

Université de Jijel

Faculté des Sciences Exactes et Informatique

Département d’Informatique

1^{ère} Année Master IA (S2)

Intitulé de l’UE : UEF3

Intitulé de la matière : Raisonnements et Exploitation de la Connaissance (REC)

Chargé de Matière Mme Kerada Ramdane Ouidad

Maintien de Cohérence dans les bases de connaissances

Partie 2

L’explication : une fonction importante des systèmes experts

1. Classification des raisonnements

Une division traditionnelle répartit les raisonnements en deux groupes complémentaires, opposés par le sens de leur démarche : la déduction et l’induction

Certains ont rejeté l’induction hors d’une théorie du raisonnement. Les uns parce qu’elle n’est pas, dans le cas général, une démarche rigoureuse, valable par sa forme seule, et qu’elle échappe ainsi aux prises d’une logique formelle. On en trouve un exemple, présenté sous la forme d’un système exhaustif, chez Pierce, qui répartit les raisonnements élémentaires en trois classes : déduction, induction, abduction (= raisonnement hypothético-déductif).

Dans un raisonnement déductif, on peut citer l’exemple suivant :

Règle : Tous les livres dans cette bibliothèque sont des livres d’histoire.

Cas : Ces livres sont empruntés de cette bibliothèque.

Résultat : Ces livres sont des livres d’histoire

Dans l’induction, on aboutit à la règle en partant d’un cas et d’un résultat :

Cas : Ces livres sont empruntés de cette bibliothèque.

Résultat : Ces livres sont des livres d’histoire

Règle : Tous les livres dans cette bibliothèque sont des livres d’histoire.

Enfin, dans l’abduction, que Pierce appelle aussi hypothèse, on aboutit au cas en partant de la règle et d’un résultat :

Règle : Tous les livres dans cette bibliothèque sont des livres d’histoire.

Résultat : Ces livres sont des livres d’histoire

Cas : Ces livres sont empruntés de cette bibliothèque.

Suite à ces raisonnement, la déduction prouve que quelque chose doit être, l’induction montre que quelque chose est effectivement, l’abduction suggère que quelque chose pourrait être. « La tendance constante du processus inductif est de se corriger lui-même : c’est là son essence. La probabilité de sa conclusion consiste uniquement dans le fait que si la vraie valeur du rapport recherché n’a pas été atteinte, une extension du processus inductif conduira à une valeur de plus en plus approchée. » (Théorie des grands nombres : Bernoulli)

2. Générer une « bonne » réponse

Des chercheurs se sont intéressés au manque de connaissances en complétant la base de connaissances par des connaissances nécessaires pour expliquer le raisonnement du système. Un système peut à présent fournir des explications plus riches. Par contre, la richesse et la complexité de la base de connaissances qui en résultent impliquent que le système soit capable de sélectionner et d'organiser en texte cohérent les informations à présenter à l'utilisateur. Le problème de sélection est d'autant plus compliqué que le système doit tenir compte du type d'utilisateur auquel il a affaire. Ces contraintes nous permettent de définir les caractéristiques que doit présenter la réponse du système pour être efficace et appropriée :

- **informative** : elle doit contenir des informations que l'utilisateur ne connaît pas encore
- **cohérente** : les informations doivent être organisées et structurées ;
- **compréhensible** : l'utilisateur doit obtenir les informations dans un langage accessible et acceptable ;
- **pertinente** : la réponse doit permettre à l'utilisateur de progresser vers la compréhension de ce qui lui posait question.

Ces caractéristiques ont déjà été largement étudiées en LN en tenant compte, de différents contextes éventuels. Par contre, les techniques utilisées dans les systèmes-experts paraissent relativement frustes. Il faut donc allier les travaux faits dans ces deux domaines.

- découpler la génération d'explications de la ligne de raisonnement du système
- étudier le problème de génération d'explications en temps que tel.

Toutefois, les techniques actuelles ne résolvent pas tous les problèmes. La production d'explications par un système-expert reste encore un champ d'investigations très actif, notamment en linguistique informatique. Ainsi, les fonctionnalités que l'on pourrait attendre d'un module d'explication restent à développer.

3. Des résolveurs de problèmes produisant des explications

Système Expert Explicatif = Système Expert + Module d'Explication

Le rôle du Module d'Explication : « rendre compte » du raisonnement suivi par le Système Expert, pour cela, il exploite en donnée une « trace » du raisonnement.

Deux types d'interlocuteurs sont à distinguer :

- L'ingénieur de la connaissance
- L'utilisateur final

➤ Des explications destinées à l'ingénieur de la connaissance

Objectif : rendre **explicite** la façon dont les connaissances sont interprétées

- Pourquoi le SE a-t-il (n'a-t-il pas) conclu sur tel fait ?
- Pourquoi / Comment le SE a-t-il utilisé telle connaissance ?
- Pourquoi le SE n'a-t-il pas utilisé telle connaissance ?

L'explication =

Une aide indispensable pour l'acquisition et la mise au point de la Base de connaissances

➤ Des explications destinées à l'utilisateur final

- Permettre de comprendre le raisonnement :
 - Pourquoi le SE recommande-t-il telle thérapie ?
 - Comment le SE a-t-il établi son diagnostic ?
- Convaincre du bien-fondé des conclusions / du raisonnement :
 - Le SE est-il certain de son diagnostic ?
 - Peut-il justifier sa conclusion ?
 - Peut-il justifier son raisonnement ?
 - Pourquoi le SE n'a-t-il pas pris telle décision ?
- Enseigner la démarche de résolution de problèmes
 - N'y avait-il pas plusieurs raisonnements possibles permettant d'aboutir à la même conclusion ?

L'explication =

Une faculté importante dont dépend l'acceptation du logiciel auprès de l'utilisateur.

➤ *Expliquer nécessite de mettre en œuvre de nombreuses connaissances*

Exemples de connaissances « explicatives »

- Des connaissances portant sur le **domaine d'application** :
 - Exp : pour justifier les connaissances de résolution de problèmes.
- Des connaissances portant sur l'**interlocuteur** :
 - Exp : pour ne pas lui réexpliquer quelque chose qu'il connaît déjà.
- Des connaissances de **rhétorique** :
 - Exp : est-il nécessaire d'argumenter sur le moment, ou bien attendre de savoir si l'utilisateur a bien compris ?
- Des connaissances de **pédagogie** :
 - Exp : lorsque l'élève demande à connaître la méthode de résolution, simplement lui indiquer la « clé » du raisonnement.

Exemple de dialogue explicatif

Un SE « fictif » en diagnostic technique utilisé par des techniciens en milieu industriel.

Technicien : Quelles sont tes conclusions ?

Sys : La panne résulte d'une défaillance du composant C26

Technicien : Ton diagnostic est-il certain ?

Sys : Les symptômes que vous m'avez communiqués m'ont fait penser à plusieurs diagnostics possibles. La plupart d'entre eux ont pu être écartés, car peu probables compte tenu de la récence du matériel. Seuls deux diagnostics pouvaient être retenus. Celui que je viens de fournir me paraît le plus plausible.

Technicien : Pourquoi la panne ne viendrait-elle pas du bloc moteur ?

Sys : Je n'ai pas envisagé cette possibilité. Mais vous auriez dans ce cas des symptômes caractéristiques autres que ceux que vous m'avez cités.

Second exemple de dialogue explicatif

PEDAG = un SE « fictif » en résolution de problèmes de mathématiques, utilisé en milieu scolaire par des élèves

Elève : Quelle méthode as-tu employée ?

PEDAG : Il fallait remarquer que la fonction était impaire et utiliser les identités remarquables pour simplifier.

Elève : Pourquoi avoir simplifié l'expression dès le début ?

PEDAG : Pour la commodité des calculs. Mais on aurait très bien pu laisser l'expression sous sa forme initiale.

5. Architecture d'un Système Expert Explicatif

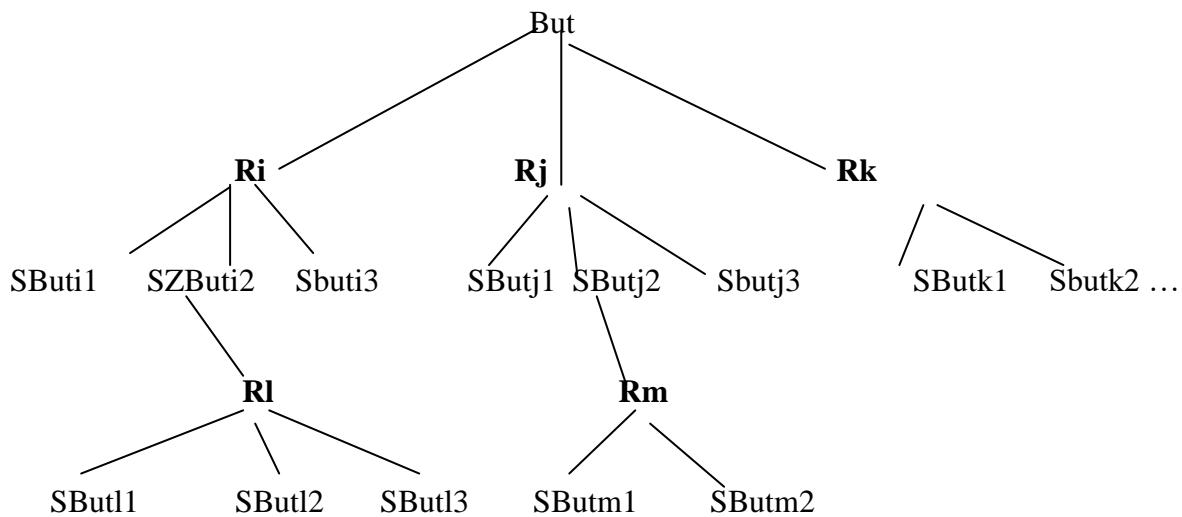
| Système Expert Explicatif | |
|---|---------------------------|
| Système Expert | Module d'Explication |
| Connaissances de résolution de problème | Connaissances Explicative |
| Mécanismes d'inférences | Générateur d'Explication |

MYCIN : un premier Système Expert Explicatif 1976 : thèse de Davis

Les capacités d'explication de MYCIN correspondent à des fonctionnalités de base que l'on trouve associées à tout moteur de base de règles

- L'Arborescence de Déduction
- Fonctionnement du Module d'Explication
- Discussion

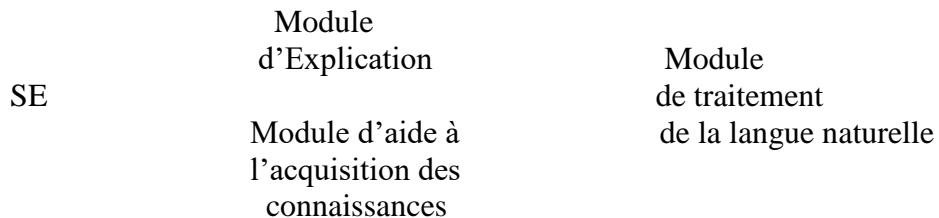
L'Arborescence de Déduction : une trace du raisonnement



Architecture du SEE MYCIN

Objectif du ME : « fournir au Système Expert une **connaissance sur ses propres connaissances** et la façon dont il les interprète »

SEE : MYCIN



Exemples de questions portant sur le contenu de la Base de Règles

Utilisateur : Prescrivez-vous parfois de la carbenicilline pour les infections par Pseudomonas ?

MYCIN : Règles concernées : REGLE024, REGLE068 et REGLE138 Lesquelles désirez-vous voir ?

Utilisateur : REGLE068

MYCIN : {imprime la règle}

Utilisateur : Le chloramphenicol est-il prescrit pour les infections par salmonelles ?

6. *Les métarègles*

Elles expriment une connaissance sur la connaissance. Elles permettent de partitionner une base de connaissance et elles représentent les stratégies de recherche, les temps de recherche sont aussi liés au nombre de règles. C'est aussi une façon de réduire l'espace de recherche.

Comment ?

- **Utiliser certaines règles de préférence :**
SI informatique **ALORS** considérer les règles **R50, R61, R63** ensuite les règles **R15, R12**.
- **Exclure certaines règles :**
SI I.A. **ALORS** ne pas utiliser les règles **R20 et R23**.
- **Proposer un plan de recherche :**
SI rougeole **ALORS** considérer les règles **R3, R4...**

Exemples de métarègles :

- **MR1 : SI** malade=femme **ALORS** considérer les règles **R15, R16**.
- **MR2 : SI** malade=enfant **ALORS** considérer les règles **R1, R2**.

7. Rôle du contexte au niveau des explications

Les raisonnements des experts ne sont pas simples à cerner et sont souvent fortement liés aux détails des situations. Le contexte joue un rôle capital dans de nombreuses expertises humaines. Actuellement, plusieurs voies sont explorées pour intégrer le contexte dans des systèmes d'aide à la décision ou dans des systèmes autonomes. Le contexte joue un rôle important dans les explications qu'un système d'aide peut apporter à un utilisateur.

Selon Leake [1992], les explications sont nécessaires quand il y a un conflit entre un événement et les événements attendus par l'utilisateur (le modèle que l'utilisateur a des événements). Les conflits de ce genre sont une propriété intrinsèque des interactions entre les événements et le contexte. Quand un tel conflit de croyance survient, les explications doivent le résoudre en identifiant les éléments ayant poussé l'utilisateur à cette attente.

Schmidt [1995] souligne que le contexte de l'utilisateur et surtout son évolution doivent être pris en compte pour la fourniture d'explications. Il prend l'exemple d'un distributeur automatique (interface homme-machine réduite à sa plus simple expression). Le concepteur a le choix de la quantité d'informations mise à disposition de l'utilisateur : plus le système fournit d'explications, moins l'utilisateur a besoin de savoir-faire. Pourtant, à force d'utiliser le système, un utilisateur acquiert le savoir-faire nécessaire et n'a donc plus besoin des explications. Cet exemple montre que les systèmes doivent adapter leurs explications aux connaissances et savoirs-faire de l'utilisateur (son contexte) et à l'évolution de ce contexte de l'utilisateur au fur et à mesure de l'interaction avec le système.

Dans une telle coopération, l'utilisateur doit également être informé de l'état du système concernant l'élaboration des explications. Le système doit donc être capable de fournir des explications sur ses propres explications, notamment en utilisant le contexte d'explication.