

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION
EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété
Intellectuelle
Bureau international



(43) Date de la publication internationale
8 février 2007 (08.02.2007)

PCT

(10) Numéro de publication internationale
WO 2007/014986 A2

(51) Classification internationale des brevets : Non classée

(21) Numéro de la demande internationale :

PCT/FR2006/001505

(22) Date de dépôt international : 28 juin 2006 (28.06.2006)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :

0508204 1 août 2005 (01.08.2005) FR

(71) Déposant (pour tous les États désignés sauf US) :

FRANCE TELECOM [FR/FR]; 6, place d'Alleray,
F-75015 Paris (FR).

(72) Inventeurs; et

(75) Inventeurs/Déposants (pour US seulement) : PAULUS,
François [FR/FR]; 14 rue de la Santé, F-35000 Rennes
(FR). BIDAULT, Alain [FR/FR]; 46 Boulevard Alexis
Carrel, F-35700 Rennes (FR).

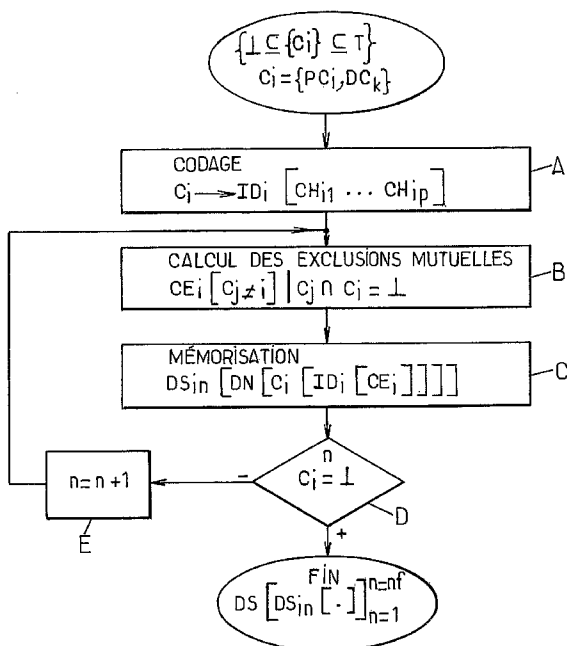
(74) Mandataires : FRECHEDE, Michel etc.; Cabinet
Plasseraud, 52 rue de la Victoire, F-75440 Paris Cedex 09
(FR).

(81) États désignés (sauf indication contraire, pour tout titre de
protection nationale disponible) : AE, AG, AL, AM, AT,
AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO,
CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB,
GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE,
KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU,
LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG,

[Suite sur la page suivante]

(54) Title: METHOD AND DEVICE FOR ENCODING AND SYNTHETICALLY REPRESENTING A PARTIALLY SATURATED ONTOLOGY, A DATA STRUCTURE AND A CORRESPONDING DATA STRUCTURE SERVER

(54) Titre : PROCÉDÉ ET SYSTÈME DE CODAGE ET DE REPRÉSENTATION SYNTHÉTIQUE D'UNE ONTOLOGIE PARTIELLEMENT SATURÉE, STRUCTURE DE DONNÉES, ET SERVEUR DE STRUCTURE DE DONNÉES CORRESPONDANTS



A... ENCODING
B... MUTUAL CONCLUSION CALCULATION
C... STORING

(57) Abstract: The invention relates to a method for encoding and synthetically representing a partially saturated ontology comprising at least a concept hierarchy formed by original concepts and at least one defined concept represented by a lattice and information items not linked to the concepts. The inventive method consists at least in encoding (A) the concept lattice by assigning an identifier (ID_i) formed by at least one integer number sequence to each concept (C_i) and to a lattice node associated thereto, wherein each integer number sequence defines an access path between the considered concept (C_i) and an universal concept (T) by means of successive head concepts, the identification comprises the all integer number sequences each of which defines the access path between the considered concept (C_i) and the universal concept (T) to which an integer number representing the considered concept (C_i) is added, in calculating (B) mutual exclusions between the concepts for each considered concept, in storing (C) the concept lattices encoded in the form of a data structure subdivided into blocs enclosing at least one concept list containing a domain name and a list containing at least the considered concept, wherein each considered concept encloses at least one identifier formed by a list of at least one path and a list of mutually exclusive concepts with the considered concept. Said invention can be used for transmitting and diffusing the ontologies through an IP network in the Internet.

[Suite sur la page suivante]



NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **États désignés** (sauf indication contraire, pour tout titre de protection régionale disponible) : ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), eurasién (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Déclaration en vertu de la règle 4.17 :

— relative à la qualité d'inventeur (règle 4.17.iv))

Publiée :

— sans rapport de recherche internationale, sera republiée dès réception de ce rapport

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) Abrégé : L'invention concerne un procédé de codage et de représentation synthétique d'une ontologie partiellement saturée, comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini représentée par un treillis et des informations non liées aux concepts. Il consiste au moins à coder (A) le treillis de concepts par attribution à chaque concept (Ci) et au noeud du treillis associé à ce dernier d'un identifiant (IDi) formé par au moins une suite de nombres entiers, chaque suite de nombres entiers définissant un chemin d'accès entre un concept considéré (Ci) et le concept universel (T), par l'intermédiaire de concepts pères successifs, l'identifiant incluant toutes les suites de nombres entiers définissant chacune un chemin d'accès entre les concepts pères et le concept universel (T), auxquelles est ajouté un nombre entier représentatif du concept considéré (Ci), calculer (B) les exclusions mutuelles entre concepts pour chaque concept considéré, mémoriser (C) le treillis de concepts codé sous forme d'une structure de données subdivisée en blocs, incluant au moins une liste de concepts incluant au moins un nom de domaine et une liste comportant au moins un concept considéré, chaque concept considéré incluant au moins un identifiant formé par une liste d'au moins un chemin et une liste de concepts mutuellement exclusifs avec le concept considéré. Application à la transmission et la diffusion d'ontologies en réseau IP, sur Internet.

PROCÉDÉ ET SYSTÈME DE CODAGE ET DE REPRÉSENTATION
SYNTHÉTIQUE D'UNE ONTOLOGIE PARTIELLEMENT SATURÉE,
STRUCTURE DE DONNÉES, ET SERVEUR DE STRUCTURE DE DONNÉES
CORRESPONDANTS

L'invention concerne un procédé et un système de codage et de représentation synthétique d'une ontologie partiellement saturée, une structure de données et un serveur de structure de données correspondants.

Une ontologie est une entité formée par un ensemble de connaissances, de faits et de règles relatifs à un domaine donné, domaine scientifique, culturel, administratif, savoir faire industriel ou commercial ou autre.

Ces informations sont traduites par des prédicats (Vol, Pays) pouvant être associés à différentes instances ("714 Sydney", "France"). Les informations implicites, obtenues par saturation des connaissances de l'ontologie, telles que règles, disjonctions, inclusions, définitions, sont calculées par un raisonneur, défini pour un langage ou une logique de description précise, par exemple Carin-ALN qui permet de combiner l'expressivité des clauses de Horn (Carin permet d'utiliser (i) des règles logique telles que *"si mon véhicule est une automobile de tourisme, alors il a 4 roues"* avec (ii) une logique de description quelconque) avec l'expressivité de la logique de description ALN qui permet d'exprimer des classes d'objets, des concepts, à partir des constructeurs associés aux lettres **A** : Top (concept universel), Bottom (concept vide), for All (restriction universelle sur certaines propriétés associées à certains concepts) et Not ; **L** : And (Et logique entre plusieurs concepts) et **N** : At least et At most (restriction de cardinalité sur certaines propriétés de certains concepts).

Les connaissances, faits et règles précités peuvent inclure une hiérarchie de concepts, lesquels représentent des concepts entre un concept vide, engendré par un prédicat qui n'a pas de sens dans l'ontologie considérée, et un concept universel, permettant de concevoir tous les concepts incorporant l'ontologie considérée.

Outre la hiérarchie de concepts précitée, une ontologie peut communément

comprendre des prédicats incluant des entités telles que concept, rôle ou prédicat ordinaire.

Pour assurer la représentation, à partir d'un codage, d'une hiérarchie de concepts d'une ontologie donnée, la méthode la plus aboutie connue actuellement a été proposée par Monsieur Alain Bidault, confer Thèse n° 6932 présentée pour obtenir le grade de docteur es Sciences de l'Université Paris XI Orsay spécialité Informatique par cet auteur, sujet : Affinement de Requêtes posées à un Médiateur au sein du système PICSEL, Paris juillet 2002.

Selon la méthode précitée, les informations contenues dans une ontologie donnée sont traduites par des prédicats, lien, association sémantique ou logique entre atomes signifiants pouvant être associés à différentes instances, valeurs spécifiques signifiantes d'une variable ou d'un atome.

Un processus de saturation de connaissances, règles, disjonctions, inclusions définitions de l'ontologie permet de calculer les informations implicites au moyen d'un raisonneur, défini pour un langage ou une logique de description précis.

Selon la méthode précitée, un concept est un prédicat unaire (n'acceptant qu'un seul argument) avec lequel il est possible de construire des formules logiques de description.

Une ontologie comportant un ensemble de concepts et une hiérarchie de concepts peut être représentée sous forme d'un treillis. Un treillis est alors défini comme une relation d'ordre partiel, pour laquelle tout couple d'éléments ou concepts (e_1 , e_2) possède un élément e_S supérieur à un élément e_I inférieur commun selon les relations :

$$e_1 < e_S \text{ et } e_2 < e_S$$

$$e_1 > e_I \text{ et } e_2 > e_I.$$

Dans ces relations, les symboles $>$ et $<$ désignent la relation d'ordre ou hiérarchie précitée.

L'ordre est partiel, car il n'est pas défini pour tous les éléments : certains éléments sont hiérarchiquement inférieurs à d'autres, mais certains éléments ne sont pas comparables à d'autres. Par ailleurs, un élément peut avoir plusieurs éléments

hiérarchiquement supérieurs et inférieurs.

La structure de treillis est plus riche qu'une structure d'arbre. En effet, un arbre ne permet pas qu'un élément ait plusieurs pères. Or, si aucun élément ou concept ne peut avoir plusieurs pères, la seule possibilité d'avoir un élément inférieur commun est d'imposer qu'un concept n'ait pas plus d'un fils. On obtient dans ces conditions une hiérarchie en ligne verticale de peu d'intérêt.

Une ontologie qui ne contient plus d'information implicite est appelée une ontologie saturée. Dans ces conditions, toute information est accessible en au plus un pas de chaînage avant. On rappelle ici qu'un pas de chaînage avant, ou de saturation, consiste à remplacer, réécrire, la partie gauche d'une règle de définition, soit la condition ou corps de la règle, par sa partie droite, soit la conclusion ou tête de la règle.

Un exemple de treillis de concepts décrit en logique de description est donné en relation avec la figure 1, exemple dans lequel la relation d'ordre du treillis est la relation d'inclusion pour les concepts primitifs A, B, C, D, E, F, G $\subseteq$:

$$\perp \subseteq A \subseteq B \subseteq C \subseteq D \subseteq T ;$$

$$\perp \subseteq G \subseteq T ;$$

$$\perp \subseteq F \subseteq E \subseteq C ;$$

$$A \subseteq E$$

$$F \subseteq B.$$

Selon la méthode précitée, on attribue à chaque concept primitif un identifiant, constitué par une ou plusieurs suites de caractères alphanumériques (a-z, 0-9). Chaque suite est unique sur le treillis de concepts et constitue un chemin d'accès au concept à partir du concept universel T. Les chemins d'accès pour le nœud A sont les suites aaaa et aaba sur la figure 1.

La relation d'ordre du treillis représenté en figure 1 est la subsomption, généralisation, entre deux concepts qui comprend toutes les instances de la relation d'inclusion entre concepts primitifs, A à G.

Une flèche allant d'un premier à un deuxième concept primitif indique que le premier concept est subsumé, généralisé, par le deuxième concept. Pour deux

concepts primitifs, cette relation se réduit à vérifier l'existence d'une relation d'inclusion. Cela dit, un treillis tel que représenté en figure 1 peut en outre incorporer des concepts définis H et I , $H=C \cap G$ avec H subsume I , comme il sera décrit ultérieurement dans la description. En revanche, déterminer si un concept défini en généralise un autre fait intervenir des calculs complexes sur les expressions de logique de description.

En référence à la figure 1 et au treillis de concepts primitifs ou définis représentés, on dit que :

- A est le fils de B et E ;
- B est le père de A et F ;
- B, E, A et F sont les descendants de C ;
- B, E, C et D sont les généralisants de A ;
- T est le concept universel ;
- $\perp$ est le concept vide.

Les définitions précitées s'appliquent en outre aux concepts définis.

La hiérarchie de concepts primitifs et définis contient implicitement toutes les relations calculables par transitivité de la relation d'inclusion, telles que, pour le concept primitif B : $B \subseteq D \subseteq T$. Ce n'est donc pas une ontologie saturée.

Les ontologies décrites en logique de description peuvent actuellement être transmises dans différents formats, ou langages de transmission.

Le format OWL pour Ontology WEB Language en anglais, le plus avancé et le plus complet, propose une vision compactée du domaine de connaissance. Le langage précité permet de décrire des concepts et des exclusions entre concepts et a été étendu via le langage swrl, pour prendre en compte également des règles vérifiant un format spécifique sur des concepts ou des rôles. Le langage swrl a lui-même été étendu par le langage swrl-P, pour prendre en compte n'importe quelle forme de prédicat.

Les techniques précitées, en particulier les formats ou langages de communication tels que xml ou swrl-P, ne permettent pas une représentation à la fois compacte et saturée d'une ontologie. Le domaine n'est ni saturé, ni exploitable de

manière informatique en tant que tel. Exploiter un tel fichier nécessite une compilation des données via un raisonneur qui permet d'obtenir une hiérarchie saturée. Les calculs à effectuer pour obtenir une telle classification sont coûteux et complexes. Ces calculs ne sont accessibles qu'à des experts.

Par ailleurs, le codage de concepts primitifs d'une hiérarchie de concepts par attribution d'un identifiant comportant au moins un chemin d'accès, tel que décrit dans le mémoire de thèse précité, permet d'obtenir un gain de temps sur les calculs liés aux concepts pour le raisonneur ou pour les traitements coopératifs mais n'est associé qu'à une représentation interne du treillis de concepts au sein d'un programme et ne prend pas en compte de nombreuses autres propriétés de l'ontologie. En effet, d'autres informations peuvent apparaître dans une ontologie, et donc dans une version saturée de celle-ci. Ces informations portent sur la synonymie entre concepts, les contraintes d'exclusion, les rôles, les règles, les prédicats n-aires, les contraintes de typage définies sur ces prédicats et sur les faits.

Ainsi, un concept peut avoir plusieurs noms différents en fonction de son contexte d'utilisation, lorsqu'en particulier il est relatif à un domaine de pointe, dont la terminologie n'a pas encore été adoptée et ne fait, donc, pas encore l'objet d'un consensus. Par extension de langage, ces noms peuvent être considérés comme des synonymes au sein de l'ontologie.

Exemple : voiture et automobile, email et courriel.

En outre, une contrainte d'exclusion permet de préciser que deux concepts sont incompatibles, ainsi que leurs descendants.

Exemple : EnOcéanie est mutuellement exclusif avec EnAfrique, ainsi qu'avec AuTogo, AuMaroc, etc.

Les rôles sont des prédicats binaires, comportant deux arguments, qui sont utilisés pour associer deux concepts entre eux. Il est souvent possible de spécifier les types de chacun des arguments du rôle.

Exemple : SituéDans localise un lieu de Résidence dans un Lieu Géographique.

Un prédicat ordinaire est un prédicat qui n'est ni un concept, ni un rôle.

Chaque argument du prédicat est le plus souvent typé.

Exemple : VolAller-Retour a pour arguments un numéro de vol, deux concepts associés à des aéroports, deux dates de départ et deux dates d'arrivée.

Une règle, dite règle de définition, permet de définir un prédicat ordinaire en fonction de l'ensemble des prédicats de l'ontologie, concepts et rôles compris. L'atome, tête de la règle, est déduit de la liste des atomes provenant du corps. Chaque règle de définition qui conclut sur un prédicat déterminé, définit ce dernier. Toutefois, ces définitions ne correspondent pas à une équivalence entre le corps de la règle et la tête car il ne s'agit que d'une simple implication.

Exemple : un vol ayant un tarif associé économique implique ($\rightarrow$) qu'il s'agit d'un vol économique.

Les faits sont des informations précises liées au domaine. Un fait associe un prédicat à une ou plusieurs constantes.

Exemple : Aéroport ("Paris-CDG"), Vol("714-Sydney").

La présente invention a pour objet de remédier aux inconvénients et limitations des techniques de l'art antérieur.

En particulier, un objet de la présente invention est la mise en œuvre d'un procédé et d'un dispositif de codage et de représentation synthétique d'une ontologie partiellement saturée dans un format ou langage de communication textuel et structuré, de type xml ou Owl.

Un autre objet de la présente invention est, en particulier, de permettre à n'importe quel Internaute d'utiliser pleinement la richesse d'ontologies, sans avoir à installer, manipuler ou utiliser de raisonneur ou de fonctions complexes pour saturer toute ontologie communiquée, ceci afin de permettre, d'une part, un gain de temps, et, d'autre part, une utilisation beaucoup plus aisée de toute ontologie par un utilisateur non averti.

Un autre objet de la présente invention est, enfin, la mise en œuvre d'un procédé et d'un dispositif de codage et de représentation synthétique d'une ontologie partiellement saturée faisant appel à un langage ou format de communication et aux règles d'écriture syntaxique, ou grammaire, d'utilisation, de ce dernier

particulièrement simple et adaptable à de nombreux formats de langages structurés de balisage, tel que langage xml ou Owl.

Le procédé de codage et de représentation synthétique d'une ontologie partiellement saturée, objet de la présente invention, cette ontologie comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini représentée par un treillis sous un concept universel et des informations non liées aux concepts, est remarquable en ce qu'il consiste au moins à coder ce treillis de concepts par attribution à chaque concept et au nœud du treillis associé à ce dernier d'un identifiant formé par au moins une suite de nombres entiers, chaque suite de nombres entiers définissant un chemin d'accès entre un concept considéré et le concept universel, par l'intermédiaire de concepts pères successifs, cet identifiant incluant toutes les suites de nombres entiers définissant chacune un chemin d'accès entre l'un de ces concepts pères et le concept universel, auxquelles est ajouté un nombre entier représentatif du concept considéré, calculer les exclusions mutuelles entre concepts pour chaque concept considéré et mémoriser le treillis de concepts codé sous forme d'une structure de données subdivisée en blocs incluant au moins une liste de concepts incluant au moins un nom de domaine et une liste comportant au moins un concept considéré, chaque concept considéré incluant au moins un identifiant, formé par une liste d'au moins un chemin et une liste de concepts mutuellement exclusifs avec le concept considéré.

Le procédé, objet de l'invention, est en outre remarquable en ce qu'il consiste à établir la structure de données de mémorisation de l'ontologie à partir d'un ensemble de règles de définition, chaque règle de définition comportant une première partie mise en relation avec une deuxième partie, cette deuxième partie constituant une définition de la première partie, à partir d'une pluralité de mots clés spécifiques à valeur syntaxique, permettant d'organiser syntaxiquement les blocs de la structure de données et à partir d'une pluralité de symboles, chaque symbole représentant une valeur sémantique attribuée à un mot clé spécifique, la structure de données subdivisée en blocs contenant des blocs terminaux constitués chacun par un mot clé et au moins un symbole, cet ensemble de règles de définition et cette pluralité de

symboles étant constitués en un langage de transmission et de communication du treillis de concepts codé, dont les règles d'écriture constituent les règles d'écriture syntaxique de cette structure de données.

Le procédé, objet de l'invention, est également remarquable en ce que, les informations non liées aux concepts incluant des rôles, des prédicats ordinaires et des faits, il consiste en outre à :

- coder chaque rôle selon un prédicat binaire incluant au moins une en-tête associée à un nom de rôle, et au moins un premier argument, de type concept, déterminé par le rôle et un deuxième argument, de type concept, déterminant dans le rôle ;
- coder chaque prédicat ordinaire selon au moins une en-tête associée à un nom de prédicat, un premier argument de type argument et un deuxième argument de liste de définition, chaque argument présentant en outre une valeur sémantique ;
- coder chaque fait selon un en-tête associé à un atome, dont l'un au moins des arguments est une constante.

Le procédé objet de l'invention, est enfin remarquable en ce que le langage de transmission et de communication comporte en outre un ensemble d'opérateurs logiques d'expressions de logique de description incluant au moins :

- un premier opérateur logique permettant la mise en relation logique d'intersection entre une première et une deuxième expression de logique de description par l'intermédiaire d'un en-tête "AND" ;
- un deuxième opérateur logique permettant la mise en relation d'un nom de rôle et d'une expression logique de description par l'intermédiaire d'un en-tête "FA" ;
- un troisième opérateur logique permettant la mise en relation d'un entier minorant et d'un nom de rôle par l'intermédiaire d'un en-tête "AL" ;
- un quatrième opérateur logique permettant la mise en relation d'un entier majorant et d'un nom de rôle par l'intermédiaire d'un en-tête "AM" ;
- un cinquième opérateur logique permettant la négation ou exclusion d'un élément de base, tel qu'un concept, par l'intermédiaire d'un en-tête "NEG".

Le système de codage et de représentation synthétique d'une ontologie partiellement saturée comportant au moins une hiérarchie de concepts formée par des concepts primitifs sous un concept universel et au moins un concept défini représentée par un treillis et des informations non liées aux concepts, objet de la présente invention, est remarquable en ce que, outre des organes d'entrée/sortie, une unité centrale de traitement et une mémoire de travail, il inclut au moins un module d'acquisition d'une ontologie permettant de mémoriser ledit treillis et lesdites informations non liées au treillis, un module de codage du treillis de concepts par attribution à chaque concept et au nœud du treillis associé à ce dernier d'un identifiant formé par au moins une suite de nombres entiers, chaque suite de nombres entiers définissant un chemin d'accès entre un concept considéré et le concept universel, par l'intermédiaire de concepts pères successifs, cet identifiant incluant toutes les suites de nombres entiers définissant chacune un chemin d'accès entre l'un de ces concepts pères et le concept universel, auxquelles est ajouté un nombre entier représentatif du concept considéré, un module de calcul des exclusions mutuelles entre concepts pour chaque concept considéré et une ressource de mémorisation du treillis de concepts codé sous forme d'une structure de données subdivisée en blocs incluant au moins une liste de concepts incluant au moins un nom de domaine et une liste comportant au moins un concept considéré, chaque concept considéré incluant au moins un identifiant formé par une liste d'au moins un chemin et une liste de concepts mutuellement exclusifs avec le concept considéré.

L'invention couvre, en outre, un système de consultation d'une ontologie saturée, comportant au moins des organes d'entrée/sortie, une unité centrale de traitement et une mémoire de travail, remarquable en ce qu'il comporte en outre une unité d'acquisition d'une structure de données subdivisée en blocs et représentative de cette ontologie comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini représentée par un treillis, précédemment mentionnée, et un module interpréteur de requête de consultation, transmise par un terminal utilisateur.

L'invention couvre, enfin, un système serveur d'accès en réseau IP à une

ontologie saturée, remarquable en ce qu'il comporte au moins, interconnectés, un portail fournisseur d'accès à cette ontologie partiellement saturée, à partir d'une requête de consultation, et, un système de consultation d'une ontologie saturée précédemment mentionné.

Le procédé et le système de codage et de représentation synthétique d'une ontologie partiellement saturée, objets de la présente invention, trouvent application particulièrement avantageuse à la transmission et à la communication d'ontologies sous forme de structures de données directement accessibles sur Internet et consultables à partir de tout langage de balisage structuré disponible en ligne, tel que le langage xml par exemple.

Ils seront mieux compris à la lecture de la description et à l'observation des dessins ci-après, dans lesquels, outre la figure 1, relative à l'art antérieur, représentative d'un treillis de concepts primitifs :

- la figure 2a représente, à titre d'exemple non limitatif, un organigramme des étapes essentielles du procédé de codage et de représentation d'une ontologie partiellement saturée, conforme à l'objet de la présente invention ;
- les figures 2b et 2c représentent, à titre d'exemple non limitatif, un organigramme des étapes essentielles de l'étape de codage proprement dite d'un treillis de concepts primitifs et définis, conforme à l'objet de la présente invention telle que représentée en figure 2a ;
- la figure 2d représente, à titre d'exemple non limitatif, un treillis comportant des concepts primitifs ou définis codés et obtenus suite à la mise en œuvre de l'étape de codage proprement dite illustrée en figure 2c ;
- la figure 3a représente, à titre purement illustratif, un mode de mise en œuvre préféré de l'étape de calcul des exclusions mutuelles entre concepts telle que décrite en figure 2a, le mode de mise en œuvre préféré précité comportant, avantageusement, une sous-étape d'initialisation d'exclusion suivie d'une sous-étape de propagation d'exclusion ;
- la figure 3b représente, à titre purement illustratif, un mode de mise en œuvre spécifique préférentiel de la sous-étape d'initialisation d'exclusion

représentée en figure 3a ;

- la figure 3c représente, à titre purement illustratif, un mode de mise en œuvre spécifique préférentiel de la sous-étape de propagation d'exclusion représentée en figure 3a ;

- la figure 4a représente, à titre d'exemple non limitatif, un schéma fonctionnel d'un système de codage et de représentation synthétique d'une ontologie partiellement saturée, conforme à l'objet de la présente invention ;

- la figure 4b représente, à titre d'exemple non limitatif, un schéma fonctionnel d'un système de consultation d'une ontologie partiellement saturée, conforme à l'objet de la présente invention ;

- la figure 4c représente, à titre d'exemple non limitatif, un schéma fonctionnel d'un système serveur d'accès en réseau IP à une ontologie partiellement saturée, conforme à l'objet de la présente invention.

Une description plus détaillée du procédé de codage et de représentation synthétique d'une ontologie partiellement saturée, conforme à l'objet de la présente invention, sera maintenant donnée en liaison avec la figure 2a et les figures suivantes.

En référence à la figure 2a, on indique que le procédé de codage et de représentation synthétique d'une ontologie partiellement saturée, conforme à l'objet de la présente invention, s'applique à toute ontologie comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini, cette ontologie étant représentée par un treillis et des informations non liées au concept.

En référence à la figure 2a, le treillis de concept est noté :

$$\{\perp \subseteq C_i \subseteq T\}.$$

Dans la relation précédente $C_i = \{PC_i, DC_k\}$ désigne chacun des concepts formant la hiérarchie de concepts formés par des concepts primitifs PC_i et au moins un concept défini DC_k , l'ensemble des concepts précités étant représenté par le treillis vérifiant la relation d'ordre précédemment mentionnée.

Dans la suite de la description, on indique que tout concept C_i est soit un concept primitif PC_i soit un concept défini DC_k , indifféremment pour la mise en

œuvre du procédé objet de la présente invention.

Enfin, l'ontologie comporte, outre le treillis précité, des informations non liées au concept dans la hiérarchie de concepts, c'est-à-dire dans le treillis, ces informations pouvant bien entendu toutefois concerner un ou plusieurs des concepts constitutifs de la hiérarchie de concepts précitée.

Ainsi que représenté en figure 2a, le procédé, objet de l'invention, consiste en une étape A à coder ledit treillis de concepts par attribution à chaque concept C_i et au nœud du treillis associé à ce dernier d'un identifiant, noté ID_i , formé par au moins une suite de nombres entiers. Chaque suite de nombres entiers définit un chemin d'accès entre un concept considéré C_i et le concept universel T, chaque chemin étant noté CH_{i1} , CH_{ip} de manière non limitative.

Chaque chemin d'accès définit un chemin d'accès entre un concept considéré C_i et le concept universel T par l'intermédiaire de concepts pères successifs.

Ainsi, l'identifiant ID_i associé au concept C_i inclut toutes les suites de nombres entiers définissant chacune un chemin d'accès entre l'un des concepts pères du concept considéré C_i et le concept universel T auquel est ajouté un nombre entier représentatif du concept considéré C_i .

L'opération de codage de l'étape A est notée par la relation symbolique :

$$C_i \rightarrow ID_i[CH_{i1}, \dots, CH_{ip}].$$

L'étape A est alors suivie d'une étape B consistant à calculer les exclusions mutuelles entre concepts pour chaque concept C_i considéré.

L'opération de calcul des exclusions mutuelles à l'étape B est notée par la relation symbolique :

$$CE_i[C_{j \neq i}] | C_j \cap C_i = \perp.$$

Dans la relation précédente CE_i désigne la liste des concepts C_j , avec $j \neq i$, vérifiant l'exclusion mutuelle avec le concept considéré C_i , l'exclusion mutuelle étant vérifiée par la relation :

$$C_j \cap C_i = \perp.$$

On comprend en particulier que toute absence d'intersection entre le concept considéré C_i et tout concept C_j appartenant à la hiérarchie des concepts représentée

par le treillis avec $j \neq i$ constitue ainsi un concept mutuellement exclusif entre les concept C_i et C_j .

L'opération de l'étape B est alors suivie d'une étape C consistant à mémoriser le treillis de concepts codé sous forme d'une structure de données subdivisée en blocs incluant au moins une liste de concepts incluant au moins un nom de domaine DN et une liste comportant au moins un concept considéré C_i . Chaque concept considéré précité inclut au moins un identifiant ID_i formé par une liste d'au moins un chemin et une liste de concepts CE_i mutuellement exclusifs avec le concept C_i considéré calculée à l'étape B précédente.

L'opération de mémorisation à l'étape C est notée par la relation symbolique :

$$DS_{in}[DN[C_i[ID_i[CE_i]]]].$$

Dans la relation précédente, on comprend que DS_{in} désigne la structure de données élémentaire subdivisée en blocs incluant la liste de concepts, incluant le nom de domaine DN et une liste comportant au moins le concept considéré C_i , l'identifiant ID_i calculé à l'étape A, la liste de concepts exclusifs mutuels CE_i calculée à l'étape B.

L'étape C peut alors être suivie d'une étape D consistant à comparer le concept considéré C_i au concept vide $\perp$, ceci pour chaque niveau de profondeur n du treillis, le calcul des étapes B et C pouvant être effectué pour chaque niveau de profondeur n du treillis.

Sur réponse négative au test de l'étape D, le concept C_i n'étant pas le concept vide et le concept vide n'ayant pas été atteint, et, par conséquent, la profondeur maximale du treillis n'ayant pas non plus été atteinte, une étape E est appelée consistant à passer au niveau de profondeur supérieur par incrémentation du niveau $n=n+1$. L'étape E est alors suivie d'un retour à l'étape B pour effectuer successivement le calcul des exclusions mutuelles et la mémorisation par actualisation de la structure de données subdivisée en blocs, pour chacun des niveaux successifs du treillis et finalement pour la totalité du treillis considéré.

Au contraire, en réponse positive au test D de la figure 2a le concept considéré correspondant au concept vide, une fin du procédé de codage est appelée.

En conséquence, on dispose d'une structure de données complète subdivisée en blocs, laquelle est notée :

$$DS[DS_{in}[.]]_{n=1}^{n=nf}.$$

Dans la relation précédente on comprend que DS désigne la structure de données globale représentative du treillis de concept codé, cette structure de données étant bien entendu formée de toutes les structures de données élémentaire DS_{in} relatives à chaque concept C_i considéré du treillis.

Une description plus détaillée de l'étape A de codage représentée en figure 2a sera maintenant donnée en liaison avec les figures 2b, 2c et 2d.

Pour la mise en œuvre du processus de codage proprement dit à l'étape A de la figure 2a, on considère la mise en œuvre du processus de codage précité de manière préférentielle sur les concepts primitifs PC_i puis sur les concepts définis DC_k dans les conditions ci-après.

En référence à la figure 1 on rappelle que le procédé, objet de l'invention, est mis en œuvre sur une ontologie quelconque, telle que représentée en figure 1 par exemple, dans laquelle chaque concept désigné ABCDEF est constitué par un concept primitif vérifiant la relation d'ordre précitée d'inclusion entre un concept universel T et un concept vide $\perp$ ou par une ontologie comportant des concepts primitifs et des concepts définis DC_k .

Pour la mise en œuvre du procédé objet de la présente invention en liaison avec la figure 2b, tout concept primitif PC_i où i désigne une adresse d'identification éventuelle d'un concept primitif donné considéré, prend la valeur A à G pour la hiérarchie des concepts primitifs représentée en figure 1 et vérifie la relation :

$$\{\perp \subseteq PC_i \subseteq T\}.$$

Dans la relation précédente on rappelle que le symbole $\subseteq$ désigne le symbole d'inclusion.

Le procédé de codage, objet de l'invention, consiste, ainsi qu'il apparaît en figure 2b, à attribuer à chaque concept primitif PC_i prenant l'une des valeurs précédentes et, bien entendu, au nœud du treillis associé à ce dernier, un identifiant formé par au moins une suite de nombres entiers, chaque identifiant étant de ce fait

représenté par au moins une suite de nombres entiers délimitée par des crochets ainsi qu'il sera expliqué ci-après.

En référence à la figure 2b, on indique qu'à chaque concept primitif PC_i est attribuée pour seule valeur sémantique la valeur de signification intrinsèque qui lui est associée par la valeur littérale de ce dernier dans la position correspondante au nœud du treillis auquel le concept est associé.

Ainsi, chaque suite de nombres entiers définit un chemin d'accès entre le concept primitif considéré PC_i et le concept universel T, par l'intermédiaire des concepts primitifs pères successifs.

Enfin, ainsi que représenté en figure 2b, l'identifiant associé à un concept primitif et au nœud du treillis correspondant inclut toutes les suites de nombres entiers définissant chacune un chemin d'accès entre les concepts primitifs pères et le concept universel T, auquel est ajouté un nombre entier représentatif du concept primitif considéré.

Un mode de mise en œuvre de codage spécifique non limitatif sera décrit en liaison avec la figure 2b, dans laquelle la structure hiérarchique de la figure 1 entre le concept universel T et le concept vide $\perp$ est préférentiellement, mais de manière non limitative, parcourue en "niveau d'abord" par exemple.

La notion de "niveau d'abord" concerne, à partir du concept universel T, le parcours de la hiérarchie de concepts par exemple par niveau de filiation successif de façon non limitative.

Ainsi, en référence à la figure 2b on considère en premier lieu les concepts D et G pour le premier niveau de filiation vis-à-vis du concept universel T, niveau $n=1$.

Pour les concepts primitifs $PC_i=D$ et respectivement $PC_i=G$ on attribue ainsi un identifiant, identifiant [1] pour le concept D, respectivement identifiant [2] pour le concept G.

Ainsi, à l'étape C_0 de la figure 2b on procède à l'attribution des identifiants pour les concepts suivants :

$$PC_i=D[1] \quad PC_i=G[2].$$

Suite au niveau de filiation 1, par exemple, on procède alors au passage au niveau de filiation 2, c'est-à-dire à l'étude de tout concept PC_i relié au concept universel T par un concept intermédiaire.

Pour l'étape C_1 de la figure 2b, on procède alors à l'attribution au deuxième niveau de filiation $n=2$:

$$PC_i = C[11].$$

L'on procède alors ainsi de suite pour tout niveau de filiation successif et, en particulier, pour le niveau de filiation $n=3$, c'est-à-dire sur la figure 1 pour les concepts B et E à l'étape C_2 selon la relation d'attribution :

$$PC_i = B[111] \quad PC_i = E[112].$$

Pour l'exécution du niveau de filiation suivant, le niveau $n=4$, l'on procède alors à l'attribution suivante :

$$PC_i = A[1111 \ 1121] \quad PC_i = F[1112 \ 1122].$$

On comprend que le codage et, en particulier, l'association d'identifiants est alors terminée lorsque l'attribution d'un identifiant à chacun des concepts primitifs autres que le concept vide $\perp$ a été effectuée, cette opération ayant été représentée à la figure 2b par une étape "fin niveau" par la relation :

$$PC_i = \perp.$$

On obtient alors un treillis de concepts noté TPC_i sur la figure 2b.

On comprend, en particulier, que l'identifiant associé à chacun des concepts PC_i inclut toutes les suites de nombres entiers définissant chacune un chemin d'accès entre l'un des concepts primitifs pères et le concept universel, auquel est ajouté un nombre entier représentatif du concept primitif considéré.

On indique que le nombre entier précité ajouté est représentatif du concept primitif considéré vis-à-vis de chaque concept primitif père ce dernier.

Ainsi, pour le concept primitif $PC_i = F$, le nombre entier ajouté vis-à-vis du concept primitif père B de ce dernier est le nombre 2 en raison de l'existence du concept fils A vis-à-vis du concept primitif père commun B.

Tout concept primitif fils, tel que le concept primitif F, ne se voit attribuer qu'un seul nombre entier d'identification, tel que le nombre 2, vis-à-vis du seul et

même concept primitif père B, indifféremment du nombre entier d'identification attribué vis-à-vis d'un autre concept primitif père, tel que E.

Ainsi, en référence à la figure 2b on indique que, de manière avantageuse, chaque nombre entier ajouté est un numéro d'ordre du concept primitif considéré représentatif du rang de filiation du concept primitif considéré précité, pris comme concept primitif fils d'une pluralité de concepts primitifs pères communs, c'est-à-dire concepts primitifs B et E sur la figure 1, vis-à-vis du concept primitif fils F.

Le procédé de codage sous forme d'un treillis de concepts, d'une hiérarchie de concepts appartenant à une ontologie, objet de la présente invention, tel que représenté et décrit en figure 2b, n'est pas limitatif.

En particulier, on comprend que le parcours de la hiérarchie de concepts représenté en figure 1a peut être effectué de toute manière adaptée et en particulier selon un processus de parcours "profondeur d'abord" lequel peut consister à parcourir la hiérarchie des concepts non plus par niveau, comme décrit en liaison avec la figure 2b, mais par branche en privilégiant le parcours par nœud successif d'une même filiation successive.

On comprend, en particulier que le procédé, objet de l'invention, tel que décrit en liaison avec la figure 2b correspond à une numérotation d'une hiérarchie qui s'inspire du tri topologique défini sur un graphe orienté.

Un graphe orienté est une relation binaire non symétrique où chaque élément de la relation est représenté par un nœud dans le graphe et où chaque occurrence de la relation fait apparaître sur le graphe un arc orienté, c'est-à-dire une flèche d'un nœud donné vers un autre nœud donné.

L'algorithme du tri topologique permet d'appliquer un traitement à un nœud une fois que tous ses antécédents ont été traités.

C'est le cas dans le mode de mise en œuvre tel que représenté en figure 2b du procédé objet de l'invention.

Ainsi, un concept primitif de la hiérarchie de concepts primitifs est numéroté une fois que tous ses pères ont été numérotés. Pour garder les informations relatives à la hiérarchie, un concept fils hérite de tous les chemins de l'identifiant de

chacun de ses pères étendu à droite par un nouvel entier, l'entier ajouté étant le même pour tous les chemins d'un même père donné à un de ses fils. Ce mode opératoire permet de garantir qu'un chemin n'est associé qu'à un seul concept de la hiérarchie de concepts et que le caractère ajouté est différent pour chacun des concepts fils considérés.

Ainsi, en référence à la figure 2b, l'ensemble des chemins étendus obtenu à partir de tous les concepts pères d'un concept primitif, considéré comme un concept fils de tous les concepts pères précités, constitue l'identifiant du concept primitif fils considéré précité.

Toutefois, certaines ontologies possèdent des redondances dans leur déclaration, dues à la propriété de transitivité de la relation d'inclusion entre concepts primitifs.

Les redondances précitées sont relativement rares dans le cadre de concepts primitifs qui ont pour seule valeur celle associée à leur position dans le treillis de concept et, bien entendu, à leur définition littérale intrinsèque.

La notion de position recouvre la notion de filiation et finalement de position de chaque nœud dans le treillis au-dessus et/ou au dessous d'un autre nœud et, en définitive, d'un concept associé à ce dernier.

La sémantique simple précitée est en général contrôlée par un expert d'ontologie, lequel définit l'ontologie en tant que telle avec la hiérarchie des concepts associée au concept primitif considéré.

Toutefois, une ontologie en tant que telle, et, en particulier une hiérarchie de concepts primitifs constitutifs de celle-ci, permet une meilleure exploitation automatique des propriétés de l'ontologie considérée, lorsqu'il n'existe pas de redondance dans la hiérarchie des concepts telle que définie.

La détection des redondances précitée pour des concepts primitifs est simple car elle consiste à vérifier l'absence de redondance par étude des instances de relation d'inclusion.

À titre d'exemple non limitatif, sur la figure 1, l'existence d'un arc entre les concepts F et C ou F et D aurait constitué une relation d'ordre redondante.

En outre, les ontologies prises en compte pour la mise en œuvre du procédé objet de l'invention vérifient la contrainte d'unicité du nom de concept donnant différents représentants des prédicats différents, mais un même nom représentant le même prédicat. L'absence d'une telle contrainte a pour conséquence qu'un même concept est dupliqué dans le treillis et le codage par numérotation a pour effet d'attribuer à ce dernier deux identifiants différents mais équivalents pour tous les traitements.

Le procédé de codage sous forme d'un treillis d'une hiérarchie de concepts appartenant à une ontologie, conforme à l'objet de la présente invention, tel que décrit en figure 2b pour des concepts primitifs, peut de manière particulièrement avantageuse être mis en œuvre, en outre, pour une hiérarchie de concepts appartenant à une ontologie comportant, non seulement, des concepts primitifs mais également au moins un concept défini, moyennant quelques précautions de mise en œuvre, lesquelles seront explicitées ci-après.

D'une manière générale, on indique que le processus de codage tel que décrit en figure 2b nécessite, en présence de concepts définis dans la hiérarchie des concepts, la mise en œuvre de certaines étapes supplémentaires, lesquelles seront décrites en liaison avec la figure 2c.

En effet, le treillis obtenu en prenant en compte des concepts définis correspond à une relation d'ordre plus large que la relation d'ordre d'inclusion d'un treillis relatif à des concepts primitifs.

En particulier, la relation d'ordre associée à un treillis comportant des concepts primitifs et des concepts définis est une relation d'ordre partiel obtenue par calcul de subsomption entre deux éléments, c'est-à-dire entre concept primitif et/ou défini. On rappelle que les tests de subsomption, propres à un langage d'ontologie, permettent d'indiquer si un concept généralise un autre concept.

Le calcul précité entre deux éléments de concept correspondants se réduit parfois à vérifier l'existence d'une instance de la relation d'inclusion, mais, en général, fait intervenir des calculs complexes sur les expressions de logique de description de l'ontologie et, en particulier, de la hiérarchie de concepts constitutive de cette

dernière. De tels calculs complexes ne seront pas décrits dans la présente description, car le mode d'exécution de tels calculs est étranger à l'objet de l'invention, puisque, précisément, l'objet de l'invention permet d'éviter la nécessité d'exécution des calculs précités.

Compte tenu des considérations précédentes, la présente invention a également pour objet, un procédé de codage d'une hiérarchie de concepts comportant des concepts primitifs et, au moins, un concept défini dans les conditions ci-après, tel que représenté en figure 2c.

Pour la mise en œuvre du procédé de codage objet de l'invention, on considère une hiérarchie de concepts comportant des concepts primitifs PC_i vérifiant la relation précédemment mentionnée dans la description :

$$\{\perp \subseteq PC_i \subseteq T\}$$

et des concepts définis notés :

$$\{DC_{kx}\}$$

chacun des concepts définis précités pouvant être le siège de redondances telles que définies précédemment.

En référence à la figure 2c, on indique que le procédé de codage, objet de l'invention, consiste au moins, en une étape notée SA, à engendrer une saturation de la hiérarchie de concepts par exécution de test d'inclusion sur les concepts primitifs. Cette opération est notée :

$$SA\{\perp \subseteq PC_i \subseteq T\} \mid PC_i \subseteq PC_i'.$$

Le test de saturation permet ainsi d'établir toutes les relations d'inclusion implicites existant entre les concepts primitifs PC_i de la hiérarchie de concepts. L'étape de saturation SA est accompagnée d'une étape de codage AC analogue à celle représentée en figure 2b, pour l'ensemble des concepts primitifs précités, l'opération de codage A permettant alors de mettre en œuvre une structure de données sous forme d'un treillis de concepts primitifs notée TPC_i , c'est-à-dire une structure analogue, à celle obtenue grâce à la mise en œuvre du procédé de codage objet de l'invention, sur les concepts primitifs représentés en figure 2b.

Cette opération à l'étape A est notée par la relation :

$$\{\perp \subseteq PC_i \subseteq T\} \rightarrow TPC_i.$$

L'étape A de codage est alors suivie d'une étape de saturation notée SU de la hiérarchie de concepts par exécution de tests de subsomption des concepts définis avec détection des synonymes.

Sur la figure 2c, l'étape de saturation par test de subsomption est notée :

$$\exists PC_i \text{ SU } DC_{kx} ? \quad \exists \text{ Syn ?}$$

Dans la relation précédente, on indique que la première identité concerne un test de subsomption de tout concept défini DC_{kx} susceptible d'ailleurs de présenter encore des redondances vis-à-vis de tout concept primitif PC_i qui a déjà fait l'objet d'un codage sous forme de treillis, le test de saturation précité permettant bien entendu de vérifier l'intégration possible de tout concept défini DC_{kx} dans la hiérarchie de concepts primitifs vérifiant la relation d'ordre total d'inclusion.

La deuxième identité d'existence des synonymes permet de traiter toute existence d'un synonyme afin, bien entendu, de ne pas perdre la relation d'unicité de désignation, le processus de traitement correspondant faisant l'objet d'une description plus détaillée ultérieurement dans la description.

L'étape de saturation par subsomption SU peut alors être suivie d'une étape de détection/suppression des redondances, notée RDC sur la figure 2, dans les concepts définis DC_{kx} , ces redondances pouvant apparaître du fait du caractère de la transitivité de la relation de subsomption vérifiée.

L'opération de détection des redondances est notée par la relation :

$$\{DC_{kx}\} \rightarrow \{DC_k\}.$$

Dans la relation précédente on comprend, en particulier, que les redondances étant supprimées, chaque concept DC_k de l'ensemble des concepts définis $\{DC_k\}$, dans lesquels les redondances ont été supprimées, peut alors être assimilé globalement à un concept primitif du point de vue du codage par numérotation vis-à-vis des concepts primitifs existant dans le treillis de concept primitif TPC_i .

En conséquence, l'étape de détection/suppression des redondances RDC de la figure 2c est suivie par une étape I d'introduction des concepts définis DC_k dans lesquels les redondances ont été supprimées dans le treillis de concepts primitifs

TPC_i .

L'opération à l'étape I précitée est notée :

$$TPC_i + \{DC_k\} \rightarrow CTDP = \{\perp \subseteq PC_i, DC_k \subseteq T\}.$$

On conçoit, en effet, que l'opération d'introduction précitée consiste en fait à introduire par exemple dans un fichier numérique l'intitulé des concepts définis DC_k dans lesquels les redondances ont été supprimées avec les intitulés des concepts primitifs PC_i pour engendrer finalement un ensemble de concepts primitifs et définis, lequel, en raison des traitements de suppression des redondances des concepts définis, peut alors être soumis à un codage semblable à celui exécuté et décrit en liaison avec la figure 2b vis-à-vis de tous les concepts présents formant la hiérarchie de concepts représentative de l'ontologie considérée.

En conséquence, l'étape I d'introduction précitée est suivie, à nouveau, d'une étape de codage A, laquelle est cette fois appliquée sur l'ensemble des concepts primitifs et définis constitutifs de la hiérarchie de concepts représentative de l'ontologie considérée.

Ainsi l'opération de codage A de la figure 2c (deuxième occurrence) porte cette fois sur l'ensemble des concepts primitifs PC_i et des concepts définis DC_k selon la relation :

$$\{\perp \subseteq PC_i, DC_k \subseteq T\} \rightarrow TDPC.$$

La relation symbolique précédente indique que par le codage du corps de treillis de concepts incluant les concepts primitifs et les concepts définis sous forme de treillis de concepts on attribue en fait à chaque concept défini DC_k et au nœud du corps de treillis associé à ce dernier, un identifiant compatible avec tout identifiant attribué à un ou plusieurs concepts primitifs et/ou défini pour former en fait un treillis de concept TDPC incluant à la fois les concepts primitifs et les concepts définis précités.

Pour une description plus détaillée des étapes SA, A, SU, RDC et I et d'un processus spécifique de détection des synonymes, on pourra utilement se reporter à la demande de brevet français 05 07326 déposée le 8 juillet 2005 au nom de la demanderesse.

Dans un exemple non limitatif de mise en œuvre, les synonymes d'un concept peuvent être simplement déclarés lors de la création de l'ontologie et du treillis de concepts. Leur détection se réduit alors à la lecture des synonymes déclarés.

La figure 2d représente une structure de données relative à un treillis numéroté comportant des concepts primitifs A à G et des concepts définis H et I codés par numérotation ainsi que décrit précédemment dans la description en liaison avec les figures 2b et 2c.

A chacun des concepts primitifs ou définis constitutifs du treillis précité, et au nœud du treillis correspondant, est associé un identifiant ID_i formé par une suite de chemins, chaque chemin étant représenté par une suite de nombres entiers conformément à la mise en œuvre du processus de codage décrit précédemment en liaison avec les figures précitées.

D'une manière générale, on indique que le codage par numérotation de concepts tel que décrit précédemment avec les figures 2b et 2c pour l'obtention d'un treillis de concepts codés, tel que représentés en figure 2c, permet, d'une manière particulièrement avantageuse, de représenter les concepts d'une ontologie saturée via une liste de concepts et leurs identifiants ID_i .

Chaque identifiant ID_i est unique sur le treillis de concepts et correspond à l'existence d'au moins un chemin entre un concept considéré C_i et le concept universel T au sommet du treillis, bien que, ainsi que mentionné précédemment, le concept universel T et le concept vide $\perp$ n'aient pas d'identifiant en tant que tel.

En référence à la demande de brevet français 05 07326 précédemment citée dans la description, on rappelle que le codage par numérotation d'un treillis tel que représenté en figure 2c permet de retrouver tous les généralisants d'un concept considéré C_i et ses descendants en parcourant l'identifiant ID_i du concept considéré C_i et la liste des concepts. Il est en outre possible de connaître les nombres maximal et minimal d'arcs qui séparent chaque concept considéré C_i du concept universel T et, par conséquent, la profondeur ou niveau du concept considéré C_i dans le treillis.

Exemple : le concept B[111] a pour concepts généralisants les concepts

définis par les chemins [11] et [1], à savoir les concepts C et D et est séparé par au plus et au moins trois arcs du concept universel T.

Le concept B de la figure 2d a pour concept descendant les concepts dont un des chemins de l'identifiant commence par [111], à savoir les concepts A et F.

Le concept I [231 1131] est séparé par au moins trois arcs et au plus par quatre arcs du concept universel T et sa profondeur est de 4.

Le codage par numérotation tel que décrit en liaison avec les figures 2b et 2c précitées permet également d'effectuer un test de subsomption entre deux concepts du treillis représentés en figure 2b. Un test de subsomption est un calcul complexe dans une ontologie décrite en logique de description qui permet de déterminer si les instances d'un concept sont incluses ou non dans celle d'un autre concept, en se basant sur la définition des concepts.

Grâce au codage par numérotation précitée, une détermination de préfixes permet alors d'être substituée à l'exécution d'un test de subsomption complexe.

Exemple : le concept C[11] subsume le concept A[1111 1121] car le premier chemin de l'identifiant de C préfixe au moins un des chemins de l'identifiant de A.

Enfin, le codage par numérotation précitée permet, en outre, de déterminer les plus petits communs généralisants, désignés ppcg, entre deux concepts C1 et C2. L'ensemble des généralisants communs ξ_{gc} contient tous les concepts qui subsument à la fois le concept C1 et le concept C2. Le ppcg est le plus grand sous-ensemble de l'ensemble des généralisants communs ξ_{gc} tel qu'aucun concept du ppcg ne subsume un autre concept du ppcg et tel qu'aucun concept du ppcg ne subsume un concept de l'ensemble reste. Le ppcg de deux concepts peut être directement accessible dans une ontologie saturée. Dans le cadre du même contexte il est alors facile d'étendre le calcul du ppcg de deux concepts à n concepts. La simplicité du calcul du ppcg qui utilise la numérotation, c'est-à-dire les identifiants et chemins, est alors d'autant plus appréciable.

Exemple : le ppcg du concept A[1111 1121] et du concept F[1112 1122] est l'ensemble des concepts associés aux chemins [111] et [112] à savoir les concepts B et E.

Le ppcg des concepts A, F et H se restreint alors au ppcg de [111 112] avec [21 113] à savoir le concept C[11]. En revanche, D et G n'ont pas de ppcg. On remarque donc que le codage par numérotation précité contient au moins autant d'informations qu'une ontologie de concepts saturée puisqu'il permet, sans pas de chaînage avant et avec quelques calculs peu coûteux en temps de calcul, d'obtenir toutes les informations inhérentes aux concepts de l'ontologie.

Une description plus détaillée de l'étape de calcul d'exclusion mutuelle B entre tout concept considéré C_i de l'ontologie et un autre concept primitif ou défini de cette même ontologie et du treillis codé, tel que représenté en figure 2d, sera maintenant donnée ci-après, en référence aux figures 3a à 3c.

D'une manière plus particulière on indique que la prise en compte des contraintes d'exclusion, pour le codage de l'ontologie précitée, a été effectué pour pouvoir communiquer les contraintes d'exclusion précitées sous une forme saturée et compacte. On comprend, en particulier, que les contraintes d'exclusion ayant été pré calculées leur communication permet ensuite, à tout utilisateur équipé de ressources ordinaires, de prendre en compte ces contraintes d'exclusion afin de manipuler l'ontologie codée objet de la présente invention.

Dans ces conditions, l'étape B de calcul des exclusions mutuelles de la figure 2a, conforme à l'objet de la présente invention, est remarquable en ce qu'elle prend en compte des contraintes d'exclusion mutuelle qui sont implicites c'est-à-dire les exclusions entre concepts induites par les définitions de ces concepts et en particulier les concepts définis.

Exemple : avec la définition Homme : = être Vivant $\cap$ nom(Femme) on définit ainsi un homme comme étant un être humain qui n'est pas une femme. On obtient donc une exclusion mutuelle implicite entre homme et femme.

Ainsi, en référence à la figure 3a, l'étape B de calcul des exclusions mutuelles comporte, avantageusement, une sous-étape B_1 d'initialisation consistant en fait à engendrer toutes les contraintes d'exclusion entre concepts définis et primitifs, en s'assurant toutefois de ne garder que les contraintes placées sur les concepts les plus généraux.

Cette opération est réalisée à la sous-étape B₁ de la figure 3a par appel d'un algorithme d'initialisation dont le pseudo code désigné ALG₁ est donné dans le tableau T1 :

Tableau T1 – Algorithme d'initialisation

Pour chaque concept C1 de l'ontologie

Def1 = Définition de C1

Pour chaque concept C2 de l'ontologie

Def2 = Définition de C2

Conjonction = Def1 $\cap$ Def2

Si l'évaluation de la conjonction aboutit sur $\perp$, alors on ajoute la contrainte

$C1 \cap C2 \subseteq \perp$.

FinPour

FinPour

Grâce au procédé de codage par numérotation exécuté à l'étape A de la figure 2a, le calcul et la mémorisation des contraintes d'exclusion mutuelle peuvent alors être optimisés.

En effet, dans le cadre de la technique antérieure, il était nécessaire pour obtenir des résultats concrets et rapides de stocker pour chaque concept C_i la liste L exhaustive des concepts vis-à-vis desquels ce concept était mutuellement exclusif.

Dans le cadre de la mise en œuvre du procédé objet de la présente invention, et, en particulier, compte tenu du procédé de codage par numérotation précité, pour chaque concept considéré C_i de l'ontologie et du treillis codé représentant celle-ci, on ne stocke en fait qu'une liste notée L_{ex} des concepts les plus généraux, avec lesquels le concept considéré C_i est mutuellement exclusif.

Ainsi, tous les concepts C_i considérés dont l'un des chemins est le préfixe d'un des concepts de la liste des concepts les plus généraux L_{ex} sont mutuellement exclusifs du concept considéré C_i.

En outre, en ne stockant pour chaque concept considéré C_i que le concept le

plus général avec lequel il est mutuellement exclusif, on peut tester l'exclusion plus rapidement qu'en stockant tous les concepts.

En effet, dans le cadre de la technique antérieure, tester si les concepts C1 C2 sont mutuellement exclusifs consistait à parcourir la liste des concepts pour y retrouver le concept exclusif C2.

Conformément au procédé de codage par numérotation objet de l'invention, il n'y a toujours qu'un seul parcours de la liste à effectuer pour rechercher la présence d'un des préfixes des chemins du concept C2. Le gain en coût de calcul réside dans la taille de la liste à parcourir, puisque la recherche d'un préfixe est aussi peu coûteuse que la recherche d'une égalité sur une chaîne de caractères entière. Enfin, le calcul et la mémorisation sélective ne seraient pas possibles sans le codage par numérotation des identifiants de chaque concept, puisque le test serait alors beaucoup plus coûteux et consisterait à vérifier l'appartenance du concept mutuellement exclusif C2 dans la liste des descendants de chacun des concepts de la liste des concepts les plus généraux L_{ex} .

Exemple : en référence au treillis représenté en figure 2d et pour le concept H[23 113] mutuellement exclusif avec, d'une part, le concept B[111] et d'autre part le concept E[112], la saturation et la propagation des deux contraintes d'exclusion doit être effectuée sur des descendants du concept H de B et de E.

L'opération de constitution de la liste des concepts le plus généraux L_{ex} est illustrée à titre d'exemple en liaison avec la figure 3b.

Pour deux concepts quelconques, un concept C_i et un concept C_j avec $j \neq i$, les opérations sont exécutées en une sous-étape B_{11} de la sous-étape B_1 d'initialisation en prenant en considération les chemins CH_i respectivement CH_j associés aux concepts précités.

Une étape de test B_{12} est alors appelée, laquelle consiste à vérifier si le chemin CH_j préfixe le chemin CH_i selon la relation :

$CH_j \text{ PREFIX } CH_i ?$

Sur réponse négative au test B_{12} , une sous-étape B_{13} est appelée, laquelle consiste à choisir le concept j suivant dont les chemins sont des chemins de concepts

généralisants, c'est-à-dire correspondant à une profondeur moindre que celle du concept considéré C_i .

Au contraire, sur réponse positive à la sous-étape B_{12} une sous-étape B_{14} est appelée, laquelle consiste à mémoriser le concept CH_j dont le chemin CH_j est inférieur au concept CH_i et préfixe ce dernier.

L'étape de mémorisation est notée à l'étape B_{14} :

$$L_{ex}=[CEG_i[C_j]]|CH_j < CH_i.$$

Par la relation symbolique précédente, on indique que l'on mémorise en fait le chemin CH_j du concept C_j qui constitue un généralisant du concept considéré CH_i et qui bien entendu préfixe ce dernier.

Suite à l'étape B_{14} est alors appelée une étape B_{15} consistant à soumettre à un test d'égalité l'identité du concept C_j au concept universel. Cette opération peut être effectuée par le calcul d'un chemin vide du concept C_j vis-à-vis du concept universel.

Sur réponse négative à la sous-étape du test B_{15} un retour est effectué à la sous-étape B_{13} laquelle consiste à appeler le concept généralisant suivant, noté j suivant, pour poursuite du processus.

Au contraire, sur réponse positive à la sous-étape de test B_{15} , une étape de fin d'initialisation est effectuée, tous les concepts généralisants mutuellement exclusifs avec le concept considéré C_i ayant été soumis au test d'existence de préfixe de chemins.

À la fin de l'initialisation, c'est-à-dire en réponse positive à la sous-étape de test B_{15} , on dispose de la liste des concepts les plus généraux avec lesquels le concept considéré C_i est mutuellement exclusif. Cette liste est notée :

$$CE_i=L_{ex}[CEG_i[C_j]]|CH_j < CH_i.$$

L'étape d'initialisation de la figure 3a peut alors être suivie d'une étape B_2 de propagation de l'exclusion.

L'étape de propagation peut alors correspondre à un processus par appel d'un algorithme ALG_2 dont le pseudo code est donné au tableau T2.

Tableau T2 – Algorithme de propagation

Initialiser les listes de concepts mutuellement exclusifs à vide.

Pour chaque contrainte d'exclusion de l'ontologie entre C1 et C2

N1 = premier chemin de C1

N2 = premier chemin de C2

Pour chaque concept de l'ontologie dont un des chemins de son identifiant possède pour préfixe N1

Associer à ce concept, si ce n'est déjà fait, le chemin N2 comme étant mutuellement exclusif.

FinPour

Pour chaque concept de l'ontologie dont un des chemins de son identifiant possède pour préfixe N2

Associer à ce concept, si ce n'est déjà fait, le chemin N1 comme étant mutuellement exclusif.

FinPour

FinPour

Exemple : pour C1=H et C2=B en liaison avec le treillis codé tel que représenté en figure 2d et pour N1=[21] et N2=[111] on associe en tant que concept mutuellement exclusif H[21] à B[111] mais également aux descendants de ce dernier, soit le concept A[1111] et le concept F[1112].

D'autre part, on associe en tant que concepts mutuellement exclusifs le chemin [111] au concept H[21] et au concept I[211]. La prise en compte de la contrainte d'exclusion entre les concepts H et E pose N1=[21] et N2=[112]. Ceci permet d'ajouter le chemin [21] au concept E[112] sans modifier ni le concept A ni le concept F mais en ajoutant le chemin [112] au concept I[211].

Un mode opératoire correspondant opérant à partir de tout concept C_j mutuellement exclusif avec un concept C_i considéré et appartenant à la liste des concepts les plus généraux, avec lesquels ce concept considéré est mutuellement exclusif, est représenté à titre purement illustratif en figure 3c.

À partir de la liste précitée L_{ex} on calcule en une sous-étape B₂₁ pour le

concept C_j considéré l'existence de chemins CH_{jx} formés par le chemin CH_j du concept C_j mutuellement exclusif précité auquel est ajouté un suffixe CH_x , tel que le suffixe CH_x ne soit pas vide vis-à-vis des concepts spécialisants du concept C_j considéré.

Lorsqu'en réponse positive à la sous-étape de test B_{21} le chemin CH_x suffixe n'est pas vide, alors une étape B_{22} permet de réactualiser la liste des concepts généralisants à partir de tous les concepts spécialisants mutuellement exclusifs avec le concept considéré C_i . Cette liste des concepts notée $\{CES_{jx}\}$ est ajoutée à la liste CEG_i précitée.

Au contraire, sur réponse négative à la sous-étape de test B_{21} la propagation est arrêtée et la liste des concepts mutuellement exclusifs est alors formée par la liste des concepts les plus généraux avec lesquels le concept considéré C_i est exclusif, cette liste ayant été mise à jour par tous les concepts spécialisants des concepts généraux mutuellement exclusifs avec le concept C_i .

Une description plus détaillée de l'étape C de mémorisation permettant la mise en œuvre du procédé, objet de la présente invention, tel qu'illustré en figure 2a sera maintenant donnée ci-après.

Alors que le codage par numérotation du treillis, tel qu'illustré précédemment et décrit relativement à l'étape A de la figure 2a précitée, permet d'effectuer un certain nombre de calculs conduits de manière optimale et peu coûteuse, en tout état de cause au moins aussi rapidement que le permet une ontologie saturée, communiquer un tel treillis codé à une application distante et, en particulier, une hiérarchie de concepts décrits en logique de description, peut alors se résumer à transmettre à cette application une liste de concepts associée à leur identifiant.

Toutes les informations nécessaires à la classification, c'est-à-dire au positionnement des concepts des uns par rapport aux autres, et au raisonnement, calcul de plus petit commun généralisant ppcg ou autre, sur des concepts sont contenus dans la liste d'identifiants précitée. Il n'est alors plus nécessaire d'effectuer des tests de subsomption complexes sur les logiques de description, ni d'effectuer un

parcours de la hiérarchie de concepts à concepts, dans le but de positionner les concepts primitifs ou définis au sein de la hiérarchie précitée.

En outre, et dans le but d'achever une communication optimale d'une hiérarchie de concepts décrite en logique de description, conformément à l'objet de la présente invention, sous forme d'une liste de concepts associés à leur identifiant, le procédé, objet de l'invention, et, en particulier, à l'étape C de mémorisation représentée en figure 2a consiste avantageusement à établir la structure de données précitée à partir d'un ensemble de règles de définition, chaque règle de définition comportant une première partie mise en relation avec une deuxième partie constituant une définition de la première partie. En outre, une pluralité de mots clés spécifiques à valeur syntaxique est mise en œuvre, ceci afin de permettre d'organiser syntaxiquement les blocs de la structure de données. Enfin, une pluralité de symboles, chaque symbole représentant une valeur sémantique attribuée à un mot clé spécifique, est en outre avantageusement mise en œuvre pour structurer la structure de données précitée selon un format spécifique.

La structure de données subdivisée en blocs contient alors des blocs terminaux constitués chacun par un mot clé et au moins un symbole. L'ensemble de règles de définition, la pluralité de mots clés et la pluralité de symboles sont constitués en un langage de transmission et de communication de la structure de données dont les règles d'écriture constituent les règles d'écriture syntaxique. Les règles d'écriture syntaxique précitées peuvent alors être définies comme une grammaire, laquelle décrit le langage de transmission d'un treillis complet de concepts.

La grammaire, ou l'ensemble des règles d'écriture syntaxique, précitée est alors composée des règles de définitions précitées de la forme :

Partie gauche : = partie droite qui permettent d'écrire des phrases ou des informations, en remplaçant les parties gauches par leur définition qui ne sont autres que les parties droites.

Les mots-clés utilisés, précédemment mentionnés, sont alors utilisés pour donner un sens au bloc utilisé dans la grammaire, c'est-à-dire finalement afin de

permettre d'organiser syntaxiquement les blocs de la structure de données précitée.

Enfin, la pluralité de symboles inclut au moins la barre verticale, les crochets, les symboles <, >, les signes +, ? et le signe * par exemple.

On indique que les symboles qui entourent les mots-clés précités permettent de renseigner sur la nature effective des blocs.

Le document final obtenu à partir des règles d'écriture syntaxique, c'est-à-dire de la grammaire précitée, ne contient en définitive que des blocs appelés blocs terminaux.

Les symboles mis en œuvre pour établir les règles d'écriture syntaxique ou grammaire précitée s'interprètent de la manière suivante :

- l'écriture d'un bloc déterminé, **bloc 1**, en partie gauche et l'écriture d'une suite ordonnée de **blocs** en partie droite d'une règle de définition, permet d'allouer à l'écriture précitée une fonction de remplacement du bloc déterminé **bloc 1** par la suite de blocs précités. Le bloc **bloc 1** ne doit pas être un bloc terminal. En outre, une virgule peut être insérée entre chacun des **blocs** en les indiquant dans l'ordre dans lequel ces derniers doivent apparaître ; Cette virgule doit apparaître à sa place dans le document final, elle est un bloc terminal ;
- l'écriture d'un premier symbole | entre deux blocs d'une pluralité de blocs permet d'allouer à l'écriture du symbole | précité, une fonction de choix entre plusieurs blocs situés à gauche du premier symbole pour remplacer la pluralité de blocs située à droite du premier symbole précité ;
- l'écriture d'une pluralité de blocs entre crochets, deuxième symbole, permet d'allouer à l'écriture précitée une fonction de remplacement de chaque bloc de cette pluralité de blocs par une chaîne de caractères, dénomination d'instance. Les chaînes de caractères précitées forment chacune un bloc terminal présent dans la structure de données finale ;
- l'écriture d'une pluralité de blocs entre le symbole < et le symbole >, constituant un troisième symbole, permet d'allouer à l'écriture précitée une fonction de remplacement de chaque bloc par sa définition à partir d'une règle de définition. Dans cette situation, les blocs ne constituent pas des blocs terminaux ;

– l'écriture d'un quatrième symbole, le signe *, associé à un bloc permet d'allouer à l'écriture du quatrième symbole précité une fonction de multiplication du nombre d'instances affectées au bloc. Les instances peuvent être différentes ou non. Une virgule apparaît entre chaque instance dans le document final ;

– l'écriture d'un cinquième symbole, le signe + associé à un bloc, permet d'allouer à l'écriture précitée une multiplication du nombre d'instances affectées au bloc sous contrainte de présence d'au moins une fois dans le document final constitutif de la structure de données ;

– l'écriture d'un sixième symbole, le signe ? associé à un bloc, permet d'allouer à l'écriture précitée la définition de champs facultatifs sous contrainte de présence une seule fois ou d'absence du bloc dans le document final constitué par la structure de données ;

– les mots en gras et lettre capitales doivent apparaître tels quels dans le document final constitutif de la structure de données ; ce sont donc des blocs terminaux.

Le langage de transmission précité et de communication de la structure de données codée est alors l'un des langages choisis dans le groupe des langages xml ou Owl.

Une description complète d'une DTD pour Document Type Definition en anglais, est donnée dans le tableau T5 ci-après. Elle satisfait aux règles d'écriture syntaxique et à la grammaire générale d'une liste de concepts transmise, ces concepts ayant été codés conformément au procédé objet de la présente invention dont la description complète de la grammaire est donnée dans les tableaux T3 et T4 ci-après..

Tableau T3

<expression de logique de description> Ce bloc est présenté dans le tableau T4 ci-après dans ce document.

<liste de concepts> ::= [nom du domaine], <concept>+

<concept> ::= C [nom de concept], [nom de synonymes]*, <identifiant>, <contraintesExclusion>?,

$\langle \text{identifiant} \rangle ::= \text{ID} \langle \text{chemin} \rangle +$
 $\langle \text{chemin} \rangle ::= \text{CH} [\text{entier}] +$
 $\langle \text{contraintesExclusion} \rangle ::= \text{CE} [\text{nom de concept}] +$

Un exemple d'expression des contraintes d'exclusion est donné ci-après pour les concepts A et F représentés sur le treillis codé de la figure 2d, ces concepts étant mutuellement exclusifs :

$C A, ID CH 1,1,1,1, CH 1,1,2,1, CEF C F, IDCH 1,1,1,2, CH 1,1,2,2 CE A.$

Dans la relation précédente :

- C désigne l'introduction d'un champ de concept A ou F ;
- ID CH désigne l'introduction d'un champ relatif à un chemin contenu dans un identifiant ID ;
- CE désigne l'introduction d'un champ de concept exclusif vis-à-vis du concept introduit dans le champ C.

En outre, conformément à l'objet de la présente invention, un concept C1 est associé au concept C2 les plus généraux avec lesquels il est mutuellement exclusif. En outre, on sait que tout concept C3 descendant de C2, c'est-à-dire fils du concept C2, est également mutuellement exclusif avec le concept C1.

On dispose donc d'une notation saturée et condensée des contraintes d'exclusion.

Les contraintes d'exclusion s'expriment alors sous la forme :

$\langle \text{contrainteExclusion} \rangle ::= \text{CE} [\text{nom de concept}]^+$

À titre d'exemple, pour le concept EnOcéanie, il est possible d'indiquer que ce concept est mutuellement exclusif avec le concept EnAfrique et EnEurope selon la relation $C \text{ EnOcéanie } CE \text{ EnAfrique, EnEurope}.$

Les règles d'écriture syntaxique formant la grammaire précitée permettant la mise en œuvre du procédé de codage objet de la présente invention ne sont pas limitées à la transmission des concepts primitifs ou définis codés par numérotation, tels que précédemment mentionnés, mais permettent de manière particulièrement avantageuse une application d'un codage aux informations non liées au concept

incluant des rôles, des prédicats ordinaires et des faits.

En particulier, le procédé de codage objet de l'invention, consiste alors à coder chaque rôle selon un prédicat binaire incluant au moins un en-tête associé à un nom de rôle, un premier argument de type concept déterminé par le rôle et un deuxième argument de type concept déterminant dans le rôle.

Ainsi, les règles d'écriture syntaxique précitées permettant la mise en œuvre du procédé de codage objet de l'invention, permettent de spécifier les types de chacun des arguments d'un rôle. Cette information est directement et complètement exploitable et n'existe que sous forme saturée.

Un rôle est alors codé sous la forme :

$\langle \text{rôle} \rangle ::= R \text{ [nom du rôle], [concept qui type l'argument 1], [concept qui type l'argument 2]}.$

À titre d'exemple :

Situé dans localise un **lieu de résidence** dans un **lieu géographique** et le rôle s'écrit alors :

R situé dans, lieu de résidence, lieu géographique.

Le procédé, objet de l'invention, consiste en outre à coder chaque prédicat ordinaire selon au moins un en-tête associé à un nom de prédicat, un premier argument de type argument et un deuxième argument de liste de définition, chaque argument présentant en outre une valeur sémantique.

En effet, un prédicat ordinaire s'identifie par son nom et par le nombre d'arguments que ce dernier accepte. Outre ces informations premières, des informations supplémentaires en vue d'un traitement coopératif peuvent être ajoutées. En particulier, chaque argument d'un prédicat ordinaire possède implicitement un type connu par l'expert en ontologie qui définit l'ontologie en tant que telle.

À titre d'exemple, le prédicat ordinaire vol AR accepte sept arguments, le premier est de type Vol, les deux suivants sont de type aéroport et les quatre derniers sont de type date, pour exprimer les dates de départ et d'arrivée des vols aller-retour.

De plus, chacun des arguments précités correspond à une information attendue qui est à la fois plus précise que le type et indépendante du type précité.

Cette information n'est autre que la sémantique de l'argument.

Ces informations relatives au type et à la sémantique des arguments sont directement et complètement exploitables. En définitive, les informations précitées n'existent que sous forme saturée. En revanche, un prédicat ordinaire peut être associé à zéro, une ou plusieurs définitions qui, elles, peuvent avoir une forme saturée.

L'expression codée d'un prédicat ordinaire conformément à la règle d'écriture syntaxique permettant la mise en œuvre du procédé de codage objet de l'invention, est alors de la forme :

$\langle \text{prédicat ordinaire} \rangle ::= \text{PO} [\text{nom du prédicat}], \langle \text{type} \rangle^+, \langle \text{définition} \rangle^*.$

À titre d'exemple, on reprend le vol AR vu précédemment sans définition.

L'expression du prédicat ordinaire précité devient :

PO vol AR, **T** vol, vol, **T** aéroport de départ, Aéroport, **T** aéroport d'arrivée, Aéroport, **T** date de départ aller, Date, **T** date d'arrivée aller, Date, **T** date de départ retour, Date, **T** date d'arrivée retour, Date.

$\langle \text{type} \rangle ::= \text{T} [\text{sémantique}], [\text{Type argument}] \text{ ex: } \text{T date d'arrivée retour, Date}.$

Les définitions d'un prédicat sont donc écrites sous forme de règles de définition dans l'ontologie. L'atome tête de la règle est déduit de la liste des atomes du corps. On dit qu'un prédicat ordinaire qui apparaît en tête de règle accepte parmi ses définitions le corps de la règle.

Deux politiques peuvent être envisagées pour ce genre d'information qui contient beaucoup d'informations implicites induites par le mécanisme de chaînage avant, c'est-à-dire de saturation d'une ontologie.

En premier lieu, on peut transmettre une version saturée des définitions avec toutes les définitions déductibles de l'ensemble des règles de l'ontologie. L'inconvénient de cette première possibilité est d'obtenir un objet d'un volume considérable par rapport à la taille initiale des règles dans l'ontologie.

En deuxième lieu, et selon un choix préféré, on ne transmet que les définitions sans redondance en tenant compte des propriétés du mécanisme de chaînage avant qui ne sont pas très complexes à mettre en œuvre et qui, par ailleurs,

sont beaucoup moins complexes que les calculs d'un raisonneur sur une logique de description.

Enfin, le procédé de codage objet de l'invention, consiste à coder chaque fait selon un en-tête associé à un atome dont au moins un des arguments est une constante. En effet, un fait est un atome dont au moins un des arguments est une constante.

Le codage d'un fait satisfaisant aux règles d'écriture syntaxique ou grammaire précitée est alors de la forme :

<fait> ::= FAIT <atome>.

À titre d'exemple un fait peut être alors codé selon la syntaxe :

FAIT A tarifAss"714"Sydney", "1230 Euros".

La définition de codage de chaque prédicat ordinaire est alors codée par un en-tête associé à une pluralité d'atomes issus d'une règle de définition.

En outre, et selon un aspect remarquable du procédé de codage objet de l'invention, à chaque règle de définition est avantageusement associé un numéro d'ordre numérique de lecture de règles, lequel permet d'ordonner tout sous-ensemble de règles de définition codées pour optimisation d'une exécution des règles de définition dans la structure de données.

Compte tenu de l'introduction d'un numéro d'ordre de lecture de règles précédemment citée, le codage d'une définition conformément aux règles d'écriture syntaxique ou grammaire permettant la mise en œuvre du procédé objet de la présente invention est alors le suivant :

<définition> ::= NUM [entier] DEF <atome>⁺.

À titre d'exemple, une règle précisant qu'un vol est à un tarif associé économique, indique que le vol est un vol économique.

Le prédicat ordinaire s'écrit alors :

PO Vol Eco, T vol, Vol T tarif, TarifEc

NUM 3 DEF A VolEco,x,p, A Vol,x, A,TarifAss,x,p, A TarifEc,p et

<atome> ::= (A [nom prédicat],[nom d'argument]⁺).

Exemple : ATarifAss,x,p.

Ainsi que mentionné précédemment dans la description, les règles de définition de l'ontologie peuvent permettre de déduire des informations par saturation, c'est-à-dire par chaînage avant. Lors d'un tel traitement des définitions de prédicat, il peut être particulièrement intéressant d'imposer l'ordre de lecture des règles de définition pour minimiser le nombre de lectures effectuées.

À titre d'exemple non limitatif, pour trois règles de définition telles que :

- 1 $P(x,y) \cap P2(tz) \rightarrow P3(t,x,z)$;
- 2 $P(x,y) \rightarrow P2(x,y)$;
- 3 $A(x,y,z) \rightarrow P(x,y)$;

le traitement dans l'ordre précité, et en partant d'un fait $A(a,b,c)$, la première lecture des trois règles permet de déduire $P(a,b)$ grâce à la règle 3.

Un deuxième passage sur les trois règles permet de déduire $P2(a,b)$ grâce à la règle 2.

Un troisième passage permet de déduire $P3(a,a,b)$ grâce à la règle 1.

Le troisième et dernier passage permet de terminer la saturation des règles puisqu'il n'aboutit à aucune déduction.

Le mode de traitement classique précité implique donc $3^2+3=12$ règles lues, soit pour n règles, n^2+n lectures.

En codant un numéro d'ordre de lecture associée à chaque définition et en imposant un traitement des règles, c'est-à-dire une lecture dans l'ordre inverse, 3, 2, 1 une seule lecture de toutes les règles permet de tout déduire, ce qui implique un nombre de lectures égal à 6 soit $2n$ lectures pour n règles.

L'introduction d'un numéro d'ordre de lecture NUM permet donc d'ordonner n'importe quel sous-ensemble de règles de l'ontologie pour une optimisation du traitement des règles de définitions précitées. Notons que dans le contexte de règles écrites en Carin-ALN, l'entier est toujours le même pour toutes les définitions d'un prédicat ordinaire donné. Nous pouvons donc également adopter les règles de grammaire suivantes pour les prédicats ordinaires :

$\langle \text{prédicat ordinaire} \rangle ::= \text{PO} [\text{nom prédicat}], \langle \text{type} \rangle^+, (\text{NUM} [\text{entier}], \langle \text{définition} \rangle^+)?$.
 $\langle \text{définition} \rangle ::= \text{DEF} \langle \text{atome} \rangle^+$.

Les règles d'écriture syntaxique permettant la mise en œuvre du procédé de codage, objet de la présente invention, sont en outre adaptées de manière à permettre la possibilité de décrire des expressions de logique de description.

En effet, alors que le codage des concepts primitifs et/ou définis par numérotation permet de privilégier très largement la communication de concepts nommés à celle d'expression de logique de description relative aux concepts décrits dans l'ontologie, une ontologie classique, c'est-à-dire une ontologie définie par un expert en ontologie dans un domaine quelconque, possède généralement quelques expressions de logique de description dans les règles de définition ou pour définir des concepts.

La mise en œuvre du procédé de codage, objet de la présente invention, permet de définir une structure de données dans laquelle les concepts définis sont restreints à leur identifiant qui n'ont plus que leur position au sein du treillis de concepts codé pour seule sémantique. La définition à laquelle ils sont associés n'est plus nécessaire pour manipuler l'ontologie et la consulter ou la compléter avec de nouveaux concepts. Bien entendu, dans le cas où l'ontologie précitée doit être complétée, il est alors possible d'exécuter cette opération et puis d'appliquer à nouveau un codage par numérotation des concepts complétés ainsi que décrit précédemment dans la description. En effet, le processus de codage par numérotation doit s'appliquer sur une ontologie complète.

Dans ces conditions, tout expert créateur d'ontologie est invité à associer à chaque expression apparaissant dans une définition un nom de concept défini auquel l'expression est équivalente. Il doit soit associer un nom à cette expression de logique de description, soit laisser un nom fictif de concepts engendré automatiquement et positionné dans l'ontologie. On comprend, en particulier, qu'en raison du fait que chaque concept dans le treillis de concepts codé n'ayant pas de valeur sémantique, seule la position de ce dernier dans l'ontologie est déterminante.

Ainsi, chaque expression de logique de description porteuse de sens est donc nommée dans l'ontologie avant le processus de codage par numérotation.

Toutefois, dans le cas où un expert créateur d'ontologie le souhaite, les

règles d'écriture syntaxique ou grammaire permettant la mise en œuvre du procédé de codage objet de la présente invention, et le langage de transmission et de communication de la structure de données constitutif de ces règles d'écriture syntaxique comportent en outre un ensemble d'opérateurs logiques d'expression de logique de description permettant d'introduire les éléments ou expressions de logique de description au gré de l'expert créateur d'ontologie, bien que ces éléments ne soient pas a priori nécessaires.

L'ensemble d'opérateurs logiques d'expression de logique de description précitée inclut au moins :

- un premier opérateur logique <and> permettant la mise en relation logique d'intersections (ET) entre une première et une deuxième expression de logique de description par l'intermédiaire d'un en-tête "AND" ;
- un deuxième opérateur logique <forall> permettant la mise en relation d'un nom de rôle et d'une expression logique de description par l'intermédiaire d'un en-tête "FA" ;
- un troisième opérateur logique <at least> permettant la mise en relation d'un entier minorant et d'un nom de rôle par l'intermédiaire d'un en-tête "AL" ;
- un quatrième opérateur logique <atmost> permettant la mise en relation d'un entier majorant et d'un nom de rôle par l'intermédiaire d'un en-tête "AM" ;
- un cinquième opérateur logique <neg> permettant la négation ou exclusion d'un élément de base tel qu'un concept par l'intermédiaire d'un en-tête "NEG".

Compte tenu des règles d'écriture syntaxique permettant la mise en œuvre du procédé de codage objet de la présente invention, les opérateurs logiques d'expression de logique de description de l'ensemble d'opérateurs logiques s'écrivent selon le tableau T4 :

Tableau T4

<expression de logique de description> ::= (<and> | <forall> | <at least> | <atmost> | <neg> | <base>)

<and> ::= **AND** (<expression de logique de description>, <expression de logique de description>+)

<forall> ::= **FA** [nom de rôle] <expression de logique de description>

<at least> ::= **AL** [entier] [nom de rôle]

<atmost> ::= **AM** [entier] [nom de rôle]

<neg> ::= **NEG** <base>

<base> ::= **C** [nom de concept]

Une description plus détaillée d'une structure de données subdivisée en blocs codée et représentative d'une ontologie comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini, cette hiérarchie de concepts étant représentée par un treillis et la structure de données subdivisée en blocs codée étant obtenue par la mise en œuvre du procédé objet de l'invention tel que précédemment décrit, sera maintenant donnée ci-après.

La structure de données subdivisée en blocs, objet de l'invention, est représentée sous forme de DTD complète et qui correspond à la mise en œuvre des règles d'écriture syntaxique ou grammaire présentée précédemment dans la description relativement à un fichier désigné *ontology.dtd*. La structure de données subdivisée en blocs, objet de l'invention, ou DTD complète est donnée dans le tableau T5 ci-après :

Tableau T5

1	< ELEMENT ontology (concept+, role*, ordinaryPredicate*, fact*)>	< ATTLIST ontology domain CDATA #REQUIRED>
2	< ELEMENT concept (synonym*, id, cExcl*, (and forall atleast almost neg base)?)>	< ATTLIST concept name CDATA #REQUIRED>
3	< ELEMENT synonym EMPTY>	< ATTLIST synonym name CDATA #REQUIRED>
4	< ELEMENT id (path+)>	
5	< ELEMENT path (integer+)>	< ATTLIST integer val CDATA #REQUIRED>
5	< ELEMENT integer EMPTY>	< ATTLIST cExcl name CDATA #REQUIRED>
6	< ELEMENT cExcl EMPTY >	< ATTLIST role name CDATA #REQUIRED>
7	< ELEMENT role (argDescr, argDescr)>	< ATTLIST ordinaryPredicate name CDATA #REQUIRED>
8	< ELEMENT ordinaryPredicate (argDescr*, (ruleNumberOfPriority, def+)?)>	< ATTLIST argDescr type CDATA #REQUIRED>
10	< ELEMENT argDescr EMPTY >	
	< ATTLIST argDescr semantic CDATA #REQUIRED>	< ATTLIST ruleNumberOfPriority val CDATA #REQUIRED>
9	< ELEMENT ruleNumberOfPriority EMPTY >	
10	< ELEMENT def (atom, atom+)>	< ATTLIST atom predicateName CDATA #REQUIRED>
15	< ELEMENT atom (argument+)>	< ATTLIST argument value CDATA #REQUIRED>
11	< ELEMENT argument EMPTY >	
11	< ELEMENT fact (atom)>	
12	< ELEMENT and ((and forAll atLeast atMost neg base), (and forAll atLeast atMost neg base)+)>	< ATTLIST forAll roleName CDATA #REQUIRED>
	< ELEMENT forAll (and forAll atLeast atMost neg base)>	< ATTLIST atLeast roleName CDATA #REQUIRED>
20	< ELEMENT atLeast EMPTY>	< ATTLIST atMost roleName CDATA #REQUIRED>
	< ELEMENT atMost EMPTY>	
	< ATTLIST atMost integer CDATA #REQUIRED>	
	< ELEMENT neg (base)>	
	< ELEMENT base EMPTY>	< ATTLIST base conceptName CDATA #REQUIRED>

Dans le tableau T5 précité, les numéros de lignes indiqués désignent des lignes de déclaration spécifiques remarquables de la mise en œuvre des règles d'écriture syntaxique, permettant la mise en œuvre du procédé de codage objet de l'invention.

En particulier, la structure de données subdivisée en blocs représentée au tableau T5 est structurée à partir d'une déclaration constituée d'une pluralité de lignes de déclaration formée par des blocs et incluant :

- une ligne 1 de déclaration de l'ontologie déclarant outre un nom de domaine de l'ontologie, une pluralité de concepts, de rôles et de prédicats ordinaires et de faits ;
- une ligne 2 de déclaration de concepts déclarant outre un nom de concept, une pluralité de synonymes, un identifiant, une pluralité de concepts mutuellement exclusifs vis-à-vis du concept déclaré et une pluralité d'opérateurs logiques d'expression de logique de description facultatif ;
- une ligne 3 de déclaration de synonymes déclarant un nom de synonyme ;
- une ligne 4 de déclaration d'identifiant, c'est-à-dire d'identifiant ID_i déclaré comme une pluralité d'entiers, ainsi que mentionné précédemment dans la description ;
- une ligne 5 de déclaration de valeur d'entier ;
- une ligne 6 de déclaration d'une liste de concepts mutuellement exclusifs d'un concept déclaré.

On indique en référence au tableau T3 introduit précédemment dans la description que le tableau T5 et en particulier les lignes 1 à 6 précédemment décrites, correspondent aux lignes de déclaration effective conformément au langage de communication d'ontologie et aux règles d'écriture syntaxique de cette dernière permettant la mise en œuvre du procédé de codage objet de la présente invention.

En outre, la structure de données subdivisée en blocs et codée, objet de l'invention, lorsque l'ontologie comprend en outre des informations non liées aux concepts incluant des rôles, des prédicats ordinaires et des faits est structurée à partir d'une déclaration constituée en outre de lignes de déclaration formées par des blocs, incluant les lignes suivantes, en particulier :

- une ligne 7 de déclaration de rôle déclarant outre un nom de rôle, un argument discriminé par le rôle et un argument discriminant dans le rôle à chaque argument étant attribué une valeur sémantique par une déclaration de type d'argument ;
- une ligne 8 de déclaration de prédicats ordinaires déclarant outre un nom de prédicat ordinaire une pluralité facultative de type d'arguments associé à un numéro d'ordre de lecture de règles et une pluralité de définitions facultatives ;
- une ligne 9 de déclaration de définitions déclarant une pluralité d'atomes ;
- une ligne 10 de déclaration d'atomes déclarant outre un nom de prédicat d'atomes une pluralité d'arguments ;
- une ligne 11 de déclaration de faits déclarant un fait comme un atome ;
- une ligne 12 et suivantes de déclaration d'opérateurs logiques d'expression de logique de description.

En particulier, la ligne de déclaration d'opérateurs logiques d'expression de logique de description précitée déclare au moins l'écriture d'un bloc déterminé en partie gauche et l'écriture d'une suite ordonnée de blocs en partie droite d'une règle de définition, à cette écriture étant allouée une fonction de remplacement du bloc déterminé par la suite de blocs, l'écriture du premier symbole | l'écriture du deuxième symbole [pluralité de blocs], l'écriture du troisième symbole, pluralité de blocs entre les symboles < et >, l'écriture du quatrième symbole * associé à un bloc, l'écriture du cinquième symbole + associé à un bloc et l'écriture du sixième symbole ? au symbole précité et à l'écriture de ces derniers étant allouées des fonctions correspondantes telles que décrites précédemment dans la description.

On comprend ainsi que la déclaration constitutive de la structure de données subdivisée en blocs et formant la DTD complète, introduite au tableau T5 précité, est alors établie en langage de balisage structuré, ce langage pouvant correspondre par exemple à un langage xml ou OWL.

On comprend, en particulier, que la déclaration précitée est établie selon le langage de balisage structuré précité. En conséquence, cette déclaration est la structure de données.

La structure de données subdivisée en blocs objet de l'invention constituant la DTD introduite au tableau T5 précité est ainsi une structure xml enrichie prenant en compte d'autres connaissances, telles que des rôles, des prédicats ordinaires comportant notamment un numéro d'ordre de lecture. La définition d'un prédicat ordinaire est en fait constitué d'une suite d'atomes commençant par l'atome de tête de la règle, cet atome tête est construit à partir du nom du prédicat suivi d'une liste d'arguments variables ou constants.

Un exemple d'utilisation d'une structure de données établie en conformité avec les règles d'écriture syntaxique, telles que décrites au tableau T5 précité, sera maintenant donné en liaison avec le tableau T6.

Tableau T6

```
<ordinaryPredicate name="VolAR">
<argDescr type="Vol" semantic="vol"/>
<argDescr type="Aéroport" semantic="lieu de départ"/>
<argDescr type="Aéroport" semantic="lieu d'arrivée"/>
<argDescr type="Date" semantic="date de départ pour l'aller"/>
<argDescr type="Date" semantic="date d'arrivée pour l'aller "/>
<argDescr type="Date" semantic="date de départ pour le retour"/>
<argDescr type="Date" semantic="date d'arrivée pour le retour"/>
<ordinaryPredicate name="VolAR">
<ordinaryPredicate name="VolEco">
<argDescr type="Vol" semantic="vol"/>
<argDescr type="TarifEco" semantic="tarif"/>
```

```

<ruleNumberOfPriory val="1"/>
<def>
  <atom predicateName="VolEco"><argument value="x"/><argument value="p"/> </atom>
  <atom predicateName="ConceptGeneral"><argument value="x"/>
    <argument value="p"/><argument value="x"/> </atom>
  <atom predicateName="TarifAss"><argument value="x"/><argument value="p"/> </atom>
  <atom predicateName="TarifEco"><argument value="x"/></atom>
</def>
</ordinaryPredicate>
<fact><atom predicateName="Aéroport"><argument value="Paris-CDG"/></atom></fact>
<fact><atom predicateName="Vol"><argument value="714-Sydney"/></atom></fact>
Avec
<concept name="Vol">
  <id><path><integer val="4"/><integer val="1"/></path></id>
</concept>
<concept name="ConceptGenerel">
  <id><path><integer val="4"/><integer val="1"/></path></id>
  <and>
    <base ConceptName="Vol"/>
    <forall rolename="TarifAss"/><neg><base ConceptName="TarifLuxe"/></neg></forall>
  </and>
</concept>

```

L'utilisation de la structure de données précitée constitutive d'une DTD spécifique est optimisée par un traitement préalable des règles, afin de leur affecter un numéro d'ordre de lecture NUM encore désigné numéro de palier.

La numérotation des règles de définition précitées permet alors d'introduire les propriétés suivantes :

- un ensemble de règles de définition à saturer est toujours ordonné selon le numéro de palier croissant ;

– un numéro de palier pour une définition d'un prédicat ordinaire **PO** se calcule de manière récursive en fonction du numéro de palier des prédicats ordinaires présent dans les définitions de prédicats ordinaires **PO**.

Les prédicats qui n'ont pas de définition ou qui ne contiennent pas de prédicat ordinaire dans leur définition, sont associés au numéro de palier 1.

L'absence de cycle dans la terminologie assure une terminaison à l'algorithme et la présence d'au moins un prédicat de palier 1.

L'algorithme d'utilisation de la structure de données subdivisée en blocs codés représentative d'une ontologie s'initialise alors avec une liste L laquelle associe chaque prédicat de l'ontologie à une liste de prédicats : chaque prédicat ordinaire **PO** est associé au prédicat ordinaire **PO** qui apparaissent dans au moins une des définitions de prédicats ordinaires (il est nécessaire d'effectuer $n*(n/2)$ lectures des éléments de la liste de L pour initialiser une liste L2 et $n*(n/2)$ lectures sont au pire nécessaires pour les suppressions de prédicats ordinaires **PO**).

Le pseudo code d'initialisation par lecture successive des règles de définition en fonction des numéros de palier est donné ci-après selon le tableau T7 :

Tableau T7

NbPrédicatTraité ← 0

Palier ← 0

Tant Que NbPrédicatTraité < NbTotalPrédicats **faire**

 Palier ← Palier +1

 L2 ← tous les prédicats de L associés à une liste vide

Pour chaque prédicat PO de L2

 NbPrédicatTraité ← NbPrédicatTraité +1

 Affecter Palier comme numéro pour chaque règle du prédicat PO

 Supprimer PO de toutes les listes contenues dans L

Fin Pour

Fin Tant Que

Deux ensembles de règles de définition R1 et R2 peuvent rapidement fusionner par application de l'algorithme ci-après en n passages pour n règles. De nombreuses règles ont le même numéro d'ordre et appartiennent au même numéro de palier et n'ont donc pas à être ordonnées les unes par rapport aux autres. Ce mode opératoire permet d'avancer plus rapidement dans un paquet de règles. Le processus algorithmique correspondant est introduit au tableau T8 ci-après :

Tableau T8

R3 ← vide

Tant Que non vide (R1) et non vide (R2) **faire**

Num1 ← PremierNuméroPalier(R1)

Num2 ← PremierNuméroPalier(R2)

Si Num1 < Num2

Alors Tant Que Num1 == PremierNuméroPalier(R1) **faire** R3 ← R3 +
ÔterPremièreRègle(R1)

Sinon Tant Que Num2 == PremierNuméroPalier(R2) **faire** R3 ← R3 +
ÔterPremièreRègle(R1)

Fin Tant Que

Si Vide(R2) **alors** R3 ← R3 + R1 **sinon** R3 ← R3 + R2

La structure de données subdivisée en blocs codés, conformément à la mise en œuvre du procédé objet de l'invention, peut alors être mémorisée et/ou communiquée à différents acteurs, cette communication pouvant être exécutée par l'intermédiaire d'un fichier texte par exemple, par l'intermédiaire d'un réseau classique en utilisant un fichier xml.

Une description plus détaillée d'un système de codage et de représentation synthétique d'une ontologie partiellement saturée, conforme à l'objet de la présente invention, cette ontologie comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini représentée par un treillis et

des informations non liées aux concepts, sera maintenant donnée en liaison avec la figure 4a.

D'une manière générale, en référence à la figure précitée, on indique que le système de codage, objet de l'invention, comporte de manière classique des organes d'entrée/sortie I/O, une unité centrale de traitement CPU et une mémoire de travail RAM.

De manière remarquable, le système de codage objet de l'invention comporte un module M_0 d'acquisition de l'ontologie permettant de mémoriser le treillis et les informations non liées aux concepts. Ces éléments d'informations peuvent, ainsi que représenté sur la figure 4a, être fournis en ligne par un fournisseur d'ontologie, le fournisseur d'ontologie éditant alors des fichiers numériques correspondants contenant le treillis et les informations non liées au concept.

En outre, le système de codage objet de l'invention comporte un module M_1 de codage du treillis de concepts par attribution à chaque concept et au nœud du treillis associé à ce dernier d'un identifiant ID_i formé par au moins une suite de nombres entiers. Chaque suite de nombres entiers définit un chemin d'accès entre un concept considéré C_i et le concept universel T , par l'intermédiaire de concepts pères successifs. L'identifiant inclut ainsi toutes les suites de nombres entiers définissant chacune un chemin d'accès entre l'un des concepts pères et le concept universel T , auxquelles est ajouté un nombre entier représentatif du concept considéré.

On comprend, en particulier, que le module de codage M_1 est avantageusement réalisé sous forme d'un module logiciel mémorisé en mémoire morte par exemple, ou sur une unité de disque dur par exemple, lequel est alors appelé par l'unité centrale CPU pour exécution, dans la mémoire de travail par exemple, lors d'une mise en œuvre du système de codage objet de l'invention.

Selon un aspect remarquable du système de codage objet de l'invention, ce dernier comporte un module M_2 de calcul des exclusions mutuelles entre concepts pour chaque concept considéré C_i de l'ontologie.

On comprend, également, que le module de calcul M_1 peut être constitué par un module de programme de calcul mettant en œuvre le procédé, objet de l'invention, tel que décrit précédemment.

Enfin, le système de codage objet de l'invention, comporte une unité de mémorisation PM du treillis de concepts codé sous forme d'une structure de données subdivisée en blocs incluant au moins un nom de domaine et une liste comportant au moins un identifiant, formé par une liste d'au moins un chemin et une liste de concepts mutuellement exclusifs avec le concept primitif considéré.

L'unité de mémorisation PM peut avantageusement être constituée par une mémoire programmable non volatile ce qui permet par exemple, lors de l'utilisation successive du système de codage objet de l'invention, d'effectuer toute mise à jour de la structure de données mémorisée dans la mémoire programmable précitée, en particulier lors de l'adjonction de concepts définis dans le treillis codé pour une mise à jour de l'ensemble de l'ontologie, ainsi que décrit précédemment dans la description.

Un système de consultation d'une l'ontologie partiellement saturée, conforme à l'objet de la présente invention, cette ontologie étant communiquée sous forme de la structure de données subdivisée en blocs telle que représentée en figure 2d, c'est-à-dire le treillis codé, accompagnée des documents texte de type DTD précédemment décrits dans la description, sera maintenant donné en liaison avec la figure 4b.

Le système de consultation représenté sur la figure précitée, conforme à l'invention, comprend au moins des organes d'entrée/sortie I/O, une unité centrale de traitement et une mémoire de travail RAM.

Il comporte en outre, ainsi que représenté sur la figure précitée, une unité M'_0 d'acquisition d'une structure de données subdivisée en blocs et représentative de l'ontologie à consulter, cette dernière comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini représentée par un treillis, cette structure de données étant représentée, ainsi que mentionné précédemment, par le treillis codé représenté en figure 2d par numérotation et par le

document DTD, tel qu'introduit par exemple par le tableau T5 ou T6, un module M_3 interpréteur de requêtes de consultation, cette requête étant transmise par un terminal utilisateur demandeur de consultations d'une ontologie donnée.

Dans un mode de mise en œuvre préférentiel, pour une requête de consultation transmise par un terminal utilisateur portant sur un premier et un deuxième concept disjoints par exemple, tel que les concepts B et H de la figure 2d, ces concepts sont alors codés dans la structure de données subdivisée en blocs comme concepts mutuellement exclusifs. On comprend, en particulier, que les concepts précités font alors partie de la liste des concepts exclusifs, liste notée CE_i telle que décrite précédemment dans la description, notamment en liaison avec les figures 3b et 3c.

Dans cette situation, le module interpréteur M_3 exécute alors une lecture des concepts dans la liste des concepts mutuellement exclusifs.

Sur identification des premier et deuxième concepts dans la liste de concepts mutuellement exclusifs, le module interpréteur M_3 exécute alors une limitation du parcours de recherche sur identification du premier et du deuxième concept B et H dans la liste des concepts mutuellement exclusifs. La limitation est alors exécutée sur les seuls concepts spécialisants du premier concept, par exemple, compte tenu de l'exclusion mutuelle des concepts spécialisants du deuxième concept.

À titre d'exemple, en supposant que la requête R transmise demande des voyages en TGV partant de l'aéroport Charles de Gaulle à Paris, alors que les concepts voyage en train et voyage en avion sont disjoints, et donc mutuellement exclusifs, et que le module interpréteur M_3 dispose des données provenant d'une compagnie aérienne pour les vols et d'une compagnie ferroviaire pour les voyages en train au départ du même aéroport, le module interpréteur M_3 , sachant que les voyages en avion ne peuvent pas être des voyages en train, et qu'un voyage en TGV est un voyage en train, limite le parcours de façon à ne pas chercher de réponse dans les données de la compagnie aérienne, ce qui, bien entendu, présente un gain de traitement considérable.

Une description plus détaillée d'un système serveur d'accès en réseau IP à une ontologie partiellement saturée, conforme à l'objet de la présente invention sera maintenant donnée en liaison avec la figure 4c.

Pour assurer un accès totalement autonome à tout utilisateur internaute à partir d'un terminal classique, le système serveur, objet de l'invention, comprend au moins interconnectés, un portail fournisseur d'accès, désigné OAP sur la figure 4c, à ladite ontologie partiellement saturée, à partir d'une requête de consultation et, bien entendu, un système de consultation de ladite ontologie saturée tel que décrit précédemment en liaison avec la figure 4b. Sur la figure 4c, le système de consultation est désigné OP.

En particulier, le système serveur peut être configuré, ainsi que représenté sur la figure 4c, par un portail fournisseur d'accès et un système de consultation physiquement distincts interconnectés en réseau ou, au contraire, par des entités correspondantes interconnectées et intégrées en une seule et même entité physique.

En particulier, le portail fournisseur d'accès comporte au moins, outre un organe d'entrée/sortie I/O et une unité centrale de traitement CPU, un module M₄ de discrimination de concepts mutuellement exclusifs dans la requête transmise par le terminal utilisateur, par comparaison des chemins d'accès contenus dans l'identifiant de chacun des concepts contenus dans la requête, et un module de calcul M₅ d'une requête cohérente par appel aux concepts généralisants des concepts mutuellement exclusifs, à partir des chemins et des identifiants des concepts généralisants précités.

En reprenant l'exemple d'utilisation décrit en liaison avec la figure 4b, on suppose que les concepts mutuellement exclusifs et donc disjoints sont les concepts A et H voyage en train et voyage en avion.

On suppose en outre que la requête demande de manière erronée, par exemple pour un utilisateur peu averti, des voyages en TGV à bord d'un avion A-320.

Dans ces conditions, le portail fournisseur d'accès à l'ontologie OAP est en mesure d'informer l'utilisateur, avant toute consultation des données, que sa requête ne peut obtenir de réponse puisqu'elle contient une incohérence, les concepts voyage en train et voyage en avion étant disjoints.

Pour ce faire, le module M_5 de calcul d'une requête cohérente fait alors appel aux concepts généralisants des concepts mutuellement exclusifs et permet, après identification dans la liste des concepts mutuellement exclusifs tel que voyage en TGV et voyage en avion, d'établir une requête cohérente du type : Voyage en TGV à bord d'un engin de transport ferroviaire européen ou voyage à bord d'un avion A-320 dans l'une des compagnies aérienne disponible dans l'ontologie.

Le procédé et le système de codage et de représentation synthétique d'une ontologie partiellement saturée, la structure de données obtenue par la mise en œuvre du procédé et du système de codage précités et le serveur de structure de données correspondant incluant un système de consultation de l'ontologie, conformes à l'objet de la présente invention, permettent ainsi à n'importe quel utilisateur internaute de pouvoir utiliser la richesse d'une ontologie, sans avoir à mettre en œuvre un raisonneur ou des fonctions complexes pour saturer cette ontologie.

En effet, l'ontologie est directement transmise sous sa forme saturée, c'est-à-dire sous la forme de la structure de données satisfaisant aux règles d'écriture syntaxique du langage de communication, tel que décrit précédemment dans la description.

Le mode de transmission d'ontologie précitée permet à la fois un gain de temps après réception du document, c'est-à-dire de la DTD complète puisqu'il n'y a aucun calcul complexe ou coûteux à effectuer sur l'ontologie déjà saturée, puis pendant la transmission et la lecture du document, puisque le fichier saturé est dans un format de type texte compact et de taille relativement réduite.

En outre, un gain d'utilisation peut être mis en évidence dans la mesure où, les règles d'écriture syntaxique, c'est-à-dire la grammaire permettant le codage de l'ontologie précitée est simple et adaptable à de nombreux formats, ainsi que mentionné précédemment dans la description, comme le format xml ou Owl par exemple.

En outre, le stockage des informations correspondantes ne nécessite pas de structure de données lourde.

Le procédé et le système de codage, la structure de données obtenue par la mise en œuvre de ces derniers et le serveur de structure de données correspondant permettent d'effectuer ainsi, tous les raisonnements sur les concepts de l'ontologie avec un moindre coût en temps d'exécution.

Un tableau de compression suivant, valable pour une ontologie de n concepts, introduit sous forme de tableau T9, illustre totalement les gains obtenus par la mise en œuvre précitée.

Dans le tableau ci-après, on appelle chaîne, une chaîne de caractères comparable au type de programmation "string".

Ph est la profondeur de la hiérarchie, ce paramètre étant égal à la longueur maximale d'un chemin CH allant du concept universel au concept vide.

Un concept considéré C_i comporte par exemple n_s synonymes et il comporte donc n_s appellations différentes.

Tout concept considéré C_i comporte également n_g concepts généralisants et n_d concepts descendants ou spécialisants pères et fils inclus.

L'ensemble des paramètres précités vérifie la relation :

$$n_g < n \text{ et } n_d < n.$$

Dans les techniques antérieures, pour différencier les concepts pères respectivement les concepts fils des autres concepts généralisants respectivement des concepts spécialisant, il fallait stocker quatre listes de noms de concepts.

Grâce à la mise en œuvre du procédé de codage objet de la présente invention, et, en particulier du codage par numérotation du treillis de concepts, toutes ces informations sont contenues dans l'identifiant ID_i de chaque concept considéré C_i qui comprend au maximum une suite d'au plus Ph-1 entier que multiplie le nombre de chemins vers le concept universel T.

Certains chemins peuvent atteindre Ph-1 entiers pour un concept considéré C_i situé au-dessus du concept vide avec au plus, Ph-1 concept généralisants. De la même façon, un concept C_i considéré situé sous le concept généralisant T comporte au plus Ph-1 concept spécialisant.

Un concept C_i considéré est mutuellement exclusif avec nce concepts, ce nombre pouvant être très proche du nombre total n de concepts.

La mise en œuvre du procédé et du système de codage et de représentation synthétique d'une ontologie partiellement saturée de la structure de données obtenue correspondante et du serveur de structures de données objet de l'invention, permet de largement réduire la valeur de nce concepts à $nce2=(n/2)$ ainsi que décrit précédemment dans la description. Le nombre nce2 est très inférieur au nombre nce.

En ce qui concerne les règles de définition, celles-ci sont très légèrement comprimées de façon à n'être transmises que comme définition d'un prédicat ordinaire. Enfin, il n'existe pas de différence pour le stockage des rôles qui ne sont pas transmis sous forme saturée.

Tableau T9

Art antérieur		Avec l'invention
Objets pour 1 concept	Taille pour 1 concept	Taille pour 1 concept
Nom du concept	1 chaîne	1 chaîne
Noms des Synonymes du concept	ns chaînes	ns chaînes
Noms des pères et autres généralisants	2 x ng chaînes	} 1 seule suite d'au plus (Ph-1) entiers x nbchemins vers T
Noms des fils et autres généralisants	2 x ng chaînes	
Contraintes d'exclusion	1 x nce chaînes	1 x nce2 chaînes ($nce2 \ll nce$)

Enfin, le procédé, le système de codage et de représentation synthétique d'une ontologie partiellement saturée, la structure de données et le serveur de structure de données correspondants, objets de la présente invention, couvrent également :

— un produit de programme d'ordinateur mémorisé sur un support de mémorisation et exécutable par un ordinateur, remarquable en ce que, lors de l'exécution du produit de programme par un ordinateur ou par un système de codage dédié, décrit en liaison avec la figure 4a, celui-ci exécute le procédé de codage et de représentation synthétique d'une ontologie partiellement saturée, ainsi que décrit précédemment dans la description en référence aux figures 2a à 3c ;

— un produit de programme d'ordinateur mémorisé sur un support de mémorisation et exécutable sur un ordinateur, remarquable en ce que lors de l'exécution de ce produit de programme par un ordinateur ou par un système de consultation dédiée d'une ontologie partiellement saturée, tel que décrit en liaison avec la figure 4b, ce produit de programme exécute une interprétation de toute requête de consultation émise par un terminal, ainsi que décrit en liaison avec la figure 4b ;

— un produit de programme d'ordinateur mémorisé sur un support de mémorisation et exécutable par un ordinateur, remarquable en ce que lors de l'exécution de ce produit de programme par un ordinateur ou par un système serveur dédié comportant un portail d'accès à une ontologie saturée, tel que décrit en liaison avec la figure 4c, ce programme exécute une discrimination de concepts mutuellement exclusifs et le calcul d'une requête cohérente par appel aux concepts généralisants des concepts mutuellement exclusifs, ainsi que décrit en liaison avec la figure 4c.

REVENDICATIONS

1. Procédé de codage et de représentation synthétique d'une ontologie partiellement saturée, ladite ontologie comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini représentée par un treillis sous un concept universel et des informations non liées aux concepts, caractérisé en ce que ledit procédé consiste au moins à :

- coder ledit treillis de concepts par attribution à chaque concept et au nœud du treillis associé à ce dernier d'un identifiant formé par au moins une suite de nombres entiers, chaque suite de nombres entiers définissant un chemin d'accès entre un concept considéré et le concept universel, par l'intermédiaire de concepts pères successifs, ledit identifiant incluant toutes les suites de nombres entiers définissant chacune un chemin d'accès entre l'un desdits concepts pères et ledit concept universel, auxquelles est ajouté un nombre entier représentatif dudit concept considéré ;

- calculer les exclusions mutuelles entre concepts pour chaque concept considéré ;

- mémoriser ledit treillis de concepts codé sous forme d'une structure de données subdivisée en blocs incluant au moins une liste de concepts incluant au moins un nom de domaine et une liste comportant au moins un concept considéré, chaque concept considéré incluant au moins un identifiant formé par une liste d'au moins un chemin et une liste de concepts mutuellement exclusifs avec ledit concept considéré.

2. Procédé selon la revendication 1, caractérisé en ce que chaque concept est constitué par une dénomination et, le cas échéant, par au moins un synonyme.

3. Procédé selon la revendication 1 ou 2, caractérisé en ce que celui-ci consiste en outre à établir ladite structure de données à partir de :

- un ensemble de règles de définition, chaque règle de définition comportant une première partie mise en relation avec une deuxième partie constituant une définition de ladite première partie ;

- une pluralité de mots clés spécifiques à valeur syntaxique, permettant d'organiser syntaxiquement lesdits blocs de ladite structure de données ;
- une pluralité de symboles, chaque symbole représentant une valeur sémantique attribuée à un mot clé spécifique, ladite structure de données subdivisée en blocs contenant des blocs terminaux constitués chacun par un mot clé et au moins un symbole, ledit ensemble de règles de définition, ladite pluralité de mots clés et ladite pluralité de symboles étant constitués en un langage de transmission et de communication de ladite structure de données, dont les règles d'écriture constituent les règles d'écriture syntaxique.

4. Procédé selon la revendication 3, caractérisé en ce que ladite pluralité de symboles inclut au moins la barre verticale, les crochets, le symbole inférieur à, le symbole supérieur à, le signe plus, le point d'interrogation, le signe étoile.

5. Procédé selon l'une des revendications 3 ou 4, caractérisé en ce que lesdites règles d'écriture syntaxique comportent au moins :

- l'écriture d'un bloc déterminé en partie gauche et l'écriture d'une suite ordonnée de blocs en partie droite d'une règle de définition, à ladite écriture étant allouée une fonction de remplacement dudit bloc déterminé par ladite suite de blocs ;
- l'écriture d'un premier symbole, la barre verticale, entre deux blocs d'une pluralité de blocs, à ladite écriture dudit premier symbole étant allouée une fonction de choix entre plusieurs blocs situés à gauche dudit premier symbole pour remplacer la pluralité de blocs située à droite dudit premier symbole ;
- l'écriture d'une pluralité de blocs entre crochets, deuxième symbole, à ladite écriture étant allouée une fonction de remplacement de chaque bloc de ladite pluralité de blocs par une chaîne de caractères, dénomination d'instance, lesdites chaînes de caractères formant chacune un bloc terminal, présent dans ladite structure de données ;
- l'écriture d'une pluralité de blocs entre les symboles inférieur à, supérieur à, troisième symbole, à ladite écriture étant allouée une fonction de

remplacement de chaque bloc par sa définition à partir d'une règle de définition, lesdits blocs ne constituant pas des blocs terminaux ;

- l'écriture d'un quatrième symbole, signe étoile, associé à un bloc, à ladite écriture du quatrième symbole étant allouée une multiplication du nombre d'instances affectées au bloc ;

- l'écriture d'un cinquième symbole, signe plus, associé à un bloc, à ladite écriture du cinquième symbole étant allouée une multiplication du nombre d'instances affectées au bloc, sous contrainte de présence d'au moins une fois dans ladite structure de données ;

- l'écriture d'un sixième symbole, signe point d'interrogation, associé à un bloc, à ladite écriture du sixième symbole étant allouée la définition de champs facultatifs, sous contrainte de présence une seule fois ou d'absence du bloc dans ladite structure de données.

6. Procédé selon l'une des revendications 3 à 5, caractérisé en ce que ledit langage de transmission et de communication de ladite structure de données est l'un des langages choisis dans le groupe des langages xml, Owl.

7. Procédé selon l'une des revendications 1 à 6, caractérisé en ce que lesdites informations non liées aux concepts incluant des rôles, des prédicats ordinaires et des faits, ledit procédé consiste en outre à :

- coder chaque rôle selon un prédicat binaire incluant au moins une en-tête associée à un nom de rôle, un premier argument, de type concept, déterminé par le rôle et un deuxième argument, de type concept, déterminant dans le rôle ;

- coder chaque prédicat ordinaire selon au moins un en-tête associé à un nom de prédicat, un premier argument de type argument et un deuxième argument de liste de définition, chaque argument présentant en outre une valeur sémantique ;

- coder chaque fait selon une en-tête associée à un atome, dont au moins un des arguments est une constante.

8. Procédé selon la revendication 7, caractérisé en ce que ladite définition est codée par une en-tête associée à une pluralité d'atomes issus d'une règle de définition.

9. Procédé selon l'une des revendications 3 à 8, caractérisé en ce que à chaque règle de définition est associé un numéro d'ordre de lecture de règle, lequel permet d'ordonner tout sous ensemble de règles de définition codées pour optimisation d'une exécution desdites règles de définition dans ladite structure de données.

10. Procédé selon l'une des revendications 3 à 9, caractérisé en ce que ledit langage de transmission et de communication de ladite structure de données comporte en outre un ensemble d'opérateurs logiques d'expressions de logique de description incluant au moins :

- un premier opérateur logique <and> permettant la mise en relation logique d'intersection (ET) entre une première et une deuxième expression de logique de description par l'intermédiaire d'un en-tête "AND" ;
- un deuxième opérateur logique <forall> permettant la mise en relation d'un nom de rôle et d'une expression logique de description par l'intermédiaire d'un en-tête "FA" ;
- un troisième opérateur logique <at least> permettant la mise en relation d'un entier minorant et d'un nom de rôle par l'intermédiaire d'un en-tête "AL" ;
- un quatrième opérateur logique <atmost> permettant la mise en relation d'un entier majorant et d'un nom de rôle par l'intermédiaire d'un en-tête "AM" ;
- un cinquième opérateur logique <neg> permettant la négation ou exclusion d'un élément de base, tel qu'un concept, par l'intermédiaire d'un en-tête "NEG".

11. Structure de données subdivisée en blocs codée et représentative d'une ontologie comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini représentée par un treillis, caractérisée en ce

que ladite structure de données est structurée à partir d'une déclaration constituée d'une pluralité de lignes de déclaration formées par des blocs, incluant :

- une ligne de déclaration de l'ontologie déclarant, outre un nom de domaine de l'ontologie, une pluralité de concepts, de rôles de prédicats ordinaires et de faits ;
- une ligne de déclaration de concepts déclarant, outre un nom de concept, une pluralité de synonymes, un identifiant, une pluralité de concepts mutuellement exclusifs vis-à-vis du concept déclaré, et une pluralité d'opérateurs logiques d'expressions de logique de description facultatifs ;
- une ligne de déclaration de synonymes déclarant un nom de synonyme ;
- une ligne de déclaration d'identifiants, déclarés comme une pluralité d'entiers ;
- une ligne de déclaration de valeur d'entier ;
- une ligne de déclaration d'une liste de concepts mutuellement exclusifs d'un concept déclaré.

12. Structure de données selon la revendication 11, caractérisée en ce que ladite ontologie comprenant en outre des informations non liées aux concepts incluant des rôles, des prédicats ordinaires et des faits, ladite structure de données est structurée à partir d'une déclaration constituée en outre de lignes de déclaration formées par des blocs incluant :

- une ligne de déclaration de rôles déclarant, outre un nom de rôle, un argument discriminé par le rôle et un argument discriminant dans le rôle, à chaque argument étant attribuée une valeur sémantique par une déclaration de type d'argument ;
- une ligne de déclaration de prédicats ordinaires, déclarant, outre un nom de prédicat ordinaire, une pluralité de types d'arguments associée à un numéro d'ordre de lecture de règle et une pluralité de définitions facultatives ;
- une ligne de déclaration de définitions déclarant une pluralité d'atomes ;

- une ligne de déclaration d'atomes déclarant, outre un nom de prédicat d'atome, une pluralité d'arguments ;
- une ligne de déclaration de faits déclarant un fait comme un atome ;
- une ligne de déclaration d'opérateurs logiques d'expressions de logique de description déclarant au moins :
 - un premier opérateur logique <and> permettant la mise en relation logique d'intersection (ET) entre une première et une deuxième expression de logique de description par l'intermédiaire d'un en-tête "AND" ;
 - un deuxième opérateur logique <forall> permettant la mise en relation d'un nom de rôle et d'une expression logique de description par l'intermédiaire d'un en-tête "FA" ;
 - un troisième opérateur logique <at least> permettant la mise en relation d'un entier minorant et d'un nom de rôle par l'intermédiaire d'un en-tête "AL" ;
 - un quatrième opérateur logique <atmost> permettant la mise en relation d'un entier majorant et d'un nom de rôle par l'intermédiaire d'un en-tête "AM" ;
 - un cinquième opérateur logique <neg> permettant la négation ou exclusion d'un élément de base, tel qu'un concept, par l'intermédiaire d'un en-tête "NEG".

13. Structure de données selon l'une des revendications 11 ou 12, caractérisé en ce que ladite déclaration est établie en langage de balisage structuré, tel que le langage xml ou Owl.

14. Structure de données selon la revendication 13, caractérisé en ce que ladite déclaration comporte des balises insérées dans lesdites lignes de déclaration, lesdites balises étant formées par l'écriture de symboles, lesdits symboles comportant au moins :

- l'écriture d'un bloc déterminé en partie gauche et l'écriture d'une suite ordonnée de blocs en partie droite d'une règle de définition, à ladite écriture

étant allouée une fonction de remplacement dudit bloc déterminé par ladite suite de blocs ;

- l'écriture d'un premier symbole, la barre verticale, entre deux blocs d'une pluralité de blocs, à ladite écriture dudit premier symbole étant allouée une fonction de choix entre plusieurs blocs situés à gauche dudit premier symbole pour remplacer la pluralité de blocs située à droite dudit premier symbole ;

- l'écriture d'une pluralité de blocs entre crochets, deuxième symbole, à ladite écriture étant allouée une fonction de remplacement de chaque bloc de ladite pluralité de blocs par une chaîne de caractères, dénomination d'instance, lesdites chaînes de caractères formant chacune un bloc terminal, présent dans ladite structure de données ;

- l'écriture d'une pluralité de blocs entre le symbole inférieur à, supérieur à, troisième symbole, à ladite écriture étant allouée une fonction de remplacement de chaque bloc par sa définition à partir d'une règle de définition, lesdits blocs ne constituant pas des blocs terminaux ;

- l'écriture d'un quatrième symbole, signe étoile, associé à un bloc, à ladite écriture du quatrième symbole étant allouée une multiplication du nombre d'instances affectées au bloc ;

- l'écriture d'un cinquième symbole, signe plus, associé à un bloc, à ladite écriture du cinquième symbole étant allouée une multiplication du nombre d'instances affectées au bloc, sous contrainte de présence d'au moins une fois dans ladite structure de données ;

- l'écriture d'un sixième symbole, signe point d'interrogation, associé à un bloc, à ladite écriture du sixième symbole étant allouée la définition de champs facultatifs, sous contrainte de présence une seule fois ou d'absence du bloc dans ladite structure de données.

15. Système de codage et de représentation synthétique d'une ontologie partiellement saturée, comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini représentée par un treillis et des informations non liées aux concepts, caractérisé en ce que, outre des organes

d'entrée/sortie, une unité centrale de traitement et une mémoire de travail, ledit système inclut au moins :

- des moyens d'acquisition de ladite ontologie permettant de mémoriser ledit treillis et lesdites informations non liées aux concepts ;
- des moyens de codage dudit treillis de concepts par attribution à chaque concept et au nœud du treillis associé à ce dernier d'un identifiant formé par au moins une suite de nombres entiers, chaque suite de nombres entiers définissant un chemin d'accès entre un concept considéré et le concept universel, par l'intermédiaire de concepts pères successifs, ledit identifiant incluant toutes les suites de nombres entiers définissant chacune un chemin d'accès entre l'un des concepts pères et le concept universel, auxquelles est ajouté un nombre entier représentatif du concept considéré ;
- des moyens de calcul des exclusions mutuelles entre concepts pour chaque concept considéré ; et,
- des moyens de mémorisation du treillis de concepts codé sous forme d'une structure de données subdivisée en blocs incluant au moins un nom de domaine et une liste comportant au moins un identifiant, formé par une liste d'au moins un chemin et une liste de concepts mutuellement exclusifs avec ledit concept primitif considéré.

16. Système de consultation d'une ontologie partiellement saturée, comportant au moins des organes d'entrée/sortie, une unité centrale de traitement et une mémoire de travail, caractérisé en ce que ledit système comporte en outre :

- des moyens d'acquisition d'une structure de données subdivisée en blocs et représentative de ladite ontologie comportant au moins une hiérarchie de concepts formée par des concepts primitifs et au moins un concept défini représentée par un treillis, selon l'une des revendications 11 à 14 ; et,
- un module interpréteur de requête de consultation, transmise par un terminal utilisateur.

17. Système selon la revendication 16, caractérisé en ce que, pour une requête de consultation transmise par un terminal utilisateur portant sur un premier et

un deuxième concept disjoints, codés comme concepts mutuellement exclusifs dans ladite structure de données subdivisée en blocs, ledit module interpréteur exécute :

- une lecture desdits concepts dans la liste des concepts mutuellement exclusifs ; et, sur identification desdits premier et deuxième concepts dans ladite liste de concepts mutuellement exclusifs,
- une limitation du parcours de recherche sur les seuls concepts spécialisant de l'un desdits premier et deuxième concepts, compte tenu de l'exclusion mutuelle desdits concepts spécialisant de l'autre desdits premier et deuxième concepts, à partir des chemins d'accès des identifiants desdits concepts.

18. Système serveur d'accès en réseau IP à une ontologie partiellement saturée, caractérisé en ce qu'il comporte au moins, interconnectés :

- un portail fournisseur d'accès à ladite ontologie partiellement saturée, à partir d'une requête de consultation ;
- un système de consultation de ladite ontologie saturée selon l'une des revendications 16 ou 17.

19. Système serveur selon la revendication 18, caractérisé en ce que ledit portail fournisseur d'accès comporte au moins :

- un module de discrimination de concepts mutuellement exclusifs dans ladite requête par comparaison des chemins d'accès contenus dans l'identifiant de chacun des concepts contenus dans ladite requête ; et,
- un module de calcul d'une requête cohérente par appel aux concepts généralisants desdits concepts mutuellement exclusifs, à partir des chemins et des identifiants desdits concepts généralisants.

20. Produit de programme d'ordinateur mémorisé sur un support de mémorisation et exécutable par un ordinateur, caractérisé en ce que lors de l'exécution dudit produit de programme par un ordinateur ou par un système de codage dédié selon la revendication 15, celui-ci exécute le procédé de codage et de représentation synthétique d'une ontologie partiellement saturée selon l'une des revendications 1 à 10.

21. Produit de programme d'ordinateur mémorisé sur un support de mémorisation et exécutable par un ordinateur, caractérisé en ce que lors de l'exécution dudit produit de programme par un ordinateur ou par un système de consultation dédié selon l'une des revendications 16 ou 17, celui-ci exécute le une interprétation de toute requête de consultation émise par un terminal, selon la revendication 17.

22. Produit de programme d'ordinateur mémorisé sur un support de mémorisation et exécutable par un ordinateur, caractérisé en ce que lors de l'exécution dudit produit de programme par un ordinateur ou par un système serveur dédié comportant un portail d'accès à une ontologie saturée selon la revendication 18 ou 19, celui-ci exécute une discrimination de concepts mutuellement exclusifs et un calcul d'une requête cohérente par appel aux concepts généralisants desdits concepts mutuellement exclusifs selon la revendication 19.

1/8

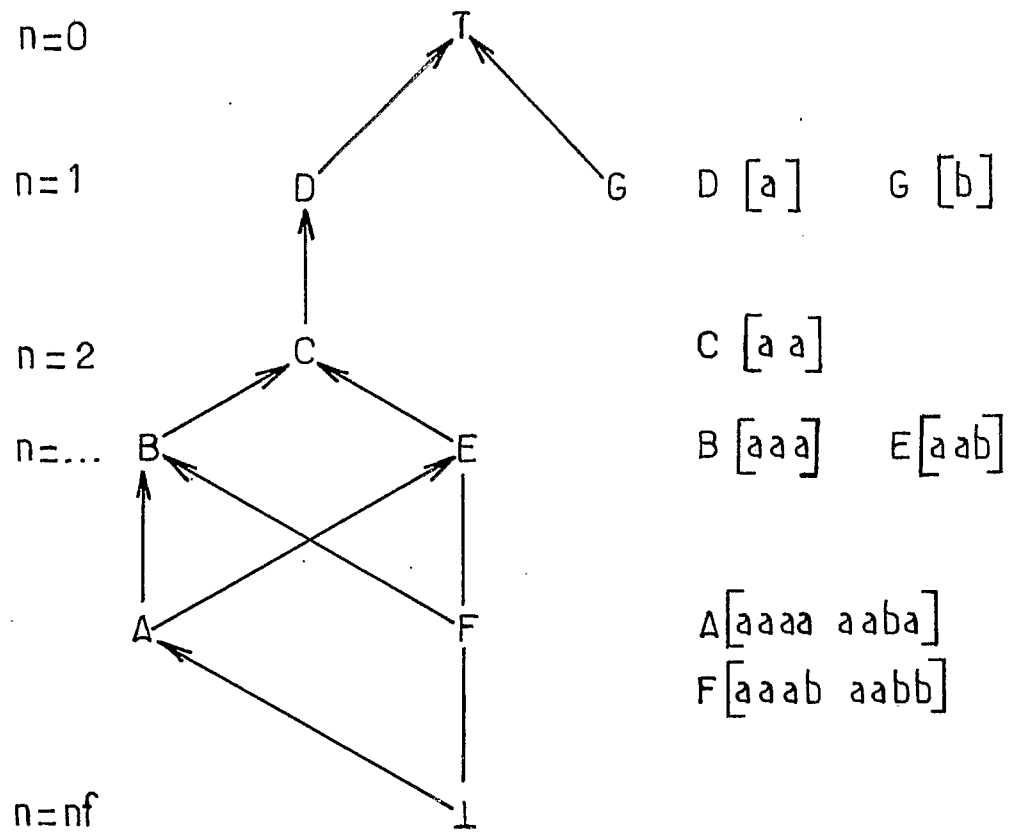


FIG.1.
(ART ANTÉRIEUR)

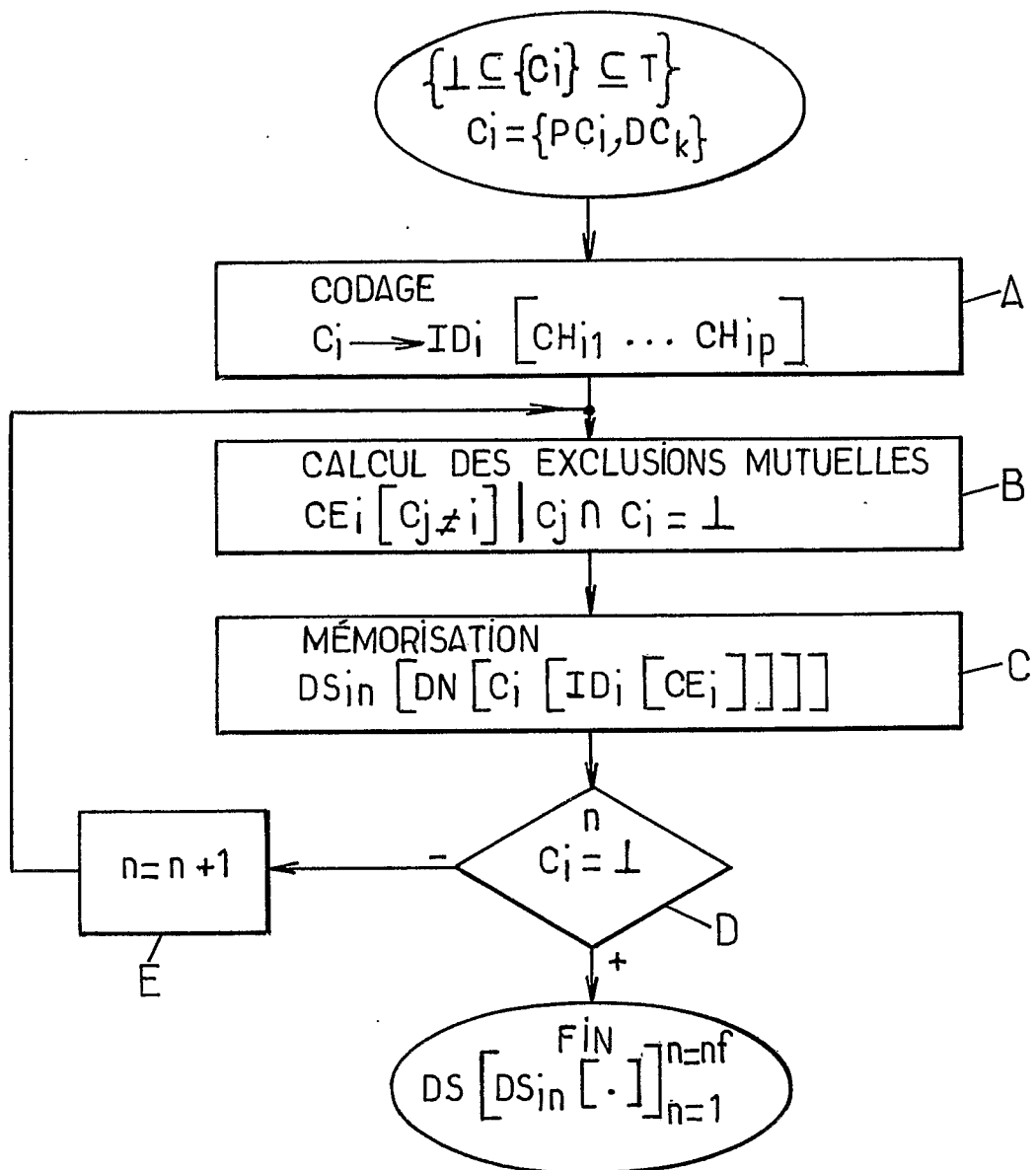


FIG.2a.

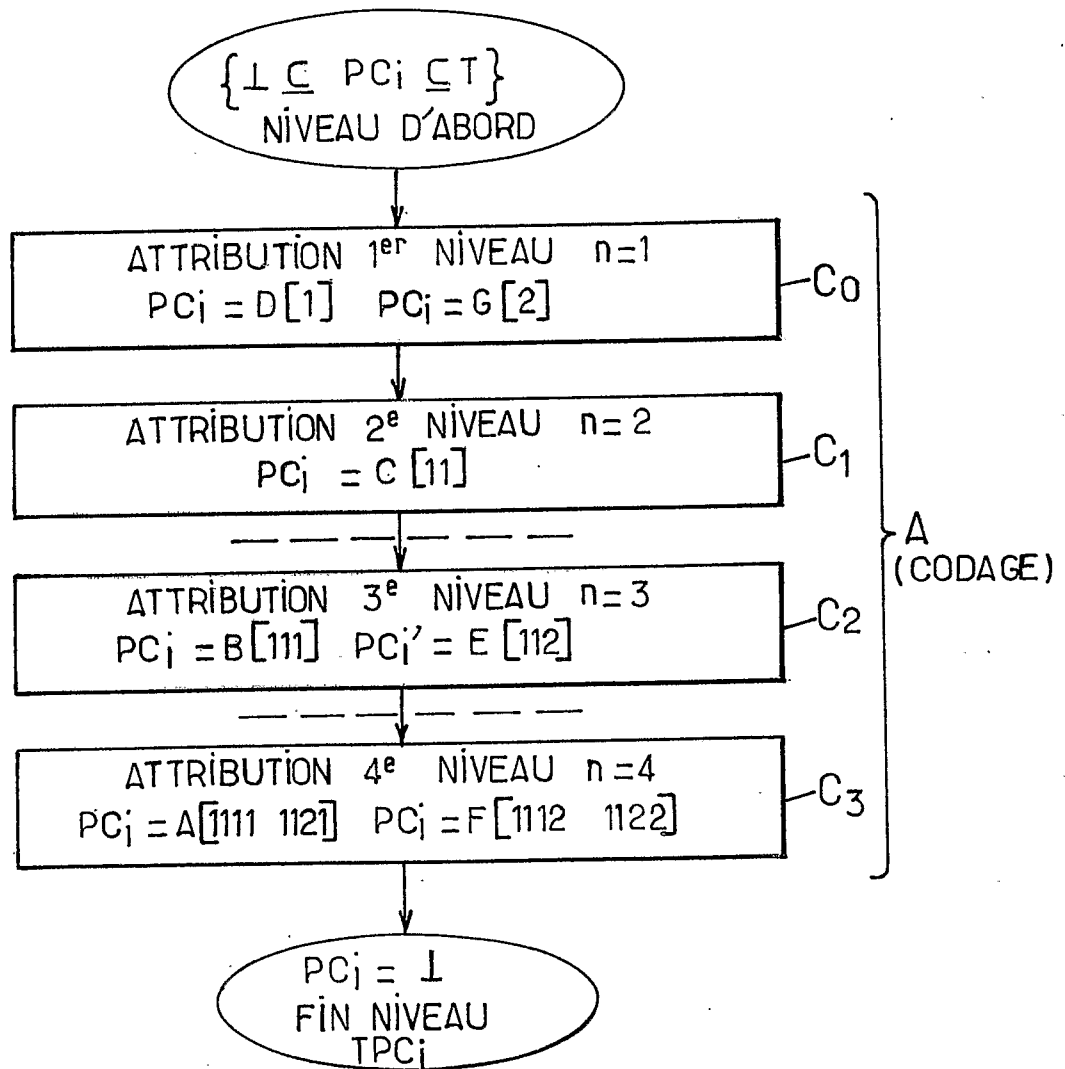


FIG. 2b.

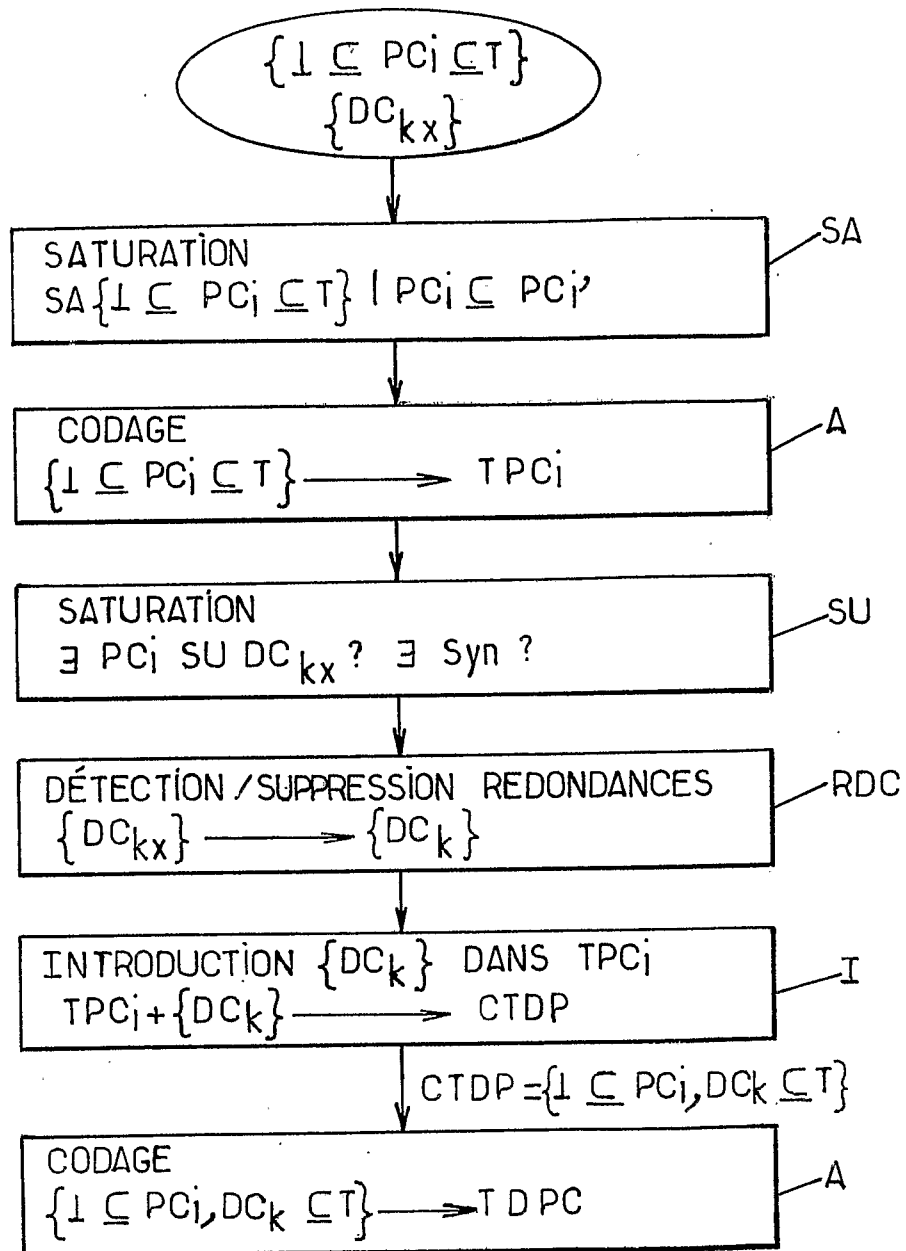


FIG.2c.

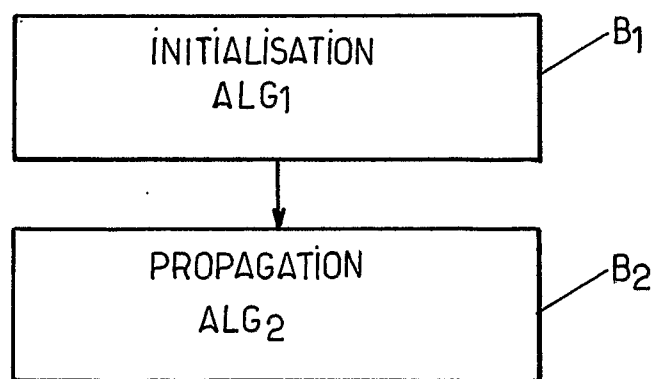
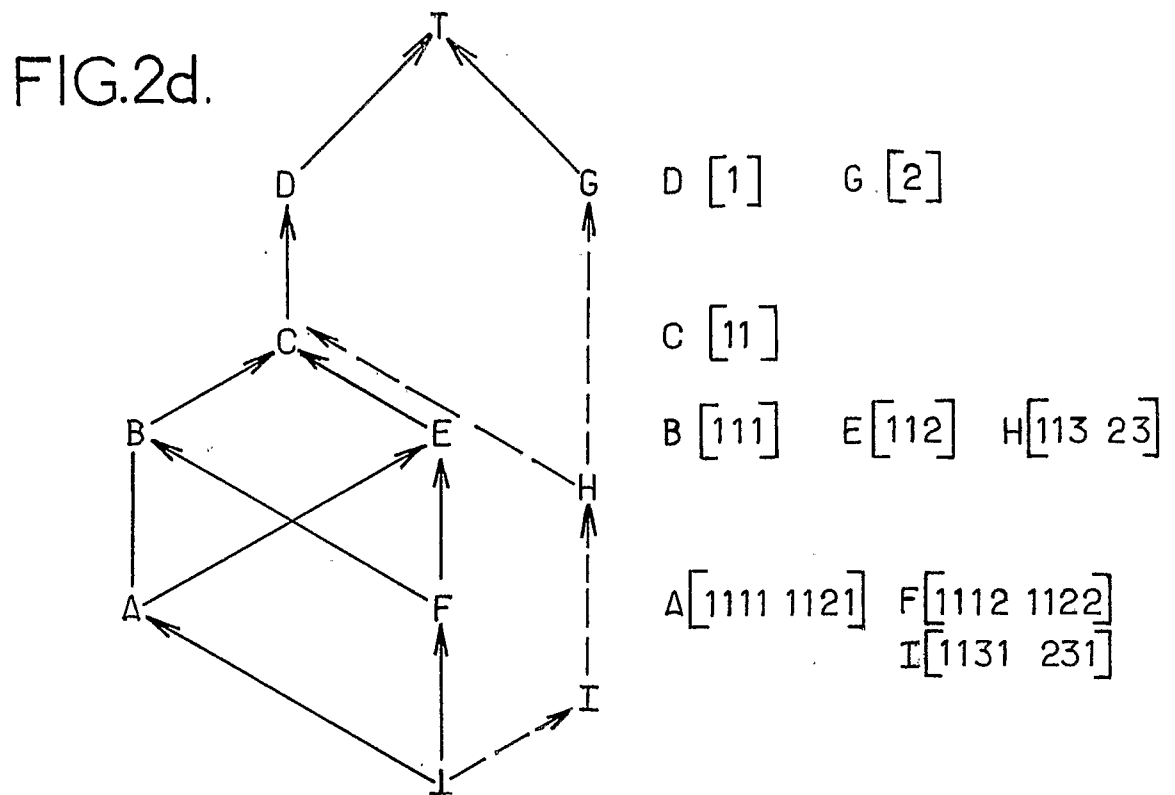
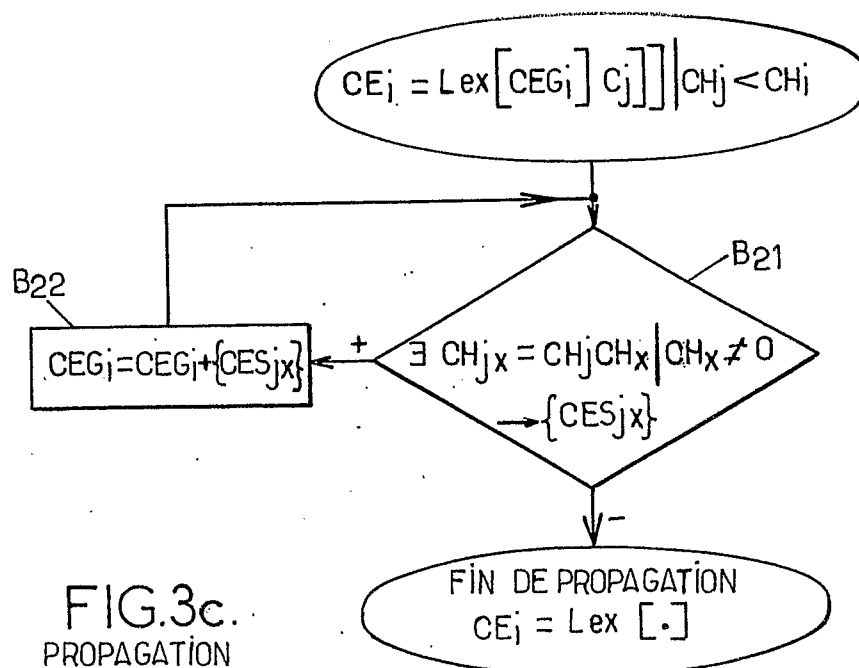
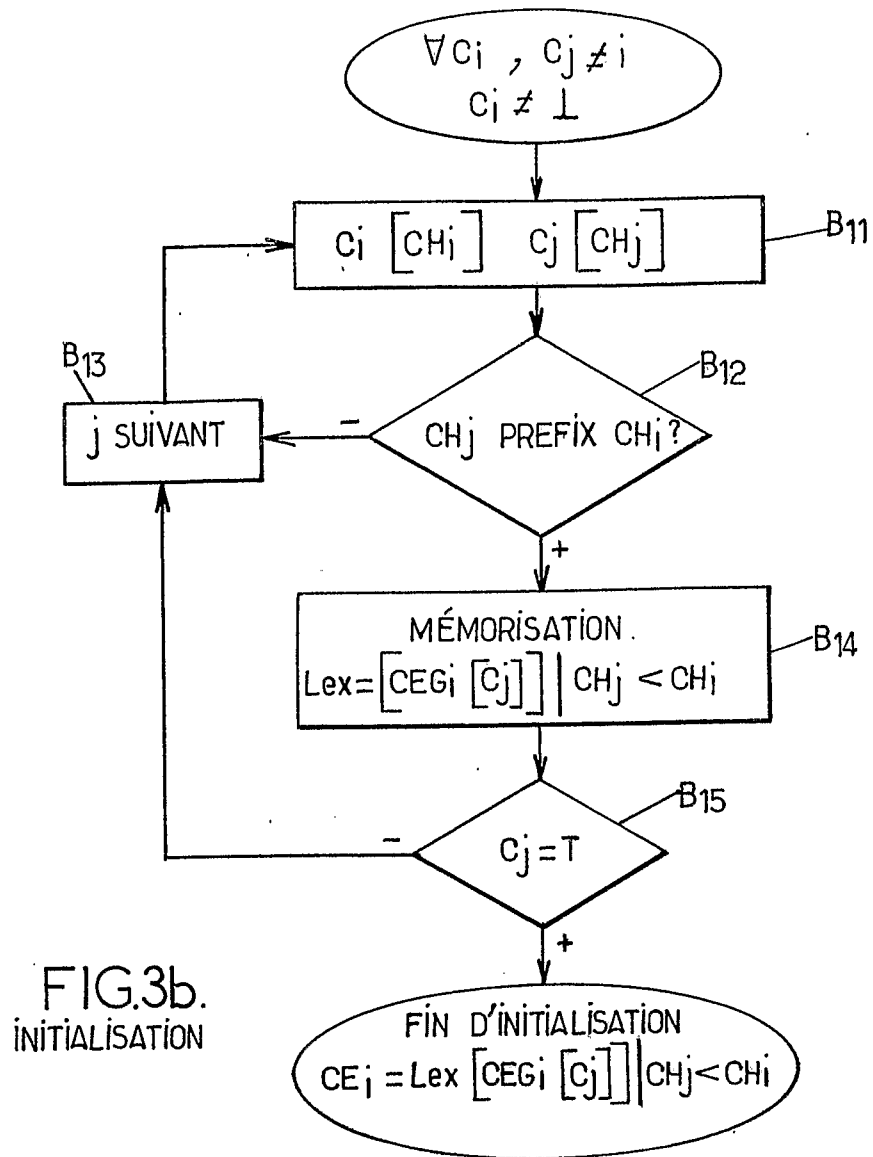
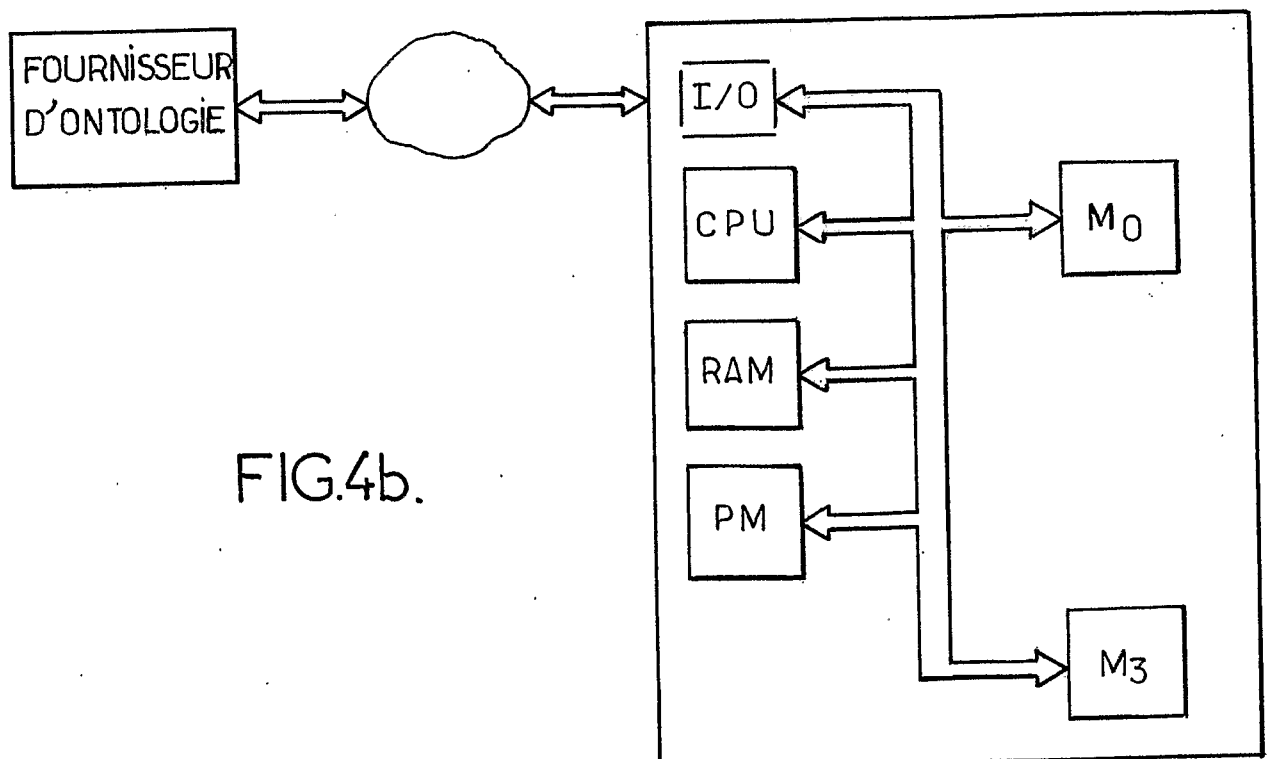
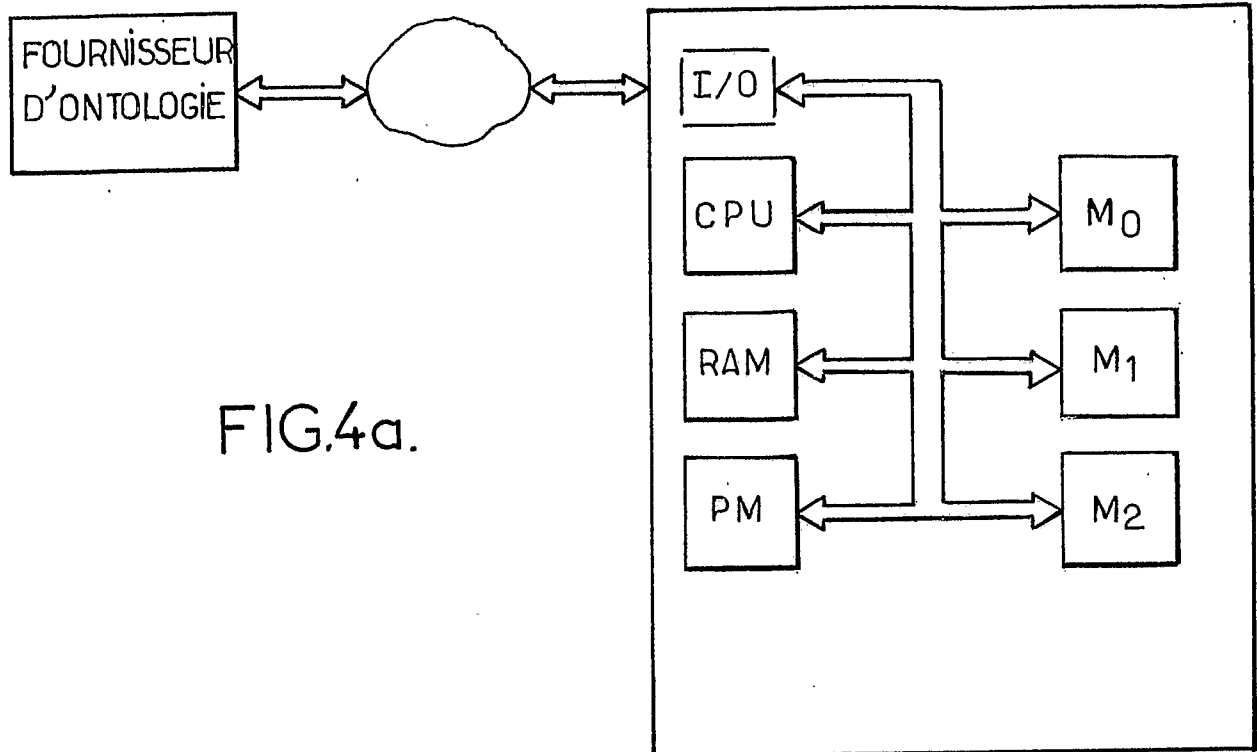


FIG.3a.





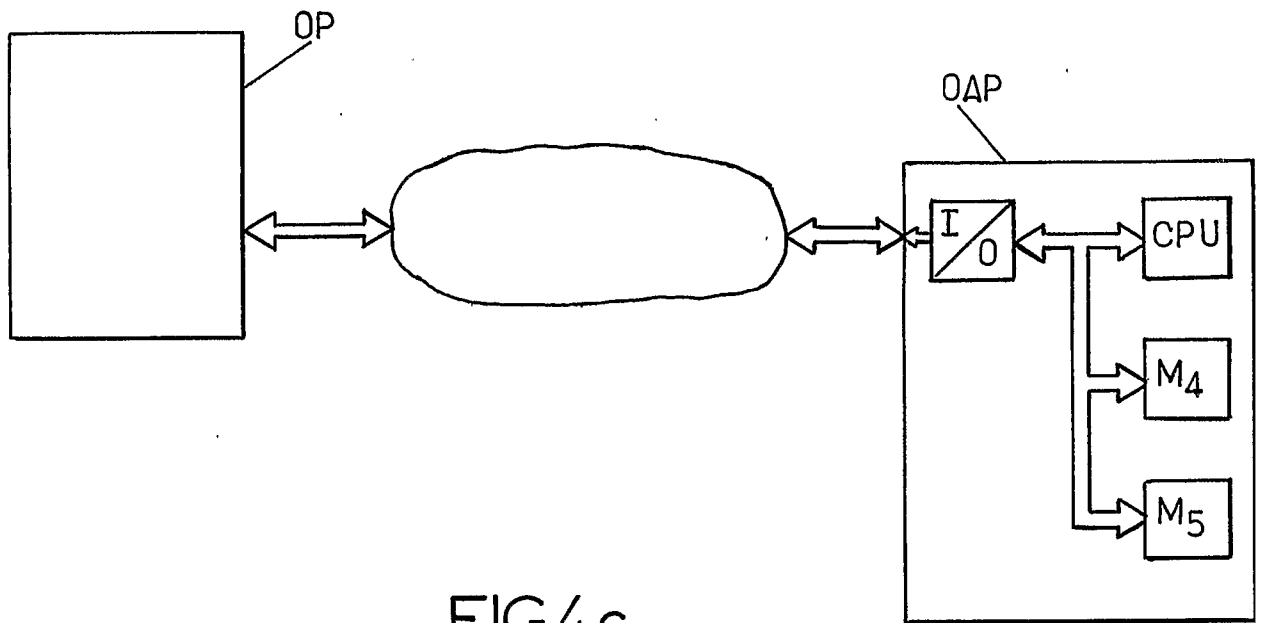


FIG.4c.