

Représentations des connaissances

73

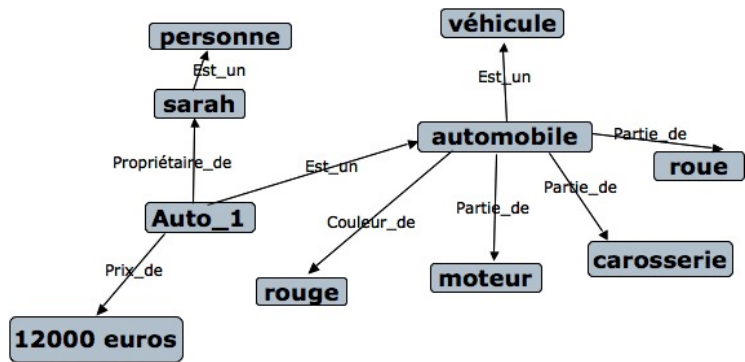
Représentation des connaissances

- Connaissances factuelles
- Connaissances opératoires

- Représentations simples
- Représentations structurées

74

Réseaux sémantiques



75

Frames (objets)

- (AutoDeSarah
(est_une (=voiture))
(couleur (=rouge))
(puissance (fiscale 9)
(unité CV)
(réelle 95))
- Tout est mis dans le frame
- Les facettes définissent la sémantique de l'attribut
- Mise en correspondance: la valeur de l'attribut peut être une valeur , un frame ou un ensemble de frames

76

Ontologies

- Description d'un domaine :
 - Concepts
 - Propriétés, attributs des concepts
 - Contraintes sur les propriétés et attributs
 - éléments individuels
 - Réseau sémantique. Concepts liés les uns aux autres par une taxonomie.
 - Permet une compréhension partagée commune du domaine.

77

Logique

78

Logique propositionnelle

- **Syntaxe**

- Propositions : a, b, c, \dots Ex: "la balle est rouge"
- Connecteurs :
 - Disjonction; OU: \vee
 - Conjonction; ET: \wedge
 - Négation; NON: \neg
 - Implication: \rightarrow Ex: "il pleut" \rightarrow "il y a des nuages"
 - Equivalence: \leftrightarrow

79

Formules

- Une proposition est une formule
- Si a et b sont des formules, alors $\neg a, a \wedge b, a \vee b, a \rightarrow b, a \leftrightarrow b$ sont aussi des formules

80

Sémantique

- Les formules sont interprétées dans $\{F, V\}$ ou $\{0,1\}$
- Tables de vérité:

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$\neg p \vee q$	$p \rightarrow q$
0	0	1	0	0	1	1
0	1	1	0	1	1	1
1	0	0	0	1	0	0
1	1	0	1	1	1	1

81

Formules

- Formule valide: toujours vraie (quelle que soit l'interprétation)
- Formule consistante: il existe une interprétation dans laquelle elle est vraie.
- Formule inconsistante: il n'existe aucune interprétation dans laquelle elle est vraie.

82

Règles

- Lois de Morgan :
 $\neg(a \vee b) \Leftrightarrow \neg a \wedge \neg b$
 $\neg(a \wedge b) \Leftrightarrow \neg a \vee \neg b$
- \wedge et \vee commutatives et associatives
- \vee et \wedge distributives et associatives l'une par rapport à l'autre

83

Règles et Equivalences

- $a \vee a \Leftrightarrow a \wedge a \Leftrightarrow a$ (idempotence)
- $a \vee \neg a$ (loi du tiers exclu)
- $\neg \neg a \Leftrightarrow a$ (double négation)
- $a \rightarrow b \Leftrightarrow \neg a \vee b$
- $\neg(a \rightarrow b) \Leftrightarrow (\neg b \rightarrow \neg a)$ (contraposition)
- $a \leftrightarrow b \Leftrightarrow (a \rightarrow b) \wedge (b \rightarrow a)$
- **$((a \rightarrow b) \wedge a) \rightarrow b$ (modus ponens) Règle d'inférence**
- **$((a \rightarrow b) \wedge \neg b) \rightarrow \neg a$ (modus tollens) Règle d'inférence**

84

Logique des Prédicats

- **Syntaxe**

- variables : x, y, z, \dots
- Fonctions; $f(), g(), \dots$
- Prédicats: p, q, \dots
- Propositions: a, b, \dots prédicats d'arité 0; $p(x), p(y)$: arité 1.
- Quantificateurs: \forall, \exists
- Connecteurs :
 - Disjonction; OU: \vee
 - Conjonction; ET: \wedge
 - Négation; NON: \neg
 - Implication: \rightarrow
 - Equivalence: \leftrightarrow

85

Atomes, formules

- Termes: Constantes, variables, fonctions de termes
- Atomes: prédicats de termes: $p(t_1, t_2, \dots)$
- Formules:
 - Un atome est une formule
 - Si F et G sont des formules, alors $\neg F, F \wedge G, F \vee G, F \rightarrow G$ sont des formules
 - Si F est une formule, $\forall x, F$ et $\exists x, F$ sont des formules

86

Clauses

- Un atome (ou sa négation) est un littéral (positif ou négatif)
- Une clause est une formule qui est une disjonction de littéraux.
- Une clause concrète ne comporte pas de variables
- Une clause de Horn est une clause comportant au plus un littéral positif

87

Principe de Résolution

- Règle d'inférence.
 - Clause1: $G = G_1 \vee G_2 \vee \dots \vee G_n$
 - Clause2: $H = \neg G_1 \vee H_2 \vee \dots \vee H_m$
 - G_1 et $\neg G_1$: littéraux complémentaires
 - Résolvante: $K = G_2 \vee \dots \vee G_n \vee H_2 \vee \dots \vee H_m$

88

Exemples

- P et $\neg P \vee Q$ se résolvent en Q (modus ponens)
- $\neg Q$ et $\neg P \vee Q$ se résolvent en $\neg P$ (modus tollens)
- $\neg G \vee H$ et $\neg H \vee K$ se résolvent en $\neg G \vee K$ (enchaînement)

89

Unification

- Pour les clauses non concrètes, l'unification est nécessaire
- Deux termes t_1 et t_2 sont unifiables s'il existe une substitution s des variables de t_1 et t_2 telle que $s(t_1) = s(t_2)$
- Exemple :
 - $SUR(X,Y)$ s'unifie avec $SUR(A,B)$ si $X = A$ et $Y=B$

90

Sémantique

- Interprétation I d'une formule G dans un domaine D
 - les constantes de G sont des éléments de D
 - les propositions de G prennent la valeur V ou F
 - les prédicats sont associés à des applications $D \rightarrow \{V, F\}$
 - les fonctions de G sont associées à des applications $D \rightarrow D$.

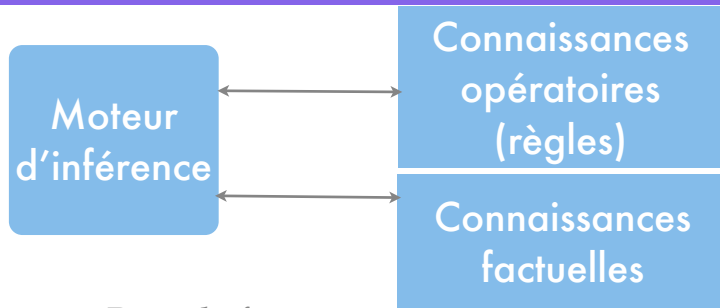
91

Indécidabilité

- Formule valide: vraie quel que soit le domaine d'interprétation (tautologie)
- Indécidabilité: Impossibilité de déterminer la validité d'une formule quelconque en un nombre fini d'opérations (Church)
- Semi-décidabilité: si la formule est valide, le nombre d'opérations est fini mais non borné.

92

Systemes à base de règles



- Base de faits
- base de règles
- moteur d'inférence

93

Principes

- Règles: Si (prémises) Alors (conclusions)
- Faits: propositions logiques
- Méthodes:
 - chaînage avant (à partir des prémisses). On cherche à établir des faits à partir de l'état initial.
 - chaînage arrière (à partir des conclusions). On cherche à résoudre un problème en remontant à l'état initial.

94

Intérêt

- Programmation déclarative
- connaissances éparses, parcellaires, granulaire
- savoir faire peu structuré (pas d'algorithme)
- évolution facile (nouvelles règles, nouveaux faits)
- Application principale: Systèmes experts

95

Contenu du cours

- <http://homepages.laas.fr/raja/Cours/P8-M2/>
- Qu'est-ce que l'IA?
- Agents intelligents; Architectures.
- Algorithmes de recherche dans les espaces d'états
- Représentations des connaissances
- Logique des propositions et des prédicats du 1er ordre
- Planification d'actions
- Raisonnement dans l'incertain, Processus décisionnels de Markov (MDP)
- Apprentissage et apprentissage par renforcement

96

Planification

97

Planification

- Décider des actions nécessaires pour atteindre un objectif: une problématique classique de l'IA
 - Objectifs abstraits
 - Actions de déplacement, de manipulation, de perception, ...
 - Raisonnement déductif et anticipation des conséquences des actions

98

Hypothèses de la planification classique

- Temps discret
- Actions déterministes
- Connaissance complète du monde
- Seul le robot agit sur le monde

99

Planification

- Entrées
 - Description de l'état initial du monde
 - Description de l'état but
 - Description des actions
 - Eventuellement: coût des actions
- Sortie:
 - Séquence d'actions qui mène de l'état initial à l'état final à coût minimal (nombre d'actions, coût des actions)

100

Questions

- Comment représenter les connaissances sur l'environnement?
- Comment représenter l'état du robot?
- Comment représenter les actions?
- Comment guider le raisonnement?

101

Cadre formel: Logique des prédicats

- L'état du monde et du robot est représenté par des propositions (vraies/ fausses).
Notion d'état.
- Les actions modifient l'état du monde et sa description
- La recherche des actions applicables repose sur l'unification logique
- L'enchaînement des actions est basé sur une recherche dans un **espace d'états**.

102

Représentation des actions

- Actions caractérisées par:
 - Qualification: ce qui est nécessaire à l'exécution l'action
 - Ramification: les conséquences de l'action
 - Cadre ou référentiel (Frame): ce qui est inchangé par l'action.

103

Exemple simple

- le "Monde des cubes"
 - Le robot fait des assemblages de cubes.
 - Le bras peut tenir un seul cube
 - Les cubes s'empilent parfaitement
 - La table est infinie
- Etat du monde = configuration des cubes
 - {on-table (G), on (B, G), clear (B), holding (R)}
- Actions: Opérateurs.
 - prendre (pick-up); poser (put-down); empiler (stack), dépiler (unstack)

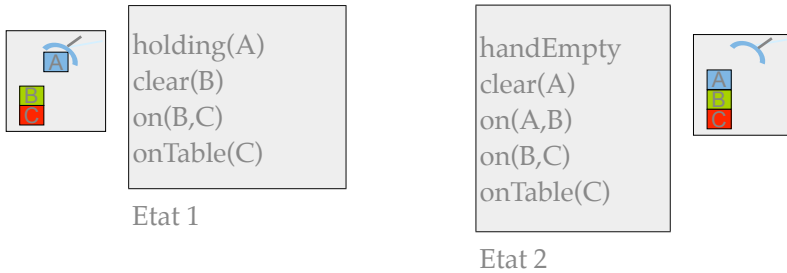


104

Représentation du monde

L'état du monde

est une liste de faits ou propositions logiques



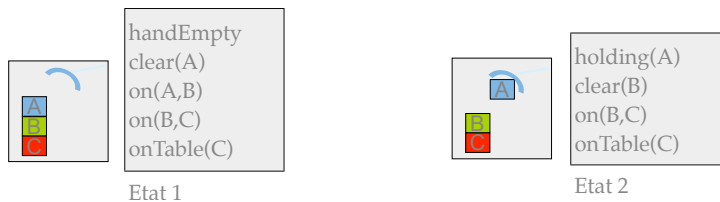
Hypothèse du Monde clos:
Les faits non mentionnés sont supposés faux

Représentation des buts

Les buts sont aussi des faits

Exemple: $\{ on(A,B) \}$ est un but

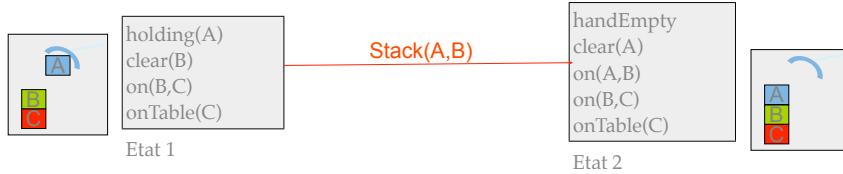
Un **Etat But** est tout état qui inclut les faits à réaliser



Etat 1 est un état but pour $\{ on(A,B) \}$.
Etat 2 n'est pas un état but.

Actions: Opérateurs

Représentations pour le planificateur STRIPS



- 1) PRE : liste de Préconditions (*qualification*)
- 2) ADD : liste d'Ajouts (*ramification*)
- 3) DEL : liste de Suppressions (*ramification*)

Exemple

Stack(A,B):

PRE: { holding(A), clear(B) }

ADD: { on(A,B), handEmpty, Clear(A) }

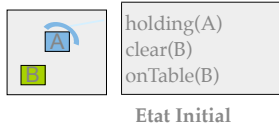
DEL: { holding(A), clear(B) }

Ce qui est inchangé par l'action n'est pas mentionné (cadre - frame)

Planification dans STRIPS

Problème:

Solution: (**Stack(A,B)**)



Stack(A,B):

PRE: { holding(A), clear(B) }

ADD: { on(A,B), handEmpty }

DEL: { holding(A), clear(B) }

Stack(B,A):

PRE: { holding(B), clear(A) }

ADD: { on(B,A), handEmpty }

DEL: { holding(B), clear(A) }

Il faut définir autant d'actions que d'objects

Schémas d'Action

Schéma générique plutôt que spécification de chaque action.

Introduction de variables

Stack(x,y):

PRE: { holding(x), clear(y) }
ADD: { on(x,y), handEmpty }
DEL: { holding(x), clear(y) }

Stack(B,A):

PRE: { holding(B), clear(A) }
ADD: { on(B,A), handEmpty }
DEL: { holding(B), clear(A) }

Stack(A,B):

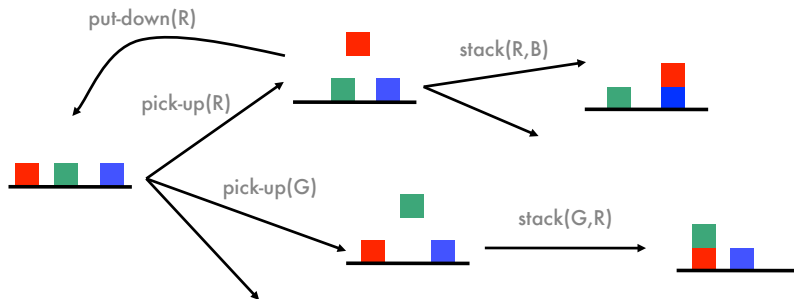
PRE: { holding(A), clear(B) }
ADD: { on(A,B), handEmpty }
DEL: { holding(A), clear(B) }

Les schémas doivent être instanciés pour être appliqués: variables x, y, ... transformées en constantes.

Problème d'*unification logique*.

Espace d'Etats

- Planifier := trouver les chemins les plus courts dans l'espace d'états représenté par un *graphe*



Algorithme STRIPS

Algorithme Strips; BUT, Etat initial E_0

Liste $L = []$, $PLAN = []$, $ETAT$

Initialisation: $L = [BUT]$, $ETAT = E_0$

Tant que L non vide

$g =$ tête de liste L ;

Si g est vrai dans $ETAT$: FIN

Sinon

Si g est une conjonction de buts cg

(maintenant en tête de L
il y a forcément un but
simple, g)

Choisir un ordre des sous-buts

Ajouter les sous-buts en tête de L

$g :=$ tête de liste L

Choisir un opérateur O dont les Ajouts s'unifient avec g

Ajouter les préconditions po de O en tête de L

Strips(po); appel récursif de Strips sur une précondition de O

$ETAT = ETAT(O)$; Appliquer O à l'état courant.

$PLAN = [PLAN, O]$; ajouter l'opérateur O au plan.

FIN

STRIPS

STRIPS(G)

S : Etat initial; G : état but

- Tantque G non vrai dans S faire:

DEBUT:

$g :=$ une composante de G non vraie dans S . (choix);

$O :=$ opérateur dont la liste d'ajouts contient un littéral qui s'unifie avec g . (choix);

$p :=$ précondition de l'instance appropriée de O ;

STRIPS(p);

$S :=$ résultat de l'application de l'instance de O à S .

FIN

Exemple



CLEAR(B) ON(C,A) **BUT:** ON(A,C) \wedge ON(C,B)
CLEAR(C) HANDEEMPTY
ONTABLE(A) ONTABLE(B)

113

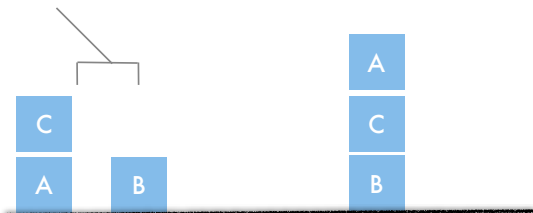
Exemple

ETAT DU MONDE

CLEAR(B)
CLEAR(C)
ON(C,A)
ONTABLE(A)
ONTABLE(B)
HANDEEMPTY

BUT:

ON(C,B) \wedge ON(A,C)



CLEAR(B) ON(C,A) **BUT:** ON(A,C) \wedge ON(C,B)
CLEAR(C) HANDEEMPTY
ONTABLE(A) ONTABLE(B)

Exemple

ETAT DU MONDE

CLEAR(B)
CLEAR(C)
ON(C,A)
ONTABLE(A)
ONTABLE(B)
HANDEEMPTY

BUT:

ON(C,B)
ON(A,C)
 $ON(C,B) \wedge ON(A,C)$

Décomposition du but

115

Exemple

ETAT DU MONDE

CLEAR(B)
CLEAR(C)
ON(C,A)
ONTABLE(A)
ONTABLE(B)
HANDEEMPTY

BUT:

ON(C,B)
ON(A,C)
 $ON(C,B) \wedge ON(A,C)$

Choix d'un opérateur

Stack(C,B)

PRE: { Holding(x), Clear(y)}
ADD: { On(x,y), HandEmpty}
DEL: { Holding(x), Clear(y)}

116

Exemple

ETAT DU MONDE

CLEAR(B)
CLEAR(C)
ON(C,A)
ONTABLE(A)
ONTABLE(B)
HANDEEMPTY

BUT:

HOLDING(C) \wedge CLEAR(B)
ON(C,B)
ON(A,C)
ON(C,B) \wedge ON(A,C)

Stack(C,B)

PRE: { Holding(x), Clear(y)}
ADD: { On(x,y), HandEmpty}
DEL: { Holding(x), Clear(y)}

117

Exemple

ETAT DU MONDE

CLEAR(B)
CLEAR(C)
ON(C,A)
ONTABLE(A)
ONTABLE(B)
HANDEEMPTY

BUT:

HOLDING(C)
CLEAR(B)
HOLDING(C) \wedge CLEAR(B)
ON(C,B)
ON(A,C)
ON(C,B) \wedge ON(A,C)

Stack(C,B)

PRE: { Holding(x), Clear(y)}
ADD: { On(x,y), HandEmpty}
DEL: { Holding(x), Clear(y)}

x=C, y=B

118

Exemple

BUT:

ETAT DU MONDE

CLEAR(B)	HOLDING(C)
CLEAR(C)	CLEAR(B)
ON(C,A)	HOLDING(C) \wedge CLEAR(B)
ONTABLE(A)	ON(C,B)
ONTABLE(B)	ON(A,C)
HANDEEMPTY	ON(C,B) \wedge ON(A,C)

Unstack(C,A)

PRE: {HandEmpty, Clear(x), On(x,y)}
ADD: {Holding(x), Clear(y)}
DEL: {HandEmpty, Clear(x), On(x,y)}

Stack(C,B)

PRE: { Holding(x), Clear(y)}
ADD: { On(x,y), HandEmpty}
DEL: { Holding(x), Clear(y)}

119

Exemple: Opérateurs

- **Pickup(x):**
PRE: {Ontable(x), Clear(x), HandEmpty}
ADD: {Holding(x)}
DEL: {Ontable(x), Clear(x), HandEmpty}
- **Putdown(x):**
PRE: {Holding(x)}
ADD: {Ontable(x), Clear(x), HandEmpty}
DEL: {Holding(x)}
- **Stack(x,y):**
PRE: { Holding(x), Clear(y)}
ADD: { On(x,y), HandEmpty}
DEL: { Holding(x), Clear(y)}
- **Unstack(x,y):**
PRE: {HandEmpty, Clear(x), On(x,y)}
ADD: {Holding(x), Clear(y)}
DEL: {HandEmpty, Clear(x), On(x,y)}

120

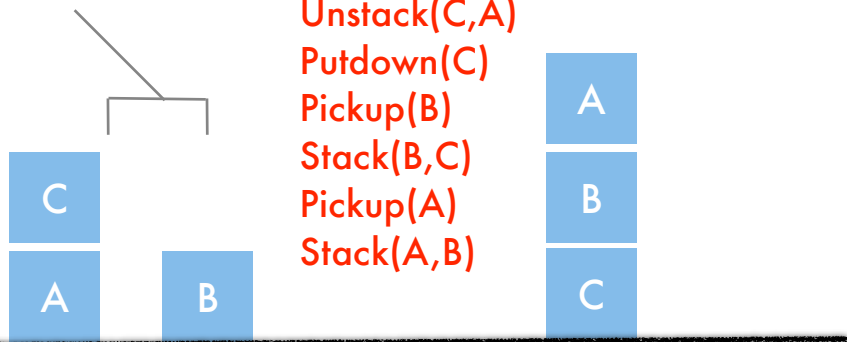
Exemple



CLEAR(B) ON(C,A) BUT: $ON(A,B) \wedge ON(B,C)$
CLEAR(C) HANDEEMPTY
ONTABLE(A) ONTABLE(B)

121

Exemple: Plan

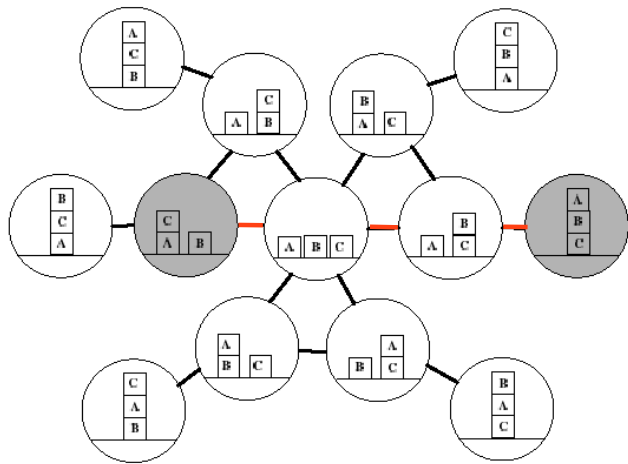


Unstack(C,A)
Putdown(C)
Pickup(B)
Stack(B,C)
Pickup(A)
Stack(A,B)

CLEAR(B) ON(C,A) BUT: $ON(A,B) \wedge ON(B,C)$
CLEAR(C) HANDEEMPTY
ONTABLE(A) ONTABLE(B)

122

Espace de recherche souvent fini



Limitations

- Limites de la logique: incertitudes, cas particuliers
- Représentation complexe des actions (liste exhaustive des conditions, des conséquences)
- Complexité du raisonnement logique: limitation à des cas relativement simples

Raisonnement Procédural

- Cadre pour des systèmes symboliques réactifs dans des environnements dynamiques
 - Robots Mobiles
 - Diagnostic de systèmes (p.ex Navette)

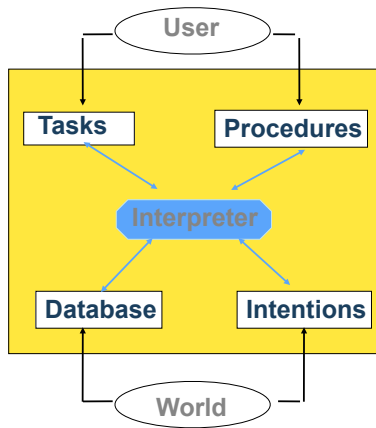
125

PRS: Procedural Reasoning System

- Connaissance: procédures pré-compilées
- Basé sur le modèle BDI (Belief, Desires, Intentions)
- Délibératif et réactif
 - Sélection de plans, exécution, perception
- Planification incrémentale et dynamique
- Orienté par le but et par les événements
- Architecture permettant d'interrompre l'exécution
- Méta-raisonnement

126

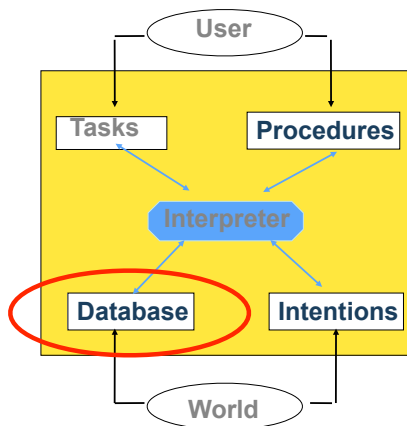
Architecture PRS



127

Base de données

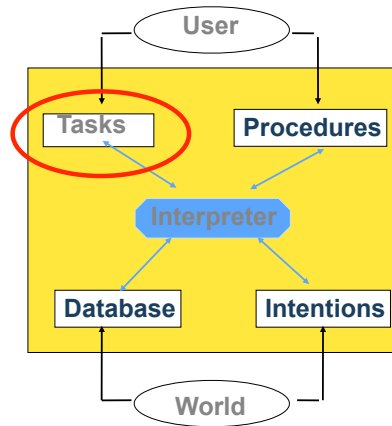
- Croyances ou faits sur le monde
- Meta-connaissances
- Ex: but G actif



128

Tâches

- Représentent le comportement désiré
- Conditions sur un intervalle temporel
- ex (move a b): ensemble d'actions permettant de se déplacer de a à b



129

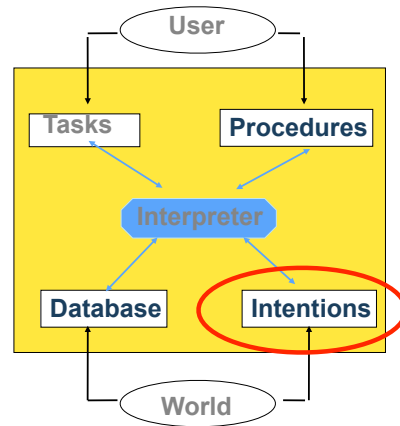
Formulation des tâches

- (! P) -- achieve P
- (? P) -- test P
- (# P) -- maintain P
- (^ C) -- wait until C
- (-> C) -- assert C
- (~> C) -- retract C

130

Intentions

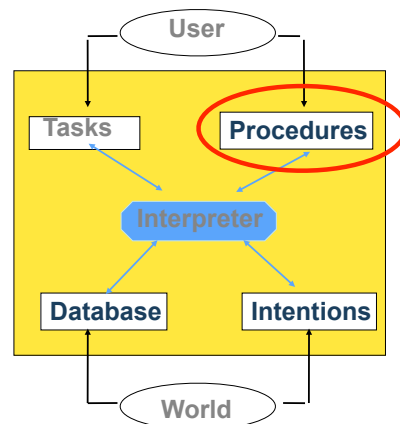
- Procédures actives (sélectionnées)
- Procédure en cours d'exécution



131

Procédures

- Procédures pré compilées
- Expriment les actions et tests pour réaliser des buts ou réagir à des faits



132

Procédures (formalisme Act)

Conditions

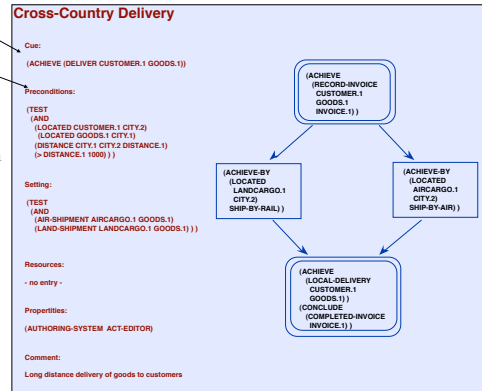
- Objectif (but ou condition)
- Critères d'applicabilité

Structure

- graphe orienté
- actions conditionnelles et parallèles partiellement ordonnées, boucles
- Exécution réussie d'un noeud si réalisation de ses buts
- actions primitives

Metapredicats

- Achieve - Achieve-By [pouv]
- Test - Conclude [effects]
- Wait-Until - Use-Resource
- Require-Until

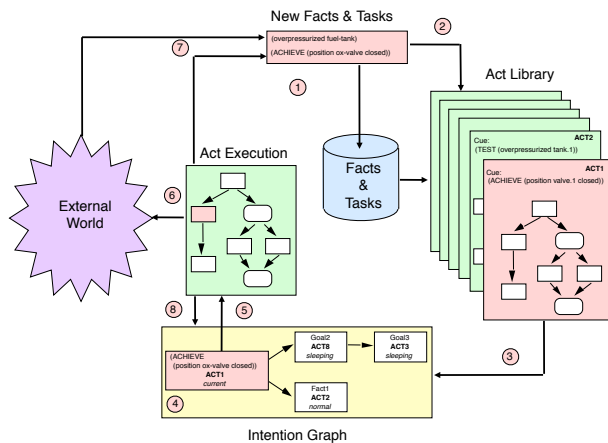


133

Interpréteur PRS

Cycle d'exécution

1. Nouveaux faits ou tâches
2. Des actions sont déclenchées par ces faits ou tâches
3. Ces actions deviennent des intentions
4. Une action est sélectionnée
5. Cette intention est activée
6. L'action est exécutée
7. Nouveaux faits ou tâches
8. Mise à jour des intentions

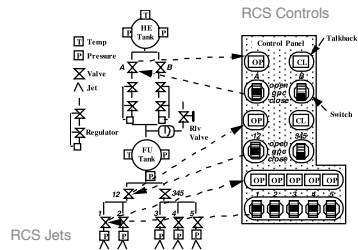


134

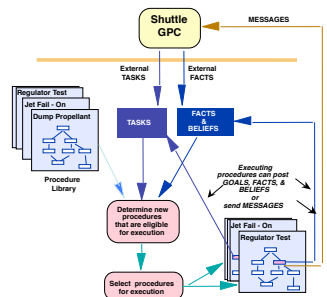
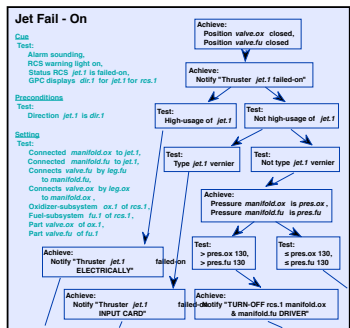
Méta-Raisonnement

- Peut inclure des procédures de méta-niveau, p. ex:
 - choix parmi plusieurs procédures applicables
 - évaluer une limite au cycle de raisonnement dans une durée donnée
 - Modalités d'exécution de buts conjoints (ET) ou disjoints (OU)

Ex: Traitement de dysfonctionnement pour la Navette spatiale



- Automatise la spécification et l'exécution des procédures du système de contrôle
- Réactions aux changements. Garantit une opération sûre pendant le diagnostic et les actions correctrices.





137

Conclusions

- Les systèmes de raisonnement procédural intégrés dans des architectures appropriées permettent de réaliser des systèmes opérationnels sur des problèmes réels
- Association avec des planificateurs comme ressources.

138