# Construction semi-automatique d'ontologies à partir de DTDs relatives à un même domaine

G. Giraldo* et C. Reynaud*†

*L.R.I., Université Paris-Sud - C.N.R.S. (L.R.I.) & INRIA (Futurs), Bât. 490, 91405 Orsay cedex
http://www.lri.fr/~cr ; e-mail : cr@lri.fr

† Université Paris X-Nanterre, 200 av. de la République, 92001 Nanterre cedex

**Résumé**

Un système médiateur se compose d'un schéma global, ou ontologie, établissant une connexion entre les sources d'information interrogées et contenant le vocabulaire utile aux utilisateurs pour exprimer des requêtes. L'objectif de cet article est de proposer des techniques pour construire le plus automatiquement possible une telle ontologie pour un serveur d'information relatif à un domaine d'application donné, regroupant un ensemble de sources d'information XML. L'ontologie dont nous cherchons à automatiser la construction est un schéma à base de classes. L'approche proposée exploite les DTDs associées aux sources d'information XML intégrées. L'article présente l'approche adoptée et décrit les résultats issus des premières expérimentations réalisées.

**Mots clés :** Ontologie, intégration d'information, Documents XML, DTDs, extraction et structuration de connaissances.

## 1 Introduction

Un système d'intégration d'information (ou système médiateur) permet à un utilisateur d'interroger des sources d'information hétérogènes et autonomes tout en simulant l'interrogation d'un système centralisé et homogène. Les sources d'information intégrées sont des sources créées indépendamment du système médiateur. Une ontologie du domaine est alors utile pour jouer le rôle de schéma global intermédiaire établissant une connexion entre chacune d'elles. Elle contient par ailleurs l'ensemble du vocabulaire utile aux utilisateurs pour exprimer des requêtes.

Le problème traité dans ce papier concerne la construction automatisée d'une ontologie au sein d'un système médiateur. A cause de l'explosion du nombre de sources d'information accessibles via le Web, le passage de l'approche médiateur à l'échelle du Web doit aujourd'hui être envisagé. Ce passage nécessite un véritable travail de recherche pour s'attaquer de façon fondamentale à certains verrous scientifiques qui sont des obstacles importants. Un de ces verrous est la construction d'ontologies. En effet, la construction manuelle d'une ontologie, même assistée par des outils conviviaux, est un travail de modélisation long et difficile. Chercher à automatiser ce processus de construction est primordial. Ceci est d'autant plus important aujourd'hui, que les ontologies sont amenées à jouer une rôle clé dans le cadre du Web Sémantique. Elles permettent de réunir un ensemble de termes partagés, précisément définis, utilisables pour la description du contenu des sources d'information accessibles via le Web. L'automatisation de leur construction permettra alors d'en faciliter l'utilisation dans des projets allant de l'intégration de quelques sources d'information à des projets développés dans le cadre du Web Sémantique.

La littérature comporte de très nombreuses définitions du terme « ontologie ». Chaque communauté adopte sa propre interprétation selon l'usage qui en est fait et le but visé. Dans [9], N. Guarino analyse plusieurs de ces interprétations dans le domaine de l'ingénierie des connaissances. Nous retiendrons, dans le cadre de notre travail la définition de Gruber [8] selon laquelle « une ontologie est une spécification explicite d'une « conceptualisation », modifiée en 1997 par Borst [3] de la façon suivante : « une ontologie est une spécification formelle d'une conceptualisation partagée ». Une conceptualisation comprend l'ensemble des entités pertinentes d'un domaine d'étude et de leurs relations. C'est une vue abstraite et simplifiée du monde que l'on veut représenter. Cette spécification est dite explicite car les types des concepts utilisés, les relations entre eux, les contraintes qu'ils doivent satisfaire et leurs propriétés sont explicitement définis. Elle est formelle car elle doit pouvoir faire l'objet de traitements machine et partagée car elle reflète un certain consensus. En règle générale, tout le monde s'accorde pour reconnaître qu'il s'agit d'une représentation d'une structuration d'un domaine, le but d'une ontologie étant de définir le vocabulaire d'un domaine pouvant être utilisé par différents acteurs (humains ou logiciels), de représenter sa signification pour faciliter la communication. Les ontologies sont ainsi utilisées en intégration d'information, notamment dans l'approche médiateur, en tant que support de

l'interface entre le système et les utilisateurs, ces derniers ayant alors l'impression qu'ils s'adressent à un système centralisé et homogène.

Notre travail se situe dans le cadre du projet PICSEL II. Nous nous intéressons à faire passer à l'échelle l'approche médiateur mise en œuvre dans le projet PICSEL I [7]. L'approche PICSEL I a eu par objectif la construction de serveurs d'information permettant d'interroger des sources d'information multiples et hétérogènes relatives à un même domaine d'application. L'objectif de notre travail, dans le cadre du projet PICSEL II, est de proposer une méthode et des techniques pour construire le plus automatiquement possible une ontologie pour un serveur d'information PICSEL regroupant un ensemble important de sources d'information XML.

Ce papier est organisé de la façon suivante. Nous présentons en section 2 le cadre de notre travail, les entrées et sorties du système logiciel que nous proposons. La méthode de construction de l'ontologie est décrite en section 3. Nous présentons et analysons les résultats issus des premières expérimentations réalisées en section 4. En section 5, nous présentons quelques travaux proches, puis nous concluons et présentons quelques perspectives en section 6.

# 2 Présentation du cadre de travail

## 2.1 De PICSEL I à PICSEL II

L'approche PICSEL I a eu par objectif la construction de serveurs d'information permettant d'interroger des sources d'information multiples et hétérogènes relatives à *un même domaine d'application*. Les serveurs d'information comportent, pour cela, un moteur de requêtes *générique* et des bases de connaissances *spécifiques* au domaine d'application du serveur. Les bases de connaissances se composent d'une description du domaine du serveur, appelée aussi *ontologie du domaine,* et des descriptions du contenu des sources d'information interrogeables. L'ontologie du domaine et le contenu des sources d'information sont décrits en utilisant le langage de représentation des connaissances CARIN [12]. Les choix en matière de représentation des connaissances ont été faits de façon à faciliter l'expression des connaissances dans le cadre d'un médiateur et pour que les traitements automatisés, effectués dans ce cadre, soient dotés de bonnes propriétés [7]. Le langage garantit, en particulier, la décidabilité du calcul complet des plans de requêtes.

La représentation d'une ontologie pour un domaine d'application réel, le domaine des produits du tourisme, a été proposée dans le projet PICSEL I, dans le formalisme logique CARIN [16]. Le résultat de ce travail sert de point de départ au travail à effectuer dans PICSEL II. Notre objectif, dans PICSEL II, est de proposer une méthode et des techniques automatisées d'aide à la construction d'une ontologie dont le modèle correspond à celui de l'ontologie PICSEL I. Nous précisons, ci-dessous, les caractéristiques du modèle auquel nous voulons aboutir, par référence au modèle représenté dans PICSEL I, puis les données à partir desquelles cette construction doit être effectuée.

## 2.2 Modèle de l'ontologie

Le modèle de l'ontologie comprend un ensemble de hiérarchies de classes qui décrivent des catégorisations de classes d'objets du domaine d'application. Dans le domaine des produits du tourisme, ce modèle comprend une hiérarchie qui représente tout ce qui peut se vendre dans le domaine du tourisme et qui regroupe les logements, les trajets, les locations, les activités, etc. Le modèle comprend également d'autres hiérarchies de classes qui décrivent des catégorisations d'ensembles d'objets de sous-domaines du domaine des produits du tourisme (lieu, loisir, prestation, service, équipement) cf. figure 1.

Chaque classe est définie au travers de ses relations avec d'autres classes. Pour une classe donnée, le modèle précise la classe qui la généralise (classe mère dans la hiérarchie) et éventuellement ses propriétés spécifiques ou bien l'ensemble des propriétés nécessaires et suffisantes d'un objet pour appartenir à cette classe. Ainsi, l'ontologie dont nous cherchons à automatiser la construction est un schéma à base de classes qui pourra être représenté dans le langage CARIN et qui pourra, de ce fait, être exploité par

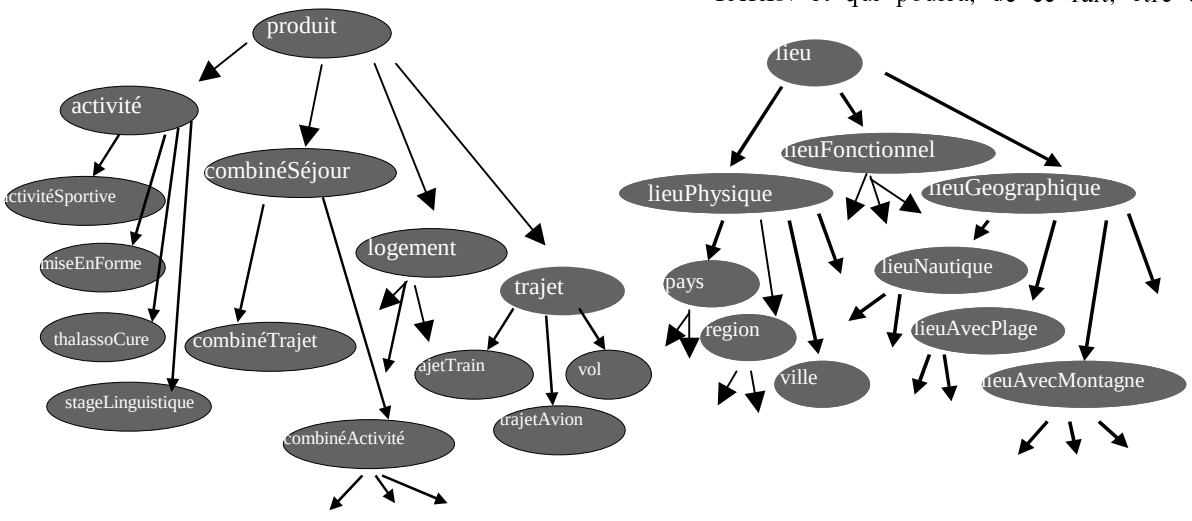


**FIG. 1.** - *Parties de hiérarchies du modèle de l'ontologie relative aux produits du tourisme*

l'ensemble des outils logiciels développés dans le cadre du projet PICSEL I, en particulier par le moteur de requêtes.

## 2.3 Documents XML

La construction de l'ontologie doit se faire à partir des données de sources d'information XML qui, dans le cadre de notre travail, sont supposées valides et auxquelles correspond une structure définie dans une DTD (Document Type Definition). Toutes les sources considérées sont homogènes par rapport au format de représentation des données puisqu'il ne s'agit que de sources XML. En revanche, elles sont hétérogènes d'un point de vue sémantique car elles ont été développées par des personnes différentes qui ont pu choisir d'utiliser des tags différents et de structurer leurs documents différemment.

Une DTD (cf. figure 2) est une représentation abstraite de documents XML qui peut être vue comme un graphe orienté (cf. figure 3) où chaque nœud du graphe correspond à un terme de la DTD, et un lien entre deux nœuds à un lien de composition entre deux termes (flèche simple) ou à un lien entre un terme et un attribut (flèche double).

```
<!ELEMENT Travel (CruiseList)>
<!ATTLIST Travel Title CDATA #IMPLIED>
<!ELEMENT CruiseList (Cruise*)>
<!ELEMENT Cruise (Destination, Amenity,
   Leaving_from, Departure_date, Return_date,
   Degriftour_Price, Public_Price)>
<!ATTLIST Cruise See_location CDATA
   #IMPLIED>
<!ELEMENT Destination (#PCDATA)>
<!ELEMENT Amenity (#PCDATA)>
<!ELEMENT Leaving_from (#PCDATA)>
<!ELEMENT Departure_date (#PCDATA)>
<!ELEMENT Return_date (#PCDATA)>
<!ELEMENT Degriftour_Price (Frs,Euro)>
<!ELEMENT Frs (#PCDATA)>
<!ELEMENT Euro (#PCDATA)>
<!ELEMENT Public_Price (Frs,Euro)>
```

***FIG. 2** - DTD Cruise*

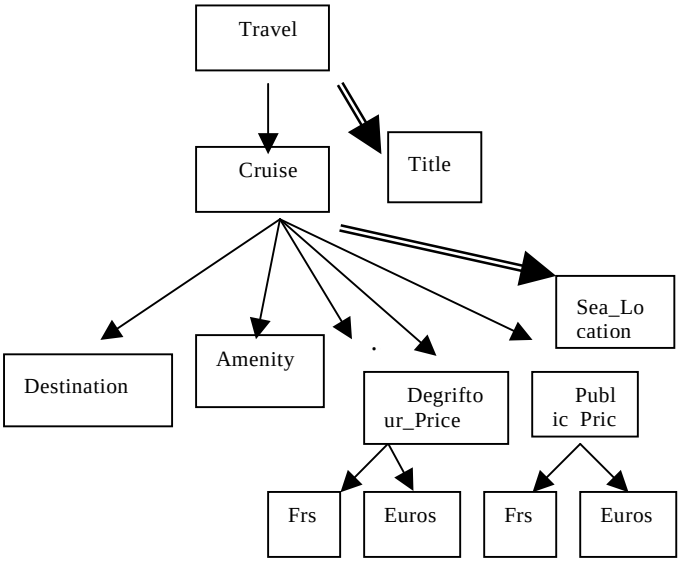


***FIG. 3** - Graphe représentant la DTD de la figure 2*

# 3 Méthode de construction automatisée d'ontologies

## 3.1 Approche générale et choix

La construction de l'ontologie devant être la plus rapide possible, les techniques que nous proposons exploitent en priorité les DTDs associées aux documents XML, non les documents eux-mêmes, de façon à réduire la masse d'information traitée.

Les DTDs sont des représentations assimilables à des *schémas* de documents, au sens des schémas de bases de données relationnelles. Ce sont des modèles *abstraits* de documents XML. Les données réelles n'y sont pas représentées. Ce niveau de représentation des données est ainsi tout à fait en accord avec la représentation d'une ontologie correspondant à une vue *abstraite* d'un domaine.

Décrivant la *structure* de documents, les DTDs ne se veulent toutefois pas des descriptions de *domaines d'application*. Le problème de la construction d'une ontologie, en tant que vue abstraite d'un domaine, ne peut ainsi résulter de la fusion de DTDs relatives à ce domaine. Les DTDs ne comportent, par ailleurs, aucune description explicite de ce que sont les classes d'un domaine, leurs propriétés, les différents types de relations entre classes. Nous proposons d'adopter une approche consistant alors non pas à fusionner des DTDs mais à extraire les classes, propriétés, relations, relatives à un certain domaine, dans des DTDs qui ne représentent pas explicitement ces éléments mais qui, néanmoins, ont été construits par des concepteurs qui avaient cette connaissance et qui l'ont utilisée dans leurs choix de représentation.

L'approche que nous proposons comporte trois phases. La première est une phase de d'extraction de composants de l'ontologie à construire : classes, propriétés, relations. La seconde phase organise les éléments acquis dans le première étape. La troisième phase représente les connaissances préalablement collectées et organisées dans le langage CARIN. Nous présentons dans les sections qui suivent les deux premières étapes de l'approche.

## 3.2 Extraction des composants de l'ontologie

La phase d'extraction s'effectue sur un échantillon de DTDs relatives à un même domaine. L'échantillon de DTDs doit être représentatif. En ce sens, il doit correspondre à un nombre pouvant être important de DTDs qui couvrent l'*ensemble* du domaine étudié. La notion de couverture du domaine ne pouvant être prouvée, là encore il est primordial que les techniques proposées soient suffisamment automatisées de façon à pouvoir être facilement et régulièrement réappliquées à partir de DTDs couvrant plus largement le domaine.

L'objectif de la phase d'extraction est de découvrir les classes du domaine, leurs propriétés et les relations entre classes : relations de généralisation

/spécialisation, relations spécifiques au domaine. Nous décrivons ci-dessous le processus d'extraction proprement dit des classes, propriétés, relations puis les traitements opérés sur les données brutes extraites de façon à obtenir un vocabulaire approprié à l'ontologie à construire.

**Extraction des termes associés à des classes, des propriétés et des relations**

La phase d'extraction présentée dans cette section est entièrement automatique. Elle exploite la structure des DTDs. L'idée de base est la suivante. Une classe est vue comme une représentation abstraite d'un ensemble d'objets auxquels on s'intéresse dans un domaine donné, ce qui signifie qu'ils ont des caractéristiques, ou propriétés, dont il est intéressant de connaître les valeurs. De ce fait, nous faisons l'hypothèse que les termes associés à des classes dans les DTDs, sont repérables par le fait qu'ils ont des propriétés, représentées dans les DTDs par des éléments qui les composent. Etant donné que l'approche repose sur l'exploitation d'un ou de DTDs représentatifs du domaine, ces composants devraient apparaître *au moins dans un* des DTDs de l'échantillon. En suivant ce raisonnement, nous en déduisons une méthode de repérage des termes associés aux propriétés : les termes associés à des éléments qui ne sont *jamais* décomposés dans *aucun* des DTDs. Le principe utilisé trouve sa justification dans le fait que les DTDs exploitées sont supposées représentatives du domaine à couvrir. La mise en oeuvre du principe décrit ci-dessus s'effectue par application des heuristiques suivantes :

$H_C$ : les classes sont désignées par les termes correspondant à au moins un nœud non feuille, dans au moins un graphe des DTDs.

$H_P$ : les propriétés sont désignées par les termes correspondant toujours uniquement à des nœuds feuilles, dans tous les graphes des DTDs où ces termes apparaissent.

$H_{R1}$ : une expression de type ELEMENT décrivant la décomposition d'un terme-classe A en plusieurs sous-termes-classes B, C, ... traduit une relation *spécifique au domaine* entre les classes A et B, A et C, etc.

$H_{R2}$ : une expression de type ATTLIST définissant les termes–propriétés P1, P2, etc. d'un terme-classe C traduit une relation de *caractérisation* entre la classe C et les propriétés P1, P2, etc..

$H_{R3}$ : une expression de type ELEMENT définissant la composition d'un terme-classe A comme une disjonction d'autres termes-classes (B, C, ...) traduit une relation de *spécialisation* entre les classes A et B, A et C, etc. (B est une classe plus spécifique que A, C est une classe plus spécifique que A, etc.)

**Traitement des termes bruts collectés**

Des termes « non signifiants » dans une ontologie sont utilisés comme tags dans les DTDs. Ces termes sont signifiants pour la personne qui a conçu la DTD et pour celles qui ont créé les documents instances des DTDs qui les contiennent. Cependant, ces termes peuvent ne pas faire partie du vocabulaire utile à la formulation de requêtes.

Les problèmes, qui se posent, sont les suivants. Les termes extraits peuvent correspondre à des mots composés qu'il est parfois utile de faire apparaître séparément dans l'ontologie. Les différents mots sont en général identifiables par la présence d'une majuscule en première lettre ou parce qu'ils sont séparés par un caractère spécial (tiret, souligné, etc.). Certains des mots, contenus dans des mots composés ou directement associés à des tags, correspondent à des abréviations qu'il serait utile d'expliquer. Exemple : Srvc pour service ou Pref pour préférences. Certains termes ne sont pas spécifiques d'un domaine d'application et ne doivent pas figurer parmi les termes retenus pour faire partie de l'ontologie. Ils sont dits « non pertinents » pour l'ontologie. Exemple : Set, List, Collection. Enfin, des termes extraits peuvent être considérés comme différents simplement parce que leur nombre (singulier ou pluriel) est différent. Exemple : access, accesses.

L'extraction des termes-classes et des termes-propriétés s'accompagne alors de traitements éliminant les doublons (cas de termes présents au singulier et au pluriel), extrayant les différents mots d'expressions composées, remplaçant les abréviations par leur signification en clair (en s'aidant de définitions) et éliminant certains termes « non pertinents » pour l'ontologie. Ces traitements exploitent d'une part des fichiers contenant les explications des abréviations et la liste des termes non pertinents. D'autre part, ils exploitent un thésaurus (WordNet [15]) pour identifier les termes pluriels et les termes composés retenus en tant que classes pour le domaine d'étude.

## 3.3 Structuration des éléments composant l'ontologie

Les éléments de l'ontologie étant identifiés (classes, propriétés, relations), il est ensuite nécessaire de proposer une organisation de ces éléments selon un modèle de classes. Proposer une telle organisation n'est pas immédiat. En effet, des choix de structuration différents ont pu être faits dans les DTDs et les relations de généralisation/spécialisation extraites reflètent l'ensemble de ces choix. Face à plusieurs solutions (incompatibles) possibles, il est difficile de déterminer de manière automatique la structure à retenir dans l'ontologie à construire.

Pour résoudre ce problème, nous proposons une approche semi-automatique. Nous demandons au concepteur du système de fournir une ébauche de

hiérarchie de classes simple, appliquée au domaine d'étude. Ceci a pour but de fixer le point de vue à adopter pour structurer les classes au sein de l'ontologie. Cette fraction de hiérarchie sert de point de départ à l'organisation de l'ensemble des classes collectées préalablement (cf. section 3.2.). Il s'agit ensuite de rechercher, parmi l'ensemble des classes extraites, celles à faire apparaître dans cette hiérarchie, de façon à la compléter. Les classes restantes doivent être organisées dans d'autres hiérarchies.

La conception de la hiérarchie initiale est effectuée par le concepteur du système qui définit les classes à représenter en fonction, par exemple, des grands thèmes sur lesquels les requêtes des utilisateurs pourront porter et qui, pour cela, peut s'inspirer d'ontologies du domaine déjà existantes.

La hiérarchie initiale est complétée en exploitant les relations de généralisation/spécialisation extraites de l'échantillon de DTDs. Le problème de structuration correspond alors à un problème de fusion de hiérarchies partielles extraites automatiquement avec une ébauche de hiérarchies de classes construites manuellement. Des thésaurus linguistiques munis de relations sémantiques entre termes, tels WordNet, sont utilisés pour faciliter ce processus de fusion, c'est-à-dire identifier les synonymes et les relations de généralisation/spécialisation entre les classes extraites et les classes de la hiérarchie initiale construite manuellement. Les classes qui ne peuvent être placées dans la hiérarchie initiale sont des éléments candidats pour faire partie d'autres hiérarchies. Les techniques

pour construire ces autres hiérarchies ne seront pas décrites dans cet article. Elles sont actuellement en cours d'étude.

# 4 Expérimentation

Le domaine des produits du tourisme a été choisi pour l'expérimentation et un prototype a été développé en Java. Comme nous l'avons dit en section 3.1, notre approche se compose de trois phases : la phase d'extraction des classes, propriétés et relations, la phase de structuration de ces éléments et la phase de représentation. Les résultats présentés dans cette section ne concernent que la phase d'extraction des termes classes et des termes propriétés. Par ailleurs, les données d'entrées dont nous disposons étant en anglais, nos résultats portent sur une extraction de termes en anglais.

## 4.1 Données

Les données d'entrées utilisées sont composées de DTDs fournies par l'OTA ou « **O**pen **T**ravel **A**lliance » (www.opentravel.org). L'OTA est un consortium qui regroupe plus de 150 organisations relatives à l'industrie du voyage : des agences de voyages, des hôtels, des agences de locations de voitures, des compagnies aériennes, etc. En association avec la DISA (Data Interchange Standards Association), cet organisme a développé des standards de communication basés sur XML pour faciliter l'emploi du commerce électronique. Parmi ces standards, on trouve 15 DTDs (317 lignes) permettant de décrire des documents portant sur *les besoins et les préférences* de voyageurs pour des voyages d'affaires, des vacances, des voyages internationaux, des voyages en avion, des locations de voitures, des séjours en hôtel, etc. Dans un deuxième temps, l'OTA a publié 77 « XML-Schema » (5717 lignes). Les descriptions fournies reprennent celles contenues dans les DTDs précédemment élaborées en les complétant. On y trouve notamment des descriptions de *réservations* de vols et d'hôtels ainsi que des descriptions de *locations* de voitures. L'expérimentation que nous avons menée porte sur les deux séries de données de l'OTA. Les descriptions fournies sous forme de XML-Schema ont été traduites sous forme de DTDs, grâce au logiciel « XML Spy » (www.xmlspy.com). Notre expérimentation a porté sur ces deux séries de données, la série 1 correspond aux DTDs de 317 lignes et la série 2 aux DTDs obtenus à partir des 77 XML-Schema. Nous avons, par ailleurs, construit un premier fichier d'abréviations donnant la signification de quelques abréviations utilisées dans le domaine des produits du tourisme, ainsi qu'un fichier de termes considérés comme non pertinents comprenant, entre autres, les termes : Collection, List, Preferences, Set. (cf. section 3.2.2.).

| | Avant traitement | | Après traitement | |
|---|---|---|---|---|
| | CLASSES | PROPRIETES | CLASSES | PROPRIETES |
| Serie 1 | 68 | 167 | 61 | 152 |
| Serie 2 | 468 | 851 | 389 | 841 |

**Tableau 1** : *Nombre de termes extraits par catégorie*

| | |
|---|---|
| CLASSES | OTA_AirBookRS, AirItinerary, AirlinesPref, CreditCard, Customer, HotelChainPref, HotelPref, TravelClub, VehicleRentalPref, BookingClassAvail, Traveler, Amenity, Attraction, Renovation, Restaurant, RoomAmenities, Destination, Package, Vehicle |

**Tableau 2.1** - *Quelques exemples de termes classes « bruts » extraits*

| | |
|---|---|
| CLASSES | air book, air itinerary, airline, credit card, customer, hotel chain, hotel, travel club, vehicle rental, booking class availability, traveler, amenity, attraction, renovation, restaurant, room amenities, destination, package, vehicle, … |
| PROPRIETES | phone, phone extension, phone number, payment type, personal service description premium cocktails fee, price, program code, program description, program name program restrictions, promo code, promotion coupon, promotion description, promotion code, property amenity, property class, property location, property name, property system name, property type, pseudo city code, quantity, … |

**Tableau 2.2** - *Liste partielle de termes extraits (après traitement)*

| N° série | Termes classes (nombre, %) | Termes propriétés (nombre, %) |
|---|---|---|
| 1 | 61 termes soit 61/213 = 28% | 152 termes soit 152/213 = 71% |
| 2 | 389 termes soit 389/1230 = 31% | 841 termes soit 841/1230 = 68% |

**Tableau 3** - *Termes classes et de termes propriétés extraits, en nombre et en pourcentage*

| N° série | Nombre de termes classes avant traitement | Nombre de termes classes après traitement | Termes classes modifiés par les traitements | Termes classes non modifiés par les traitements |
|---|---|---|---|---|
| 1 | 68 | 61 | 47 (47/68 = 69%) | 21 (21/68 = 31%) |
| 2 | 468 | 389 | 396 (396/468 = 84%) | 72 (72/468 = 16%) |

**Tableau 4** : *Impact des traitements sur les données brutes extraites*

## 4.2 Résultats obtenus

Nous présentons, dans le tableau 1, le nombre de termes-classes et de termes-propriétés, extraits de façon entièrement automatique, par le logiciel, à partir des deux séries de données. La partie gauche du tableau donne le nombre de termes bruts, la partie droite donne le nombre de termes après traitements (cf. section 3.2.2.). Nous présentons, dans le tableau 2.1, quelques termes-classes bruts extraits et dans le tableau 2.2, une liste partielle des termes-classes et des termes-propriétés extraits après traitement.

## 4.3 Analyse des résultats

L'approche permet de bien isoler les termes correspondant à des classes (environ 30 %) des termes correspondant à des propriétés (cf. tableau 3). L'ontologie construite étant un modèle de classes, sa structuration s'effectue autour des classes du domaine. Une telle approche nous permet alors de réduire considérablement le nombre de termes à considérer lors de la construction des hiérarchies de classes.

Les traitements effectués sur les termes bruts extraits, bien que très simples, sont assez efficaces (cf. tableau 4). En effet, les ensembles de termes après traitement et avant traitement sont très peu différents de par le nombre de leurs éléments (68 et 61 termes classes pour la série 1, 468 et 389 termes classes pour la série 2), mais ils ne contiennent que très peu d'éléments identiques (31% pour la série 1, 16 % pour la série 2). 69 % des termes classes de la série 1 (respectivement 84 % pour la série 2) ont été modifiés par les traitements effectués. Ces derniers permettent l'obtention de termes plus pertinents que les termes bruts extraits, tout en n'augmentant pas le volume des données.

Certains des traitements effectués sur les données brutes reposent sur l'utilisation d'un thésaurus en ligne, qui s'est avérée assez efficace. L'interrogation de Wordnet s'avère en particulier très utile pour reconnaître les termes au pluriel et pour indiquer le terme singulier correspondant. Ainsi, dans la série 2, 87 termes, soit 22% des termes-classes bruts, ont été identifiés par WordNet comme étant des pluriels. Certains de ces mots existaient déjà au singulier parmi la liste des termes classes extraits. Ils ont donc été supprimés. Par ailleurs, WordNet est utile dans les traitements portant sur les termes composés. Ces derniers ne doivent pas tous être décomposés en termes « élémentaires », certaines expressions composées pouvant correspondre à un concept du domaine. WordNet renseigne sur les expressions composées à conserver et à associer à des classes.

Ainsi, par exemple, lors des traitements effectués sur la série 2, WordNet a reconnu les expressions composées suivantes : bank account, credit card, guest room, phone number, reference point, postal code, company name, given name, insurance company, due date, seating capacity.

Nous nous sommes, par ailleurs, livrés à un petit exercice, consistant à extraire manuellement les termes classes et propriétés à partir des données de la série 1. Le tableau 5 fournit le nombre des termes-classes et de

| | |
|---|---|
| *Termes Classes* | |
| Nombre de termes classes extraits manuellement | 19 |
| Nombre de termes classes extraits automatiquement | 61 |
| Nombre de termes classes extraits manuellement et non automatiquement<br>Nombre de termes classes extraits manuellement et non automatiquement / Nombre de termes classes extraits manuellement | 4<br>21 % |
| Nombre de termes classes extraits automatiquement et non manuellement<br>Nombre de termes classes extraits automatiquement et non manuellement /<br>Nombre de termes classes extraits automatiquement | 46<br>75 % |
| Nombre de termes extraits automatiquement et manuellement<br>Nombre de termes extraits automatiquement et manuellement / Nombre de termes classes extraits manuellement | 15<br>79 % |
| *Termes Propriétés* | |
| Nombre de termes propriétés extraits manuellement | 160 |
| Nombre de termes propriétés extraits automatiquement | 152 |
| Nombre de termes propriétés extraits manuellement et non automatiquement<br>Nombre de termes propriétés extraits manuellement et non automatiquement /<br>Nombre de termes propriétés extraits manuellement | 35<br>22 % |
| Nombre de termes propriétés extraits automatiquement et non manuellement<br>Nombre de termes propriétés extraits automatiquement et non manuellement /<br>Nombre de termes propriétés extraits automatiquement | 27<br>17 % |
| Nombre de termes extraits automatiquement et manuellement<br>Nombre de termes extraits automatiquement et manuellement / Nombre de termes propriétés extraits manuellement | 125<br>78 % |

**Tableau 5** : *Extraction manuelle comparée à une extraction automatique*

termes-propriétés trouvés dans le cas d'une extraction manuelle et rapproche ces données quantitatives des résultats obtenus à partir d'une extraction complètement automatisée. Selon les chiffres, 79 % des termes extraits manuellement l'ont également été de façon automatique, ce qui signifie, un nombre de termes « oubliés » par le processus automatique d'environ 21 %. Par ailleurs, on remarque un nombre relativement important de termes classes extraits automatiquement et non pertinents (75 %) alors que le même phénomène ne se produit pas pour les termes propriétés (17 %).

Ces résultats s'expliquent de la façon suivante. Tout d'abord, tous les oublis de termes classes correspondent à des termes qui figurent parmi l'ensemble des termes classes extraits automatiquement mais sous un autre nom. En effet, manuellement, le concepteur interprète la signification des termes extraits des DTD et déduit les terme-classe du domaine qui lui semblent devoir être représentés dans l'ontologie. Mais le logiciel n'a pas toutes ces capacités d'interprétation. De cette façon, par exemple, RelatedTraveler fait partie des termes-classes extraits automatiquement alors qu'à la main, le concepteur a extrait Traveler. On observe le même phénomène pour les termes propriétés. Par exemple, le processus automatique extrait params et air equip alors que le processus manuel extrait parameter et equipment.

L'importance des termes non pertinents provient de la méthode d'extraction utilisée et également de la nature des DTDs utilisées en entrées de notre système. Les techniques utilisées conduisent à considérer que tout terme décomposé dans une DTD correspond à un terme-classe.

Dans la pratique, il existe des exceptions. Ainsi, le terme address, bien qu'étant souvent décomposé en street number, building room, address line, city name, state prov, country name, correspond davantage à une propriété composée qu'à une classe. Nous rencontrons le même problème pour person name qui est décomposé en name prefix, given name, middle name, surname, name suffix, name title. Par ailleurs, les DTDs que nous exploitons sont proposés par l'OTA pour décrire toutes sortes de documents liés au tourisme. Parmi les descriptions proposées, figure, par exemple, toute une partie spécifique au commerce électronique. On y trouve les termes control, session, sendBy, warnings, errors. Ces termes ne font pas partie du vocabulaire du domaine du tourisme utile à des utilisateurs pour exprimer des requêtes dans le domaine. Ils n'ont pas été retenus par le processus d'extraction manuelle. Les DTDs de l'OTA comptent de très nombreux termes de ce type.

Ces premiers résultats sont encourageants même si de meilleurs résultats auraient peut-être pu être obtenus si nous avions exploité des DTDs encore plus proches du domaine de l'ontologie à construire.

Notre objectif est d'élaborer un processus de construction d'ontologies semi-automatique. Le travail d'expérimentation présenté dans cet article permet de mieux préciser la part des traitements qui peuvent être entièrement automatisés et le travail du concepteur dans la phase d'extraction de classes et de propriétés. Il ressort de l'analyse effectuée qu'il est tout à fait envisageable de débuter par une extraction entièrement automatique puis de définir de nouvelles abréviations dans les fichiers prévus à cet effet. Ceci est très important car, suite à ces modifications, le nombre de termes-classes non pertinents extraits automatiquement peut fortement diminuer. Une fois la liste des termes non pertinents et la liste des abréviations arrêtée, le concepteur analyse les résultats produits par le logiciel. Cela peut le conduire à éliminer des termes, modifier certains noms, transformer certains termes-classes en termes-propriétés. Il prend ainsi en charge les interprétations spécifiques de termes à effectuer et le traitement des exceptions (au niveau des classes). Nous envisageons de tirer parti de l'étude présentée dans cet article pour définir précisément la façon dont le concepteur peut être guidé dans cette tâche (ex : présentation de termes classés selon que certains ont été ou non reconnus par WordNet comme des termes ayant un sens, etc.).

# 5 Travaux proches

Dans les dernières années, les ontologies ont eu une importance considérable. Des méthodes de construction d'ontologies à partir d'analyses de corpus de textes disponibles [1], [13] ainsi que des techniques de réutilisation d'ontologies [5] ont été proposées. Dans le cadre de l'intégration d'information, on distingue deux types de travaux. Certains ont été fortement influencés par les résultats obtenus dans le domaine des bases de données fédérées considérant qu'il s'agit d'une fusion d'ontologies locales. D'autres se sont inspirés des travaux effectués en Apprentissage Automatique [17], [11], [6]. Ces derniers se donnent pour objectif de construire une représentation en accord avec un ensemble d'ontologies locales (c'est-à-dire cohérente et consistante) mais dont l'objectif n'est pas forcément d'y retrouver l'intégralité de leur contenu.

Notre approche débute par une phase d'extraction de termes et s'apparente, de par l'objectif recherché, aux travaux cités ci-dessus en apprentissage automatique. Néanmoins, la méthode retenue est particulière. Elle exploite la structure des documents en entrée du système. Une fois les classes, propriétés et relations extraites, le problème consiste ensuite à intégrer des hiérarchies partielles de classes extraites automatiquement et une ébauche de hiérarchie de classes construite manuellement. La nature du problème change alors et se rapproche des problèmes traités dans les approches Bases de Données Fédérées. Parmi les techniques utilisées dans ces approches, on peut citer celles qui exploitent les services inférentiels attachés au langage de représentation des connaissances [2], les techniques basées sur l'utilisation d'heuristiques [10], sur l'exécution d'opérateurs de transformations [4] ou les interfaces intelligentes assistant l'utilisateur dans la fusion d'ontologies [14]. Notre approche se distingue des précédentes dans la mesure où les techniques utilisées sont principalement basées sur l'exploitation de connaissances linguistiques (extraites actuellement du thésaurus WordNet) qui guident le processus de fusion.

# 6 Conclusion et perspectives

Dans cet article, nous avons décrit une approche de construction semi-automatique d'une ontologie utilisable dans le cadre d'une approche médiateur et les résultats issus des premières expérimentations réalisées concernant la phase d'extraction.

La phase d'extraction est aujourd'hui opérationnelle, de même que la phase de « fusion » des hiérarchies de classes partielles extraites et de la hiérarchie initiale construite manuellement. L'étape de construction des autres hiérarchies de classes rassemblant toutes les classes ne pouvant se greffer sur la hiérarchie initiale est en cours d'étude. Elle nécessite la mise en oeuvre de mécanisme de clustering de façon à identifier les classes pouvant être regroupées au sein d'une même hiérarchie, puis d'identifier les relations existant entre classes.

# Références


[1] N. AUSSENAC, B. BIEBOW, et S. SULZMAN. Revisiting Ontology Design: A method Based on Corpus Analysis. *EKAW, LNAI 1937*, pages 172-188, 2000.

[2] S. BERGAMACHI, S. CASTANO, M. VINCINI et D. BENEVENTANO. Intelligent Techniques For Extraction And Integration of Heterogeneous Information. *IJCAI*, 1999.

[3] W. N. BORST. Construction of Engineering Ontologies, *PhD Thesis*, University of Twente, Enschede, 1997.

[4] H. CHAPULSKY. OntoMorph : A Translation System for Symbolic Knowledge, *Principles of Knowledge representation and reasoning: Proceedings of the seventh International Conference* (*KR2000*), San Francisco, CA : A. G. Cohn, F. Giunchiglia F., Selman B., editors, Morgan Kaufmann Publishers, 2000.

[5] P. CLARK, J. THOMPSON, H. HOLMBACK, L. DUNCAN. Exploiting Thesaurus-Based Semantic net for Knowledge-Based Search, *AAAI*, pages 988-995, 2000.

[6] A. DOAN, P. DOMINGOS, A. LEVY. Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 509-520, Santa Barbara, CA: ACM Press, 2001.

[7] F. GOASDOUE, V. LATTES, M-C. ROUSSET. The Use of CARIN language and algorithms for information integration: the PICSEL Project, *International Journal of Cooperative Information Systems*, Vol. 9 N°4, pages 383-401 , 2000.

[8] T.R GRUBER. A Translation Approach to portable Ontology specifications, *Knowledge Acquisition*, 5(2), pages 199-220, 1993.

[9] N. GUARINO. Formal Ontology, Conceptual Analysis and Knowledge Representation, *Int. J. Human-Computer Studies*, 43, pages 625-640, 1995.

[10] E. HOVY. Combining and Standardizing large-Scale, practical Ontologies for Machine Translation and Other Uses, *Proc. 1st Intl. Conf. On language Resources and Evaluation,* Granada, 1998.

[11] E. JEONG, C-N. HSU. Integration and Reuse of Heterogeneous XML DTDs For Information Agent. *In proceedings the Second Asia-Pacific Conference on Intelligent Agent Technology* (IAT), 2001.

[12] A. LEVY, M-C. ROUSSET. Combining Horn Rules and Description Logics in CARIN, *Artificial Intelligence*, n°104,, pages 165-209, 1998.

[13] A. MAEDCHE, S. STAAB. Mining Ontologies from Text, *EKAW*, *LNAI 1937*, pages 189-202, 2000.

[14] D. L. Mc GUINENESS., R. FIKES, J. RICE, S. WILDER. An environment for Merging and Testing Large Ontologies, *Proc. Of KR'00*, pages 483-493, 2000.

[15] G.A. MILLER. WordNet: A Lexical Database for English, *Communications of the ACM*, Vol. 38, N°11, pages 39-41, 1995.

[16] C. REYNAUD, B. SAFAR, H. GAGLIARDI. Une expérience de représentation d'une ontologie dans le médiateur PICSEL, *Conférence IC2001*, Plateforme AFIA, pages 329-348 Grenoble, 2001.

[17] M. SINTEK, M. JUNKER, L. ELST, A. ABECKER. Position Statement for IJCAI-2001 Workshop on Ontology Learning, 2001