

FRL



Prof. Santucci Jean-François
SPE - UMR CNRS 6134
UNIVERSITE DE CORSE
email : santucci@univ-corse.fr

Les frames : concepts de base

- Un frame est une unité de connaissance (prototype) décrivant une situation ou un objet [Minsky 1975]
- Un frame possède des attributs décrits par des facettes
- Les facettes sont
 - déclaratives (domaine, valeur, défaut, ...)
 - procédurales (réflexes, démons, ...)

Les frames : concepts de base

→ Un exemple de frame (syntaxe Lisp)

```
(AutoDeSarah  
  (est-un    (= Voiture))  
  (couleur   (= Rouge))  
  (puissance (fiscale 9)  
              (unité cv)  
              (réelle 95)))
```

→ Tout est placé dans le frame

→ Les facettes définissent la sémantique de l'attribut

→ Mise en correspondance : la valeur de l'attribut d'un frame peut être un frame ou un ensemble de frames

Les frames : Concepts de Base

→ L'article de Minsky n'est que conceptuel...

→ Concrétisation de ces idées dans les Représentations de Connaissances par Objets (RCO) ou représentations post-frames ou à base de frames

Les frames : concepts de base

→ Les RCO privilégient l'aspect instanciation

- héritage de propriétés entre des objets
- distinction classe/instance parmi les objets

→ Les RCO rationalisent l'usage des facettes

- de typage
- d'inférence
- de service

Les Frames : Concepts de base

→ Exemple : Différences LOO/RCO

LOO	RPO
langage de programmation	outil de représentation de connaissances
héritage "statique"	héritage dynamique
méthodes de classes	procédures attachées aux attributs
envoi de message	primitive d'accès et activation de réflexes
abstraction	pas de données privées

Les frames : concepts de base

- Un objet au sens RCO est une structure à 3 niveaux : attributs, facettes, valeurs
- Distinction Classe/Instance le plus souvent
- Une hiérarchie de spécialisation décrit les liens entre les classes d'objets et leurs sous-classes : simple ou multiple
- Dans les RCO adoptant la distinction Classe/Instance
 - une classe décrit un modèle structurel d'un objet :
 - ▢ ensemble des propriétés décrivant sa structure
 - ▢ ensemble des opérations qui lui sont applicables
 - une classe décrit une famille (un concept) de l'univers de discours

Les frames : concepts de base

- L'extension d'une classe est l'ensemble de ses instances
- L'intension d'une classe est l'ensemble des descriptions de ses attributs
- Une instance est un représentant particulier d'une classe
- L'extension d'une classe est l'ensemble des individus de cette classe effectivement représentés dans la base de connaissances

Les frames : concepts de base

- Stocker et organiser la connaissance autour de la notion d'objet
- Fournir des services inférentiels de bas niveau destinés à compléter l'information disponible ou à dériver la connaissance implicite
- Faciliter la modélisation et la manipulation des objets
- Deux notions sont mises en avant: *réification* et *déclarativité*

Les frames : concepts de base

→ La classe définit deux aspects de ses instances:

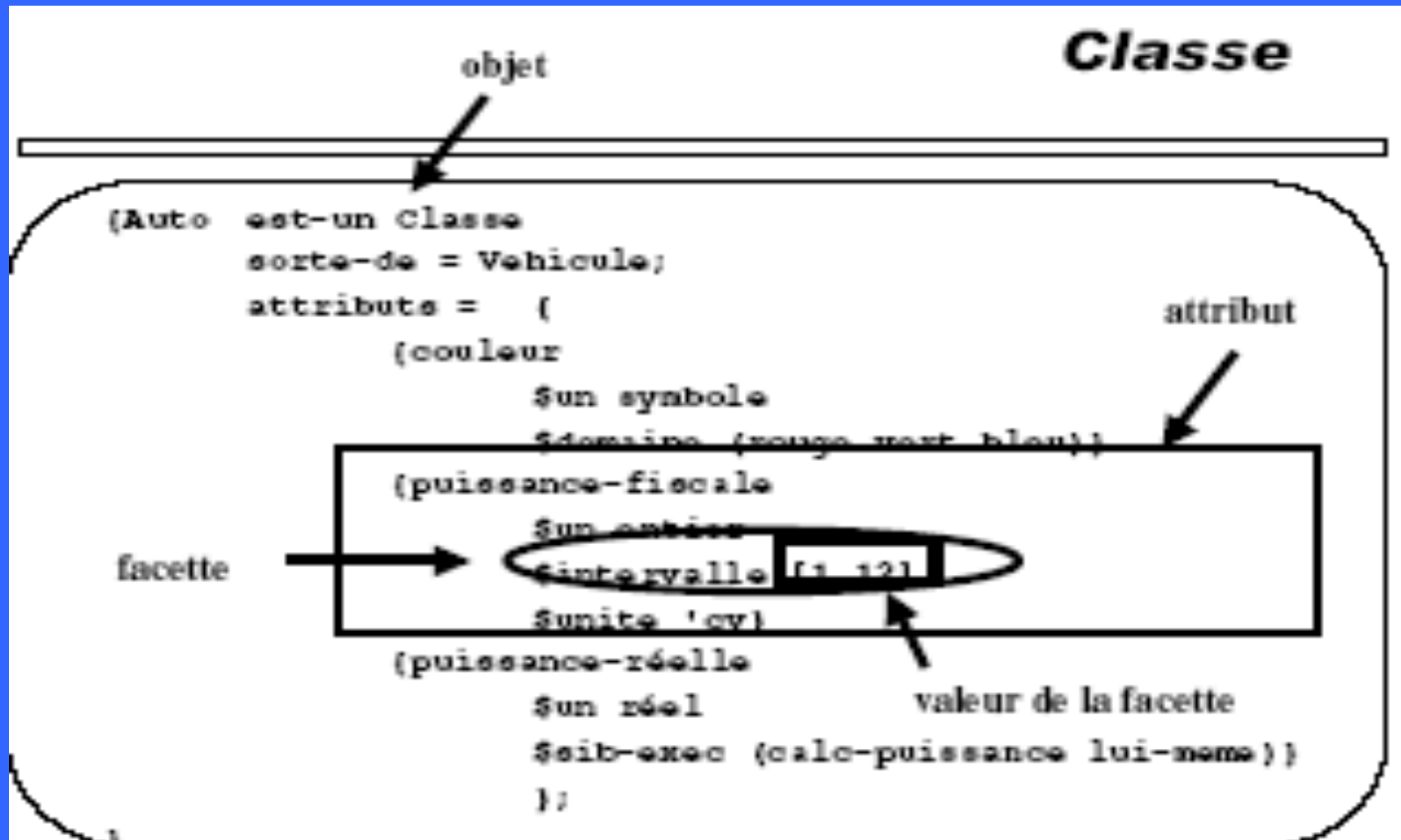
1) Aspect structurel

- Liste des attributs,
- Type de ces attributs (pas toujours pris en compte car entraîne des vérifications),
- Mono ou multi-valués,
- { Contraintes }

2) Aspect inférentiel

- Mécanismes d'inférence des valeurs d'attributs.

Les frames : concepts de base



Les frames : concepts de base

- Couples attributs-valeurs
- Attributs possiblement vides (valeur inconnue)
- Attributs référençant d'autres objets

```
{AutoDeSarah est-un Auto
  couleur = rouge ;
  p-réelle = inc ;
  p-fiscale = 9;
  moteur = {Moteur#72
    est-un Moteur
    cylindrée =2;
  }
}
```



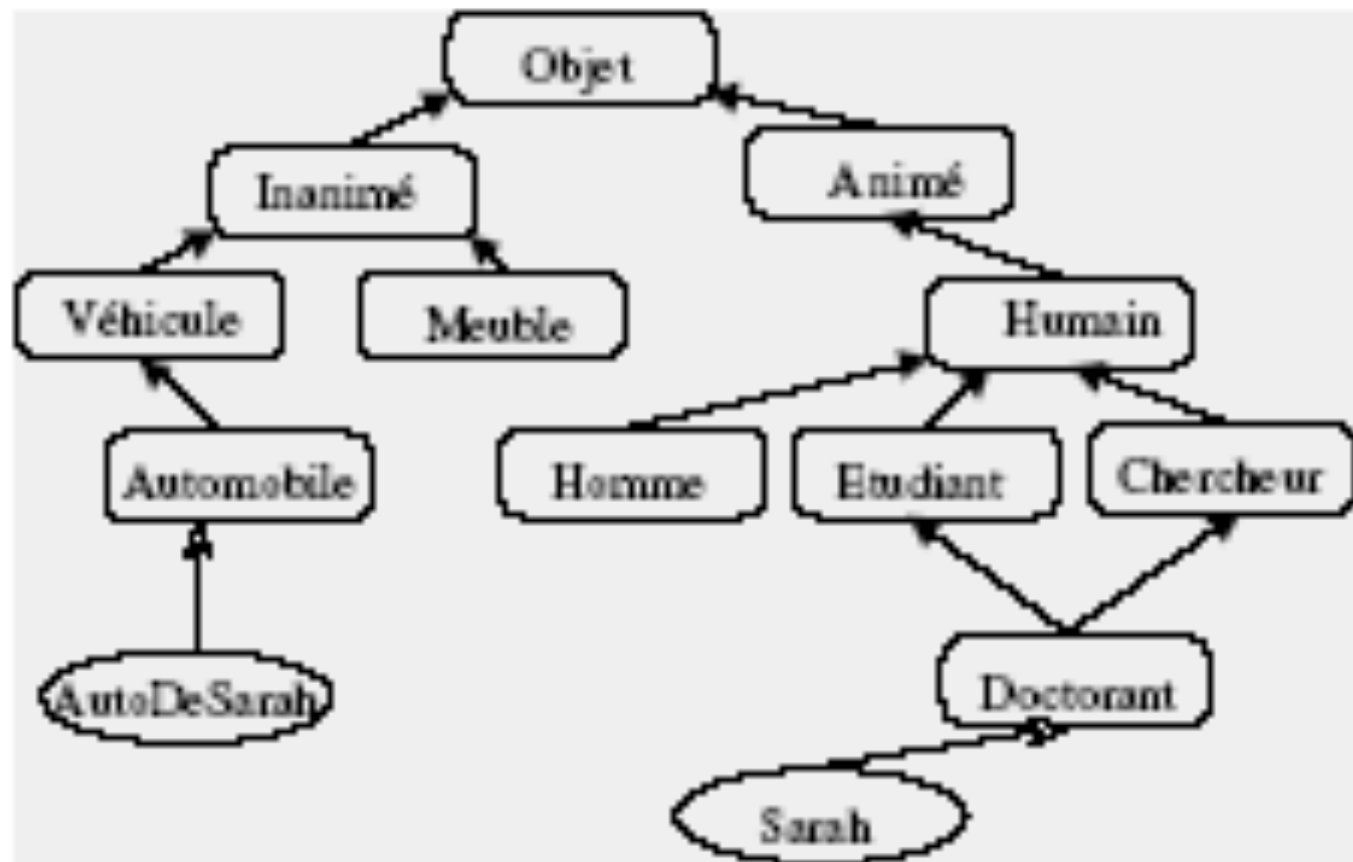
Les frames : concepts de base

Attribut

→ Il peut représenter :

- une propriété de l'objet (âge, nom, *etc.*) : il a alors un type simple (entier, chaîne de caractères, *etc.*)
- une relation dans laquelle l'objet peut être impliqué (propriétaire-de, père-de, *etc.*) : il a alors pour type une classe (sa valeur sera une instance) de la base de connaissances
- un composant (moteur, roue, *etc.*) : c'est un cas particulier du précédent. L'objet est un *objet composite*, l'attribut désigne l'un de ses *composants*. Une sémantique particulière peut être associée à la *relation de composition*

Graphe de Spécialisation



Graphe de Spécialisation

On parle de *multi-spécialisation* lorsqu'une classe de la hiérarchie est sous-classe de plusieurs autres classes.

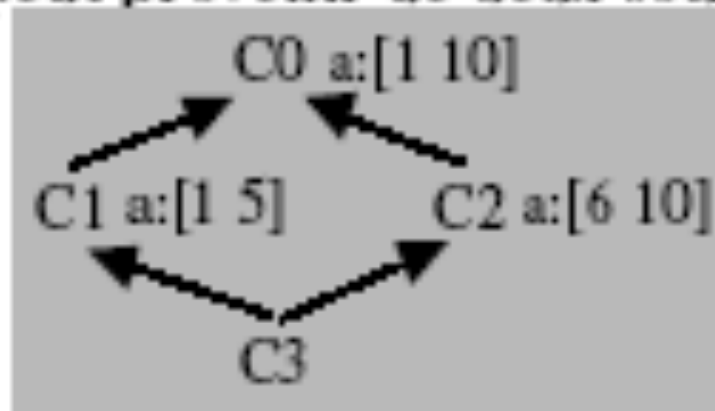
En *mono-spécialisation* (toute classe n'a qu'une sur-classe), la hiérarchie de classes est un arbre

En multi-spécialisation, la hiérarchie de classes est un graphe acyclique orienté

→ Comme dans le cas de la mono-spécialisation, la classe doit respecter la sémantique de la spécialisation → affinement ou ajout d'attributs

Graphe de Spécialisation

Il peut arriver que deux attributs de même nom imposent des contraintes inconsistantes. La classe ne peut donc pas avoir d'instance (car cela conduirait à l'inconsistance de la base). Cela peut provenir de deux sources:



- Les deux sur-classes sont incompatibles
- Les deux attributs représentent des attributs distincts:
l'utilisateur aurait dû faire attention

Vérification

Possibilité, lors d'une modification de l'objet de vérifier des conditions exprimées sous forme de prédicats. Elles peuvent porter sur un ou plusieurs attributs :

```
{Resistance est-un Classe
  sorte-de = Composant;
  attributs = {
    {v-entree $un entier}
    {v-sortie $un entier}
    {intensite $un entier}
    {resistance $un entier} ;;}
```

```
$a-verifier = v-entree - v-sortie = intensite*resistance;
```

Valeurs par défaut

→ La valeur par défaut est un mécanisme d'inférence qui réussit toujours !

→ Elle spécifie la valeur que va prendre l'attribut si aucune valeur n'est fournie. Elle peut cependant être masquée

Exemple : «Par défaut, le nombre de pieds d'une chaise est 4»

```
{Chaise est-un Classe
    sorte-de = Meuble;
    attribute = ({ nb-pieds $un entier $default 4)...
};
}
```

→ La valeur par défaut (facette \$ défaut) alliée à une valeur de classe (facette \$ valeur) est une solution possible au traitement des exceptions

Valeurs par défaut

→ Indique que la valeur d'un attribut donné est celle d'un autre attribut (lorsque le premier est inconnu et le second est connu)

Exemple : "Par défaut, le nom de famille d'une femme est celui de son époux"

(Femme est-un Classe

 sorte-de = Personne;

 attributes = {

 (non-famille \$un chaine \$← nom-epoux)

 (epoux \$un (Homme

 attributes = {

 (non-famille \$var non-epoux)

 ...})) ...})

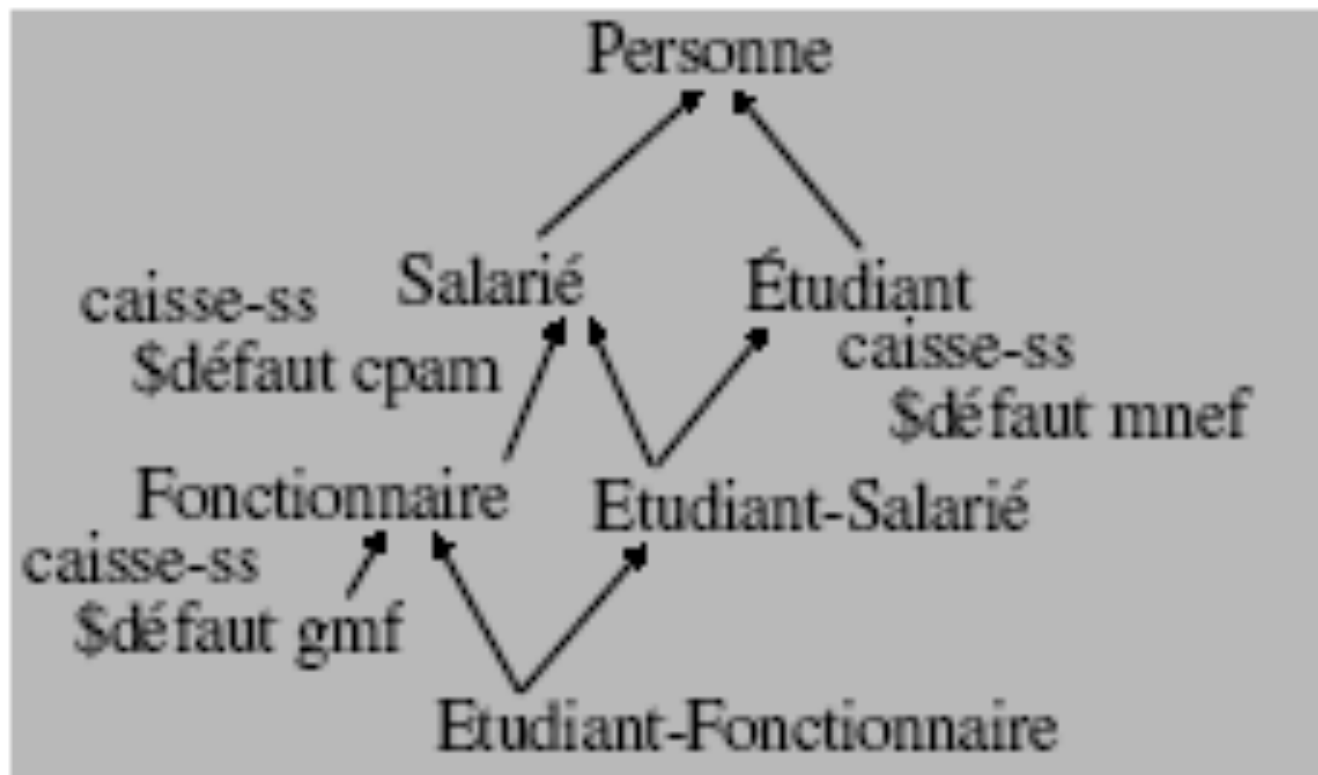
}

Recherche de Valeurs : Mécanismes d'inférence

- Les mécanismes d'inférence sont définis dans les classes
- ils sont applicables aux instances de ces classes et aux instances de leurs sous-classes
- Le long de la hiérarchie, pour un attribut dans une même instance, chaque mécanisme d'inférence peut *a priori* retourner une valeur différente adaptée, en général, à la classe dans laquelle il est présent
- Si plusieurs mécanismes d'inférence disponibles pour un attribut dans une classe → le mécanisme le plus prioritaire est déclenché en premier

Conflits d'héritage

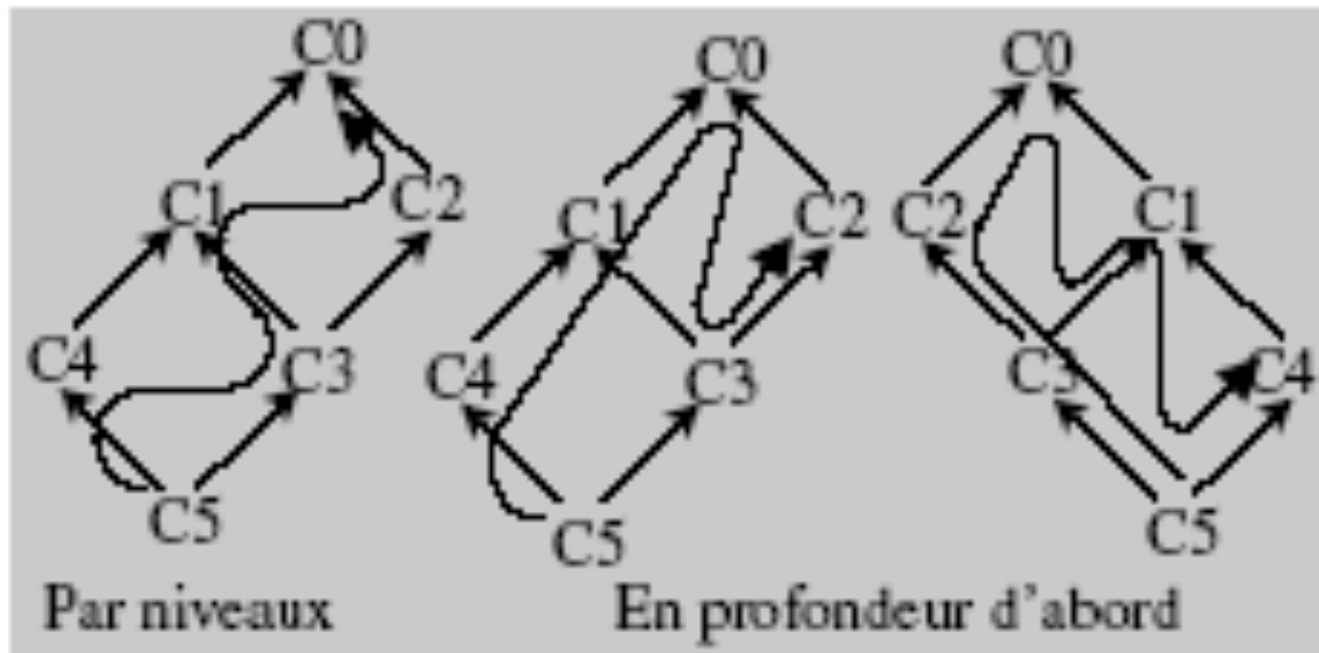
→ La multi-spécialisation pose des problèmes...



→ Certains systèmes la prennent en compte en proposant des stratégies de résolution de conflits

Stratégies de Résolution de Conflits d'héritage

→ Linéarisation globale



→ Linéarisation locale à chaque classe

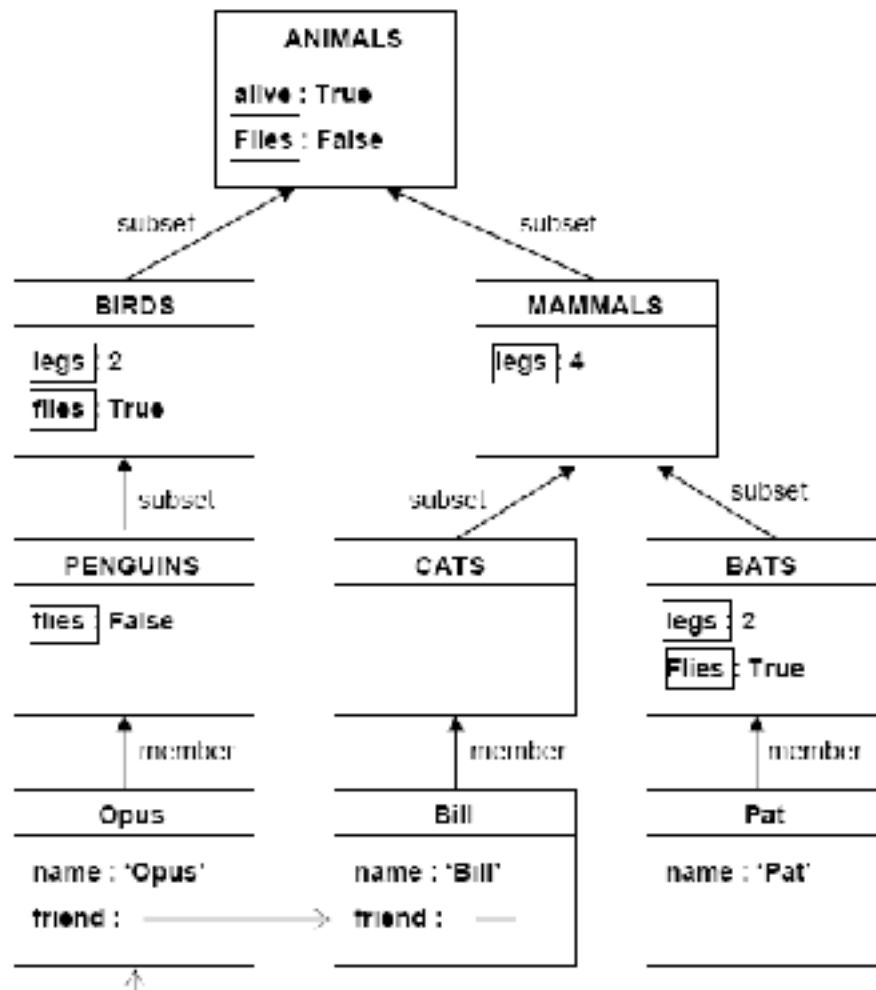
→ Linéarisation locale à chaque attribut

Les différents types de facets

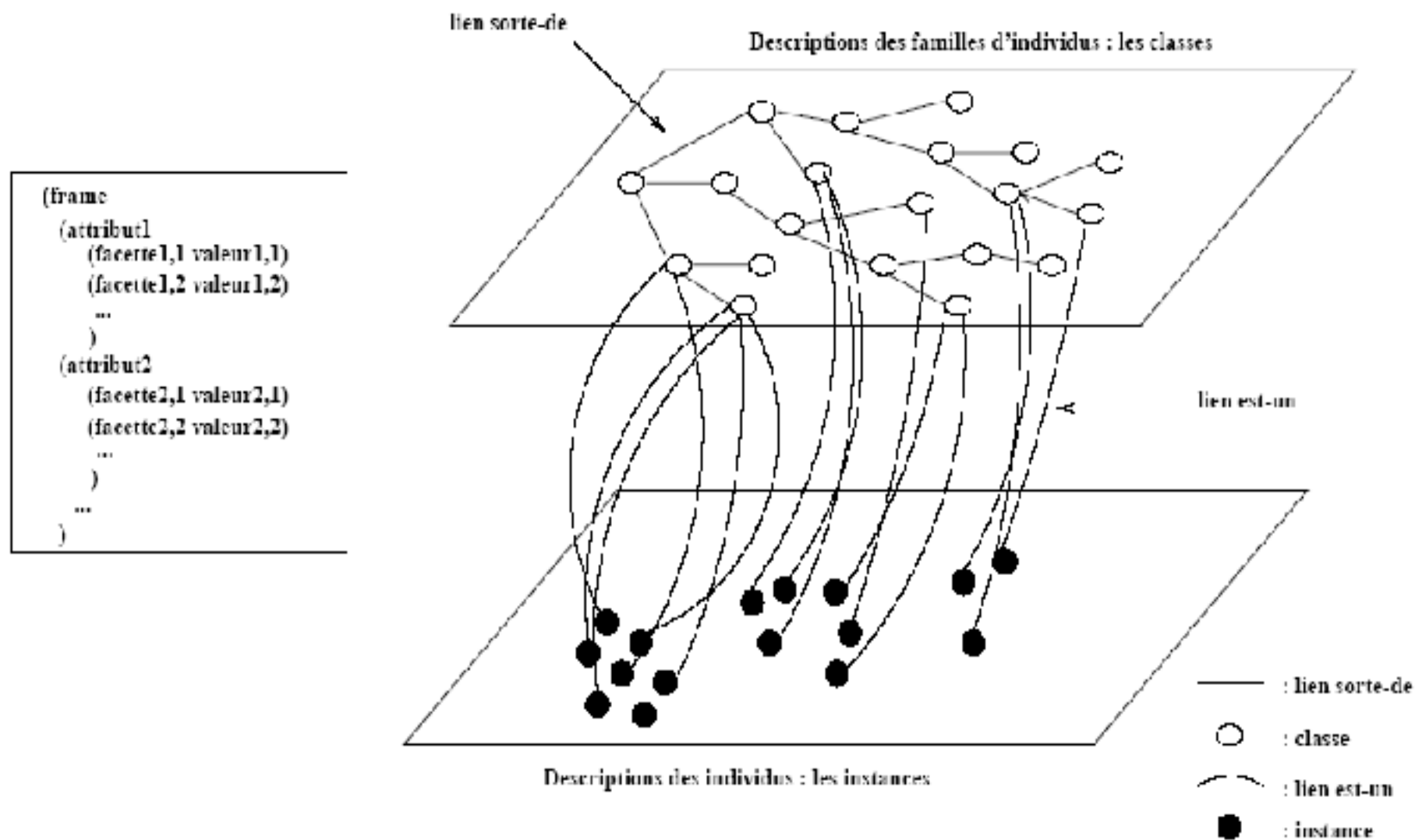
- Facet déclaratives :
 - Type `$one` : type intensif (Integer, ...) ; `$oneof` : type extensif (liste de valeurs possibles)
 - Restrictions de type : `$domain`, `$interval`
 - Cardinalité : `$min`, `$max`
 - Valeur : `$default`, `$value`
- Facet procédurales :
 - Si-besoin : calcule la valeur de l'attribut associé
 - Si-possible : vérifie une valeur avant de la placer dans un attribut
 - Si-ajout : est activé comme un démon après une modification de valeur
 - Si-enlève : est activé comme un démon après une suppression de valeur

L'accès aux valeurs des attributs en mode lecture et/ou écriture provoque des réactions (appelés réflexes) de la part des frames qui sont spécifiés dans la partie procédurale.

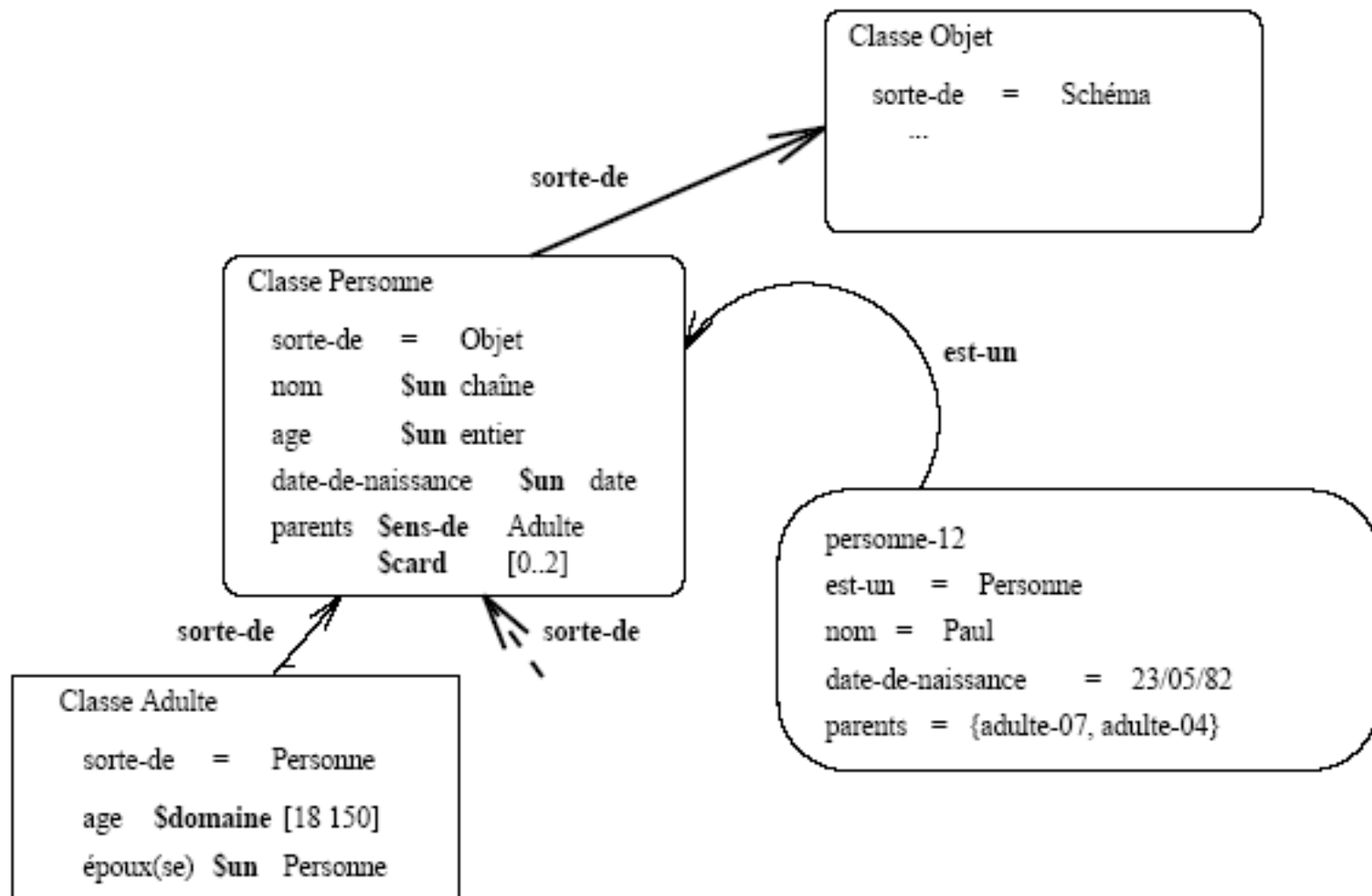
Exemple de Frames



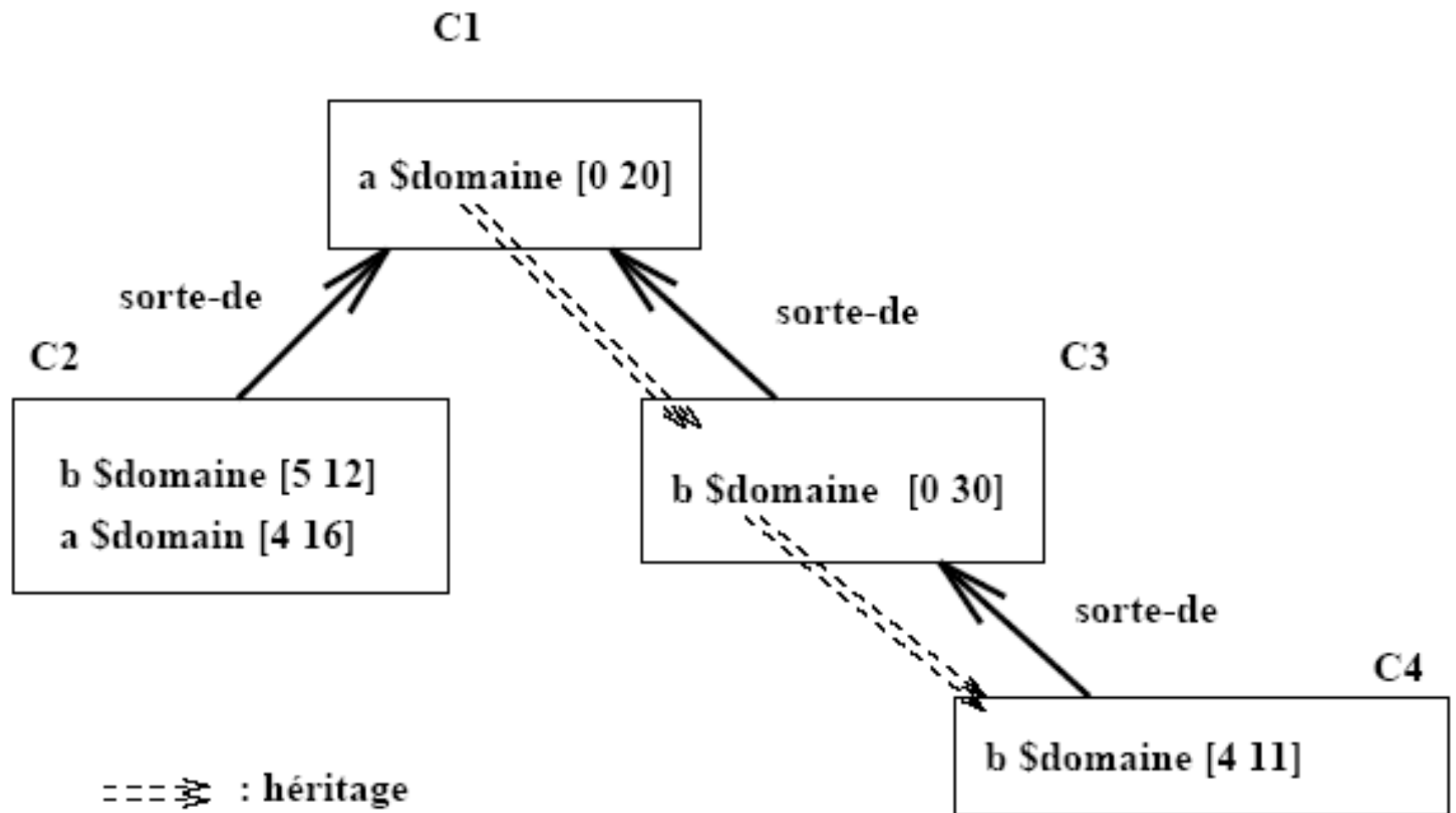
Description d'un frame



Classe et Instances



Héritage



Objets composites

