FRL

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

Session type en FRLC Etape 1

1. **Charger FRLC**

A partir de l'interpréteur SCHEME : charger FRLC par l'instruction load « FRLC.scm ».

Ceci a pour résultat :

1. d'ajouter au SCHEME l'ensemble des fonctions constituant la sur-couche
2. d'initialiser les variables système du langage
3. de créer le frame primitif Objet.

Session type en FRLC Etape 2

Développement d'une application en FRLC

Pour développer une application, l'utilisateur a alors le choix entre deux méthodes :

1. Ecrire sous un éditeur le script de son application à l'aide des fonctions de FRLC

2. L'utilisateur décrit les frames à l'aide de fonctions FRLC.

Utiliser un menu de construction de frames sous l'interpréteur Lisp (fonction Fmenu). Cette fonction permet de définir l'ensemble de l'arborescence des frames de l'application en décrivant leurs slots, facettes, valeurs.

Les différents types de fonctions (1)

- Fonctions de construction et destruction

Fput, Fput+, fremove, Fremove+,Fcreate,
Finst, Fgename

- Fonctions de consultation de valeurs

Fget, Fget-I,Fget-N, Fget-Z, Fgetclasses,
Fgetframes , Fchildren, Frame, Frame?, Fname,
Fname?, Finstance?, Fgeneric?, Fslots?, Ffacets?,

Les différents types de fonctions (2)

- Fonctions de vérification d'information

Flink?, Fako?, Fcheck

- Fonctions d'interface

Fmenu, Fwriteframe, Fimprim, Fsave, Fload

Conseil pour l'implémentation

D'autres fonctions peuvent être offertes telles que :

- fonctions de vérification des valeurs avec des contraintes associées grâce à des facettes prédéfinies telles que : required, intervalle, etc… ;
- fonctions de gestion des variables globales de l'utilisateur
- fonctions permettant d'implémenter l'envoi de messages et le déclenchement de méthodes.

Conseil pour l'implémentation : ordre

Fget, Fgetframe, Fput, Fremove, Fcreate, Finst,
Fput+, Fremove+
Fget-I, Fget-N, Fget-Z, Fgetclasses, Fgename,
Fchildren, Frame, Frames?, Fname, Fnames?,
Flink?,Fako?, Finstance?, Fgeneric?, Fcheck
Fmenu, Fwriteframe, Fimprim, Fsave, Fload
et
toutes autres fonctions que vous pouvez définir afin d'améliorer le langage.

Conseil pour l'implémentation

Bonne connaissance de Lisp :

1. Rplacd : remplace physiquement le cdr d'une liste par l'expression donnée en paramètre

Exemple :

>((define L '(a b c))

=(a b c)

>(rplacd L '(e f))

=(a e f)

> L

= (a e f)

2. Last : retroune le dernier élément d'une liste en paramètre sous forme de liste

>(last '(a b c))

=(c)

Fget

Syntaxe : (Fget <frame> <slot> <facet>)

Spécification : Recherche d'information. Elle retourne (sous forme de liste) la ou les valeurs associées au frame <frame> pour le slot <slot> et la facette <facet>. Fget retourne nil si le frame <frame<, le slot <slot> ou la facette <facet> n'est pas définie.

Fget

Exemple d'utilisation

>(fget 'henry 'age 'value)

=(23)

Si on avait avant (fput 'henry 'age 'value '23)

Fget

```
(defun fget (frame slot facet)
(mapcar ‘car
    (cdr (assoc facet
            (cdr (assoc slot
                    (cdr (get frame ‘frame)))))))
```

Fgetframe

Syntaxe : (fgetframe <frame>)

Spécification : Retourne la structure de liste liée au frame <frame> si le frame est définie et sinon initialise la variable *frames* avec le nom du frame et positionne dans la propriété frame la liste : (<frame>)

Implémentation :

```
(defun fgetframe (frame)
(cond ((get frame ‘frame))
        (t (setq *frames* (append *frames* (list frame)))
           (putprop frame (list frame) ‘frame))))
```

Fput

Syntaxe : (fput <frame> <slot> <facet> <valeur>)

Spécification : permet de positionner la facette <facet> du slot <slot> du frame <frame> à la valeur <valeur>. Elle retourne nil si cette valeur est déjà positionnée et <valeur> sinon.

Implementation de Fput

```
(defun fput (frame slot facet valeur)
(cond ((null (fget frame slot facet)) nil)
        (t (fassoc valeur
                (fassoc facet
                   (fassoc slot
                      (fgetframe frame)))) valeur)))

(defun fassoc (cle aliste)
  (cond ((assoc  (cle (cdr aliste)))
          (t  (cadr (rplacd (last aliste) (list (list cle)))))))
```

Fcreate et Finst

```
(defun fcreate (frame name)
(fput name ‘ako ‘valeur frame)
(fput name ‘classification ‘valeur ‘instance))

(defun finst (frame name)
(fcreate frame name)
Etc…..
```