**Chapitre 3 : 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. une base de faits
3. 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 e 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. Dans un deuxième temps, le raisonnement incertain est présenté. Les approches probabilistes classiques comme celle de Bayes est étudiée dans le cadre d’un moteur d’inférence incertain. Une approche plus évoluée est l’approche de Dempster-shafer car elle a beaucoup moins de contraintes que l’approche probabiliste. Ensuite, la théorie des facteurs de certitude et celle des ensembles flous qui s’intègrent plus naturellement avec les règles de productions sont présentées.

Les autres types de représentation de connaissances sont présentés comme les réseaux sémantiques, les graphes conceptuels et les frames.

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.

* 1. **Structure et fonctionnement d’un système expert**

L’architecture d’un système expert typique est constitué de plusieurs modules qui s’interagissent :

* **L’interface utilisateur** sert à simplifier la communication, elle peut utiliser la forme question-réponse, le menu, le langage naturel etc.
* **La base de connaissances** contient les connaissances concernant la résolution du problème.
* **Le moteur d’inférence** applique une stratégie de résolution en utilisant les connaissances et ceci pour en dériver une nouvelle information.
* **La base de faits** 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 module d’explication** permet au système expert d’expliquer son raisonnement.
* **L’éditeur** permet l’édition des connaissances dans la base. Il est très important de remarquer la séparation faite entre les connaissances et l’inférence.
* – Cette séparation permet d’utiliser un codage différent, cela nous permet par exemple d’utiliser le langage naturel pour représenter les connaissances (sous forme Si .. ALORS.. par exemple).
* – Cette séparation permet au programmeur de se focaliser au codage des connaissances sans se soucier trop de la façon du codage du moteur d’inférence.
* – Cette séparation permet aussi de modifier les connaissances sans avoir un effet sur le codage du moteur d’inférence.
* – Cette séparation permet également de pouvoir tester plusieurs types d’inférence sur la même base de connaissances.
	1. **Domaines d’application** Les systèmes experts ont été conçus pour résoudre certains types de problèmes comme en médecine, en droit, en chimie, en éducation etc. Les catégories de problèmes abordés par les systèmes experts sont :

 – L’interprétation ou la construction d’une description abstraite à partir de données.

– Le prédiction des conséquences à partir de situations données.

– Le diagnostic d’une défaillance à partir d’un ensemble d’observations.

– La conception d’une configuration de composants à partir d’un ensemble de contraintes.

 – La planification d’une séquence d’actions pour l’accomplissement d’un ensemble de buts à partir de certaines conditions de départ et en présence de certaines contraintes. – La réparation d’un dysfonctionnement.

– le contrôle du comportement d’un environnement complexe.

**1.3 Problèmes adaptés aux systèmes experts** Les chercheurs ont défini un ensemble informel de critères pour déterminer si un problème est adapté ou non à être résolu par la technologie système expert :

 a. Le besoin d’une solution doit justifier le coût et l’effort de la construction d’un système expert.

b. L’expertise humaine n’est pas valable dans toutes les situations dont on a besoin.

c- Le problème peut être résolu en utilisant une technique de raisonnement symbolique.

d. Le domaine est bien structuré.

e. Le problème ne peut pas être résolu en utilisant des méthodes traditionnelles de calcul.

f. La coopération entre experts de domaine existe.

g. Le problème est de taille considérable.

* 1. **Processus d’ingénierie de connaissance** Les personnes concernées par le développement d’un système expert sont :

 – l’ingénieur de connaissance qui est un expert en langage IA. Son rôle est de trouver les outils et les logiciels nécessaire pour l’accomplissement du projet, d’aider l’expert du domaine à expliciter sa connaissance et d’implanter cette connaissance dans la base de connaissances.

– l’expert du domaine qui fournit les connaissances nécessaires liées au problème.

– l’utilisateur final dont le rôle est de spécifier l’application et de déterminer les contraintes de la conception.

En général, le travail commence par un interview entre l’ingénieur de connaissance et l’expert du domaine. L’ingénieur essaie de comprendre le domaine, d’observer l’expert pendant son travail. Une fois l’expert ait obtenu des informations complètes et précises sur le domaine ainsi que sur la résolution du problème, il pourrait entamer la tâche de la conception du système. Il choisit la façon de la représentation des connaissances, Il détermine le type du raisonnement et la stratégie utilisée (chaînage avant ou arrière, en profondeur ou en largeur). Il conçoit de même l’interface utilisateur. Le prototype obtenu doit être capable de résoudre correctement un problème typique. Ce prototype doit être testé et affiné par l’ingénieur et l’expert du domaine en même temps. Le prototype peut être complété au fur et à mesure en ajoutant des nouveaux éléments dans la base de connaissance. Souvent, à la fin de ce travail progressif, l’ingénieur serait amené à refaire une version plus propre qui réécrit la connaissance d’une façon plus sommaire. Dans son rôle, l’ingénieur de connaissance doit traduire l’expertise informelle en un langage formel adapté au mode du raisonnement du système.

**1.5 Acquisition de connaissance** Dans son rôle, l’ingénieur de connaissance doit traduire l’expertise informelle en un langage formel adapté au mode du raisonnement du système. Plusieurs points doivent être soulevés concernant l’acquisition des connaissances :

a. La compétence humaine n’est pas souvent accessible via la conscience. Avec l’expérience acquise, la compétence et la performance d’un expert s’installe et opère dans l’inconscient. Par conséquence, il est difficile aux experts d’expliciter son savoir-faire.

b. L’expertise humaine prend souvent la forme du savoir comment plus que la forme du savoir quoi.

c. L’expertise humaine représente un modèle individuel ou un modèle de communauté. Ces modèles sont soumis aux conventions et aux procédés sociaux.

d. L’expertise change et peut subir des reformulations radicales. A cause de la complexité et de l’ambiguïté posées par le problème de l’acquisition de connaissances, l’ingénieur de connaissances doit avoir un modèle conceptuel lui permettant de faire la liaison entre l’expertise humaine et le langage de programmation, ce modèle constituera ce qu’on appellera représentation de connaissances

**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 ¬ actif

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

 – Dans un système d'ordre 1, on pourra trouver : ∃ X maladie(X) et X ≠ 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 prémisse et s'il existe des règles mentionnant des klebsiellas dans une prémisse 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,...

**Moteur d’inférences :**

 **Chaînage 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 union 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 vérfiées.

– 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 chainage 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 pour un 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 seismologiques 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 requis 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.