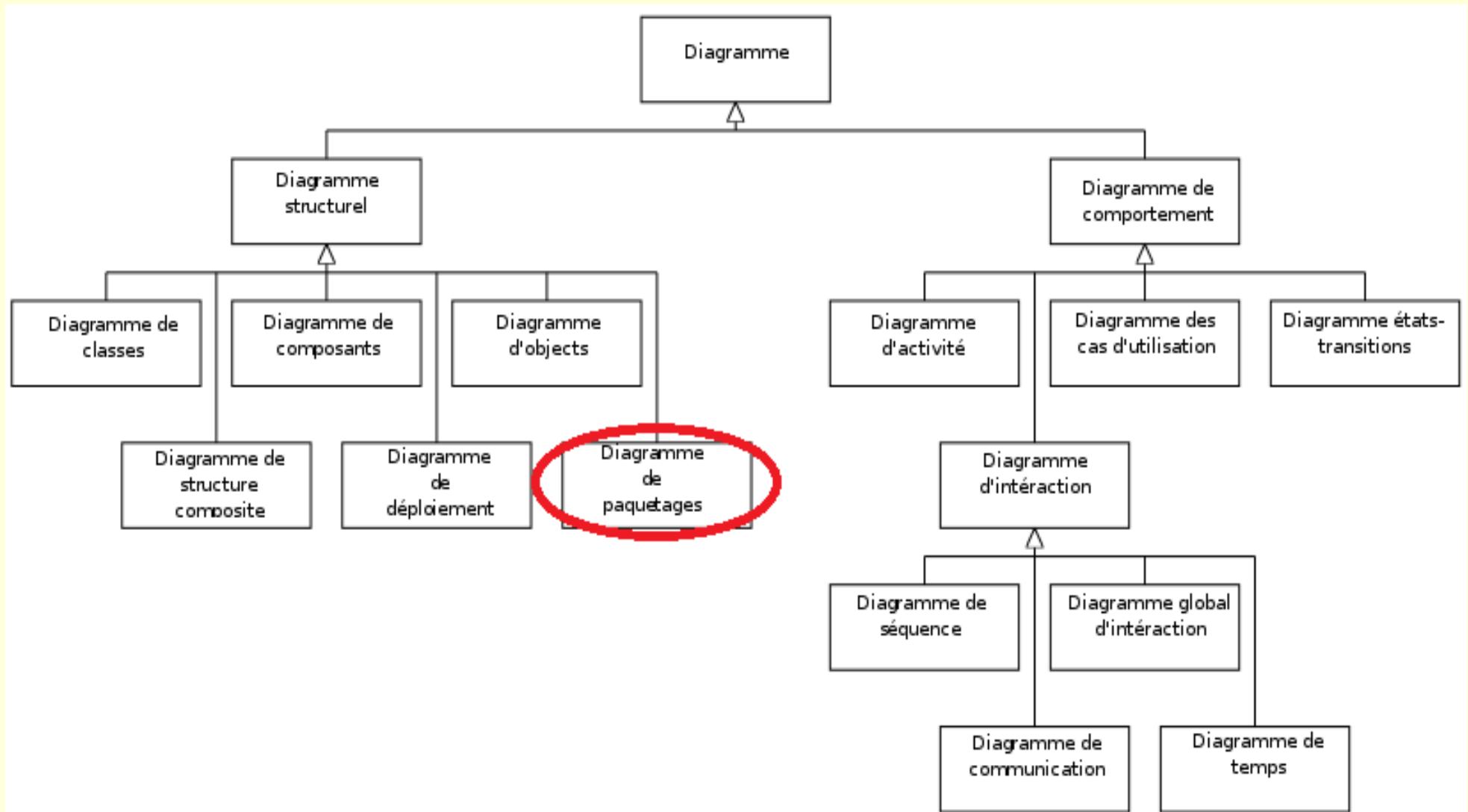


Application

Etablir le diagramme de classes associé à ce diagramme d'objets :



Les diagrammes



Paquetages (packages)

- Groupe d'éléments (classes, associations, objets, diagrammes) partageant un thème commun.
- Notation :

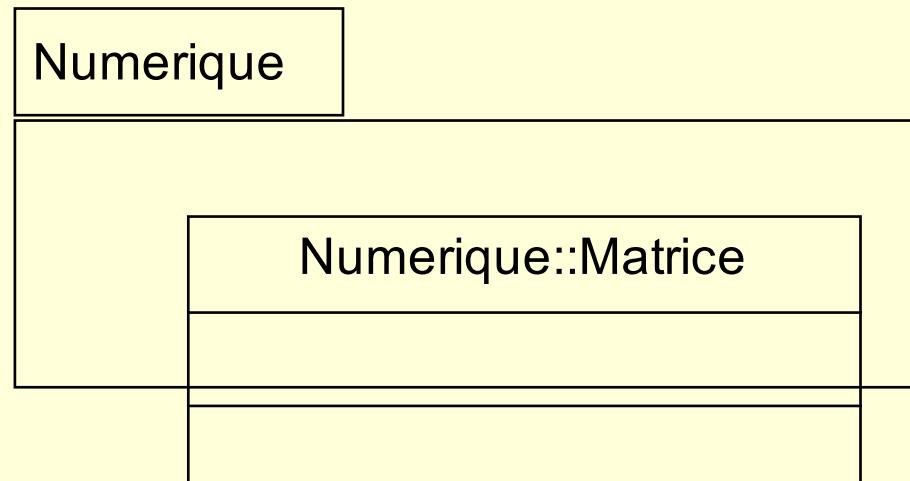


Diagramme de paquetage

Permet de définir des domaines et des dépendances entre ces domaines

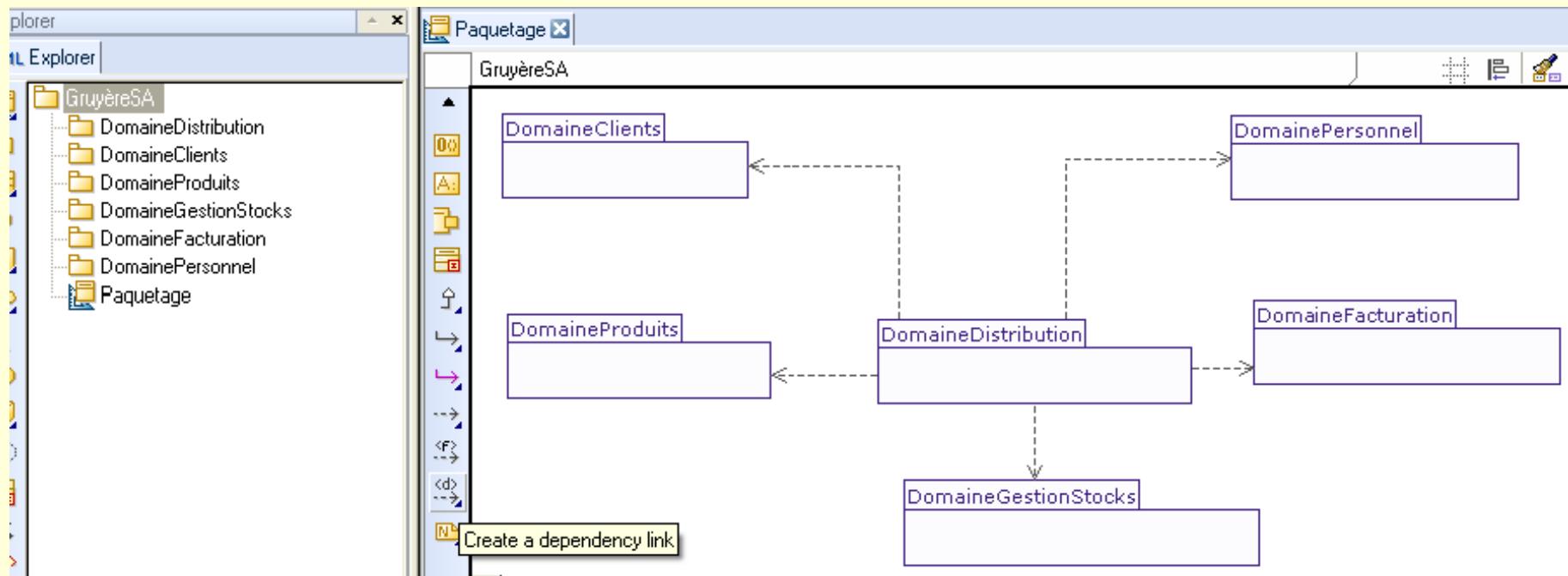
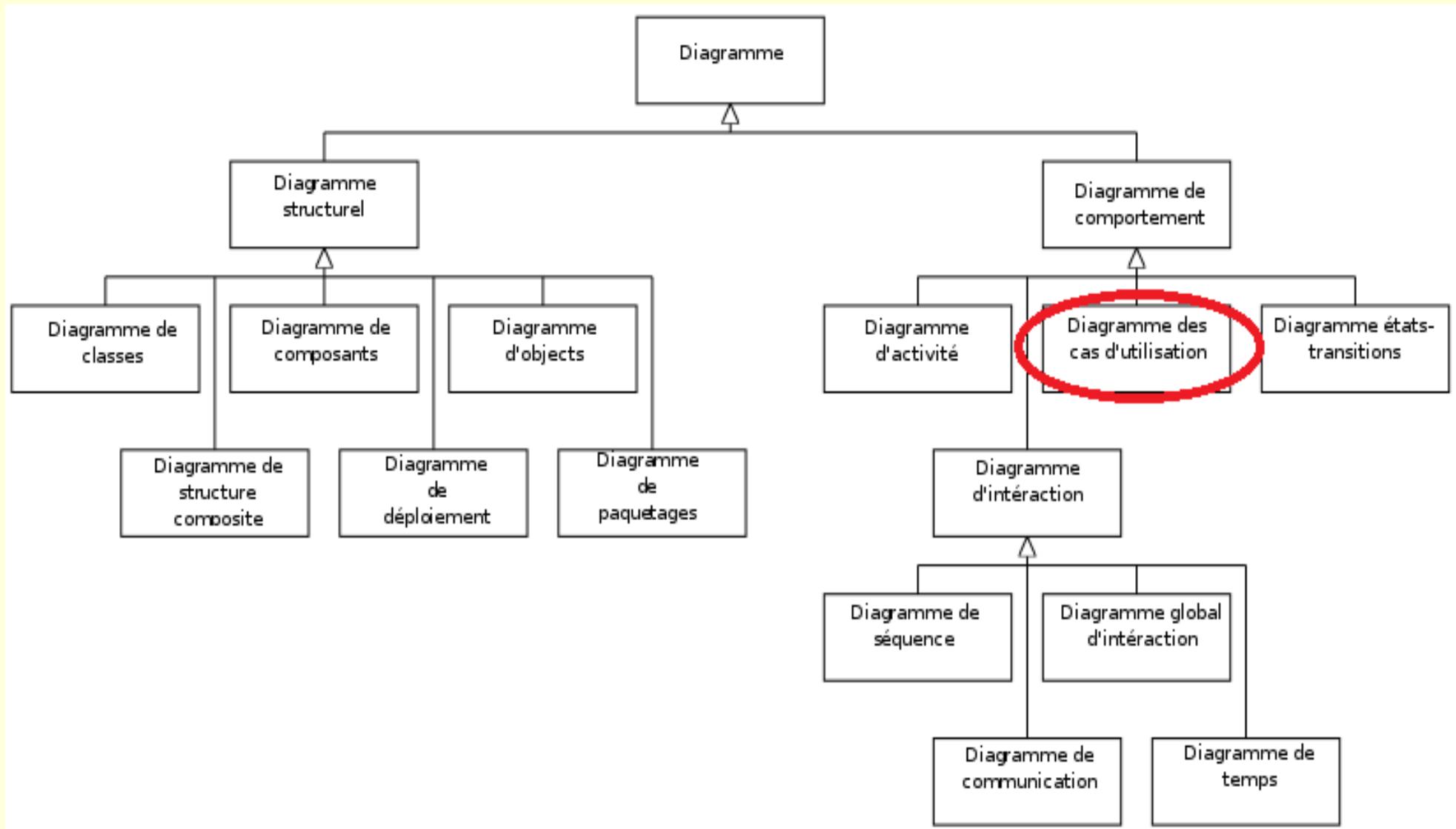


Diagramme de paquetage

- Dépendance ; 1 classe d'un paquetage (origine flèche) demande les service d'une classe d'un autre paquetage (destination)
- Il ne faut pas de dépendance circulaire
- Les paquetages dont dépendent beaucoup d'autres paquetages doivent peu évoluer

3. Modélisation des comportements

Les diagrammes



Les vues

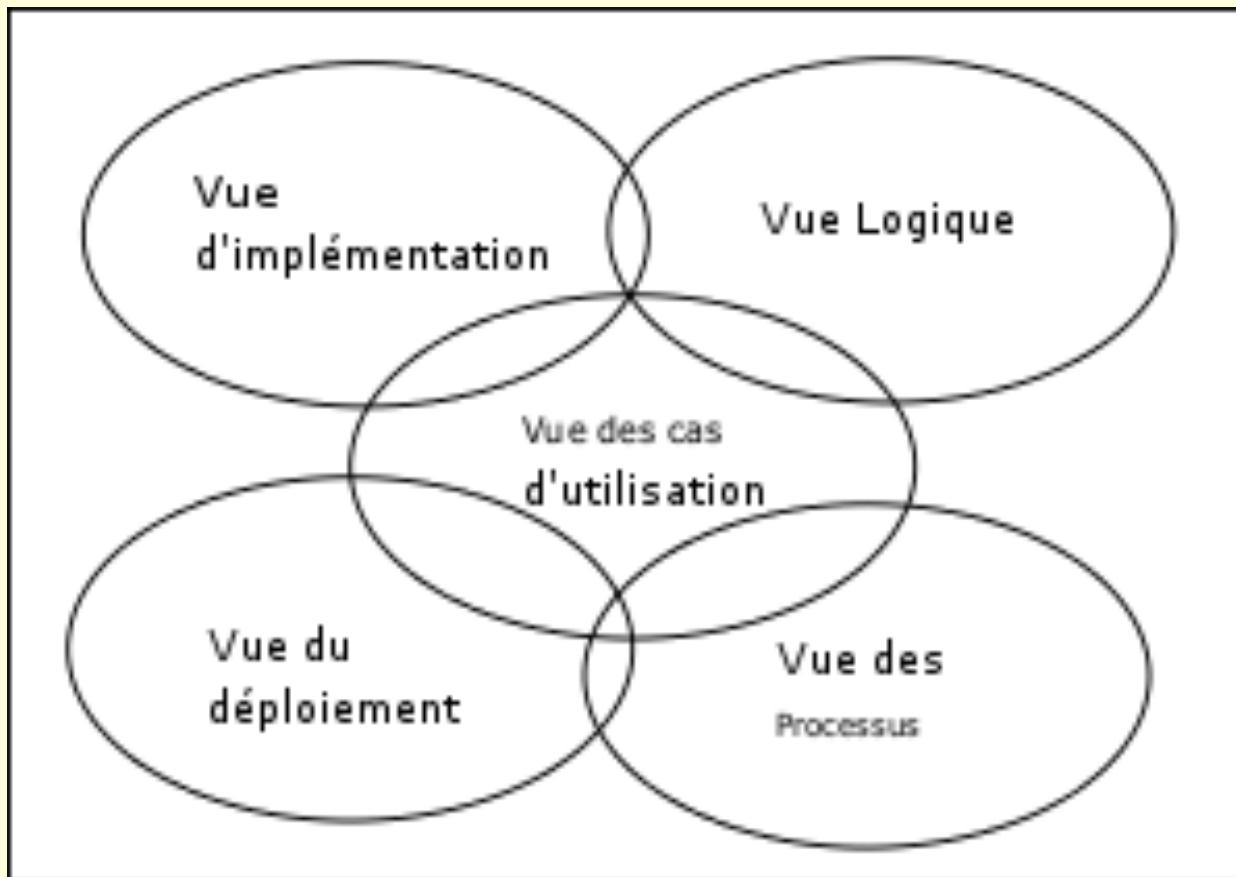


Diagramme de cas d'utilisation : aspect central d'UML.

Principes du diagramme de cas d'utilisation

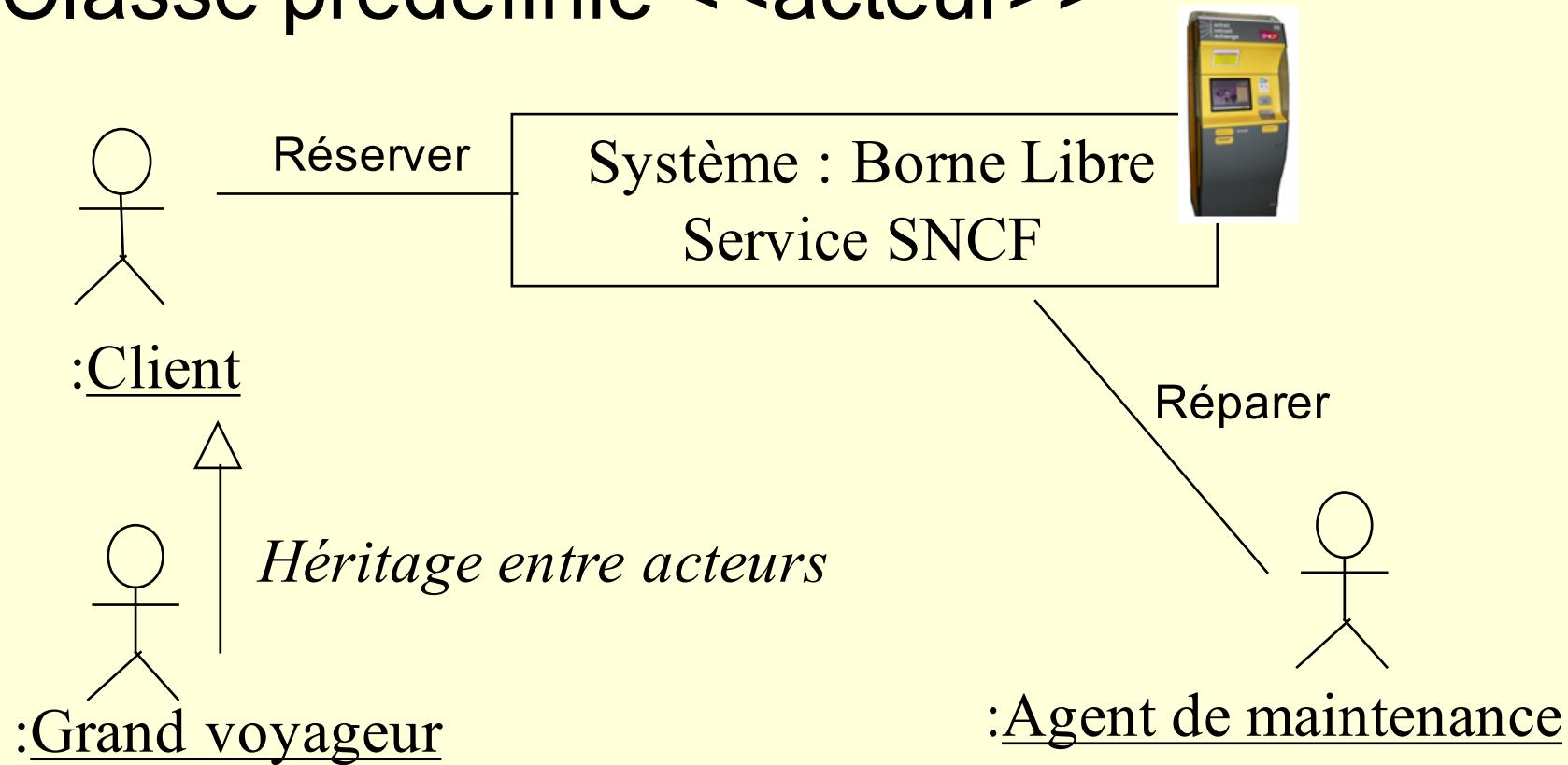
- Modèle servant à spécifier le comportement d'un système ou d'un sous-système.
- Permet à l'utilisateur de comprendre comment utiliser les fonctionnalités de l'élément.
- Montre plus précisément :
 - Limites du système
 - Acteurs extérieurs
 - Interactions entre acteurs et fonctions
 - Fonctions internes au système

Acteurs du système

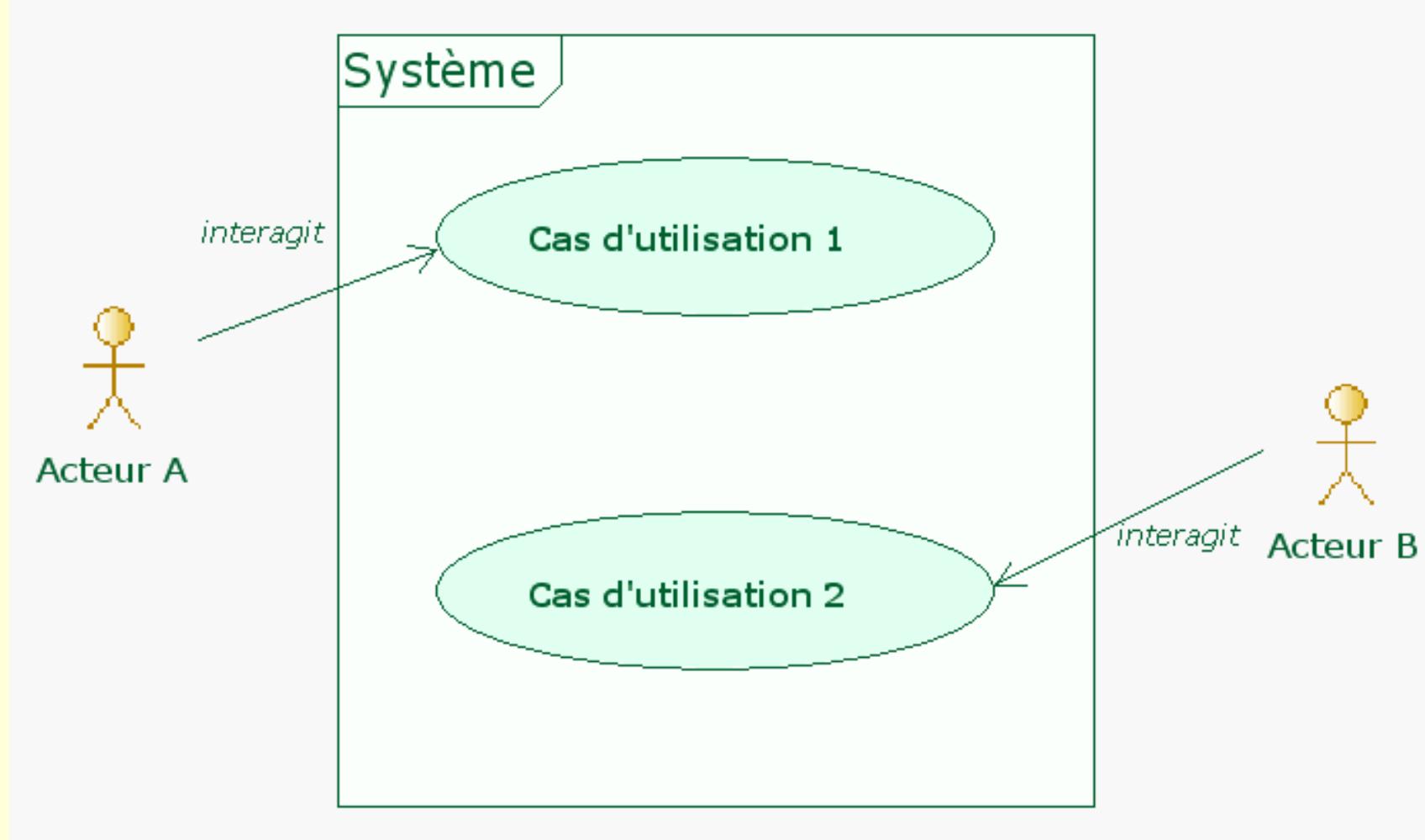
- Entités externes au sous-système.
- Interagissent avec le système par des évènements ou des messages.
- Sont définis par le rôle qui leur est attribué.
- Peuvent correspondre aussi bien à des personnes qu'à du matériel ou à un autre système.
- On modélise généralement les acteurs principaux à gauche et les secondaires à droite.

Acteurs : formalisme

Classe prédéfinie <<acteur>>



Les cas d'utilisation



Intérêt des cas d'utilisations

- Constituent un moyen de déterminer les besoins d'un système
- Utilisés par les utilisateurs finaux pour exprimer leurs **besoins**
 - élaboration des **spécifications** du système
- Permettent d'impliquer les utilisateurs dès les premiers stades du développement
- Constituent une base pour les tests fonctionnels

Scenario

- Format de représentation textuelle souple et riche, qui permet de préciser :
 - La description des interactions dans un cas typique de succès (le cas nominal)
 - Les variations du cas (cas particuliers)
 - Les contraintes diverses, questions ouvertes, etc.
- Outil de description des cas d'utilisation pour le dialogue utilisateur- informaticiens
- Outil pour la conception des scenarii de tests

Scenario

Cas d'utilisation : <ici le nom du cas d'utilisation>

Acteur : <ici le nom du ou des acteurs principaux, déclencheurs du cas>

Parties prenantes et intérêts : <listes des parties prenantes et leurs intérêts>

Niveau : <ici une valeur parmi (*Stratégique, Objectif utilisateur, Sous-fonctionnalité*)>

Portée : <ici la portée du cas> (voir plus loin)

[Pré-conditions] : <ici les préconditions éventuelles>

[Post conditions] : <ici les postconditions éventuelles>

Scénario nominal

1. <description de l'action>
2. <description de l'action>
3. ...
4. <description de la dernière action avant la fin du cas>

[Extensions]

- <numéro de l'étape> : <condition> : <action ou sous-cas d'utilisation>
- <numéro de l'étape> : <condition> : <action ou sous-cas d'utilisation>

[Contraintes]

...

[Question-ouverte]

...

[Annexe]

...

Exemple de scenario



Description

Objectifs :

Ce cas traite des actions conduites pour assurer des stocks satisfaisants.

Acteur :

Le gérant ou son adjoint

Actions :

1. Scénario principal
 - 1.1 déterminer les besoins
 - 1.2 choisir le fournisseur
 - 1.3 passer la commande
2. Scénario : pas de fournisseur
 - 2.1 référencement d'un fournisseur
 - 2.2 retour à 1.3

Règle de gestion :

- On favorise le fournisseur le plus proche

Document :

- Catalogue fournisseur

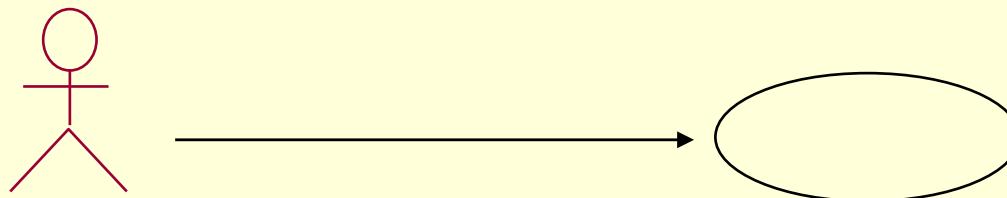
Scenario

- R1 : Partir du sommet (les grandes fonctions), et se maintenir le plus possible au niveau objectif utilisateur
- R2 : Centrer son attention sur le cas nominal (un scénario typique de succès)
- R3 : Préciser toujours les parties prenantes et leur intérêts
- R4 : Utiliser un verbe au présent de l'indicatif à chaque étape
- R5 : Utiliser la voie active décrivant les sous-objectifs en cours de satisfaction
- R6 : Le sujet doit être clairement localisable (en début de phrase généralement)
- R7 : Rester concis et pertinent (éviter les longs documents)
- R8 : Eviter les si, et placer les comportements alternatifs dans les extensions.
- R9 : Signaler les sous-cas d'utilisation. Ils sont toujours représentés par la relation d'inclusion include d'UML.
- ...
- R12 : Laisser de côté l'interface utilisateur

Diagramme de cas d'utilisation

Relations

- **Communication, association** : entre un acteur et un cas d'utilisation : Echange d'informations entre l'acteur et le système.



- **Inclusion** entre deux cas d'utilisation. Le cas d'utilisation source comprend également le comportement décrit par le cas d'utilisation destinataire.

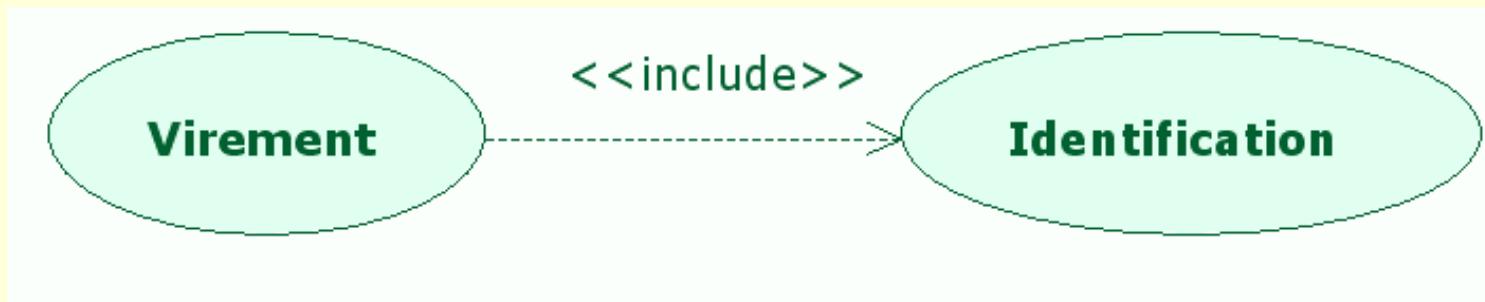


Diagramme de cas d'utilisation

Relations (suite)

- **Extension** entre deux cas d'utilisation.

Le cas d'utilisation source est une spécialisation du cas d'utilisation cible, il ne comprend que la description particulière.

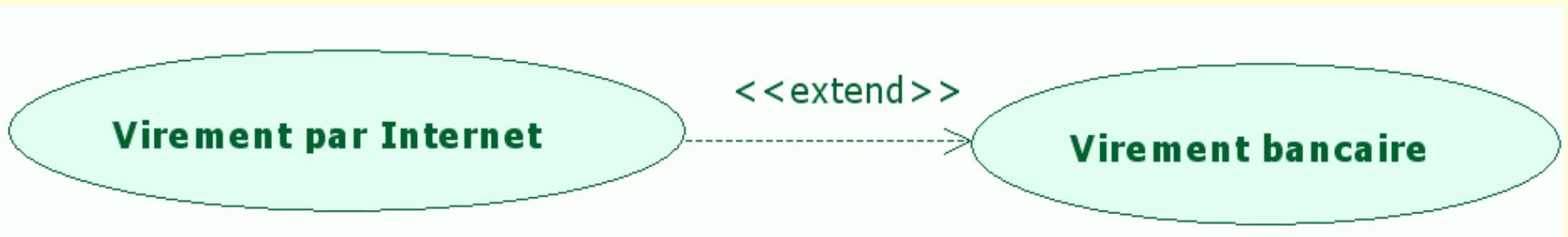
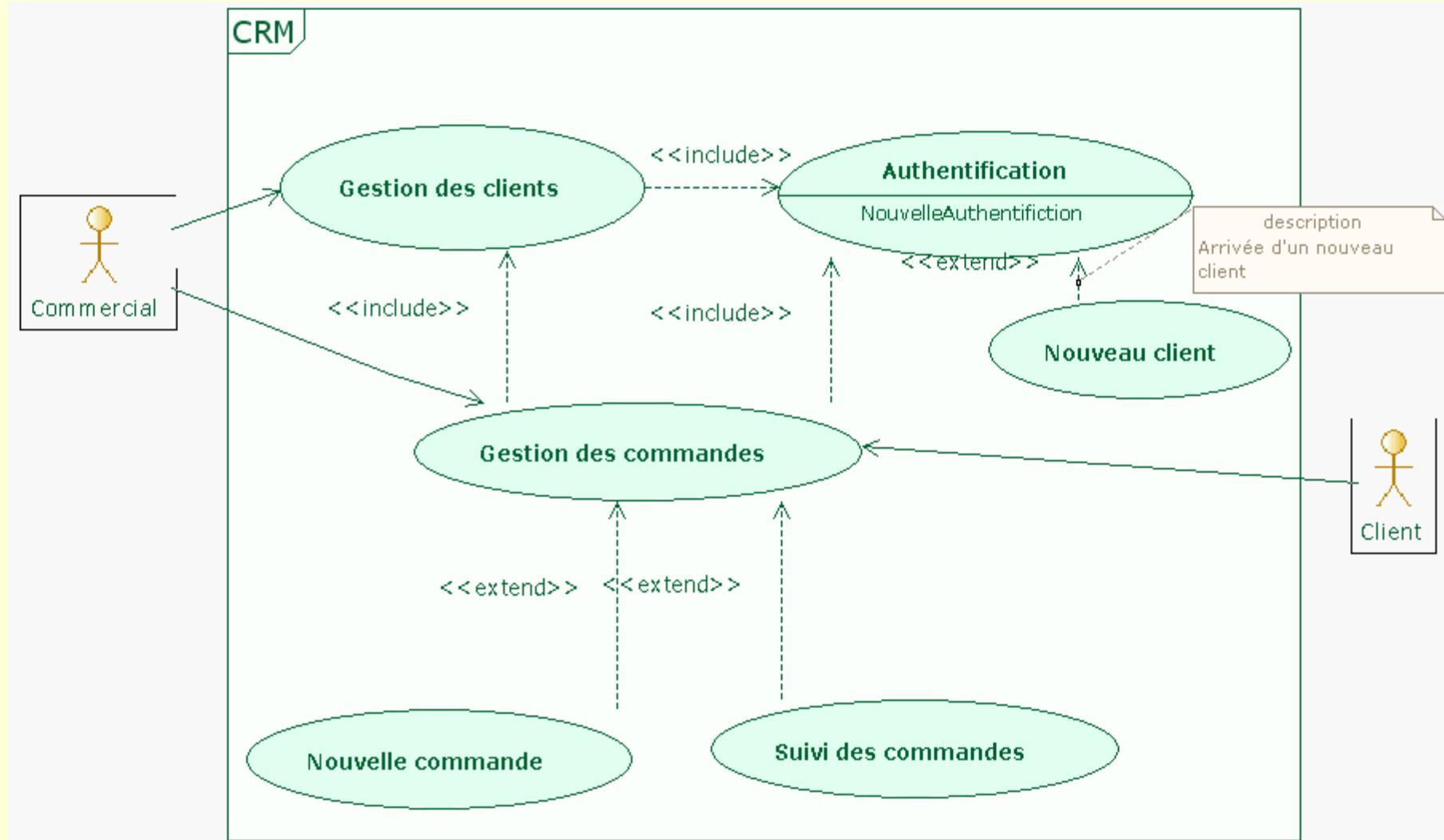


Diagramme de cas d'utilisation

Exemple



Application : guichet automatique bancaire

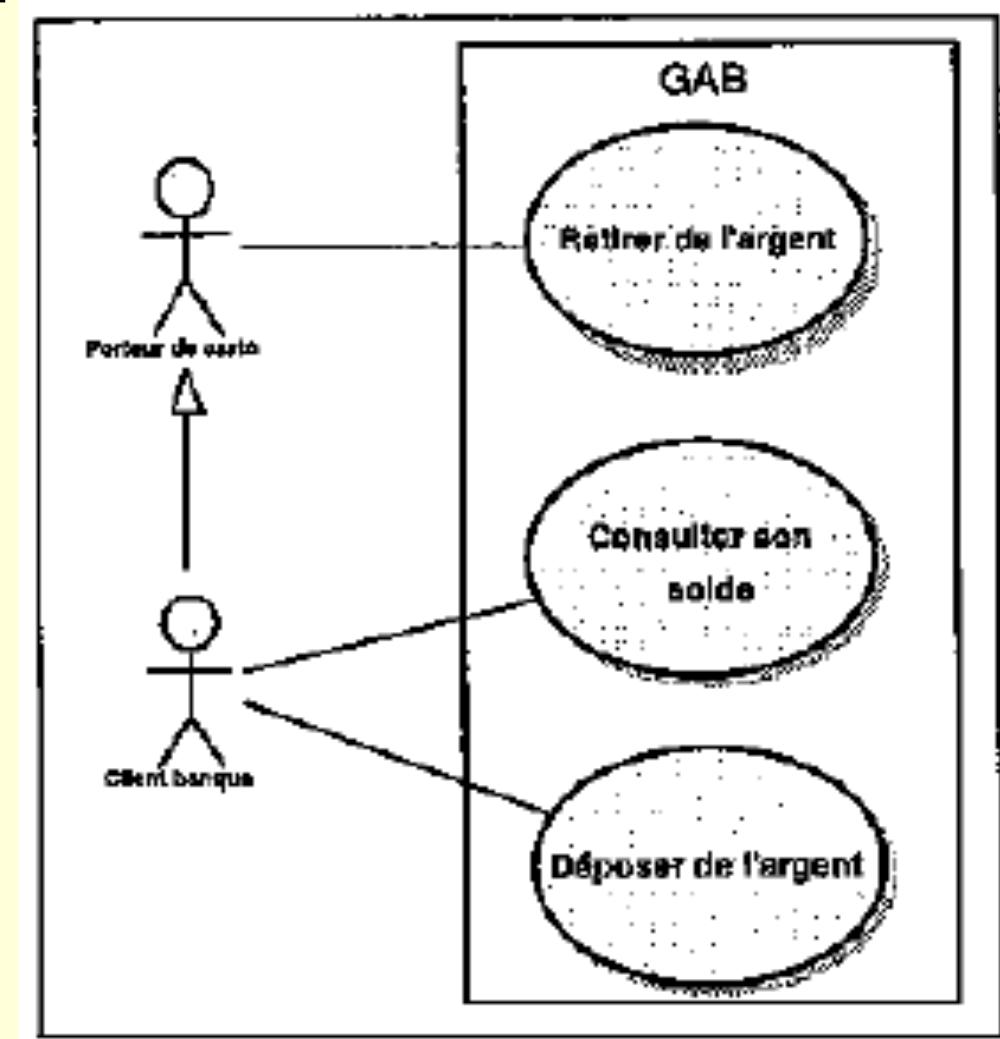
Un porteur de carte bancaire peut utiliser le guichet pour retirer de l'argent.

Si c'est un client de la banque, il peut également consulter son solde ou déposer de l'argent

Application : guichet automatique bancaire

Un porteur de carte bancaire peut utiliser le guichet pour retirer de l'argent.

Si c'est un client de la banque, il peut également consulter son solde ou déposer de l'argent



Les diagrammes

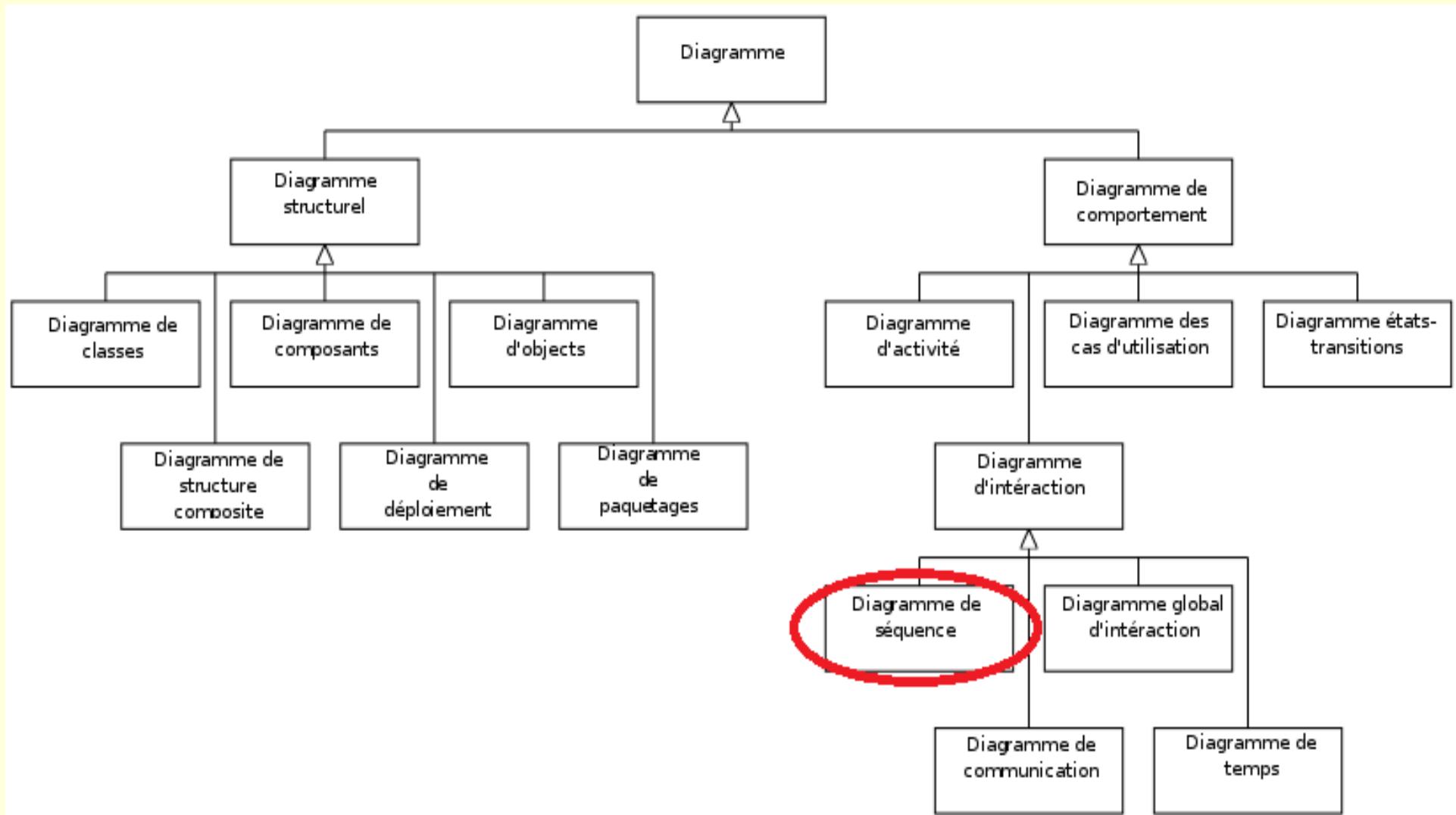


Diagramme de Séquences

Exprime la séquence des interactions entre objets du système selon un point de vue temporel, pour réaliser le cas d'utilisation.

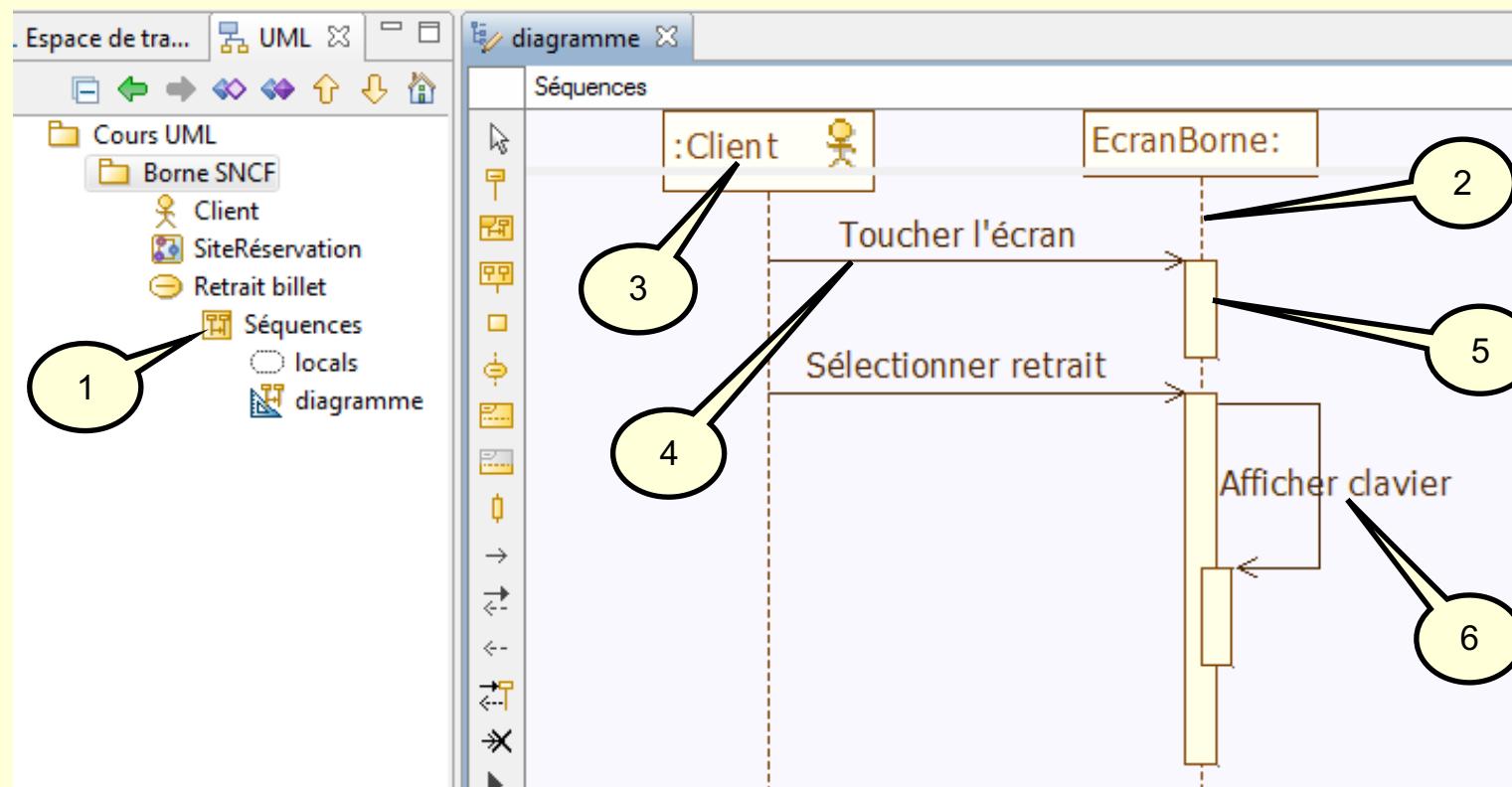


Diagramme de Séquence

Représente les objets impliqués dans le processus.

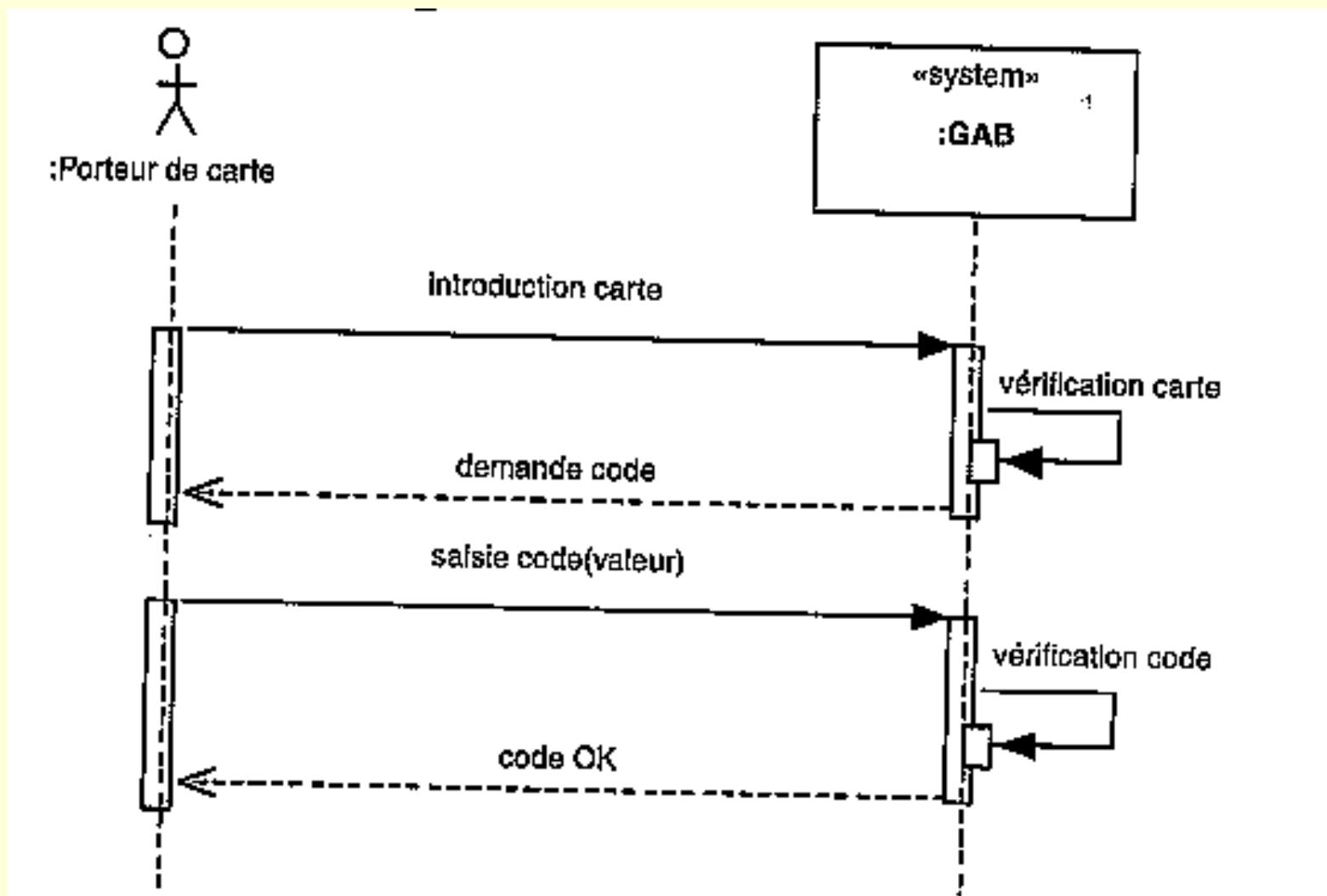
Un objet peut-être une instance de classe ou un acteur.

Ligne de vie d'un objet (verticale) : le temps s'écoule de haut en bas. Permet de voir les messages envoyés ou reçus par l'objet (flèches) et l'activité d'un objet (surépaisseur).

Un message reçu porte généralement le nom d'une opération de la classe.

Les messages peuvent être asynchrones ou synchrones.

Diagramme de séquence : exemple



Application

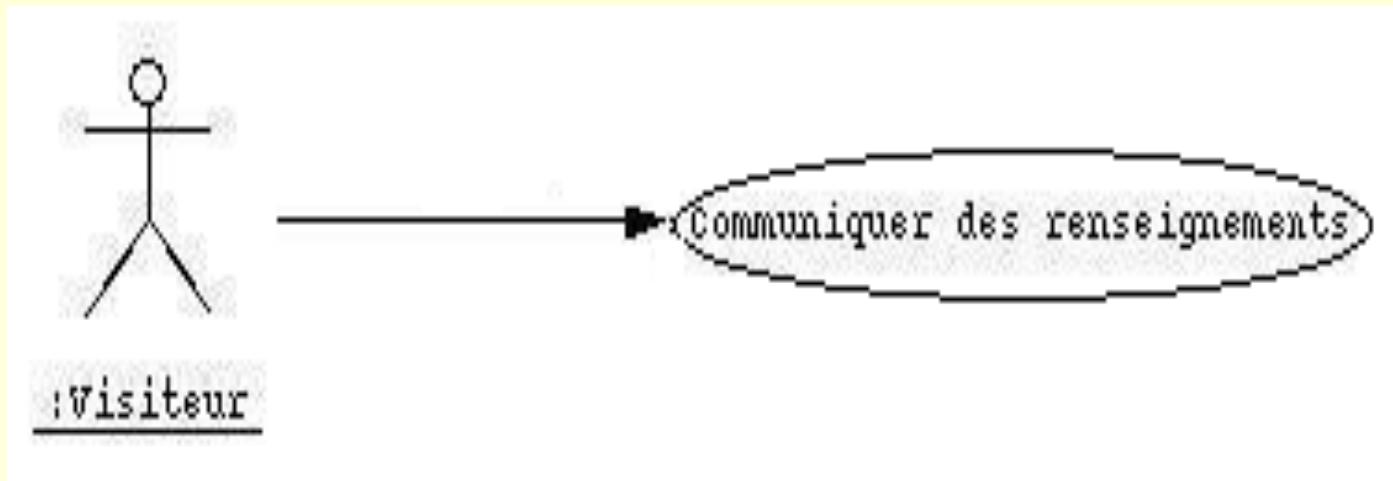
Etablir les diagrammes de cas d'utilisation et de séquence correspondant à ce cas :

Un établissement public d'enseignement technique met en place un système de porte ouverte permanent sur la toile. Les visiteurs sont invités à communiquer leurs coordonnées et divers renseignements.

- **Cas d'utilisation** : Communiquer des renseignements
- **Acteur** : visiteur
- **Parties prenantes et intérêts** :
 - Visiteur : il veut communiquer ses coordonnées à l'établissement afin d'être contacté ultérieurement sur les formations qu'il aura sélectionnées.
 - Service Administratif : il veut pouvoir contacter, en temps voulu, les personnes ayant manifesté un intérêt pour les formations disponibles dans l'établissement.
- **Pré-conditions** : aucune
- **Post-conditions** : Les données communiquées par le visiteur sont accessibles par le service administratif.
- **Scénario nominal**
 1. Le visiteur communique au système un choix de formation.
 2. Le système communique un formulaire d'identification.
 3. Le visiteur s'identifie.
 4. Le système présente un formulaire correspondant à la formation choisie.
 5. Le visiteur renseigne le formulaire et le soumet au système.
 6. Le système lui confirme l'enregistrement des informations.
- **Extensions**
 - *. A tout moment : le visiteur peut abandonner l'opération en cours.
 - 1a. Le visiteur est déjà authentifié (suivi de session) : aller en 4
 - 2a. Le visiteur n'est pas connu du système : sous-cas d'utilisation Création visiteur
 - 5a. Certains champs obligatoires se sont pas renseignés : aller en 4.

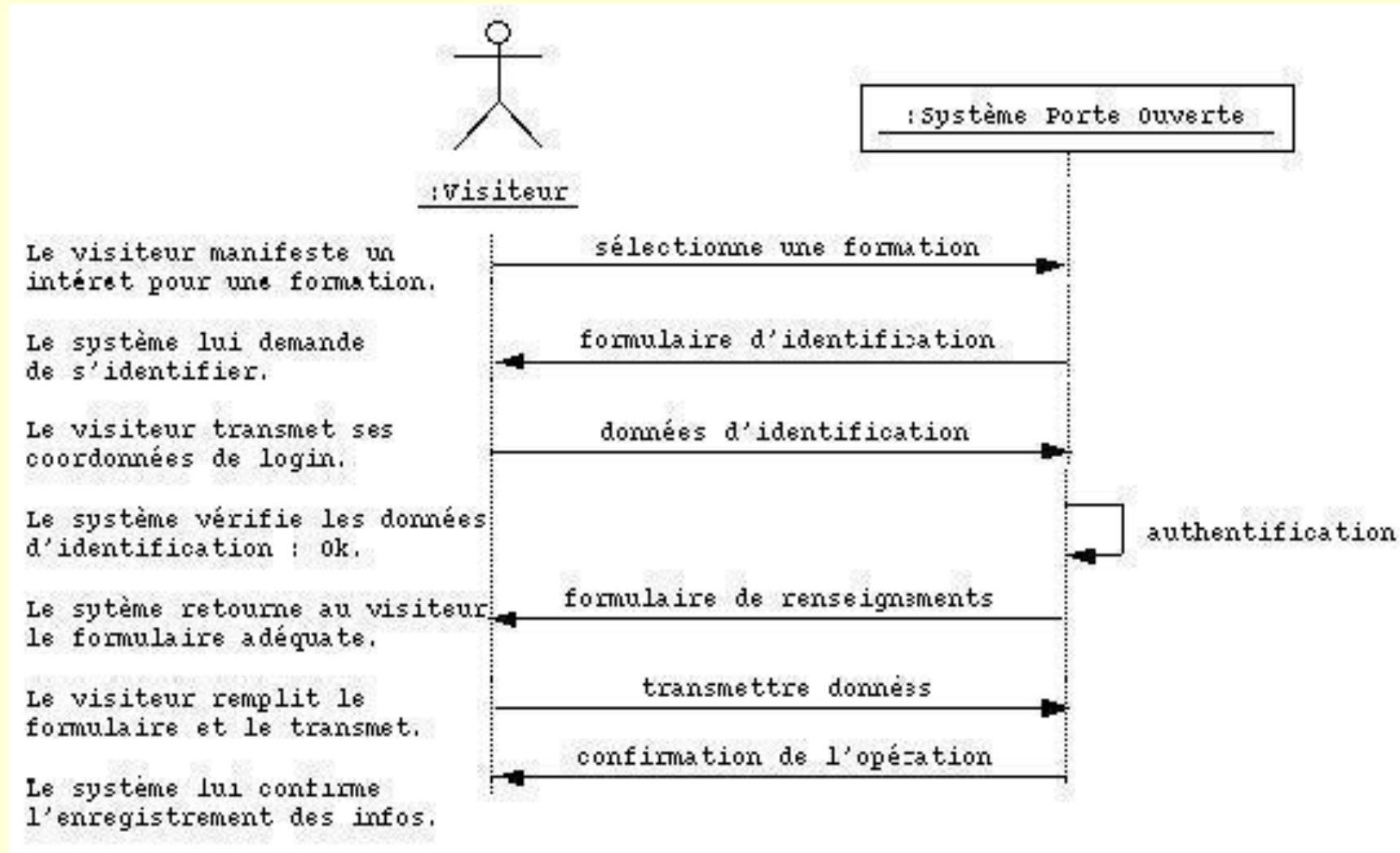
Application

Diagramme de cas d'utilisation :



Application

Diagramme de séquence:



Les diagrammes

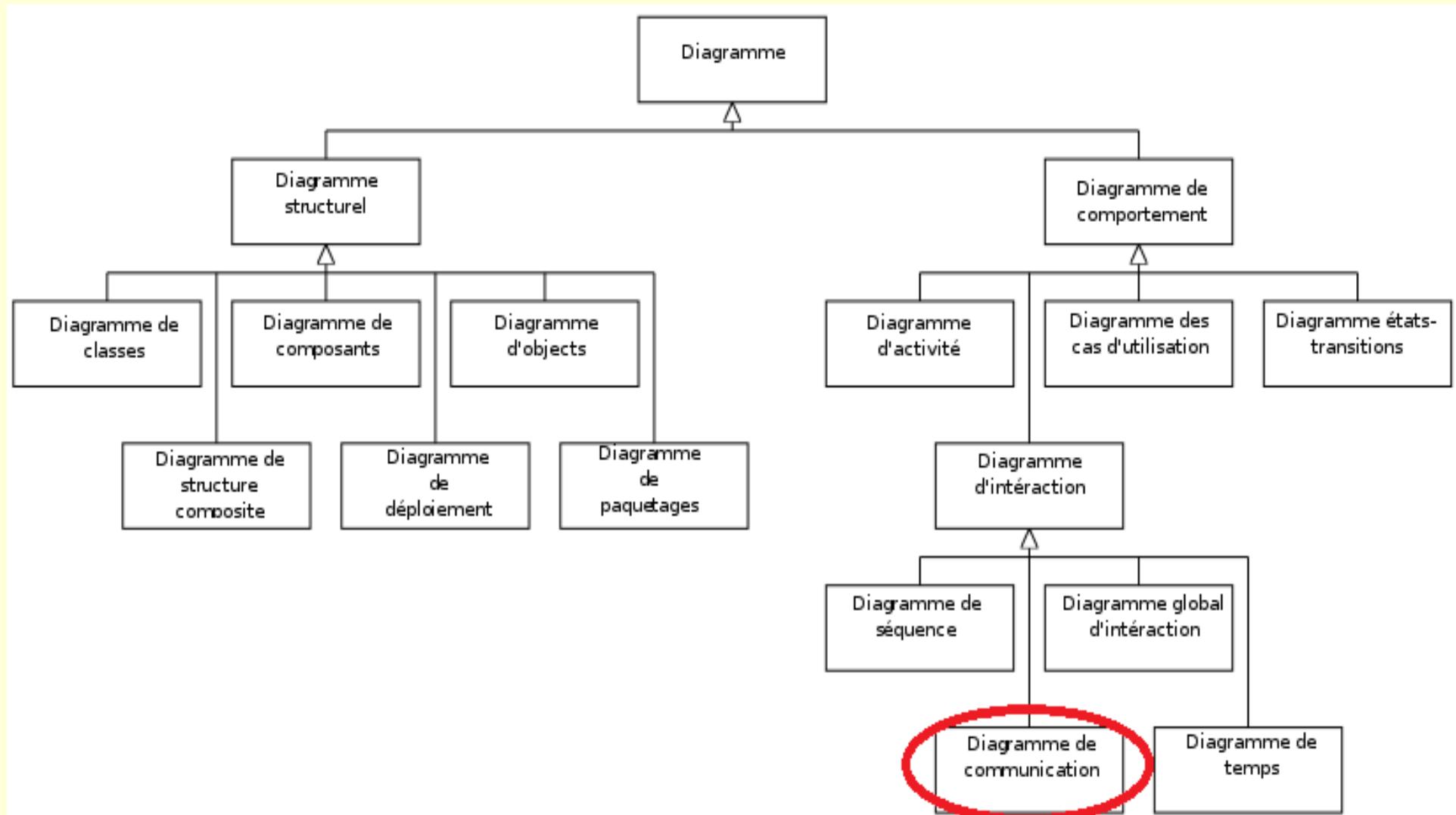
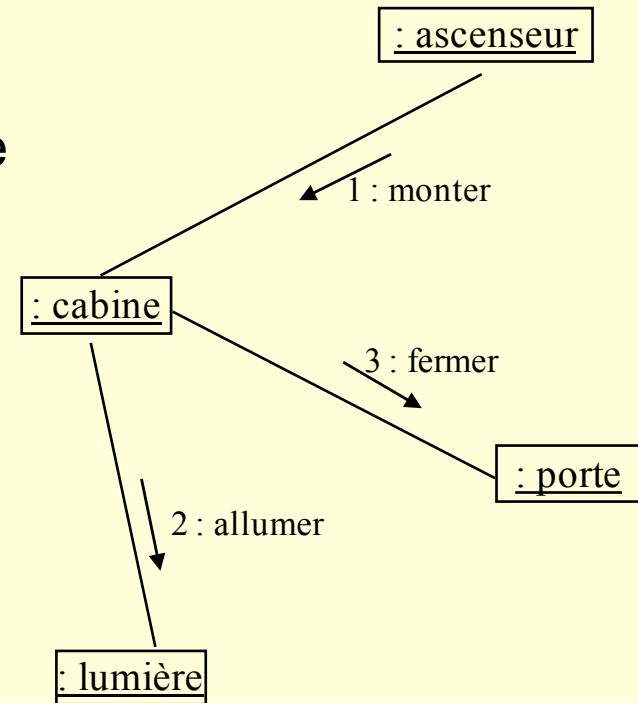


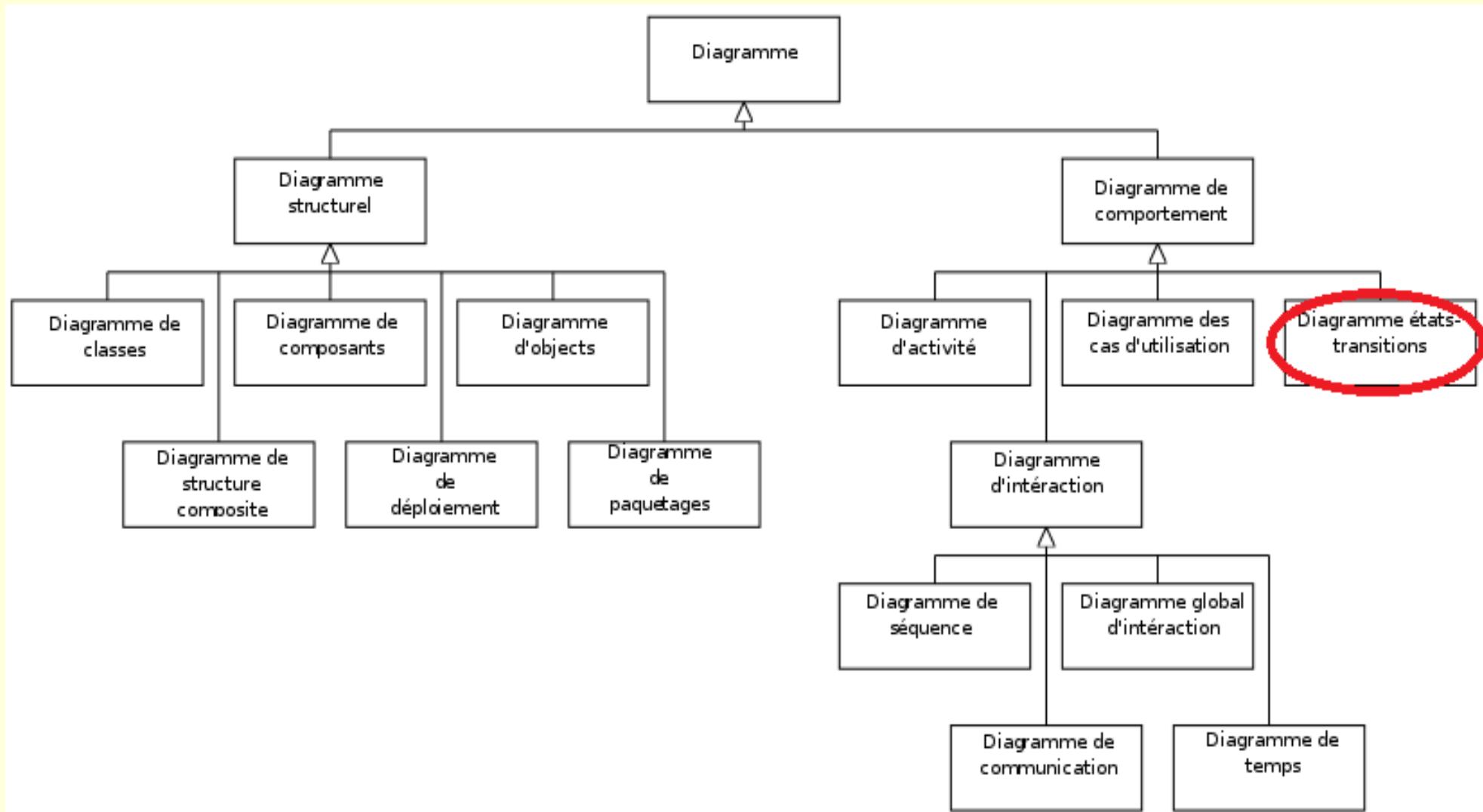
Diagramme de communication

Décrit, par une représentation spatiale statique,
les échanges de messages entre les objets

On approche du diagramme de séquence
sans la dimension temporelle.



Les diagrammes



Modélisation du comportement des objets

- Comportement d'un objet en réponse à des stimuli, des messages externes
 - Etats : passif, actif
 - Transitions : passages d'un état à un autre
 - Evènements : faits marquants, occurrences dans le temps
 - Action, activité : traitement effectué par l'objet
- Automate fini, machine à états finis

Diagramme états-transitions

- Représentations du cycle de vie auquel doivent se conformer toutes les instances d'une classe donnée

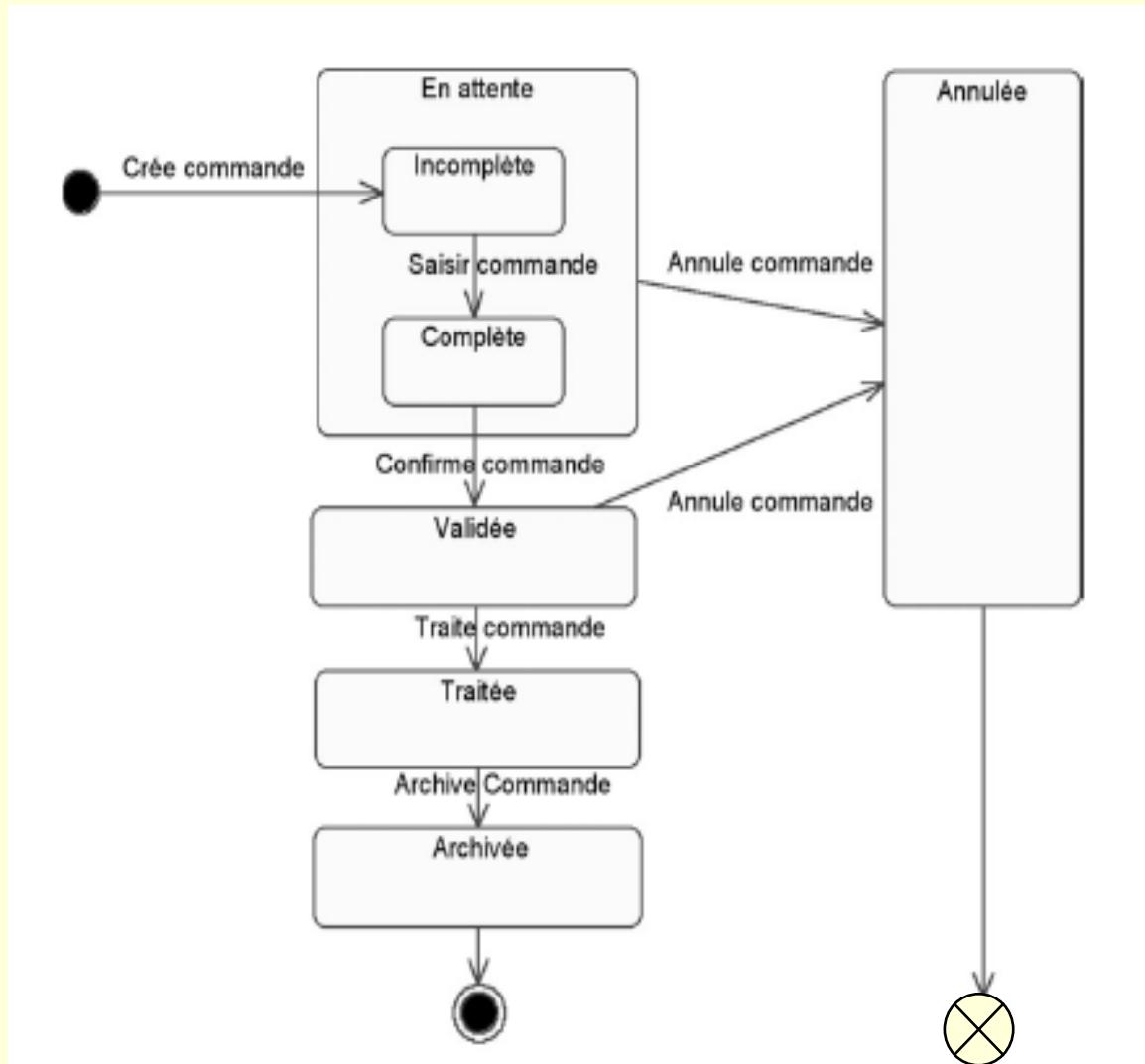


Diagramme d'Etats : Définitions

- Un état décrit une période durant la vie d'un objet, caractérisée par
 - un ensemble de valeurs d'attributs portées par l'objet
 - l'attente par l'objet d'un événement particulier (ou plusieurs)
 - ou la réalisation par l'objet d'une activité particulière
- Une transition décrit la réaction d'un objet lorsqu'un événement se produit (généralement l'objet change d'état)

Diagramme d'Etats : Définitions

- Un état peut porter une activité
 - complexe, et décomposable en activités ou en actions
 - qui peut être interrompue par un événement
- Un état ou une transition peut porter une action
 - simple, non décomposable
 - atomique, qui ne peut être interrompue
- Les paramètres des événements matérialisent le flot d'information
- Garde : condition booléenne qui valide ou non le déclenchement d'une transition lors de l'occurrence d'un événement

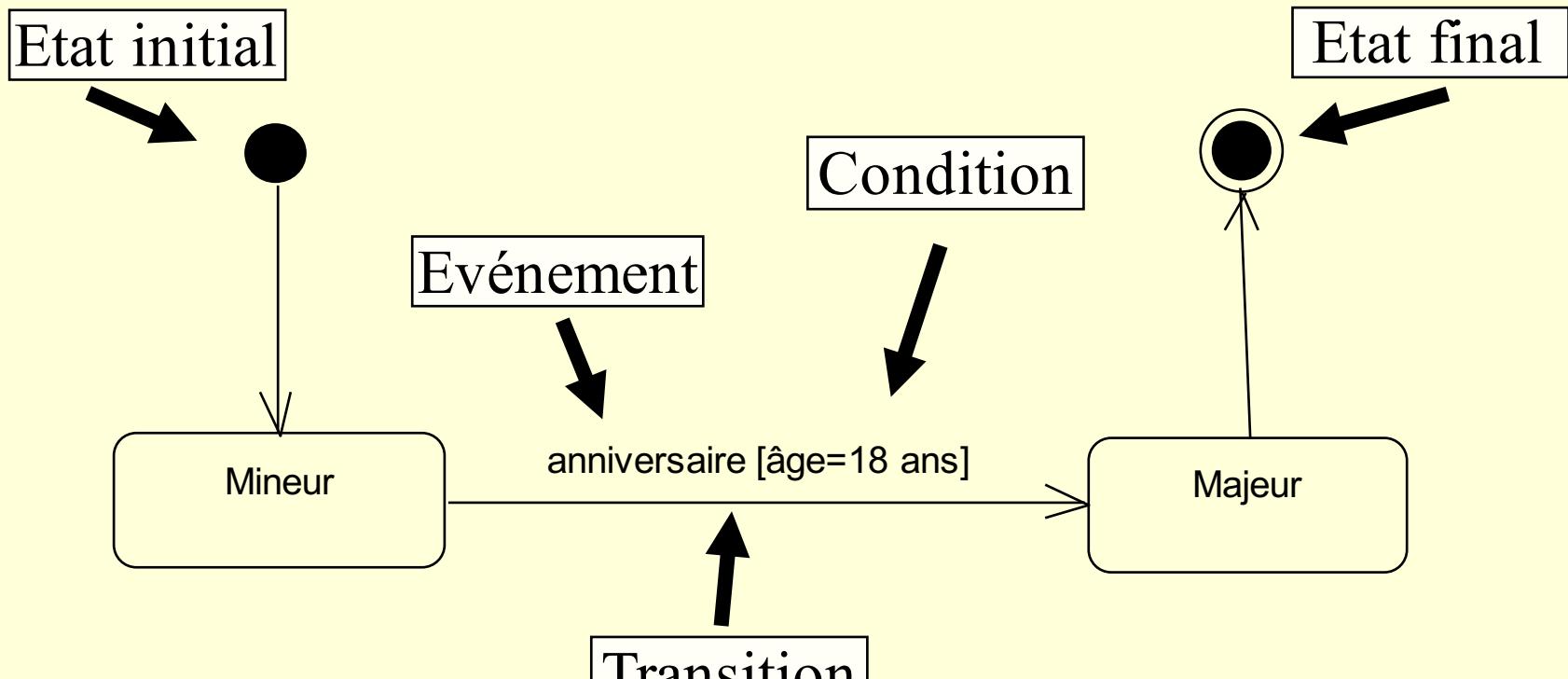
Etat

- Chaque objet est à un moment donné dans un état particulier :
 - Etat Initial : état d'une instance juste après sa création (un seul état initial)
 - Etat Intermédiaire : un objet est toujours dans un état donné pour un certain temps
 - Etat Final : état d'une instance à la fin du diagramme
 - Sortie, entrée : utilisé dans les sous états



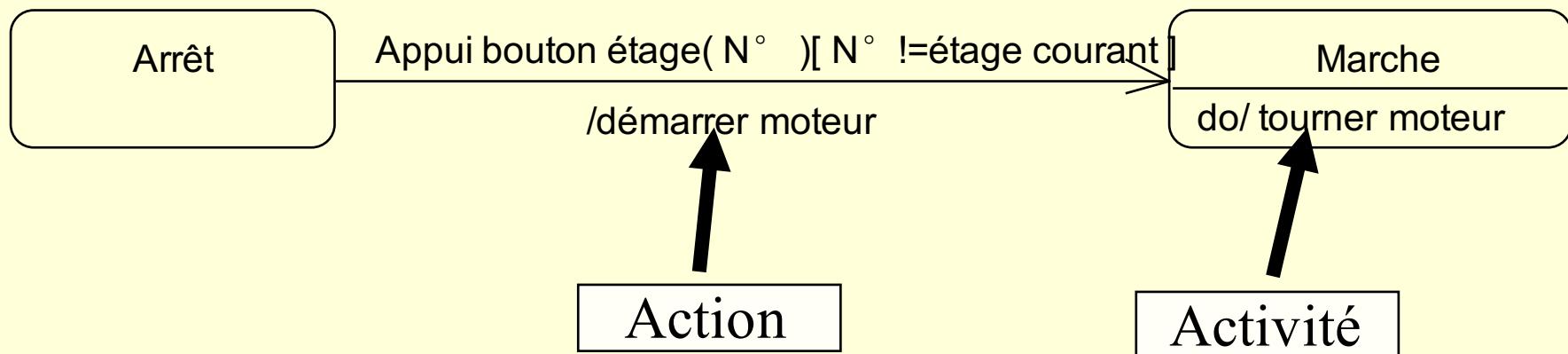
Transition

- Transition : relation entre 2 états indiquant qu'un objet dans le premier état va exécuter une action et entrer dans le deuxième état quand un événement apparaîtra
- Condition : expression booléenne devant être vérifiée pour permettre la transition

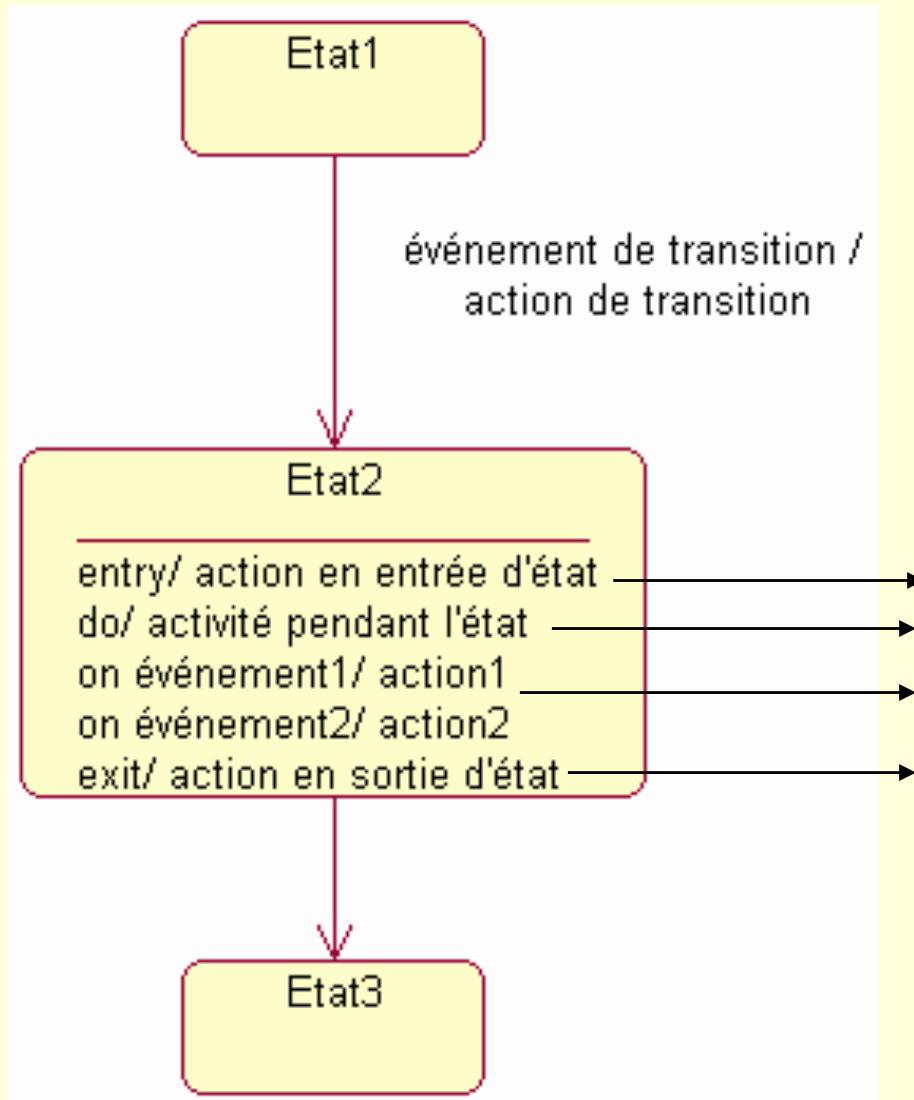


Activités et actions

- Action : opération atomique (non interruptible) déclenchée par une transition
- Activité : opération qui dure un certain temps (interruptible) dans un état particulier



Activités et actions (suite)



Action au cours d'une transition

Action en entrée d'état

Activité

Actions en réponse à un événement

Action en sortie

Evènements

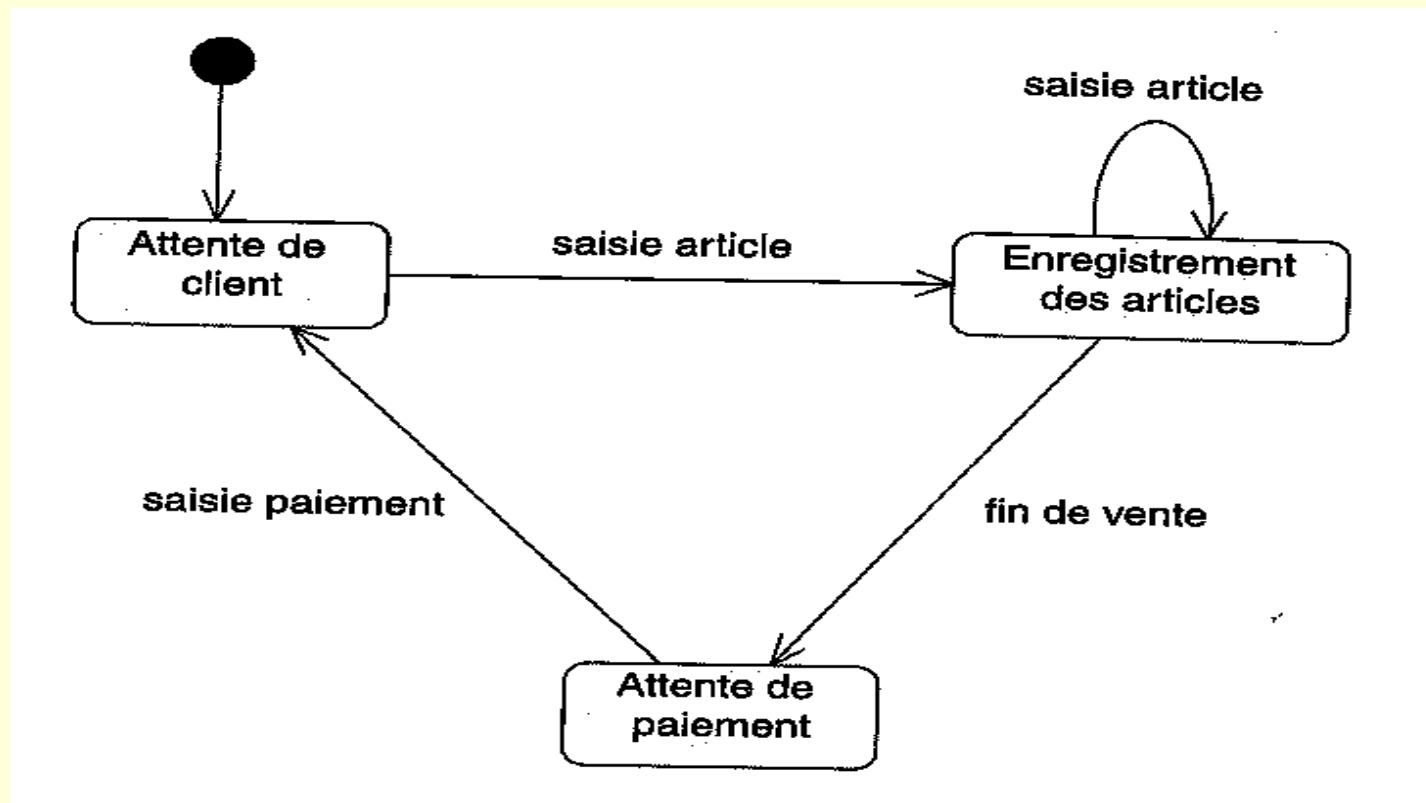
- Signal : envoi et réception asynchrone sans retour du récepteur
- Appel d'opération : demande d'exécution d'une fonction avec éventuellement des paramètres
- Temporel : déclenchement à une date donnée (when) ou après une temporisation (after)

Application

Modélisation d'une caisse automatique : L'état le plus courant est celui de l'attente d'un client. Lorsqu'un client scanne un article, elle se met à enregistrer les articles les uns après les autres. Si le client sélectionne « fin de vente », la caisse attend le paiement. Une fois le paiement saisi, la caisse attend le client suivant.

Application

Modélisation d'une caisse automatique : L'état le plus courant est celui de l'attente d'un client. Lorsqu'un client scanne un article, elle se met à enregistrer les articles les uns après les autres. Si le client sélectionne « fin de vente », la caisse attend le paiement. Une fois le paiement saisi, la caisse attend le client suivant.



Application

Cas Réveil-matin :

On peut mettre l'alarme en fonction ou hors fonction.

Quand l'heure courante devient égale à l'heure d'alarme, le réveil sonne.

On peut interrompre la sonnerie avec le bouton « Stop » (et le réveil re-sonnera le lendemain) ou en mettant l'alarme hors fonction, sinon la sonnerie s'arrête au bout de 59 minutes.

Etablir le diagramme d'états-transitions.

4. Modélisation de l'installation

Les diagrammes

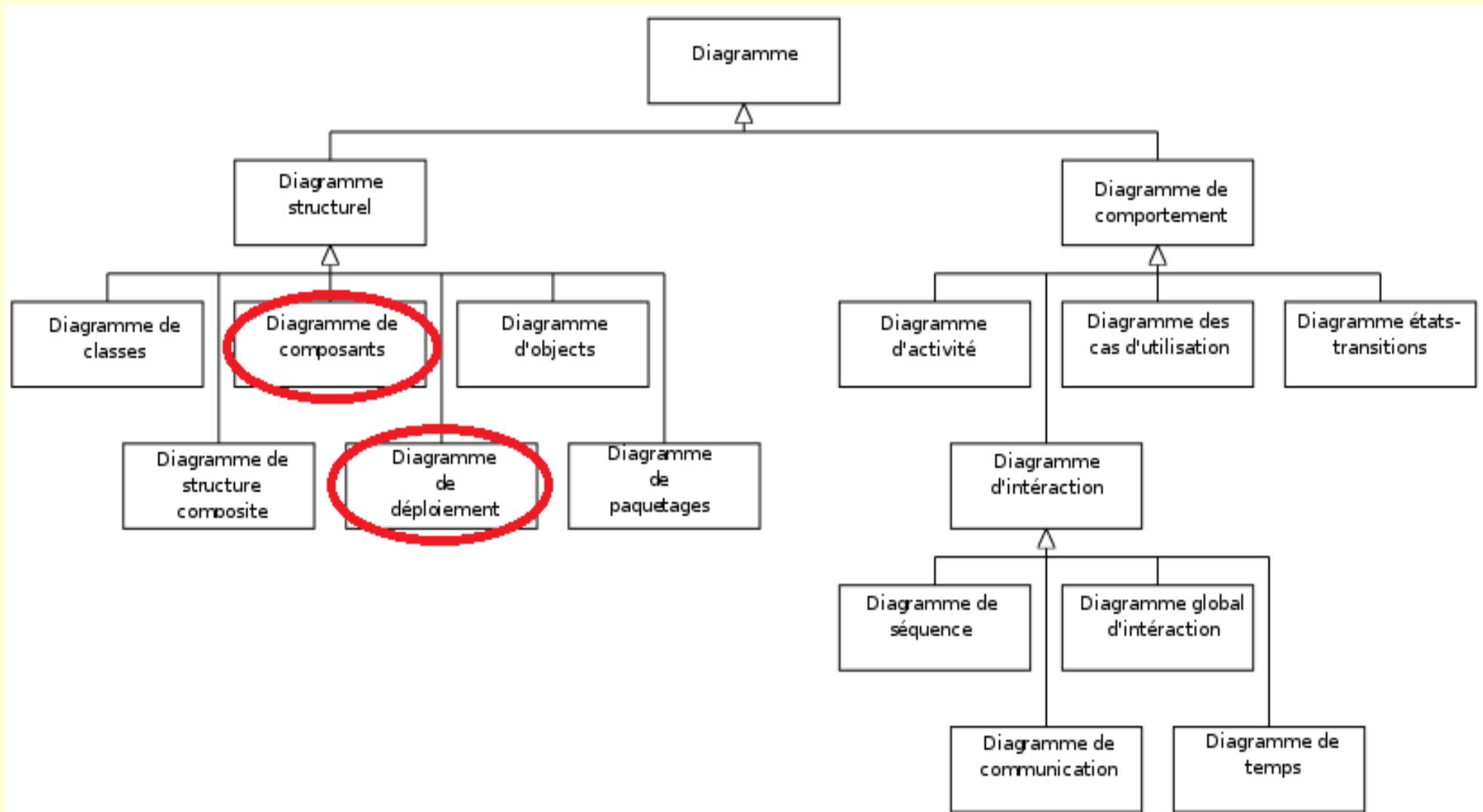


Diagramme de Déploiement

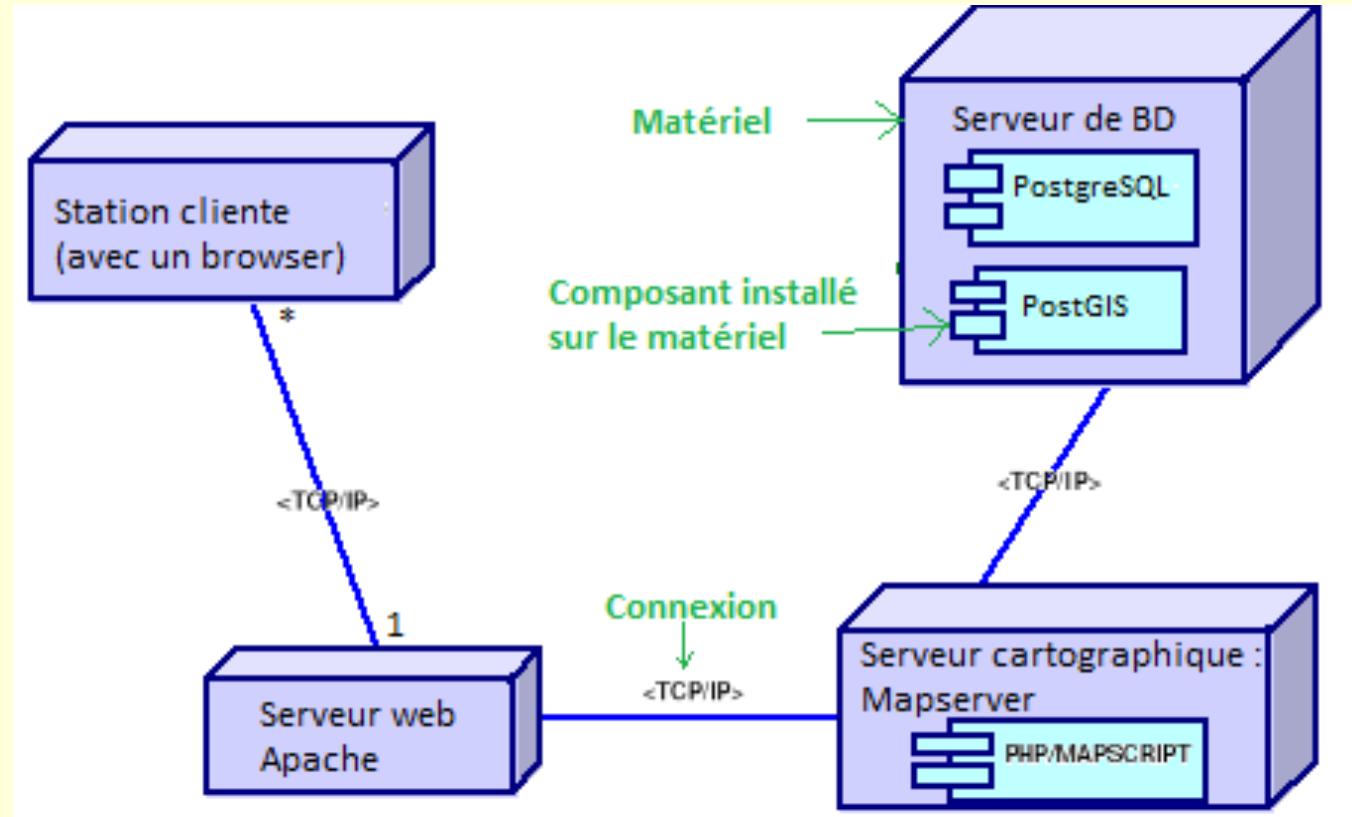
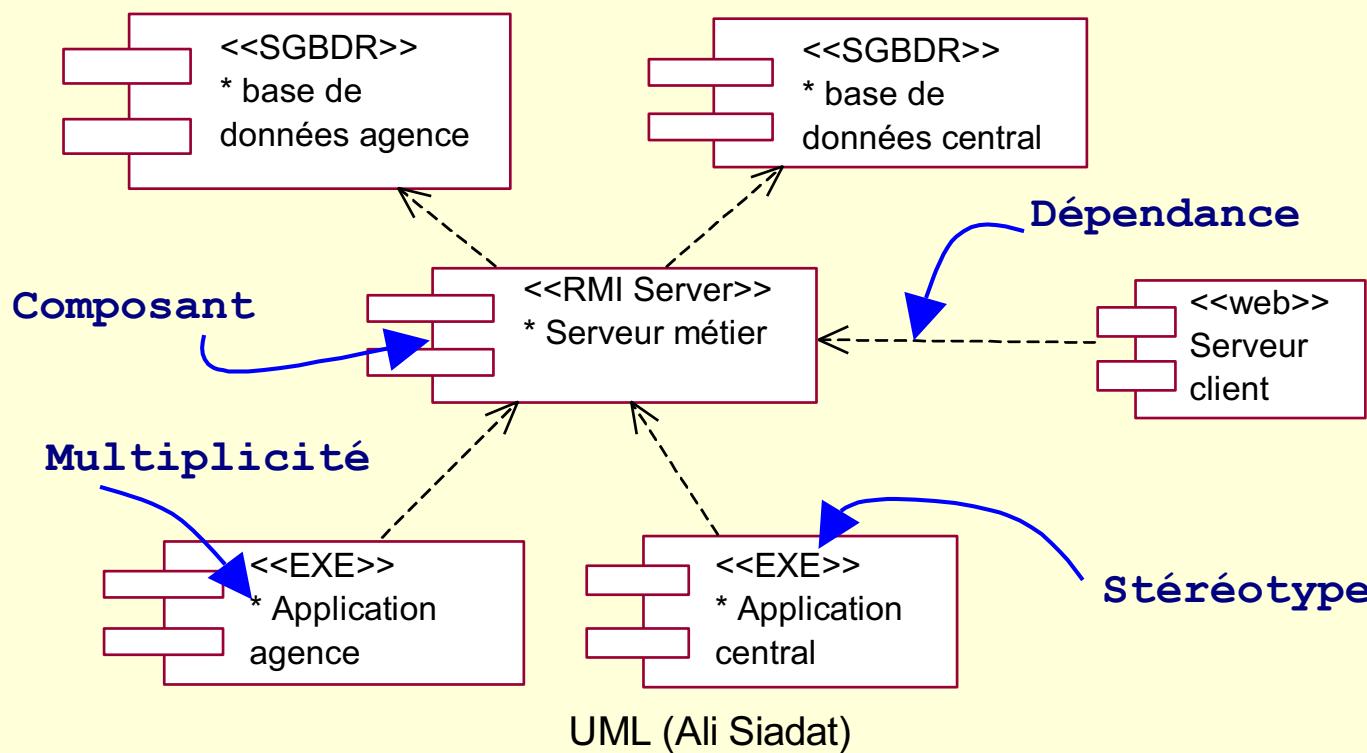


Diagramme de Composants

- Permet éventuellement de décrire les composants logiciels connus, dans le cadre d'une politique de réutilisation
- **Composant** = partie physique et remplaçable d'un système réalisant un ensemble d'interfaces



5. Applications

Diag. classes

Modéliser les relations entre les concepts suivants :

- a - un rectangle et un carré
- b - un segment et ses deux extrémités
- c - un château de cartes
- d - une classe et ses élèves
- e - les articles du magasin, leur référence magasin, les fournisseurs auprès desquels ils peuvent être acquis, leur référence chez leurs différents fournisseurs
- f - une poupée russe (c'est-à-dire une série de poupées contenues les unes dans les autres)
- g - une voiture, son modèle, son immatriculation
- h - un atelier et ses machines : meule, perceuse, presse
- i - un livre et ses pages
- j - les pages web et les images qui y apparaissent
- k - un hotel et ses chambres

Diag. classes : Cours

Dans un cursus universitaire, une matière (exemple : PML) est enseignée par plusieurs enseignants. Un enseignant peut enseigner plusieurs matières (ex : stratégie et GRH).

Le cours d'une matière dans un cursus (ex : la stratégie en master MFTIGS) est assuré par un enseignant, et correspond à un nombre d'heures de cours, de TD et de TP.

Diag. classes : Edition

Un auteur écrit des ouvrages. Un ouvrage est édité par des éditeurs. Un ouvrage peut être édité plusieurs fois par des éditeurs différents (réédition au format poche par exemple). Une édition fait l'objet d'un contrat. Les conditions générales sont stipulées par des contrats-types. Chaque contrat est rattaché à un contrat type et peut stipuler des conditions particulières. On doit connaître le nom et le prénom des auteurs, le titre des ouvrages, la date, le montant et les clauses des contrats, le nom des éditeurs.

Diag. classes : CE

Les activités offertes sont gérées par des sections de deux types :

- les **sections sportives** (squash, tennis, etc.). Chaque section peut être affiliée à une fédération sportive dont on veut conserver les coordonnées ;

- les **sections culturelles** (informatique, photographie, etc.).

On distingue deux sortes d'adhérents aux sections :

- les **adhérents internes** qui sont de deux types :

- les employés (qui appartiennent à un service de l'entreprise),
- les stagiaires (pour lesquels l'association désire connaître le nom de leur établissement de formation)

- les **adhérents externes** : une personne ne travaillant pas dans l'entreprise peut devenir membre de l'association si elle est parrainée par deux adhérents internes. Les conjoints et les enfants des employés sont souvent adhérents externes.

Le CE souhaite garder trace du nom des parrains de chaque adhérent externe.

Diag. UC : Vente en ligne

Le site internet B2B d'une entreprise s'adresse aux distributeurs potentiels de ses produits. Il permet de présenter les gammes de produits.

Le site propose également de passer des commandes. S'il n'y a pas de clients dans la région du distributeur, une proposition de parrainage lui est faite (un dispositif lui permettant d'attaquer ce nouveau marché).

Diag. UC, séquence : Cabinet médical

Un patient contacte le cabinet médical pour prendre rendez-vous. Le secrétariat lui propose une date et une heure de rendez-vous.

Le jour de son rendez-vous, il transmet sa carte vitale et d'éventuels renseignements pour la mise à jour de son dossier.

Il consulte le médecin, qui peut éventuellement lui prescrire un traitement sur une ordonnance.

Le patient règle le prix de la consultation auprès de la secrétaire. Suite au règlement du patient, le cabinet médical adresse directement la feuille de soin électronique à la sécurité sociale qui rembourse le patient.

Diag. UC : Magasin électroménager

Le client entre dans le magasin, passe dans les rayons, demande éventuellement des renseignements aux vendeurs, choisit des articles, les demande à un vendeur, qui vérifie si le stock est suffisant et lui remplit une fiche. Le client règle son produit en caisse, puis récupère son produit auprès du service enlèvement.

Le règlement peut s'effectuer de différentes façons : chèque, CB, espèces (Les procédures diffèrent d'un mode à l'autre). Le client peut bénéficier d'une réduction ou d'un avoir. Il peut demander une livraison si le produit est encombrant.

Le client peut effectuer une réclamation

Le client peut demander une prolongation de sa garantie pour ses achats

Le paiement par carte bancaire comprend la vérification auprès du groupement des banques l'autorisation de paiement liée à la carte.

Diag. activité : Salle des fêtes

Pour louer la salle de fêtes du village de Greuville, les habitants doivent en faire la demande auprès de la mairie. Ils précisent la date choisie. Si la salle est libre, la mairie leur propose de réserver la salle en signant un contrat de réservation et en versant la moitié du prix de la location. Ils doivent demander une attestation à leur assurance.

Trois mois avant la date de location, ils doivent transmettre cette attestation à la mairie, ainsi que le solde du prix de location et un chèque de caution qui n'est pas encaissé.

Le jour de la location, un état des lieux et un inventaire du mobilier sont effectués par la mairie.

A la fin de la location, un nouvel état des lieux et un nouvel inventaire du mobilier sont dressés. S'il n'y a pas eu de dégradation, le chèque de caution est rendu, sinon celui-ci est encaissé.

Dans tous les cas, une facture est remise.

Diag. activité : G° commandes

Voici le processus de la gestion des commandes dans le service commercial d'une grande entreprise.

Le client passe une commande. On identifie le client ; s'il est nouveau, on le crée, si la commande comporte une anomalie, on la rejette et le client reçoit un avis de rejet explicatif ; sinon, on enregistre la commande. On vérifie alors si tous les articles sont en stock suffisant ; si l'un d'entre eux est insuffisant, on met la commande en attente et on transmet au service des achats une demande de réapprovisionnement et au client, un avis d'indisponibilité. Si tous les articles sont en stock, la commande est confirmée et le client reçoit une confirmation. Le service des livraisons reçoit la commande à livrer et celui de l'administration, la commande à facturer.

Quand le service des achats envoie un avis d'entrée en stock, les commandes en attente sont réexaminées ; certaines peuvent alors être confirmées et le client reçoit une confirmation.

Diag. Etats-Trans. : Candidature

Pour demander à entrer dans une école privée, les candidats envoient leur dossier avant une date fixée. Une première commission examine les candidatures à l'inscription. A l'issue de cet examen, les dossiers sont soit rejetés, soit acceptés, soit mis en attente (dossier intéressant mais incomplet). Ces derniers seront réexaminés ultérieurement par une seconde commission, après qu'on ait reçu de la part des candidats, dans un certain délai, un dossier complété des informations supplémentaires demandées. Sans ce renvoi, le dossier est refusé, de même que lors du non respect des délais.

Lorsqu'un candidat reçoit un avis favorable, il doit retourner, dans un certain délai, un chèque égal au montant des droits d'inscription. Le dossier est alors clos et l'étudiant reçoit alors sa carte.

Diag. Etats-Trans. : Réparation

Le client doit prendre rendez-vous auprès du garage en indiquant le type de réparation souhaité sur son véhicule pour en estimer la durée. Le jour convenu, il présente sa voiture. Le client indique précisément les révisions et réparations à effectuer, notées sur une fiche suiveuse qui accompagnera le véhicule à réparer.

Si des réparations complémentaires conséquentes sont détectées on prend contact avec le client et on lui demande son accord sur le complément de réparation. S'il refuse, on ne procédera qu'aux travaux initialement envisagés.

A l'issue de la réparation, la facture est éditée.

Quand le client se présente pour retirer sa voiture, on lui remet la facture et on encaisse son chèque. La voiture est alors remise au client.

Diag. Etats-Trans. : Sas d'entrée

Un sas est à l'état fermé en permanence (porte d'entrée ouverte, porte de sortie bloquée)

Lorsqu'une personne entre dans le sas, la porte d'entrée se ferme (le sas est à l'état « complet »). La personne doit présenter son badge devant le lecteur dans les 3 minutes qui suivent son entrée.

Si le badge est valide, le sas s'ouvre (porte de sortie ouverte) pendant 2 minutes.

Si le badge n'est pas valide ou pas présenté dans le délai imparti, une sonnerie prévient le gardien. Celui-ci peut alors gérer manuellement l'ouverture de la porte de sortie du sas (personne autorisée) ou le déblocage de la porte d'entrée pour que la personne puisse repartir, ou prendre les mesures de sécurité nécessaires.

Diag. Etats-Trans. : GPS

Lorsque l'utilisateur allume le navigateur, celui-ci calcule sa position et propose alors son menu. Sur ce menu, l'utilisateur peut choisir entre le calcul d'un itinéraire, l'affichage du plan autour de la position en cours ou l'arrêt du GPS. Si l'utilisateur choisit le calcul d'un itinéraire, le navigateur GPS affiche l'écran de saisie avec un clavier interactif de lettres. Lorsque l'utilisateur appuie sur Valider, le navigateur vérifie la validité de l'adresse et passe en mode navigation (affichage de l'itinéraire et annonce des instructions de conduite). Si l'adresse saisie est invalide, le navigateur affiche un message d'erreur et réaffiche immédiatement l'écran de saisie de l'adresse. A tout moment, l'utilisateur peut retourner au menu.