PDF pour 25440-02 sommaire

Le présent fichier PDF **Sommaire** fournit des extraits du document

Mylog, générateur de système experts

Manuel de Référence

Edité par la **Société Delphia**

Ce Manuel de référence a été édité an 1987 sous forme de classeur (183 pages) par la société Delphia, size à Seyssinet près de Grenoble.

Ce document est disponible sous le numéro **AC 25440-02** dans l'inventaire de la collection de l'ACONIT, Association pour un conservatoire de l'informatique et de la télématique, Grenoble.

Le présent PDF fournit un scan de :

-La page d'en-tête,
-La page d'identification de la société Delphia,
-La table des matières (7 pages),
-L'introduction (14 pages),
-Copie de quelques extraits constituant un résumé du système expert.

Fiche rédigée le 22/5/2020 par Jean Ricodeau.

MYLOG

Générateur
de
Systèmes Experts

Manuel de Référence

Delphia

27 Avenue de la République
38170 Seyssinet
France

Le **Manuel de Référence de Mylog** est composé de sept sections.
Ce sont dans l'ordre :

1 L'Introduction

qui est une présentation générale des Systèmes Experts et de Mylog.

2 La Représentation de la connaissance

qui contient la description de la syntaxe de Mylog.

3 Le Fonctionnement du moteur d'inférence

qui présente, de manière d'abord générale, les concepts du fonctionnement, puis leur application dans Mylog.

4 Les Prémisses prédéfinies

qui répertorie les outils de contrôle et de manipulation de certaines formes de représentation de la connaissance.

5 La Représentation centrée objets

qui est dédié à ce formalisme particulier que sont les objets.

6 Manuel du concepteur

qui décrit l'interface de Mylog destinée au Concepteur de Systèmes Experts.

7 Annexes

La carte syntaxique et l'index des mots réservés.

Table des matières

2 REPRESENTATION DE LA CONNAISSANCE

5 REPRESENTATION CENTREE OBJETS

6 MANUEL DE CONCEPTION

1 Introduction

Dans de nombreuses activités intellectuelles, relevant du domaine des sciences exactes, sociales, humaines ou même de la vie courante, le savoir-faire des experts n'est en général pas structuré de manière à pouvoir être représenté sous forme d'algorithmes.

Dans tous ces domaines, le savoir-faire peut souvent être représenté comme un ensemble de connaissances spécifiques appropriées à une situation donnée. L'enchaînement de ces connaissances se décidant au fur et à mesure de l'évolution de la situation étudiée.

Une approche informatique classique convient donc assez mal à la modélisation du raisonnement humain, puisqu'il n'est pas envisageable de décrire systématiquement dans un ordre prévu à l'avance toutes les étapes qu'il parcourt.

C'est pourquoi les recherches se sont tournées vers de nouveaux outils et en particulier l'Intelligence Artificielle pour créer les Systèmes Experts.

1.1 Présentation d'un système expert

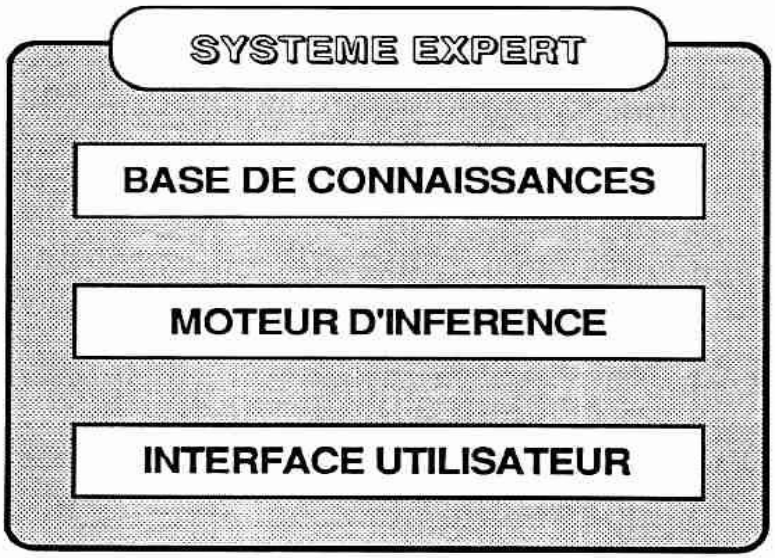
Un système expert est un ensemble de programmes dont l'objectif est de simuler le raisonnement humain. La simulation du raisonnement s'effectue sur des domaines très ciblés tels le contrôle de processus, l'analyse financière, le diagnostic médical, etc. La simulation du raisonnement humain de manière générale, fait encore partie des recherches en cours.

Les systèmes experts sont certainement les applications les plus prometteuses issues des recherches en Intelligence Artificielle. Ils sont d'ailleurs souvent confondus avec l'Intelligence Artificielle elle-même, car ils en sont la première application véritablement opérationnelle et intéressante pour l'industrie.

Les caractéristiques essentielles des systèmes experts sont :

- la séparation faite entre l'ensemble des connaissances mises en œuvre et les algorithmes de résolution qui les utilisent,
- le fait que la solution du problème ne passe pas par l'exécution d'une séquence d'instructions dans un ordre prévu à l'avance.

La première obligation pour créer un système expert, est évidemment la participation d'un expert du domaine cible à l'écriture du système expert. Cet expert doit formaliser son savoir, qui est introduit dans ce qui est appelé une **base de connaissances**. L'utilisation de ces connaissances est réalisée par le **moteur d'inférence** du système expert. Enfin une **interface utilisateur** est nécessaire pour que le système expert créé puisse être exploité par l'utilisateur final. Celle-ci doit être la plus conviviale possible.



1.1.1 La base de connaissances

La base de connaissances est la partie que doit construire l'expert avec ou sans l'aide d'un informaticien selon le degré de complexité du formalisme utilisé par le système expert pour représenter la connaissance.

Les connaissances constituant cette base sont de nature déclarative. Elles sont l'expression de relations, de situations, de faits et ne préjugent pas de l'utilisation qui en sera faite.

La base de connaissances est propre au domaine traité. Elle rassemble :

- toutes les connaissances qu'utilise un expert du domaine choisi, c'est-à-dire la description des objets et de leurs relations,
- la mise en évidence de cas particuliers,
- les différentes stratégies de résolution ainsi que leurs conditions d'applications.

Pour produire des résultats intéressants, la base de connaissances d'un système expert doit contenir des informations aussi précises, complètes et détaillées que celles d'un expert du domaine.

1.1.2 Le moteur d'inférence

Le moteur d'inférence est le programme qui construit les raisonnements à partir de la base de connaissances. Il est indépendant de l'application développée. Il suffit souvent de changer de base de connaissances pour que le système expert soit compétent dans un autre domaine.

L'ensemble des connaissances est donné en vrac, ou plus précisement dans un ordre qui ne constitue pas une séquence d'instructions mais une collection d'assertions. Le rôle du moteur d'inférence est de décider selon les besoins et donc en fonction de chaque situation nouvelle, quelles sont les connaissances à mettre en œuvre et dans quel ordre les utiliser.

1.1.3 L'interface utilisateur

Au moteur d'inférence s'ajoutent des interfaces indispensables pour assurer le dialogue homme-machine, interfaces destinées à l'utilisateur final du système expert.

Sans avoir besoin de connaissances particulières en informatique, tout utilisateur doit pouvoir :

- répondre aux questions posées par le système expert,
- suivre son raisonnement,
- lui poser des questions ou lui demander des explications.

1.2 Présentation d'un générateur de systèmes experts

La réunion d'un moteur d'inférence et d'un environnement de développement forme ce que l'on appelle un **générateur de systèmes experts.**

Les outils de l'environnement de développement

L'environnement de développement d'un générateur de systèmes experts comprend un certain nombre d'outils. Ces outils permettent d'aider le concepteur dans toutes les phases de la création d'un système expert.

Cet environnement de développement, s'il n'est d'aucune influence sur la valeur des raisonnements, joue un rôle essentiel dans la rédaction de la base de connaissances et la mise au point du système expert.

Les environnements de développement de générateurs de systèmes experts disponibles à l'heure actuelle comprennent tous au minimum :

- un analyseur syntaxique, donnant précisement le nombre et la nature des erreurs détectées à la lecture de la base de connaissances,
- un système de trace pas à pas de l'exécution du système expert, permettant une mise au point rapide de l'application,
- une protection de l'application développée, afin de préserver la confidentialité des connaissances.

Cet environnement de développement doit permettre entre autres choses :

- de détecter rapidement les connaissances induisant un comportement anormal du système expert et d'ajouter ou supprimer facilement des connaissances. Une base de connaissances n'est jamais définitivement complète ou immuable, le domaine d'expertise étant lui-même amené à évoluer.
- de justifier les résultats du système expert en indiquant les différentes étapes de son raisonnement ainsi que les connaissances mises en œuvre. Ces explications sont utiles à la fois au concepteur au moment de la mise au point, et à l'utilisateur final afin de l'aider à comprendre les résultats obtenus par le système expert.

1.3 Représentation de la connaissance

Il existe différents modes de représentation des connaissances, les principaux étant les deux suivants :

- les **règles de production**, qui peuvent être définies suivant la logique des propositions (ordre 0) ou suivant la logique des prédicats (ordre 1),
- la **représentation centrée objet**, qui est une représentation hiérarchisée des connaissances autorisant une manipulation dynamique des attributs d'objets et utilisant le puissant mécanisme d'héritage.

1.3.1 Les règles de production :

Les règles de production sont de la forme suivante :

si les prémisses A et B et C ... sont vérifiées
alors les conclusions D, E, F ... sont vérifiées.

Avec ce type de formalisme les connaissances d'un expert se décomposent en trois sous-ensembles :

- les **faits**, qui représentent la partie affirmée de l'univers d'expertise (connaissance objective),
- les **règles**, qui permettent de déduire à partir de la connaissance affirmée de nouveaux faits pour arriver aux conclusions du système expert,
- la **métaconnaissance**, qui précise l'utilisation des règles et définit ainsi des heuristiques pour aboutir aux conclusions de la meilleure façon possible.

Deux des propriétés essentielles des règles de production sont :

- leur indépendance les unes par rapport aux autres,
- leur caractère déclaratif.

L'indépendance des règles de production permet une mise à jour aisée de la base de connaissances, simplement par ajout, modification ou suppression d'une règle. L'exploitation des connaissances représentées sous forme déclarative, par opposition au caractère directif de la séquence d'instructions d'un programme, correspond à une étape importante dans l'évolution de l'informatique.

La représentation des connaissances sous forme de règles de production est un formalisme très adapté pour tous les raisonnements de type déductif et pour la traduction de la connaissance heuristique des experts, spécialement quand il s'agit de problèmes de type diagnostic, prévision, simulation, ...

1.3.2 La représentation centrée objets

La représentation centrée objets est un formalisme particulièrement bien étudié pour l'expression des connaissances de nature structurelle. Ce formalisme manipule des **objets** regroupés dans des classes en fonction de leurs caractéristiques communes.

Chaque classe est définie comme une **spécialisation** d'autres classes. Il en résulte une organisation hiérarchique des classes sous la forme d'un treillis, chaque classe possédant au minimum une classe ancêtre à partir de laquelle elle a été définie. Chaque objet d'une classe est aussi un objet des classes ancêtres, directes ou éloignées, de la classe à laquelle il appartient.

L'organisation hiérarchique des classes permet donc à un objet d'une classe donnée de bénéficier des informations des classes ancêtres de la sienne. Ce mécanisme, appelé **héritage**, permet une centralisation efficace de l'information.

Chaque objet ou **instance** de la classe possède un certain nombre de caractéristiques définies dans la classe, certaines de ces caractéristiques sont appelées **attributs** de l'instance. A un attribut peuvent être associées une valeur et des propriétés appelées **facettes**, permettant de calculer cette valeur ou de restreindre son domaine de définition.

A partir d'un objet défini dans une classe, il est possible de chercher si cet objet peut être une instance d'une classe plus **specialisée**, c'est-à-dire dépendante plus ou moins directement de la classe d'appartenance de l'objet. Ce mécanisme appelé **classification**, permet de formaliser facilement les problèmes d'identification d'objets et, plus généralement, les problèmes de diagnostics.

1.4 Caractéristiques d'un moteur d'inférence

Les caractéristiques d'un moteur d'inférence sont :

- l'ordre du moteur,
- les différents modes de chaînage disponibles.

1.4.1 L'ordre du moteur

L'ordre du moteur correspond à l'ordre de la logique utilisée lorsque les connaissances mises en œuvre sont représentées sous forme de règles de production. Cette logique est soit la logique des **propositions**, soit celle des **prédicats**.

1.4.1.1 L'ordre 0 et la logique des propositions

Dans la logique des propositions ou logique d'ordre 0, la représentation de la connaissance s'effectue à l'aide d'énoncés complètement instanciés.

Exemple :

```
dubois est cadre .
```

La logique des propositions ne permet pas de manipuler des ensembles d'entités ou des variables. Par exemple, il est impossible de traduire sous forme de proposition le fait que tout cadre est aussi un salarié de l'entreprise. Pour cela, il faudrait l'énoncer pour tous les cadres.

Exemple :

```
dubois est salarie.
dufour  est salarie.
 martin est salarie.
```

etc...

1.4.1.2 L'ordre 1 et la logique des prédicats

Dans la logique des prédicats ou logique d'ordre 1, il est possible d'introduire des variables dans l'énoncé des connaissances.

Exemple :

```
X est salarie si
  X est cadre.
```

C'est un formalisme très puissant englobant généralement d'autres types de représentation des connaissances et notamment l'algèbre relationnelle.

1.4.2 Les deux modes de chaînage

Il existe deux modes de chaînage traduisant deux modes de raisonnements différents. Suivant le type de raisonnement désiré, le moteur doit fonctionner avec l'un des deux modes de chaînage **avant** ou **arrière**.

Le premier est le raisonnement déductif, c'est-à-dire que le moteur d'inférence utilise la partie affirmée de la base de connaissances (les faits) pour tenter de déduire de nouvelles affirmations à l'aide des règles. Ce mode de fonctionnement est appelé chaînage avant.

En chaînage avant, la règle "si A alors B" n'est "déclenchée" par le moteur d'inférence que lorsque les **conditions** exprimées par la **prémisse** A sont vérifiées dans la base de connaissances. Ce type de raisonnement est dit **guidé par les données**.

Le second mode de raisonnement consiste à vérifier un certain nombre d'hypothèses. Le moteur d'inférence tente de prouver les hypothèses déclarées dans la base de connaissances l'une à la suite de l'autre, en remplaçant chacune des hypothèses par un ensemble d'assertions qui, si elles-mêmes peuvent être prouvées, permettront de déduire l'hypothèse de départ. Ce mode de fonctionnement est appelé chaînage arrière.

En chaînage arrière, la règle "si A alors B" est utilisée pour remplacer la **conclusion** B à démontrer, par un ensemble de **conclusions** déduit de la prémisse A. Ce type de raisonnement est dit **guidé par les buts**.

1.5 Caractéristiques de Mylog

Mylog est un kit de développement de systèmes experts, hybride puisqu'utilisant deux modes de représentation des connaissances, règles de production et représentation centrée objets.

Le moteur d'inférence de Mylog est un moteur d'ordre 1 pouvant fonctionner suivant les deux modes de chaînage, avant ou arrière.

Mylog permet la structuration de la base de connaissances en modules indépendants. Pour chaque module, le mode de chaînage et l'utilisation de coefficients de vraisemblance est à préciser.

Mylog offre des moyens puissants de structuration et de représentation des connaissances par l'intermédiaire :

- du découpage modulaire de la base de connaissances,
- des règles en logique du premier ordre,
- de la représentation centrée objets,
- de l'utilisation de coefficients de vraisemblance,
- des deux modes de chaînage possibles.

De plus Mylog est doté d'un environnement complet pour le développement de grosses applications, mettant à la disposition des concepteurs des outils très ergonomiques pour la création, la mise au point et l'exploitation de bases de connaissances importantes.

1.5.1 Modularité

Les bases de connaissances Mylog sont structurées en modules constituant des sous-bases indépendantes et communiquant par l'intermédiaire d'interfaces. Cette structuration amène des avantages décisifs au niveau :

- de l'analyse du problème,
- de la réalisation de la base de connaissances,
- de l'utilisation finale,
- des performances d'exécution.

La décomposition modulaire de la base de connaissances est une structuration qui réduit la complexité du problème par un découpage en éléments plus faciles à maîtriser. Les experts peuvent alors travailler à plusieurs et indépendamment, et mettre au point les modules séparement. Cette technique rend envisageable la réalisation de bases de connaissances importantes mettant à profit les connaissances de plusieurs experts.

La modularité, en réduisant la portée des règles et des définitions de classes, permet :

- d'optimiser la recherche des règles ou des classes candidates,
- d'éviter que les performances ne se dégradent sur les bases de connaissances de grande taille.

L'utilisation de l'espace mémoire est également améliorée par le découpage modulaire, qui autorise le dépassement des limites de la mémoire centrale ou évite l'encombrement des mémoires virtuelles, par une gestion en overlay des modules.

La modularité permet aussi d'activer pour chaque module un mode de fonctionnement différent du moteur d'inférence. Lorsque, sur une même base de connaissances, le moteur utilise les deux modes de chaînage, le moteur fonctionne en chaînage **mixte**.

1.5.2 Règles de production et représentation centrée objets

Une des principales richesses de Mylog réside dans la combinaison des règles de production d'ordre 1 et de la représentation centrée objet (RCO). Cette représentation hybride permet de couvrir tous les aspects de la représentation des connaissances, depuis les éléments structurels pour les RCO jusqu'aux aspects les plus déductifs pour les règles de production.

L'utilisation dans une même base de connaissances, de règles et d'objets nécessite une communication bidirectionnelle entre les deux formalismes.

La liaison règles -> objets se fait principalement par l'intermédiaire de **demandes de valeurs,** permettant d'obtenir une instance d'une classe sur laquelle peuvent être imposées un certain nombre de contraintes.

La liaison objets -> règles s'établit au niveau des attributs d'une classe de deux façons différentes, soit par l'attachement procédural, soit par la définition de contraintes. Suivant le cas, la règle est déclenchée lors de l'initialisation ou de la consultation de l'attribut.

1.5.3 Les coefficients de vraisemblance

Les règles de production qui traduisent l'expérience de l'expert ne reflètent pas toujours des implications logiques certaines, mais parfois seulement ses propres convictions. La traduction de la confiance de l'expert dans chacune des règles se fait par l'association de coefficients de vraisemblance à chacune d'elles.

Le mode de propagation de ces coefficients est le même que celui utilisé dans le système expert Mycin et peut être, de plus, redéfini par l'expert.

La logique des prédicats s'appuie sur la logique formelle. L'introduction de coefficients de vraisemblance dans la logique des prédicats permet une approche des concepts de la logique floue.

1.6 Environnement de développement de Mylog

L'environnement de développement Mylog comprend un ensemble d'outils destinés à faciliter la mise au point, la validation et l'exploitation des bases de connaissances créées.

1.6.1 L'analyseur syntaxique

L'analyseur syntaxique transforme le texte d'une base de connaissances en une forme exploitable par le moteur d'inférence et fournit une liste détaillée des erreurs rencontrées. A la fin de l'analyse deux fichiers sont générés qui sont respectivement :

- le ficher exécutable par le moteur d'inférence,
- le dictionnaire texte.

1.6.2 Le dictionnaire texte

A la fin de l'analyse, Mylog fournit un dictionnaire texte contenant des renseignements essentiels sur les connaissances contenues dans le module analysé, pour la mise au point de ce module.

Ces informations permettent de vérifier la cohérence de la base de connaissances, de détecter éventuellement des fautes d'orthographe, d'isoler les règles qui ne sont pas utilisées et de découvrir les informations manquantes ou redondantes.

1.6.3 La trace interactive

En cours de développement, l'exécution pas à pas du moteur d'inférence peut être visualisé. A l'aide de ce mécanisme de trace, l'expert peut suivre le cheminement du moteur d'inférence et le diriger interactivement.

Cet outil de haut niveau permet de déceler les causes d'un comportement imprévu du système expert et présente un important intérêt didactique pour l'expert devant se familiariser avec les mécanismes d'inférence de Mylog.

1.6.4 Le journal

Chaque session Mylog peut être conservée grâce à un journal répertoriant :

- les questions posées par le système expert avec les réponses que l'utilisateur a fournies,
- les explications et commentaires fournis par le système expert,
- les différents messages produits par Mylog.

1.6.5 La sauvegarde des réponses

A la fin de chaque exécution, l'utilisateur a la possibilité de sauvegarder une partie ou la totalité des réponses qu'il a fournies au cours de l'exécution précédente.

La sauvegarde des réponses permet :

- d'arrêter momentanément une expertise pour la reprendre au même endroit un peu plus tard,
- de conserver des informations qui pourront être utilisées au cours d'une nouvelle session,
- de modifier une partie des réponses afin d'étudier le comportement du système expert dans ce cas,
- d'effectuer des séries de tests systématiques afin d'étudier le comportement du système expert dans des situations classiques jusqu'aux cas extrêmes.

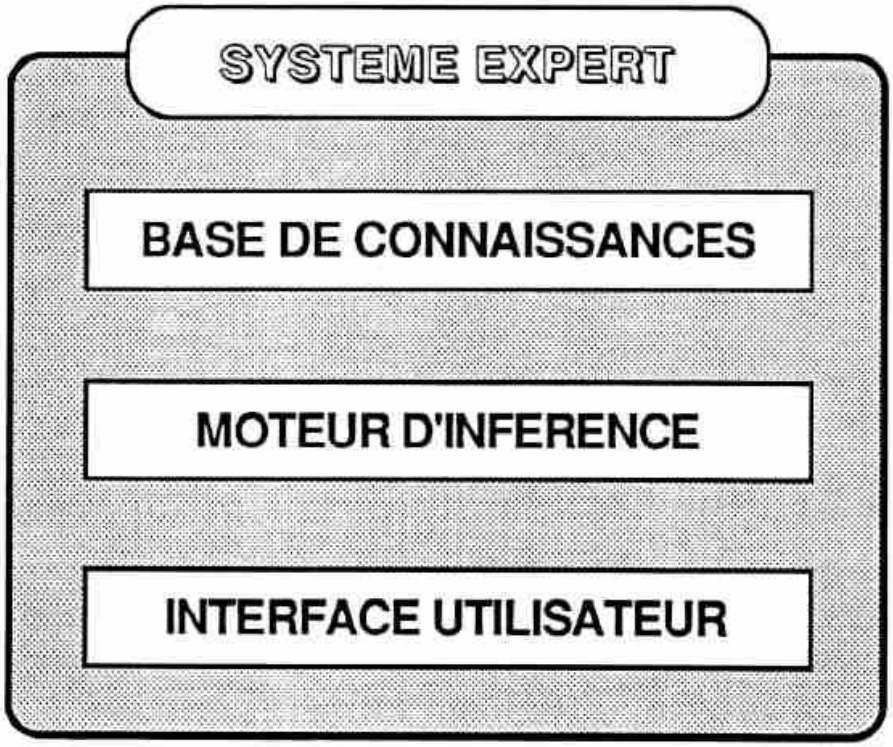
1.6.6 La protection de l'application

A l'aide des modules exécutables de la base de connaissances et d'une version réduite de Mylog, l'expert peut fournir à l'utilisateur un système expert protégé. Les modules sont des fichiers non accessibles sous éditeur. La version réduite de Mylog ne permet ni l'analyse, ni l'accès au dictionnaire et à la trace.

1.7 Ouverture

L'accès à un langage externe est indispensable pour permettre au système expert de communiquer avec son environnement (capteurs, SGBD, graphique...). D'autre part, les applications classiques tendent à intégrer des modules experts dans leur structure. Il est donc primordial qu'une communication et des appels bidirectionnels puissent avoir lieu.

Mylog permet, via Delphia Prolog, d'accéder à tout langage et peut même être invoqué avec une base de connaissances depuis un langage externe. L'accès aux systèmes de gestion de base de données relationnelles est immédiat, par l'utilisation de la bibliothèque **STORIA**.



Delphia Manuel de Référence Mylog

1.1.1 La base de connaissances

La base de connaissances est la partie que doit construire l'expert avec ou sans l'aide d'un informaticien selon le degré de complexité du formalisme utilisé par le système expert pour représenter la connaissance.

Les connaissances constituant cette base sont de nature déclarative. Elles sont l'expression de relations, de situations, de faits et ne préjugent pas de l'utilisation qui en sera faite.

La base de connaissances est propre au domaine traité. Elle rassemble :

- toutes les connaissances qu'utilise un expert du domaine choisi, c'est-à-dire la description des objets et de leurs relations,
- la mise en évidence de cas particuliers,
- les différentes stratégies de résolution ainsi que leurs conditions d'applications.

Pour produire des résultats intéressants, la base de connaissances d'un système expert doit contenir des informations aussi précises, complètes et détaillées que celles d'un expert du domaine.

1.1.2 Le moteur d'inférence

Le moteur d'inférence est le programme qui construit les raisonnements à partir de la base de connaissances. Il est indépendant de l'application développée. Il suffit souvent de changer de base de connaissances pour que le système expert soit compétent dans un autre domaine.

L'ensemble des connaissances est donné en vrac, ou plus précisement dans un ordre qui ne constitue pas une séquence d'instructions mais une collection d'assertions. Le rôle du moteur d'inférence est de décider selon les besoins et donc en fonction de chaque situation nouvelle, quelles sont les connaissances à mettre en œuvre et dans quel ordre les utiliser.

Les règles

Les règles permettent de déduire de nouveaux faits à partir de ceux déjà présents dans la base de connaissances ou pouvant être déduits.

Une règle est composée d'une partie **conclusions** (ce qui peut être déduit) et d'une partie **prémisses** (ce qui doit permettre la déduction), séparées par le mot réservé **si**.

La partie prémisses est en général formée de plusieurs prémisses séparées par le mot réservé **et**.

Exemple :

participation a la construction **si**
nombre salaries est X **et** X > 9.

Si la prémisse *nombre salaries est 123* est prouvée dans la base de connaissances, le moteur d'inférence évalue l'expression booléenne avec la variable *X* associée à la valeur *123*, le résultat étant vrai, la conclusion *participation a la construction* est déduite.

Les déclarations de buts

Lorsque le moteur d'inférence simule un raisonnement qui consiste à vérifier des hypothèses formulées par l'expert, il fonctionne en chaînage arrière.

Les buts sont les hypothèses que doit vérifier le moteur d'inférence à partir des informations contenues dans la base de connaissances.

Les buts d'un module sont déclarés dans des phrases commençant par le mot réservé **but** et sont séparés par le mot réservé **ou**.

Exemple :

but montant charges sociales employeur Z pour X
ou montant charges sociales salarie Z pour X .

La base de connaissances contenant cette déclaration de buts, permet de calculer le montant des charges sociales par salarié pour l'employeur et pour le salarié.

Pour le moteur d'inférence, cette déclaration de buts est équivalente aux deux suivantes :

but montant charges sociales employeur Z pour X.
but montant charges sociales salarie Z pour X.

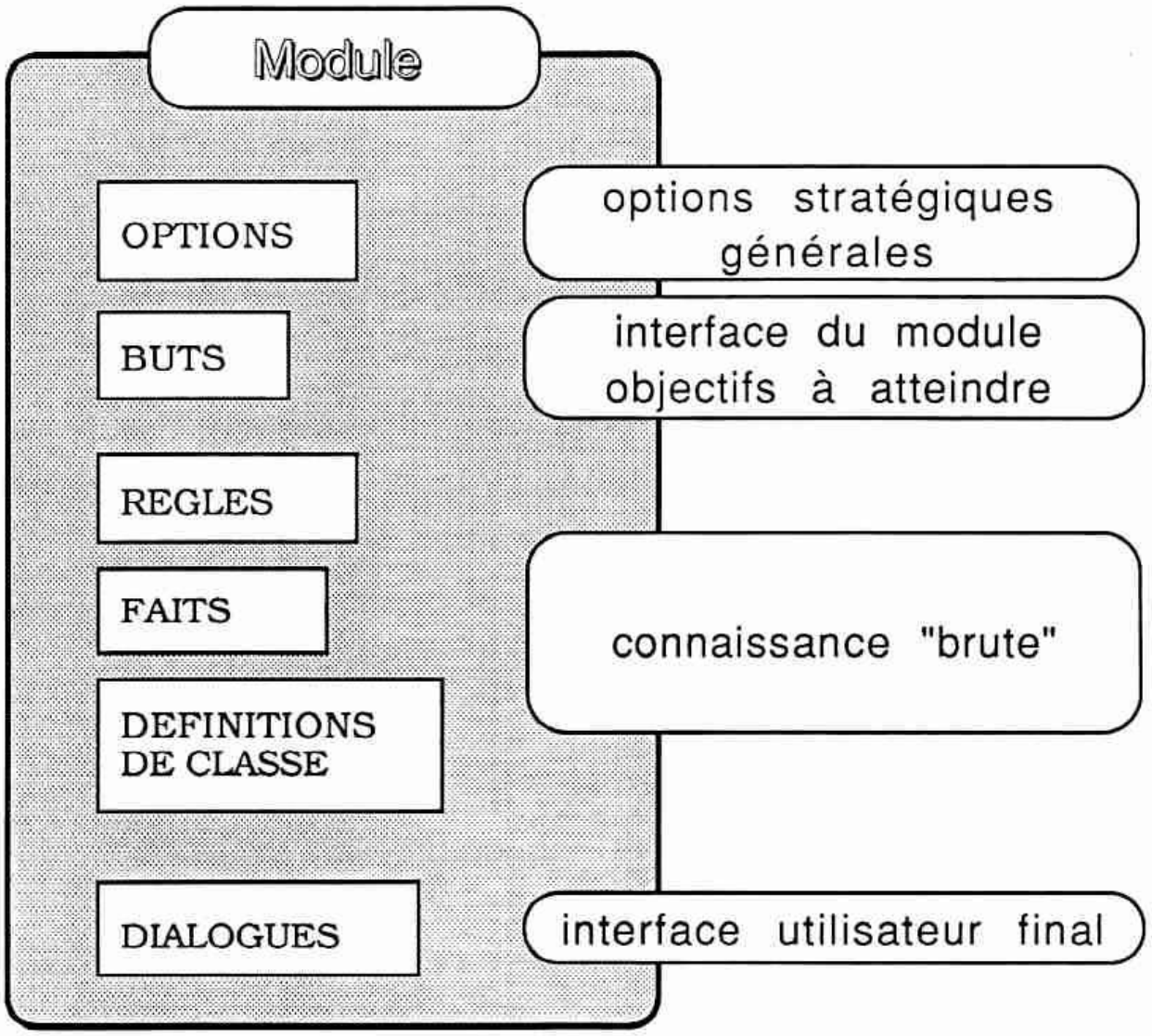
La **génération d'hypothèses** est une technique qui permet de ne pas déclarer explicitement les buts d'un module d'une base de connaissances, mais de les faire générer par un autre module de cette base.

Cette technique est très utile en particulier pour les problèmes de diagnostic, le module générant les hypothèses éclaircit le problème rapidement à partir de faits, règles et dialogues assez généraux et exporte ses résultats dans le module demandant la génération d'hypothèses. Les résultats exportés deviennent les buts de ce module qui peut effectuer une analyse plus fine du problème. Le nombre d'hypothèses probables ayant été réduit, le temps d'exécution du module est d'autant diminué.

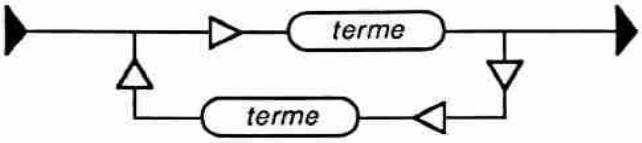
Les modules

A partir des éléments de base de Mylog qui viennent d'être présentés, il est possible de construire une petite base de connaissances, ou un module d'une base de connaissances importante.

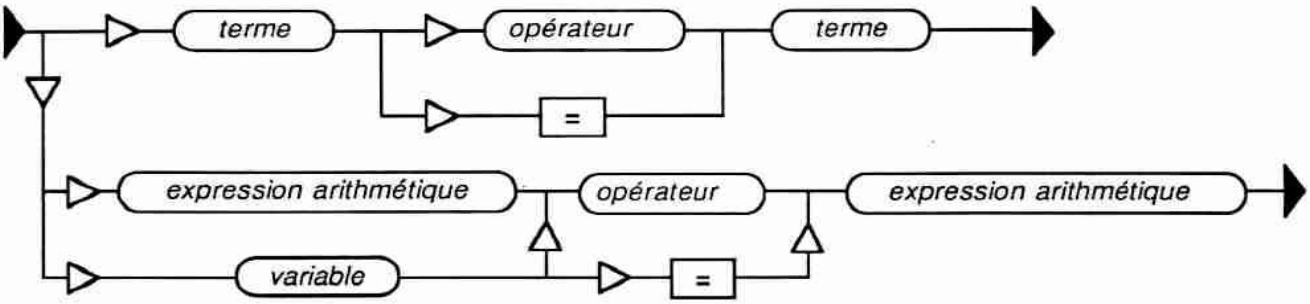
Tous les modules d'un système expert Mylog sont construits de la façon suivante :



Littéral

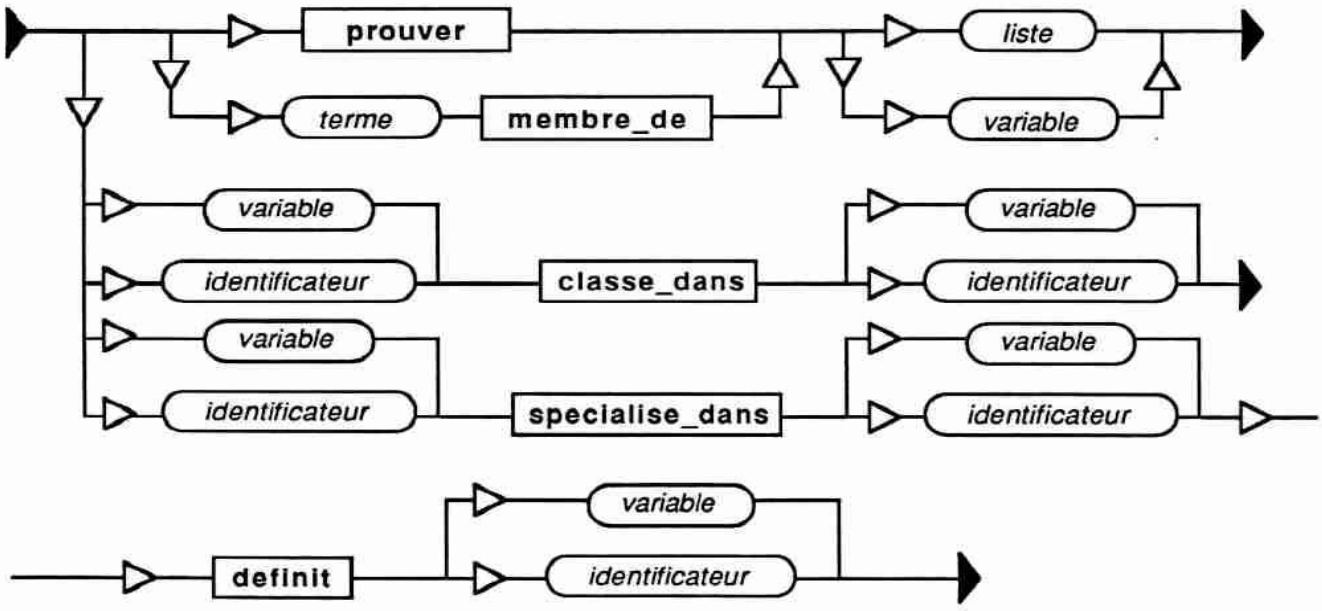


Expression booléenne

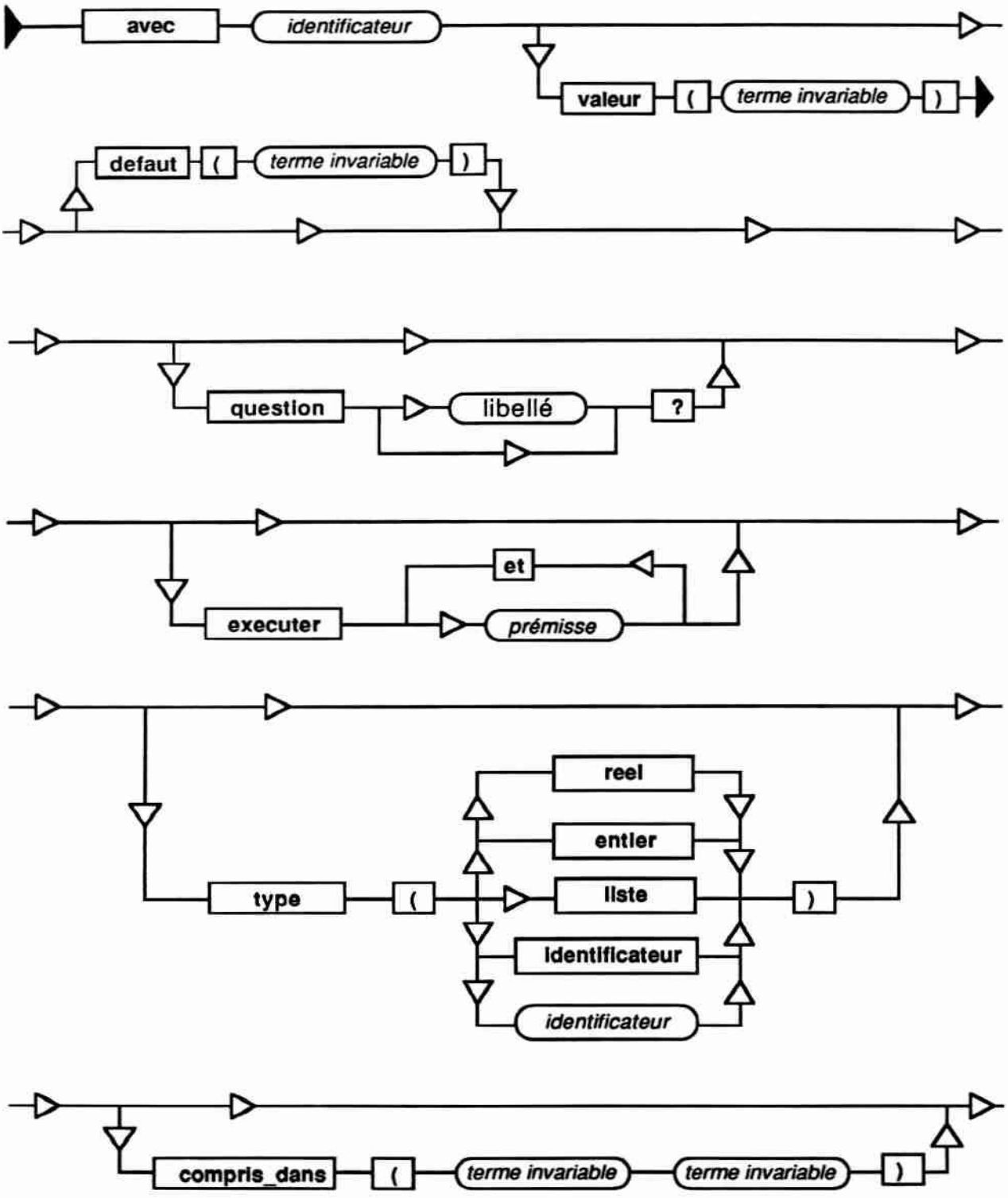


Dialogue booléen

Prémisse prédéfinie booléenne



Attribut


avec
identificateur
valeur
(
terme invariable
)
defaut
(
terme invariable
)
question
libellé
?
et
executer
prémisse
reel
entier
type
(
liste
)
identificateur
identificateur
compris_dans
(
terme invariable
terme invariable
)