

AIC-Automatisation avancée, intelligence artificielle et cognitive

Annexe - Archive Systèmes-experts

Jean-Daniel Dessimoz

heig-vd

Haute Ecole d'Ingénieurs et de Gestion
du Canton de Vaud

Institut d'
Automatisation
Industrielle



Hes·SO

Haute Ecole Spécialisée
de Suisse occidentale

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

1

AIC-Automatisation avancée, intelligence artificielle et cognitive

Contenu

- Introduction
- Notion de modèle ; métrique pour le traitement d'information et pour la cognitive
- Choix d'une structure de commande
- Intelligence artificielle
- Commande à logique floue
- Commande neuronale
- Commande multimodale
- Commande à algorithme génétique
- Robots mobiles autonomes
- Conclusion

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

2

Contenu des *Exposés et exercices*

Notion de modèle ; métrique pour le traitement d'information et pour la cognitive	4p
Choix d'une structure de commande	2p
Intelligence artificielle	2p
Commande à logique floue	2p
Commande neuronale	2p
Commande multimodale	2p
Commande à algorithme génétique	2p
Robots mobiles autonomes	4p
Réserve et contrôle continu (TE, corr.)	6p

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

3

Travaux de laboratoire associés

Estimation de grandeurs cognitives (essais en simulation avec programmes d'évitement d'obstacles)	AIC-1
Test d'intelligence artificielle selon Turing et utilisation d'Eliza	AIC-2
Commande neuronale	AIC-3
Commande à logique floue	AIC-4
Commande à algorithme génétique	AIC-5
Commande multimodale	AIC-6
Robots mobiles autonomes	AIC-7
Sur demande, l'étudiant peut échanger l'une des manipulations ci-dessus par un autre sujet (cf. manipulations LaRA)	

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

4

AIC-Automatisation avancée, intelligence artificielle et cognitive

Archives

- Systèmes-experts

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

5

Systèmes-experts

Systèmes-experts

- 1 Généralités
- 2 Exemple: Déplacement en ville
- 3 Exemple: Manutention de cubes en piles

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

6

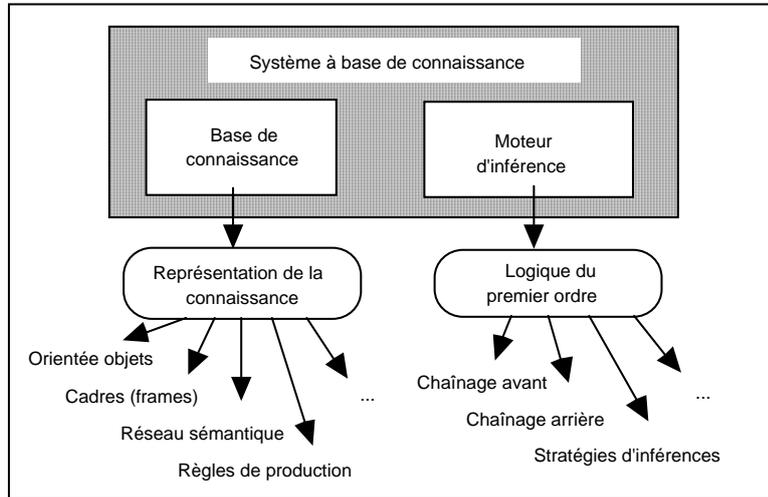
1 SE – Généralités (1 de 5)

- “Expert”: très bonnes performances, comparables à celle d’un spécialiste
- Systemes à bases de connaissances
- Systemes à base de règles
- Programmation déclarative, et non procédurale

1 SE – Généralités (2 de 5)

- Quel langage?
 - Langage procédural (C, Java, Basic, Pascal, etc.)
 - Langage spécialisé Prolog (Programmimg in Logic), Lisp
 - L4G (Personal consultant +..., etc.-obsolete)
 - Langage procédural (“beyond prototyping”) (C, Ada, Pascal, etc.)

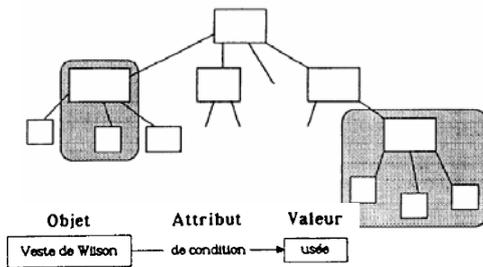
1 SE - Généralités (3 de 5)



HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

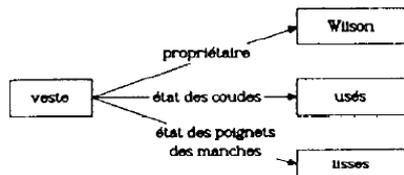
9

1 SE - Généralités (4 de 5)



Dualité de formalismes de représentation

Un objet est défini par plusieurs attributs:



Veste	
Compartiments	Valeurs
propriétaire	Wilson
état des manches	usé
état des poignets	usé
nombre de bras	défait. 2
poches	défait. oui
taille	si nécessaire, trouver la taille et le poids du propriétaire et utiliser la table 2

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

10

1 SE – Généralités (5 de 5)

Niveaux de réalisation

Example	Traditional products	Knowledge-based products	Example
ACME Co Accounting	Application	Packaged KBS	Planning advisor
Multiplan	Application generator	Shells	S.1
Pascal	High Level Language	Toolkits	KEE
68'000 Assembly	Assembly language	Object Oriented Language	LOOPS
Personal computer	Hardware	Progr. languages	Common LISP
		Hardware	Personal computer

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

11

2 SE – Exemple « en ville »

Règles de production (1 de 4)

<p>Règle 1 Si moyen = conduire et lieu = ville alors action = taxi</p>	<p>Règle 2 Si moyen = conduire et lieu = pas en ville alors action = conduire</p>
<p>Règle 3 Si moyen = marcher et temps = mauvais alors action = imperméable et marcher</p>	<p>Règle 4 Si moyen = marcher et temps = bon alors action = marcher</p>
<p>Règle 5 Si distance > 5 km et lieu=ville alors action = taxi</p>	<p>Règle 6 Si distance > 1 km et délai < 15 min alors moyen = conduire</p>
<p>Règle 7 Si distance > 1 km et délai > 15 min alors moyen = marcher</p>	<p>Règle 8 Si distance < 1 km alors moyen = marcher</p>

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

12

2 SE - Exemple « en ville »

Consultation

(2 de 4)

Question 1 Quelle est la distance?	Réponse 1 distance = 2 km
Question 2 Combien de temps avant la réunion?	Réponse 2 délai = 2 min.
Question 3 Où se trouve la réunion?	Réponse 3 lieu = ville

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

13

2 SE - Exemple « en ville »

Chaînage avant

(3 de 4)

Règles satisfaites	Faits en mémoire
	distance = 2 km délai = 10 min. lieu = ville
1er passage Règle 6 ->	moyen = conduire
2ième passage Règle 1 ->	action = taxi
3ième passage	pas de nouvelle conclusion

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

14

2 SE – Exemple « en ville »

(4 de 4)

Chaînage arrière

action?

Règle 1 -> action=taxi.

moyen?

lieu?

lieu=ville.

Règle 5 ----- Règle 6 -> moyen = conduire.

distance?

distance?

délai?

distance = 2 km.

délai= 10min.

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

15

Systemes-experts

3 SE – Exemple «CUBES»

3.1 Implémentation en Basic

3.2 Implémentation en Prolog

3.3 Implémentation en Lisp

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

16

3.1 SE-«CUBES»-Basic

(1 de 2)

Commandes:

<HLP>,<PRG>,<FIN>,<DBG>,<GO>,<Ctrl/C>

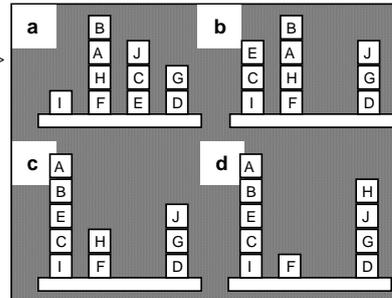
H B I
E A F
C D G J
1 2 3 4

Donnez la consigne sous la forme:

< nom_du_cube | pile | niveau >

Ex.: B32 ou FIN

Consigne: ?



HMI

Solution pour H44

3.1 SE-«CUBES»-Basic

(2 de 2)

```

6400 ! REGLE 8 : Tant qu'il y a un cube au-dessus de celui a prendre &
      ! alors c'est celui-la qui est a prendre.
6500 IF CUB_A_PR$<>"" &
      THEN WHILE
SEG$(PILE$(POSI_P%),POSI_N%+INC%,POSI_N%+INC%)<>"" &
\     POSI_N%=POSI_N%+INC% &
\     CUB_A_PR$=SEG$(PILE$(POSI_P%),POSI_N%,POSI_N%) &
\     PRINT " R8: >>>";POSI_N% IF DBG% &
\     PRINT " CUB_A_PR=";CUB_A_PR$ IF DBG% &
\     NEXT &
\     IF CUB_A_PR$<>CUBES$ AND DESTI_P%=PILE% AND
DESTI_N%=NIVEAU% & THEN UNTIL ALEA%<>PILE% AND ALEA%<>POSI_P%
AND
\     LEN(PILE$(ALEA%))<MAXNIVEAU% &
\     ALEA%=INT(RND*4) &
\     NEXT &
\     DESTI_P%=ALEA% &
\     DESTI_N%=LEN(PILE$(DESTI_P%))+INC%&
\     END IF &
\     PRINT " R8b: DESTI_P=";DESTI_P% IF DBG% &
\     PRINT " DESTI_N=";DESTI_N% IF DBG%&
\     PRINT " CUB_A_PR=";CUB_A_PR$ IF DBG% &
\     END IF &

```

3 SE – Exemple «CUBES»

Implémentation Pascal

(* Regle 5: si il n'y a pas de cube a la position de consigne
qu'il y en a un en dessous
ou que c'est le premier de la pile alors
c'est le cube de consigne qui est a prendre

*)

```

Cube_1 := Copy(Pile[PileCube-1],NivCube,1);
Cube_2 := Copy(Pile[PileCube-1],NivCube+Dec,1);
If (Cube_1 <> "") and (Cube_2 <> "") or (NivCube = 1) Then
Begin
  Cube_A_Pr := Cube;
  Pile_Dest:= PileCube;
  Niv_Dest:= NivCube;
End;

```

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

19

3 SE – Exemple «CUBES»

Implémentation Prolog (1 de 5)

```

cube(a,1,1);
cube(b,1,2);
cube(c,1,3);
cube(d,1,4);
cube(0,1,5);
...

```

```

/ ?- deplace(a,3,4).
d
c
b f
a e
1 2 3 4
prendre le cube d
mettre le cube sur la pile 3
c
b f
a e d
1 2 3 4
...

```

```

...
prendre le cube a
mettre le cube sur la pile 3

```

```

a
b
c
d
1 2 3 4

```

```

Solution 1 :
ok
next solution ? Y/N y
One solution.
Frame stack size : 1281
Number of created (pushed)
frames : 1604

```

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

20

3 SE – Exemple «CUBES»

Implémentation Prolog (2 de 5)

```
/ ?- cube(Cube,2,Niveau).
Solution 1 :
  Cube=e,Niveau=1
next solution ? Y/N y
Solution 2 :
  Cube=f,Niveau=2
next solution ? Y/N y
Solution 3 :
  Cube=0,Niveau=3
next solution ? Y/N y
3 solutions.
Frame stack size : 1281
Number of created (pushed) frames : 1607
```

SI
- il existe un certain cube "C", dans
n'importe quelle pile et à n'importe
quel niveau,
- et qu'au niveau immédiatement
supérieur
- il y a une fin de pile, et ceci dans la
même pile

ALORS
ce cube, C, est accessible.

Règle R en pseudocode

Solutions Prolog pour variables (à majuscules)

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

21

3 SE – Exemple «CUBES»

Implémentation Prolog (3 de 5)

```
/*Clause définissant un cube accessible,
donc au sommet d'une pile.*/
cube_accessible(C):-cube(C,P,H),is(VAR,H + 1),cube(0,P,VAR).
```

Règle R en prolog (clause R)

POUR QUE
un certain cube, C, soit accessible
IL FAUT QUE
- il existe un certain cube "C", dans n'importe
quelle pile et à n'importe quel niveau,
- et qu'au niveau immédiatement supérieur
- il y ait une fin de pile, et ceci dans la même pile

Reformulation de la règle R en pseudocode de style prolog (clause R)

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

22

3 SE – Exemple «CUBES»

Implémentation Prolog (4 de 5)

```
/*Clause definissant l'initialisation du déplacement d'un cube. */  
deplace (C,P,H) -> affiche (10), deplacer_cube(C,P,H);
```

```
/*Clauses definissant le déplacement du cube C dans la pile P a */  
/*la hauteur H. */  
deplacer_cube(C,P,H):-cube(C,P,H),!,print("Le cube est deja en place") ;  
deplacer_cube(C,P,H):-cube(C,P,H1),cube(0,PT,HT),diff(P,PT),  
    rendre_cube_accessible(C,PT,HT),prendre(C,P),  
    mettre(C,PT),deplacer_cube(C,P,H),! ;  
deplacer_cube(C,P,H):-rendre_cube_accessible(C,P,H),cube(C,P1,H1),  
    rendre_position_accessible(P1,P,H),prendre(C,P1),  
    mettre(C,P),!  
deplacer_cube(C,P,H):-prendre(C,P1),cube(0,P2,_),diff(P2,P),  
    diff(P1,P2),mettre(C,P2),deplacer_cube(C,P,H),!;
```

Règle R2 en prolog (clauses R2i)

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

23

3 SE – Exemple «CUBES»

Implémentation Prolog (5 de 5)

POUR

déplacer un certain cube, C, sur la pile P au niveau H

IL FAUT

- qu'il y ait ce cube "C" dans la bonne pile "P", et ceci à n'importe quel niveau,
- qu'il y ait un sommet de pile dans une pile quelconque...
- ... mais différente de celle où est C,
- y déplacer les cubes qui gênent l'accès à C
- puis prendre C et l'amener sur cette pile temporaire
- et rappeler la clause déplacer_cube (implémentation récursive!)

*« Traduction » de la règle R2
originellement écrite en prolog (clauses R2i)*

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

24

3 SE – Exemple «CUBES»

Implémentation en Lisp (1 de 3)

```
;definition de la classe pour le cube
;;;
;;; Class cube
;;;
(define-class cube
  (instvars (identite 'A)
            (NOM #\A)
            (pile 1) ;repere horizontal, des lettres
            (niveau 1)) ;reperes vertical, des chiffres
  (options gettable-variables
           settable-variables
           inittable-variables))
(compile-class cube)
```

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

25

3 SE – Exemple «CUBES»

:si le cube de la pile de depart n'est pas libre [on ne peut pas le
deplacer car il y a des cubes au-dessus], il faut le rendre libre

```
(if (>
    (- (length
        (list-ref probleme
                  (send (eval cube-a-deplacer) get-pile)
                  )
        )
        1
    )
    (send (eval cube-a-deplacer) get-niveau)
)
;then
(set! possible
 (regles (list-ref
          (list-ref probleme
                    (send (eval cube-a-deplacer) get-pile)
                    );sélection de la pile
          (+ 1 (send (eval cube-a-deplacer) get-niveau))
          );sélection du cube au-dessus
        )
 pile-consigne niveau-consigne
) ;regles
);set!
);if
```

Implémentation en Lisp (2 de 3)

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008

26

3 SE – Exemple «CUBES»

```
(if possible
  ;then
  ; maintenant, on a un cube a deplacer
  ; il faut le poser quelque part
  (set! pile-destination
    (cherche-destination niveau-consigne
      pile-consigne
      (send (eval cube-a-deplacer)
        get-pile)
      )
    );set!
  );if
  ;principal
  (define (resou cube-a-deplacer pile-consigne niveau-consigne)
    (define possible)
    (eff-lgn (+ 1 l-consigne))
    (eff-lgn l-consigne)
    (window-set-cursor! 'console (+ 1 l-consigne) 3)
    (write " But:cube=") (write cube-a-deplacer)
    (write " pile=")(write pile-consigne)
    (write " niveau=")(write niveau-consigne)
    (window-set-cursor! 'console (+ 1 l-consigne) 1)
    (if (verif-consigne cube-a-deplacer pile-consigne niveau-consigne)
      ;then
      (toutes-les-regles cube-a-deplacer pile-consigne niveau-consigne)
      ;else
      (write " Consigne erronée")
    )
    (eff-lgn (- l-consigne 1))
  )
)
```

HESSO.HEIG-VD, J.-D. Dessimoz, 15.02.2008 27

AIC-Automatisation avancée, intelligence
artificielle et cognitive

FIN