

Systemes Intelligents : Raisonnement et Reconnaissance

James L. Crowley

Deuxième Année ENSIMAG

Deuxième Semestre 2004/2005

Séance 1

4 février 2005

Intelligence, Connaissance et Raisonnement

Plan de la séance:

Intelligence, Connaissance et Raisonnement	2
Qu'est-ce que l'intelligence?.....	2
Qu'est-ce que la connaissance?.....	2
Qu'est-ce que la raisonnement?.....	2
Qu'est-ce qu'un symbole?.....	3
Sortes de connaissance.....	3
Introduction des Systemes Experts.....	4
Domaines d'Application des Systemes Experts.....	4
Techniques de Programmation des Systemes Experts.....	5
Le Systeme MYCIN.....	7
Représentation des Faits.....	9
Combinaison des hypothèses :.....	12
Contrôle de raisonnement.....	14

Notes des Cours sur Web :

<http://www-prima.imag.fr/Prima/Homepages/jlc/Courses/Courses.html>

Exercices de Programmation en CLIPS :

<http://www.ghg.net/clips/CLIPS.html>

Intelligence, Connaissance et Raisonnement

Qu'est-ce que l'intelligence?

INTELLIGENCE :

(Petit Robert) "La faculté de connaître et comprendre,
incluant la perception, l'apprentissage, l'intuition, le jugement et la
conception."

(Dictionnaire American Heritage) "La faculté de connaître et de raisonner."

(Newell et Simon) :

"Application de la connaissance à la résolution de problèmes"

Qu'est-ce que la connaissance?

Qu'est-ce que la connaissance ? la compétence

Ce qui permet la résolution de problèmes.

En d'autres termes :

la connaissance est définie par sa FONCTION,

c.-à-d. se caractérise par ce qu'elle fait, non pas par son contenu structurel.

Qu'est-ce que la raisonnement?

La génération de nouveaux connaissance par inférence.

Quelques techniques d'inférence :

Déduction : $(p \quad (p \quad q)) \quad (q)$

Abduction : $(q \quad (p \quad q)) \quad \text{peut-être}(p)$

Induction: $p(A) \quad q, p(B) \quad q, \dots \quad x (p(x) \quad q)$

Dans la technologie des système expert, les raisonnement est accompli
par la manipulation de symboles.

Qu'est-ce qu'un symbole?

Un Symbole est une relation ternaire entre :

- Un signe
- Une chose
- Un interprète

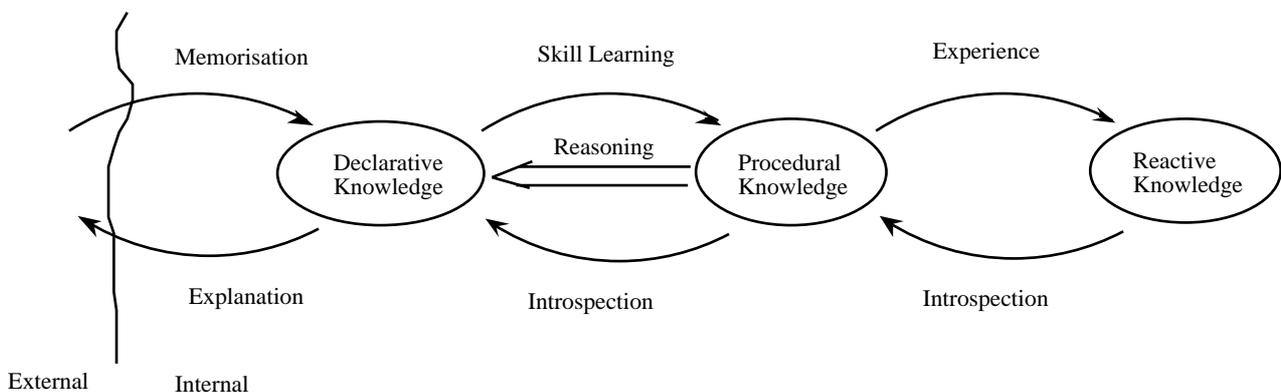
Élément central de l'I.A. est l'Hypothèse de Systèmes de Symboles.
d'A. Newell ("Physical Symbol Systems").

Sortes de connaissance

Déclarative : expression symbolique (abstraite) d'une compétence.
Utiliser pour communiquer et pour raisonner sur des connaissances.

Procédurale : expression "compilée" d'une compétence.
Utiliser pour optimiser le temps d'exécution.

Réactive : association "stimuli" - "Réponse"



Une autre distinction est la connaissance "superficielle" et connaissance "profonde"

Connaissance profonde : un modèle permettant le raisonnement par simulation

Superficielle : une expression symbolique des associations des faits permettant un raisonnement "abstrait".

La plupart des systèmes experts fonctionnent avec une connaissance "superficielle".

Introduction des Systèmes Experts

Domaines d'Application des Systèmes Experts

Les systèmes experts sont applicables pour les domaines :

Subjectifs
mal-formalisés, et
exigeant des jugements

Par nature, ces domaines ne se prêtent pas à la programmation "Algorithmique".
Donc, le processus d'extraction de connaissance d'un expert domine.

Il comprend

Définition du problème
Formalisation des faits et des relations du domaine.
Élaboration des règles
Raffinement.

C'est un processus itératif. Il est difficile de faire avec un cycle de vie classique.

Les plupart des systèmes experts fonctionnent avec une connaissance "superficielle".

Algorithmique : programme = algorithme plus donné

La programmation algorithmique s'applique aux domaines où il existe une théorie centrale qui prévoit le comportement des phénomènes.

Les exemples incluent la physique, la mécanique, et le calcul mathématique.

Système expert : programme = connaissance plus inférence (et contrôle)

Elle s'applique aux domaines où il n'existe pas de théorie centrale, mais seulement un ensemble de connaissances peu structurées.

Par exemple : la médecine, la loi et la conception de tout sort

Pour les Systèmes Experts, la compétence repose sur

- 1) un ensemble important des faits peu structurés,
- 2) et un peu de raisonnement par association.

Les faits : ce sont les expressions symboliques de description et relation

Le raisonnement : la génération de nouveau fait - l'inférence.

Un système expert est

- 1) une expression symbolique de la connaissance d'un domaine, combinée avec
- 2) un moteur d'inférence

Le principe est de séparer la connaissance du domaine du mécanisme du raisonnement.

Ceci pose le problème de “technique de représentation de la connaissance”

Techniques de Programmation des Systèmes Experts

Trois techniques sont utilisées pour représenter la connaissance :

Les règles :

Les schémas :

La logique.

Souvent les systèmes contiennent deux ou même trois formes.

Les règles :

si <CONDITIONS> alors <ACTIONS>

ou si <CONDITIONS> alors <CONCLUSIONS> do <ACTIONS>

Les règles encodent les associations et les lois causales.

Une partie du contrôle est codé dans la règle.

Ils sont une forme de réponses conditionnées semblable à celle des animaux.

Pour l' I. A., les règles sont une technique pour fournir une représentation des connaissances pour le raisonnement.

Tous les calculs sont exprimés par des règles, avec chaînage "avant".

(Règles = base des connaissances)

Basé sur un algorithme de "mise en correspondance" rapide permettant un nombre important de règles.

Exemples : OPS-5 -> OPS-83
-> ART -> CLIPS

Les schémas :

Représentation déclarative de connaissances sur la forme des "objets"

Les schémas sont les structures de données (les objets).

Ils codent les relations entre faits.

Ils permettent un raisonnement par "héritage", et une communication par message.

Calculs : héritages, démons et règles (chaînage avant et arrière)

Interrogation : calcul des prédicats

Exemples :

KRL -> KEE -> Nexpert (intégration des règles, prédicats et schémas)
-> Knowledge Craft (Intégration d'OPS-5, Prolog et "KRL").

MYCIN -> EMYCIN -> S1 -> ???

La logique :

Outil d'excellence pour l'analyse.

La logique est un moyen de communication "sans ambiguïté"

Le prolog mélange les expressions logiques (clause d'Horn)
avec une sémantique de programmation.

Tous les calculs sont exprimés par des règles en clause d'Horn
(Règles avec chaînage arrière).

Exemple : Prolog-1 => Prolog-2 => Prolog-3

Le Système MYCIN

Dans les années 1970, une équipe de Stanford University, sous la direction d'Ed Feigenbaum, a fait l'hypothèse que l'intelligence repose sur le stockage des grandes quantités de connaissance.

Ils ont cherché des techniques de "représentation de la connaissance".

= > "Heuristic Programming Project" (HPP)

Dans les années 1970 - 1973 ils ont fait un système d'analyse des données d'un analyseur spectrographique. => Dendral.

Vers la fin du projet ils ont trouvé utile de représenter la connaissance sous forme de règles.

En 1973 ils ont cherché un nouveau domaine pour refaire une étude. Ils ont retenu le domaine de "Thérapie Anti-Biotique".

Le résultat a été le système MYCIN : Conçu 1973 - 1978
E-MYCIN (Essential Mycin) - généralisation

Domaine : anti-microbien ou antibiotique

Il existe un grand choix de médicaments anti-microbiens ainsi qu'un grand nombre des microbes.

Chaque antibiotique agit différemment avec chaque microbe.

À cause de cette grande variété, seuls certains médecins spécialisés connaissent bien ce domaine.

Buts à atteindre :

Facile à utiliser

Fiable

Manipule un grand nombre de connaissances

Utilise des renseignements inexacts ou incomplets

Explique et justifie ses conseils

MYCIN est un "programme de consultation" :

Donner des avis aux médecins concernant les thérapies anti-microbiennes.

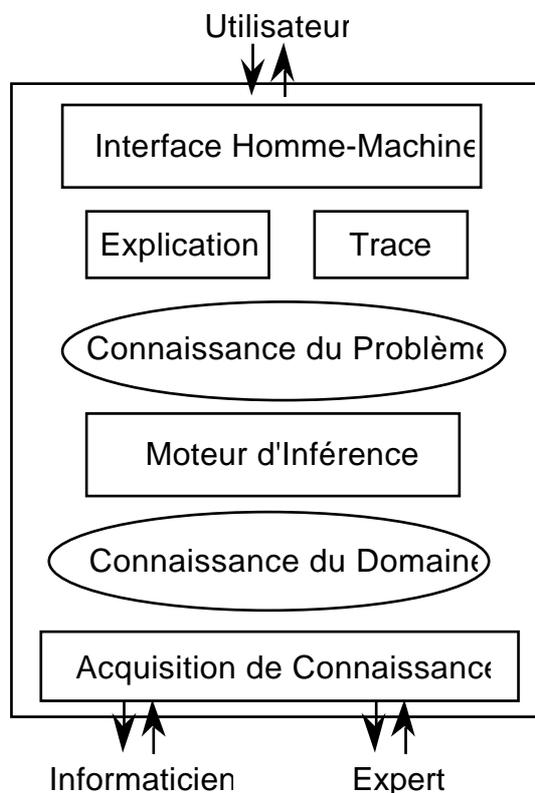
Le premier "vrai" système expert

- Programmation heuristique
- Connaissance experte du domaine
- Explication interactive
- Connaissance et jugements

Approximativement 500 règles avec les faits fortement typés.

Le système :

- 1) Demande des informations sur le cas.
- 2) Applique ses connaissances.
- 3) Donne son jugement et conseille.
- 4) Répond aux questions sur son raisonnement.



Représentation des Faits

Les faits : (contexte paramètre valeur CF)
en termes orientés objet : (objet attribut valeur CF)

En Mycin il y a 10 contextes.

Les CONTEXT contrôle le raisonnement en MYCIN.

Contextes :

Personne : le malade

OPERS : opérations antérieures

CurCuls: prélèvements effectués (et mis en culture)

CurDrugs : médicaments pris

CurOrgs : organismes présents dans une culture

OpDrugs : médicaments donnés pour une opération

PosTher : thérapies possibles

PriorCuls : prélèvements effectués dans le passé

PriorDrugs : médicaments pris dans le passé

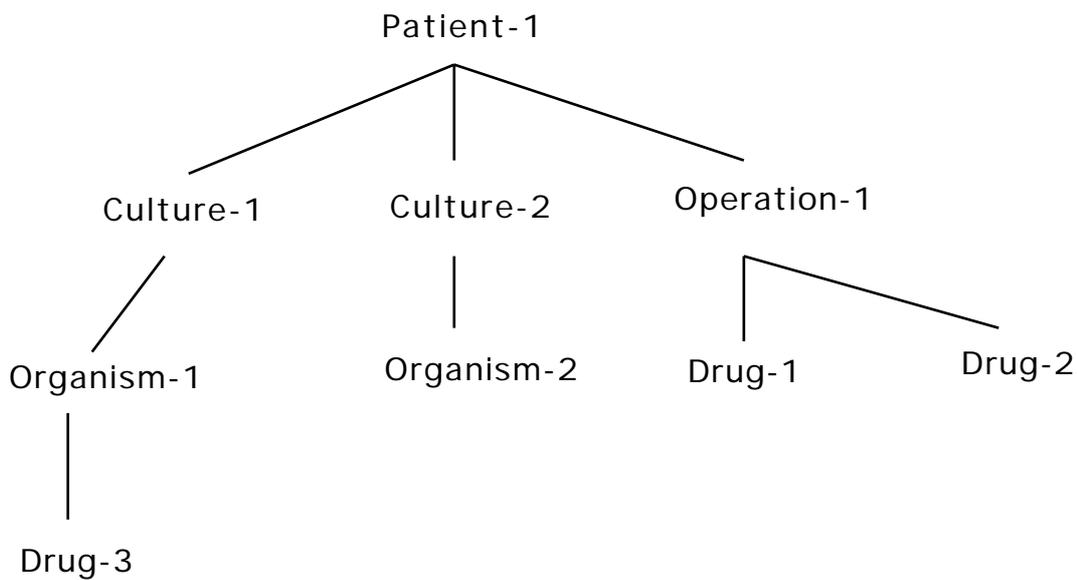
PriorOrgs : organismes trouvés dans le passé

Mycin répond à quatre questions :

- 1) Quelles infections importantes existent ?
- 2) Quel microbe est la cause de chaque infection ?
- 3) Quels médicaments sont efficaces ?
- 4) Quel est le meilleur médicament ?

Ceci correspond à l'élaboration de l'arbre des contextes.

L'arbre "dynamique" est composé des instances des contextes.



En général, MYCIN suit les hypothèses les plus sûres, de plus il existe des méta-règles pour "focaliser" la recherche.

Les paramètres

Les paramètres sont fortement typés.

Les paramètres sont étiquetés avec :

Expect : {Y/N, NUMB, ONE_OF, ANY_OF}

PROMPT : phrase pour demander une valeur

LABDATA : mesuré dans le laboratoire : Y/N : utilisé pour le contrôle
(si Y le système demande la valeur , sinon il infère pour la trouver)

LOOKAHEAD : liste des règles faisant référence à ce paramètre

TRANS : traduction en anglais ...

Trois catégories:

Single-Valued : (exclusive) beaucoup de valeurs possibles mais
une seule peut être vraie. Exemple : nom

Multi-Valued : pas exclusive

Exemple : médicaments, allergies

Binaire: Oui/Non

Croyance:

1) Tous les faits sont étiquetés par un facteur de croyance, CF [-1, 1]

2) Toutes les règles sont étiquetées par une "force" CF [-1, 1].

Les Règles :

Raisonnement : chaînage arrière, dirigé par un but.

Les règles en MYCIN sont sous forme d'abduction

$$A \quad B \quad \underline{CF_R} \rightarrow C$$

A tout instant, MYCIN a un "but" (C) qu'il cherche à prouver.

Pour prouver C il faut prouver A et B, pour prouver A ...

Légal :

$$A \quad B \quad C \rightarrow D$$

$$A \quad (B \quad C) \rightarrow D$$

$$(A \quad B \quad C) \& (D \quad E) \rightarrow F$$

Pas légal

$$A \quad B \quad C \rightarrow D$$

$$A \quad (B \quad (C \quad D)) \rightarrow E$$

MYCIN contient des "templates" qui permettent au système d'interpréter ses propres règles et même de former automatiquement de nouvelles règles.

Combinaison des hypothèses :

Conjonction : croyance de H_1 et H_2 : CF_1, CF_2

$$CF(H_1 \quad H_2) = \min\{ CF(H_1), CF(H_2) \}$$

Disjonction : croyance de H_1 ou H_2

$$CF(H_1 \quad H_2) = \max\{ CF(H_1), CF(H_2) \}$$

Les règles portent une force "CF_R"

$$A \quad B \quad \underline{CF_R} \rightarrow C \quad CF_C = CF_R * \min\{ CF_A, CF_B \}$$

$$A \quad (B \quad C) \quad \underline{CF_R} \rightarrow D \quad CF_D = CF_R * \min\{ CF_A, \max\{CF_B, CF_C\}\}$$

Si on arrive à la même hypothèse par deux voies indépendantes ?

Fonction générale pour combiner des hypothèses :

Combine(CF_1, CF_2) =

$$CF_1 + CF_2 (1 - CF_1) \quad \text{Si } CF_1 \geq 0 \text{ et Si } CF_2 \geq 0$$

$$\frac{CF_1 + CF_2}{1 - \min\{ |CF_1|, |CF_2| \}} \quad \text{Si } CF_1 \cdot CF_2 < 0$$

$$-Combine(-CF_1, -CF_2) \quad \text{Si } CF_1 \leq 0 \text{ et Si } CF_2 \leq 0$$

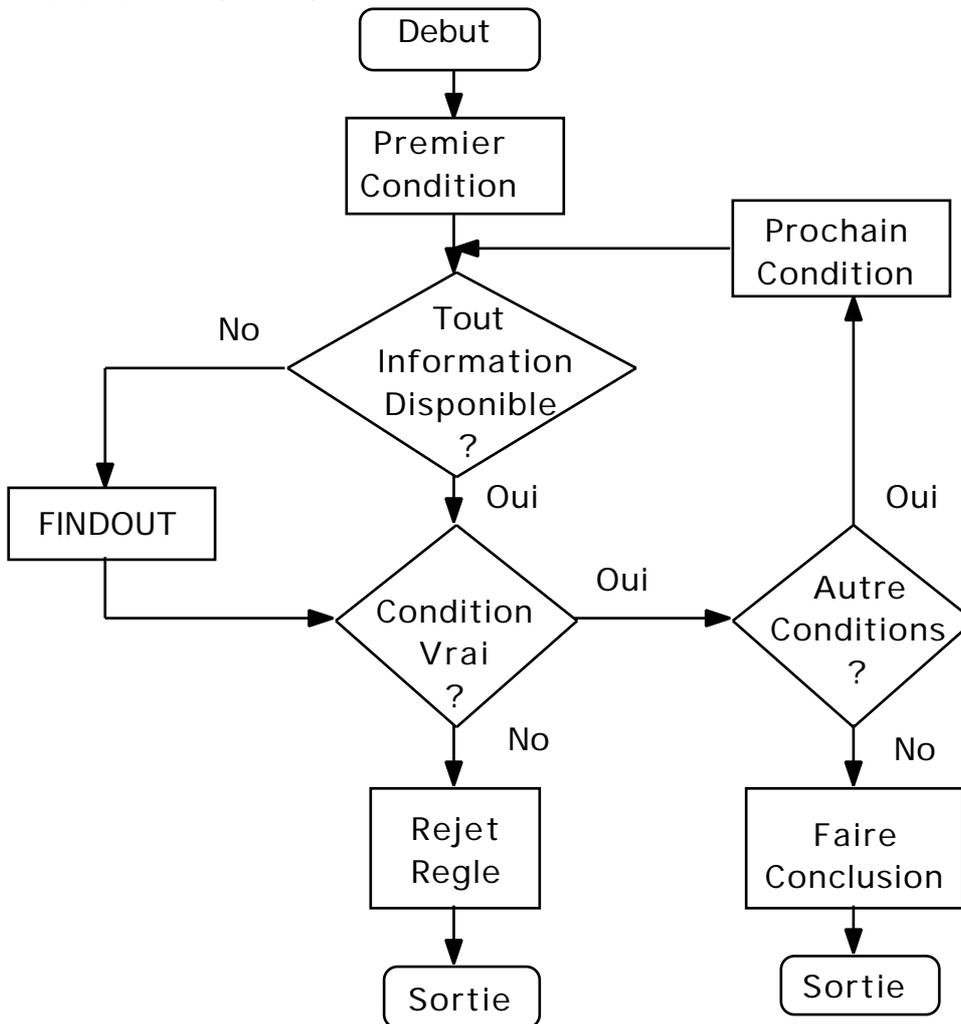
Alors :

$$(H_1, CF_1) \text{ et } (H_2, CF_2) \text{ et } (H_1 = H_2) \rightarrow (H_1, CF_1 = Combine(CF_1, CF_2))$$

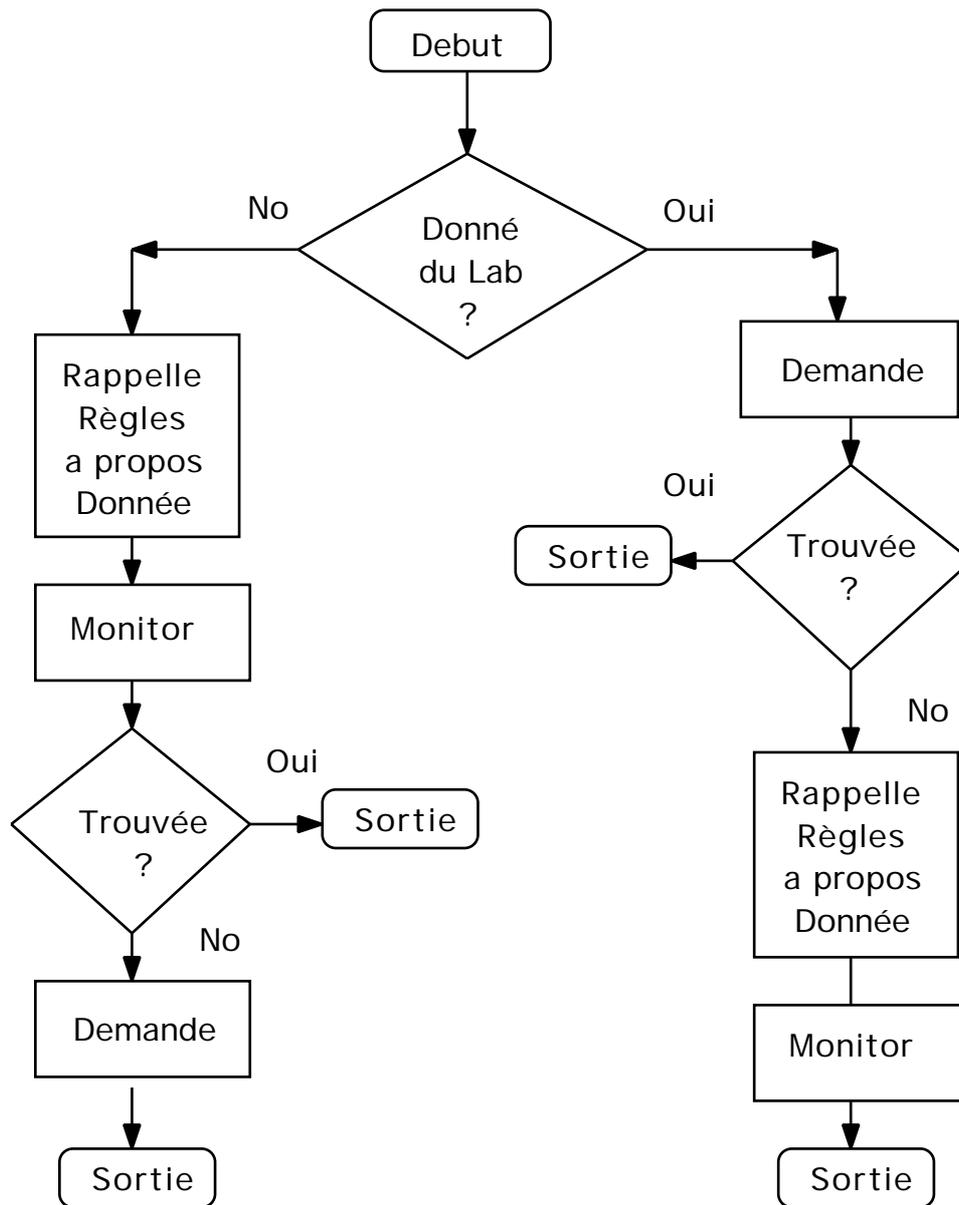
En MYCIN $CF(\neg H) = -CF(H)$

Contrôle de raisonnement

Procédure MONITOR :



Procédure FINDOUT:



Interactions :

A tout instant, l'utilisateur peut demander

WHY ? - Pourquoi ? - trace de la chaîne de raisonnement en cours.

How ? - Comment ? - trace la source d'un fait.

Le système est capable de "dé-compiler" les règles et les contextes en pseudo "anglais".