

Introduction à l'ingénierie des connaissances

Ingénierie des connaissances (Master 2 ISC)

IC : Qu'est-ce que c'est ?

Ingénierie évoque un ensemble de techniques et de méthodes, *appliquées* pour la résolution de problèmes complexes.

Connaissance : « ensemble des notions et des principes qu'une personne acquiert par l'étude, l'observation ou l'expérience et qu'elle peut intégrer à des habiletés ». (définition sur le web)

En informatique, l'**ingénierie des connaissances** évoquerait les techniques pour manipuler des connaissances sur ordinateur.

Autre définition

Dictionnaire terminologique québécois, définition de 1996 (donc un peu dépassée).

Ingénierie de la connaissance, équivalent anglais de **Knowledge engineering**, serait synonyme de **Génie cognitif** :

- ▶ Intégration des techniques d'intelligence artificielle et du génie logiciel en vue de concevoir et de construire des systèmes experts.
- ▶ Discipline étudiant l'extraction et la formalisation de connaissances provenant d'un expert humain en vue de leur intégration dans des systèmes experts.

On voit apparaître surtout la notion de **système expert**.

Systeme expert

Même dictionnaire.

Systeme expert (SE) : Systeme informatique permettant de résoudre les problèmes dans un domaine d'application déterminé à l'aide d'une base de connaissances établie à partir de l'expertise humaine.

Systeme à base de connaissances (SBC) : Systeme informatique fonctionnant avec une base de connaissances sur un sujet donné.

Les systèmes experts « classiques » serait un cas particuliers des systèmes à base de connaissances. Tout système informatique utilise de la connaissance, mais dans un système à base de connaissances, celle-ci est représentée de façon **explicite**.

Les systèmes experts

Objectif : reproduire les compétences d'un expert pour résoudre un problème. Il lui faut :

1. **connaître** des faits dans le domaine d'expertise.
2. **manipuler** ses connaissances dans l'objectif de la résolution d'un problème.
3. **choisir** la réponse la plus adaptée de façon heuristique, en acceptant des raisonnements approchés.

Un système expert effectue une tâche *knowledge-intensive*.

Différences avec un programme classique

Différence entre **donnée**, **information**, **connaissance**.

- ▶ la donnée transporte l'information ;
- ▶ l'information est extraite de la donnée ;
- ▶ la connaissance permet de relier les informations.

La manipulation des connaissances relève de **l'intelligence artificielle**. Permettrait de traiter des problèmes mal connus, flous ou évolutifs.

Tâches utilisant des connaissances

Classement issu de la *méthodologie CommonKADS* :

- ▶ Tâches d'analyse : classification, diagnostique, assessement, supervision, prediction.
- ▶ Tâches de synthèse : conception/configuration, modélisation, planification, ordonnancement, répartition.

Ce classement se veut relativement exhaustif des différentes tâches demandant de la connaissance.

Exemple de classification

Je dispose d'un champ sur lequel je veux cultiver des pommes de terre. Quelle variété cultiver ?

Divers facteurs peuvent rentrer en compte : le type de terrain, la présence de parasites/maladies, les éventuels traitements phytosanitaires envisagés, les débouchés prévus (alimentation humaine, animale, industrie)...

Le programme doit *étudier une situation existante et proposer une variété* (parmi les 3000 existantes, nombre fixé). On peut ranger ça dans les **classifications**.

Exemple (suite)

On pourrait ranger les caractéristiques de chaque variété dans une base de **données** :

Variété	Catégorie	Chair	Rendement	Résistance	...
Bintje	Consommation humaine	Farineuse	Excellent	Moyenne	...
...					

Cette table représente (en gros) la **connaissance du domaine**, mais ne permet pas de déterminer la meilleure variété : il manque les questions à poser, les critères prioritaires, etc...

Exemple (suite)

On peut imaginer des **règles** :

Si telle maladie est présente **alors** sélectionner les variétés peu sensibles à celle-ci.

Cette inférence mélange une connaissance du domaine (une variété doit résister aux maladies présentes pour être cultivée) **et** une connaissance de la tâche voulue (on cherche à sélectionner une variété (et non, par exemple, sélectionner un champ)).

Exemple (suite)

On peut généraliser cette règle à différents critères. Restent à déterminer un **ordre** sur ces règles, par exemple :

- ▶ Le premier critère est la catégorie souhaitée (consommation humaine ou animale ou industrie), puis ...
- ▶ On arrête s'il ne reste plus de critères *ou* moins de 2 variétés encore possibles.

Là encore, on fait intervenir connaissances du domaine et connaissances de la tâche. Séparer la représentation de ces connaissances peut être souhaitable et difficile.

Exemple (classique) de système expert : MYCIN

MYCIN : un des premiers systèmes experts (développé à Stanford à partir de 1972).

Objectif : diagnostique et traitement d'une infection bactérienne.

1. déterminer l'importance de l'infection ;
2. déterminer l'organisme responsable ;
3. identifier les médicaments possibles ;
4. choisir le meilleur traitement.

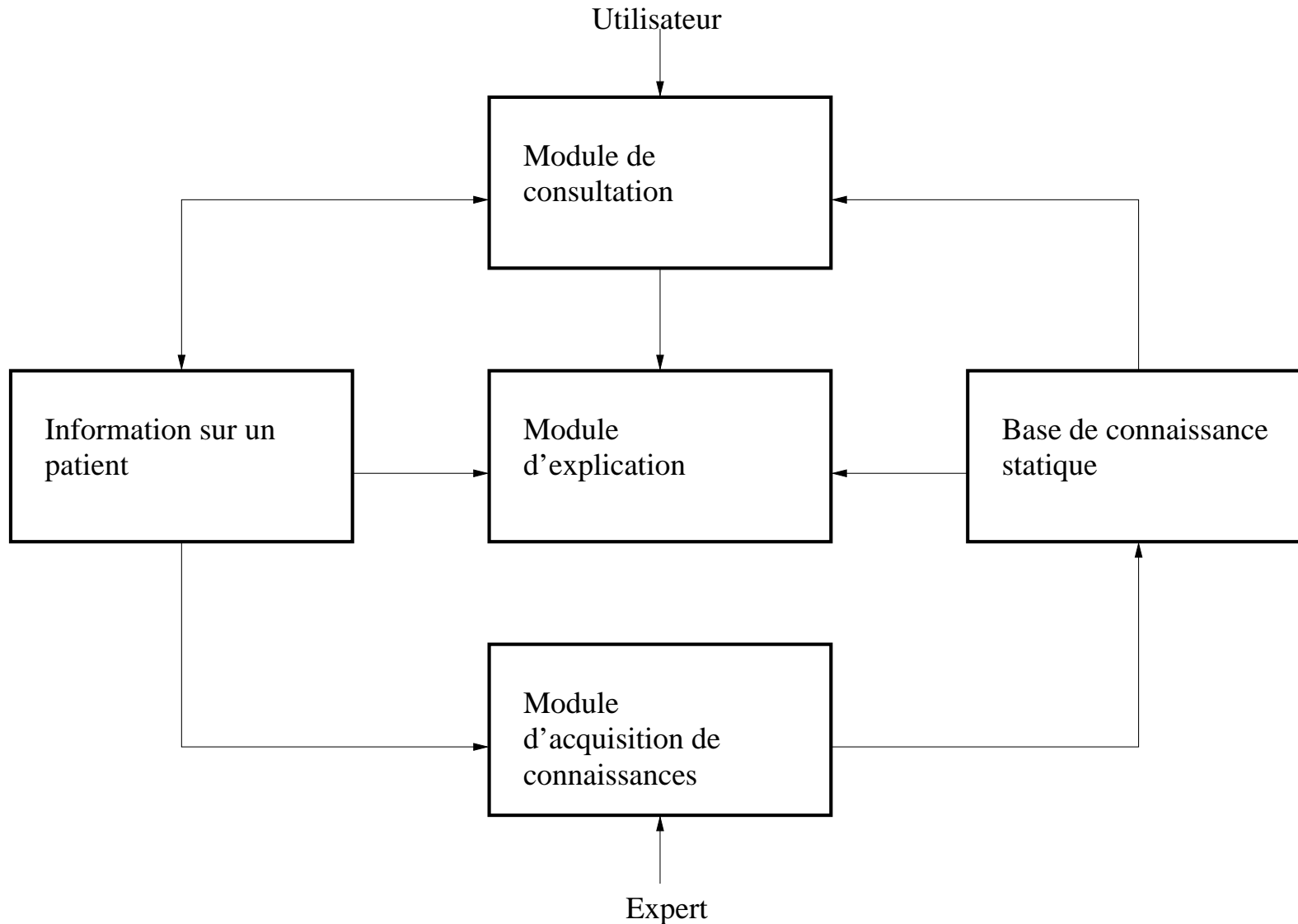
Le tout à partir de données incomplètes.

MYCIN (2)

Évaluation de MYCIN en 1979 : en « compétition » face à 8 médecins (plus le traitement appliqué) sur 10 cas réel, MYCIN arrive premier.

MYCIN était un programme de recherche et n'a jamais été réellement utilisé à l'hôpital : incomplet, difficile à évaluer, mauvaise interface... Mais il a montré qu'on peut approcher un domaine d'expertise.

Organisation de MYCIN



La connaissance dans MYCIN

La base de connaissances statique comporte des **règles** :

Si :

- ▶ la coloration de l'organisme est GRAM négatif
- ▶ sa morphologie est un bâtonnet
- ▶ il est aérobic

Alors :

- ▶ il est vraisemblable (0,8) que l'organisme est un Enterobacteriaceae

On dispose donc d'un ensemble de *règles* (ici, de déductions), approximatives.

Les inférences

Moteur d'inférence : déduit des faits à partir de faits initiaux et des règles.

Deux approches de base :

- ▶ partir de ce qu'on veut trouver et remonter vers les faits (chaînage arrière) ;
- ▶ partir des faits et aller vers ce qu'on veut trouver (chaînage avant).

⇒ étude des **logiques** (classiques ou non), et des modes de **raisonnements** (en intelligence artificielle).

La structure « classique » des systèmes experts utilisaient toujours un ensemble de *règles de production*.

Limites de MYCIN

1. Peu de flexibilité (adapté à un problème précis) ;
2. Connaissances difficiles à entrer :
 - ▶ beaucoup de règles ;
 - ▶ dépendantes du système d'inférence.
3. Manque d'explications sur le résultat.

MYCIN n'a jamais été réellement utilisé (mais ses successeurs oui).

Questions fondamentales des systèmes experts

1. Acquisition de la connaissance
2. Représentation de la connaissance
3. Contrôle du raisonnement
4. Explication de la connaissance

L'acquisition de la connaissance

Problème essentiel :

1. Communication expert - ingénieur - machine ;
2. définition du vocabulaire ;
3. choix de la représentation ;
4. processus (très) lent (plusieurs mois, voire années).

Approches :

1. Apprentissage automatique ?
2. Meilleure modélisation et interface (dialogue expert-machine) ?
3. Mise en place d'une méthodologie rigoureuse (CommonKADS) ?

Représentation des connaissances

Syntaxe et sémantique non ambiguë, loin de la représentation humaine ?

- ▶ Règles de production (MYCIN), programmes logiques ?
- ▶ Structures (arbres, réseaux, objets, cadres) ?

Comment représenter l'incertitude, la probabilité ?

Contrôle du raisonnement

Stratégie de raisonnements, heuristique (quelles questions, quels tests).

Au début, ordre du raisonnement (des conclusions aux axiomes ou l'inverse).

Exemple : recherche dans Prolog (stratégie simple).

Expressions de métarègles, décrivant la façon dont les règles sont utilisées (« sous la circonstantes A, préférer les règles de type R »).

Explication des connaissances

Approche la plus simple : donner la **trace** du raisonnement.

Lié à la représentation des connaissances, difficile dans certains cas (apprentissage automatique).

Évolutions de l'ingénierie des connaissances

« Echech » des systèmes experts. Explosion de la mémoire, arrivée d'internet.

1. Systèmes à base de connaissances, méthodologies
2. Modélisation des règles métiers.
3. Ontologies
4. Ingénierie documentaire

L'« échec » des systèmes experts

Sortir d'un domaine d'application spécifique, changement des besoins.

Limites de la représentation des connaissances, mauvaises tentatives de reproduction.

Modèle

Modélisation du système « physique » :

- ▶ fonctionnel : la description se fait pour un cadre et un objectif particulier ;
- ▶ qualitatif : éviter la quantification des mesures.

Modélisation du raisonnement :

- ▶ méthode de parcours du modèle ;
- ▶ stratégies.

Buts d'un modèle

Un modèle est une **abstraction** du monde pour diminuer sa complexité, en se focalisant sur certains buts.

- ▶ il conceptualise les entités du domaine et les interactions ;
- ▶ il offre un intermédiaire entre les connaissances et le code ;
- ▶ il devrait permettre de manipuler les objets, d'interpréter les résultats.

Les manipulations ne sont pas forcément liées à l'interprétation.

Modèles de connaissances

Deux types de connaissances :

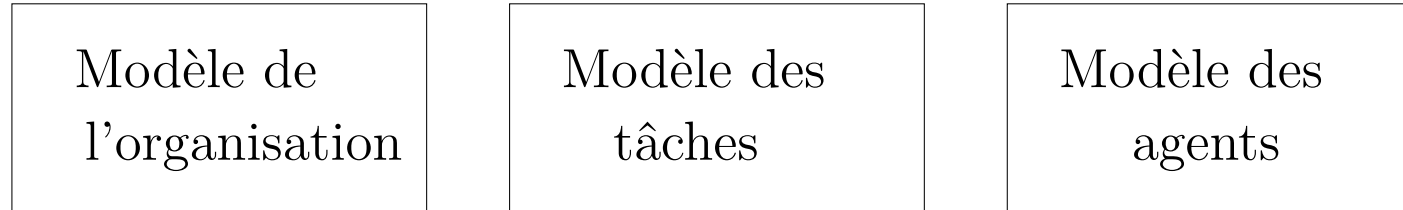
1. connaissances du raisonnement ;
2. connaissances du domaine dans lequel le raisonnement sera fait (entités du domaine, relations entre ces entités).

La mise en place d'un *shéma du domaine* (ou « modèle du domaine ») accélère l'acquisition et la manipulation des connaissances.

⇒ bien structurer les connaissances du domaine, indépendamment de l'implémentation.

CommonKADS (knowledge analysis and design system)

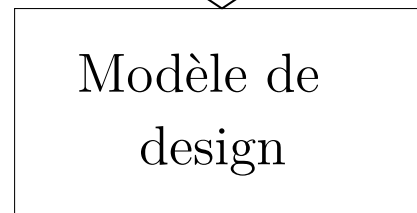
contexte



concepts



produit



Ontologies

Objectifs :

1. réutiliser des connaissances de domaines génériques ;
2. partages, communications entre SBCs.

Mises en place de standards, de logiciels génériques, de langages d'échange.

Approche du cours

1. Divers aspects liés aux SBC (représentation des connaissances, raisonnements), utilisation d'un langage adapté (drools) ;
2. Quelques bases d'apprentissage.
3. Méthodologie de modélisations (CommonKADS).