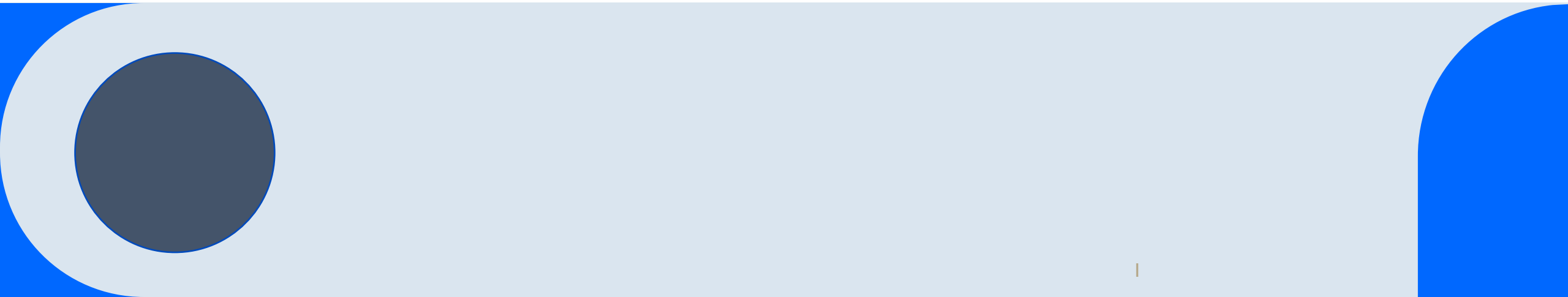


Analyse et conception des systèmes d'informations



Vision globale d'une entreprise

Décider des actions à conduire

Système de DECISION

**Collecter, mémoriser,
traiter, distribuer
l'information**

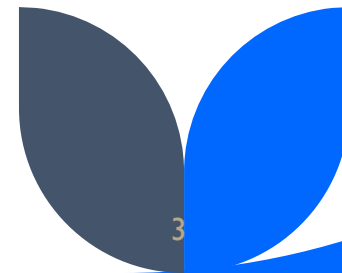
**Système
d'INFORMATION**

Assurer les fonctions :
· facturer les clients,
· fabriquer les produits,
....

Système OPERANT

L'information...?

- ❑ Ensemble complexe, composé de données et de liens.
- ❑ Décrit une activité.
- ❑ Plusieurs valeurs possibles.

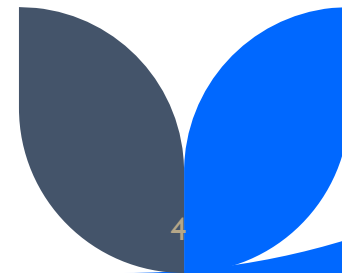


Système d'information - Définition

Un système d'Information (noté SI) représente l'ensemble des éléments participant :

- à la gestion
- au traitement
- au transport
- à la diffusion

de l'information au sein de l'organisation.



Comment réaliser un « bon » système d'information ?

La réponse sur les techniques et démarches classiques du Génie Logiciel :

- **Analyse**

- de l'existant et des besoins de l'utilisateur.

- **Conception**

- du système et du logiciel,

- **Réalisation**

- Traduction des algorithmes dans un langage choisi.

- **Tests du logiciel**

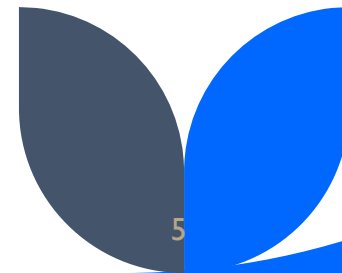
- Vérification et validation du logiciel.
- Tests de non régression.

- **Exploitation**

- Utiliser le logiciel une fois installé.

- **Maintenance**

- Correction des erreurs.
- Ajouts de fonctionnalité.
- ...



Analyse et Conception de Système d'Information

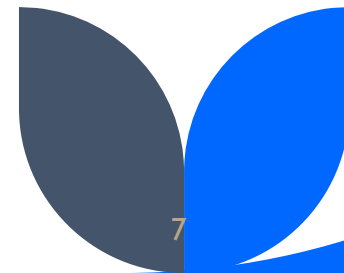
- ✓ **MODELES**
- ✓ **METHODOLOGIES**

Modèle?

Un modèle est une représentation abstraite, d'une partie du monde réel, exprimée dans un langage de représentation.

Ce langage peut être :

- **Formel** : ayant une syntaxe et une sémantique bien définies comme :
 - ❖ la logique
 - ❖ un langage informatique
- **Semi-formel** : notation graphique normalisée.
- **Informel** : description en langage naturel.



Modèle...pourquoi?

Les principales motivations sont :

- **Comprendre** et **analyser** la structure et le fonctionnement de l'entreprise
- **Prévoir** (de manière fiable) le comportement et les performances des processus opérationnels avant de les implémenter.
- **Choisir** la (ou les) meilleure(s) alternative(s) d'implantation
- **Identifier** les risques d'implantation à gérer
- **Justifier** les choix d'implantation sur des critères liés aux ressources et aux coûts
- **Bâtir** une vision commune du fonctionnement de l'entreprise et la **Communiquer** facilement au plus grand ensemble possible du personnel.



Méthodologie...pourquoi?

Formalisation claire et complète du problème informationnel.

Maîtrise de la résolution du problème par l'utilisation de critères objectifs pour évaluer les solutions.

Construction de SI pertinents, complets, cohérents, fiables flexibles et adaptatifs.

Évaluation du SI à tout moment de son cycle de vie.

Faciliter la coopération entre concepteurs, informaticiens gestionnaires, utilisateurs.

Rigueur dans l'élaboration de la solution.

Réduire les coûts et les délais.



Méthodologie... Typologie?

- **Approche cartésienne.**
Orientée traitements.
- **Approche systémique.**
Orientée données.
- **Approche Objet.**
Orientée données et traitements.

Outils de modélisation



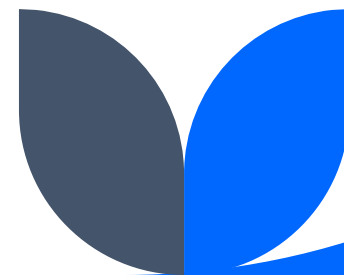
Lucidchart est une plateforme de collaboration en ligne, basée sur le cloud, permettant la :

- ✓ Création de diagrammes
- ✓ Visualisation de données

Lien vers le site : <https://www.lucidchart.com/pages/>

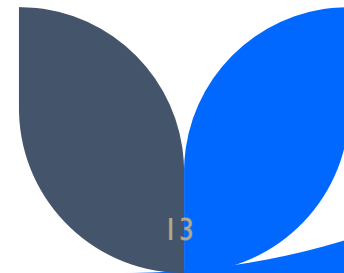
Méthodologie... Exemple?

**MERISE : Méthode d'Etude et de Réalisation
Informatique pour les Systèmes d'Entreprises**



Merise - Définition

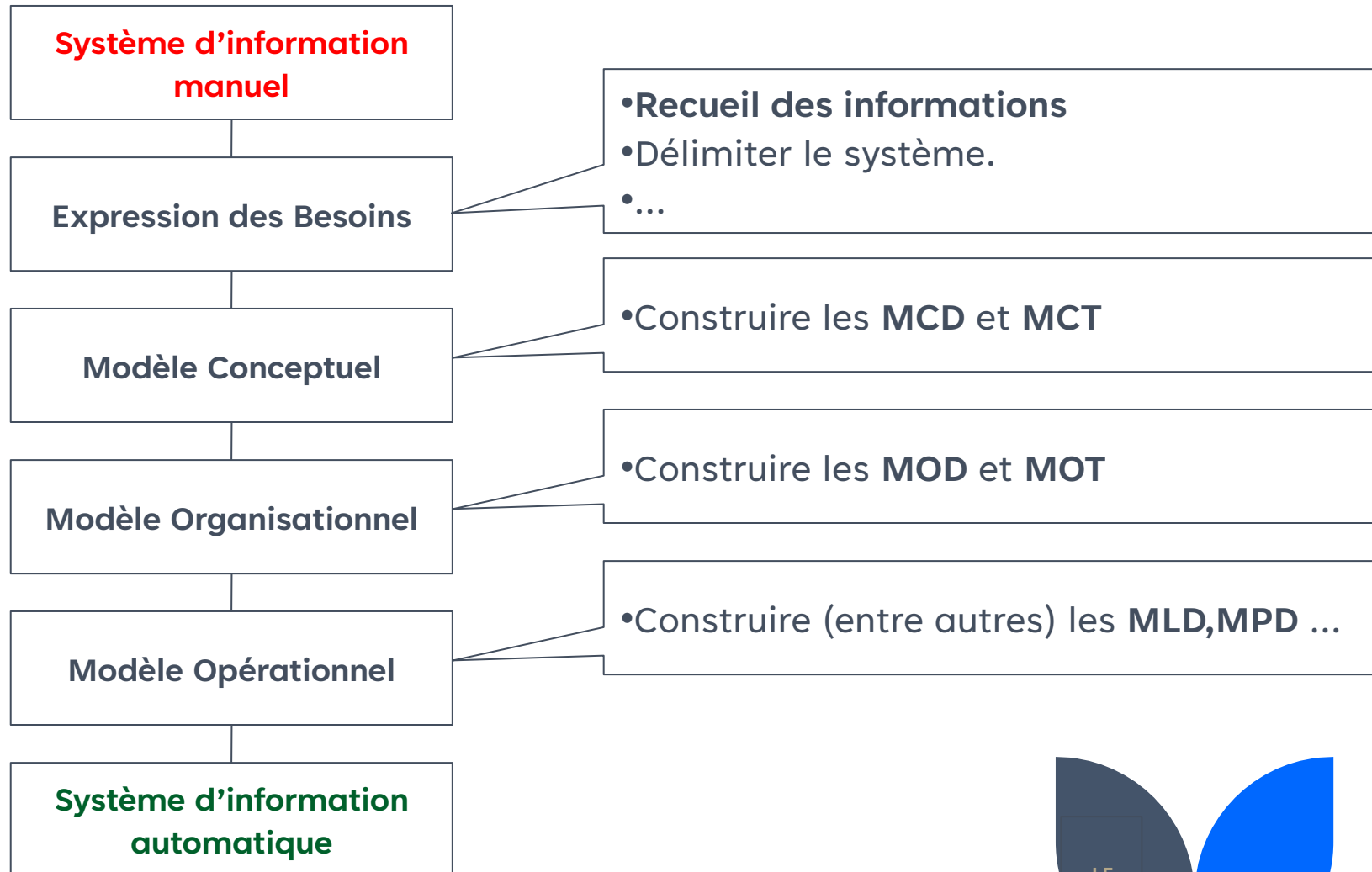
- MERISE est une méthode d'analyse informatique et une démarche de construction des systèmes d'information (SI)
- Elle s'est apparue à la fin des années 1970
- Elle utilise un ensemble de signes graphiques pour représenter un modèle
- Elle permet de faire un lien de communication entre les différents acteurs d'un projet



Merise...deux modèles

- Deux modèles : *données* et *traitements*.
- Elaborés séparément.
- Déclinés sur trois niveaux :
 1. Conceptuel.
 2. Organisationnel.
 3. Opérationnel.
- Validation données / traitements.

Merise...cycle d'abstraction

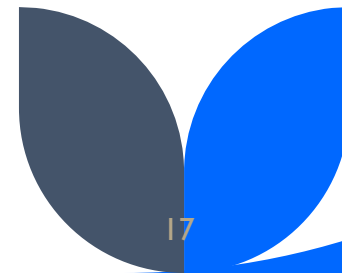




MCD : Modèle Conceptuel de données

Modèle Conceptuel de Données (MCD)

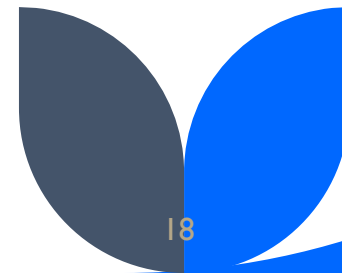
- Toute donnée recensée doit être mémorisée.
- Le MCD modélise cette mémoire (collective) du système.
- Un formalise de référence :
 - Modèle **Entité-Association**.
 - Concepts d'entités et d'associations.
 - Particulièrement adapté aux Base de Données relationnelles.
- Redondance interdite !



Modèle Conceptuel de Données (MCD)

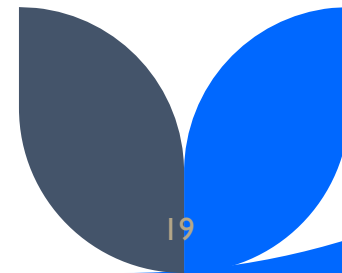
L'élaboration du MCD passe par les étapes suivantes :

1. La mise en place de **règles de gestion**
2. L'élaboration du **dictionnaire des données**
3. L'élaboration du MCD (création des **entités** puis des **associations** puis ajout des **cardinalités**).



Modèle Conceptuel de Données (MCD) règles de gestion

Les règles de gestion sont l'ensemble des informations qui vont nous aider à modéliser et concevoir le SI



Modèle Conceptuel de Données (MCD)

règles de gestion -Exemple

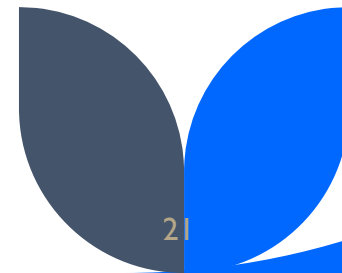
Les règles de gestion du SI d'une bibliothèque :

- Pour chaque livre, on doit connaître le titre, l'année de parution, un résumé et le type (roman, poésie, science-fiction...);
- un livre peut être rédigé par aucun (dans le cas d'une œuvre anonyme), un ou plusieurs auteurs dont on connaît le nom, le prénom, la date de naissance et le pays d'origine ;
- Chaque exemplaire d'un livre est identifié par une référence composée de lettres et de chiffres et ne peut être paru que dans une et une seule édition ;
- Un inscrit est identifié par un numéro et on doit mémoriser son nom, prénom, adresse, téléphone et adresse e-mail ;
- Un inscrit peut faire zéro, un ou plusieurs emprunts qui concernent chacun un et un seul exemplaire. Pour chaque emprunt, on connaît la date et le délai accordé (en nombre de jours).

Modèle Conceptuel de Données (MCD) règles de gestion

Une agence immobilier désire avoir une application dans laquelle doit inscrire son fichier des maisons, des propriétaires et des locataires.

Travail à faire: trouver l'ensemble des données à utiliser dans l'application.



Modèle Conceptuel de Données (MCD) règles de gestion

Une agence de location de voitures vous demande de réaliser une application pour gérer les clients et les voitures.

Travail à faire: trouver l'ensemble des données à utiliser dans l'application.

Modèle Conceptuel de Données (MCD) dictionnaire des données

Le dictionnaire de données est un document qui permet de recenser, de classer et de trier toutes les données collectées lors des entretiens et/ou de l'étude des documents.

Modèle Conceptuel de Données (MCD)

dictionnaire des données

Pour chaque donnée, il indique :

Code mnémonique : libellé désignant une donnée

Désignation : une mention décrivant ce à quoi la donnée correspond

Type de donnée :

A ou Alphabétique : lorsque la donnée est uniquement composée de caractères alphabétiques (de 'A' à 'Z' et de 'a' à 'z')

N ou Numérique : lorsque la donnée est composée uniquement de nombres (entiers ou réels)

AN ou Alphanumérique : lorsque la donnée peut être composée à la fois de caractères alphabétiques et numériques

Date : lorsque la donnée est une date (au format AAAA-MM-JJ),

Booléen : Vrai ou Faux

Taille : nombre de caractères ou de chiffres

Remarques ou observations complémentaires (par exemple si une donnée est strictement supérieure à 0, etc.)

Modèle Conceptuel de Données (MCD)

dictionnaire des données -Exemple

Reprenons l'exemple de notre bibliothèque et du système de gestion des emprunts que nous sommes chargés d'informatiser. Après l'étude des règles de gestion, nous pouvons établir le dictionnaire des données suivant :

Code mnémonique	Désignation	Type	Taille	Remarque
id_i	Identifiant numérique d'un inscrit	N		
nom_i	Nom d'un inscrit	A	30	
prenom_i	Prénom d'un inscrit	A	30	
rue_i	Rue où habite un inscrit	AN	50	
ville_i	Ville où habite un inscrit	A	50	
cp_i	Code postal d'un inscrit	AN	5	
tel_i	Numéro de téléphone fixe d'un inscrit	AN	15	
tel_port_i	Numéro de téléphone portable d'un inscrit	AN	15	
email_i	Adresse e-mail d'un inscrit	AN	100	
date_naissance_i	Date de naissance d'un inscrit	Date	10	Au format AAAA-JJ-MM
id_l	Identifiant numérique d'un livre	N		
titre_l	Titre d'un livre	AN	50	
annee_l	Année de parution d'un livre	N	4	
resume_l	Résumé d'un livre	AN	1000	
ref_e	Code de référence d'un exemplaire d'un livre	AN	15	Cette référence servira également d'identifiant dans ce système
id_t	Identifiant numérique d'un type de livre	N		
libelle_t	Libellé d'un type de livre	AN	30	
id_ed	Identifiant numérique d'une édition de livre	N	6	
nom_ed	Nom d'une édition de livre	AN	30	
id_a	Identifiant numérique d'un auteur	N		
nom_a	Nom d'un auteur	A	30	
prenom_a	Prénom d'un auteur	A	30	
date_naissance_a	Date de naissance d'un auteur	Date		Au format AAAA-JJ-MM
id_p	Identifiant numérique d'un pays	N		
nom_p	Nom d'un pays	A	50	
id_em	Identifiant numérique d'un emprunt	N		
date_em	Date de l'emprunt	Date		Au format AAAA-JJ-MM
delais_em	Délai autorisé lors de l'emprunt du livre	N	3	S'exprime en nombre de jours

Modèle Conceptuel de Données (MCD)

dictionnaire des données -Exemple

La fiche suivant est une fiche d'adhérent de l'association Geev. Elaborer son dictionnaire de données.

Association Geev

Fiche d'adhérent

Identifiant : GHJ123

Nom : Bennay

Prénom : Nael

Adresse : 58 quartier marne

Code postal :75000

Ville : Paris :

Téléphone : 06-55-77-88-99

Email : bennay.nael@gmail.com

Date d'adhésion : 20/03/2022

Modèle Conceptuel de Données (MCD)

dictionnaire des données - Exemple - Geev

Code mnémonique	Désignation	Type	Taille	Remarque
Id_adherent	Identifiant	AN	6	3 caractères -3 chiffres
nom	Nom d'adhérent	A	30	
pren	Prénom d'adhérent	A	30	
adr_adherent	Adresse d'adhérent	AN	50	
cp	Code postal	N	5	
Ville	Ville d'adhérent	A	50	
tel	Numéro téléphone d'adhérent	AN	10	
mail	Email d'adhérent	AN	50	
date_adherent	Date d'adhérent	Date		JJ-MM-AAAA

Modèle Conceptuel de Données (MCD)

Définition

Les éléments de base de MCD sont :

- ☐ Les propriétés
- ☐ Les entités
- ☐ Les relations

Modèle Conceptuel de Données (MCD)

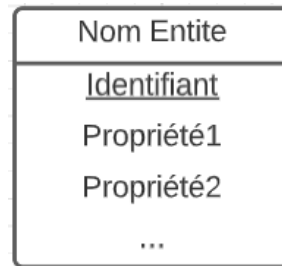
Propriétés

- ❑ Les propriétés sont les informations de base qui décrivent les éléments(les entités) d'un SI.
- ❑ Les propriétés sont appelées **attributs** ou **caractéristiques**
- ❑ Exemple: identifiant auteur, nom auteur, prénom auteur, adresse auteur sont des propriétés qui décrivent l'élément(l'entité) Auteur.
- ❑ Chaque propriété dispose d'un **type**(alphabétique, alphanumérique, numérique, date,...).

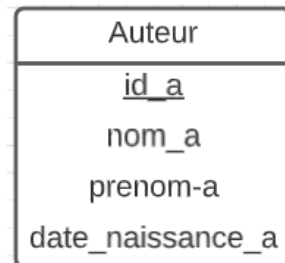
Modèle Conceptuel de Données (MCD)

Entités

- ❑ Une entité est la représentation d'un élément dans un SI.
- ❑ Chaque entité regroupe un ensemble de propriétés.
- ❑ La représentation d'une entité s'appelle une occurrence,
- ❑ Le formalisme d'une entité est le suivant :



- ❑ Exemple : si on reprend notre dictionnaire de données précédent, on schématise par exemple une entité «**Auteur**» comme ceci :



Modèle Conceptuel de Données (MCD)

Occurrences d'une entité

- ❑ Une occurrence est un élément particulier d'une entité
- ❑ Une occurrence à des valeurs pour chaque propriété de l'entité.
- ❑ Une occurrences est parfois appelée **tuple**
- ❑ Une entité peut n'avoir aucune, une ou plusieurs occurrences.

Voici un exemple de table d'occurrences de l'entité Auteur :

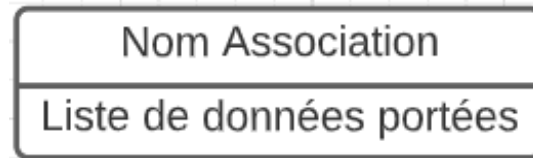
id_a	nom_a	prenom_a	date_naissance_a
1	Hugo	Victor	1802-02-26
2	Rimbaud	Arthur	1854-10-20
3	de Maupassant	Guy	1850-08-05

Rmq : **L'identifiant** est une propriété qui permet de connaître de façon unique et sûre les occurrences d'une entité donnée.

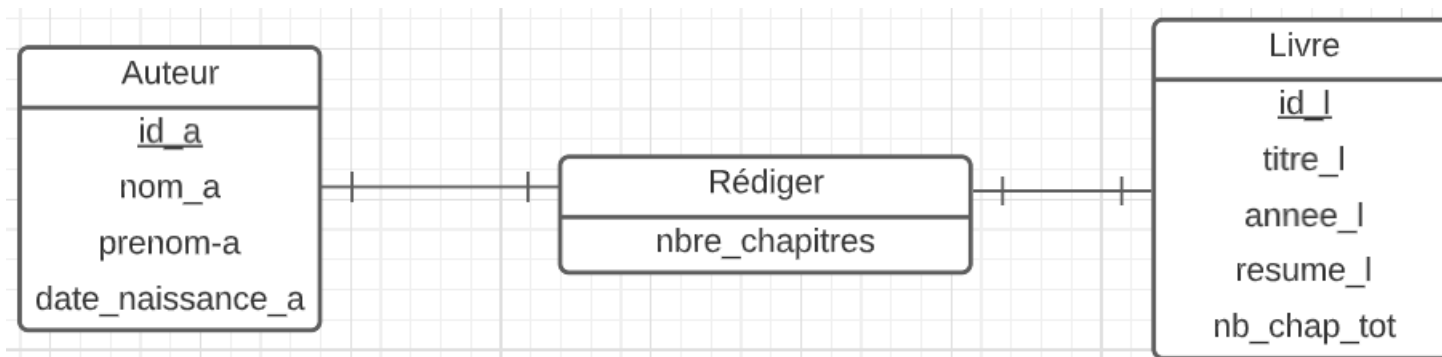
Modèle Conceptuel de Données (MCD)

Relation ou Association

- ❑ Une relation ou association est la liaison qui lie entre les entités du SI.
- ❑ Le formalisme d'une association est le suivant :



- ❑ Généralement le nom de l'association est un verbe définissant le lien entre les entités qui sont reliées par cette dernière.
- ❑ Exemple : Une partie du MCD pour SI d'une bibliothèque :



Modèle Conceptuel de Données (MCD)

Cardinalité

- ❑ Une cardinalité est le nombre de fois où l'occurrence d'une entité participe aux occurrences de la relation.
- ❑ Elle est définie comme ceci : **minimum, maximum**
 - ❖ La cardinalité minimale (**0 ou 1**) exprime le nombre de fois minimum qu'une occurrence d'une entité participe aux occurrences d'une relation.
 - ❖ La cardinalité maximale (**1 ou n**) exprime le nombre de fois maximal qu'une occurrence d'une entité participe aux occurrences d'une relation.
- ❑ Les cardinalités les plus répandues sont les suivantes :
0,n 1,n 0,1 1,1

Modèle Conceptuel de Données (MCD)

Cardinalité -Exemple

❑ Dans notre exemple nous devons poser les questions suivantes:

○ Entité Auteur :

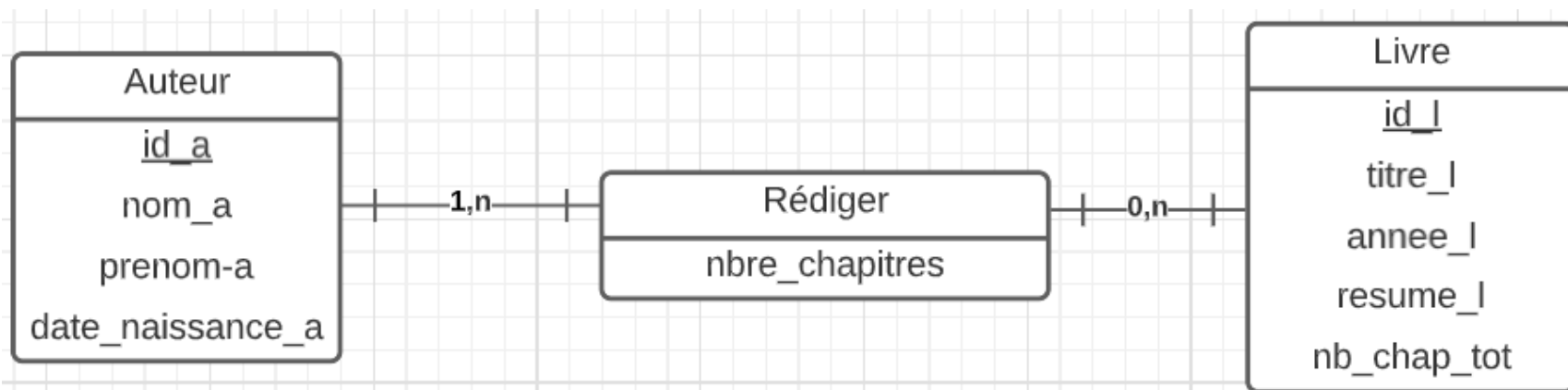
1. Combien de fois au minimum un auteur peut rédiger un livre ?
2. Combien de fois au maximum un auteur peut rédiger un livre ?

un auteur rédige au moins un ou plusieurs livres (1,n)

○ Entité Livre

1. Quelle est le nombre minimum de chapitres rédigés par l'auteur ?
2. Quelle est le nombre maximum de chapitres rédigés par l'auteur ?

pour chaque livre, on connaît le nombre de chapitres rédigés par l'auteur



Modèle Conceptuel de Données (MCD)

Cardinalité - Remarque

- ❑ Pour les cardinalités, il n'y a pas de règles exactes à suivre, tout est question d'interprétation, au sein d'une équipe de développement, il peut y avoir des divergences de point de vue.
- ❑ Pour les cardinalités, il faut être le plus logique possible, se référer aux règles de gestion édictées par le commanditaire de l'application.

Exercice

SI – Gestion des commandes des clients

1. Définir les règles de gestion du SI?

- Indications : Acteurs, relations entre eux, ...
- Un client est identifié par un numéro unique et caractérisé par un nom, un prénom, un âge, une adresse et une ville.
- Un produit est identifié par un numéro unique et caractérisé par une désignation, un prix d'achat et un prix de vente.

2. Définir le dictionnaire de données du SI :

3. Elaborer le MCD du SI en question

Solution

SI – Gestion des commandes des clients

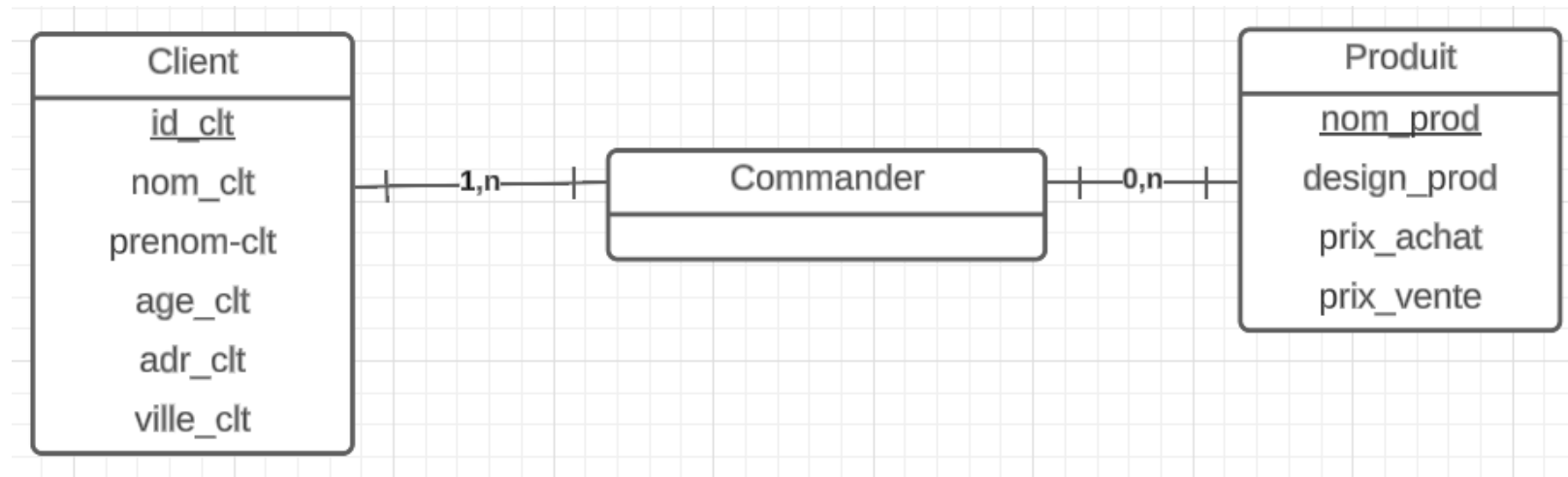
1. Les règles de gestion du SI sont les suivants :
 - Un client peut commander au moins un produit (1,n)
 - Un produit peut être acheté par aucun ou plusieurs clients (0,n)
2. Le dictionnaire de données du SI est le suivant :

Code mnémonique	Désignation	Type	Taille
id_clt	Numéro du client	N	
nom_clt	Nom du client	AN	30
prenom_clt	Prénom du client	AN	30
age_clt	Âge du client	N	2
adr_clt	Adresse du client	AN	70
ville_clt	Ville du client	A	50
num_prod	Numéro du produit	AN	12
design_prod	Désignation produit	AN	56
prix_achat	Prix d'achat	N	
prix_vente	Prix de vente	N	

Solution

SI – Gestion des commandes des clients

3 . MCD



Exercice

SI – Gestion de l'hôtellerie

Un groupe hôtelier vous demande de réaliser une application de gestion hôtelière. Cette application doit permettre la gestion de 8 hôtels, chaque hôtel possède 100 chambres maximum.

Ces hôtels sont répartis en 5 classes(*,**,***,****,*****).

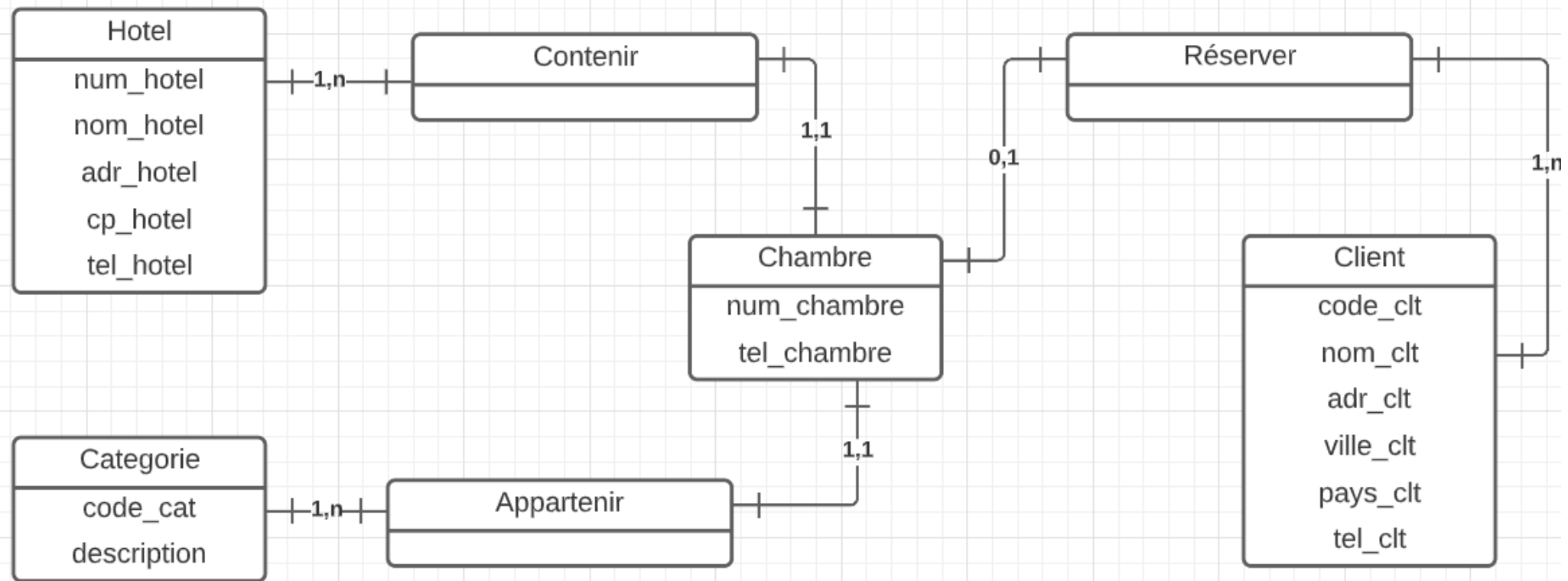
Pour chaque hôtel il y a au maximum 9 catégories de chambres différentes (capacité, degré de confort,...)

Cette application doit gérer aussi, les clients ainsi que leurs réservations.

- Travail à faire:
 - 1- Trouver les entités en proposant leurs propriétés.
 - 2- Trouver les associations et les cardinalités.
 - 3- Elaborer le Modèle Conceptuel de Données.

Solution

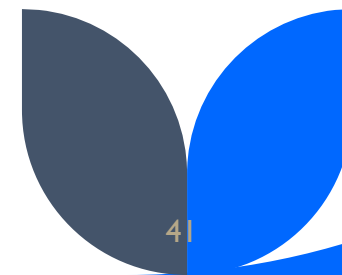
SI – Gestion de l'hôtellerie



Le modèle conceptuel de données : MCD

Notion de dépendances fonctionnelles (DF)

- La dépendance fonctionnelle signifie qu'un élément B dépend fonctionnellement de A si la connaissance de A implique la connaissance de B et on note $A \rightarrow B$.
- La DF s'applique dans les cas suivants :
 - ✓ Entre les attributs d'une même entité: il s'agit d'une DF entre l'identifiant d'une entité et les autres attributs de l'entité.
 - ✓ Entre les attributs de plusieurs entités: il s'agit d'une DF entre les identifiants des entités participantes à une association et les attributs de cette dernière.





MLD : Modèle Logique de données

Passage de MLD au SQL

Modèle logique de Données (MLD)

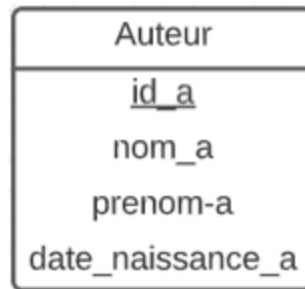
- ❑ Modèle relationnel ou MLD : Ensemble de schémas relationnels de la forme : Relation(clé1, ... clé, att1, ... attm)
- ❑ Les relations sont à la fois issues des entités du MCD, mais aussi d'associations.
- ❑ Une relation possède :
 - **Un nom** : correspond à celui d'une entité ou d'une association
 - **Une clef primaire** : permet d'identifier sans ambiguïté chaque occurrence de la relation en question. Elle est composée d'une ou plusieurs attributs.
 - **Un ensemble d'attributs** : sont des données élémentaires issues des propriétés des différentes entités, mais aussi des identifiants et des données portées par certaines associations.
 - **La clef étrangère** : est un attribut d'une relation qui fait référence à la clef primaire d'une autre relation
Par convention, on fait précéder ou suivre la clef étrangère du symbole #

NB : Une relation peut posséder aucune, une ou plusieurs clefs étrangères, mais possède toujours une et une seule clef primaire.

Passage de MCD à MLD - Règles de conversion

Conversion d'une entité

- ❑ En règle générale, toute entité du MCD devient une relation dont la clef est l'identifiant de cette entité.
- ❑ Chaque propriété de l'entité devient un attribut de la relation correspondante.
- ❑ Exemple :

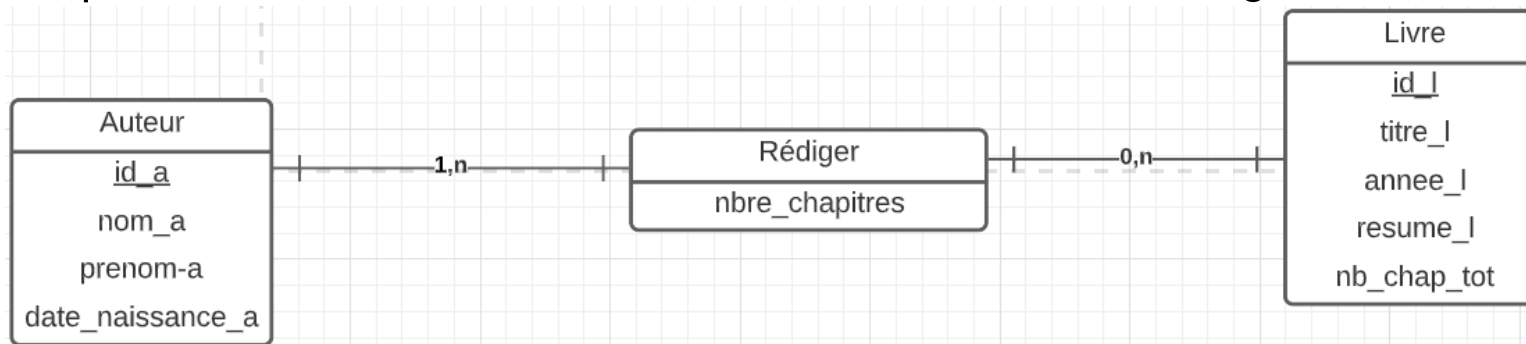


Conversion de l'entité Auteur à la relation suivante : Auteur(id_a, nom_a, prenom_a, date_naissance_a)

Passage de MCD à MLD - Règles de conversion

Conversion d'associations n'ayant que des cardinalités de type 0/1,N

- ❑ Une association ayant des cardinalités 0,N ou 1,N de part et d'autre devient une relation, appelé **relation associative**, dont la clef est constituée des identifiants des entités reliées par cette association.
- ❑ Ces identifiants seront donc également des clefs étrangères respectives.
- ❑ Voici un exemple de relation associative issu de l'association « rédiger » de notre MCD :



L'association « rédiger » est traduite comme ceci : Rediger (id_a#, id_l#, nb_chapitres)

Passage de MCD à MLD - Règles de conversion

Conversion des associations ayant au moins une cardinalité de type 1,1

On dispose plusieurs possibilités dans ce cas de figure :

❑ Méthode 1 :

- ❖ La règle de conversion la plus répandue est d'ajouter une clef étrangère dans la relation qui correspond à l'entité se situant du côté de cette cardinalité 1,1. Cette clef étrangère fera donc référence à la clef de la relation correspondant à la seconde entité reliée par l'association.
- ❖ Prenons un exemple issu de l'association « être originaire de » et des entités « Auteur » et « Pays » :



- Pays(id_p,nom_p)
 - Auteur (id_a, nom_a, prenom_a, date_naissance_a, id_p #)
-
- ❖ Lorsque l'on applique cette règle de conversion, deux restrictions s'imposent :
 - l'association ne peut être porteuse de données.
 - l'association doit être binaire (c'est-à-dire relier uniquement deux entités et pas plus).

Passage de MCD à MLD - Règles de conversion

Conversion des associations ayant au moins une cardinalité de type 1,1

On dispose plusieurs possibilités dans ce cas de figure :

❑ Méthode 2 :

- ❖ Lorsque deux entités sont toutes deux reliées avec une cardinalité 1,1 par une même association, on peut placer la clef étrangère de n'importe quel côté.

Par convention, on choisit de la placer du côté de la relation correspondant à l'entité ayant le plus de liaisons avec les autres

Passage de MCD à MLD - Règles de conversion

Conversion des associations ayant au moins une cardinalité de type 1,1

On dispose plusieurs possibilités dans ce cas de figure :

❑ **Méthode 3 :**

- ❖ Créer une relation associative dont la clef est cette fois composée uniquement de la clef étrangère qui fait référence à l'identifiant de l'entité du côté à la cardinalité 1,1.

Si on reprend le même exemple, voici ce que l'on devrait obtenir :

- Pays (id_p, nom_p)
- Auteur (id_a, nom_a, prenom_a, date_naissance_a)
- EtreOriginaireDe (id_a#, id_p#)

NB : l'association peut être porteuse de données. Ces dernières deviendront donc des attributs de la relation associative comme dans le cas des cardinalités 0,1/N.

Passage de MCD à MLD - Règles de conversion

Conversion des associations ayant au moins une cardinalité de type 0,1

Nous avons deux possibilités :

- ❖ Créer la clef étrangère dans la relation correspondant à l'entité du côté de la cardinalité 0,1.

NB :

L'association ne peut pas être porteuse de données ;

Une association ayant une cardinalité 0,1 doit être binaire

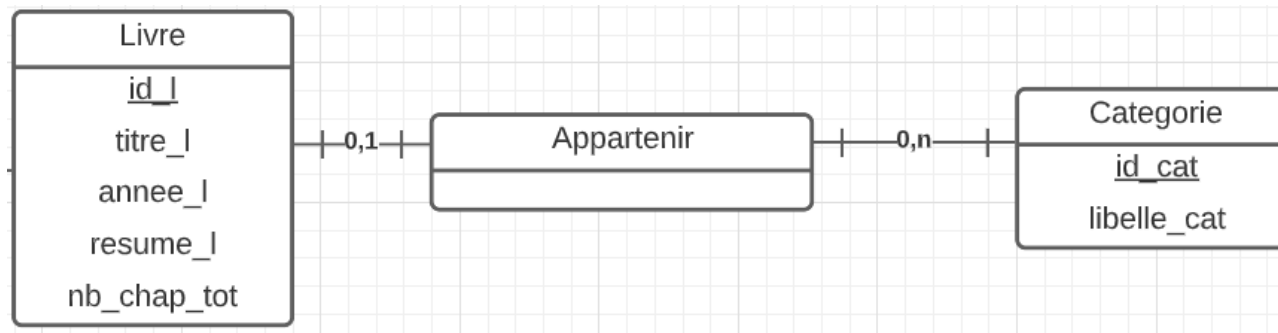
- ❖ Créer une **relation associative** dont la clef est cette fois composée uniquement de la clef étrangère qui fait référence à l'identifiant de l'entité du côté opposé à la cardinalité 0,1.

Passage de MCD à MLD - Règles de conversion

Conversion des associations ayant au moins une cardinalité de type 0,1

Exemple :

Un livre peut appartenir à 0 ou 1 catégorie, on obtient le MCD suivant :



➤ Méthode 1 :

Categorie (id_cat, libelle_cat)

Livre (id_l, titre_l, annee_l, resume_l, id_cat#)

➤ Méthode 2 :

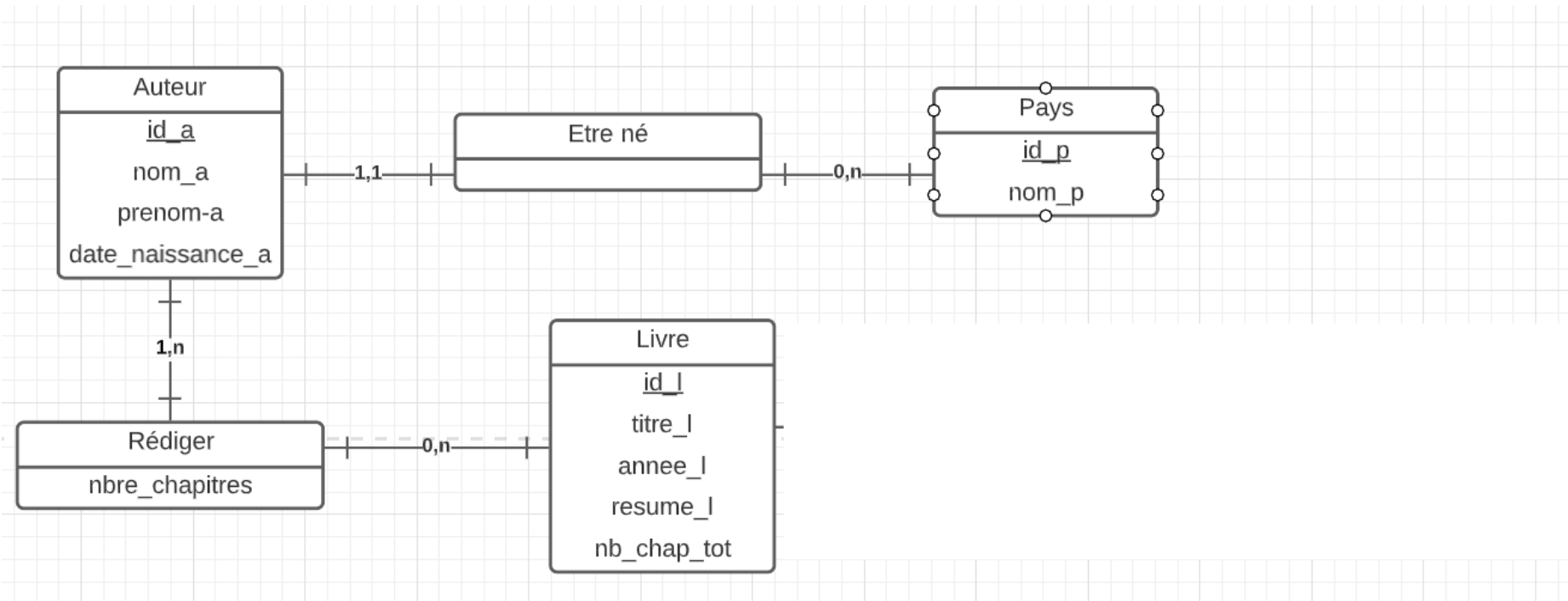
Categorie (id_cat, libelle_cat)

Livre (id_l, titre_l, annee_l, resume_l)

Appartenir (id_l#, id_cat#)

Élaboration du MLD et passage au SQL

Elaborer le modèle relationnel de MCD suivant



Élaboration du MLD et passage au SQL

Voici le MLD :

- **Pays** (id_p, nom_p)
- **Auteur** (id_a, nom_a, prenom_a, date_naissance_a, id_p#)
- **Livre** (id_l, titre_l, annee_l, resume_l)
- **Rediger** (id_a#, id_l#)

Légende :

x : relation

x : clef primaire

x# : clef étrangère

Comme vous pouvez le constater, le schéma de la base est déjà fait.
Les règles de passage au SQL sont assez simples :

Chaque **relation** devient une **table** ;

Chaque **attribut** de la relation devient une **colonne** de la table correspondante ;

Chaque **clef primaire** devient une **PRIMARY KEY** ;

Chaque **clef étrangère** devient une **FOREIGN KEY**.

```
CREATE TABLE Pays (  
    id_p INT NOT NULL,  
    nom_p VARCHAR(50),  
    PRIMARY KEY (id_p)  
);
```

```
CREATE TABLE Auteur (  
    id_a INT NOT NULL,  
    nom_a VARCHAR (30),  
    prenom_a VARCHAR (30),  
    date_naissance_a DATE,  
    id_p INT NOT NULL,  
    FOREIGN KEY (id_p) REFERENCES Pays(id_p),  
    PRIMARY KEY (id_a)  
);
```

```
CREATE TABLE Livre (  
    id_l INT NOT NULL,  
    titre_l VARCHAR (254),  
    annee_l VARCHAR (4),  
    resume_l TEXT,  
    PRIMARY KEY (id_l)  
);
```

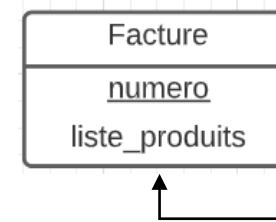
```
CREATE TABLE Rediger (  
    id_a INT NOT NULL,  
    id_l INT NOT NULL,  
    FOREIGN KEY (id_a) REFERENCES Auteur(id_a),  
    FOREIGN KEY (id_l) REFERENCES Livre (id_l),  
    PRIMARY KEY (id_a, id_l)  
);
```

Règles de vérification des niveaux de normalisation

1 ère Forme Normale (1FN) :

Toutes les entités et les association possèdent un identifiant

Aucune propriété n'est à valeurs **multiples** (propriétés atomiques)



Ici, « liste des produits » n'est pas atomique, c'est une liste

2 -ème Forme Normale (2FN) :

Le modèle est en 1FN

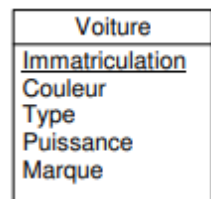
Toutes les DF entre les propriétés sont **élémentaires**

➡ Les propriétés d'une entité ne doivent dépendre que de l'identifiant de l'entité et non d'une partie de cet identifiant

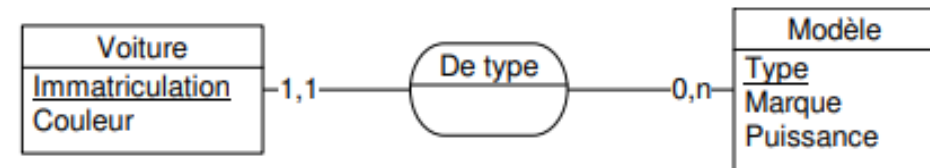
3 e Forme Normale (3FN)

Le modèle est en 2FN

Toutes les DF entre les propriétés sont **directes**



Type → Marque, Puissance alors que Type n'est pas un identifiant



NB : Pour que le MLD soit valide, il faut que chacune de ses relations soit au moins en 3FN.



D'autres Modèles de MERISE

Modèle Conceptuel de Traitement (MCT)

- Son objectif est la description de la transformation des informations.
- Se base sur plusieurs notions :
 - Activité : décrit perception globale du fonctionnement du système, et est, par le fait, complexe.
 - Traitement : décrit l'un des composants de l'activité du système.
 - Action : décrit une fonctionnalité atomique dans un traitement (consultation, mise à jour...).

Modèles Organisationnelles de Données et de Traitements (MOD, MOT)

- Concepts identiques à ceux du MCD et MCT sauf que ...
 - l'intégration de notions supplémentaires, comme
 - les lieux (où ?),
 - Les personnes (qui ?),
 - Les ressources (comment ?)
- =>les contraintes spatiales et temporelles,
- imposent que,
 - la redondance de données soit tolérée et que les traitements soient raffinés.

Modèles Opérationnels : Logique et Physique

Le modèle logique représente un choix logiciel pour le système d'information.

Le modèle physique reflète un choix matériel pour le système d'information.

Etude de cas : Forum de discussion

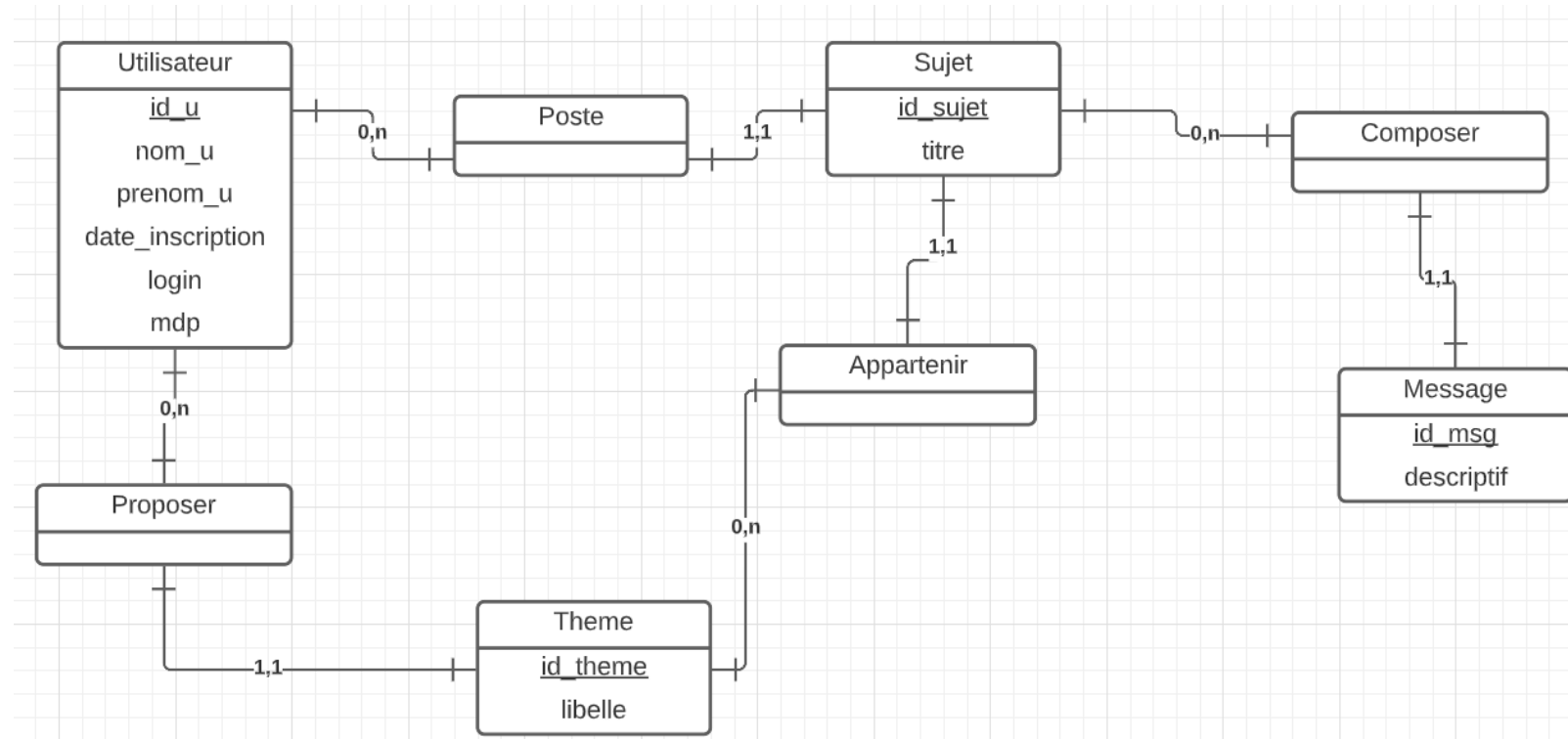
Dans notre étude de cas on abordera le sujet du Forum d'une façon générale. Le Forum contient une liste des sujets créés par le propriétaire du site ou proposés par des participants membres du Forum. Pour devenir membre du Forum, l'internaute doit s'inscrire et par la suite participer aux différents sujets proposés dans le Forum.

- Le Forum comporte une liste des thèmes
- Les thèmes sont proposés par les participants
- Pour participer au Forum, il faut s'inscrire
- Un membre peut participer au forum en sélectionnant un thème
- Pour un thème donné, le participant peut poser des questions, voir ou inclure ses propres réponses.

1. Identifier les règles de gestion
2. Etablir le dictionnaire de données
3. Réaliser un MCD
4. Trouver MLD

Etude de cas : Forum de discussion

MCD



MLD

Utilisateur (id_u, nom_u, prenom_u, date_inscription, login, mdp)

Sujet (id_sujet, titre, id_u#, id_theme#)

Message (id_msg, descriptif, id_sujet#)

Theme (id_theme, libelle, id_u#)

Etude de cas : Institut de formation

Les cours sont organisés en modules, chaque module est caractérisé par un numéro de module, un intitulé, une durée en heures et un type.

Les étudiants suivent des enseignements portant sur plusieurs modules. Chaque étudiant est caractérisé par un numéro d'inscription unique, un nom, un prénom et une adresse et une date de naissance, un étudiant est évalué trois fois pour chaque module et possède une note de fin de module.

Chaque étudiant appartient à un groupe caractérisé par un code, une spécialité et le nombre d'étudiants qu'il comporte.

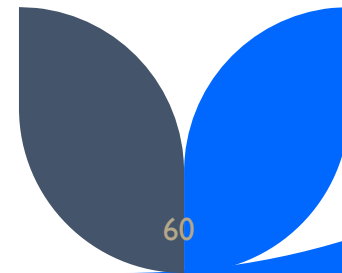
Un enseignant intervient dans un module pour un groupe donné à une date donnée, chaque enseignant est caractérisé par un code, un nom, un prénom et une adresse.

Un enseignant intervient habituellement dans plusieurs modules.

On désire aussi mémoriser le nombre d'heures effectué par chaque enseignant dans un module donné pour un groupe donné.

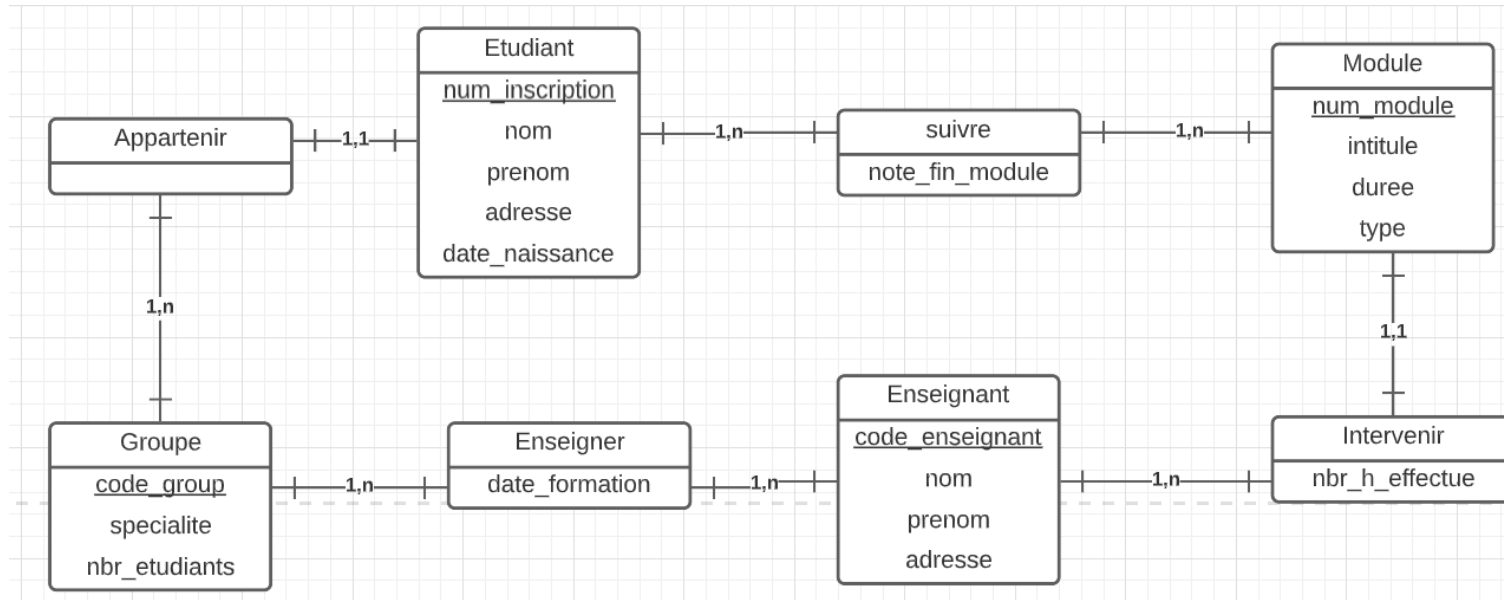
T.A.F :

1. Établir le dictionnaire de données.
2. Identifier les règles de gestion.
3. Réaliser un MCD.
4. Trouver MLD.



Etude de cas : Institut de formation

MCD



MLD

Etudiant (num_inscription, nom, prenom, adresse, date_naissance, code_group#)

EnseignantModule (code_enseignant#, num_module#, nbr_h_effectue)

Groupe (code_group, specialite, nbr_etudiants)

Enseignant (code_enseignant, nom, prenom, adresse)

Module (num_module, intitule, duree, type)

Suivre(num_inscription#, num_module#, note_fin_module)

Enseigner(code_group#, code_enseignant#, date_formation)

Limites MERISE

- ❖ Sépare les données des traitements
- ❖ Ne permet pas de modéliser des programmes orientées objet

Approche Orientée Objets

« Objet »

Système = une collection **d'objets** dissociés, identifiés et possédant des caractéristique

Un Objet =

- Identité
- Etat (Attributs)
- Comportement (Méthodes)

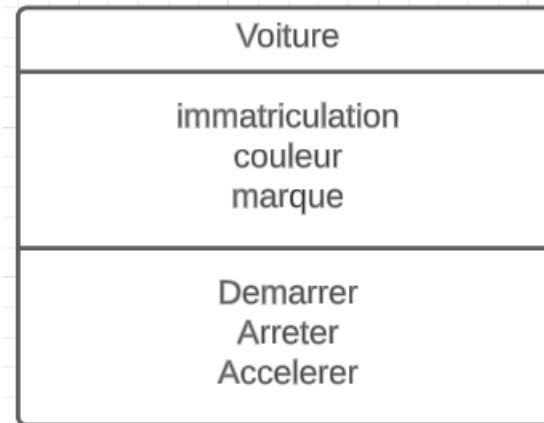
Toto:Voiture
Immatriculation= 995 Couleur = grise marque = Mercedes
Démarrer Arrêter Accélérer

→ Rapprocher données et traitements

Approche Orientée Objets

« Classe »

Classe : un type de données abstrait caractérisée par des propriétés (attributs et méthodes) communes à des objets





UML : Unified Modeling Language

UML - Définition

UML n'est pas une méthode de conception, C'est un langage de modélisation orientée objet.

Langage :

- Syntaxe et règles d'écriture
- Notations graphiques normalisées

De Modélisation

- Abstraction
- Spécification et conception

Unifié :

- Fusion de plusieurs notations antérieures
- Standard défini par OMG

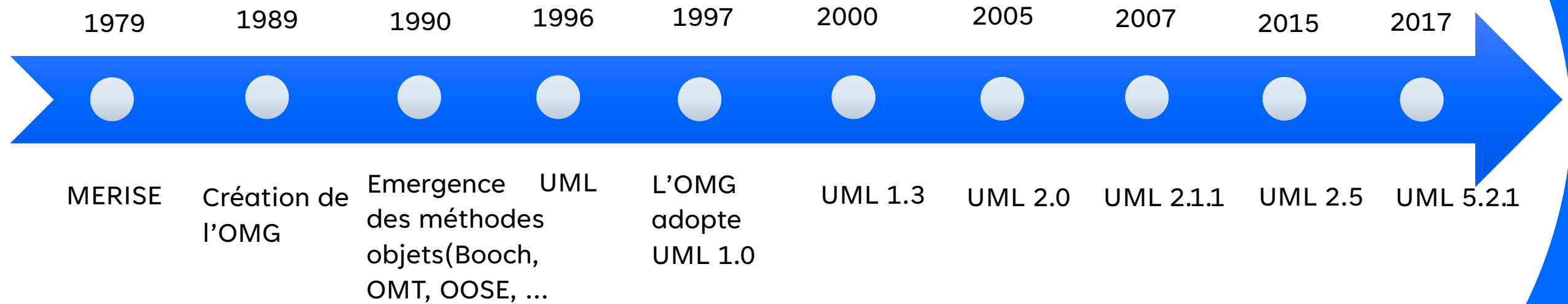
OMG : Consortium d'entreprises créé en 1989, à but non lucratif, afin de promouvoir la technologie objet dans le développement du logiciel,

UML - Objectif

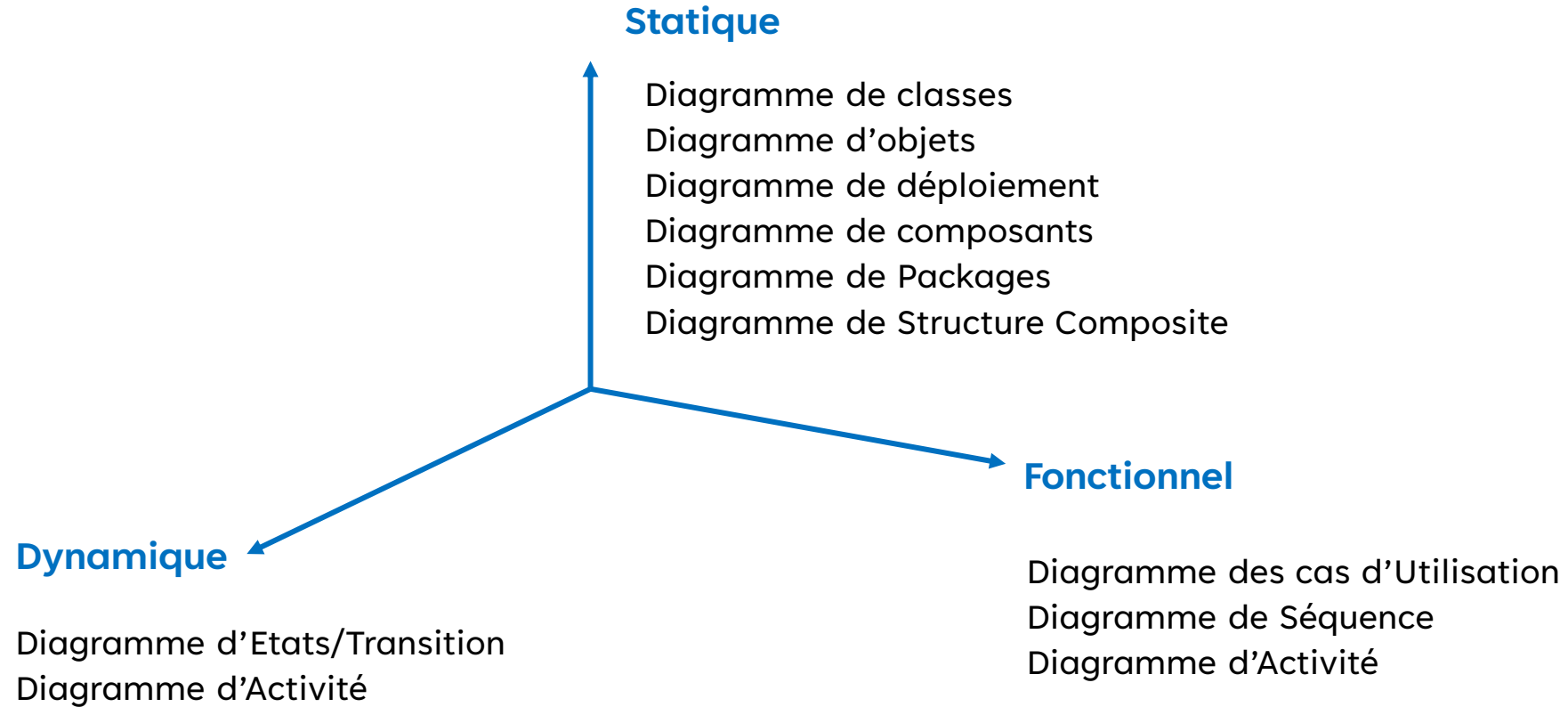
UML permet de :

- ✓ **Visualiser** le système
- ✓ **Spécifier** la structure et la comportement du système
- ✓ **Aider** à la construction du système
- ✓ **Documenter** les décisions

UML - Histoire



UML – Axes de modélisation





UML

Diagramme des cas d'utilisation

Diagramme des cas d'utilisation

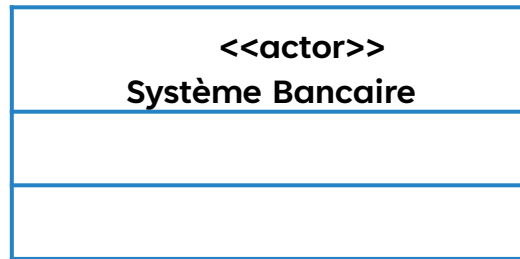
Définition

- Capturer les exigences fonctionnelles d'un système
- Décrire les interactions typiques entre les utilisateurs d'un système et le système lui-même

Diagramme des cas d'utilisation

Acteur

Acteur est l'abstraction d'un rôle joué par des entités externes (**utilisateur, dispositif matériel ou autre système**) qui interagissent directement avec le système étudié.



Remarques :

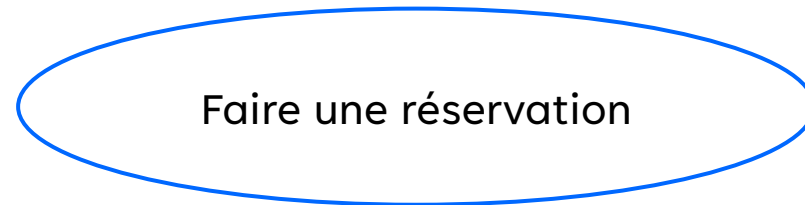
- Une même entité externe concrète peut jouer successivement différents rôles par rapport au système étudié
 - ➔ modélisée par plusieurs acteurs
- Un même rôle peut être joué simultanément par plusieurs entités externes concrètes
 - ➔ modélisée par le même acteur

Diagramme des cas d'utilisation

Cas d'utilisation

Un cas d'utilisation est un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier.

Exemple :



(^_^) L'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système

Diagramme des cas d'utilisation

Exemple de Système de location des voitures

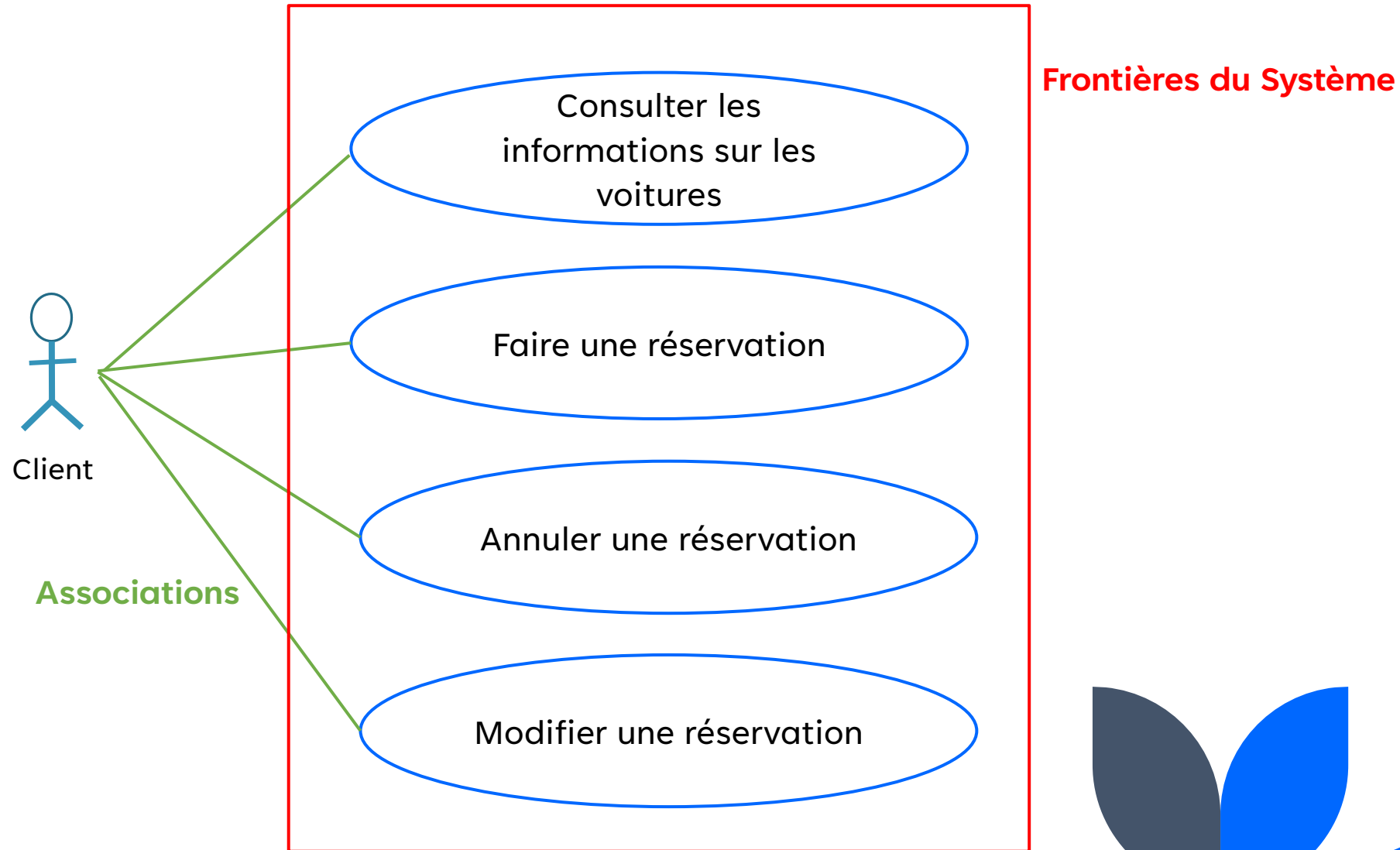


Diagramme des cas d'utilisation

Relations

On dispose deux types de relation :

1. Relation entre les cas d'utilisation

- Inclusion
- Extension
- Généralisation/Spécialisation

2. Relation entre les acteurs

- Généralisation/Spécialisation

Diagramme des cas d'utilisation

Relation entre les cas d'utilisation - Inclusion

Un cas d'utilisation A **contient** le comportement défini dans un autre cas d'utilisation B



On dit que B est une partie essentielle de A.

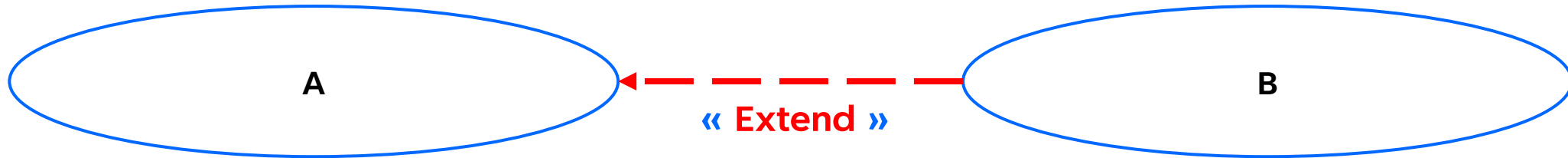
Exemple :



Diagramme des cas d'utilisation

Relation entre les cas d'utilisation - Extension

L'instance d'un cas d'utilisation **A** peut être augmentée avec un comportement quelconque défini dans un cas d'utilisation étendu **B**



On dit que B est une partie optionnelle de A

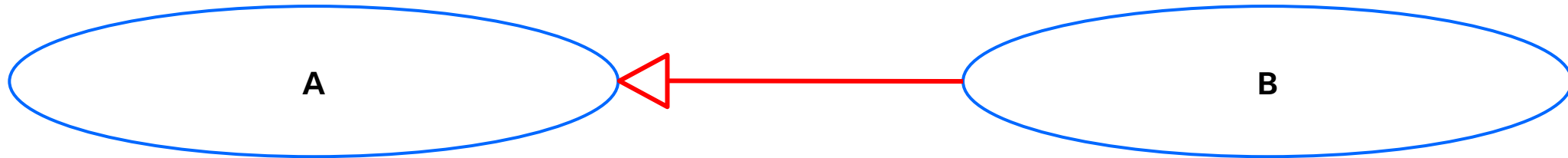
Exemple :



Diagramme des cas d'utilisation

Relation entre les cas d'utilisation – Généralisation/Spécialisation

Un cas d'utilisation **B** est plus spécifique qu'un cas d'utilisation général **A**



On dit que B est une spécialisation de A et A est une généralisation de B

Exemple :

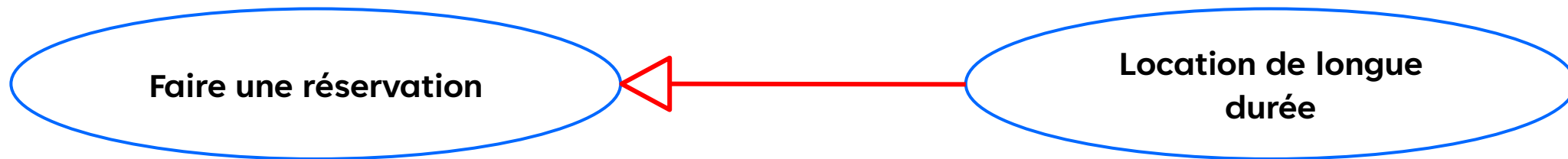


Diagramme des cas d'utilisation

Distributeurs de boissons chaudes et machines à café

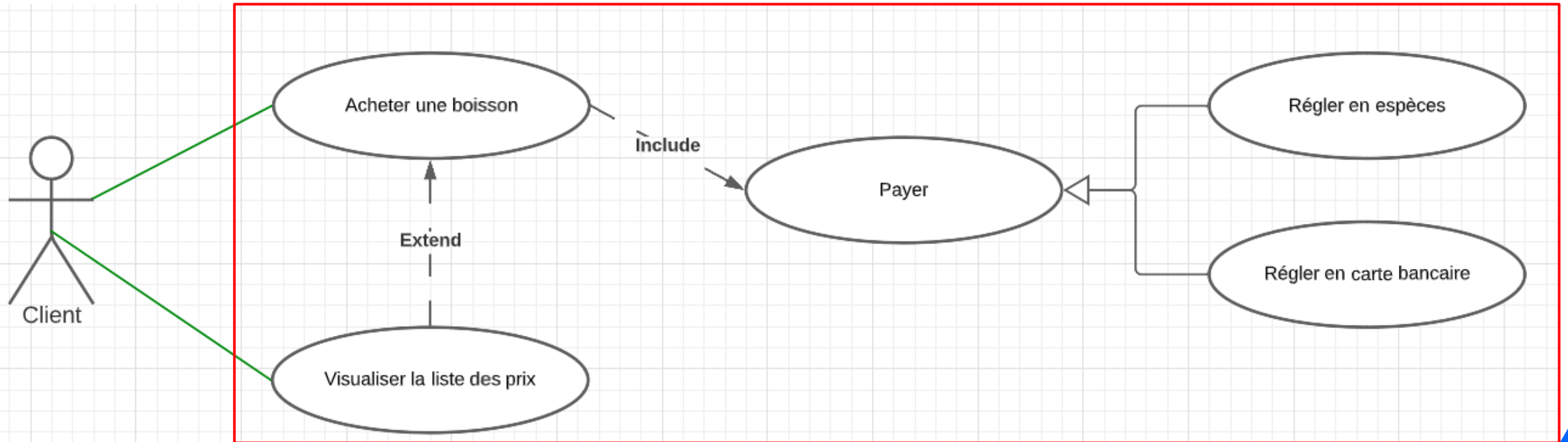


Diagramme des cas d'utilisation

Relation entre les acteurs – Généralisation/Spécialisation

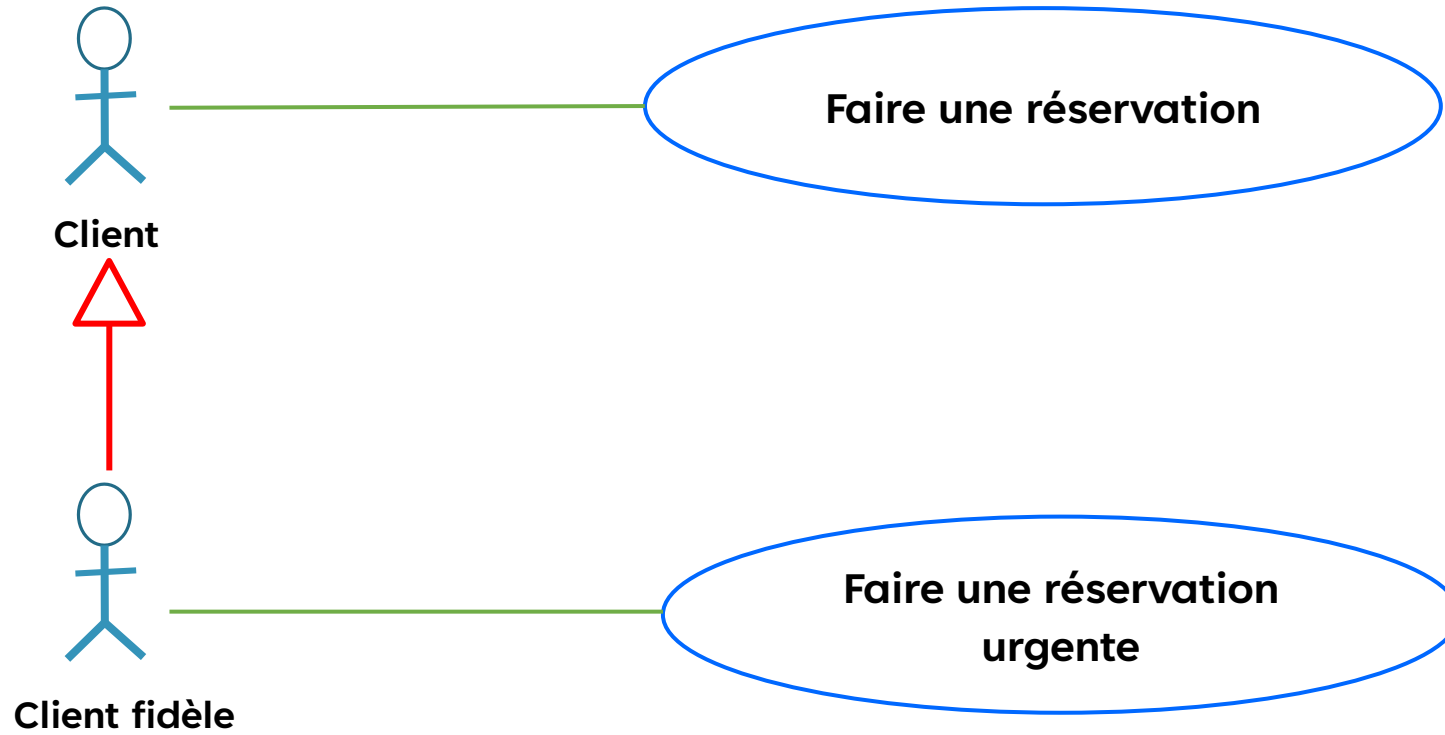


Diagramme des cas d'utilisation

Exercice - Entreprise

Elaborer le diagramme des cas d'utilisation du système suivant :

- ❑ Tous les personnels de l'entreprise peuvent consulter le système. Toute consultation doit être précédée d'une authentification légère dans laquelle la personne précise son nom et son service à des fins de statistiques ultérieures.
- ❑ Les ingénieurs peuvent effectuer différentes opérations de mise à jour pour les produits dont ils sont responsables : ajout, retrait, modification des informations sur les produits. Ces opérations doivent être précédées d'une authentification plus approfondie lors de laquelle l'ingénieur précise son nom, son service et donne un mot de passe qui est vérifié en contactant le système de gestion des personnels.
- ❑ Toutes les opérations (consultation et mise à jour) peuvent optionnellement s'accompagner d'une impression des documents accédés

Diagramme des cas d'utilisation

Solution - Entreprise

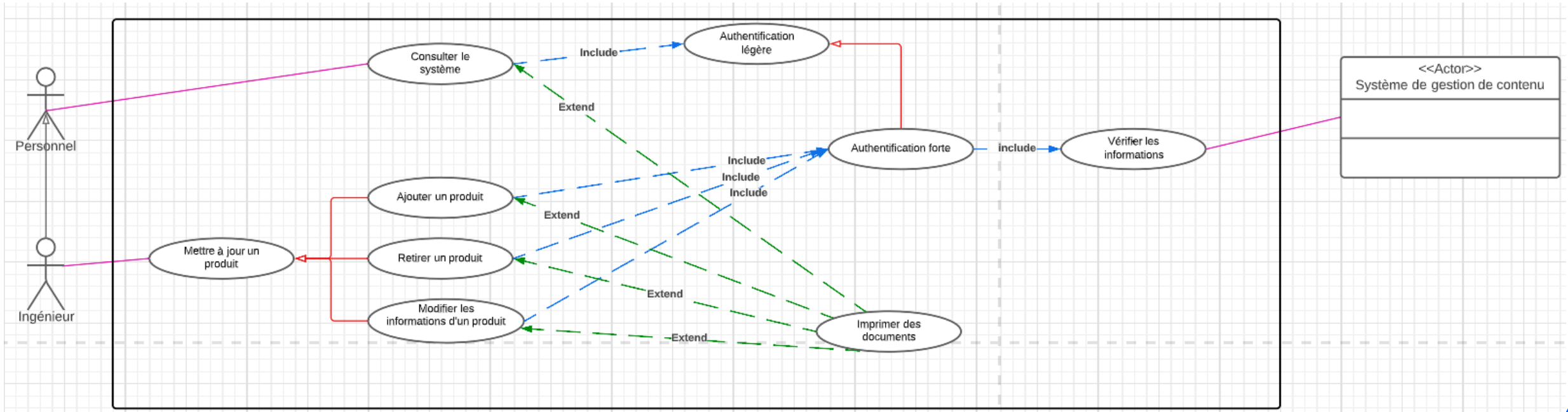


Diagramme des cas d'utilisation

Exercice – Système gestion de stock d'articles

Dans un magasin, un commerçant dispose d'un système de gestion de son stock d'articles, dont les fonctionnalités sont les suivantes :

- Edition de la fiche d'un fournisseur
- Possibilité d'ajouter un nouvel article (dans ce cas, la fiche fournisseur est automatiquement éditée. Si le fournisseur n'existe pas, on peut alors le créer)
- Edition de l'inventaire. Depuis cet écran, on a le choix d'imprimer l'inventaire, d'effacer un article ou d'éditer la fiche d'un article).

Modéliser cette situation par un diagramme de cas d'utilisation

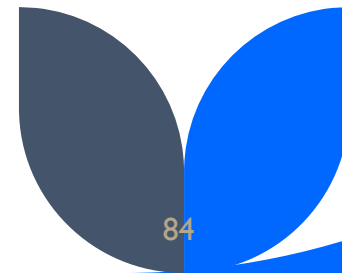


Diagramme des cas d'utilisation

Acteur principal / Secondaire

- **Acteur principal** tire réellement bénéfice du cas d'utilisation. Généralement, c'est lui le déclencheur du cas d'utilisation.
- **Acteur Secondaire** sollicité par le système pour obtenir des informations complémentaires.

Exemple :

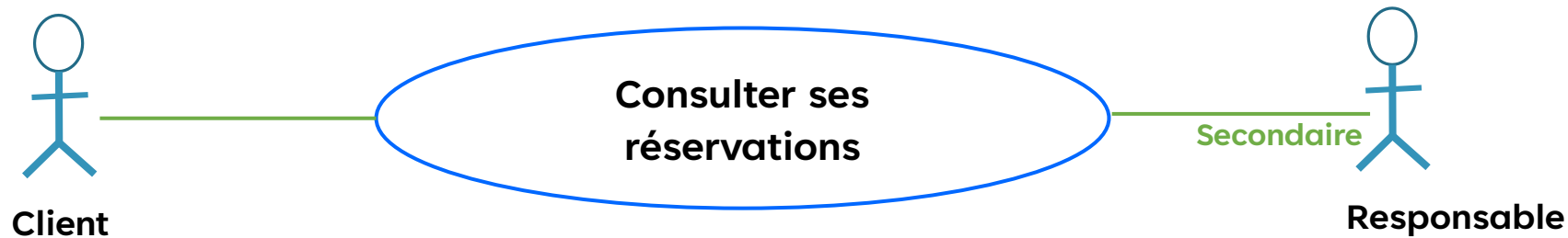


Diagramme des cas d'utilisation

Scénarios d'un cas d'utilisation

- **Un cas d'utilisation** est un ensemble de séquences d'interactions entre le **système** et un **acteur**
- **Un scénario** est une **séquence d'étapes** décrivant une **interaction** entre un acteur et le système
 - ➡ Il représente une **succession particulière d'enchaînements**, qui s'exécutent **du début à la fin du cas d'utilisation**

Les différents types de scénarios sont :

Le scénario nominal : correspond au fonctionnement «**normal**» du cas d'utilisation

Les autres scénarios sont des cas particuliers :

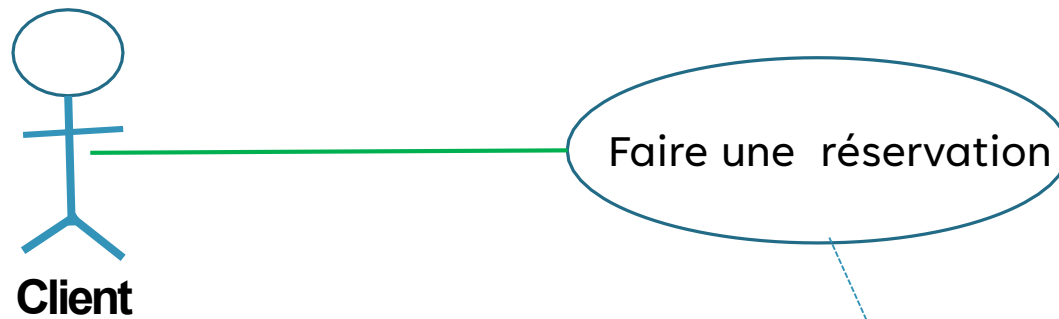
Les scénarios alternatifs conduisent à un retour à une étape du cas nominal

Les scénarios d'exceptions conduisent à la fin du cas d'utilisation

Diagramme des cas d'utilisation

Scénarios d'un cas d'utilisation - Exemple

Exemple : Créer les différents scénarios possibles pour le cas d'utilisation « Faire une réservation »



- Saisir la période désirée
- Choisir une voiture parmi les voitures disponible
- Consulter le prix estimé

Diagramme des cas d'utilisation

Scénario normal du cas d'utilisation <Faire une réservation>

Le scénario nominal est le suivant :

1. Le client choisit de faire une réservation
2. Le système affiche l'interface permettant de faire une réservation
3. Le client insère la période désirée pour la location
4. Le système récupère à partir du système de gestion de parc automobile les voitures disponibles durant cette période
5. Le système affiche la liste des voitures disponibles
6. Le client sélectionne la voiture désirée
7. Le système affiche le prix estimé de la location
8. Le client confirme la réservation

Diagramme des cas d'utilisation

Scénarios alternatifs du cas d'utilisation <Faire une réservation>

Les enchaînements alternatifs sont les suivants :

- En (3)** Si le client insère une date de début de location inférieure à J+1, le système l'invite à choisir une autre date
- En (4)** Si aucune voiture n'est disponible durant la période voulue, le système invite le client à choisir une autre période
- En (5)** Si le client n'est pas satisfait des voitures disponibles, il peut choisir une autre période ou sortir sans confirmer sans terminer la demande
- En (7)** Si le client n'est pas satisfait du prix de la location, il peut choisir une autre voiture ou sortir sans confirmer la réservation

Diagramme des cas d'utilisation

Description textuelle d'un cas d'utilisation

Pour chaque cas d'utilisation , il faut définir :

- Un **sommaire d'identification** (obligatoire) est composé de :
titre, but, résumé, dates, version, responsable, acteurs...
- Une **description des enchaînements** (obligatoire) est composé de :
le scénario nominal, les enchaînements alternatifs, les enchaînements d'exception, les préconditions, et les postconditions
- Les **besoins d'IHM** (optionnel)
- Les **exigences non fonctionnelles** (optionnel)
disponibilité, fiabilité, intégrité, confidentialité, performances, ...

Diagramme des cas d'utilisation

Description textuelle du cas d'utilisation <Faire une réservation>

Sommaire d'identification

Titre: Faire une réservation

But : Détailler les étapes permettant à un client de faire une réservation de voiture

Acteurs : Client, Système de gestion du parc automobile (secondaire)

Date : DD/MM/YYYY

Responsables : X

Version : 1.0

Description des enchaînements

Préconditions

Le client est authentifié

Scénario nominal :

1. Le client choisit de faire une réservation
2. Le système affiche l'interface permettant de faire une réservation
3. Le client insère la période désirée pour la location
4. Le système récupère à partir du système de gestion de parc automobile les voitures disponibles durant cette période
5. Le système affiche la liste des voitures disponibles
6. Le client sélectionne la voiture désirée
7. Le système affiche le prix estimé de la location
8. Le client confirme la réservation

Enchaînements alternatifs :

En (3) Si le client insère une date de début de location inférieure à J+1, le système l'invite à choisir une autre date

En (4) Si aucune voiture n'est disponible durant la période voulue, le système invite le client à choisir une autre période

En (5) Si le client n'est pas satisfait des voitures disponibles, il peut choisir une autre période ou sortir sans confirmer sans terminer la demande

En (8) Si le client n'est pas satisfait du prix de la location, il peut choisir une autre voiture ou sortir sans confirmer la réservation

Postconditions

La demande de réservation est envoyée au responsable

Besoins d'IHM

Le client doit être guidée lors de sa première utilisation

Exigences non fonctionnelles

Confidentialité : les informations concernant le client ne doivent pas être divulgués



UML

Diagramme de séquence

Diagramme de séquence

Définition

Pour un cas d'utilisation donné, la séquence des échanges entre l'acteur et le système peut être présentée par :

- ✓ **Une Fiche de description textuelle du cas d'utilisation**
- ✓ **Un Diagramme de Séquence**

Diagramme de séquence Présentation

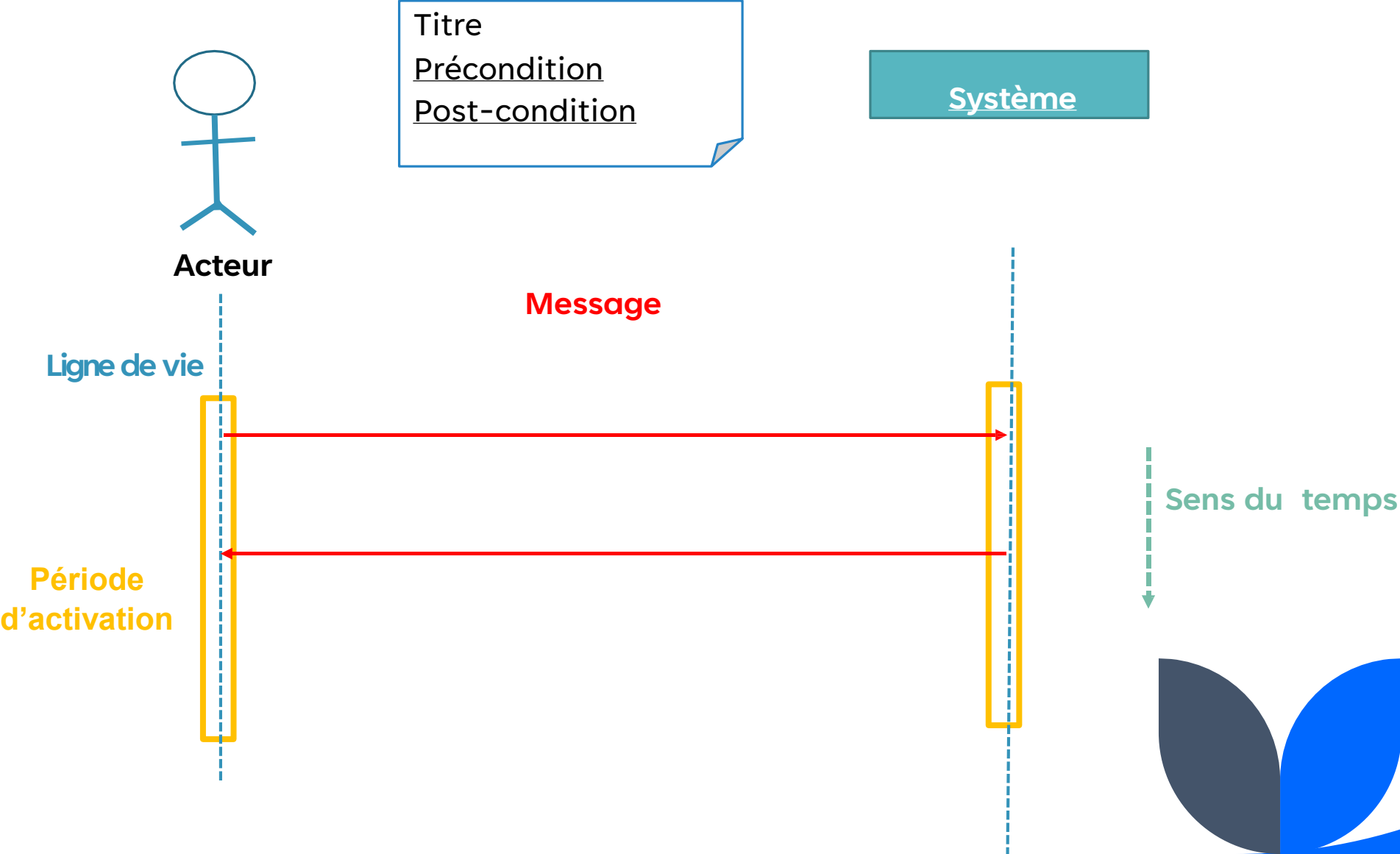


Diagramme de séquence

Cas d'utilisation <Enregistrer sous>

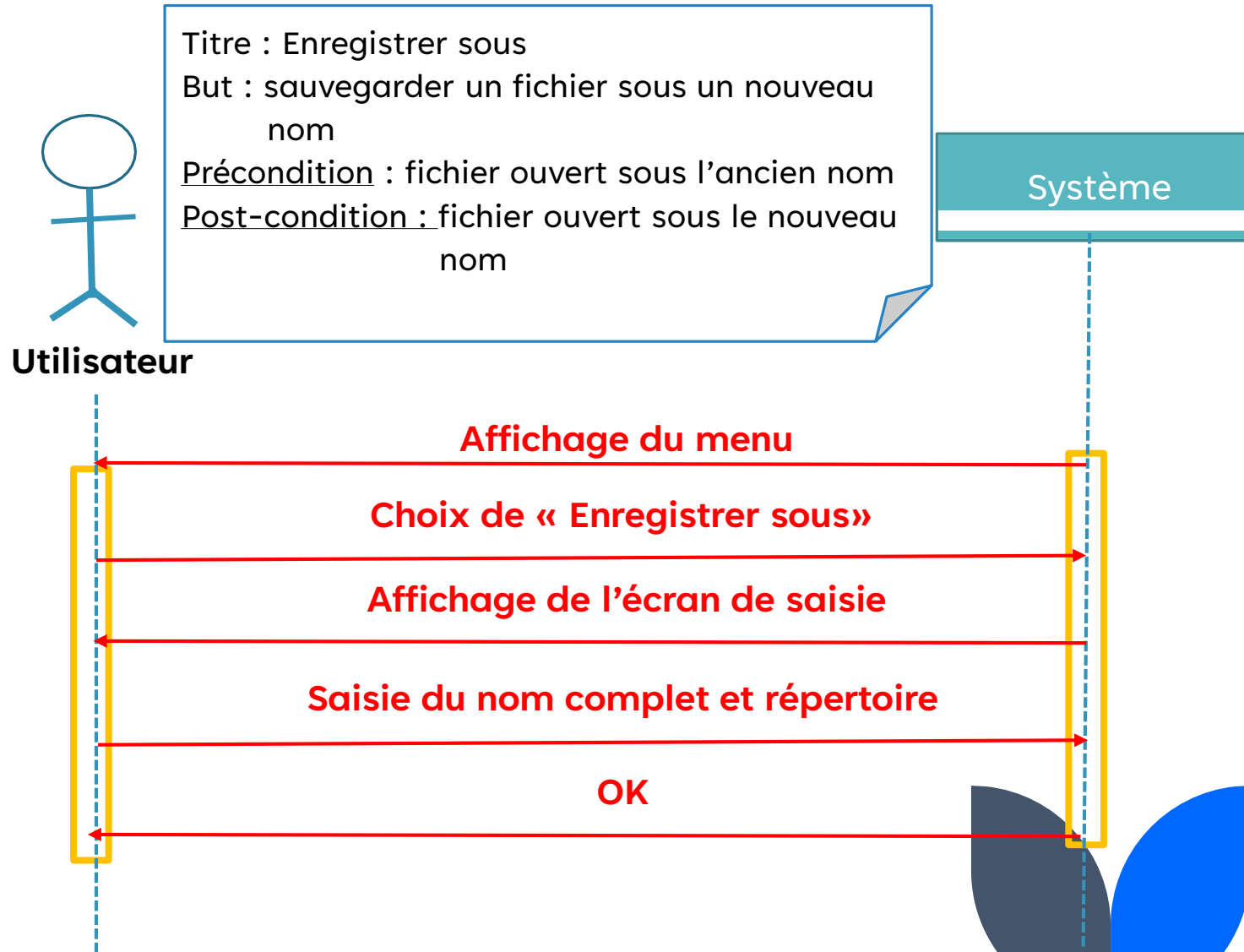


Diagramme de séquence

Cas d'utilisation <Faire une réservation>

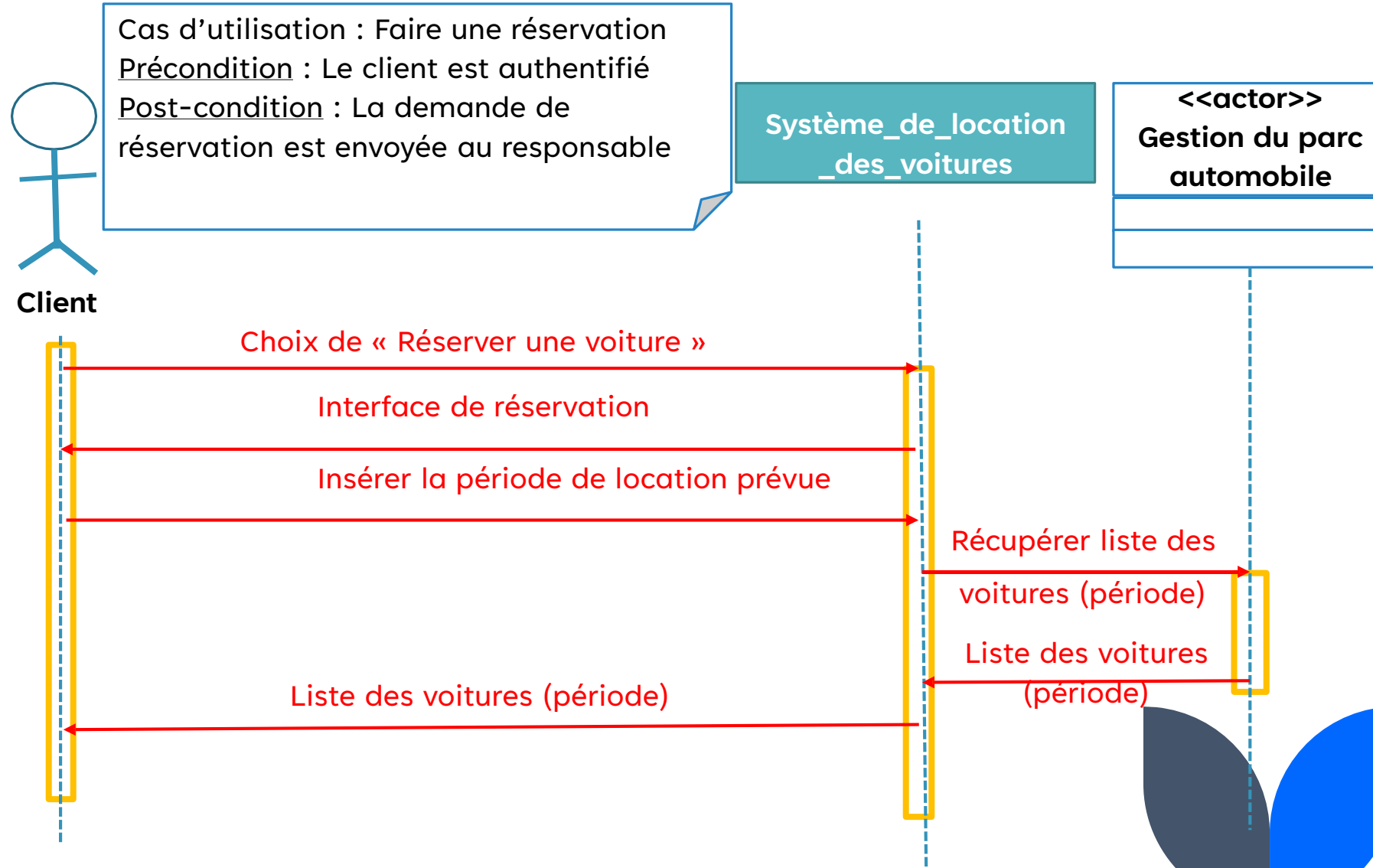


Diagramme de séquence

Communication entre sous-systèmes

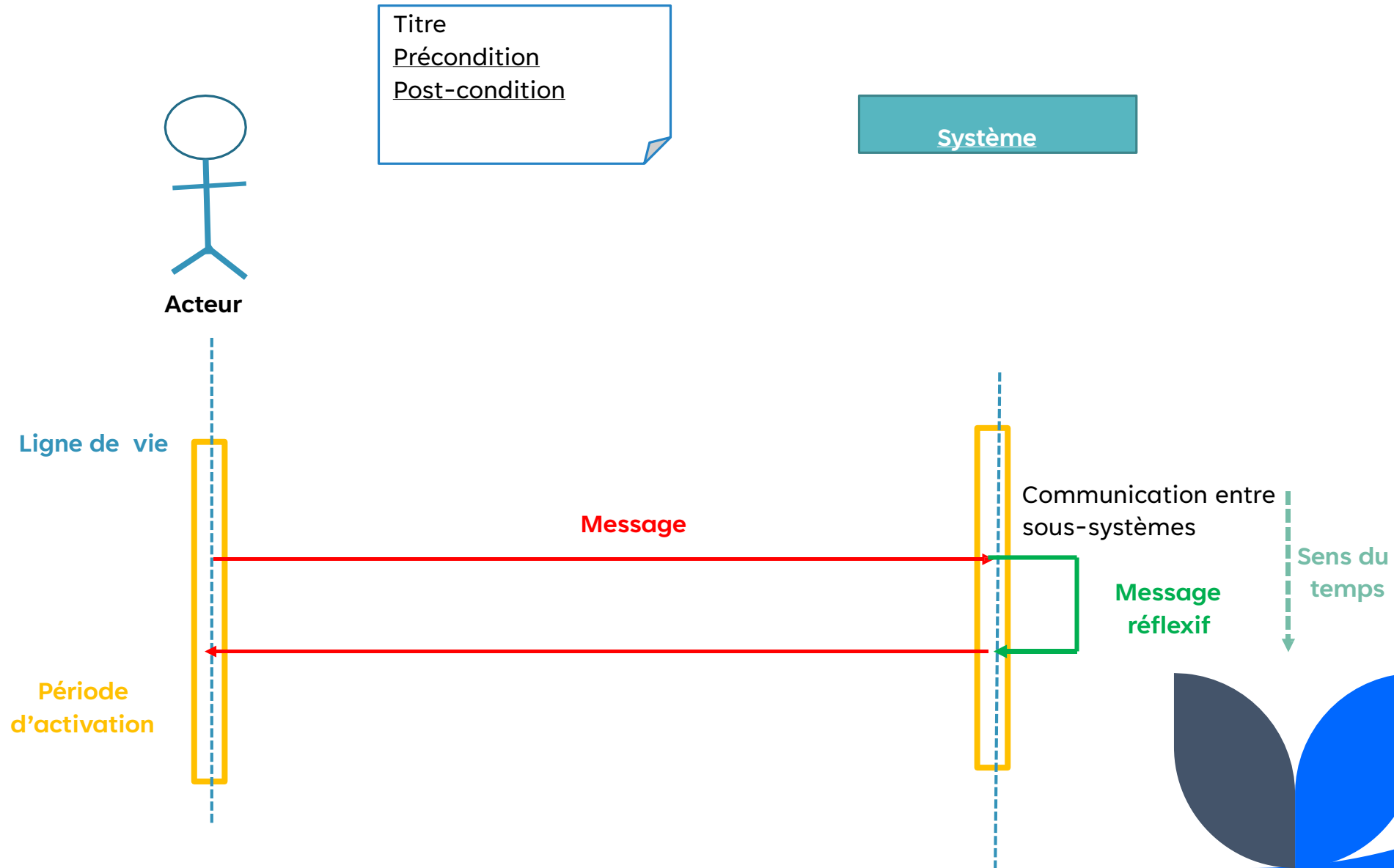


Diagramme de séquence
Communication entre sous-systèmes
Cas d'utilisation <Faire une réservation>

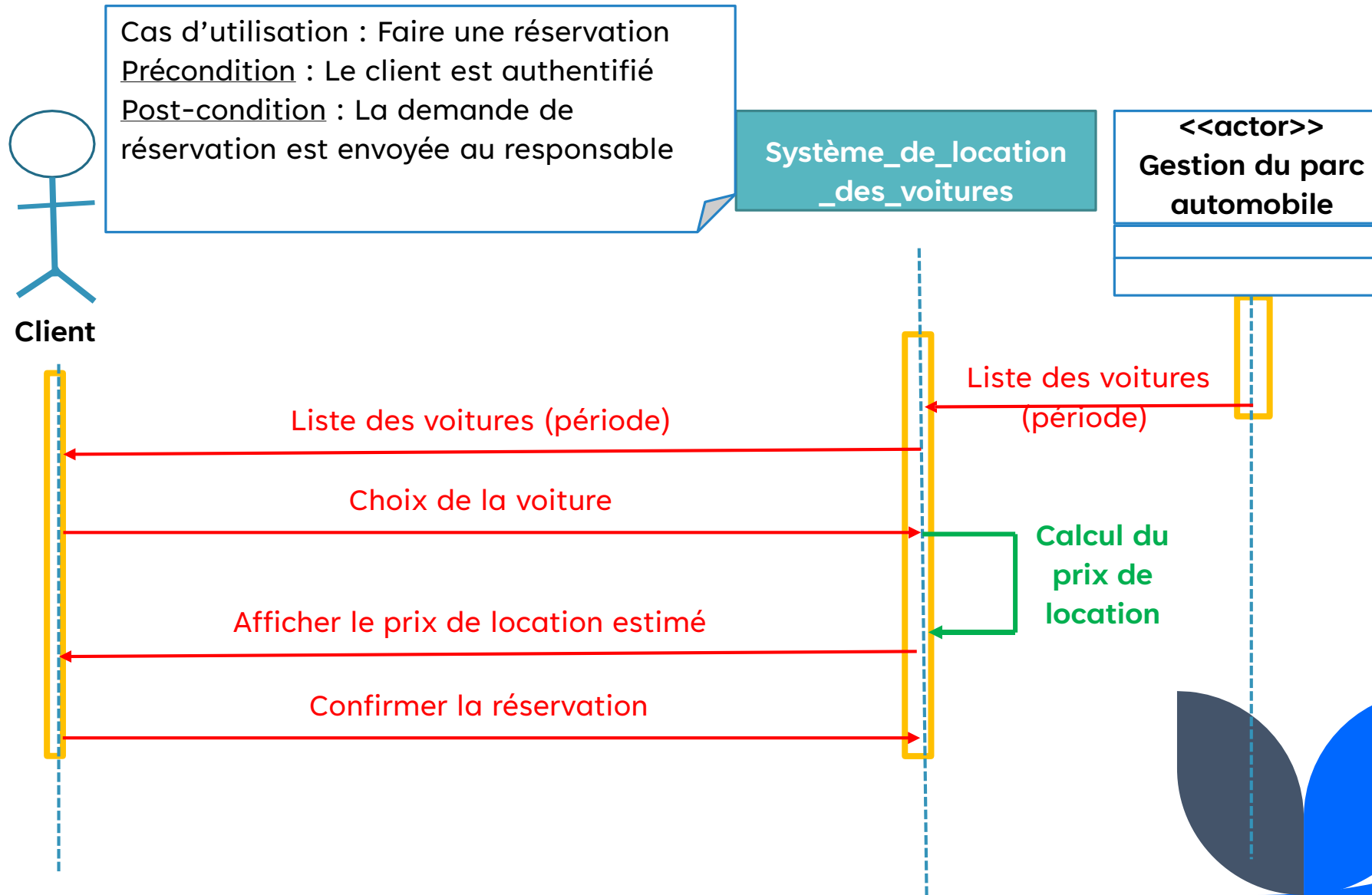
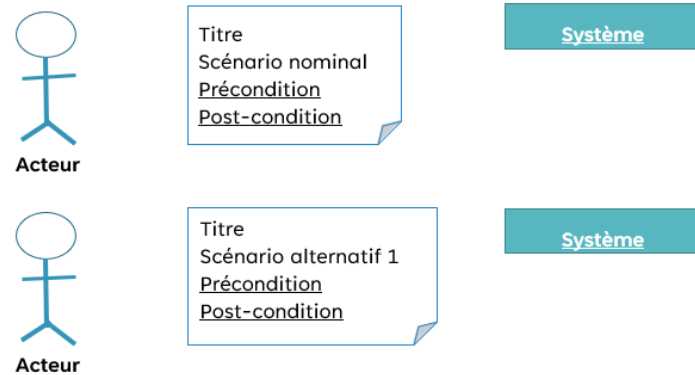


Diagramme de séquence

Comment représenter les différents scénarios d'un cas d'utilisation?

Nous disposons deux méthodes pour représenter les scénarios d'un cas d'utilisation :

Méthode 1 : Diagramme de séquence pour chaque scénario



Méthode 2 : Utiliser le concept des **fragments combinés** :

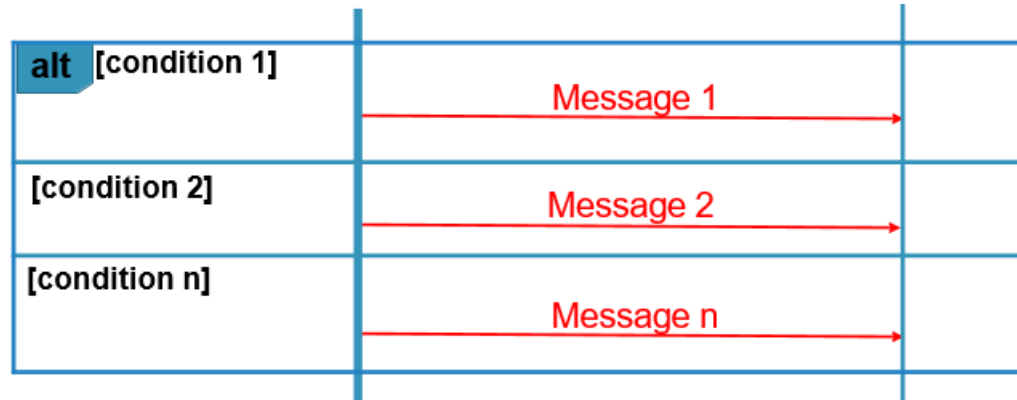
➡ décomposer une interaction complexe en fragments suffisamment simples pour être compris

- a. Alternative « **alt** »
- b. Optionnel « **opt** »
- c. Parallèle « **par** »
- d. Boucle « **loop** »
- e. Référence « **ref** »

Diagramme de séquence

Méthode 2 : fragments combinés « alt »

Représentation de « alt »



Exemple : Si aucune voiture n'est disponible durant la période voulue, le système invite le client à choisir une autre période

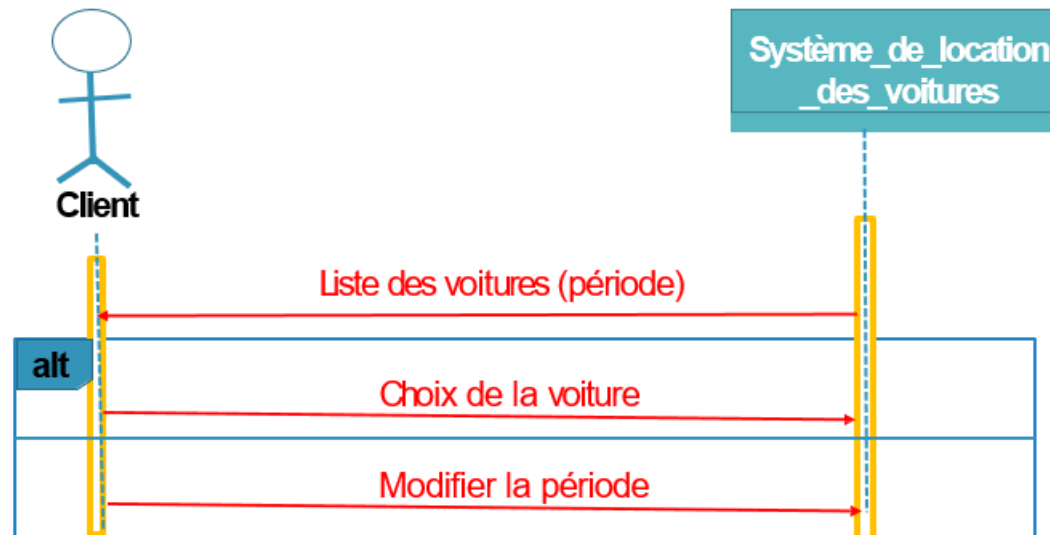


Diagramme de séquence

Méthode 2 : fragments combinés

« opt »

Représentation de « opt »



Exemple : Si le client insère une date de début de location inférieure à J+1, le système l'invite à choisir une autre date

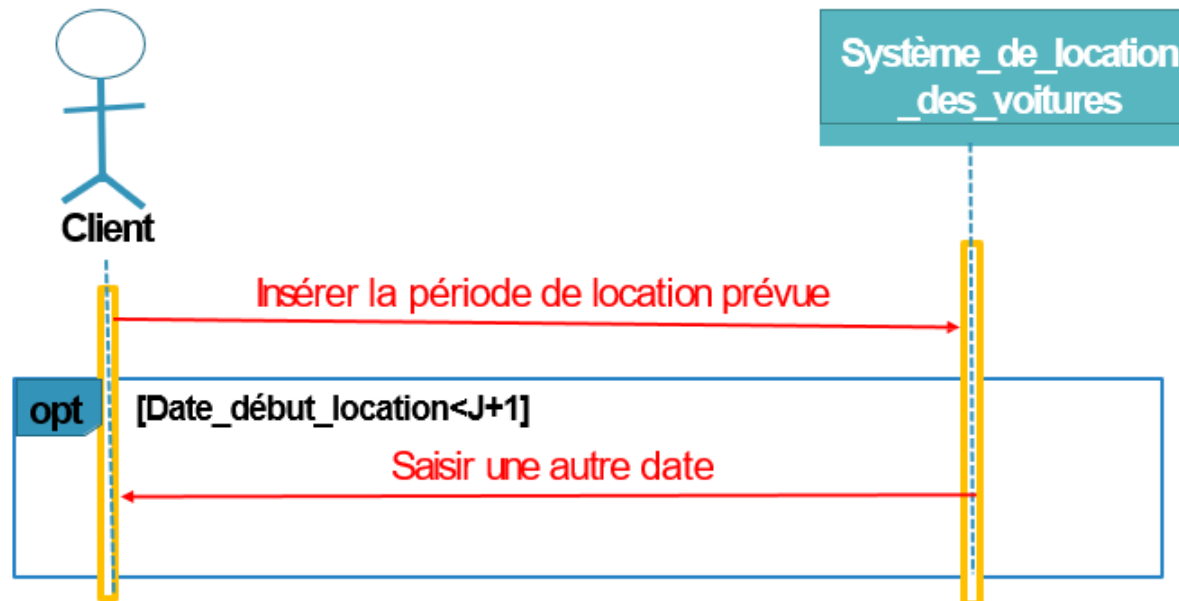


Diagramme de séquence

Méthode 2 : fragments combinés

« par »

Représentation de « par »

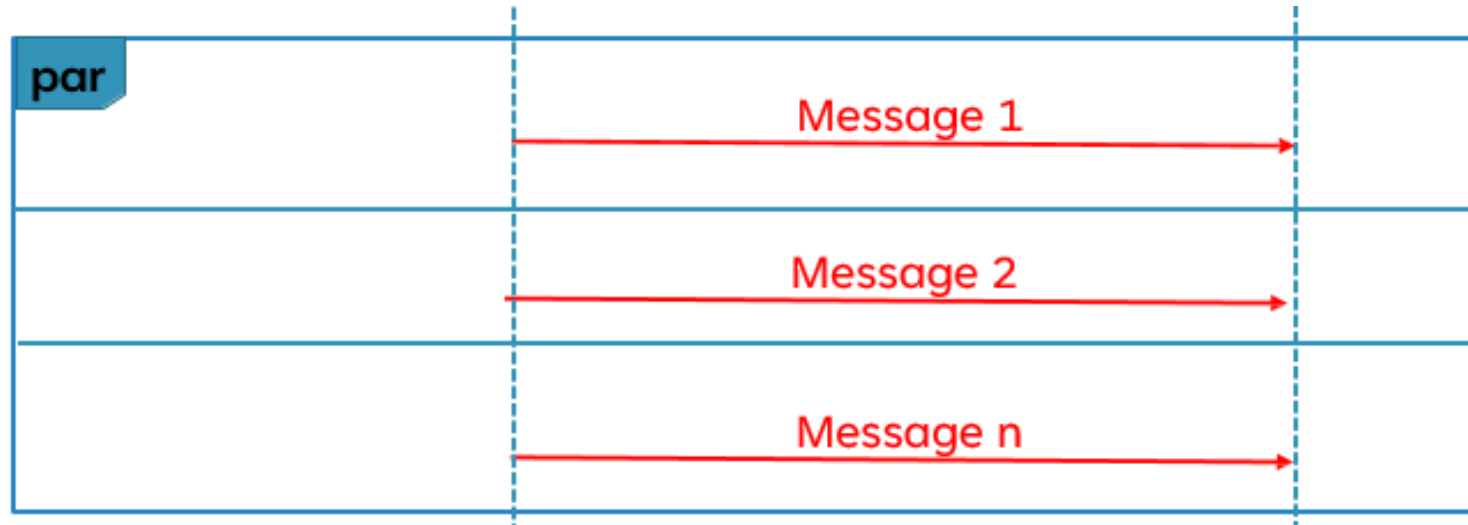
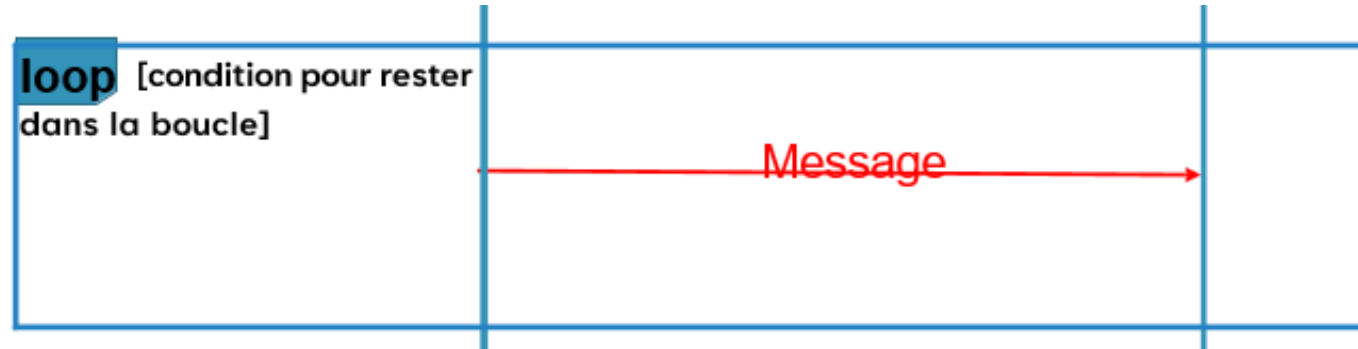


Diagramme de séquence

Méthode 2 : fragments combinés « loop »

Représentation de « loop »



Exemple : Si le client insère une date de début de location inférieure à J+1, le système l'invite à choisir une autre date

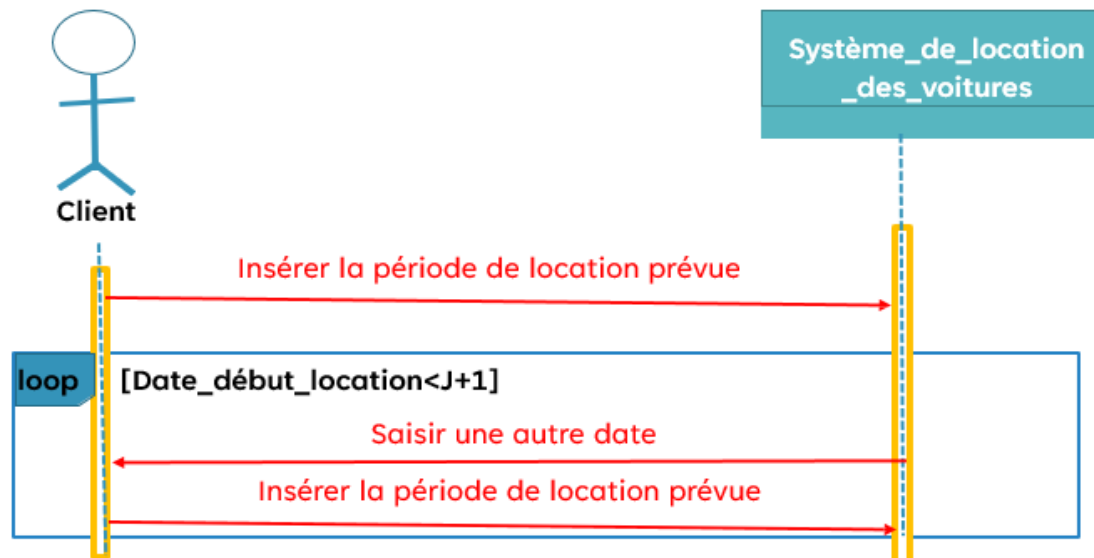


Diagramme de séquence

Méthode 2 : fragments combinés

« ref »

Pour faire une réservation, il faut d'abord s'authentifier



Diagramme de séquence avec le fragment ref

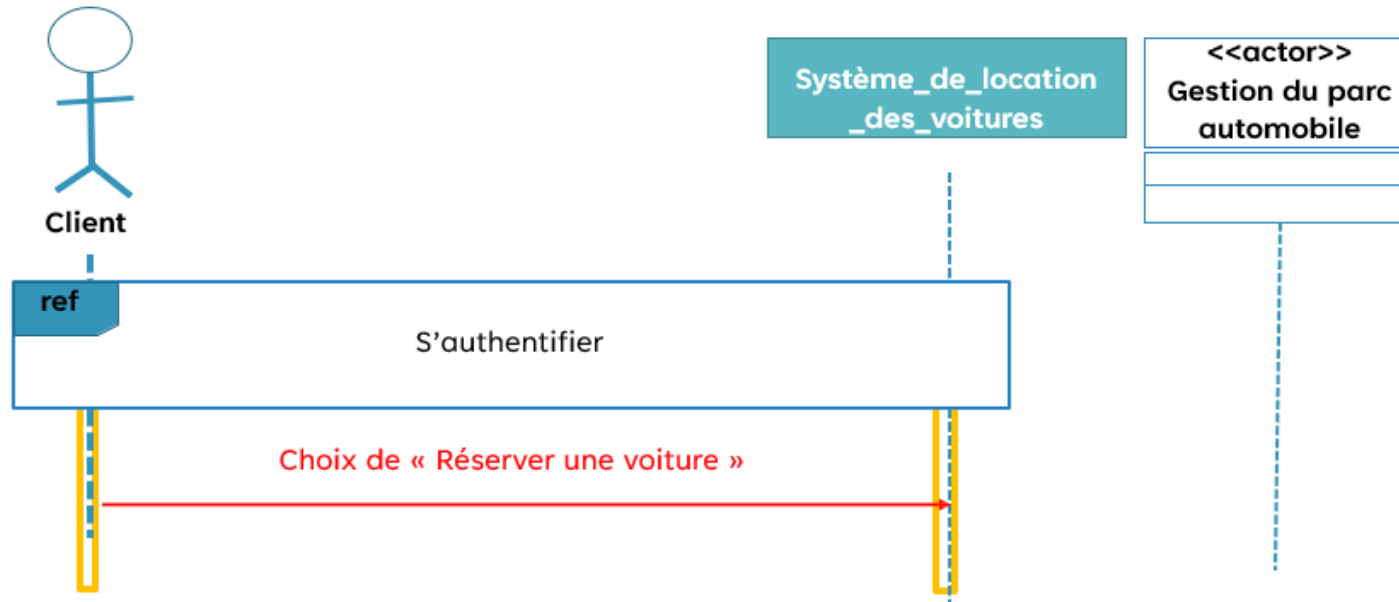


Diagramme de séquence

Exercice – Caisse d'un supermarché

Le déroulement normal d'utilisation d'une caisse de supermarché est le suivant :

Un client arrive à la caisse avec ses articles à payer

Le caissier enregistre le numéro d'identification de chaque article, ainsi que la quantité si elle est supérieure à 1

La caisse affiche le prix de chaque article et son libellé

Lorsque tous les achats sont enregistrés, le caissier signale la fin de la vente

La caisse affiche le total des achats

Le caissier annonce au client le montant total à payer

Le client choisit son mode de paiement

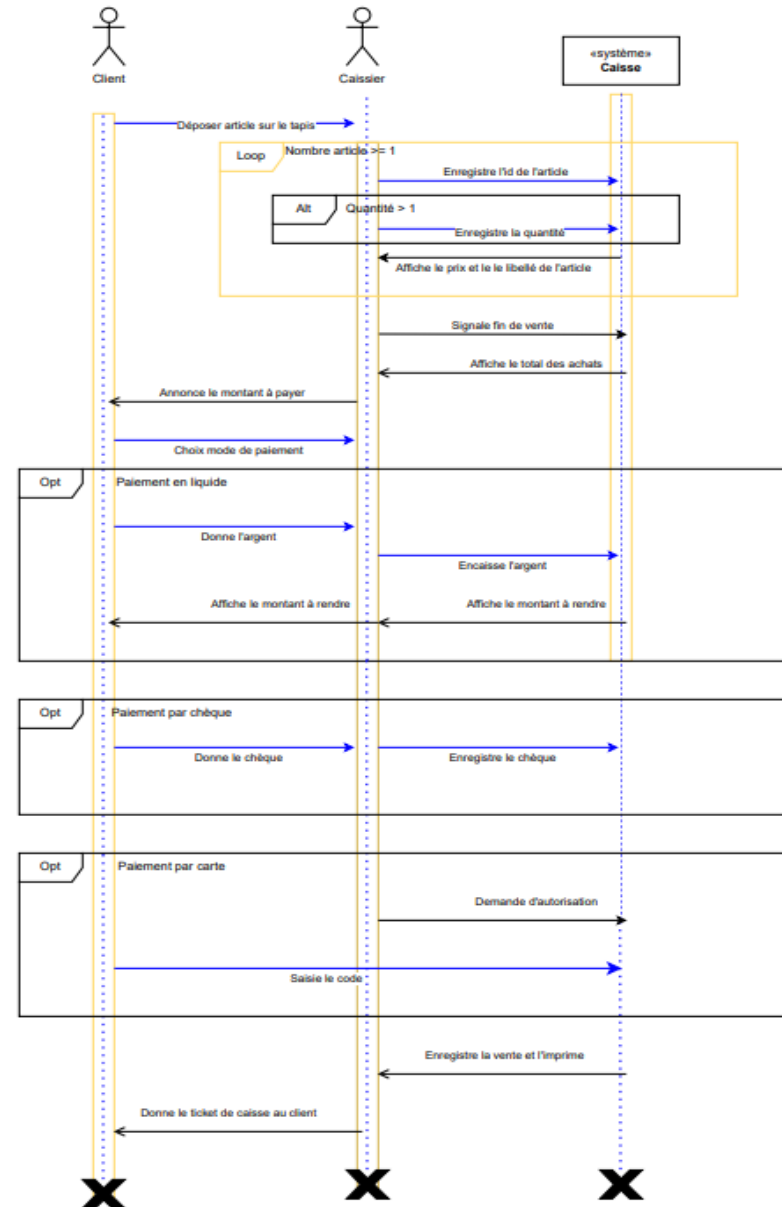
- ❖ Liquide : le caissier encaisse l'argent, la caisse indique le montant à rendre au client
- ❖ Chèque : le caissier note le numéro de pièce d'identité du client
- ❖ Carte de crédit : la demande d'autorisation est envoyée avant la saisie

La caisse enregistre la vente et l'imprime

Le caissier donne le ticket de caisse au client

Modéliser cette situation à l'aide d'un diagramme de séquence en ne prenant en compte que le cas du paiement en liquide.

Titre: Encaissement des articles
But: encaisser les articles
Précondition: l'utilisateur a au moins un article dans son panier
Postcondition: le client repart avec ses articles





UML

Diagramme d'activité

Diagramme d'activité

Activité + Action

Le diagramme d'activité montre l'enchaînement des *actions* et des décisions :

- au sein d'une *activité* du système

ou

- au sein de **tout** le système

Activité : représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles

Action : une partie de l'activité



Diagramme d'activité

Transition

Transition : relie deux actions ou activités entre elles



- montre le passage d'une activité/action à l'autre
- déclenchée par la fin du comportement de l'activité/l'action source
- provoque automatiquement le début immédiat de l'activité cible
- peut être réflexive (une transition conditionnelle)

Diagramme d'activité

Activités/Actions composites

Les activités/actions peuvent être imbriquées hiérarchiquement :

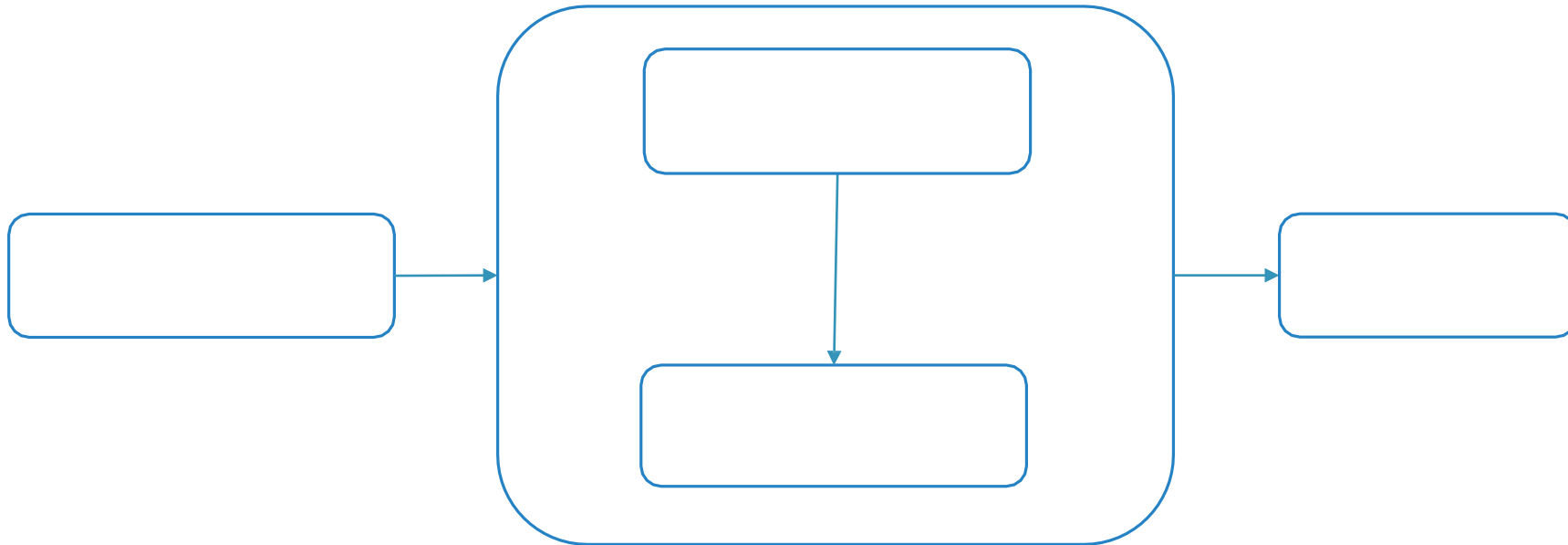


Diagramme d'activité

Nœuds de contrôle

* *Nœud initial* : montre le point de départ de la première activité/action

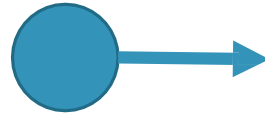


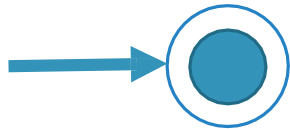
Diagramme d'activité

Nœuds de contrôle

* **Nœud final** : montre le point d'arrivée de la dernière activité/action

- Nœud de fin d'activité

l'exécution de l'activité enveloppante s'achève et tout nœud ou flot actif au sein de l'activité enveloppante est abandonné



- Nœud de fin de flot

le flot est terminé, mais cette fin de flot n'a aucune incidence sur les autres flots actifs de l'activité enveloppante

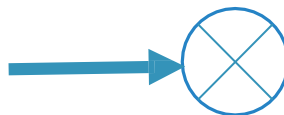


Diagramme d'activité

Nœuds de contrôle

* *Nœud de décision*

Transition conditionnelle

= C'est un aiguillage dans une transition

- On peut préciser la condition de l'aiguillage
- On précise les conditions de passage dans les différentes branches de l'aiguillage

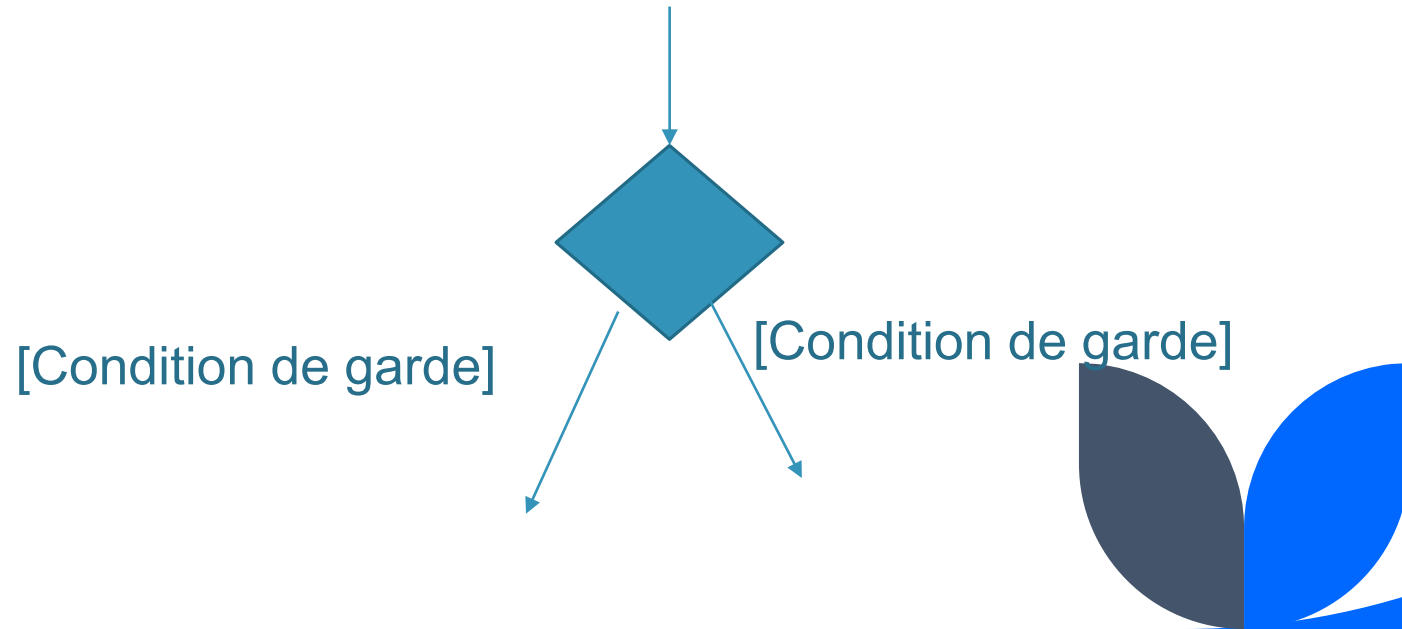


Diagramme d'activité

Nœuds de contrôle

* *Nœud de fusion*

Activités/actions déclenchées par la fin d'au moins une autre activité/action

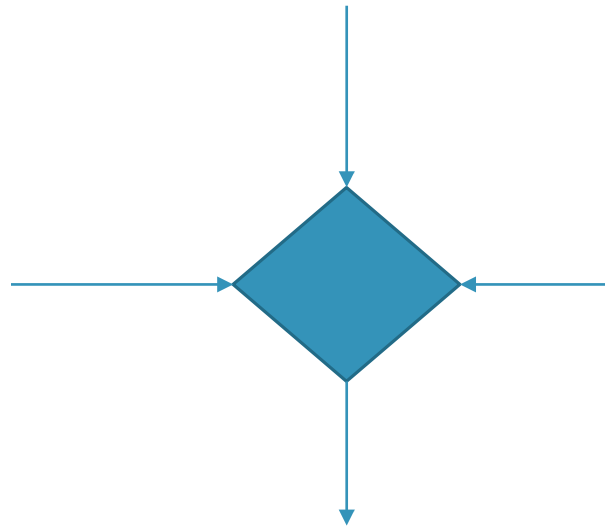


Diagramme d'activité

Nœuds de contrôle

* *Nœud de bifurcation*

Transition (ou synchronisation) fourche

Activités/actions déclenchées en parallèle

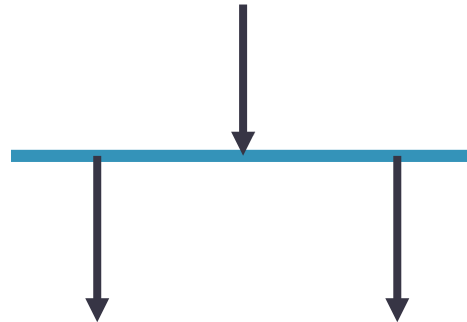


Diagramme d'activité

Nœuds de contrôle

* *Nœud d'union (jointure)*

Transition (ou synchronisation) jonction

Activités/actions déclenchées par la fin coordonnée de plusieurs autres

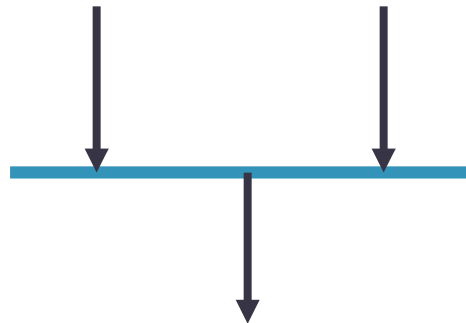


Diagramme d'activité

Exemple

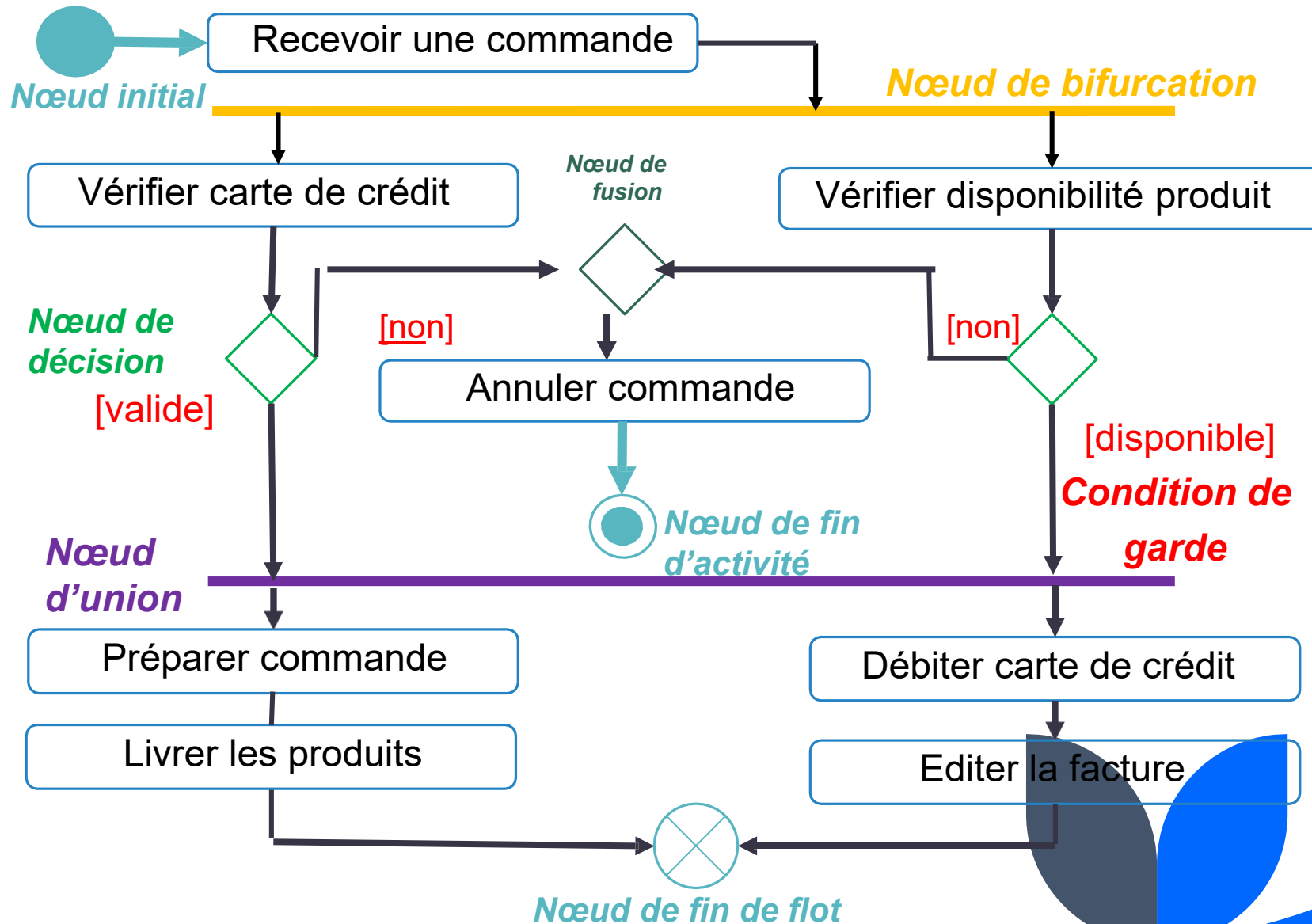


Diagramme d'activité

Exercice

Présentez le ***diagramme d'activité*** qui décrit la recette de la mousse au chocolat suivante :

- Commencer par casser le chocolat en morceaux, puis le faire fondre.
- En parallèle, casser les œufs en séparant les blancs des jaunes.
- Quand le chocolat est fondu, ajouter les jaunes d'œuf.
- Battre les blancs en neige jusqu'à ce qu'ils soient bien fermes.
- Les incorporer délicatement à la préparation chocolat sans les briser.
- Verser dans des ramequins individuels.
- Mettre au frais au moins 3 heures au réfrigérateur avant de servir.



UML

Diagramme de classes

Diagramme de classe

Définition

Le diagramme de classes spécifie la structure et les liens entre les **types d'objets** dont le système est composé.

➡ Ces objets interagissent entre eux et avec les acteurs pour réaliser les cas d'utilisation

❖ **Classe** : La description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques.

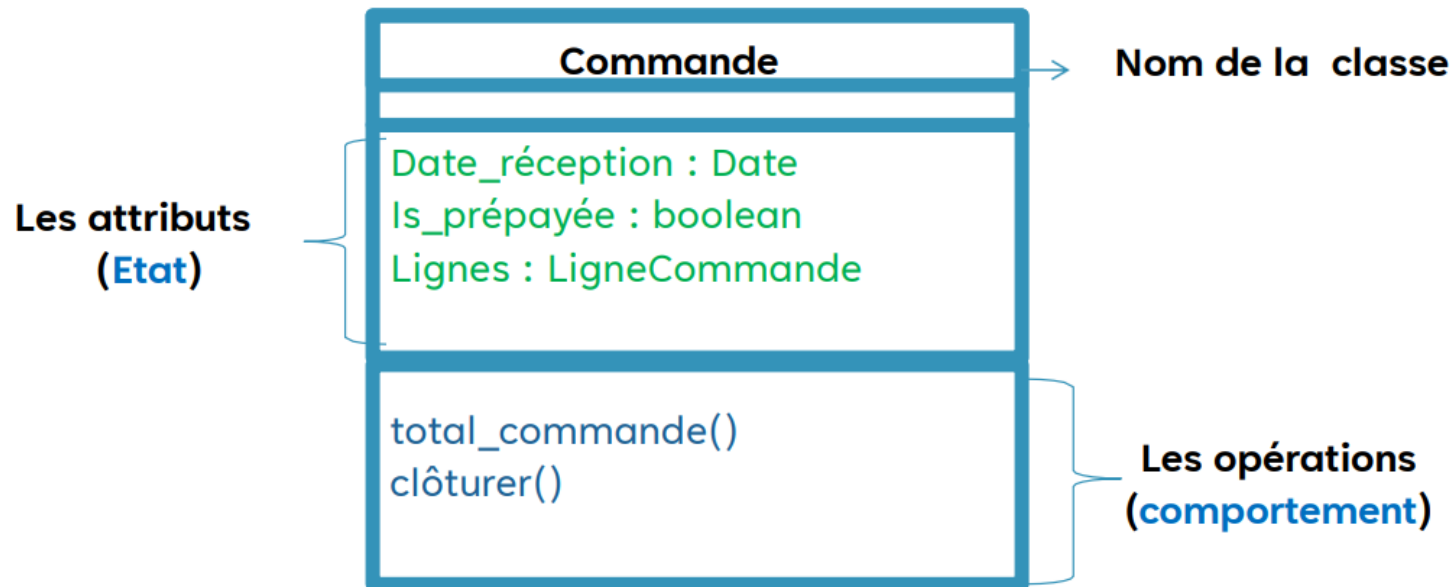


Diagramme de classe

Composants d'une classe

- **Attribut** : Type d'information contenu dans une classe

Visibilité nom : type = valeur_défaut



+ publique
- privée
protégée

Exemple : - nom : String = « x »

- **Opération** : Un élément du comportement contenu dans une classe

Visibilité nom (liste_paramètres) : type_retour



+ publique
- privée
protégée

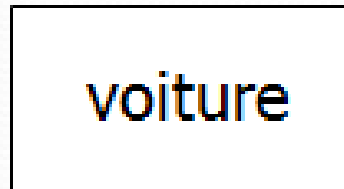


nom : type = valeur_par_défaut

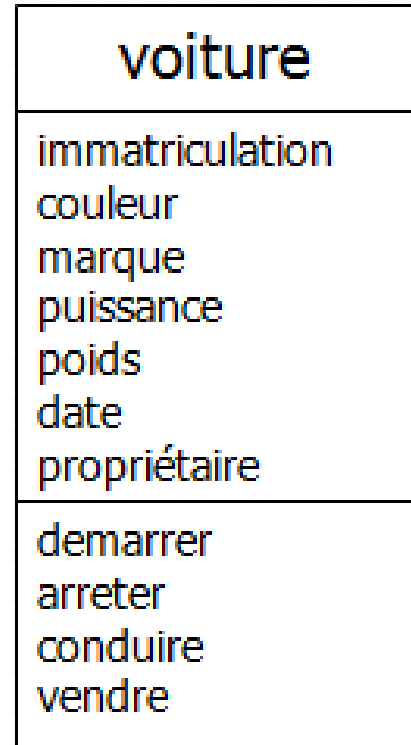
Exemple : + calcul_benefice (date: Date): float

Diagramme de classe

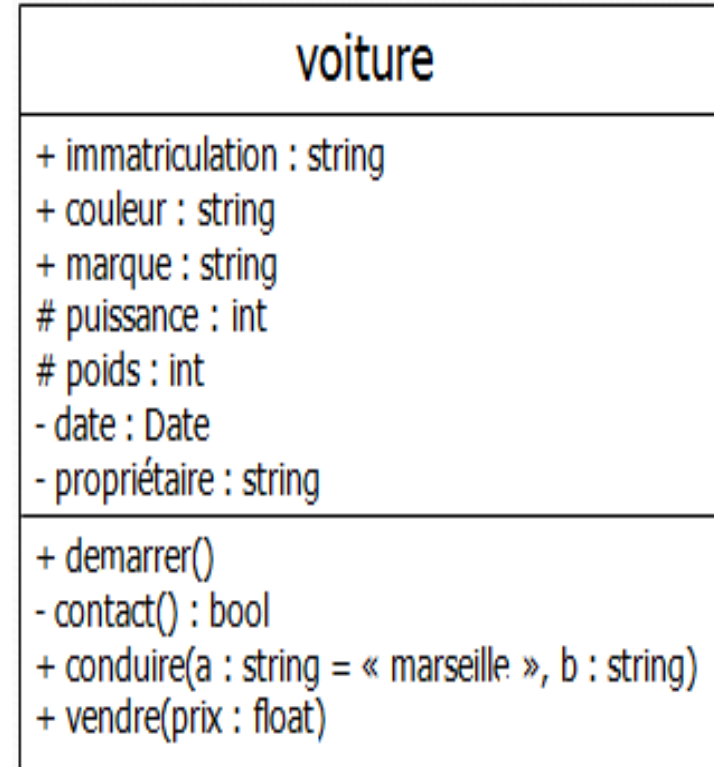
Niveaux d'abstraction



Classe non documentée



Classe documentée



Classe détaillée

Diagramme de classe

Association

- Une **association** est une relation sémantique durable entre deux classes. Elle est caractérisée par une **multiplicité**.
- Une **multiplicité** donne une indication sur le nombre d'objets qui peuvent compléter la propriété (association). Elle est définie avec :
 - Une borne inférieure (**nombre positif ou 0**)
 - Une borne supérieure (**nombre positif ou ***)

On peut utiliser :

Optional : Borne inférieure = 0

Mandatory : Borne inférieure > 0

Single-valued : Borne supérieure = 1

Multivalued : Borne supérieure > 1(généralement *)

Diagramme de classe

Association – Multiplicité (Exemple)

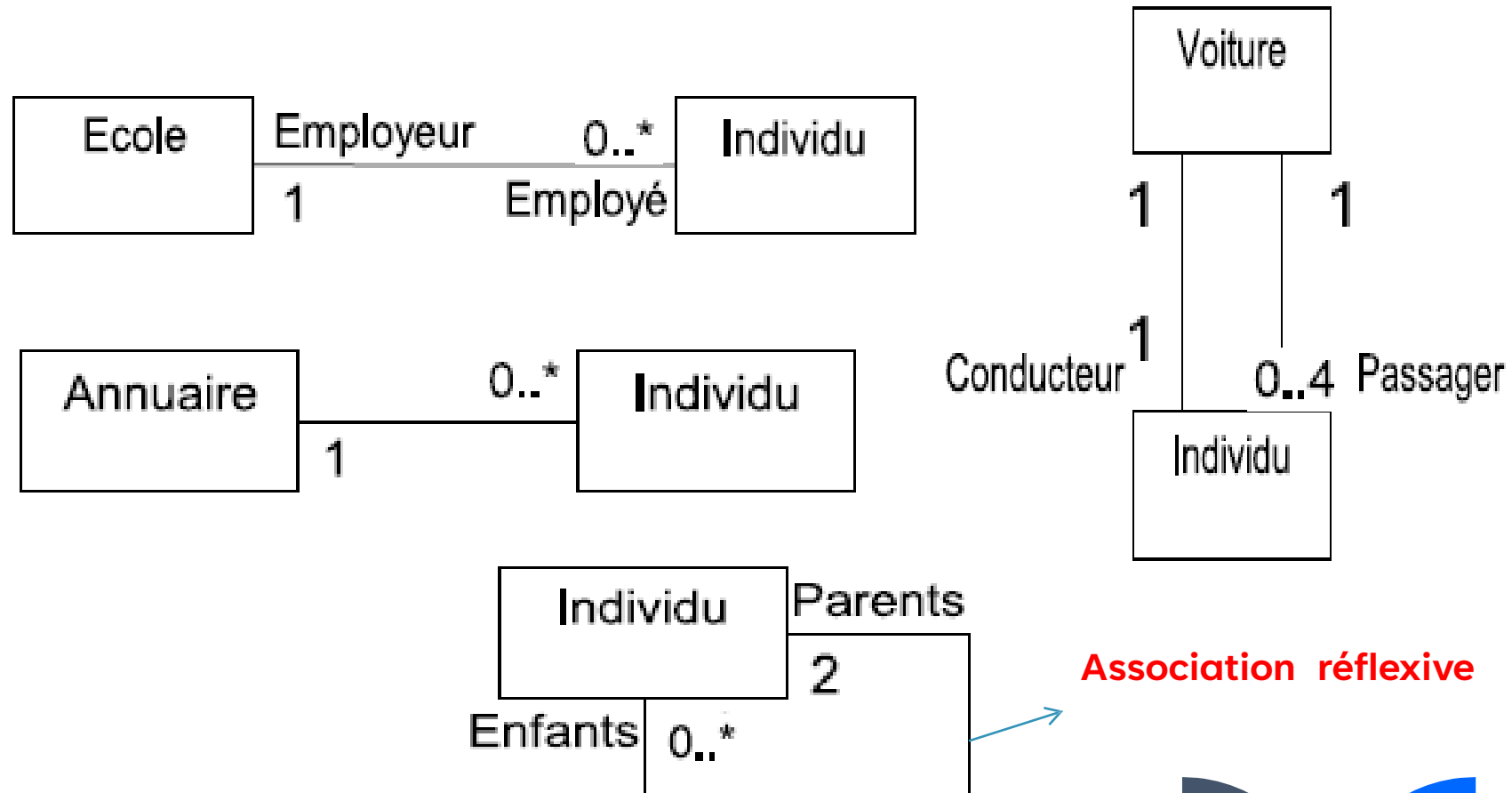


Diagramme de classe

Types d'association

1. **Classe d'association**
2. **Agrégation**
3. **Composition**
4. **Généralisation / Spécialisation**
5. **Association n-aire**
6. **Navigabilité**

Diagramme de classe

Classe d'association

Représenter une association par une classe :

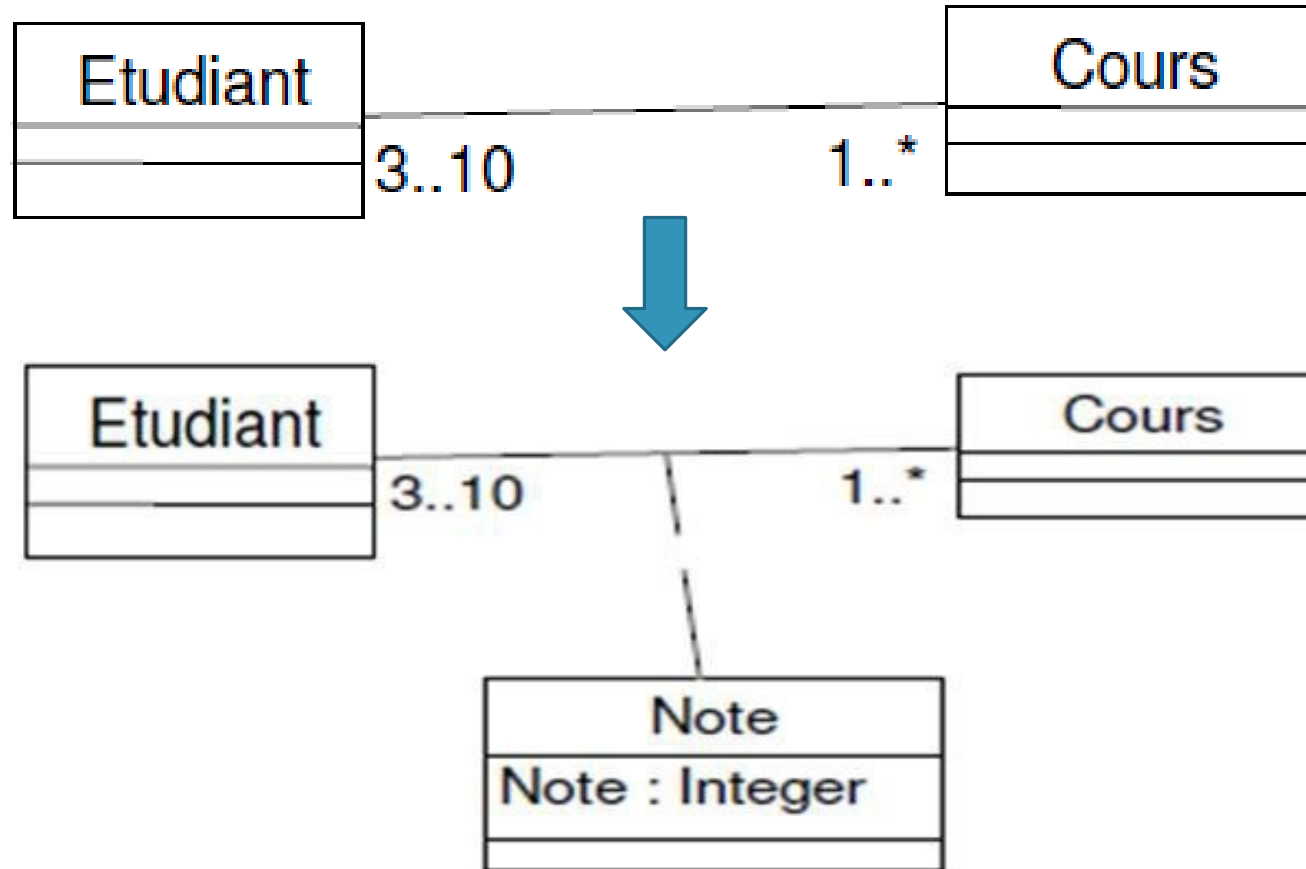
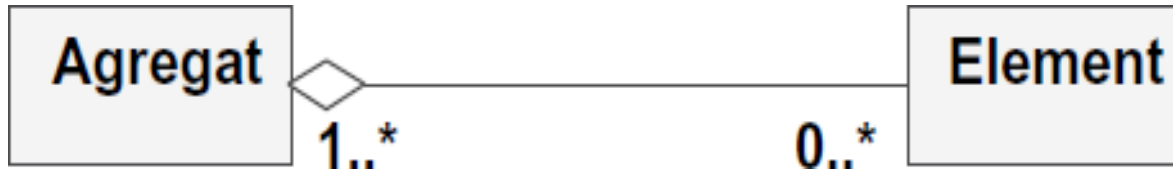


Diagramme de classe

Agrégation

Une agrégation permet d'exprimer une relation de contenance « **contient** »



Exemples :



Une classe peut appartenir à plusieurs agrégat :

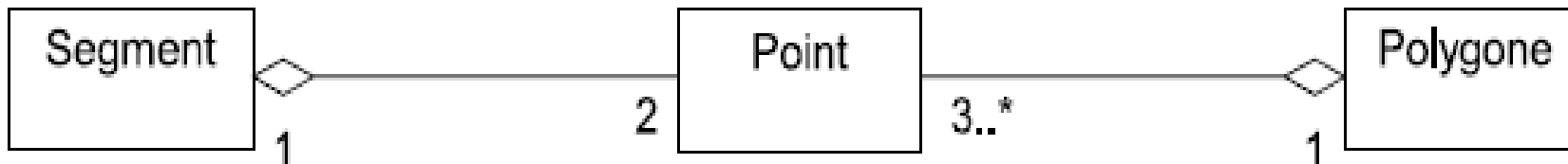


Diagramme de classe

Composition

Une composition est une **agrégation forte**, C'est-à-dire :

- Un élément ne peut appartenir qu'à **un seul agrégat composite**
- La destruction de l'agrégat composite entraîne la destruction de tous ses éléments

Une composition permet d'exprimer une relation « **est composé de** »



Exemple :

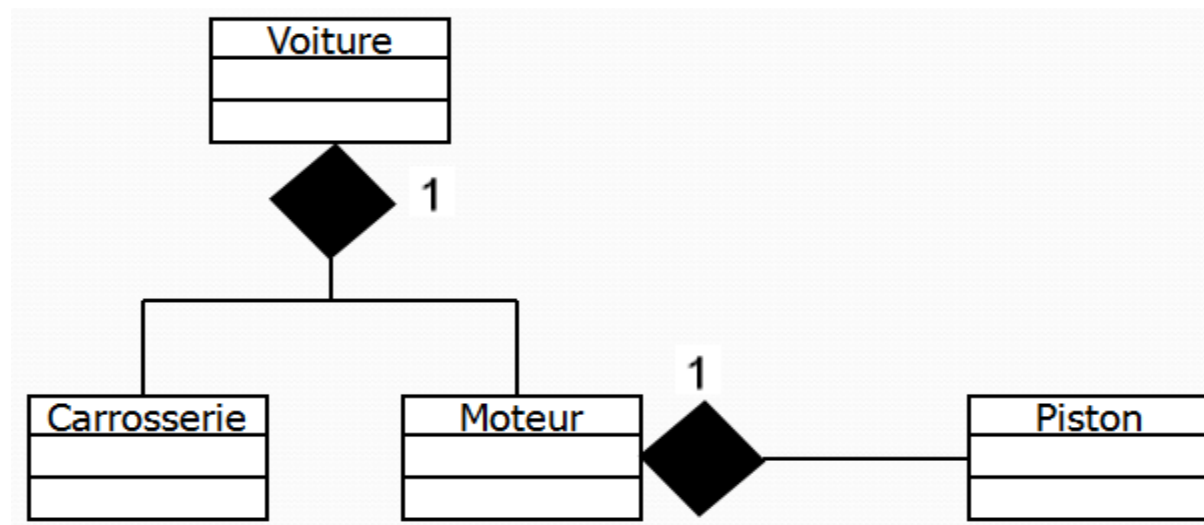


Diagramme de classe

Composition Vs Agrégation

Indicateurs de la composition :

- Si l'expression « **est un composant** » est utilisée pour décrire la relation
La porte est un composant de la voiture
- S'il existe des opérations, s'appliquant sur le tout, s'appliquent aussi sur les composants
La voiture se déplace, la porte se déplace
- S'il existe des attributs dont les valeurs se propagent à ceux des composants
La voiture est bleue, la porte est bleue
- Si la relation est asymétrique
La porte est un composant de la voiture, la voiture n'est pas un composant de la porte

Diagramme de classe

Composition Vs Agrégation

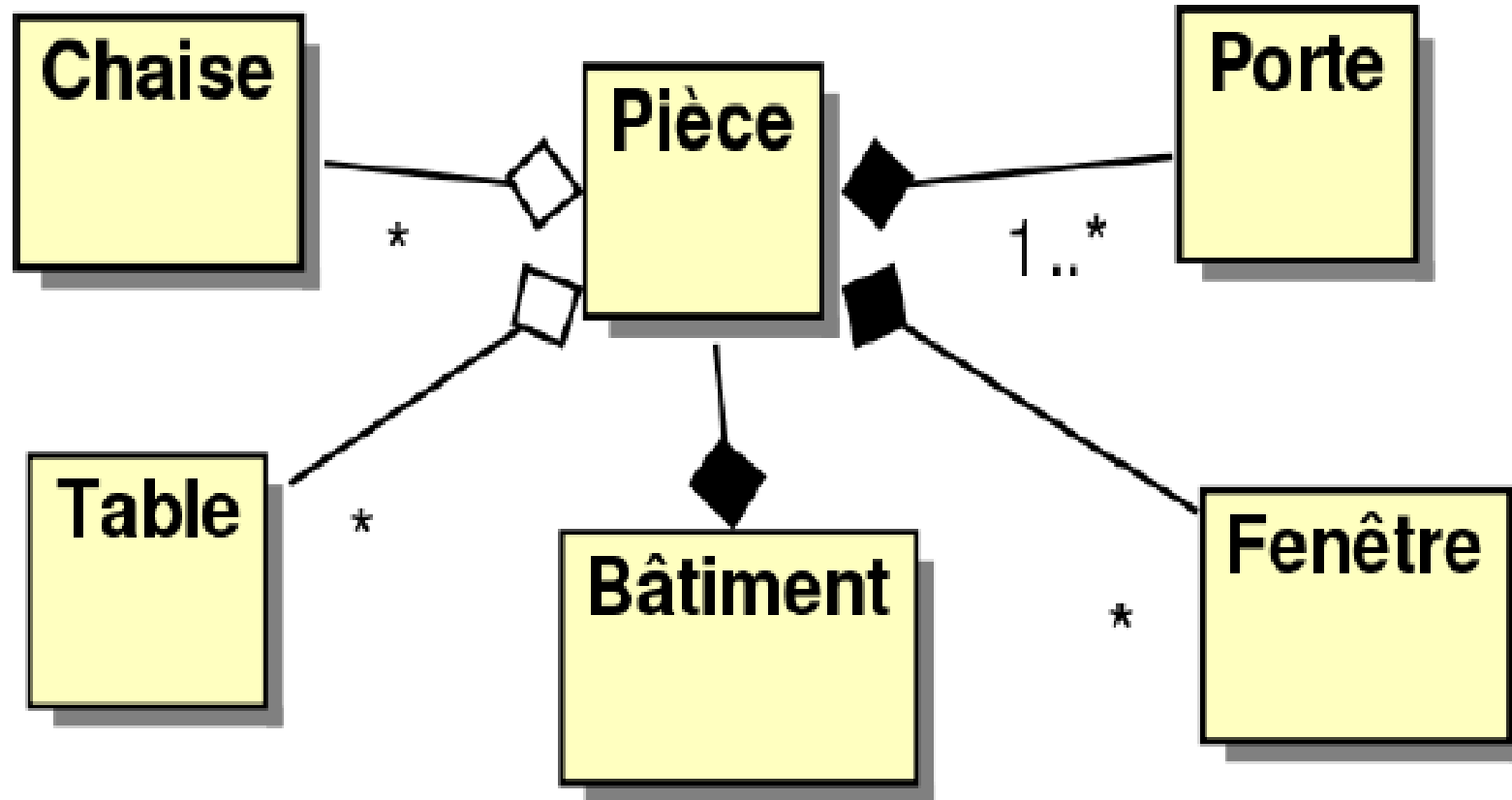


Diagramme de classe

Généralisation / Spécialisation

- **Généralisation** : La création de **superclasses** regroupant les propriétés et les comportements communs aux **sous-classes** (factorisation)
- **Spécialisation** de la classe existante par ajout d'attributs et/ou ajout/redéfinition de méthodes
- Ce type d'association permet d'exprimer une relation « **est un** »

Exemple :

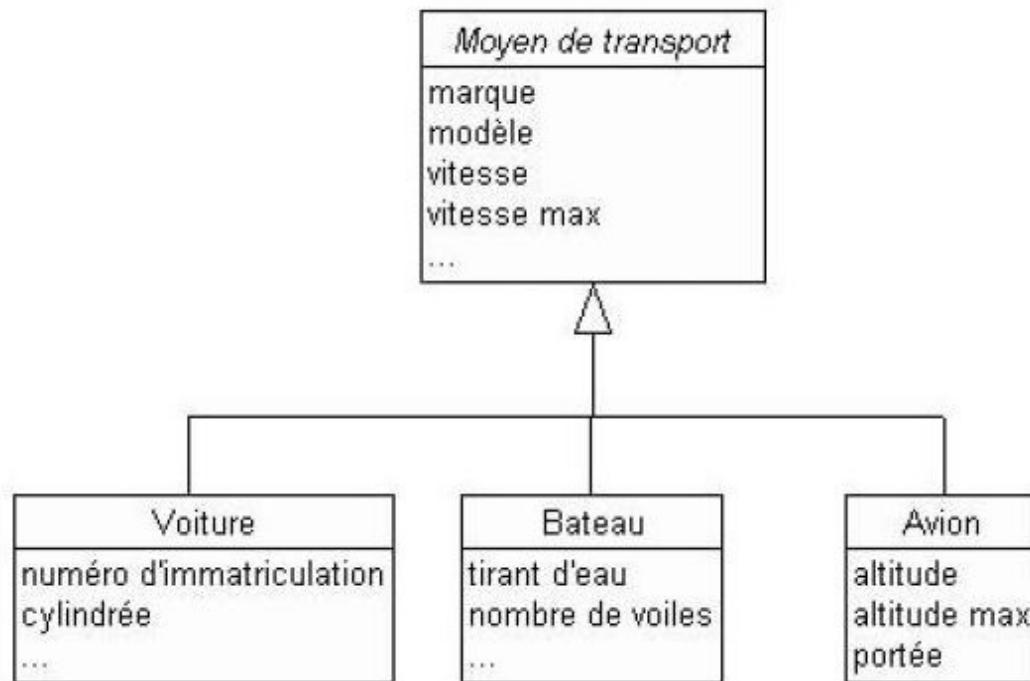


Diagramme de classe

Association n-aire

Association n-aire lie deux classe

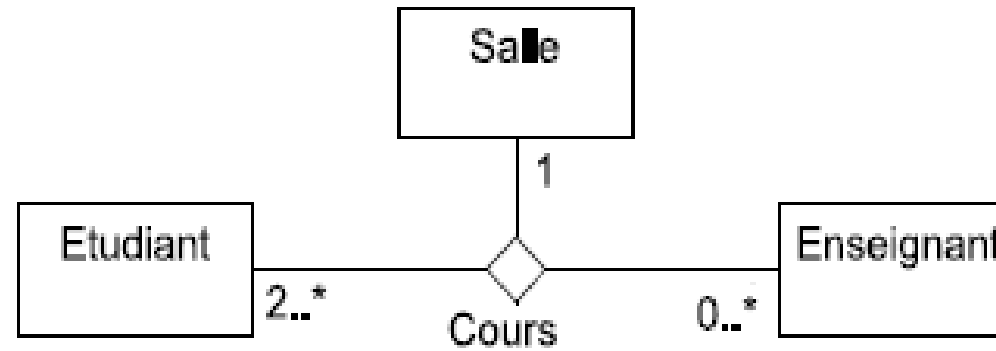
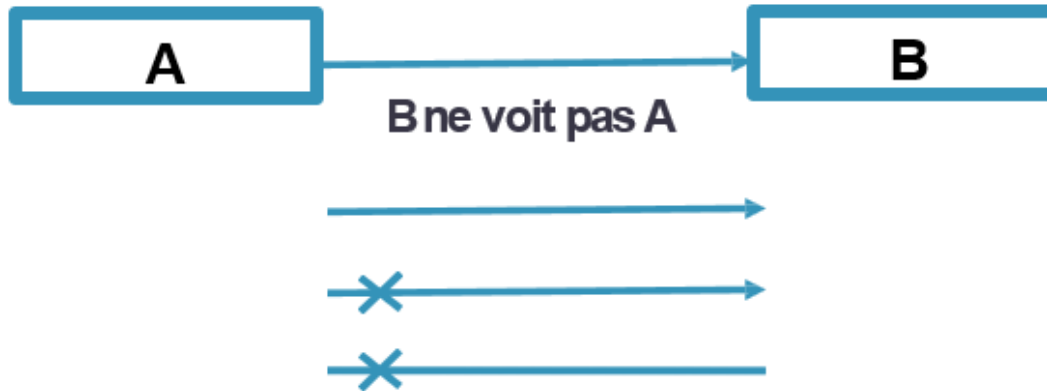


Diagramme de classe

Navigabilité

Une association peut ne nécessiter qu'un seul sens de navigation



Exemple :



Diagramme de classe

Navigabilité

Association bidirectionnelle : paire de propriétés liées

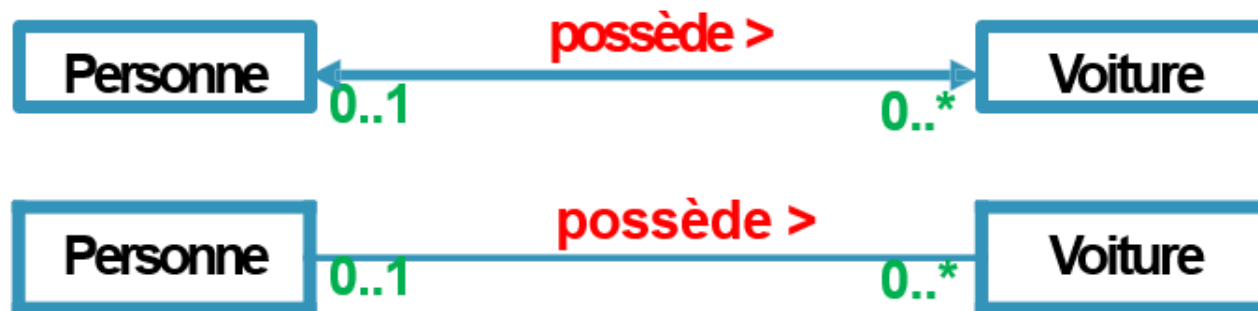


Diagramme de classe

Interface

- Une interface permet de regrouper un ensemble d'opérations assurant un service cohérent par une classe.
- Les classes implémentant une interface doivent implémenter toutes les opérations décrites dans l'interface

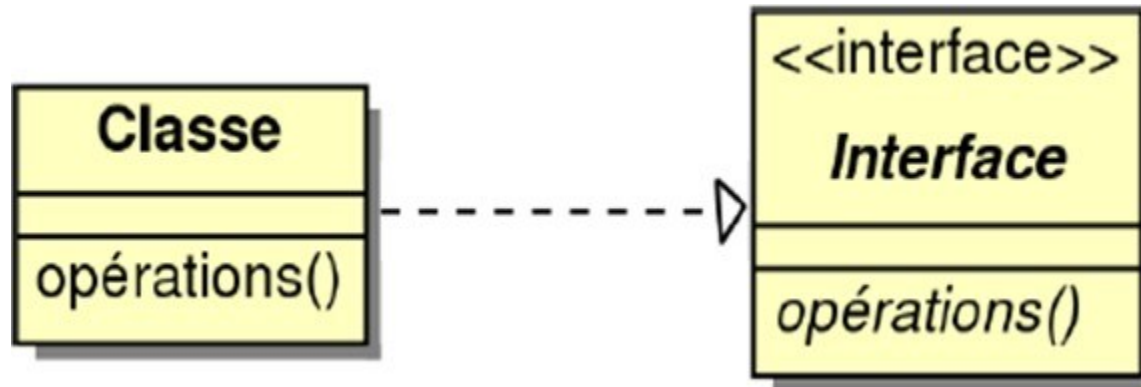


Diagramme de classe

Synthèse

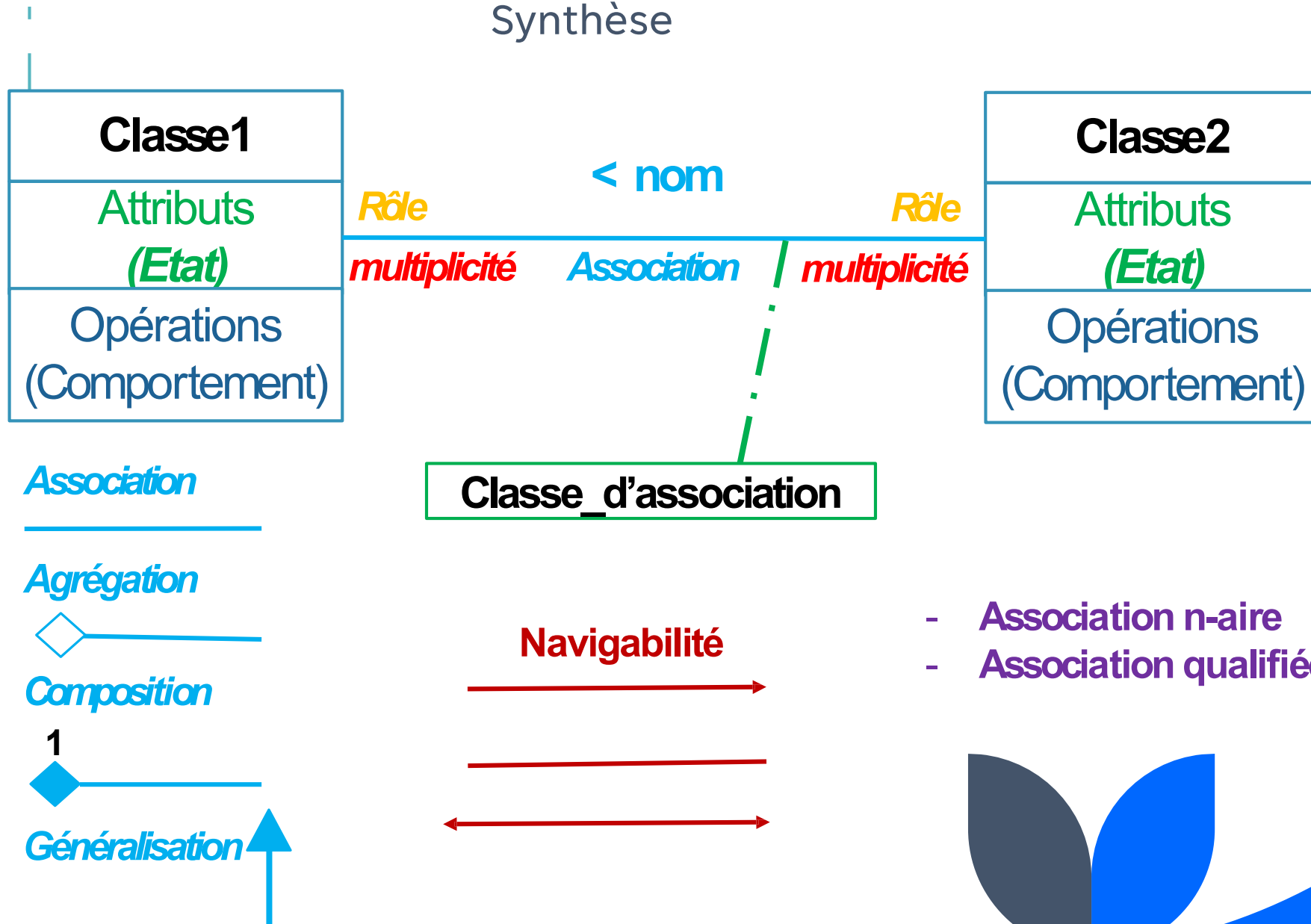


Diagramme de classes

Exercice

Donnez les diagrammes de classes correspondant aux situations suivantes :

1. Un répertoire contient des fichiers
2. Une pièce est composée de murs
3. Les modems et les claviers sont des périphériques d'entrée /sortie
4. Une transaction boursière est un achat ou une vente
5. Un compte bancaire peut appartenir à une personne physique ou morale

Diagramme de classes

Solution

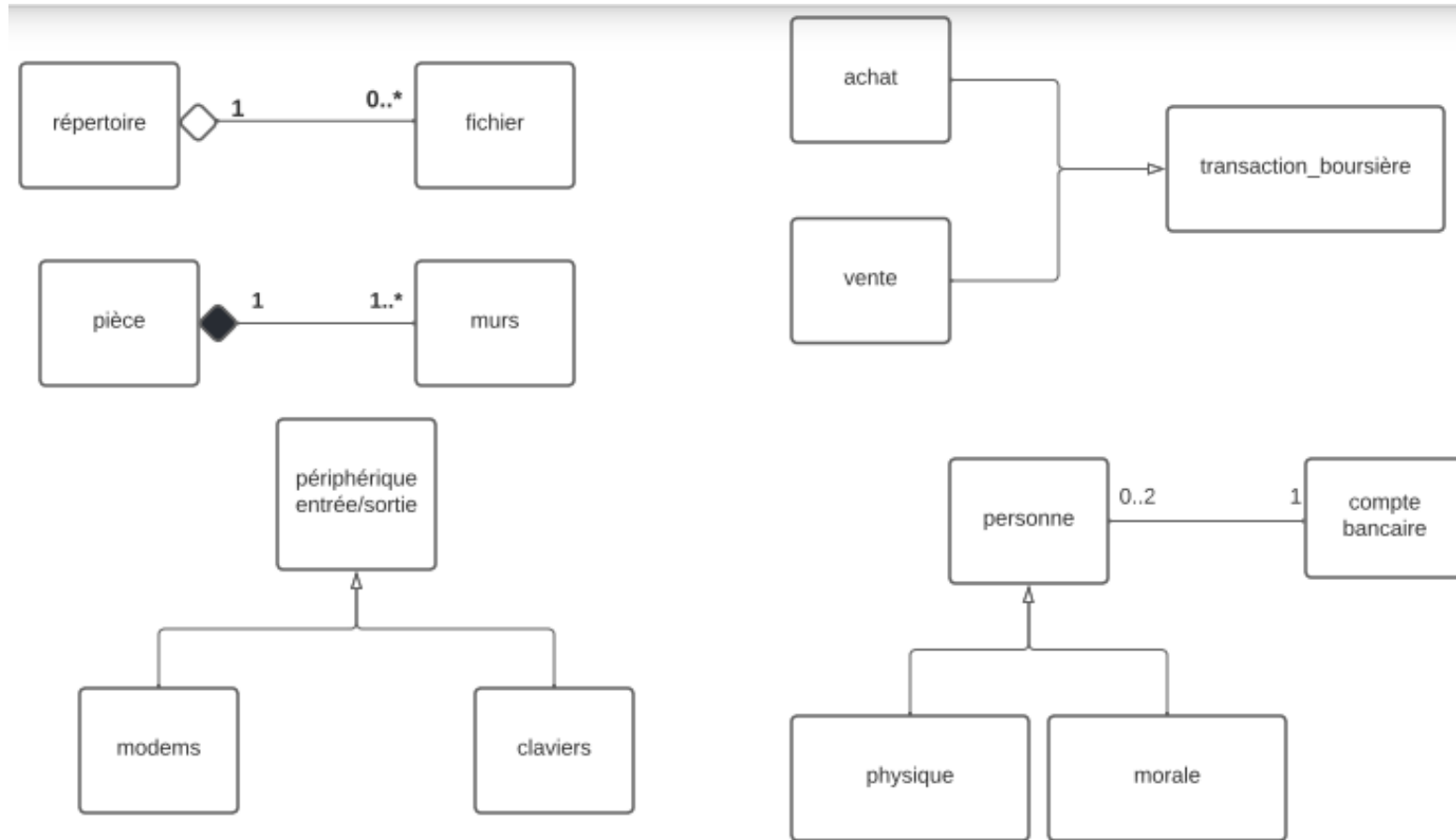


Diagramme d'objets

Définition

Le diagramme d'objets permet de représenter les objets du système à un instant donné afin de :

- Illustrer quelques classes du système
- Mettre en évidence des détails imperceptibles dans le diagramme de classes
- Modéliser des cas particuliers

Représentation des objets en UML :

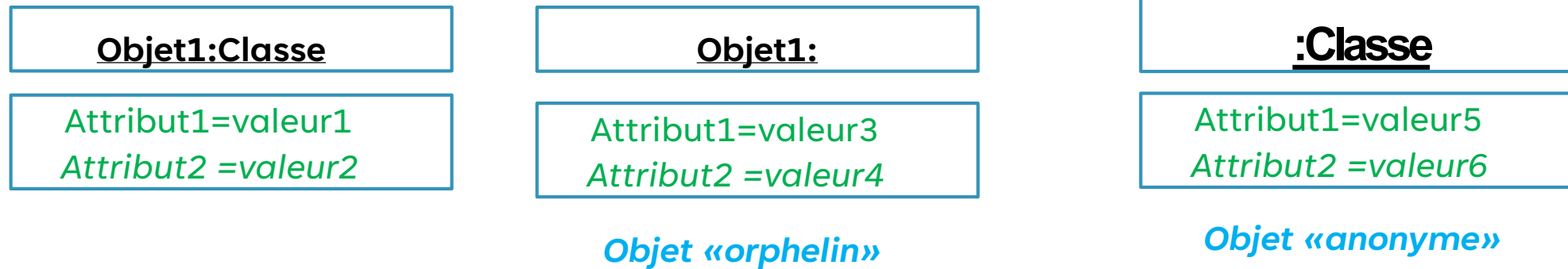


Diagramme d'objets

Exemple

- Diagramme de classes :



- Diagrammes d'objets :

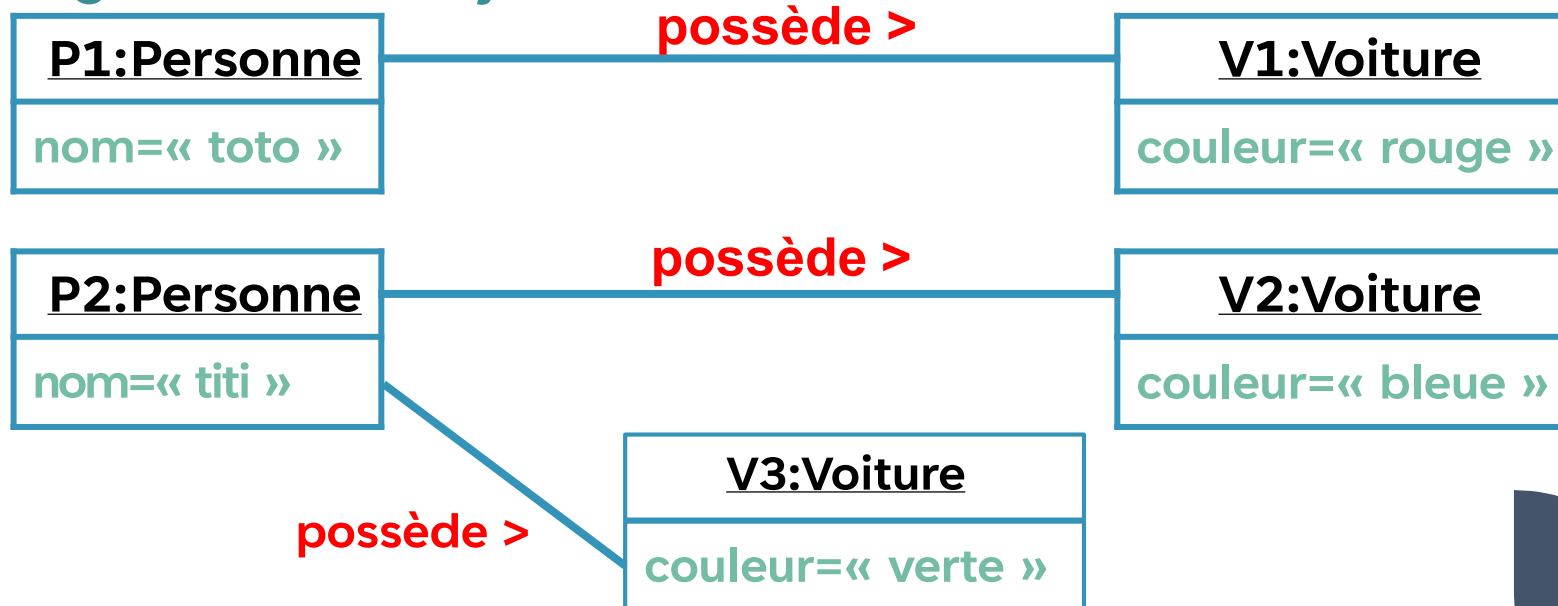


Diagramme d'objets

Exercice

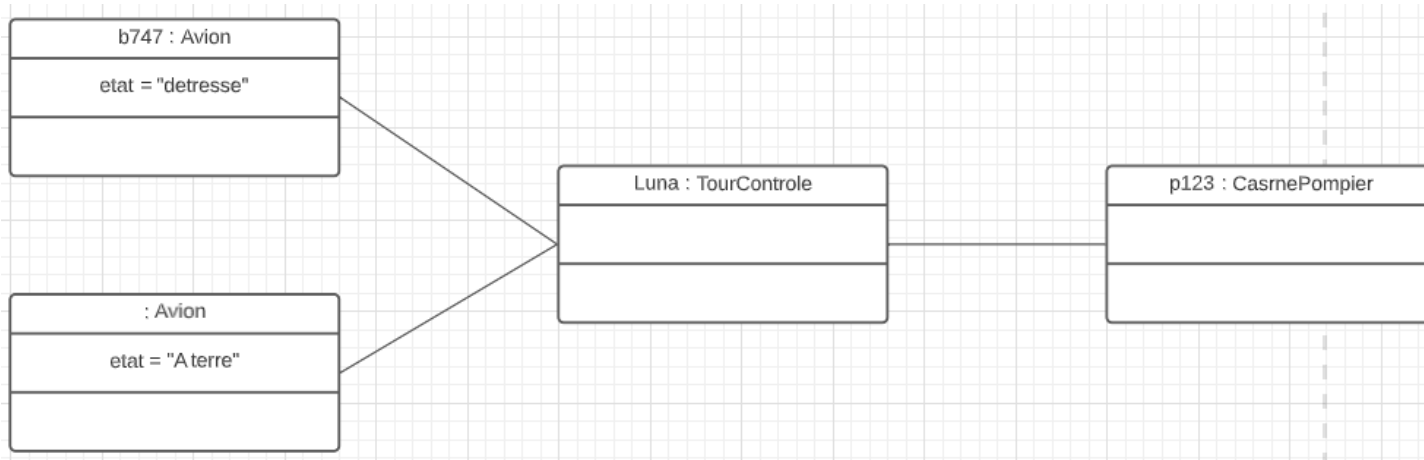
Donnez les diagramme d'objets correspondants aux situations suivantes:

- a) Un objet nommé b747 de classe Avion et en état « détresse » est en relation avec luna, une tour de contrôle. Un ensemble d'autres avions anonymes dont l'état est « à terre » sont aussi liés à luna. La tour de contrôle communique avec p123, une caserne de pompiers.
- b) Etienne et Amaury sont des programmeurs. Etienne est un analyste-concepteur. Ils font partie de l'équipe " DEV " qui représente l'entreprise " DevCom". Le logo de l'équipe "DEV" est " UML". Tous les développeurs utilisent un ordinateur.

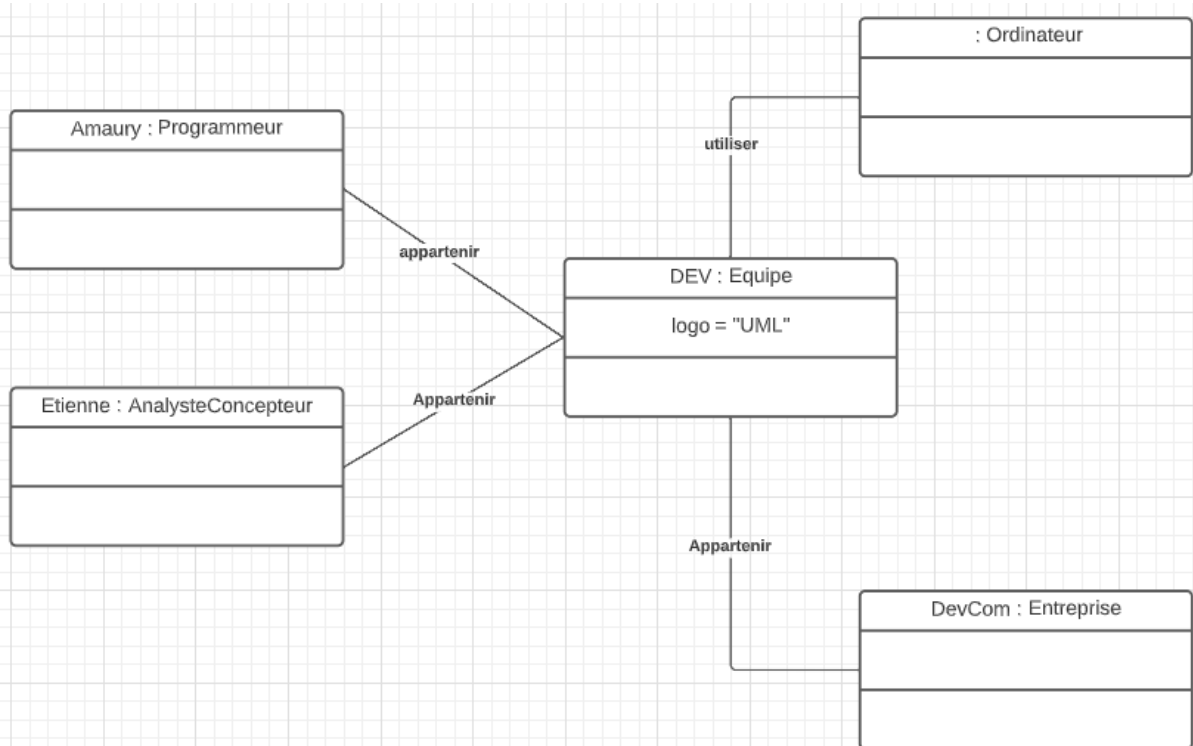
Diagramme d'objets

Solution

a)



b)



Passage de diagramme de classe (UML) vers un modèle relationnel

1. Transformation des classes
2. Transformation des associations
3. Les Transformation d'agrégations / composition
4. Transformation de la relation d'héritage

1 - Transformation d'une classe dans le modèle relationnel

A chaque classe on peut faire correspondre une relation telle que :

- le nom de la relation reprend le nom de la classe
- le nom des attributs de la relation sont issus des noms des attributs de la classe.

Exemple :

Voiture
<u>num</u> : String
marque : String

Voiture(num, marque)
V1: Voiture('1AA45', 'renault')

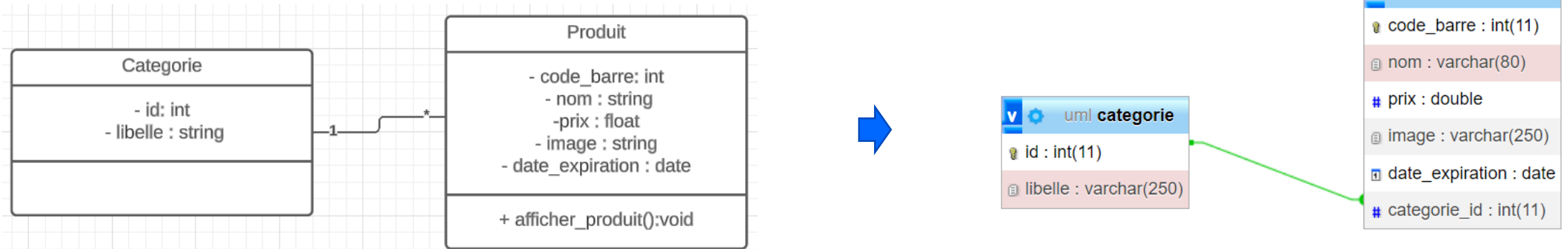
2 - Transformation des associations - un à plusieurs

La clé primaire de la classe mère va être migrer dans la sous classe comme une clé étrangère.

Exemple :

Une catégorie peut contenir plusieurs produit,

Un produit ne peut être contenir que dans une seule catégorie



2 - Transformation des associations - un à plusieurs

```
CREATE TABLE categorie(  
    id int primary key,  
    libelle varchar(250)) ENGINE=INNODB;  
  
CREATE table produit(  
    code_barre int primary key,  
    nom varchar(80),  
    prix double,  
    image varchar(250),  
    date_expiration date,  
    categorie_id int,  
    FOREIGN KEY (categorie_id) REFERENCES categorie(id)  
    )ENGINE=INNODB;
```

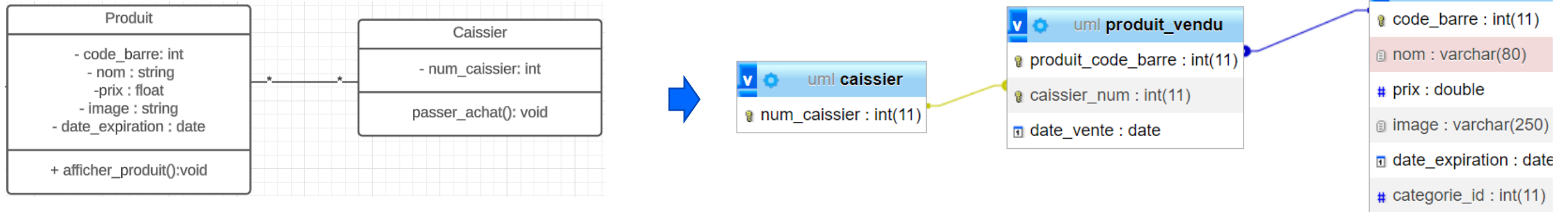
2 - Transformation des associations - plusieurs à plusieurs

Une table doit être créée contenant les deux clés primaires des deux tables,

Exemple :

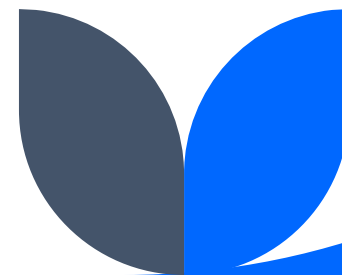
Le produit doit être vendu par plusieurs caissiers

Un caissier va vendre plusieurs produits



2 - Transformation des associations - plusieurs à plusieurs

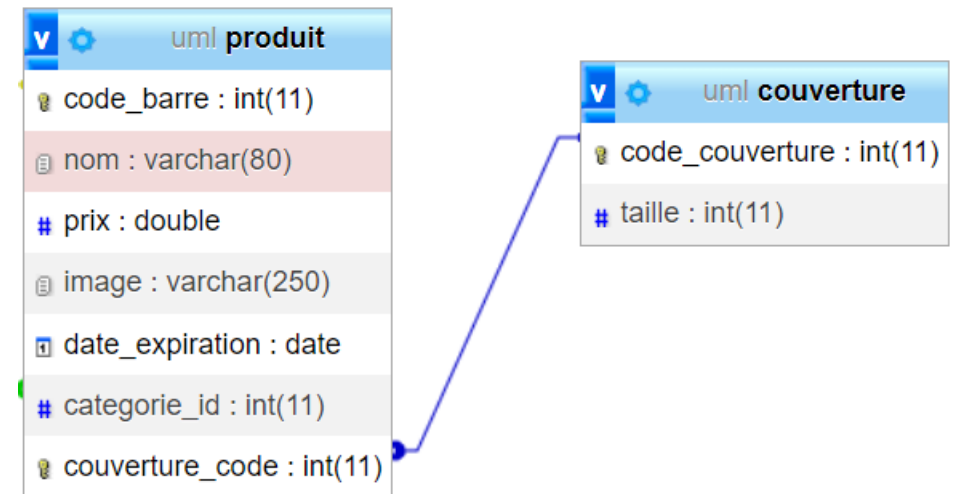
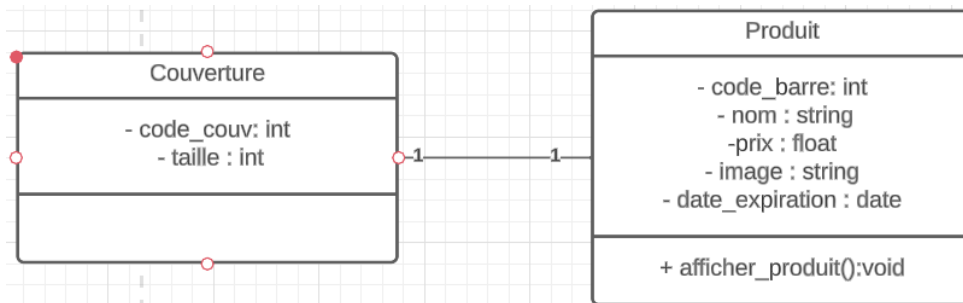
```
CREATE TABLE caissier(  
    num_caissier int primary key) ENGINE=INNODB;  
  
CREATE table produit_vendu(  
    produit_code_barre int,  
    caissier_num int,  
    date_vente date,  
    PRIMARY KEY (produit_code_barre,caissier_num),  
    FOREIGN KEY (produit_code_barre) REFERENCES produit(code_barre),  
    FOREIGN KEY (caissier_num) REFERENCES caissier(num_caissier)  
    )ENGINE=INNODB;
```



2 - Transformation des associations - un à un

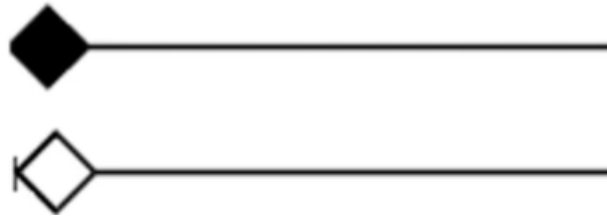
Vous avez le choix de migrer la clé primaire de couverture vers produit ou l'inverse.

Exemple :



2 - Transformation d'agrégation/ composition

Ces types d'associations de traitent de la même façon que les associations classiques, en fonction de la multiplicité (cardinalité) données

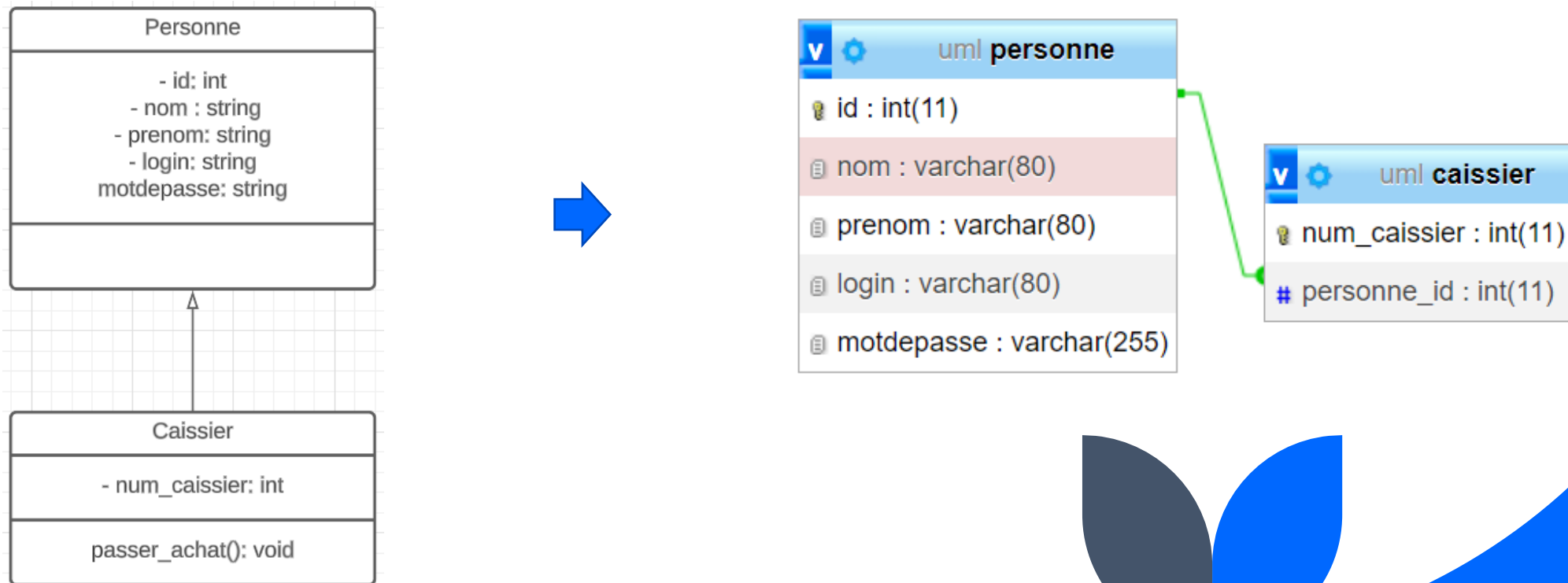


2 - Transformation de la relation d'héritage- Distinction

Nous disposons de trois méthodes distinction, push-down, push-up.

Distinction : La clé primaire de la classe mère va migrer vers la classe fille

Exemple :

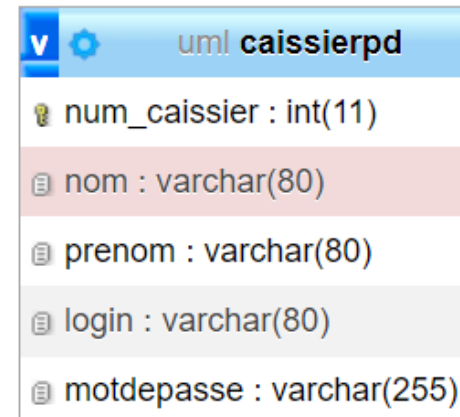
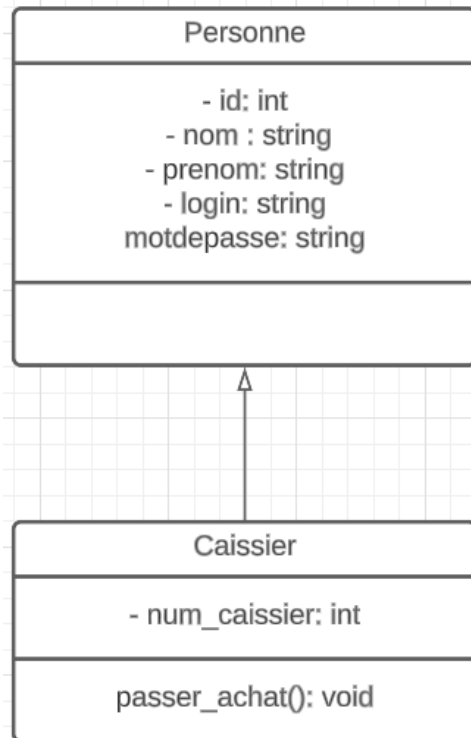


2 - Transformation de la relation d'héritage- Push-Down

Nous disposons de trois méthodes distinction, push-down, push-up.

Push-Down : Pousser en bas, on va pousser les attributs de la classe mère vers la classe fille

Exemple :

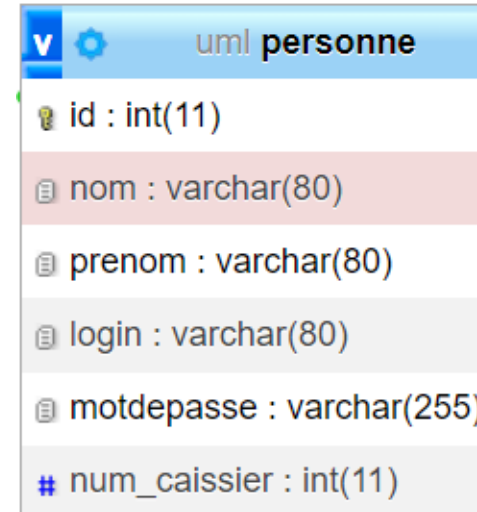
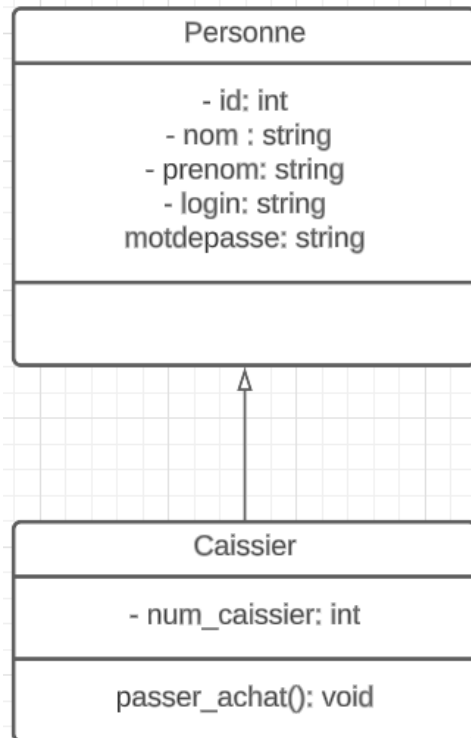


2 - Transformation de la relation d'héritage- Push-up

Nous disposons de trois méthodes distinction, push-down, push-up.

Push-Up: Pousser en haut, on va pousser les attributs de la classe fille vers la classe mère

Exemple :





Merci

Sana EL ATCHIA

Sana.elatchia@lagrandeclasse.fr