



Cours 9 - Les Représentations Objets (II)

MARIE-HÉLÈNE ABEL

Table des matières

I - Les logiques de description	5
A. Le langage KL-ONE.....	5
B. Caractéristiques des logiques de description.....	6
1. Représentation bipartite.....	6
2. Primitives épistémologiques.....	6
C. Mise en œuvre dans KL-ONE.....	9
1. Les concepts.....	9
2. Les rôles.....	10
3. Synthèse.....	13
II - Les graphes conceptuels	15
A. Introduction.....	15
B. Définition formelle et notations.....	15
1. Définition formelle.....	15
2. Deux notations.....	16
C. Positionnement.....	17
1. GC et Frames.....	17
2. GC et logique.....	17
D. Mise en œuvre.....	18
1. Les concepts.....	18
2. Les relations conceptuelles.....	18
3. Les graphes canoniques.....	19
4. Les opérations.....	20
III - Les langages à objets	23
A. Introduction.....	23
B. Principe de base : la classe.....	24
C. Synthèse.....	25
D. POO versus Programmation classique.....	25
1. Rappel de la programmation "classique".....	25
2. Comparaison POO/Programmation "classique".....	26

- Les Logiques de Description, Brachman et Schmolze [77], offrent un modèle à mi-chemin entre les réseaux sémantiques et les Frames de Minsky.
- Le premier représentant s'appelle KL-ONE.

A. Le langage KL-ONE

Préambule

- KL-ONE a été créée par Brachman en 1985.
- Objectif : langage de représentation dont la sémantique est bien fondée, c'est-à-dire externe à la représentation et aux algorithmes qui opèrent sur celle-ci.
- Domaines d'application initiaux : compréhension de la langue naturelle, génération automatique de langue naturelle, etc.



Définition : KL-ONE est un langage de représentation des connaissances à la frontière des réseaux sémantiques et des frames :

- Langage de représentation de concepts proche de la notion de sémantique lexicale ;
- Les concepts de KL-ONE sont munis d'attributs appelés rôle ;
- Des contraintes sur le type et la cardinalité de sa valeur sont associées à tout rôle, à la manière des facets d'un langage de frames.



Remarque : KL-ONE est basé sur la subsumption de termes.

C'est un système à héritage structuré.

KL-ONE Family

KL-ONE est à l'origine de nombreux autres langages :

BACK, CANDIDE, CLASSIC, DRL, KANDOR, KL-CONC, KL-MAGMA, KL-TWO, KNET, KREME, K-REP, KRYPTON, L-LILOG, LOOM, MANTRA, MESON, NIKL, SB-ONE, SPHINX, DAML+OIL, OWL

B. Caractéristiques des logiques de description

Les principales caractéristiques sont une représentation bipartite et l'usage de primitives épistémologiques

1. Représentation bipartite



Définition

- Partie descriptive appelée Tbox pour « Terminological Box ».
- Partie assertionnelle appelée Abox pour « Assertional Box ».



Attention

KL-ONE n'est pas bipartite, c'est un langage de description de concepts, sous la forme de prédicats, sans assertions sur le monde.

Raison d'être de Krypton :

Composant terminologique et assertionnel fondé sur la logique des prédicats.

- Les opérations sur les connaissances et les « faits » ne sont pas les mêmes.
- Induit des problèmes : liaison entre les deux composants.

2. Primitives épistémologiques

Une logique de description fournit des primitives épistémologiques (primitives de représentation) :

- Des concepts primitifs ou définis,
- Des rôles,
- Des opérations (and, or, not, some, all, atleast, atmost...) sur ces éléments primitifs du langage,
- Un mécanisme de classification fondé sur la relation de subsumption entre concepts ou rôles.
-> Cela induit généralement un graphe orienté sans circuit entre les concepts et entre les rôles.



Définition : Concepts primitifs

Ce sont des définitions incomplètes des concepts.

-> Conditions Nécessaires mais non suffisantes.



Exemple : Définition du concept primitif polygone

- Polygone peut être défini comme une figure géométrique avec N côtés qui sont des segments de droite.
- Nous avons des CN mais non suffisantes pour définir qu'un individu appartient à cette classe : toute figure non fermée satisfait ces conditions.



Définition : Concepts définis

Ce sont des définitions complètes des concepts.

-> Conditions Nécessaires et Suffisantes (CNS).



Exemple : Définition du concept défini triangle

Triangle peut être un concept défini, dérivé de Polygone (un triangle est un

polygone à exactement trois côtés).



Attention

La classification ne peut opérer que sur des concepts définis.



Exemple : Définition du concept moto

(Cprim Véhicule
(and
 (atleast 1 Roue)
 (atleast 1 Moteur)))

```
(Cdef Moto
  (and
    (atleast 2 Roue)
    (atleast 1 Moteur)))
```

Moto est un Véhicule ?

image



Exemple : Définition2 du concept moto

(Cprim Véhicule
 (and
 (atleast 1 Roue)
 (atleast 1 Moteur)))

(Cdef Moto
 (and
 Véhicule
 (atleast 2 Roue)))

Moto est un Véhicule ?

image2

C. Mise en œuvre dans KL-ONE

Illustrons la représentation des concepts et des rôles avec KL-ONE

1. Les concepts

Les concepts sont organisés sous la forme d'une taxinomie, éventuellement multiple, utilisant le lien de subsumption de façon à permettre l'héritage de leurs propriétés.

Deux types de concepts :

- Concept générique : extension = plusieurs individus

- Concept individuel : extension = un individu (exemples : le roi de France, étoile du soir, étoile du matin)

Taxinomie

Deux liens particuliers permettent d'organiser la taxinomie de concept :

- Lien "sorte-de" : permet de relier des concepts génériques.
- Lien "est-un" : correspond à la relation d'instanciation entre un concept générique et un concept individuel.

2. Les rôles



Définition : Un rôle

- Permet de mettre en relation des concepts.
- Permet d'exprimer les conditions de la définition d'un concept.
- Peut être générique ou individuel.



Attention

- Les rôles génériques ne relient que des concepts génériques.
- Les rôles individuels permettent de relier des concepts individuels.



Remarque : Tous les rôles peuvent être munis de propriétés

- Restriction de valeur.
- Restriction de nombre.
- Différentiation.
- Description structurelle.

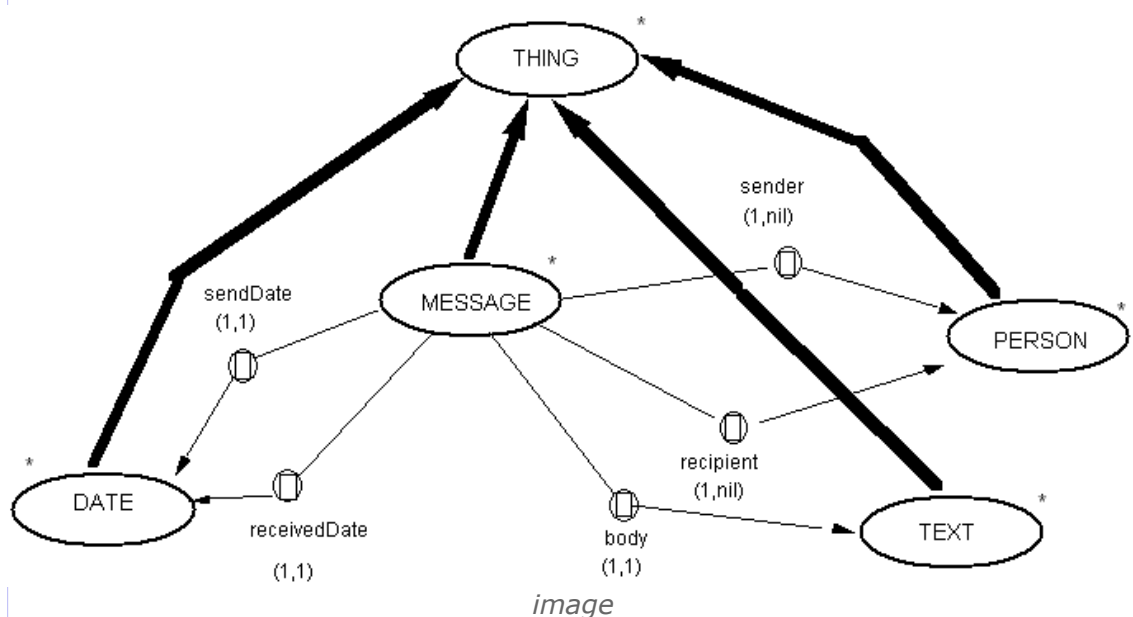


Définition : Rôle générique

- Un rôle générique est décrit ou spécifié par une restriction de valeur et possède une restriction de nombre ou cardinalité $[n, m]$.
- Les restrictions de valeurs sont de type nécessaire.



Exemple



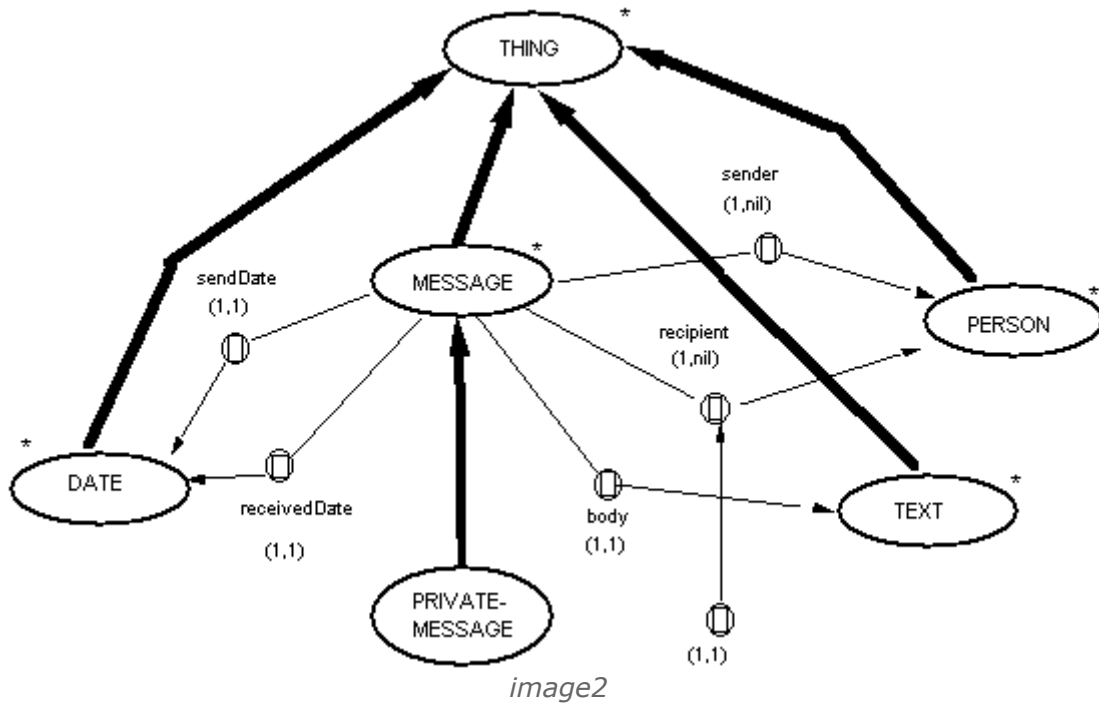


Définition : Restriction de rôle

Jouent un rôle similaire à la relation de subsomption, mais se rapportent aux rôles génériques



Exemple

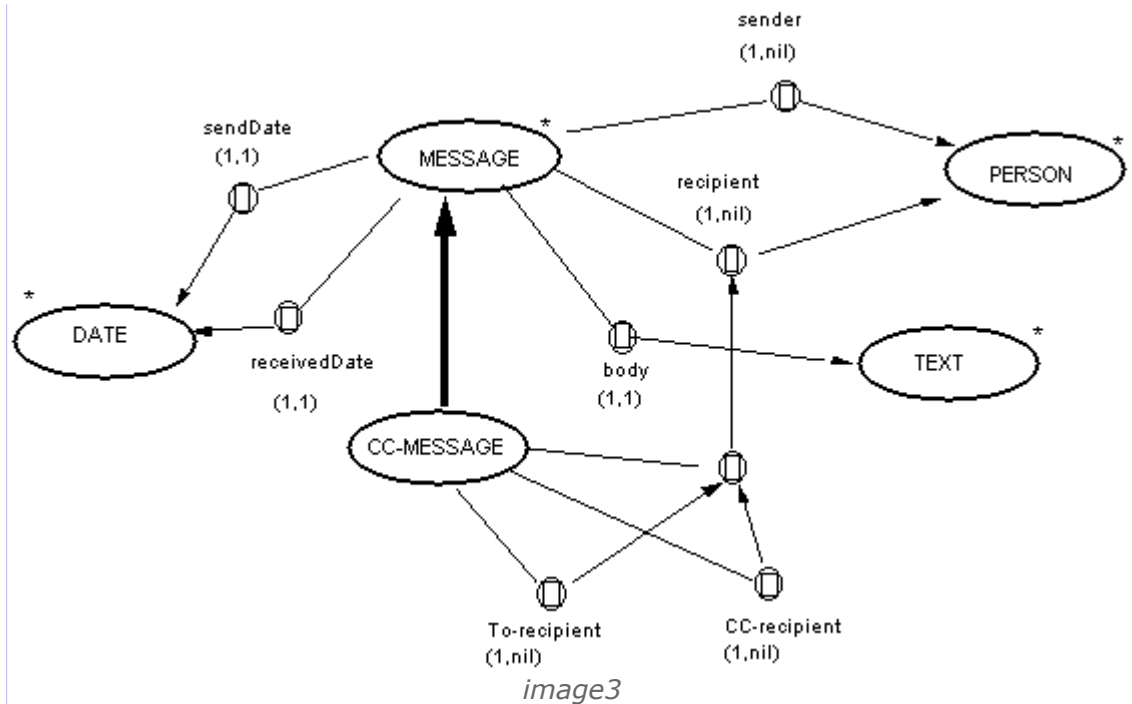


Définition : Différentiation de rôles

Un rôle générique différencie un autre s'il dénote une sous-relation de la relation dénotée par l'autre



Exemple



3. Synthèse

- KL-ONE est un langage de description de concepts sous la forme de prédicats.
- Il ne comporte que des connaissances terminologiques : Triangle = polygone à trois côtés

Brachman préconise l'utilisation de connaissances assertionnelles, ce qu'il applique dans le système KRYPTON :

- TBox : logique des prédicats, c'est la terminologie qui décrit les propriétés du monde en intension.
- ABox : proche de la logique du premier ordre, c'est l'ensemble des connaissances concernant les individus peuplant effectivement le monde.

Les graphes conceptuels



Introduction	15
Définition formelle et notations	15
Positionnement	17
Mise en œuvre	18

A. Introduction

La théorie des Graphes Conceptuels, développée par SOWA (1984), est un formalisme général de représentation des connaissances fondé sur la logique.

Langage de représentation des connaissances basé sur :

La linguistique,
La psychologie,
La philosophie.

Objectif

Développer un système de logique qui représente de façon simple et pratique les structures du langage naturel.

B. Définition formelle et notations

Après avoir précisé la notation formelle des GCs, nous en donnons deux façons de les noter.

1. Définition formelle



Définition

- Les Graphes Conceptuels sont des graphes finis, connexes et bipartites.
- Les deux sortes de nœuds d'un graphe bipartite sont les concepts et les relations conceptuelles.
- Chaque relation conceptuelle possède un arc ou plus, chacun d'entre eux doit être lié à un concept.



Remarque

- Si une relation a n arcs, elle est dite n -adique.

- Un unique concept peut former un graphe conceptuel, mais chaque arc de chaque relation conceptuelle doit être lié à un concept quelconque.

2. Deux notations



Définition : Notation graphique

- Concepts : rectangles
- Relations : ovales
- Arcs : flèches



Exemple

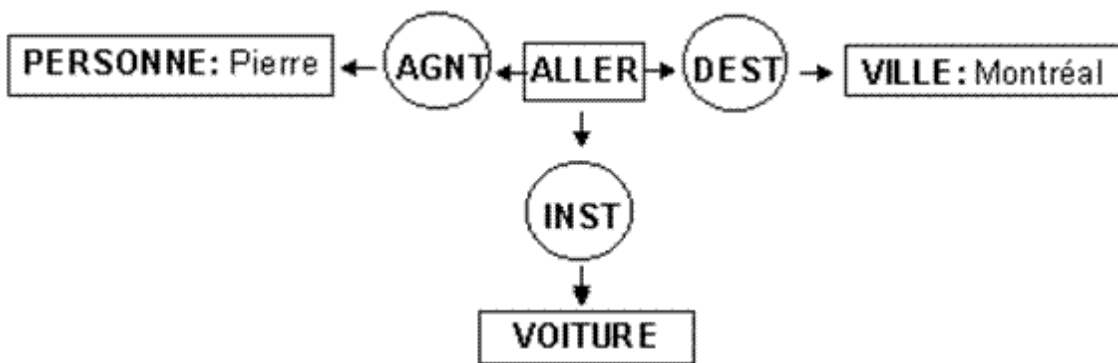


image4

Notation linéaire

- Concepts : entre crochets
- Relations entre parenthèses
- Arcs : flèches



Exemple

[ALLER]-
(AGNT) ->[PERSONNE: Pierre]
(DEST) ->[VILLE: Montréal]
(INST) ->[VOITURE]

image3

C. Positionnement

Précisons les liens qui existent entre les GCs et :

- les Frames
- la logique

1. GC et Frames

Un G.C. ressemble à un Frame :

- Les relations conceptuelles s'apparentent aux "slots".
- Les étiquettes de type s'apparentent aux contraintes.
- Les noms Pierre et Montréal s'apparentent aux contenus des "slots".



Remarque

Contrairement aux Frames qui ne savent représenter qu'une sous section de la logique, les G.C. sont plus généraux et permettent de représenter toute la logique.

2. GC et logique

Les G.C. reposent sur un système de logique et peuvent ainsi être représentés par la logique des prédicats.



Exemple : L'analyse de l'exemple précédent révèle :

- Prédicats : $\text{personne}(x)$, $\text{aller}(x)$, $\text{ville}(x)$, et $\text{voiture}(x)$.
- Les relations conceptuelles : $\text{agnt}(x,y)$, $\text{dest}(x,y)$, $\text{int}(x,y)$.



Exemple : L'expression de ce graphe en logique des prédicats du premier ordre donne donc l'expression suivante :

$$(\exists x) (\exists y) (\text{personne}(\text{Pierre}) \wedge \text{aller}(x) \wedge \text{ville}(\text{Montréal}) \wedge \text{voiture}(y) \\ \wedge \text{agnt}(x, \text{Pierre}) \wedge \text{inst}(x, y) \wedge \text{dest}(x, \text{Montréal})).$$

image5

D. Mise en œuvre

Illustrons la représentation des concepts, relations conceptuelles, les graphes canoniques et certaines opérations.

1. Les concepts



Définition

Un concept est un couple : (type, référent)



Exemple : Concept générique

[personne] ou [personne : *]



Exemple : Concept individuel

[personne : #1]

-> la personne ou cette personne



Exemple : Nom propre

[personne : Jean]



Exemple : Ensemble générique

[personnes : {*.}]



Exemple : Ensemble de cardinal spécifié

[personne : {*.}@3]

-> 3 personnes



Exemple : Mesure associée à une unité

[hauteur : @3m]



Exemple : Ensemble défini en extension

[personne : {Jean, Marc, Arthur}]

2. Les relations conceptuelles



Définition : Agent

(agnt) relie [action] à [animé]

-> Le concept animé représente l'acteur de l'action



Définition : Experienceur

(expr) relie [etat] à [animé]

-> Le concept animé ressent cet état (Marie a chaud)



Définition : Instrument

(inst) relie [entité] à [action]

-> Entité est impliquée de manière causale



Définition : Objet

(obj) relie [action] à [entité] sur laquelle porte l'action

-> Le chien mange un os (os = objet)

3. Les graphes canoniques



Définition

Les graphes canoniques définissent des conditions d'emploi des concepts ou des possibilités de combinaisons de concepts.



Remarque

Les autres graphes seront dérivés à partir des graphes canoniques.



Exemple : Interne

[interne]-
 (agnt)<-[pratiquer]-
 (obj)->[medecine]
 (lieu)->[hopital]

image4



Exemple : Juger

[juger]-
 (agnt) -> [personne]
 (obj)-> [proposition]

image5



Exemple : Les internes jugent une proposition importante

[juger]-

(agnt) -> [interne]

(obj) -> [proposition]-

(attr) -> [important]

image6

4. Les opérations

Les opérations sur les graphes permettant de manipuler les GC lors d'une analyse sémantique ou d'exprimer des phrases plus ou moins concises lors d'une génération



Exemple

- Copie
- Restriction
- Simplification
- Jointure
 - Jointure maximale
 - Jointure dirigée



Les langages à objets



Introduction	23
Principe de base : la classe	24
Synthèse	25
POO versus Programmation classique	25

A. Introduction

L'origine de langage objet peut être trouvée dans les réseaux sémantiques

En fixant une relation hiérarchique de base (est-un) et une relation d'appartenance, on a la structure d'un langage objet.



Remarque : Les autres relations se cachent dans les attributs des entités.

- Les entités hiérarchisées sont les classes (avec au sommet une classe racine souvent appelée OBJET).
- Les autres entités, les objets proprement dits, appartiennent aux différentes classes.

Types d'implémentation

L'idée de langage à objets connaît différents types d'implémentation ayant chacun ses caractéristiques avec quelques dénominateurs communs.



Remarque : Les premiers objets ont été définis comme une structure de données en LISP (les flavors)

SMALLTALK est un des premiers langages qui a mis en œuvre cette méthode et constitue le modèle prototypique et le plus complet des langages à objets.



Remarque : Des langages classiques ont ajouté cette possibilité à leur noyau habituel

- Turbo
- Pascal
- Cobol
- C

...



B. Principe de base : la classe



Définition : Une classe est définie par :

- des données (variables d'instance, champs)
- des méthodes : procédures liées à la classe, identifiées par un sélecteur.



Attention

- Pour utiliser les méthodes, il est nécessaire d'envoyer des messages aux objets de la classe.
- En principe, aucun accès direct aux données d'un objet n'est possible (encapsulation des données).



Exemple : Classe rectangle

- Variables
 - (longueur (valeur : un nombre))
 - (largeur (valeur : un nombre))
 - (coin (valeur : un point))
- Méthodes
 - (aire (longueur * largeur))
 - (display (draw-rect (coin longueur largeur)))
 - (init (a b p) (longueur <- a)(largeur <- b)(coin <- p))



Exemple : Classe point

- Variables
 - (cx (valeur : un nombre))
 - (cy (valeur : un nombre))
- Méthode
 - (display (plot (cx cy)))
 - (init (a b) (cx <- a)(cy <- b))



Exemple : Définition des objets rec-1 et pointA, instances des classes rectangle et point

```
(new rectangle rec-1)
(new point pointA)
```



Exemple : Envoi de messages

```
(send pointA init 25 10) ; ce message initialise les valeurs de pointA
(send rec-1 init 25 10 pointA) ; ce message initialise les valeurs de rec-1
(send aire rec-1); ce message permet d'obtenir la valeur de l'aire de rec-1 : 250
```



Définition : Polymorphisme

La méthode **init** est différente selon l'objet auquel elle s'applique



Définition : Héritage

Une classe peut être définie par rapport à une classe déjà existante.



Exemple : Classe rectangle-plein (rectangle)

- Variables
 - (couleur (possibilités : jaune bleu rouge))

- Méthodes
 - (display (super display)(fill couleur))
 - (init (a b p c)(super init a b p)(couleur <- c))



Remarque

Cette nouvelle classe possède les propriétés et les méthodes de la classe rectangle

C. Synthèse

Classe

Un objet est déterminé par différents attributs définis dans sa classe. Chaque attribut peut être typé.

Héritage

Avec la possibilité de définir des classes "emboîtées", on utilise de façon fondamentale les mécanismes d'héritage.

Méthodes

Les procédures associées aux classes spécifient le comportement de ces dernières : ce sont les méthodes.

Messages

Les objets communiquent par l'envoi de messages qui invoquent les méthodes à mettre en œuvre.

D. POO versus Programmation classique

Après un bref rappel de ce qu'on entend par programmation "classique", nous effectuons un positionnement entre les deux approches suivies.

1. Rappel de la programmation "classique"



Définition

Par programmation classique nous regroupons les programmations procédurale, fonctionnelle, logique et déclarative.

Organisation des programmes

Les programmes sont découpés en modules ou fonctions qui reçoivent des paramètres en entrée et retournent des valeurs en sortie.



Méthode

Le programmeur doit choisir quelle fonction doit être appelée en fonction de ce qu'il faut faire et parfois le type des paramètres.



Remarque : Cela implique que :

- Le programmeur doit se rappeler le nom des différentes fonctions et sélectionner la bonne.
- Si le type de l'argument change, il faut éditer les appels correspondants

dans le programme.

- Si le travail est fait par le compilateur, l'ajout d'un nouveau type ou l'extension d'une fonction à un type existant ne peut être réalisée sans une modification du compilateur.

2. Comparaison POO/Programmation "classique"

Programmation classique

Le contrôle se fait au niveau des appels de fonctions :

- On choisit la fonction à exécuter,
- Le contrôle est passé à la routine,
- Puis les valeurs sont passées en arguments.

POO

La programmation objet consiste à inverser les priorités de la programmation classique et à privilégier l'argument par rapport à la fonction.



Méthode

L'argument connaît son type et peut donc choisir la bonne séquence de code à exécuter.