

FRL



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

PLAN

1. Introduction : Notions de Representation de la connaissance
2. Frames
3. Concepts de base d'un langage à base de frames
4. Implementation d'un langage à base de Frames

Spécifications

Un frame est une structure de données qui représente une situation caractéristique ou prototypique

- Elle se compose de plusieurs éléments, appelés slots ou éléments terminaux (sur différents niveaux).
- Un frame est une sorte de réseau de relations et de noeuds.
- Chaque slot peut être lui-même un frame.
- Chaque slot peut prendre une valeur par défaut qui est remplacée afin de mieux cadrer avec la réalité et servir de variable.

Spécifications

Un frame est une structure à trois niveaux qui sont emboîtés : le triplet (frame, slot, facette)

❖ (Frames

```
(slot1      (facette_value)
             (facette_value)
             ... )
(slotn      (facette_value)
             (facette_value)))
```

❖ Un slot décrit les différentes propriétés d'un frame

❖ Une facette est une modalité descriptive ou comportementale d'un slot. A une facette est toujours associée une valeur.

Spécifications



Facettes déclaratives et procédurales.

- Facettes de typage : \$un, \$liste-de, \$domaine, \$sauf, \$intervalle, \$a-verifier, \$card-min, \$card-max
- Facettes de valeurs : \$valeur, \$default, etc. ...
- Facettes procédurales : \$si-besoin, \$si-possible, \$si-ajout, \$si-enlève, \$si-succes \$si-echec \$avant-ajout \$apres-ajout \$avant-mod \$apres-mod \$avant-sup \$apres-sup \$sib-filtre etc. ...

Exemples de Frames

➤ (date

sorte-de \$valeur

objet

jour

\$un

entier

\$intervalle

[1 31]

mois

\$un

chaîne

\$domaine

"janvier" "février" "mars" "avril" "mai"

"juin" "juillet" "août" "septembre" "octobre" "novembre" "décembre"

année \$un entier)

➤ (date-23

est-un \$valeur date

jour

\$valeur 15

mois

\$valeur "avril"

année

\$valeur 1987)

Exemples de Frames

- (division-euclidienne
sorte-de \$valeur méthode
dividende \$un entier
diviseur \$un entier
\$sauf 0
reste \$un entier)
- Lors de la lecture d'un slot, la valeur peut être obtenue par la facette \$valeur, \$default ou \$si-besoin.
 - ❖ Mais avec le mécanisme d'héritage, comment se comportent ces trois facettes si elles sont présentes pour un slot donné dans tous les frames plus généraux que celui considéré

Lecture d'un slot

- Lecture d'un slot peut entraîner le parcours de la hiérarchie considérée.
- Trois stratégies principales : en I, en Z et en N.
 - ❖ En I, seules les facettes \$valeur sont consultées dans la hiérarchie de l'objet.
 - ❖ En Z, les facettes \$valeur, \$default et \$si-besoin sont consultées dans cet ordre à chaque niveau hiérarchique du bas vers le haut.
 - ❖ En N, les facettes \$valeur sont consultées sur toute la hiérarchie (du bas vers le haut) puis on fait de même avec les facettes \$default puis \$si-besoin.

Ecriture d'un slot



- Mise en place de la valeur et enfin propagations des modifications, déclenchement des réflexes si-ajout dans l'ordre descendant de la hiérarchie.

Suppression d'une valeur d'un slot



- il y a d'abord l'effacement de la valeur puis propagation des réflexes "si-enlève" dans l'ordre ascendant de la hiérarchie.

Modèle de Frames

❖ Modèle Classes/Instances

(meuble

sorte-de \$valeur objet)

(table

sorte-de \$valeur meuble

forme \$un chaîne

\$domaine "ronde" "ovale" "rectangulaire" "carrée"
"triangulaire"

nbr-pieds \$un entier

\$intervalle [1 15]

composé-de \$liste-de \$un plateau \$liste-de pieds)

(plateau sorte-de \$valeur objet)

(pieds sorte-de \$valeur objet)

Modèle de Frames

❖ Modèle Prototype

```
( pieds
  sorte-de $valeur objet )
( meuble
  sorte-de $valeur objet )
( table
  sorte-de $valeur meuble
  forme $un chaîne
           $défaut "rectangulaire"
  nbr-pieds $un entier
              $défaut 4
  composé-de $liste-de $un plateau $liste-de pieds )
( plateau sorte-de $valeur objet )
```

Concepts à implémenter

- z Définition d'une structure informatique permettant de :
 - y Stoker des frames
 - y 4 niveaux (frame, slot, facet, valeur)
- z Possibilité de définir des frames prototype
- z Possibilités de définir des frames instance
- z Création d'instances à partir de prototype
- z Gestion de la hiérarchie entre prototypes :
 - y héritage de propriétés
 - y masquage de l'information
- z Gestion des valeurs exactes et valeurs par défaut
- z Gestion des reflexes : si-besoin, si-ajout, si-retrait

PLAN



1. Introduction : Notions de Representation de la connaissance
2. Frames
3. Concepts de base d'un langage à base de frames
4. Implementation d'un langage à base de Frames

Stratégie d'implémentation

- z Définir votre langage de manipulation de frames comme sur-langage de SCHEME :
 - y Choix d'une structure informatique en SCHEME permettant d'implémenter un frame.
 - y définir un ensemble de fonctions qui permet de traiter les points précédents

Stratégie d'implémentation

- z Partir des listes de propriétés de Lisp et des A-listes afin de définir une structure informatique permettant d'implémenter des Frames.
- z Les listes de propriétés sont manipulées par les fonctions :
 - y getprop
 - y put
- z Les A-listes sont manipulées par la fonction : assoc.

Stratégie d'implémentation

z On associera à la propriété frame d'un atome une liste décrivant un frame.

z Exemple pour représenter le frame suivant :

```
Henry  ako    =    homme
      age    default  25
      si-besoin demande-age
```

On écrira :

```
(put 'henry '(henry (ako (= (homme))) (age (default (25))
      (si-besoin (demande-age)))) 'frame)
```

Stratégie d'implémentation

z Intérêt :

y (getprop 'henry 'frame)

y Retourne : (henry (ako (= (homme))) (age (default (25))
(si-besoin (demande-age))))

z Remarque :

(cdr (getprop 'henry 'frame))

Retourne ((ako (= (homme))) (age (default (25)) (si-besoin (demande-age))))

z Donc (assoc 'age (cdr (getprop 'henry 'frame)))

Retourne (age (default (25)) (si-besoin (demande-age)))

z Même raisonnement pour les facettes.

Spécification du langage : Variables globales

- Z ***frames*** : liste des noms de tous les frames du système
- Z ***frame** : le nom du frame courant. Cette variable est particulièrement utile au programmeur lorsque celui ci définit des démons. Effet il peut se révéler utile de connaître le nom du frame qui a initié un déclenchement du démon (frame tient le rôle de la variable *self* ou *this* dans les langages orientés objets).
- Z ***slot** : contient le nom du slot courant. Cette variable joue un rôle similaire à celui de la variable ***frame** et est affectée aussi lors du déclenchement de démons.
- Z ***facet** : contient le nom de la facette courante. Cette variable est utilisée dans le même esprit que les deux précédentes.
- Z ***value** : contient la valeur courante. Cette variable est comme les trois précédentes utile pour l'écriture de démons par l'utilisateur 'spécialement pour les démons de type if-added ou if-removed.

Spécification du langage : Frames et slots prédéfinis

z Frame prédéfini :

Les frames créés par l'utilisateur doivent s'insérer dans la hiérarchie des frames existantes : pour cela le système crée à l'initialisation un premier frame racine de l'arborescence : le frame **Objet**. Donc tous les frames créés par l'utilisateur seront des descendants de **Objet**.

z Slots prédéfinis offerts à l'utilisateur :

- y Le slot **AKO** : définition du graphe d'héritage
- y Le slot **CLASSIFICATION** : permet de distinguer les frames prototypes des frames instances.

Spécification du langage : Facettes prédefinies

1. **Value** : permet de définir une valeur exacte pour un slot donné d'un frame donné.
2. **default** : permet de définir une valeur par défaut pour un slot donné d'un frame donné
3. **if-added** : permet de spécifier qu'un démon doit être déclenché dès qu'une valeur est ajouté dans le slot du frame qui possède cette facette.
4. **if-removed** : permet de spécifier qu'un démon doit être déclenché dès qu'une valeur est effacée dans le slot du frame qui possède cette facette.
5. **if-needed** : permet de spécifier q'un démon doit être déclenché dès que l'on a besoin d'une valeur pour le slot du frame qui possède cette facette.