
Méthodologie de construction des ontologies pour la résolution de conflits de Systèmes d'Information

Mohamed Mhiri

*Laboratoire MIRACL
Institut supérieur d'informatique et du multimédia de Sfax
BP 3030 - 3018 Sfax TUNISIE
{mohamed.mhiri, faiez.gargouri}@fsegs.rnu.tn*

RÉSUMÉ. La modélisation d'un système d'information (SI) nécessite une parfaite connaissance du domaine concerné et une étude approfondie des exigences, sans cesse croissantes, des besoins des utilisateurs. Cette tâche devient très difficile en raison de la complexité des applications actuelles utilisant notamment des nouvelles technologies (e-learning, e-commerce, data warehouse, ...) et la masse énorme de concepts provenant de sources hétérogènes. Ces problèmes peuvent engendrer différents types de conflits (syntaxiques, sémantiques et structurels). Comme solution, nous proposons l'utilisation des ontologies comme un moyen d'assistance aux concepteurs durant leur tâche de modélisation pour un domaine donné. Dans cet article, nous proposons une méthodologie de construction des ontologies pour la résolution des conflits des systèmes d'information.

ABSTRACT. The information system (IS) modelling requires a perfect knowledge of the concerned field and a good study of user's requirements. This task becomes very difficult because the complexity of the current applications in particular using new technologies (e-learning, E-trade, data warehouse...) and the enormous mass of concepts coming from heterogeneous sources. These problems can generate various types of conflicts (syntactic, semantic and structural), we propose the ontology using to assist the designers during their task of modelling for a given field. In this paper, we propose an ontology building methodology to resolve their IS conflicts.

MOTS-CLÉS : conception de systèmes d'information, ontologie, relations sémantiques, relations conceptuelles, UMLOnto.

KEYWORDS: Information systems design, ontology, semantic relationships, conceptual relationships, UMLOnto.

1. Introduction

Le système d'information (SI) est considéré comme une pièce maîtresse pour la survie de toute organisation. En effet, informatisé, le SI d'une organisation, garantit la bonne circulation, la cohérence et l'arrivée juste à temps des informations utiles pour la prise de décision. De nos jours, ces systèmes sont de plus en plus complexes, nécessitant un énorme travail de modélisation. Les concepteurs sont souvent confrontés à un ensemble de difficultés liées principalement à la méconnaissance du champ d'étude et à la multitude des concepts utilisés. Ces problèmes sont dus au manque d'un consensus sur les vocabulaires utilisés pour un champ d'étude donné. En conséquence, les concepteurs peuvent, dans certains cas, commettre des erreurs syntaxiques, structurelles et/ou sémantiques. Ces erreurs auront des conséquences sur la cohérence des représentations conceptuelles (RC) résultats et, par suite, sur la qualité de l'implantation du SI.

D'un autre côté, la tâche d'analyse des besoins devient aussi très fastidieuse en raison, en particulier, de la nouvelle orientation des applications actuelles, utilisant massivement les nouvelles technologies. Ces applications se basent sur une énorme quantité de concepts provenant de sources hétérogènes. Une telle hétérogénéité entraîne souvent des ambiguïtés sémantiques au niveau des RC résultats. Les (ateliers de génie logiciel) AGL actuels ne permettent pas la détection ni la résolution de toutes les ambiguïtés énoncées. En effet, en raison de l'aspect sémantique de ces erreurs et du fait que les concepteurs ne disposent pas d'informations précises sur le domaine d'étude, ces erreurs ne sont pas détectables. Les AGL se limitent actuellement à la vérification de certaines erreurs syntaxiques relatives aux spécificités des diagrammes utilisés. Par exemple, ils peuvent vérifier que le nom d'une classe soit unique dans un diagramme donné. Paradoxalement, dans une RC donnée, un concepteur peut indiquer qu'une *classe Voiture* hérite d'une autre *classe Personne*, sans que cette erreur ne soit détectée.

D'autre part, plusieurs chercheurs (Gomez *et al*, 1999), (Fürst, 2002), (Gandon, 2002) ont pu démontrer que le concept d'ontologie permet d'analyser le savoir dans un domaine en modélisant les concepts pertinents pour une ou plusieurs applications de ce domaine. En effet, une ontologie comprend tout ce à quoi il faut penser dans un domaine, tous les objets de pensée d'un domaine. La façon d'y penser ou d'en parler indique le type ontologique de l'objet (Bachimont, 2000).

Une fois une ontologie est exprimée sous une forme assimilable par la machine, celle-ci devrait être en mesure d'assurer la tâche de synthèse de l'information qui est aujourd'hui à la charge de l'humain. La difficulté est donc transférée à la création de ces ontologies, qui doivent refléter fidèlement la sémantique du domaine ciblé. Les acteurs de ce domaine sont donc les plus qualifiés pour effectuer ce travail, à condition qu'ils soient capables de l'exprimer explicitement (Rousse, 2003). Partant de ce principe, nous pensons qu'une ontologie, peut être associée à un AGL afin de combler les insuffisances sémantiques de ce dernier.

Cet article est organisé comme suit. Nous commençons par présenter la classification de différents conflits de SI. Ensuite, nous présentons notre approche de construction des ontologies dédiées pour la conception des SI. Nous terminons par une conclusion.

2. Classification de conflits des SI

La conception des SI peut engendrer des difficultés qui concernent la représentation du monde réel. Nous trouvons parfois un domaine très complexe dont la modélisation conceptuelle engendre plusieurs entraves comme celle d'avoir une représentation complète et fidèle du monde réel. D'autres types de problèmes peuvent émerger, comme la présence d'erreurs sémantiques dues aux ambiguïtés existantes dans les relations entre les concepts. De plus, nous pouvons trouver des erreurs conceptuelles ou structurelles reliées aux interprétations des concepts spécifiques à un domaine, attribuées selon différents niveaux d'abstraction, par différents concepteurs.

Par ailleurs, dans le cas d'une conception collective d'un SI d'un domaine considéré comme « complexe », plusieurs problèmes peuvent émerger. En effet, dans ces conditions, la conception collective nécessite la présence de plusieurs concepteurs, et chaque concepteur peut modéliser le SI à sa manière et selon son point de vue. D'un autre côté, la conception et le développement d'un produit s'effectuent, de plus en plus, entre différentes équipes distribuées à la fois géographiquement et temporellement (Dartigues, 2002). De ce fait, l'intégration des différents RC résultats provoque généralement des éléments redondants (classes, attributs, méthodes) ou d'apercevoir des informations communes mais qui sont modélisées de façons différentes (Gargouri, 2002).

En plus, les concepteurs sont confrontés à des conflits de différents niveaux d'abstraction. Ces conflits ne sont pas détectables par les AGL. Ils peuvent provoquer des erreurs conceptuelles. Ils sont classés comme suit :

- *Les conflits syntaxiques* sont liés à la capacité de représentation du modèle de données. Une même information peut être représentée par différents modèles avec une syntaxe et des concepts différents. Ils sont conséquents aux différences entre les terminologies utilisées, lors de la conception, par les différents intervenants sur une même application donnée. Différents types de conflits terminologiques existent comme, par exemple, la synonymie.

- *Les conflits structurels* sont relatifs aux choix de conception établis pour chaque composant coopérant pour représenter les informations. Une même information peut être structurée autrement sur deux SI. Par exemple, concernant l'attribution de différents niveaux d'abstraction, par différents concepteurs, à un même concept (Classe/Attribut, Attribut/Méthode, ...). Comme exemple, un attribut Auteur dans la classe *Livre* peut être considéré comme classe dans une autre RC.

- Les conflits sémantiques concernent les ambiguïtés qui peuvent être engendrées par les relations entre les concepts, comme par exemple, l'inclusion et l'intersection de classes qui sont cachées et les contraintes non représentées entre classes (Berrabah, 2004).

Ces conflits peuvent, dans certains cas, provoquer des erreurs structurelles et sémantiques. Comme exemple, nous pouvons avoir cette représentation erronée : *une Personne hérite d'une Voiture*.

Comme moyen de résolution de ces conflits, nous proposons l'utilisation de l'ontologie pour un domaine (par exemple le commerce électronique). En effet, une ontologie permet d'analyser le savoir dans un domaine en modélisant les concepts pertinents d'un domaine donné. En plus, elle définit et fournit une sémantique d'un domaine. Dans notre cas, elle représente un moyen d'aide pour assister le concepteur lors de l'étape de modélisation.

Cela nécessite évidemment la construction d'ontologies dédiées à la conception de SI. Cette construction utilise un ensemble d'étapes permettant l'extraction des concepts et les relations conceptuelles et sémantiques entre ces concepts, l'utilisation d'un langage pour la représentation de l'ontologie résultat et une mise à jour incrémentale assurant sa réutilisation. Dans la section suivante, nous présentons notre approche de construction d'ontologie pour la conception de SI

3. Approche de construction d'une ontologie pour la conception de SI

La construction d'une ontologie nécessite l'utilisation d'un processus permettant de passer des données brutes à l'ontologie. Notre ontologie sera construite à partir d'un ensemble de RC orientées objet traitant un domaine donné et prises comme corpus (Mhiri *et al*, 2005a). Nous supposons que les RC, composant notre corpus, soient validées par leurs concepteurs ainsi que par des experts du domaine.

Les étapes de notre approche sont au nombre de quatre :

- Construction de l'ontologie initiale.
- Alimentation de l'ontologie.
- Pour chaque RC du corpus :
 - Extraction des concepts relatifs à RC.
 - Comparaison des concepts extraits avec ceux de l'ontologie initiale (pour le premier passage et l'ontologie Vi-1 pour le ième passage) : détermination des relations sémantiques.
 - Alimentation de l'ontologie avec les concepts de RC en respectant les liens sémantiques établis.
 - Alimentation de l'ontologie avec les liens conceptuels de RC.
- Représentation l'ontologie avec UMLOnto.

- Etude de la maintenance de l'ontologie.

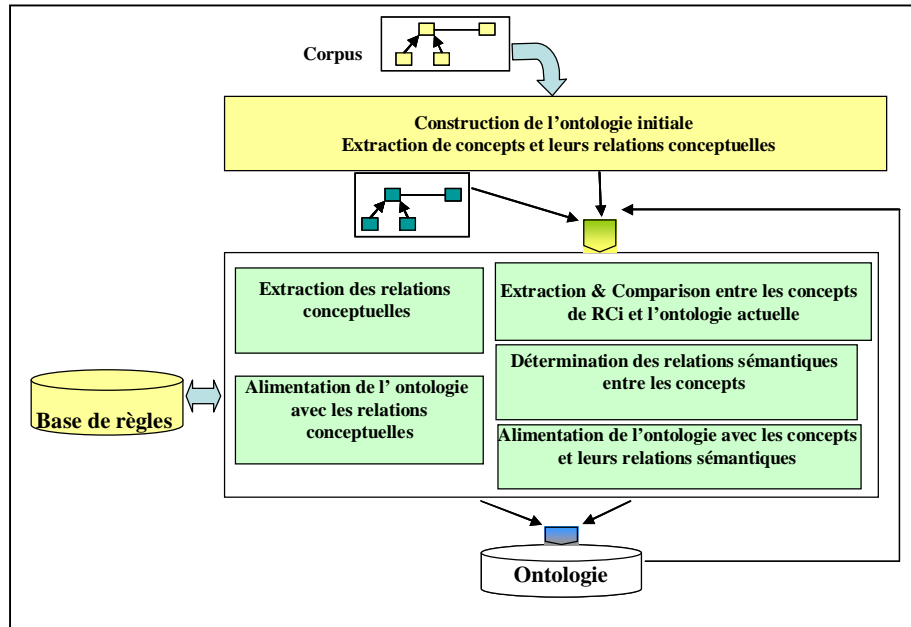


Figure 1. Approche de construction d'une ontologie pour la conception de SI

3.1. Construction de l'ontologie initiale

Pour la construction de l'ontologie initiale, nous optons à la prise en compte de la RC représentant le plus grand nombre de concepts pour un domaine donné. Ce choix est justifié par le fait que cette RC contient le plus grand ensemble de classes, ainsi que leurs liens possibles, permettant ainsi de représenter un univers de discours (du domaine concerné) plus large que les autres RC. En plus cette RC, prise comme corpus, doit être validée par l'expert du domaine et bien documentée (dictionnaire de données et leur description). Cette ontologie (construite manuellement) contient un ensemble de concepts et ses liens conceptuels.

Nous définissons un concept comme étant un terme caractérisé par une syntaxe et une sémantique représentées sous forme de chaînes de caractères. Il est caractérisé par une identité et contient un ensemble d'attributs et un ensemble d'opérations. Il peut être une classe ou bien une classe d'association (association porteuse de données).

La construction de l'ontologie initiale consiste à l'extraction des noms de classes, leurs propriétés (attributs et opérations) ainsi que leurs relations

conceptuelles. Par relation conceptuelle nous sous-entendons tous les liens supportés par UML.

Afin d'illustrer notre démarche, on se basera dans la suite sur des RC relatives au domaine du commerce électronique.

La médiathèque (Figure 2) représente une application de commerce électronique pour l'achat en ligne des produits (CD, livre, etc.). Elle renferme les classes *Usine* et *Consommateur* qui représentent l'usine qui fabrique les produits et l'acheteur ou le consommateur de ces produits. Chacune de ces deux classes possède une association avec la classe *Broker*, qui joue le rôle de courtier entre l'usine et le consommateur. La classe *Usine* a une autre relation avec la classe *Broker*. Cette association admet une classe de lien *Enchère*. La classe *Enchère* concerne la négociation qui peut se faire suivant une enchère donnée. La classe *Usine* est en association avec la classe *Produit*. Un *produit* peut être un *CD* ou un *Livre*. Cette relation est traduite par une relation de *généralisation-spécialisation*. La relation qui existe entre la classe *Consommateur* et la classe *Broker* admet une classe d'association *Ordre*. Cet ordre concerne le paiement, représenté par la classe *Paiement*, et la livraison représentée par la classe *Livraison*. Cette dernière peut être *Livraison_direct* ou livraison *Par_post*. Pour la classe *Paiement*, il existe deux relations. La première est une relation d'agrégation avec la classe *Effet*. La deuxième est une relation de *généralisation-spécialisation* avec les classes *Carte* et *Monnaie_électronique*. Les deux classes *Carte_crédit* et *Carte_débit* héritent de la classe *Carte*.

La médiathèque sera prise en compte pour la construction de l'ontologie initiale. Ainsi, le tableau suivant (table 1) représente une expression de la structure de notre ontologie initiale. Comme déjà mentionné, les relations conceptuelles qui peuvent exister entre les concepts. Ces relations peuvent être :

- une association : relation conceptuel entre deux concepts.
- une agrégation : relation d'ensemble/élément.
- une composition : relation de composé/composant.
- une généralisation-spécialisation : une relation de classification.

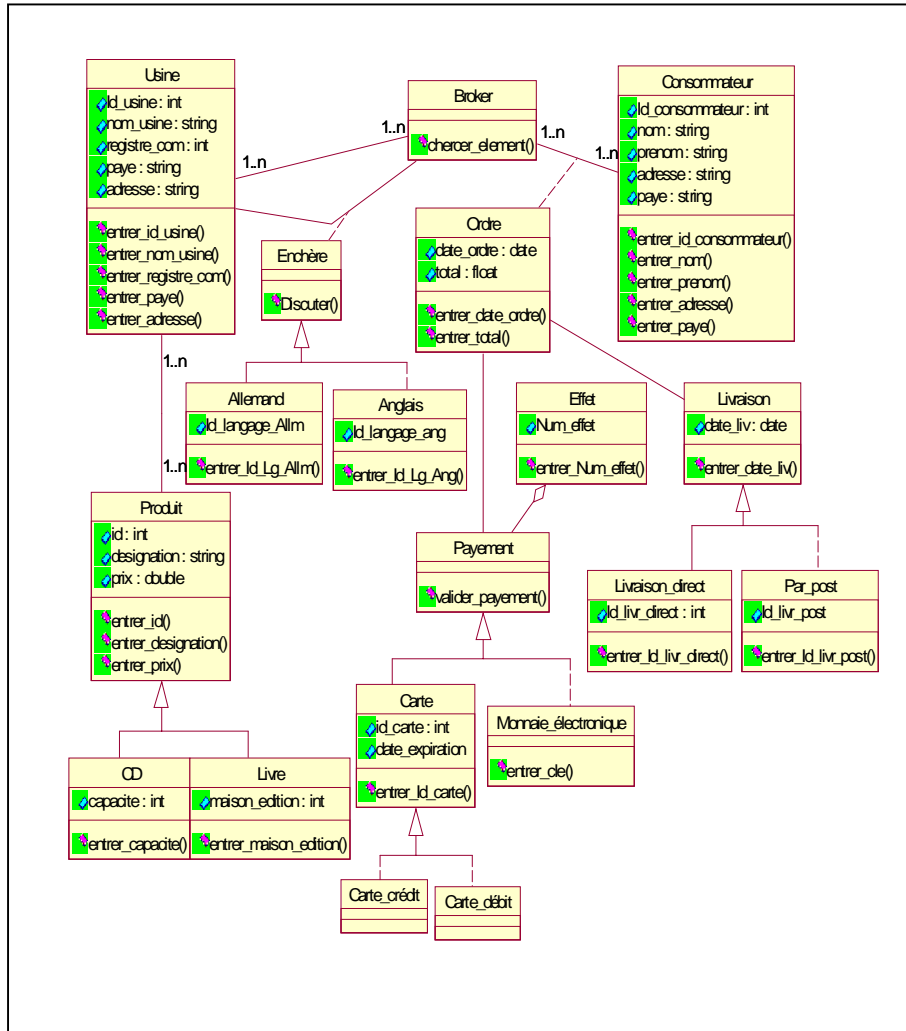


Figure 2. Example1 « Médiathèque »

Concept1	Concept2	Relation
Usine	Broker	Association
Usine	Produit	Association
Enchère	{ Usine, Broker }	Classe d'association
Enchère	{ Allemand, Anglais }	Généralisation
Produit	{ CD, Livre }	Généralisation

Ordre	{Broker, Consommateur }	Classe d'association
Ordre	Livraison	Association
Effet	Payement	Agrégation
Livraison	{Livraison_direct, par poste}	Généralisation
Payement	{carte, monnaie électronique}	Généralisation
Carte	{carte crédit, carte débit}	Généralisation

Table 1. Liste des relations entre les concepts de la médiathèque

3.2. Alimentation de l'ontologie

L'alimentation d'une ontologie, consiste à l'ajout des nouveaux concepts à partir des RC introduites par les concepteurs. Elle contient les phases suivantes.

Pour chaque RC :

- Extraction des concepts, leurs propriétés et leurs liens relatifs à la RC.
- Comparaison des concepts extraits avec ceux de l'ontologie initiale (Vi-1) : détermination des relations sémantiques.
- Alimentation de l'ontologie avec les concepts de RC en respectant les liens sémantiques établis.
- Alimentation de l'ontologie avec les liens conceptuels (de RC) entre les concepts ajoutés dans l'ontologie.

Dans les paragraphes suivants nous présentons ces différentes phases.

3.2.1. Phase 1 : Extraction des concepts relatifs à RC

Cette extraction est similaire à l'étape précédente, mais elle se base sur les RC élaborées par les concepteurs.

Prenons comme exemple la RC de vente des livres on line.

Tunisia Book est un diagramme de classes pour la vente et l'achat des livres en ligne. Dans ce diagramme de classes, représenté par la figure 3, les deux classes *Consommateur* et *Fournisseur* héritent de la classe *Tiers*. Chaque tiers possède une *carte_crédit* pour le payement électronique. Il accède à un agent mobile selon un login et un mot de passe. Chaque tiers doit s'abonner à un broker qui joue le rôle de courtier et un broker peut être abonné par un ou plusieurs tiers. Chaque fournisseur peut fournir un ou plusieurs livres et un livre peut être fournit par un ou plusieurs fournisseurs. Un fournisseur possède une certification, représentée par la relation d'agrégation entre la classe *Fournisseur* et la classe *Certification*. Le consommateur sélectionne les offres retournées par le broker après la phase de recherche dans la classe *Livre*. La classe d'association *Panier* contient la liste des

livres choisis par le consommateur. Le paiement se fait suivant un ordre désigné par la classe *Ordre*.

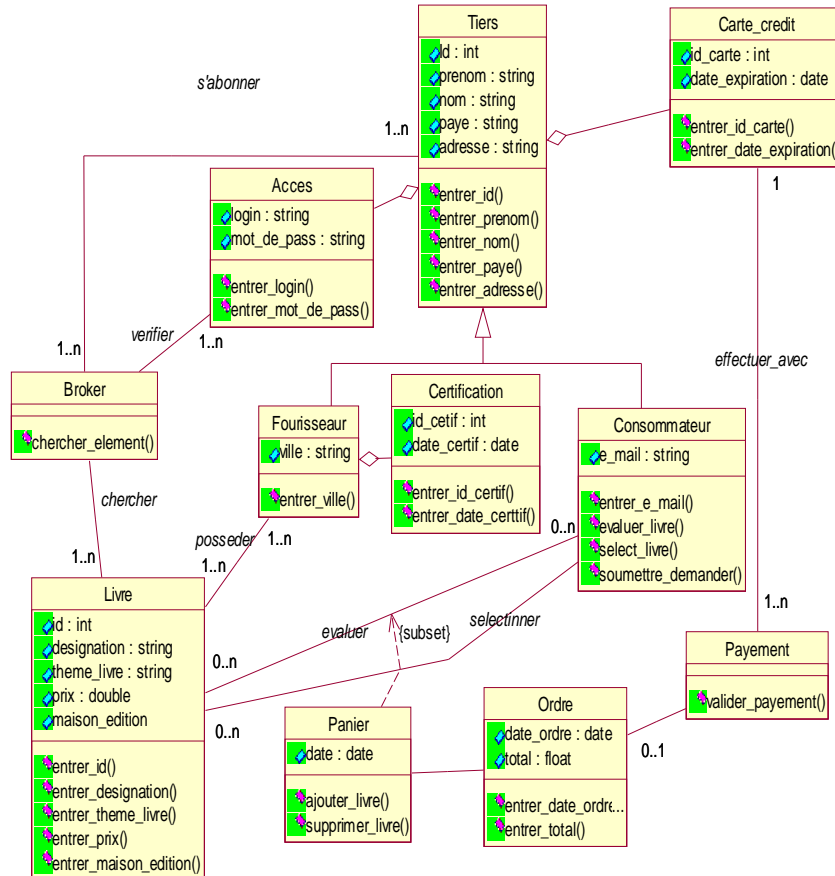


Figure 3. Un exemple d'une RC « Tunisia Book »

Le tableau 2 représente l'ensemble des concepts extraits de cette RC.

Nous remarquons que les concepts utilisés peuvent avoir des liens sémantiques avec ceux utilisés par l'ontologie (par exemple usine et fournisseur etc.).

La détermination de ces liens nécessite une comparaison entre les concepts d'une RC avec l'ontologie.

Liste des concepts
Tiers
Accès
Broker
Livre
Fournisseur
Certification
Consommateur
Panier
Ordre
Payement
Carte crédit

Table 2. Liste des concepts de Tunisia Book

3.2.2. Phase 2 : Détermination automatique des relations sémantiques entre les concepts de l'ontologie

Par relation sémantique, nous sous-entendons toute relation, autre que celles habituellement utilisées dans les RC UML (héritage, agrégation, ...) et qui est susceptible d'exister entre deux concepts. Ces relations peuvent être :

- *Relation d'identité* : c'est une relation sémantique qui existe entre deux concepts qui ont la même syntaxe, les attributs et les mêmes opérations. Exemple : *Identité*(Personne, Personne).
- *Relation de Synonymie* : c'est une relation sémantique qui existe entre deux concepts qui expriment le même sens. Exemple : *Synonymie*(Personne, Individu).
- Relation de classification *sorte_de*, entre deux concepts exprimant que l'un est un cas particulier de l'autre. Exemple : *Sorte_de* (Canari, Oiseaux).
- Relation d'*Homonymie* : un même concept peut avoir deux sens différents. Exemple : Homonymie (rose (couleur), rose (fleur)).
- Relation d'Equivalence : une relation sémantique qui existe entre deux concepts jouant le même rôle dans des RC différentes. Exemple : Equivalence(Usine, Fournisseur).
- Relation d'*Antonymie* : est utilisé entre deux concepts totalement disjoints. Exemple : *Antonymie*(Marié, célibataire).

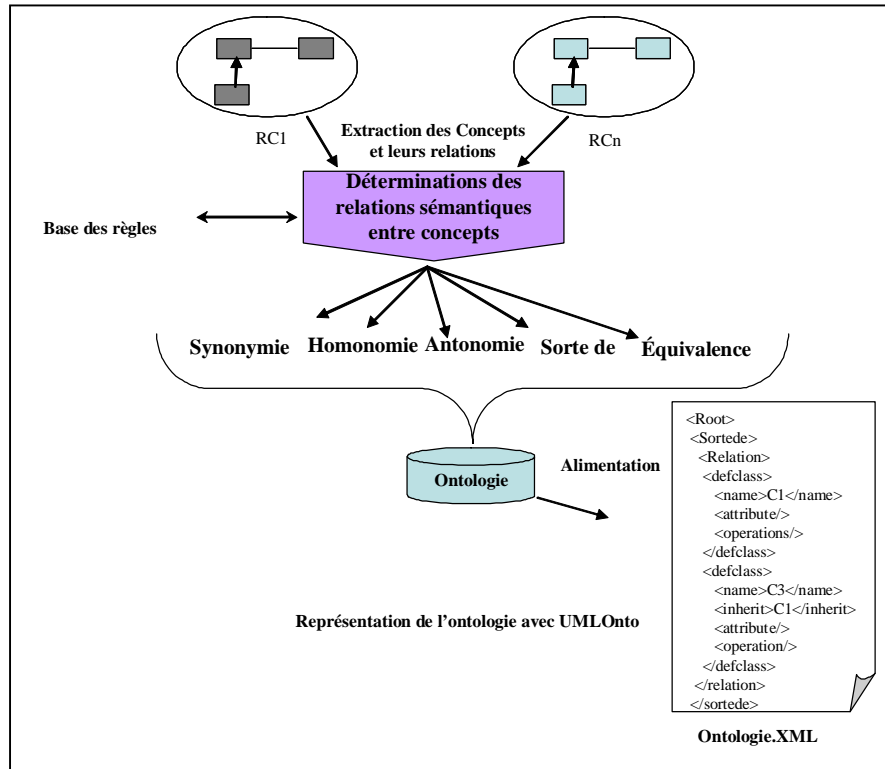


Figure 4. Processus de détermination des relations sémantiques

Après l'extraction des différents concepts, de la RC courante, une analyse sémantique est effectuée sur les concepts de noms des classes et leurs propriétés pouvant exister entre les concepts de classes (Mhiri *et al*, 2006) (Kraft *et al*, 2007). La figure 4 précise le processus de détermination des relations sémantiques. Cette analyse utilise un ensemble de mesures permettant de calculer la similitude entre la *syntaxe* des concepts (chaînes de caractères), la *syntaxe* des attributs (chaînes de caractères) et leur *sémantique*. Cette dernière concerne le sens des concepts représenté par ses attributs et ses relations conceptuelles (association, agrégation, composition et généralisation-spécialisation). Pour mesurer la similitude entre deux concepts, l'un de RC d'entrée et l'autre de l'ontologie, nous combinons l'appariement syntaxique entre les chaînes de caractères et l'appariement sémantique réalisé par (Hess *et al*, 2004), (Holt, 2000), (Jiang *et al*, 1997).

3.2.3. Phase 3 : Alimentation de l'ontologie avec les concepts de RC en respectant les liens sémantiques établis

Cette phase consiste à l'alimentation incrémentale des concepts de la RC qui ont une ou plusieurs relations sémantiques avec l'ontologie. Elle peut être automatique utilisant exclusivement les définitions formelles des relations sémantiques ou bien assistée par les concepteurs surtout dans le cas où les concepts sont ambigus. Nous sous-entendons par ambiguë les concepts qui n'ont pas la même syntaxe, ni les mêmes attributs ni les mêmes opérations. Dans ce cas, le concepteur peut envisager deux solutions :

- 1- ajouter le concept introduit comme une nouvelle entrée à l'ontologie.
- 2- choisir une relation sémantique avec la liste des concepts composant notre ontologie.

Dans le tableau suivant, nous présentons un exemple d'alimentation en se basant sur les définitions des relations sémantiques.

RCi	Ontologie	Relation sémantique
Usine	Fournisseur	Equivalence
Broker	Broker	Identité
Consommateur	Acheteur	Synonymie
Livre	Livre (monnaie)	Homonymie
Marié	Célibataire	Antonymie
Usine	Tiers	Sorte de

Table 3. Alimentation de l'ontologie avec les concepts de RC en respectant les liens sémantiques établis

3.2.4. Phase 4 : Alimentation de l'ontologie avec les liens conceptuels de RC

Après la troisième phase, une nouvelle lecture de la RC est effectuée pour extraire les relations conceptuelles entre les concepts alimentés dans l'ontologie. Ces relations sont nécessaires pour représenter le contexte d'un concept dans le domaine de l'étude. Elles seront à leurs tours alimentées dans l'ontologie. Par conséquent, l'ontologie sera enrichie par les nouveaux concepts et leurs relations conceptuelles en plus des liens sémantiques entre les concepts de l'ontologie version i-1.

Le tableau suivant représente un exemple d'alimentation des liens conceptuels (exemple du concept tiers).

Tiers	Carte crédit	Agrégation
Tiers	{ fournisseur, consommateur }	Généralisation
Tiers	Broker	Association
Tiers	Accès	Agrégation

Table 4. Liste de liens conceptuels du concept tiers

3.3. Représenter l'ontologie avec UMLOnto

Une ontologie peut s'exprimer selon plusieurs degrés de formalisation allant des définitions les plus informelles, en langage naturel, aux expressions écrites en un langage totalement ou partiellement formel (par exemple la logique du premier ordre) devant respecter une syntaxe et sémantique très stricte. Le degré de formalisation de l'ontologie va dépendre principalement des besoins.

Il existe plusieurs formalismes qui permettent la représentation des concepts d'une ontologie. Les plus utilisés sont :

- des formalismes basé sur la logique,
- des formalismes basé sur les réseaux sémantiques,
- des formalismes basé sur les graphes conceptuels,
- des langages basé sur les frames,
- des langages dédiés pour un domaine (médical, génétique, agriculture, ...etc).

Pour notre approche, nous proposons un langage spécifique pour la description des ontologies. Il s'agit d'une extension du standard de la conception UML baptisée UMLOnto (Mhiri *et al*, 2005b) (Figure 4). La proposition d'une extension de UML a été motivée essentiellement par deux raisons (i) UML, bien qu'étant un standard, n'est pas adapté pour les spécificités des ontologies (en particulier les différents types de liens entre les concepts), et (ii) les différentes autres propositions pour la représentation des ontologies (les réseaux sémantiques, ...) deviennent vite illisibles et incompréhensibles par les spécialistes.

Les objectifs que nous avons fixés pour UMLOnto sont :

- une extension de UML,
- une extensibilité et une adaptabilité de ses concepts,
- une représentation simple et concise des concepts d'un domaine ainsi que de leurs relations,
- une généricité des concepts,
- une représentation compréhensible par les concepteurs et les utilisateurs.

La modélisation d'une ontologie nécessite l'introduction de nouvelles représentations précisant les concepts et les différentes relations qui peuvent exister entre ces concepts.

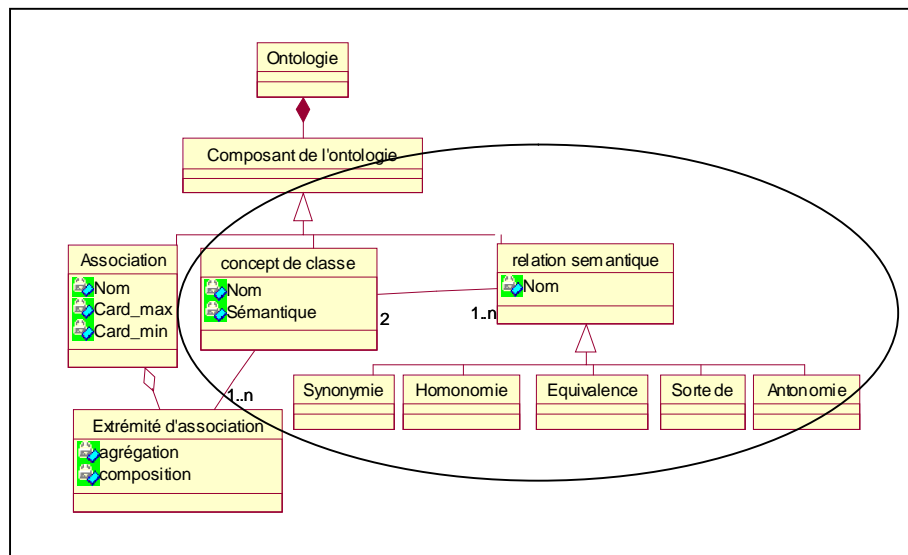


Figure 5. Les concepts du langage UMLOnto

3.4. Etude de Maintenance de l'ontologie

Une fois l'ontologie est construite, une étude de sa maintenance est effectuée. Cette étape regroupe la mise en oeuvre opérationnelle de l'ontologie et son utilisation. Elle comprend également la maintenance corrective (liée aux erreurs résiduelles) et la maintenance adaptative ou évolutive (liée à l'évolution des besoins). Elle a pour but de tester la cohérence de l'ontologie et d'effectuer des tests d'utilisation dans le but de corriger ou d'ajuster les éléments nécessaires (par un retour à la phase de construction).

La maintenance des ontologies est généralement réalisée en prenant en compte à la fois la correspondance entre l'ontologie et le monde réel et la qualité d'expression de l'ontologie. Par exemple, (Gomez-Perez *et al*, 1999) proposent trois critères pour mesurer la qualité d'une ontologie:

- La précision de la modélisation elle-même (clarté, standardisation du vocabulaire, suppression des concepts quasi-homonymes).
- La fiabilité de l'ontologie (complétude, cohérence, extensibilité).
- La qualité de la structuration (disjonction des classes, utilisation de l'héritage multiple, modularité).

Dans notre cas, nous proposons la définition d'un ensemble d'opérateurs pour assurer la maintenance corrective et la maintenance évolutive de notre ontologie (Mhiri *et al*, 2007). Ces opérateurs ont pour objectifs :

- l'adéquation des changements effectués aux besoins des concepteurs.
- la suppression des concepts isolés,
- d'éliminer la redondance des relations conceptuelles,
- d'éliminer la redondance des propriétés d'un concept,

Dans les paragraphes suivants, nous présentons la définition de ces opérateurs.

3.4.1. Définitions des Opérateurs

Nous considérons une ontologie (Ocsi) dédiée pour la conception des SI, caractérisée par un ensemble de concepts (C), un ensemble des propriétés de concepts (attributs (Ac) et opérations (Opc)) et un ensemble de relations (Rel) entre les concepts : conceptuels et sémantiques. Elle est représentée par le tuple suivant :

$$Ocsi = \langle C, Ac, Opc, Rel \rangle$$

Un opérateur permet d'effectuer un ou plusieurs changements dans une ontologie. Cet opérateur est défini par les pré-conditions minimales qu'un concept doit satisfaire avant l'opération et les post-conditions maximales que concept vérifie après chaque transformation.

En plus, il peut propager le changement dans toute la structure de l'ontologie. Il est caractérisé par une syntaxe et une sémantique. Il peut avoir un ou plusieurs arguments, les arguments d'un opérateur ne peuvent être que des éléments des ensembles des attributs, des opérations, des concepts ou des relations de l'ontologie. Dans les paragraphes suivants, nous définissons quelques opérateurs.

- **Supprimer un concept** : *Supprimer-Concept(Ocsi, c : C)*

Il est utilisé pour la suppression d'un concept c à partir de l'ontologie Ocsi. La tâche de suppression d'un concept de l'ontologie ne se consacre pas seulement à sa destruction de l'ontologie, mais il s'agit de prévoir la façon de gérer les effets de cette suppression sur la structure de l'ontologie. Cet opérateur fait appel à l'opérateur Supprimer-Prop(Ocsi, $c : C, p : P$) pour supprimer ses propriétés.

- **Ajouter un concept** : *Ajouter-Concept(Ocsi, c')*

Il permet d'ajouter un nouveau concept c' à l'ontologie. Cela nécessite évidemment l'ajout de ses propriétés (attributs et opérations).

- **Renommer un concept** : *Renommer-Concept(Ocsi, c : C, c')*

Il est utilisé pour effectuer une opération de renommage d'un concept: l'ancien nom c du concept est remplacé par le nouveau nom c' . Cet opérateur permet de corriger les erreurs et des ambiguïtés syntaxiques possibles qu'on peut trouver dans une ontologie.

- **Renommer une propriété** : *Renommer-Prop(Ocsi, c : C, p : P, p')*

Il est utilisé pour renommer la propriété p du concept c par une nouvelle propriété p' de l'ontologie ciblée par l'opérateur de renommage.

- Déplacer une propriété : *Déplacer-Prop* ($Ocsi, p : P, cs : C, cd : C$)

Il permet de déplacer une propriété p d'un concept source cs vers un concept destination sd .

- Regrouper deux concepts : *Regrouper-Concept* ($Ocsi, c : C, ci : C, cj : C, cg$)

Il permet de regrouper deux concepts (ci, cj) en un concept plus générique cg . Il sert à la généralisation des propriétés de deux concepts de l'ontologie. La factorisation de propriétés communes pour deux concepts permet d'effectuer une optimisation de l'ontologie en renforçant le degré de granularité et de modularité de l'ontologie afin de faciliter sa réutilisation. L'agrégation des propriétés communes doit être accompagnée de l'opération de la comparaison des propriétés afin d'extraire celles qui sont communes et cohérentes sémantiquement.

L'utilisation des opérateurs (*ajouter-concept, supprimer-concept, renommer-concept et regrouper-concept*) peut influencer la qualité et la structure d'une ontologie. En effet, l'utilisation de tels opérateurs peut provoquer des concepts isolés (sans relation conceptuelle ou sémantique), ou bien des relations redondantes ou erronées. Comme solution, nous proposons d'autres opérateurs permettant l'ajout, le remplacement, la suppression des relations entre les concepts. Nous supposons que ces relations soient binaires.

- Ajouter une relation : *Ajouter-relation* ($Ocsi, rel, type, ci : C, cj : C$)

Il permet d'ajouter une relation conceptuelle entre deux concepts ci et cj . Cette relation peut être une association, une agrégation, une composition ou un héritage. L'ajout d'un type de relation doit vérifier les conditions relatives à la sémantique des concepts. Il est utilisé dans le cas de l'ajout d'un nouveau concept, la suppression d'un concept, le renommage d'un concept ou bien le regroupement de deux concepts.

- Remplacer une relation : *Remplacer-relation* ($Ocsi, rel : Rel, rel', type, ci : C, cj : C$)

Il permet de remplacer une relation existante rel par une autre rel' . Il a comme objectifs la correction des erreurs conceptuelles entre concepts et la propagation des changements possibles effectués par les opérateurs d'évolution des concepts.

- Supprimer une relation : *Supprimer-relation* ($Ocsi, rel : Rel, type, ci : C, cj : C$)

Il permet de supprimer une relation rel entre deux concepts ci et cj . Cette relation peut être une relation conceptuelle ou sémantique.

L'utilisation de ces opérateurs peut nécessiter la modification ou bien la détermination des liens sémantiques entre les concepts. La déduction de ces liens nécessite une nouvelle comparaison entre les propriétés des concepts ajoutés ou modifiés.

3.4.3 Exemple d'utilisation des opérateurs de maintenance

Afin de représenter un exemple d'utilisation des opérateurs de maintenance, nous prenons le cas de l'ontologie du commerce électronique (*Oec*) qui contient un ensemble de concepts (*cc*). L'exemple consiste à la suppression de deux concepts selon deux cas.

- Le concept à supprimer représente un concept dérivé

Prenons l'exemple de paiement d'une commande par un acheteur. Le paiement peut être effectué par une carte (de crédit ou de débit) ou bien par une monnaie électronique (figure 6). Nous voudrions supprimer le concept *monnaie électronique*.

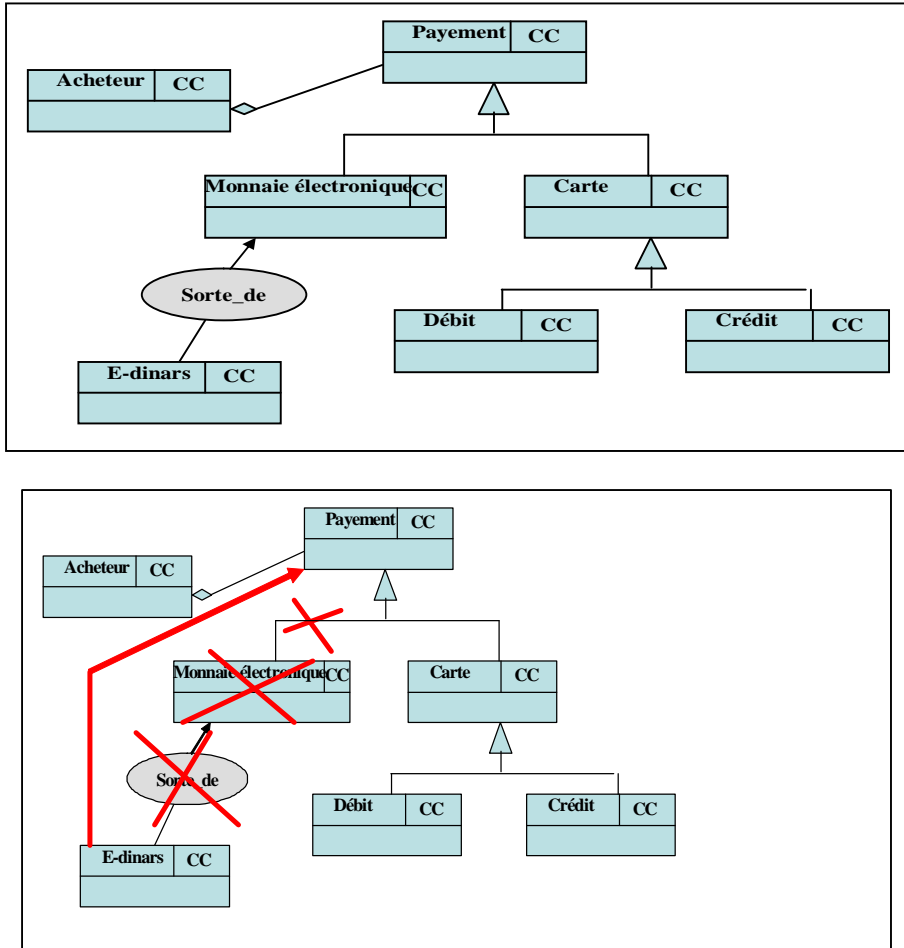


Figure 6. Extrait d'une ontologie pour le e-commerce

Les phases à entreprendre sont :

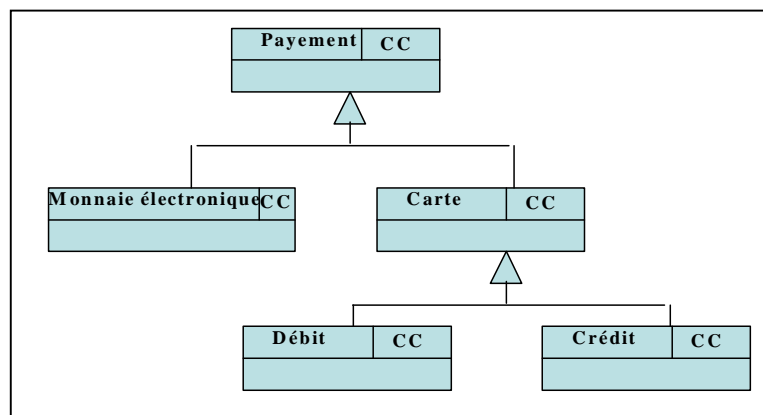
1. Supprimer l'ensemble de propriétés du concept *Monnaie électronique* en utilisant l'opérateur *Supprimer-prop(Oec, Monnaie-électronique, P)*.
3. Supprimer le lien d'héritage entre le concept générique *Payement* et le concept spécifique *Monnaie électronique* en utilisant l'opérateur *Supprimer-relation(Oec, Hérit, Héritage, Payement, Monnaie-électronique)*.
3. Supprimer le lien sémantique entre le concept *Monnaie-électronique* et *E-dinars* en utilisant l'opérateur *Supprimer-relation(Oec, Syn, Synonymie, Monnaie-électronique, e-dinars)*.
4. Ajouter un lien d'héritage entre le concept générique *Payement* et le concept spécifique *E-dinars* en utilisant l'opérateur *Ajouter-relation (Oec, Hérit, Héritage, Payement, E-dinars)*.
5. Supprimer le concept *Monnaie-electrique* en utilisant l'opérateur *Supprimer-concept (Oec, Monnaie-électronique)*.
6. Valider la suppression avec sa propagation.

- Le concept à supprimer représente un concept générique

Nous proposons la suppression du concept générique *Payement* représentée par la figure 7.

Les phases à entreprendre sont :

- 1- Déplacer les propriétés du concept *Payement* en utilisant les opérateurs *Déplacer-prop (Oec, p, Payement, Monnaie-électronique)* et *Déplacer-prop(Oec, p, Payement, Carte)*.
- 2- Supprimer le lien d'héritage entre les concepts *Payement* et *carte* en utilisant l'opérateur *Supprimer-relation(Oec, Hérit, Héritage, Payement, carte)*.
- 3- Supprimer le lien d'héritage entre les concepts *Payement* et *Monnaie-électronique* *Supprimer-relation (Oec, Hérit, Héritage, Payement, Monnaie-électronique)*.
- 4- Supprimer le concept *Payement* en utilisant l'opérateur *Supprimer-concept (Oec, Payement)*.



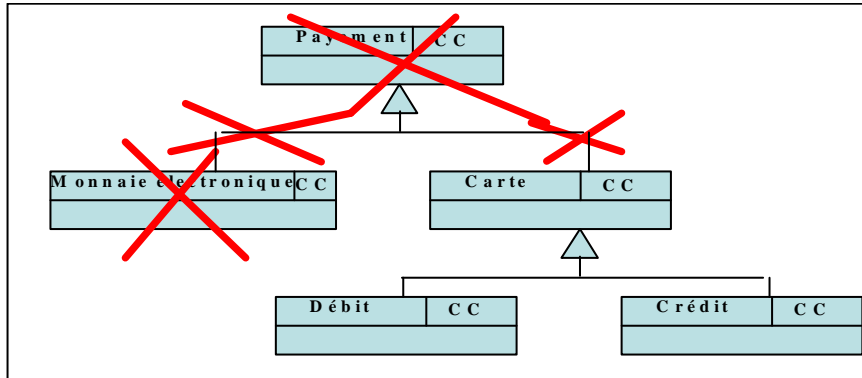


Figure 7. Extrait d'une ontologie contenant le concept Payment

5- Supprimer le concept isolé Monnaie-électronique en utilisant les opérateurs Supprimer-prop(*Oec*, *Monnaie-électronique*, *P*) et Supprimer-concept(*Oec*,*Monnaie-électronique*) .

6- Valider la suppression par le concepteur.

4. Conclusion

Dans cet article, nous avons proposé une approche de construction des ontologies dédiées pour la conception de SI. Cette approche représente un moyen d'assistance pour les concepteurs pour la résolution des différents conflits. En effet, cette ontologie de domaine, une fois construite, représente un référentiel contenant l'ensemble de concepts et leurs relations (conceptuelles et sémantiques). Elle peut être intégrée dans l'outil d'aide à conception, assurant le couplage avec une RC introduite par un concepteur. Ce couplage permet de communiquer avec l'utilisateur (à travers un API), de proposer une aide de conception en fournissant l'ensemble de concepts) et de corriger les erreurs conceptuelles.

5. Bibliographie

- Bachimont B., « Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances », In J. Charlet, M. Zacklad, G. Kassel, D. Bourigault éditions, Ingénierie des connaissances : évolutions récentes et nouveaux défis, p.305-323, Eyrolles, 2000.
- Berrabah D., Boufares F., Ducateau C.F., Gargouri F., Heiwy V., « Les conflits entre les contraintes dans les schémas conceptuels de bases de données : UML-EER », *eighth Maghrebian Conference on Software Engineering and Artificial Intelligence*, Sousse Tunisia, pp. 437, 9-12 may, 2004.

- Dartigues C., Etat de l'art sur les méthodes de fusion des ontologies, Rapport Technique N°RR02, Laboratoire d'Informatique Graphique Image et Modélisation, Université Claude Bernard Lyon 1, France, 2002
- Frédéric F., L'ingénierie ontologique, Institut de Recherche en Informatique de Nantes, Rapport de recherche No 02-07, Octobre 2002.
- Gandon F., Ontology Engineering : a survey and a return on experience, rapport de recherche n° 4396, INRIA, 2002.
- Gargouri F., Modélisation de la complexité des SI à travers la coopération par intégration de représentations conceptuelles, H.D.R, Université Tunis II, Mai 2002.
- Gomez P.A., Rojas-Amaya D., « Ontological Reengineering for Reuse», Fensel D., Studer R., Eds., *11th European Workshop on Knowledge Acquisition, Modeling and Management (EKAW-99)*, vol. 1621 de LNAI, Berlin, 26–29 1999, Springer, p. 139–156.
- Hess G., Iochpe N., Cirano., « Ontology-driven resolution of semantic heterogeneities in GDB conceptual schemas », *6th Brazilian Symposium of GeoInformatic*, 2004.
- Holt A., « Understanding environment and geographical complexities through similarity matching ».. In *Complexity International*, number 7, 2000.
- Kraft B., Nagl M., « Visual knowledge specification for conceptual design: Definition and tool support », *Advanced Engineering Informatics* number 21, 67–83, 2007.
- Jiang J., Conrath D., « Semantic Similarity Based in Corpus Statistics and Lexical Taxonomy », *International Conference Research in Computational Linguistics*, Taiwan, 1997.
- Mhiri M., Mtibaa A., Gargouri F., « Towards an approach for building information systems'ontologies », *1st workshop Formal Ontologies Meet Industry*, 9-10, June, Verona, Italy, 2005.
- Mhiri M., Mtibaa A., Gargouri F., « UMLOnto: Towards a language for the specification of information systems'ontologies », *the Seventeenth International Conference on Software Engineering and Knowledge Engineering*, Taiwan, China, 14-16 July, 2005, p.743-746.
- Mhiri M., Gargouri F., Benslimane D., « Détermination automatique des relations sémantiques entre les concepts d'une ontologie », *INFORSID'2006*, Hammamet, 31 Mai au 3 juin 2006.
- Mhiri M., Gargouri F., « Using ontologies to resolve semantic conflicts in information systems design », *The first International Conference on Information and Communication Technology & Accessibility* April 12-14, 2007, Hammamet – Tunisia.
- Rousse G., Vers une ontologie de la systématique, les objets-métier du domaine, thèse de doctorat, Laboratoire Informatique et Systématique, université pierre de marie curie, Paris 6, 2003.