

Chapitre 2 : Systèmes experts

La notion de systèmes experts est apparue dans les années 70 avec l'apparition du système expert célèbre MYCIN dont le but était d'aider les médecins à effectuer le diagnostic des maladies infectieuses du sang. La version de base contenait 200 règles ensuite 300 règles concernant les méningites ont été ajoutées.

Aujourd'hui, les systèmes experts constituent une technologie bien définie faisant partie des systèmes à base de connaissances. Les systèmes experts ont comme finalité la modélisation de la connaissance et de raisonnement d'un expert (ou d'un ensemble d'experts) dans un domaine donné. Pour cela, trois acteurs principaux doivent contribuer à l'élaboration d'un système expert à savoir :

- 1- L'utilisateur final,
- 2- L'expert du domaine
- 3- Et l'ingénieur de connaissances.

L'interaction entre ces trois acteurs amènera à l'élaboration d'une première version de systèmes experts contenant

- 1- Une base de connaissances,
- 2- Et un moteur d'inférence effectuant une forme définie de raisonnement.

Dans la suite de ce chapitre, nous allons présenter les points suivants :

L'architecture générale d'un système expert ainsi que le processus d'acquisition de connaissances. Dans un premier temps, nous allons voir les systèmes experts à base de règles de production.

Les algorithmes d'inférence appelés chaînage avant et chaînage arrière. Les différentes stratégies de résolution que peut utiliser un moteur d'inférence sont aussi présentées.

1. Technologie des systèmes experts

Les experts humains sont capables d'effectuer un niveau élevé de raisonnement à cause de leur grande expérience et connaissance sur leurs domaines d'expertise.

Un système expert utilise la connaissance correspondante à un domaine spécifique afin de fournir une performance comparable à l'expert humain. En général, les concepteurs de systèmes experts effectuent l'acquisition de connaissance grâce à un ou plusieurs interviews avec l'expert ou les experts du domaine. Les humains qui enrichissent le système avec leurs connaissances ne fournissent pas seulement leur connaissance théorique ou académique mais aussi des heuristiques qu'ils ont acquises grâce à l'utilisation de leurs connaissances. Contrairement à la modélisation cognitive, les systèmes experts n'ont pas comme finalité de s'inspirer des théories du fonctionnement du cerveau humain mais ce sont des programmes qui utilisent des stratégies heuristiques pour la résolution des problèmes spécifiques. Le raisonnement effectué par un système expert doit être objet à l'inspection, et ceci en fournissant d'information sur l'état de la résolution du problème et des explications sur les choix et les décisions du système. D'un autre côté, la solution fournie par le système doit être évaluée par un expert humain et ceci dans le but de modifier l'information contenue dans la base de connaissances.

2. Structure et fonctionnement d'un système expert

Architecture d'un système expert

Les composantes principales d'un système expert sont :

- La base de connaissances,
- Le moteur d'inférence,
- Le programme utilisateur.

La base de connaissances : La base de connaissances sert à l'organisation et la centralisation de tous les éléments de connaissance du système expert. Il s'agit de la base du système et sa qualité influera grandement sur les capacités du système expert.

Une base de connaissances contient toutes les connaissances spécifiques sur un domaine et constitue un élément essentiel pour tout système expert. En effet, pour prendre une bonne décision et faire de bons raisonnements, il faut se baser sur des connaissances de bonne qualité et bien organisées. Il va donc de pair que les connaissances d'un système expert soient organisées dans une base de connaissances. Des connaissances bien organisées peuvent être définies comme étant facilement utilisables, retrouvables et modifiables dans la base de connaissances. De plus, mieux organisées sont les connaissances, moins une modification d'une d'entre elles a des répercussions sur les autres.

• Le moteur d'inférence : (du verbe « inférer » qui signifie « déduire »), cette partie du système expert donne la capacité de raisonnement de ce dernier. Le moteur d'inférence utilise les connaissances contenues dans la base de connaissances afin d'appliquer le raisonnement propre au système expert. Il est le cerveau du système.

• **L'interface utilisateur** : Ce programme permet à l'utilisateur d'interagir avec le système expert.

La figure 1 illustre le système expert :

- ✓ A partir d'entretiens avec des experts, on extrait une base de données avec des connaissances spécifiques à ce domaine. Généralement, ce sont des règles du type:

Si p et q alors r

Et une **base de faits** qui contient les données spécifiques liées à l'application traitée. Elle peut contenir aussi les solutions intermédiaires ou les conclusions partielles trouvées lors de l'inférence.

- ✓ Le moteur d'inférences établit une conclusion grâce à l'information fourni par l'utilisateur et aux bases des connaissances.
- ✓ La base de connaissances ainsi que le système d'inférence sont fondés sur la logique.

L'interaction avec l'utilisateur prend typiquement la forme d'un dialogue, où le système expert pose des questions pour aider à trouver la bonne solution.

3. Domaines d'applications d'un SE :

Les systèmes experts se retrouvent utilisés dans divers domaines d'applications tels que :

La médecine : diagnostic médical, classification radiographique.

L'informatique : composante de systèmes tutoriels intelligents, maintenance électronique.

Le marketing : sélection d'hôtel.

L'industrie : surveillance, diagnostics de pannes, gestion d'incidents

La géologie : aide à la prospection géologique et minière Etc

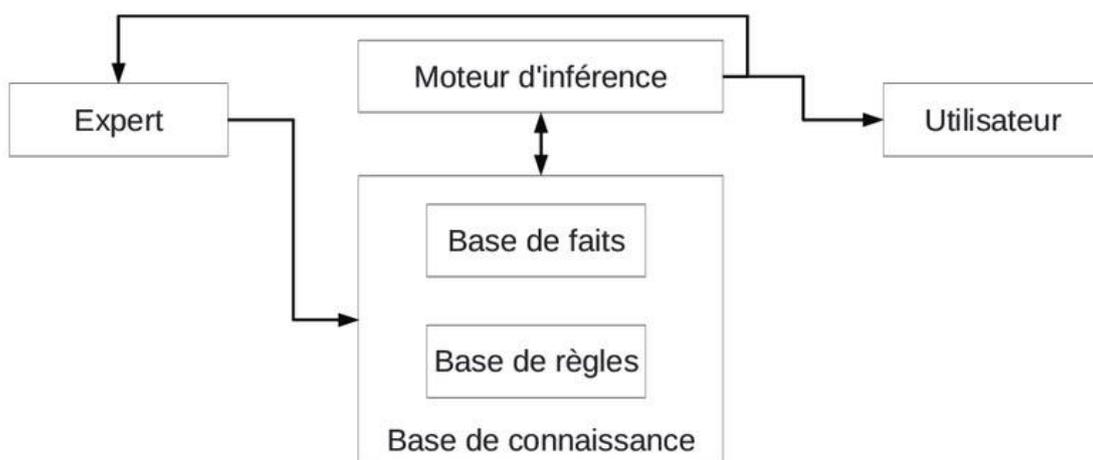


Figure 1. Architecture d'un système expert.

Chapitre 3 : Fonctionnement des systèmes expert

1. Formalisme de représentation

Pour la représentation des connaissances on utilise des règles :

Une structure **IF –THEN** qui relie de l'information ou des faits de la partie IF avec l'action dans la partie **THEN**.

- **Ex: IF** «lumière est verte» **THEN** « action est go»
- **Ex: IF** «lumière est rouge» **THEN** « action est stop»
- **IF** «condition est vraie» **THEN** «faire quelque action».
- Un langage de programmation est une façon de représenter des
- connaissances.

Connaissances procédurales:

- «comment».
- **Connaissances sur comment accomplir une tâche.**

Connaissances déclaratives:

- «qu'est-ce que c'est»
- **La capacité de formuler ou décrire quelque chose**

2. Les règles de production

Une règle est constituée de deux parties:

1. Antécédent: la partie IF
2. Conséquence: la partie THEN

IF «antécédent» THEN «conséquence»

- Une règle est déclenchée lorsque l'antécédent est vrai et que la partie conséquence est exécutée

La base de connaissances

- La base de faits
- La base de règles
- Les métarègles et la métaconnaissance
- La représentation des connaissances incertaines

Base de faits

- Mémoire de travail

– Au début de la session : contient ce que l'on sait du cas examiné avant toute intervention du moteur d'inférences.

– Puis : complétée par les faits déduits par le moteur ou demandés à l'utilisateur.

– Exemple : dans le domaine médical,

– Base de faits = liste de symptômes en début de session et un diagnostic lorsque celle-ci se terminera.

- Le type d'un fait

– les faits élémentaires sont :

- booléens : vrai, faux
- symboliques : c'est-à-dire appartenant à un domaine fini de symboles

- réels : pour représenter les faits continus.

– Par exemple, actif est un fait booléen, profession est un fait symbolique et rémunération est un fait réel.

- Les formules ou conditions

– Dans un système expert d'ordre 0, on pourra par exemple écrire des formules de la forme : actif ou \neg actif

– Dans un système d'ordre 0+, on pourra trouver les formules : actif et (profession \neq medecin ou rémunération \leq 20000)

– Dans un système d'ordre 1, on pourra trouver : $\exists X$ maladie(X) et $X \neq$ grippe et symptome(X) = forteFievre

- **Métafaits et métavaleurs**

–Savoir si une valeur a été attribuée à un fait

- Demandée à l'utilisateur

- S'il ne peut pas répondre, il faut que le système puisse le savoir

– Il n'est pas envisageable qu'un médecin demande à son patient <<quelle maladie avez-vous ? >>

- demandable (diagnostic) est un métafait booléen (et une métacondition).

La base de règles

- Elle rassemble la connaissance et le savoir-faire de l'expert. Elle n'évolue donc pas au cours d'une session de travail.

- Une règle est de la forme :

Si <conjonction de conditions> **alors** <conclusion>

où les conclusions sont de la forme : <Fait> = <valeur>.

- Une base de règles est un ensemble de règles et sa signification logique est la conjonction de la signification logique de chacune des règles.

– si A ou B alors C ou

– si A alors B et C

Il n'en est par contre pas de même de

– si A alors B ou C

Les métarègles et la métaconnaissance

- **Métarègles** : règles sur la manière d'utiliser les règles

- On trouve par exemple dans MYCIN les métarègles suivantes :

– si le patient est un hôte à risque et s'il existe des règles mentionnant des pseudo-monias dans une prémisses et s'il existe des règles mentionnant des klebsiellas dans une prémisses alors il est probable qu'il faille utiliser les premières avant les secondes

La représentation des connaissances incertaines

- On peut reconnaître globalement un objet sans être capable d'identifier à 100% chacun de ses détails.
 - théorie des probabilités : degré de vraisemblance d'un fait.
 - De nombreux générateurs de systèmes experts offrent la possibilité aux utilisateurs de nuancer leur certitude concernant un fait en leur associant un degré de vraisemblance.
 - il n'est pas raisonnable d'attendre d'un être humain, expert ou non, qu'il puisse définir avec précision de tels degrés de vraisemblance.
- logique floue, logique modale,...

3. Moteur d'inférences :

Chainage avant

- Saisie des faits initiaux
- Début
 - Phase de filtrage => Détermination des règles applicables
 - Tant que ensemble de règles applicables n'est pas vide ET que le problème n'est pas résolu
- Faire
 - Phase de choix => Résolution des conflits
 - Appliquer la règle choisie (exécution)
 - Modifier (éventuellement) l'ensemble des règles applicables
- Fin faire
- Fin

ALGORITHME DU CHAINAGE AVANT

ENTREE : BF, BR, F

- DEBUT
- TANT QUE F n'est pas dans BF ET qu'il existe dans BR une règle applicable
- FAIRE
 - choisir une règle applicable R (étape de résolution de conflits, utilisation d'heuristiques, de métarègles)
 - $BR = BR - R$ (désactivation de R)
 - $BF = BF \cup \text{concl}(R)$ (déclenchement de la règle R, sa conclusion est rajoutée à la base de faits)
- FIN DU TANT QUE

SI F appartient à BF ALORS

- F est établi
- SINON
 - F n'est pas établi

- FIN

Exemple : les règles

• REGLE r1 SI animal vole ET animal pond des œufs ALORS animal est un oiseau

• REGLE r2 SI animal a des plumes ALORS animal est un oiseau • REGLE r3
SI animal est un oiseau ET animal a un long cou ET
animal a de longues pattes
ALORS animal est une autruche

Exemple : les faits

- F1 : animal a des plumes
- F2 : animal a un long cou
- F3 : animal a de longues pattes

Chaînage arrière

- Le principe est le suivant :
 - Le moteur recherche les règles qui concluent sur le but à vérifier, et s'assurent que ces règles sont "déclenchables".
 - La règle est déclenchable si ses prémisses sont vérifiées.
 - Si parmi les règles sélectionnées, une règle est déclenchable, alors le but est vérifié.
 - Si ce n'est pas le cas, alors les prémisses à vérifier deviennent de nouveaux buts, appelés sous-buts, et le processus est réitéré.

- Les principales conditions d'arrêt :
 - L'ensemble des sous-buts est vide (succès) = tous les sous-buts ont été vérifiés et le problème est résolu
 - Impasse ou échec : Soit un des sous - buts n'est pas vérifiable avec la règle courante et il faut choisir une nouvelle règle pour le vérifier, et si cela n'est pas possible, alors il y a échec.

Moteur d'inférences : Chaînage arrière

- Phase de filtrage
- Si l'ensemble des règles sélectionnées est vide Alors questionner l'utilisateur
- Sinon
 - Tant que le but n'est pas résolu ET qu'il reste des règles sélectionnées
Faire
 - Phase de choix
 - Ajouter les sous-buts (partie gauche de la règle choisie)
 - Si un sous-but n'est pas résolu Alors mettre le sous-but en but à résoudre

– Fin faire

Le chaînage mixte

- Le chaînage mixte alterne les chaînages arrière et avant pour optimiser les conclusions.
- Principe :
 - Tant que j’obtiens de nouvelles conclusions:
 - J’applique un chaînage arrière
 - J’applique un chaînage avant

Exemple

Base des faits initiaux a, c, d, e, g, h, k

Base de règles

1. si k et w et m alors i »
2. si i et w et j alors q »
3. si f et h alors b »
4. si a et b alors q »
5. si c et h alors r »
6. si r et j et m alors s »
7. si g alors f »
8. si w et n et o et p alors q

On demande d’établir le fait q par un chaînage avant

Limites

- Les connaissances d’un expert peuvent être difficiles à extraire (implicites ou non verbales)
- Concevoir une base de faits et une base de règles fonctionnelles peut être compliqué.
- Il est difficile de concevoir une interface homme / machine efficace et sexy pour système expert.

Conclusions :

Les systèmes experts sont partout

- terme "système expert" un peu suspect de nos jours
 - dans le monde commercial, on parle souvent de "bases de données" plutôt que de "systèmes experts"
 - Par exemple, la réservation d'un vol sur une ligne aérienne implique nécessairement

- (1) une base de connaissance (les vols assurés par la compagnie),
 - (2) un ensemble de faits à considérer (votre demande de destination avec une gamme de départs et arrivées possibles),
 - (3) une logique de règles ayant trait aux trajets disponibles, aux dates de départ et aux réservations déjà faites.
- Un tel système correspond donc à la définition générale d'un système expert

Haton & Haton (1989) distinguent différents types de systèmes experts

- Systèmes d'interprétation de données. – P.ex., systèmes de diagnostic en médecine ("de quelle maladie s'agit-il?"), système d'interprétation géologique ("les mesures sismologiques permettent-elles de croire à l'existence de dépôts minéraux importants?"), systèmes d'évaluation psychologique ("s'agit-il d'un cas suicidaire?"), etc.
- Systèmes de prédiction. – P.ex., systèmes de prédiction météorologique ("Il pleut aujourd'hui en France. Va-t-il pleuvoir en Suisse demain?"), prédictions géopolitiques ("Les conflits de guerre sont particulièrement fréquents en situation de crise économique. Quelles combinaisons précises de facteurs économiques, sociologiques et politiques prédisent un déclenchement d'hostilités?"), etc.
- Systèmes de planification. – P.ex., système de réservation de vols aériens, planification des altitudes de vol selon les vents connus et les corridors disponibles, planification des actions d'assemblage d'un robot industriel, planification des interventions requises pour la construction d'un bâtiment, etc.
- Systèmes de conception. – P.ex., Développement et simplification de circuits intégrés, aménagement d'une cuisine optimale dans un espace donné, clonage de gènes, création d'un nouveau composé chimique, etc.