

Systemes Intelligents : Raisonnement et Reconnaissance

James L. Crowley

Deuxième Année ENSIMAG

Deuxième Sémestre 2005/2006

Séance 5

8 mars 2006

Exercice :

L'objet de cette exercice est de réaliser un système de classes permettant de répondre aux questions concernant les relations entre les membres d'une famille. Les réponses sont générées par des "handlers".

a) Définir la classe "personne" avec les slots "nom", "pere", "mere", "freres" et "soeurs" . Les slots freres et soeurs peuvent contenir une liste. Définir la classe "homme" (sous-classe de personne) avec le slot "epouse" et le slot "sexe" avec une valeur fixe "masculin". Définir la classe "femme" (sous-classe de personne) avec le slot "époux" et le slot "sexe" avec une valeur fixe "feminin".

b) Définir des "message-handlers" pour la classe PERSONNE qui déterminent la "gandmere" et le "grandpere" paternels.

c) Définir le message handler pour la classe PERSONNE qui détermine les oncles (freres du pere et freres de la mere). Aide : On peut composer une liste avec create\$. Ex : (a b c) <- (create\$ a b c)

d) Définir le message-handler pour la classe PERSONNE qui détermine la liste des noms de tous les grands-parents.

Tester les handlres avec :

(make-instance [A] of HOMME (nom A) (epouse []) (pere [F]) (mere [G])(freres []) (souers [B]))

(make-instance [B] of FEMME (nom B) (epoux []) (pere [F]) (mere [G])(freres [A]) (souers []))

(make-instance [C] of HOMME (nom C) (epouse []) (pere [J]) (mere [K])(freres []) (souers [D]))

(make-instance [D] of FEMME (nom D) (epoux []) (pere [J]) (mere [K])(freres [C]) (souers []))

(make-instance [E] of HOMME (nom E) (epouse []) (pere [O]) (mere [N])(freres [F]) (souers []))

(make-instance [F] of HOMME (nom F) (epouse []) (pere [O]) (mere [N])(freres [E]) (souers []))

(make-instance [G] of FEMME (nom G) (epoux []) (pere []) (mere [])(freres [H]) (souers []))

(make-instance [H] of HOMME (nom H) (epouse []) (pere []) (mere [])(freres []) (souers [G]))

(make-instance [I] of FEMME (nom I) (epoux []) (pere [T]) (mere [S])(freres []) (souers []))

(make-instance [J] of FEMME (nom J) (epoux []) (pere [T]) (mere [S])(freres []) (souers []))

(make-instance [K] of HOMME (nom K) (epouse []) (pere [W]) (mere [V])(freres []) (souers []))
(make-instance [L] of FEMME (nom L) (epoux []) (pere [W]) (mere [V])(freres [K]) (souers []))
(make-instance [M] of HOMME (nom M) (epouse []) (pere []) (mere [])(freres []) (souers [N]))
(make-instance [N] of FEMME (nom N) (epoux []) (pere []) (mere [])(freres [M]) (souers []))
(make-instance [O] of HOMME (nom O) (epouse []) (pere []) (mere [])(freres [P]) (souers []))
(make-instance [P] of HOMME (nom P) (epouse []) (pere []) (mere [])(freres [O]) (souers []))
(make-instance [R] of HOMME (nom R) (epouse []) (pere []) (mere [])(freres []) (souers [S]))
(make-instance [S] of FEMME (nom S) (epoux []) (pere []) (mere [])(freres [R]) (souers []))
(make-instance [T] of HOMME (nom T) (epouse []) (pere []) (mere [])(freres []) (souers [U]))
(make-instance [U] of FEMME (nom U) (epoux []) (pere []) (mere [])(freres [T]) (souers []))
(make-instance [V] of FEMME (nom V) (epoux []) (pere []) (mere [])(freres []) (souers []))
(make-instance [W] of HOMME (nom W) (epouse []) (pere []) (mere [])(freres []) (souers []))