



**HAL**  
open science

# UNE METHODE ORIENTEE OBJET POUR L'ELABORATION DE SYSTEMES EXPERTS

Danielle Boulanger, Joël Colloc

► **To cite this version:**

Danielle Boulanger, Joël Colloc. UNE METHODE ORIENTEE OBJET POUR L'ELABORATION DE SYSTEMES EXPERTS. VOLCANS-IA 93, 1993, Clermont-Ferrand, France. <hal-02586114>

**HAL Id: hal-02586114**

**<https://hal.science/hal-02586114v1>**

Submitted on 15 May 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# UNE METHODE ORIENTEE OBJET POUR L'ELABORATION DE SYSTEMES EXPERTS

D. BOULANGER, J. COLLOC

I.A.E. - Université Jean Moulin Lyon III  
URA CNRS 1257  
15, Quai Claude Bernard  
69007 LYON FRANCE

Tél : (33) 72.72.20.36

Fax : (33) 72.72.20.50

## RESUME

Ce travail présente une méthode exploitant les concepts d'un modèle orienté objet, pour la construction de systèmes experts dans des domaines empiriques comme la Gestion, les Sciences Humaines, la Médecine...

Cette méthode repose sur un modèle original dont les points forts sont : un double niveau conceptuel (intérieur et extérieur à l'objet), un héritage multiple ascendant intérieur et sans conflit, des fonctions d'évaluation et des fonctions dynamiques, un effet zoom offrant une perception adaptée du domaine de l'expert.

Les différentes phases de la méthode sont décrites. La méthode sera mise en oeuvre ultérieurement à travers un Atelier de Conception de Systèmes Experts (ACSE).

**MOTS CLES** : modèle orienté objet, méthode O-O, système expert.

## ABSTRACT

This paper presents a method exploiting O-O model concepts to build an expert system in empirical domains as Management, Human Sciences, Medecine.

This method relies on a particular model whose specific features are : a twin conceptual level (internal and external level), a multiple ascending inner inheritance without conflicts, evaluation and dynamical functions, a zoom effect providing an adapted perception of the expert's domain.

The method steps are described in turn. The method will be implemented by an automated tool (ACSE) to design expert systems.

**KEYWORDS** : Object oriented model, O-O method, expert system.

*Catégorie : Recherche*

## 1. INTRODUCTION

La plupart des méthodes de construction de Systèmes Experts (S.E.) peuvent se répertorier suivant trois niveaux (VAUD 90) relatifs :

### - **Aux entreprises**

Ce sont "les méthodes supports de l'impact des SE dans les organisations"; elles décrivent les différentes étapes en les assortissant de "conseils" et de produits logiciels (BONN 86).

### - **Aux projets**

Ce sont les "méthodes supports du développement de SE"; le découpage en phases permet la visibilité du développement. Ces méthodes s'appuient sur un cycle de vie logiciel et accompagnent les acteurs depuis les spécifications jusqu'à la validation du SE (VOGE 88).

### - **Aux activités**

Ce sont les "méthodes supports de la réalisation d'activités"; elles proposent des enchaînements d'outils d'assistance. Chacun de ces niveaux traite un aspect particulier de la vie du SE.

Nous proposons une méthode dont les étapes successives permettent au cogniticien d'exprimer les connaissances d'experts d'un domaine donné, à l'aide d'un modèle orienté objet original. L'objectif de cette méthode est de couvrir les trois niveaux déclarés ci-dessus avec en final, l'élaboration d'un **Atelier de Conception de SE** nommé **ACSE**. Les objectifs assignés sont voisins de ceux de MACAO (AUSS 89)

Ce document est organisé de la façon suivante : nous rappellerons tout d'abord les concepts du modèle, puis nous présenterons les différentes étapes de la méthode et nous conclurons en proposant une ébauche d'atelier de conception.

## 2. LES CARACTERISTIQUES DU MODELE ORIENTE OBJET

Dans notre modèle (COLL 89)(COLL 90), les objets sont imbriqués les uns dans les autres. Selon les applications, une unité correspondant à un type d'objet de référence (appelé niveau 0) doit être défini : chaque instance de ce type est appelée **objet unique**.

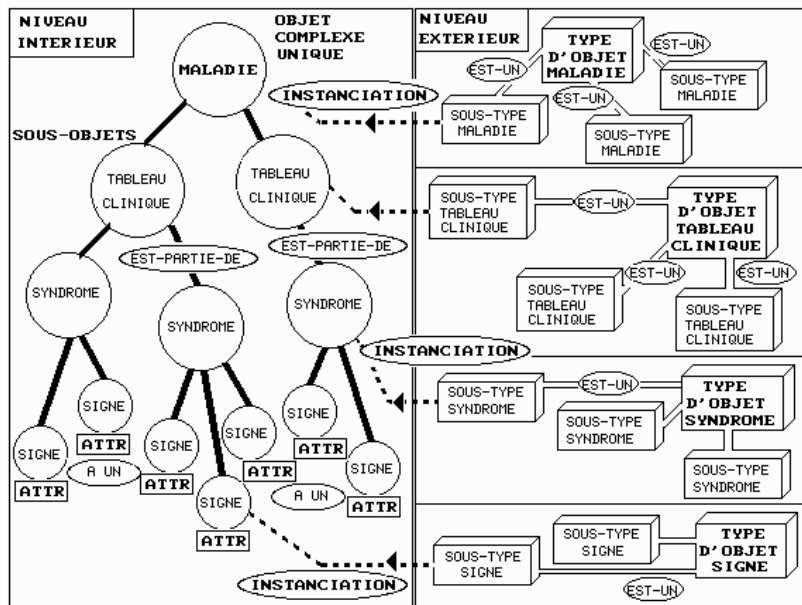
Cette notion définit une frontière entre le niveau **extérieur** et le niveau **intérieur** : tous les objets formant la structure d'un objet unique (instance du type de référence) sont appelés ses sous-objets et appartiennent à son niveau intérieur. Les autres objets forment le niveau extérieur, c'est à dire l'environnement de l'objet unique.

**Au niveau intérieur** : l'ensemble des propriétés des sous-objets (attributs, fonctions...) détermine une partition sur l'ensemble des propriétés de l'objet unique.

**Au niveau extérieur** sont exprimées les relations de l'objet avec les autres objets de son environnement.

Chaque sous-objet peut être considéré à un moment donné comme étant au niveau 0 en sélectionnant son type comme type de référence. Nous appelons **effet zoom**, cette perception adaptée des objets du monde réel.

Le modèle comporte un double niveau conceptuel (figure 1)



**Figure 1 : Le double niveau conceptuel, Exemple : un objet maladie**

## 2.1. LE NIVEAU INTERIEUR OU "ENDOBJET"

Il inclut les relations d'agrégation des sous-objets (EST-PARTIE-DE) et les relations regroupant les propriétés (attributs, fonctions) d'un objet (A-UN).

Les relations d'agrégation et la partition des ensembles de propriétés qui en découle produisent un héritage intérieur multiple ascendant qui transmet les propriétés des sous-objets à l'objet unique qu'ils composent.

Au niveau intérieur, attributs et fonctions représentent les caractéristiques statiques et le comportement des objets. Les contraintes sur les objets sont soit statiques (contraintes de structure, attributs, cardinalités), soit dynamiques (par des fonctions du temps).

L'évolution spontanée de l'état des objets est traduite par des **fonctions dynamiques** qui modifient automatiquement, au cours du temps, les liens de composition (structure) et les domaines ou les valeurs des attributs (propriétés).

Des **fonctions d'évaluation** comparent la structure ou les propriétés de l'objet unique à celles d'autres objets.

## 2.2 LE NIVEAU EXTERIEUR OU "EXOBJET"

Il inclut les relations de généralisation et de spécialisation (EST-UN) existant entre les types d'objet et les sous-types.

Ces relations établissent une hiérarchie de type et un **héritage simple descendant** qui transmet les propriétés d'un type d'objet vers ses sous-types.

Des **fonctions d'évaluation globales** situées dans **des instances** du type d'objet application permettent de comparer les instances des types d'objet de la base de connaissances.

Le modèle distingue le concept de **type** qui définit la structure et les propriétés des instances, du concept de **classe** qui représente un ensemble d'instances.

## 2.3. LIENS ENTRE LE NIVEAU EXTERIEUR ET LE NIVEAU INTERIEUR

### L'instanciation

Les niveaux intérieurs et extérieurs sont reliés par le mécanisme d'instanciation qui crée de nouveaux objets conformément à chaque type.

Les **attributs qualifiants** sont des attributs particuliers (mentionnés dans le type d'objet) et dont la présence est obligatoire dans les nouvelles instances créées. Les autres attributs sont facultatifs.

La communication entre objets a lieu par messages et grâce à des fonctions d'interface. Les messages transportent des données provenant d'un objet émetteur vers un ou plusieurs objets récepteurs.

### **Encapsulation et fonctions d'interface**

Les fonctions d'entrée (ou d'affectation) vérifient les valeurs à donner aux attributs de l'endobjet puis les affectent. Les fonctions de sortie (ou de lecture) fournissent l'état de l'objet après réception d'un message de requête validé.

### **Type d'objet application**

Le **type d'objet application** est un type système prédéfini dont les instances décrivent les applications conformes aux besoins des utilisateurs.

Les objets de type application encapsulent :

- **des données** : les métaattributs représentant les états intermédiaires et les résultats de traitements ou d'étapes de raisonnement.
- **des traitements** : des métafonctions conçues pour définir les relations existantes et les accès nécessaires aux objets du domaine de connaissances, intervenant à chaque étape de raisonnement.

Dans des applications système expert, selon l'effet zoom, les objets applications peuvent sélectionner les types d'objet connaissances de référence pour adapter le niveau de raisonnement et déclencher les fonctions d'évaluation nécessaires.

Ils obtiennent les résultats de comparaisons adaptées entre les objets de la base de connaissances et ceux introduits par l'utilisateur.

Les **principaux points originaux de ce modèle** sont :

- Le double niveau conceptuel : intérieur, extérieur
- L'héritage intérieur multiple ascendant
- Les fonctions d'évaluation et fonctions dynamiques qui représentent respectivement l'état courant et l'évolution des objets. Les fonctions d'évaluation utilisent des algorithmes de recherche de similitude structurale partielle (BOUL 92).
- Les types d'objet application.

*NB : les méthodes sont appelées fonctions dans notre modèle.*

## **3. UNE METHODE DE CONCEPTION DE SYSTEME EXPERT.**

### **3.1. L'ETAPE PREALABLE D'UN PROJET DE SYSTEME EXPERT**

#### **3.1.1. L'ETUDE D'OPPORTUNITE**

L'étude d'opportunité, identifie les raisons justifiant la création d'un système expert.

Elle exploite une classification des problèmes où les SE sont habituellement préconisés, indépendamment des domaines d'applications :

- La déficience qualitative d'une ou plusieurs fonctions d'expertise : diagnostic, reconnaissance d'états, décision, conception, simulation et prédiction, apprentissage.
- La mauvaise performance des prestations en quantité, en coût, en vitesse d'exécution...
- La nécessité d'un transfert de connaissances, par mesure conservatoire (rareté du savoir), par besoin d'autonomie scientifique, pour des raisons stratégiques ou de

protection industrielle (secrets de fabrication), pour assurer la pérennité d'un savoir spécialisé.

-La volonté de formaliser un raisonnement empirique dans un but de recherche.

### 3.1.2. L'ETUDE DE FAISABILITE

Il s'agit d'une étude de faisabilité classique qui évalue :

- l'intérêt du projet de construction de SE,
- les moyens humains disponibles (expert(s), cogniticien, informaticien(s)),
- les moyens matériels (machines et outils logiciels),
- la durée nécessaire à la réalisation du SE (ordonnancement),
- la rapidité d'évolution du domaine et des techniques (risque d'obsolescence)
- les risques d'échecs

A l'issue des études d'opportunité et de faisabilité, une décision est prise quant à la poursuite du projet ou à son arrêt.

## 3.2. L'ETAPE CONCEPTUELLE DE DESCRIPTION DU DOMAINE

Indépendamment des applications envisagées, cette étape décrit le ou les domaines de connaissances (les contextes où s'exerce l'expertise)(BREU 89) et l'exprime en termes de types d'objets.

### 3.2.1. LE RECUEIL DE CONNAISSANCES

Il s'agit d'une étape primordiale dans la réalisation d'un SE, menée par le cogniticien et les experts.

L'acquisition de connaissances comprend : une phase de recueil systématique, généralement basée sur des techniques d'expression orale, une phase de conceptualisation initiale conduisant à identifier les concepts, propriétés, relations, procédures, une phase d'analyse plus approfondie à l'aide d'outils structurants permettant une validation de la connaissance (JOHN 88) (LIOU 92).

Les connaissances obtenues peuvent être classées selon leur niveau d'abstraction, leur degré de dépendance vis à vis du contexte, les formes qu'elles peuvent prendre pour l'utilisateur final, la dimension des unités d'expertise analysées. Ces différents niveaux d'abstraction seront exploités à l'aide de l'effet zoom.

Le cogniticien utilise :

-**Les méthodes directes** de recueil de connaissances, basées sur des entretiens plus ou moins structurés, correspondent généralement à la phase de recueil systématique. L'expert verbalise ses connaissances et répond éventuellement à un questionnaire préparé par le cogniticien. Cette phase produit une liste d'objets pourvus de propriétés remarquables et des relations existant entre eux. Le cogniticien utilisera avec profit une série de fiches identifiant les objets (les synonymes), recensant les attributs et les actions associées.

-**Les méthodes indirectes** complètent les résultats des méthodes directes.

La technique des grilles répertoires basée sur les travaux de Kelly (KELL 55) permet d'effectuer une classification des éléments de connaissances et des inférences entre eux. Dans une première phase de recueil de données du domaine d'expertise, l'expert est amené à comparer une vingtaine d'objets entre eux et à dégager les

propriétés qui les rapprochent et celles qui les distinguent. Il est invité à établir une échelle de valeurs pour chaque propriété recensée.

Durant la deuxième phase de formalisation, le cogniticien établit un tableau des distances entre les objets et entre les attributs. Cette matrice de distance est construite en sommant les écarts des évaluations de toutes les caractéristiques. On obtient ainsi, un arbre de "ressemblance" entre objets (REIT 87) (GALL 88) (BOOS 88) (BOOS 89). Cette méthode semble bien adaptée à la modélisation objet.

### 3.2.2. DEFINIR LE OU LES DOMAINES DE CONNAISSANCES

Définir les limites du ou des domaines d'expertise par un ou plusieurs intervalles de types d'objets de référence dont les instances interviennent au sein des hiérarchies de composition d'objets qui constituent autant de domaines distincts (effet zoom) (COLL 89). Les échanges entre les objets de ces domaines ont lieu exclusivement par messages. Plusieurs applications peuvent coexister sur un ou plusieurs domaines.

### 3.2.3. RECENSER LES TYPES D'OBJETS INTERVENANT DANS CHAQUE DOMAINE

Durant le classement des objets des domaines d'expertise, il faut éviter : de confondre objet et attribut, de définir une action comme un objet, le concepteur ne pouvant expliquer ce qu'est l'objet mais uniquement ce qu'il fait. L'objet se résume alors à un traitement.

### 3.2.4. CLASSER LES OBJETS EN DEUX CATEGORIES :

**Les Types d'objets "concrets" ou physiques** décrivent des objets existant dans le monde réel, perceptibles à l'aide d'au moins un des cinq sens humains ou tout appareillage prolongeant leur activité.

**Les Types d'objets "abstraites"** ou concepts générés par l'activité cognitive humaine : l'imagination, la réflexion, le raisonnement sont des abstractions, des agrégats issus du filtrage pragmatique des propriétés pertinentes remarquées dans le monde réel.

Les systèmes experts exploitent plutôt des concepts précaires émergents de la recherche, dont l'évolution rapide est étroitement liée aux changements perçus dans l'environnement et les techniques.

Les types d'objets abstraits sont à rapprocher des connaissances superficielles des modèles à règles de production (HARM 90). On cherchera à s'appuyer sur la plus grande stabilité des objets physiques.

**Règle** : On ne mélange pas des objets concrets et des objets abstraits dans une hiérarchie de composition. On recense les types d'objets de chaque domaine selon la place de leurs instances dans la hiérarchie de composition et leur caractère concret ou abstrait.

### 3.2.5. DEFINIR LA STRUCTURE DES TYPES D'OBJETS DU DOMAINE

#### a. La démarche descendante :

Pour chaque type d'objet complexe du domaine, selon la précision désirée par les experts : **1.**déterminer le nombre de niveaux de sous-objets nécessaires pour bâtir la hiérarchie intérieure de leurs instances, **2.**Sélectionner des instances significatives et énumérer les sous-objets parties de ces instances, recommencer jusqu'au niveau feuille où figurent les sous-objets simples ne comportant que des attributs, **3.**Définir les types d'objet décrivant les sous-objets intervenant à chaque niveau de la hiérarchie de composition.

A l'inverse des modèles à règles de production, en cas de désaccord des experts on accumule les avis de chacun et on crée plusieurs versions de types d'objet qui coexistent sans conflit, du fait de l'indépendance et de la protection des objets produites par l'encapsulation. Les variantes nécessitent de dériver des sous-types. On recense les attributs de chaque type d'objet, avec leurs contraintes de domaine, de cardinalité et d'intégrité.

On représente la structure des instances des types d'objets de chaque domaine par un diagramme, soumis aux experts pour approbation et correction.

### **b. La démarche ascendante**

On constitue une bibliothèque de types d'objet simples (ROCH 91) dont on recense les attributs qualifiants et leurs contraintes. Puis on exploite **la composition des types d'objets** pour construire, à l'aide des types d'objets simples, les types d'objets complexes fixés comme buts en respectant la règle ci-dessus. Par une approche **incrémentale** d'enrichissements successifs, les nouveaux types d'objets, obtenus par composition, grossissent la bibliothèque et servent à construire d'autres types d'objets plus complexes. On peut à tout moment "rajouter" des types d'objets oubliés. L'approche ascendante présente l'avantage d'utiliser des types d'objets simples, dont on peut tester la qualité, avant de les impliquer dans des types d'objets plus complexes.

Les démarches ascendante et descendante sont combinées, d'une manière très souple au gré des concepteurs du fait de la modularité et des facultés de composition/décomposition offertes par le modèle.

Il faut vérifier que les types d'objets créés sont conformes aux spécificités préconisées par les experts, et éventuellement dériver des sous-types. Un **dictionnaire de référence** recense les symboles identifiant les types d'objets de chaque domaine, établissant ainsi un vocabulaire commun.

#### **3.2.6. DEFINITION DU COMPORTEMENT DES INSTANCES DES TYPES D'OBJETS RECENSES**

Dans cette étape est défini le comportement spontané des instances des types de chaque domaine : modifications au cours du temps de la structure globale des objets ou de certaines parties seulement (sous-objets), changement des contraintes de leurs attributs. Pour chaque objet subissant une modification de structure, on définit une fonction dynamique et on détermine à chaque instant, les sous-objets intervenant. Les modifications de contraintes des attributs sont exprimées à l'aide de fonctions du temps. L'établissement de ces fonctions nécessite une étude (domaine de définition, sens de variation...) dont les résultats sont transcrits dans des descripteurs au sein des types d'objets.

#### **3.2.7. DEFINITION DES INTERFACES D'ENTREE / SORTIE POUR LES TYPES D'OBJETS RECENSES**

Les experts et le cogniticien définissent les fonctions d'interface nécessaires à la communication entre les objets du domaine.

#### **3.2.8. LES FONCTIONS D'EVALUATION INTERIEURES**

Le cogniticien et les experts construisent des fonctions d'évaluation pour obtenir l'état de certains objets.

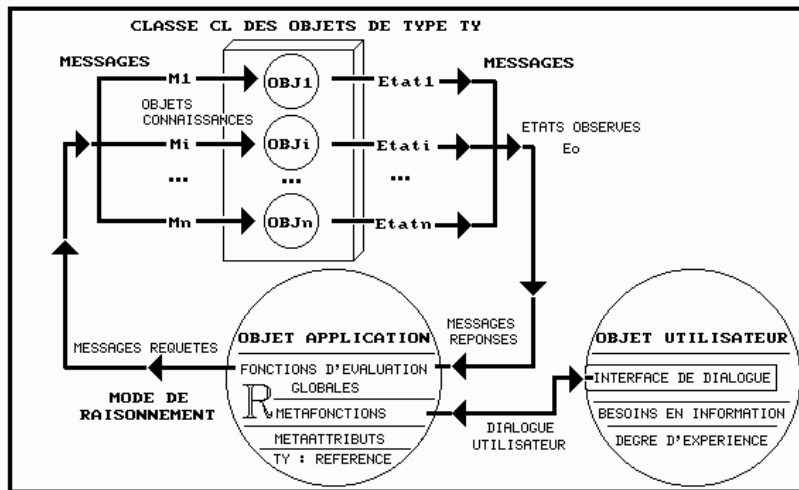
Les fonctions d'évaluation globales résident dans les objets applications et utilisent les fonctions d'entrée/sortie, pour connaître la valeur de certains attributs des objets connaissances. Cette approche améliore la modularité puisque les types d'objets

représentant la connaissance et leurs instances demeurent indépendants des applications bâties.

### 3.3. ETAPE DESCRIPTIVE DES APPLICATIONS

Cette étape établit les relations existant entre les types d'objets du fait, de la ou des applications envisagées. La simulation des modes de raisonnement des experts s'exprime par des actions que devront exécuter les objets.

#### 3.3.1. DEFINITION CONCEPTUELLE DES MODES DE RAISONNEMENT (fig . 2)



**Figure 2 : Le cycle de raisonnement**

Une phase de raisonnement (figure 2) peut se définir comme un ensemble de messages  $\{M1, \dots, Mi, \dots, Mn\}$  envoyés par l'objet application selon un mode de raisonnement noté R, sur une classe d'objets connaissances CL de type TY.

L'ensemble d'états des objets  $\{Et1, \dots, Eti, \dots, Etn\}$ , conformément au problème posé est interprété à l'aide de fonctions d'évaluation adéquates, exprimant un résultat.

L'héritage multiple ascendant permet de raisonner au niveau des sous-objets tout en produisant des états **observables** au niveau de l'objet unique (en modifiant le type d'objet de référence). Un objet application est créé et détermine, de manière interactive, l'ordre de succession des phases de raisonnement.

On détermine les types d'objets nécessaires au raisonnement R et les flux d'informations échangés entre eux.

Les états des objets notés (Eo) peuvent se traduire sous des formes diverses : par des valeurs booléennes, des scores établissant le degré de similitude structurale d'un objet comparé à ceux de la classe CL... La forme de la réponse dépend de la nature de la fonction d'évaluation choisie dans le cadre de l'application.

Les modes de raisonnement proposés :

**La similitude structurale** évalue le degré de parenté structurale de deux objets. Elle peut être partielle : concerner un ou plusieurs sous-objets d'un objet complexe. Les fonctions d'évaluation exploitent des algorithmes de recherche de similitude structurale partielle entre deux objets (BOUL 92). Nous proposons en annexe un algorithme de ce type.

**La similitude d'états** compare l'état de deux objets ayant une parenté structurale établie : la valeur de leurs attributs ou de leurs fonctions intérieures.

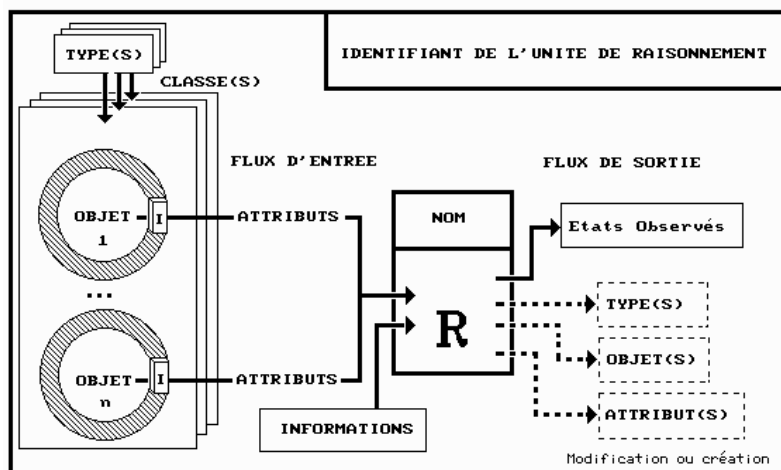
**L'induction** cherche à généraliser de nouvelles propriétés observées sur quelques objets.

La **déduction** transforme une ou plusieurs connaissances en une nouvelle, plus appropriée à la résolution d'un problème donné.

### 3.3.2. LES AUTRES TRAITEMENTS

D'autres fonctions de traitement peuvent être implantées dans les objets applications, comme les tris, les calculs de maximums et de minimums, les comparaisons de valeurs... Leur but est d'exprimer les états des objets sous une forme adaptée au problème de l'utilisateur.

### 3.3.3. LE TRAVAIL DES EXPERTS DU DOMAINE



**Figure 3 : Schéma cognitif : une unité de raisonnement**

Le cogniticien fait exprimer aux experts les phases de raisonnement nécessaires à l'expertise. Chaque unité de raisonnement R (figure 3) est représentée par un identifiant, les types d'objets concernés, le mode de raisonnement employé, la nature des informations nécessaires (flux d'entrée), la forme d'expression des états des objets (flux de sortie), d'éventuels nouveaux types d'objets, instances, ou attributs induction). Le **schéma cognitif** décrit les unités de raisonnement mises en oeuvre par les experts.

### 3.4. LA REPRESENTATION DES CONNAISSANCES

A partir de quelques cas typiques, le cogniticien, avec l'aide des experts essaie d'instancier les **types d'objets définis précédemment**. Il tient compte des propriétés des instances, des contraintes des attributs et des fonctions... exprimées par les types d'objets définis lors des étapes précédentes. Si des divergences sont constatées, de nouveaux types d'objet répondant à ces contraintes spécifiques peuvent être établis. Cette dernière étape conceptuelle procure toutes les informations nécessaires pour construire un prototype de SE et organiser sa communication avec d'autres catégories d'applications. Il s'agit d'une étape itérative :

- Durant l'étape de construction** des types d'objet et des objets, le cogniticien exploite les approches ascendante et descendante décrites plus haut.
- Durant l'étape de validation**, les experts contrôlent la conformité des objets.

### 3.5. LA REALISATION DU SYSTEME EXPERT.

Cette étape bénéficie de la souplesse et des facultés de prototypage procurées par l'approche orientée objet; elle s'appuie sur les définitions des types d'objets représentant les connaissances et les modes de raisonnement.

La réalisation comprend l'élaboration d'un prototype, son contrôle et sa validation, la montée en charge de la base de connaissances et l'achèvement du système expert.

**La réalisation d'un prototype** vise à valider les schémas cognitifs sur une base de connaissances restreinte.

**L'aspect structural de la connaissance** est traité comme suit : sélectionner quelques cas pratiques significatifs, créer les types d'objets recensés pour le domaine d'expertise concerné, établir les liens formant la hiérarchie extérieure entre les différents types d'objets. Il faut ensuite créer les objets simples correspondants, implanter les attributs de chacun d'eux, définir les liens formant la hiérarchie intérieure des objets complexes.

**L'aspect comportemental** est ensuite envisagé. L'informaticien établit le comportement spontané des objets statiques qui viennent d'être construits. Les types d'objets sont enrichis de la liste des fonctions chargées de définir le comportement de leurs instances.

**Les applications sur la base de connaissances** sont réalisées en exploitant les schémas cognitifs et en construisant un ou plusieurs objets abstraits de type application exprimant les besoins des utilisateurs du système expert. Les objets application comportent des métafonctions de gestion et des métafonctions de raisonnement propres aux systèmes experts.

**L'acheminement des résultats** a lieu par messages vers l'objet application. Ils sont stockés dans les métaattributs en vue de leur lecture par l'utilisateur.

A l'issue de cette phase, on dispose d'un ou plusieurs objets application constituant, avec les objets connaissances, un prototype fonctionnel de système expert.

### 3.6. L'ETAPE D'EVALUATION ET DE VALIDATION

#### 3.6.1. LE CONTROLE ET LA VALIDATION DU PROTOTYPE

Il comprend : le repérage et la correction des erreurs techniques de conception et l'évaluation du comportement et des résultats du prototype face à des cas pratiques ou à des situations réelles qu'on lui soumet.

#### 3.6.2. LES ERREURS TECHNIQUES DE CONCEPTION

Elles concernent les redondances, les contradictions. Elles sont grandement limitées par l'héritage extérieur sur les types d'objet et l'héritage intérieur sur les objets.

On remédie à ces erreurs techniques en effectuant la **journalisation** des messages d'erreurs. La correction des erreurs statiques est effectuée à l'aide d'outils d'édition des types d'objets et des objets.

Le repérage des erreurs comportementales ou dynamiques est réalisée en simulant l'évolution des objets (les valeurs successives prises par les différents attributs) et grâce à des outils d'étude de fonctions.

#### 3.6.3. L'EVALUATION DU PROTOTYPE DE SYSTEME EXPERT

Le prototype, doit être confronté aux objectifs mentionnés dans le cahier des charges, établi à l'issue de l'étape de faisabilité, qui délimite les buts à atteindre.

Il faut vérifier la conformité du prototype aux spécifications et la variation de performance du personnel utilisateur en situation d'expertise.

\* **La conformité** : Les connaissances et les étapes de raisonnement doivent être conformes aux attentes des experts ayant participé à l'élaboration du prototype. Les

schémas cognitifs sont réexaminés et au besoin modifiés. Les connaissances produites par les experts sont considérées comme exactes et font donc référence.

\* **La variation de performance** s'étudie entre le couple utilisateur/prototype et utilisateur seul (BONN 86).

\* **Le protocole d'évaluation :**

Il est effectué un rappel des objectifs définis au sein du cahier des charges, les experts sélectionnent des cas pratiques validés parmi ceux implantés dans le prototype, les experts évaluent la fidélité avec laquelle le prototype restitue leurs connaissances. Si le taux d'échecs est important, l'évaluation est suspendue, un cycle de correction est amorcé.

Les réponses du prototype sont réparties par les experts dans les classes suivantes :

A: considérée comme la meilleure solution	+ 3 points
B: une des meilleures solutions, ne fait pas le consensus.	+ 2 points
C: solution n'est pas des plus intéressantes.	+ 1 point
D: solution inadéquate, ne répond pas au problème posé.	0 point
E: La solution est mauvaise ou fausse.	- 10 point
F: La solution est dangereuse, néfaste...	- 100 point

On calcule la différence des scores obtenus lorsque les utilisateurs sont seuls et lorsqu'ils sont assistés du prototype. Le prototype est considéré comme acceptable si les scores des couples utilisateur/prototype sont positifs.

Enfin, on teste le prototype auprès d'experts du domaine n'ayant pas participé à la réalisation du SE.

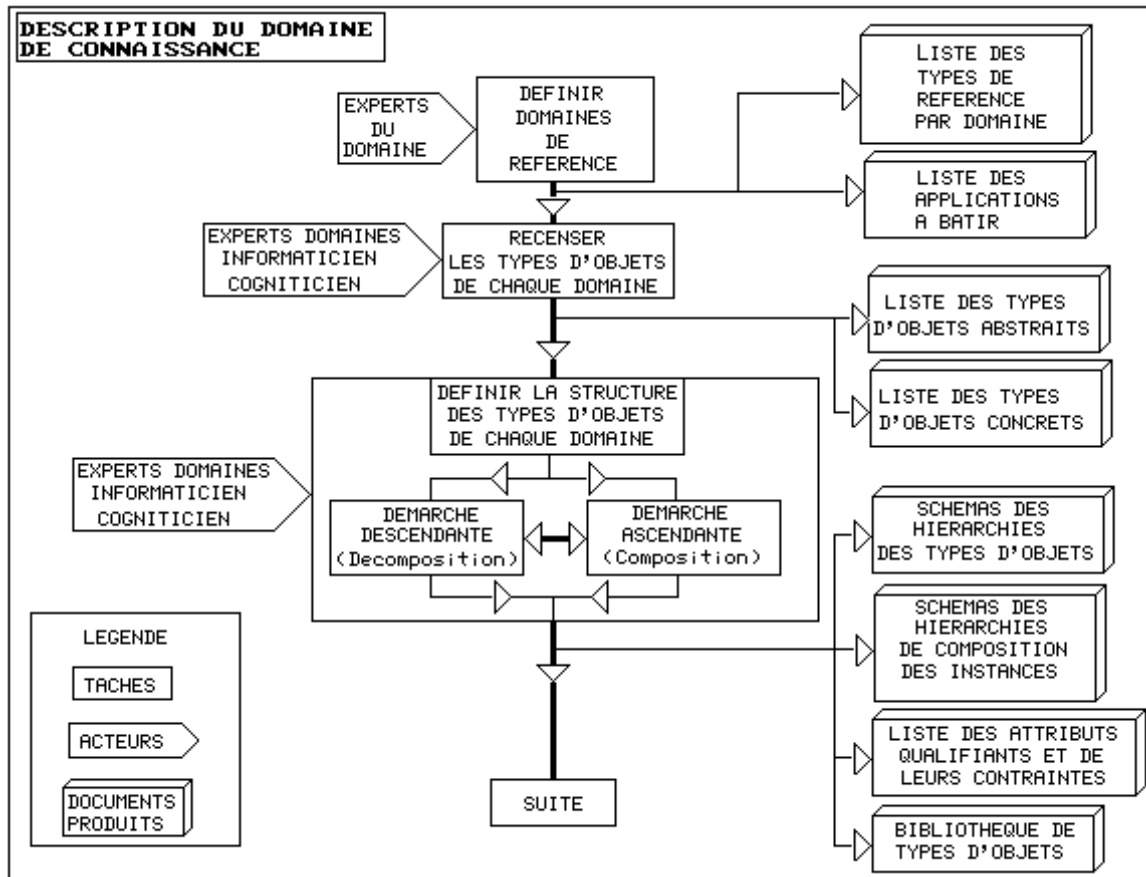
#### 3.6.4. LA MONTEE EN CHARGE ET LA MAINTENANCE

Le prototype ayant été jugé satisfaisant, il convient de compléter les classes d'objets constituant la base de connaissances. En principe, il suffit d'instancier les types d'objets existant mais certains cas particuliers peuvent encore nécessiter la spécialisation de quelques nouveaux sous-types.

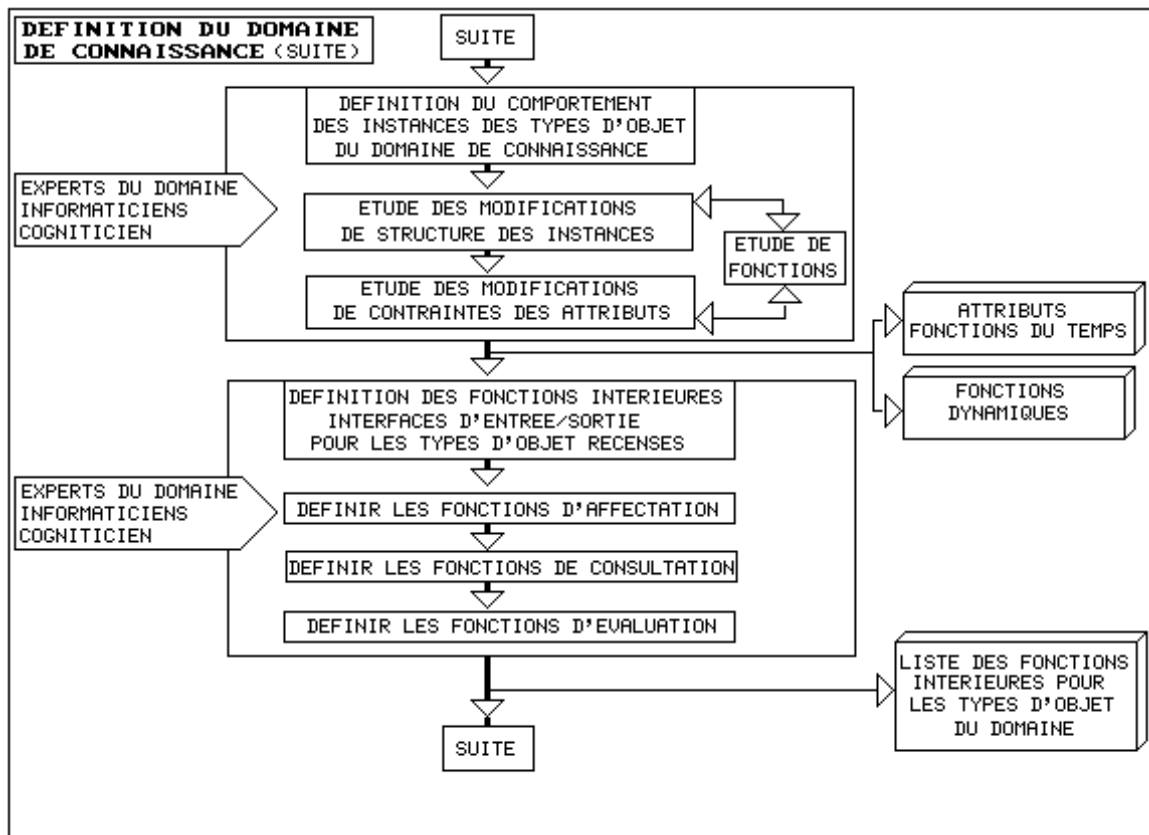
\* **La maintenance**, dans le cas particulier d'un système expert, comprend la correction des erreurs techniques de conception et celles qui correspondent à une mauvaise adaptation du système aux situations réelles.

Le système expert devra donc être périodiquement réévalué et le contenu de la base de connaissances mis à jour.

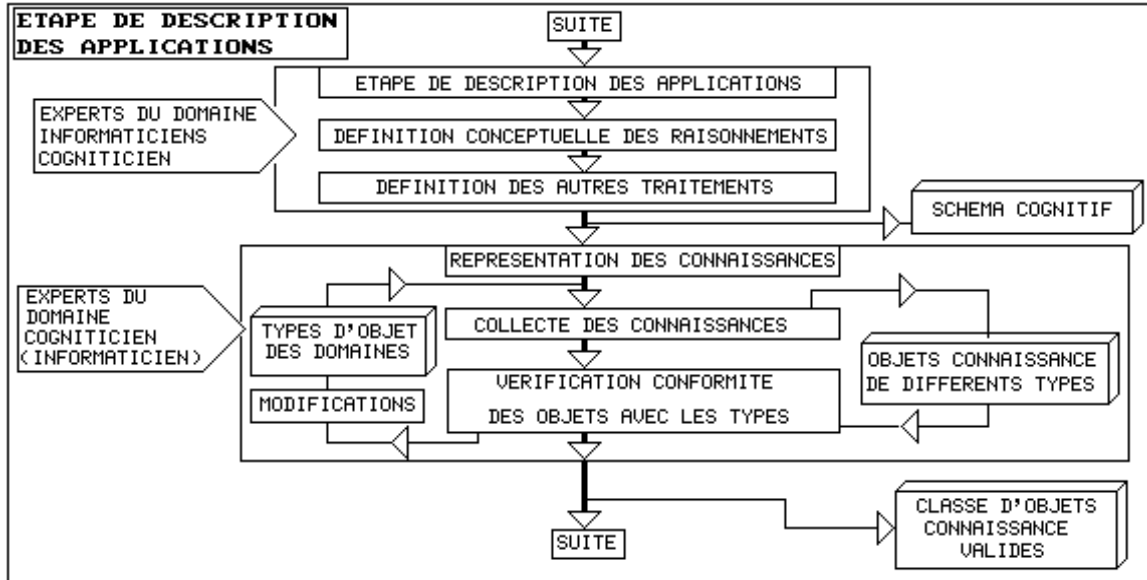
Les étapes de la méthode sont illustrées par les figures 4, 4', 5 et 6.



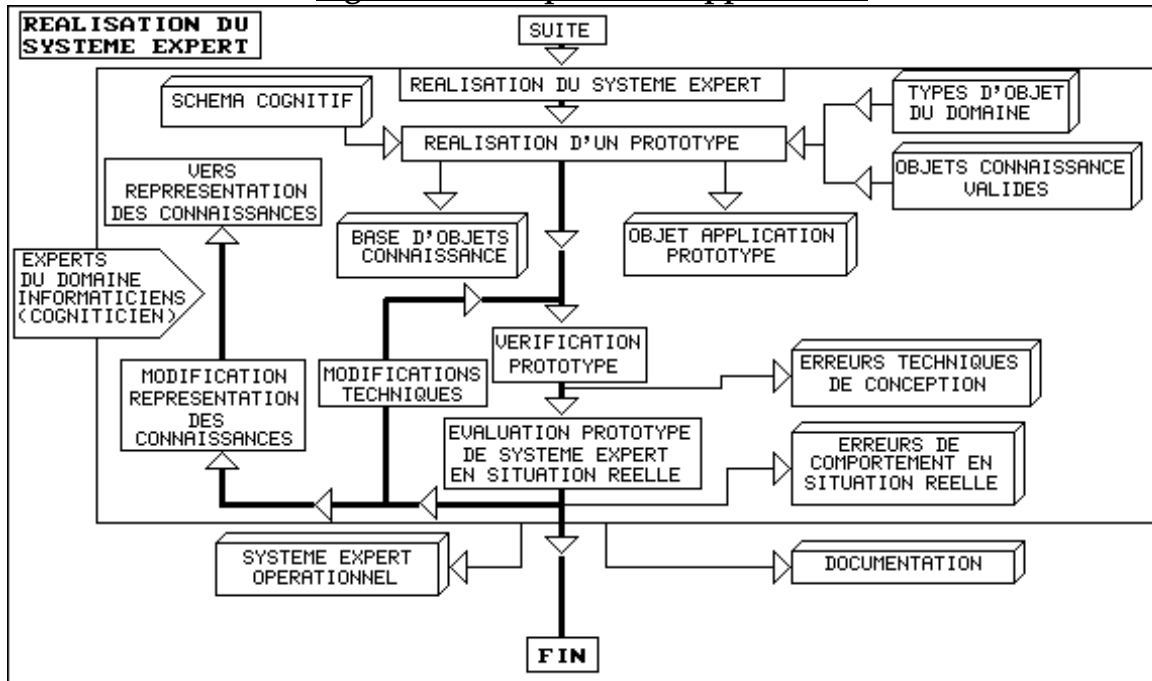
**Figure 4 : Domaine de connaissance**



**Figure 4' : Domaine de connaissance (suite)**



**Figure 5: Description des applications**



**Figure 6: Réalisation du S.E.**

**4. CONCLUSION : EBAUCHE D'UN ATELIER DE CONCEPTION DE SE**

Cet atelier nommé ACSE est en cours de réalisation. Il prend en compte les étapes de la méthode et génère des SE qui sont composés de trois bases communiquant par messages :

- Une Base d'Objets Connaissances qui modélise les connaissances du domaine d'expertise,
- Une Base d'Objets Applications qui représente les modes de raisonnement de l'expert (exprimés par les schémas cognitifs) utilisés lors de l'expertise,
- Une Base d'Objets Utilisateurs qui exprime les besoins en information des utilisateurs, notamment leur degré d'expérience du domaine et d'utilisation du système (GAUT 89). Les bases BOC, BOA, BOU exploitent des ressources situées dans le module système :

-Le serveur dynamique d'objets permet un prototypage et une modification interactive des objets.

Ce module a été réalisé en C++ sur PC et est porté actuellement sur station SUN IPX (R). Le SGBD orienté objet ONTOS (R) reste à confirmer pour la gestion des bases.

-Le gestionnaire de messages assure l'acheminement des messages vers les objets destinataires.

-L'interpréteur de fonctions traduit et exécute les fonctions présentes dans les objets.

-Le module de mise à jour des fonctions temporelle gère la dynamique des objets.

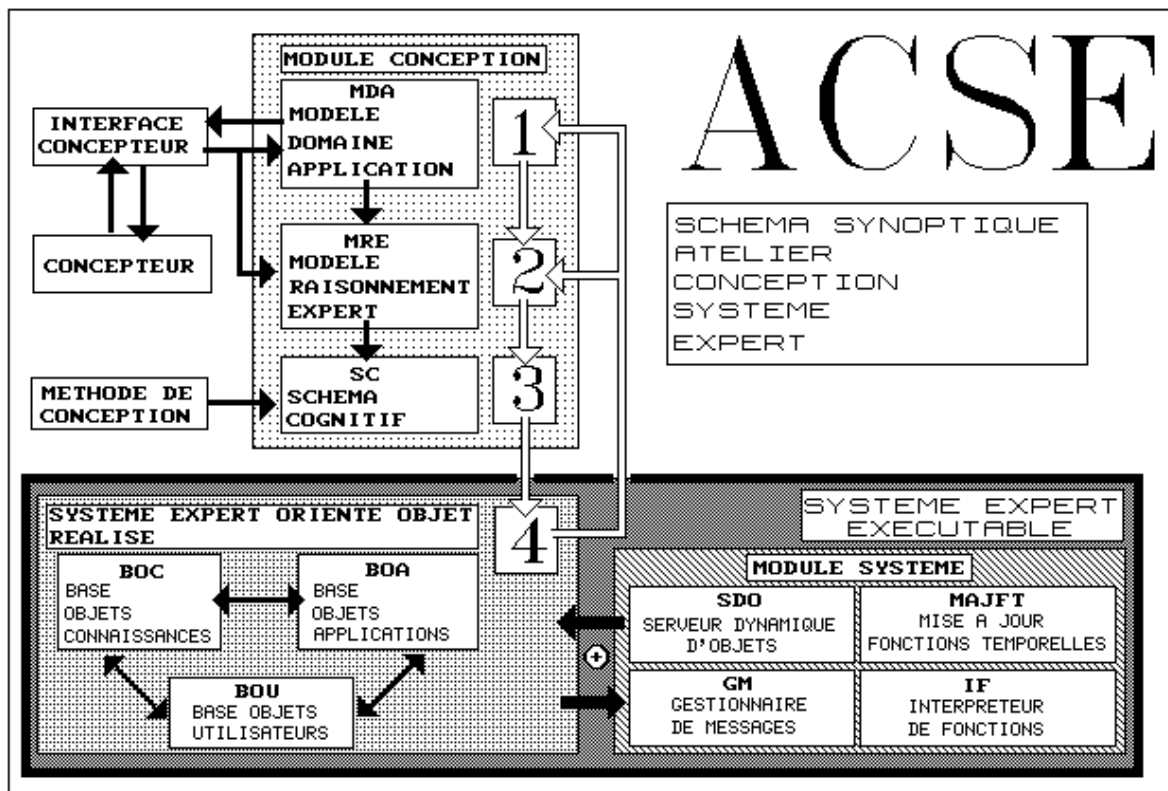


Figure 7 : Schéma synoptique de ACSE

## REFERENCES

- (AUSS 89) AUSSENAC N., FRONTIN J., RIVIERE M.H., SOUBIE J.L.  
 "A Mediating Representation to Assist Knowledge Acquisition with MACAO."

In: 3rd European Workshop on Knowledge Acquisition for Knowledge-Based Systems, ed. by J. BOOSE, B. GAINES, J.G. GANASCIA, Paris, 1989.

**(BONN 86) BONNET A., HATON J.P., TRUONG-NGOC J.M.**

"Systèmes experts, vers la maîtrise industrielle"

InterEditions 1986.

**(BOOS 88) BOOSE J.H.**

"Uses of Repertory Grid - Centred Knowledge Acquisition Tools for Knowledge-Based Systems."

International Journal of Man-Machine Studies, Vol 29, n°3, pp. 287-310.

**(BOOS 89) BOOSE J.H.**

"Interactive Knowledge Acquisition Tools"

In: 3rd European Workshop on Knowledge Acquisition for Knowledge-Based Systems, ed. by J. BOOSE, B. GAINES, J.G. GANASCIA, Paris, 1989, pp. 437-447.

**(BOUL 88) BOULANGER D., COLLOC J.**

"Conception d'une base d'objets maladies : exploitation par un Système Expert d'aide au diagnostic"

In : "La science des systèmes dans le domaine de la santé", tome1 : L'informatique dans les systèmes de santé.". Ed. Duru G et al. - Lyon : Masson, 1988, pp. 345-348.

**(BOUL 92) BOULANGER D., COLLOC J.**

"Detecting Heterogeneity in a Multidatabase Environment Through an O-O Model"

IFIP, DS-5 Conference, Semantics of Interoperable Database Systems, Lorne, Victoria, Australia, November 1992.

**(BREU 89) BREUKER A., WIELINGA B.J.**

"Knowledge Acquisitions as Modelling Expertise : the KADS Methodology"

Proceedings First European Workshop on Knowledge Acquisition for Knowledge-Based Systems, 1989.

**(COLL 89) COLLOC J., BOULANGER D.**

"Un modèle objet pour la représentation de connaissances empiriques"

In: "Colloque ICO'89 Informatique Cognitive des Organisations. Quebec, Canada, Juin 1989."

Ed. by Moulin B. et Simian G. - Limonest : L'interdisciplinaire, 1989, pp. 119-140.

**(COLL 90) COLLOC J.**

"Une approche orientée objet pour l'élaboration d'applications médicales."

Thèse de Sciences : Informatique et Automatique Appliquées : Institut National des Sciences Appliquées de Lyon, 1990, °0054, 391 p.

**(FERB 90) FERBER J.**

"Conception et programmation par objets",

Paris, Ed. Hermès, 1990, 110 p.

**(GALL 88) GALLOÛIN J.F.**

"Transfert de connaissances. Systèmes experts : techniques et méthodes",

Paris, Ed. Eyrolles, 1988, 168 p.

**(GAUT 89) GAUTHIER G., IMBEAU G.**

"Un système tutoriel intelligent : modèle théorique et réalisation."

Actes du 6ème Symposium Canadien sur la technologie pédagogique, Halifax, N.E. : 1989, CNRC, pp. 109-114.

**(HARM 90) HARMON P., SAWYER B.**

"Creating Expert-Systems for Business Industry".

New-York : Ed. John Wiley & Sons Inc. : 1990, 329 pages. ISBN 0-471-61496-3.

**(JOHN 88) JOHNSON N., TOMLINSON C.M.,**

"Knowledge elicitation for second generation expert systems",

Proceedings of the 2nd EKAW 88, GMD Studien Nr 143, Bonn, 1988, pp. 23.1-23.10.

**(KELL 55) KELLY G.A.**

"The Psychology of Personal Constructs", New-York : Norton.

**(LIOU 92) LIOU Y.T.**

"Knowledge Acquisition Issues, Techniques and Methodology."

DATABASE, winter 92, pp. 59-64.

**(REIT 87) REITMAN-OLSON J., REUTER H.**

"Extracting expertise from experts : methods for knowledge acquisition, Expert systems (4), n°3, 1987, pp. 157-168.

**(ROCH 91) ROCHFELD A.**

"Modèle externe de données et modèle externe objet"

Actes du 4e congrès génie logiciel, Toulouse, Decembre 1991.

**(VAUD 90) VAUDET J.P., GUYOT J., DELMAS D.**

"Des méthodes pour les systèmes experts"

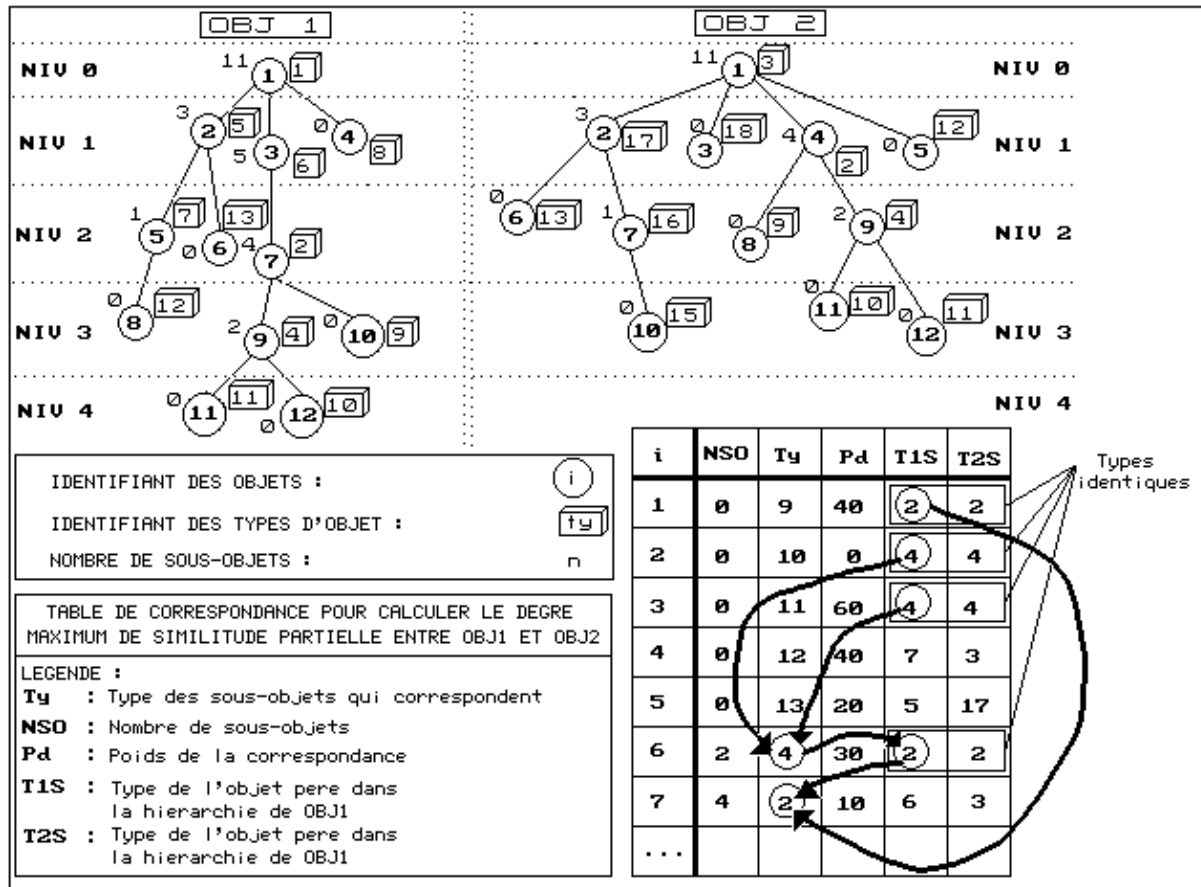
Revue Génie Logiciel n°19, mai 1990, pp. 4-8.

**(VOGE 88) VOGEL C.**

"Génie cognitif"

Masson Collection Sciences cognitives, 1988.

## ANNEXE



**Figure : Comparaison des hiérarchies intérieures de deux objets**

Nous présentons l'algorithme de similarité structurale partielle dans le cas où les deux structures OBJ1 et OBJ2 appartiennent à deux types différents.

L'algorithme compare les hiérarchies de composition intérieures des deux objets OBJ1 et OBJ2 et tente de trouver le plus de sous-objets possible ayant le même type.

- Les arbres sont parcourus en profondeur d'abord car :
- les hiérarchies ont généralement des profondeurs différentes.
  - il faut comparer les sous-objets de même complexité.

**Etape 1 :**

Les arbres de composition des objets OBJ1 et OBJ2 doivent être comparés en tenant compte du nombre de sous-objets que chaque sous-objet comporte dans sa propre hiérarchie de composition.

**Etape 2 :**

Ainsi, l'algorithme construit deux listes de noeuds, (une pour OBJ1 et une pour OBJ2) triées en ordre croissant selon le nombre respectif de sous-objets (NSO) que chaque sous-objet contient dans sa propre hiérarchie de composition.

**Etape 3 :**

Puis, l'algorithme compare les deux listes, il compte les sous-objets appartenant au même type et construit une table de correspondance (voir figure) pour calculer le degré maximum de similitude structurale.

**Etape 4 :**

La table contient les correspondances de sous-objets, triées selon leur complexité, c'est à dire le nombre de sous-objets (NSO) situés dans leur hiérarchie intérieure.

Le poids des correspondances est calculée par la fonction maximum.

L'algorithme compare les types respectifs T1S et T2S des objets pères des deux sous-objets correspondants.

S'ils appartiennent au même type, c'est à dire T1S et T2S ont le même identifiant, une nouvelle correspondance de sous-objets de ce type est recherchée dans la table en examinant la colonne Ty.

Si elle est trouvée, le même processus se poursuit jusqu'à ce que T1S et T2S soient différents. La dissimilitude implique l'arrêt du processus.

Par exemple, les sous-objets de la correspondance n°2 ont le même type d'objet père, T1S=4 et T2S=4. L'algorithme recherche le type n°4 dans la colonne Ty.

Les sous-objets intervenant dans la correspondance n°6 sont de type n°4 et ils montrent à leur tour un même type d'objet père : T1S=2, T2S=2. Le type n°2 est recherché dans la colonne Ty. La correspondance n°7 concerne des sous-objets de type n°2 mais ils n'ont pas même type d'objet père : T1S=6 et T2S=3. Donc, l'algorithme a trouvé deux sous-arbres similaires et a déterminé leur degré maximum de similitude structurale.

L'algorithme se répète jusqu'à épuisement des correspondances inscrites dans la table à la recherche d'éventuels autres sous-arbres similaires.