

# Analyse et conception d'un système d'information

Antoine Lambert

March 6, 2017

## Contents

<b>1</b>	<b>Le diagramme de classes</b>	<b>2</b>
1.1	introduction . . . . .	2
1.2	Identification des classes . . . . .	3
1.3	identification des attributs . . . . .	4
1.4	Identification des méthodes . . . . .	4
1.5	Résumé Synoptique . . . . .	5
1.5.1	Types de données . . . . .	6
1.5.2	Exercice supplémentaire . . . . .	7
<b>2</b>	<b>Partie 2 : Les Phases</b>	<b>8</b>
2.1	Une phase . . . . .	8
2.2	Les Types de Phases . . . . .	8
<b>3</b>	<b>Partie 3 : ??</b>	<b>10</b>

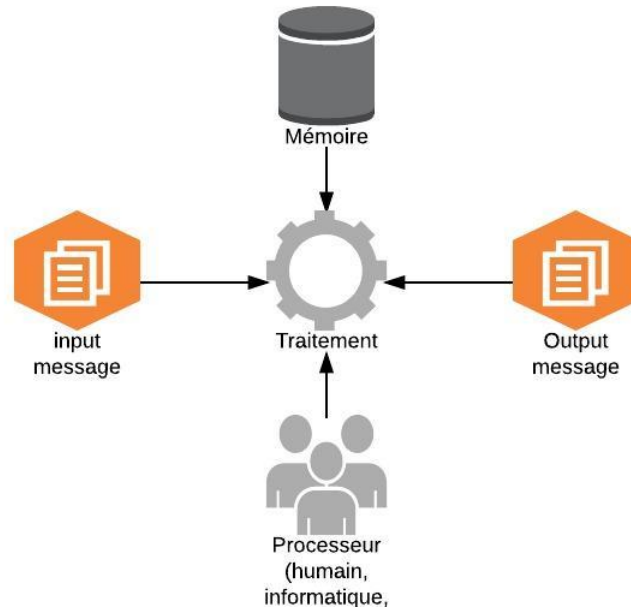
# 1 Le diagramme de classes

## 1.1 introduction

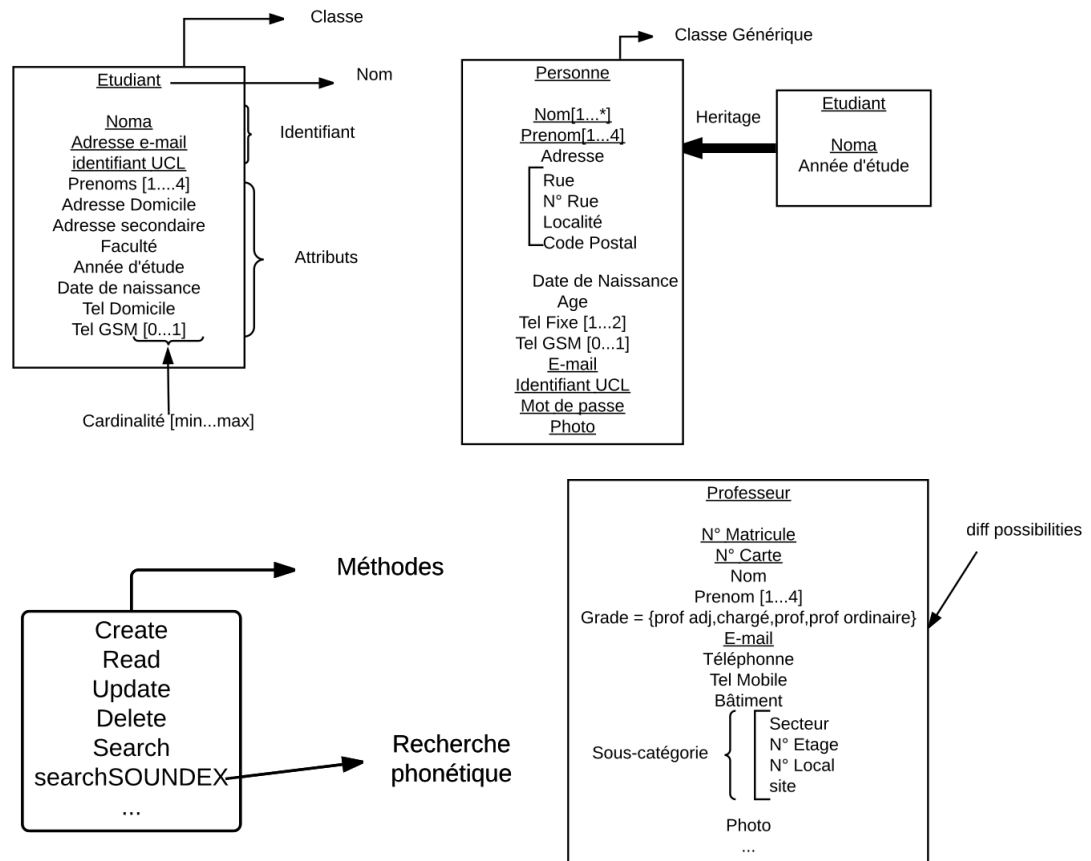
Le diagramme de classe est un **modèle du domaine** (modèle qui définit les objets que l'utilisateur final peut voir, accéder et manipuler à travers le système avec les relations entre ces objets).

Le diagramme de classe est un **modèle structurel** (Une vue d'un système qui met l'accent sur la structure des objets, avec leurs classificateurs, leurs relations, leurs attributs et leurs opérations).

Le modèle du domaine est une extension du modèle des données.



## 1.2 Identification des classes



**UML** : unified modelling language.

**Objet** : une abstraction de quelque chose présent dans le domaine d'intérêt qui doit refléter une capacité du système d'information à maintenir cette information et à interagir avec elle.

**Classe** : description d'un ou de plusieurs objets de manière uniformisée, avec moyens de gérer les objets de cette classe.

**Diagramme de classe** : moyen de décrire les classes en vue de leur implémentation.

Cette implémentation est utile pour permettre une abstraction commune et permettre une implémentation assez semblable quel que soit le langage utilisé. Pour chaque item on peut modifier la visibilité des attributs :

- Public (+): visible et utilisable par toute autre classe (utilisation très limitée)
- Protégé (#): visible et utilisable par toute spécialisation de la classe
- Privé (-): visible uniquement par la classe elle-même

- Dérivé (/): calculé à partir d'autres attributs
- Autre (?): inconnu
- Non spécifié (): indéterminé

### 1.3 identification des attributs

Un attribut :

- un **concept tangible**, concret
- est **adapté** à sa classe
- sont dans une structure d'**héritage**

**Héritage :** l'héritage est la capacité d'une classe à posséder toutes les caractéristiques ( méthodes ou informations) de la classe Parent(SuperClasse).

Un identifiant est un moyen de désigner de manière univoque n'importe quel objet d'une classe. Il est unique et permet de désigner un objet précis dans une table. ( simple(noma) - multiple(noma et Prenom) - groupés(nom-prenom) )

### 1.4 Identification des méthodes

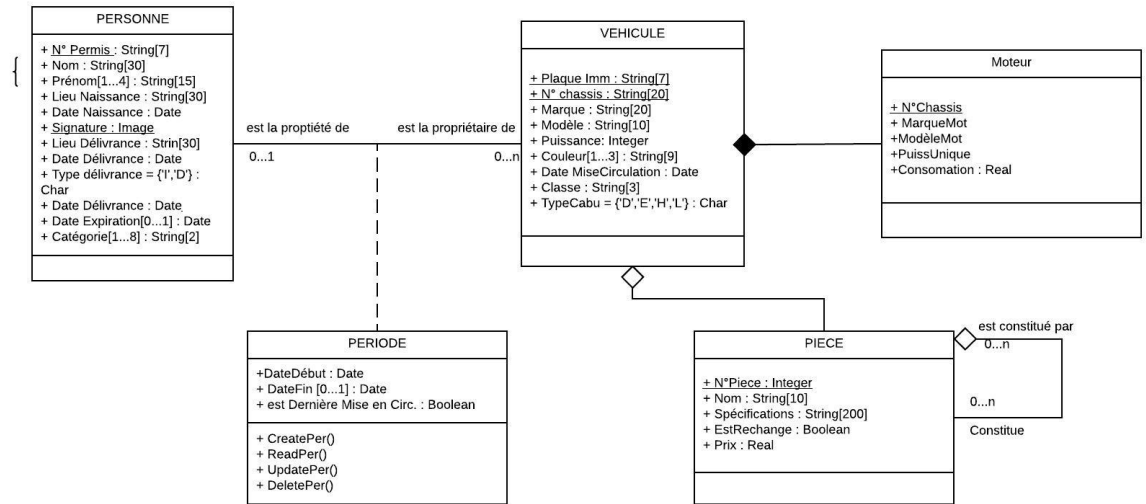
**methode :** action exécutable par une classe.

**état :** tout changement causé par l'utilisation d'une méthode.

\* = cardinalité infinie Repeated -; information qu se repête ( ex domicile 1 et Domicile 2 qui reprends adresse rue pays ...)

Pour l'héritage et la relation de composition il n'y a pas de cardinalité car elle est implicite ( [0...1]et[1...n] )

## 1.5 Résumé Synoptique



- I : indéterminé
- D : date
- /
- D : diesel
- E : essence
- H : hybride
- L : électrique

Ici :

- période est une classe d'association.
- Moteur – relation composition : toujours au moins un
- Piece : relation d'agrégation : toujours facultatif

la relation de pièce à pièce est appelée récursive car une pièce peut elle même être composée de pièces.

On peut avoir la récursion sur l'agégation et non avec la composition, JAMAIS

### 1.5.1 Types de données

Boolean = 0,1V,Foui,non

Hour

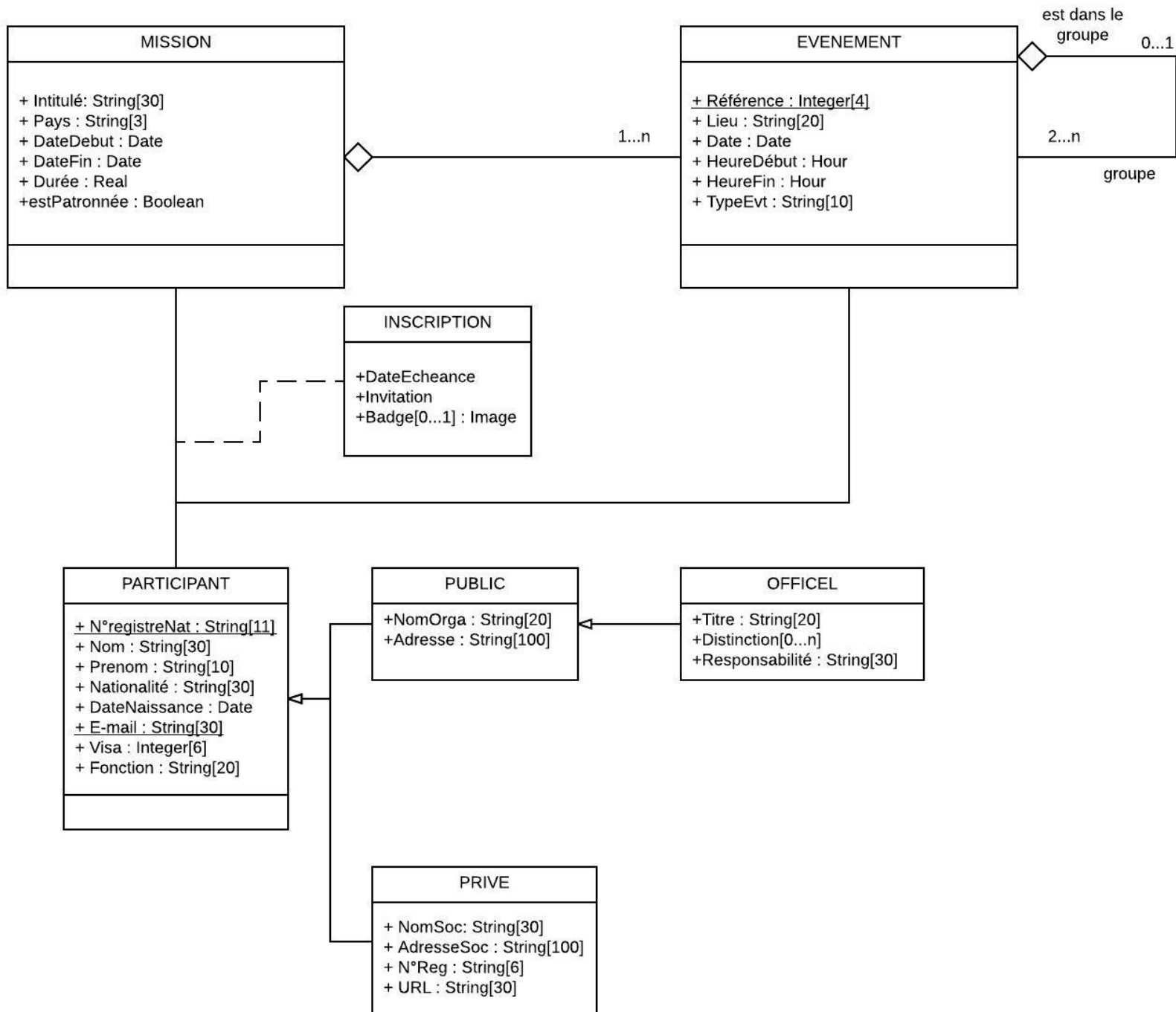
Date

Integer[4] = Liste Codes Post

Real

Char(= String[1])

### 1.5.2 Exercice supplémentaire





## 2 Partie 2 : Les Phases

### 2.1 Une phase

- traitement
- unité de temps
- unité de lieu
- unité de ressources

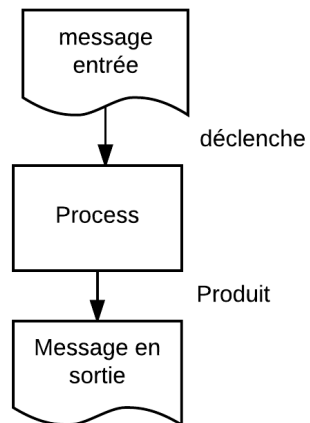
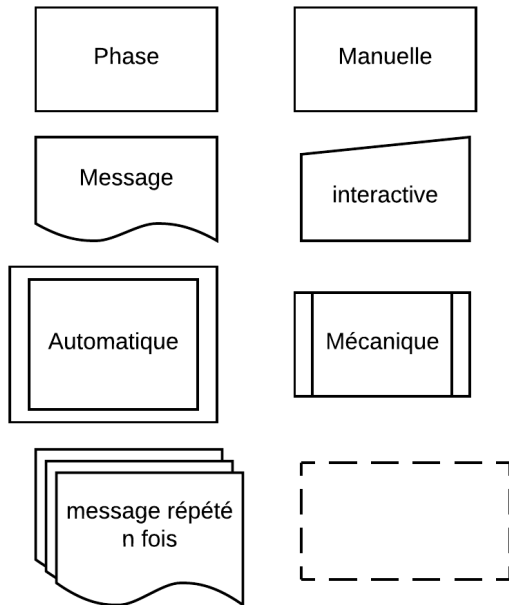
à compléter sur slides

### 2.2 Les Types de Phases

- **automatique** : nécessite des ressources de type système
- **Interactive** : nécessite des ressources de type système + Humaine
- **Manuelle** : nécessite des ressources de type Humaine
- **Mécanique** : Humaine et machine (ex : mise sous pli)

N°	Nom	Préd	Justification	Nature
1	PréparerBon	/	/	Manuelle
2	EnregistrerCde	1	chg lieu : lieu(prépa) -> lieu(opérateurs)	Interactive
2	//	//	chg ress pers : +opérateur	//
2	//	//	chg ress l/m : +SI + terminaux	//
2	//	//	chg ress info : + commande correcte	//
2	//	//	pt att déc : si complète	//
2	//	//	diff périod : $f(P2) \neq f(P1)$	//
3	Préparer Requis	2	chg ress l/m : -terminaux	Automatique
3	//	//	chg ress pers : -opérateurs	//
3	//	//	chg ress l/m : +SI + terminaux	//
3	//	//	pt att déc : si correcte	//
3	//	//	diff périod : $f(P3) \neq f(P2)$	//
4	PréparerSérie	3	chg ress info : +série	Automatique
4	//	//	pt att accumulation : "n bons"	//
4	//	//	diff périod : $f(P4) \neq f(P3)$	//
5	ExécuterSérie	4	chg lieu : -> lieu(entrepôt)	Manuelle(now:mécanique)
5	//	//	chg ress pers : +magasinier	//
5	//	//	chg ress l/m : -SI	//
5	//	//	chg ress info : +série exécutée	//
5	//	//	diff périod : $f(P5) = 1 \neq f(P4)$	//
6	ConstituerColis	5	chg lieu : lieu(entrepôt) -> lieu(salle de colisage)	Manuelle
6	//	//	chg ress pers : -magasinier, +opérateur(embal)	//
6	//	//	chg ress l/m : +SI + terminaux	//
6	//	//	chg ress info : +colis(+ docs colisage)	//
6	//	//	deff périod : $f(P6) \neq f(P5)$	//
7	ExpédierColis	6	chg lieu : lieu(salle de colisage) -> lieu(expéditeur)	Manuelle
7	//	//	chg ress pers : -opérateur(embal) +Société d'expédition	//
7	//	//	chg ress info : +colis expédié	//
7	//	//	deff périod : $f(P7) \neq f(P6)$	//
8	SélectionnerCommandeDiff	3	chg ress info : +nouvelle livraison	Automatique
8	//	//	pt att acc : si nouvelle livraison	//
8	//	//	deff périod : $f(P8) \neq f(P3)$	//

### 3 Partie 3 : ??



relations

