Extraction et Intégration d'Informations Semi-structurées dans les pages Web - Projet Chimère

Marie-Sophie Segret, Pierre Pompidor, Danièle Hérin

Extraction et Intégration d'Informations Semi-structurées dans les pages Web - Projet Chimère ⋆

Marie-Sophie Segret, Pierre Pompidor, Danièle Hérin

L.I.R.M.M.
Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier,
161 rue Ada, 34392 - Montpellier Cedex 5
segret@lirmm.fr, pompidor@lirmm.fr, dh@lirmm.fr

Résumé

Ce papier se situe dans le cadre du projet Chimère dont objectif est de faciliter l'accès à plusieurs serveurs d'information spécifiques à un domaine donné et dont la particularité est de délivrer des informations extraites de bases de données après que l'utilisateur ait rempli des formulaires. Les pages de tels serveurs incluent ces formulaires qui se composent de zones de saisies et de données textuelles apparaissant à proximité immédiate de ces zones. Ces types de pages sont très fréquents dans le domaine de la réservation de moyens de transport (ex. Air France, SNCF) et dans le commerce électronique.
Un tel contexte pose deux problèmes à résoudre : i)l'extraction des informations de telles pages "formulaires" en combinant les données structurées et les "brèves" données textuelles, ii) l'intégration et l'exploitation des informations extraites de différents sites et répondant à un même service, sachant que les problèmes d'hétérogénéité doivent être transparents pour l'utilisateur. Pour traiter i) nous effectuons une analyse des pages "formulaires" à partir d'une ontologie du domaine et d'une analyse syntaxique et sémantique de texte. ii) est un problème de modélisation de la partie de chaque site qui relève du domaine en utilisant un vocabulaire et un langage identique pour tous les sites concernés.
Nous avons retenu une approche incrémentale consistant à développer dans un premier temps un prototype minimal qui montre la faisabilité de l'approche retenue.

Mots clef : web; ontologies ; intégration d'informations ; ingénierie des connaissances.

Thèmes:
6. Intranet/Internet et Ingénierie des Connaissances
5. Systèmes d'Information et Ingénierie des Connaissances

1 Introduction

La grande majorité des serveurs (en particulier commerciaux) reposent sur l'exploitation de bases de données qui, en général, ne sont accessibles qu'à travers des formulaires HTML ou XML. Or, le besoin croissant d'automatiser l'usage du web requiert l'appréhension, même partielle, de la structure des vues de ces bases. Nous avons à traiter des informations hétérogènes relevant de différents sites et comprenant, d'une part des informations structurées mais dont la structure est inconnue a priori, et d'autre part des informations semi-structurées [2] où la structure (sous forme par exemple d'éléments ou d'attributs XML) est directement associée aux données. Nous appellerons informations semi-structurées des informations construites à partir des données structurées et de l'analyse de ces "textes brefs" qui typiquement composent les formulaires.
L'extraction et l'intégration d'informations issues du Web a donné lieu à de nombreux travaux de recherche. La partie modélisation et intégration a été étudiée, par exemple, dans les projets Information Manifold [20], OBSERVER [14], SIMS [4], TSIMMIS [7], Ontobroker [11], Picsel [13]). Notre approche consiste à modéliser le domaine d'application par la construction d'une ontologie globale, à extraire la structure des pages qui composent les sites (notamment celles des formulaires), à modéliser le contenu des sites à partir de l'ontologie globale et des informations extraites. L'originalité de notre approche repose d'une part sur la spécificité de notre problématique, l'analyse des formulaires composés de zones de saisie qui nous permettent de définir la structure des bases de données sous-jacentes; d'autre part sur l'approche que nous avons retenue pour résoudre cette problématique. Cette approche repose sur cinq points :

– Les données des serveurs sont modélisées à partir d'ontologies du domaine, ce qui permet de gérer l'interopérabilité sémantique et d'assurer la cohérence entre les données des différents serveurs.

– Les pages web sont analysées, d'une part en analysant les données à saisir et les informations résultantes issues des bases de données, et d'autre part en effectuant l'analyse sémantique des informations textuelles qui apparaissent sur les pages à proximité immédiate de ces données. C'est la conjonction des résultats de ces deux analyses qui constitue nos données semi-structurées.

– Les données semi-structurées et les ontologies sont stockées dans des agents informationnels à l'aide d'une logique de description.

⋆ Marché Université Montpellier II - CNET n° 981B129

– La gestion et la manipulation de ces agents informationnels permettra de résoudre les requêtes des utilisateurs.
– Un apprentissage est effectué en cours de fonctionnement du système, au fur et à mesure de l'enrichissement des agents informationnels, pour réviser la modélisation du domaine d'application.

Nous avons développé un prototype conçu de manière incrémentale. La version 1, Chimère V1 est actuellement opérationnelle. Elle n'inclut pas la fonction d'apprentissage. Nous avons retenu la planification de voyages comme domaine d'application, qui inclut la prise en compte des opérateurs de voyages (SNCF [1], Air France [2], ...), et des chaînes hôtelières (Ibis [3], ...).

Lorsqu'on traite pour la première fois d'un domaine d'application, on suppose qu'une ontologie a été construite par les experts du domaine. Le fait qu'elle ne soit pas complète n'est pas très génant puisque l'un de nos objectifs est de faire évoluer cette ontologie que nous appelons ontologie globale. Dans le cadre de notre prototype, nous avons construit manuellement une telle ontologie sur le domaine des voyages. Cette ontologie décrit les concepts du domaine et recense les contraintes sur les domaines de valeur de ces concepts.
Le processus d'extraction et d'intégration des informations dans les pages Web se décompose en deux étapes :

- Analyse syntaxique et sémantique. Cette étape, entièrement automatisée, permet d'identifier les principaux serveurs et d'importer des pages XML d'un serveur donné. Pour chaque page, une analyse syntaxique et une analyse sémantique du code XML sont réalisées. L'analyse syntaxique permet d'identifier des formulaires en analysant leur entête, les listes déroulantes, les zones de saisie et des séries de boutons radio. L'analyse sémantique consiste à analyser le titre de la page web, les noms des variables (en identifiant les concepts associés) et le texte qui les accompagne, à sauvegarder le contexte sémantique de la page et à renseigner le formulaire.

- Enrichissement de la base de connaissances. Cette étape, entièrement automatisée, permet de décrire le contexte sémantique des pages du serveur par la construction d'une ontologie locale comprenant les concepts et instances trouvés dans les pages.

Lorsqu'un utilisateur pose une requête, le système recherche des ontologies locales satisfaisantes. A chaque ontologie locale sélectionnée, les serveurs correspondants sont connectés et les formulaires sont renseignés de manière automatique. Les réponses des différents serveurs sont comparées et la proposition la plus intéressante est retournée à l'utilisateur.

Ce papier décrit d'une part la partie analyse du texte des pages du système d'extraction des informations (section 2), qui permet d'alimenter la base de connaissances; d'autre part nous présenterons la modélisation des connaissances extraites des pages web et par là même la structure de la base de connaissances, en utilisant le formalisme UML et le langage de représentation Classic (section 3).

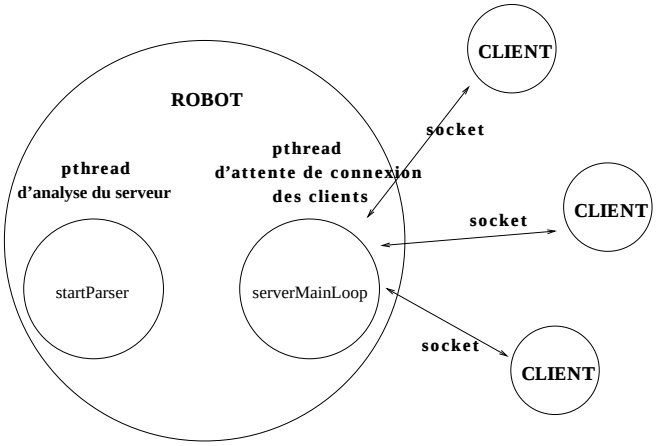
1. http://www.sncf.fr
2. http://www.airfrance.fr
3. http://www.ibis.tm.fr

2 Extraction des informations dans les pages Web

Pour l'instant, le processus d'extraction des informations dans les pages web se nourrit essentiellement de l'analyse simultanée des données et des informations textuelles qui composent et environnent les formulaires. Cette focalisation sur les formulaires ne nous a pas donné l'occasion de travailler sur des pages écrites en XML, les serveurs relatifs à notre problématique (opérateurs de voyage, prestataires hôteliers, ...) n'en proposant point. Les exemples donnés ci-dessous sont donc tirés de pages HTML, ce qui n'occulte en rien le fait qu'une structuration "XML" des informations (textuelles ou non textuelles) qui y apparaissent, faciliterait grandement notre travail.
Avant d'effectuer l'analyse des pages, celle-ci sont importées à l'aide d'un robot, décrit succinctement en section 2.1, la phase d'analyse des informations textuelles étant elle décrite de manière plus détaillée en section 2.2.

2.1 Importation des pages Web

Le robot se décompose en deux processus : l'un qui réalise l'importation les pages Web d'un site donné, l'autre qui permet à un éventuel "espion" (client Java) de se connecter sur le robot pour être informé en temps réel de ce qu'il fait.



Après avoir réussi l'importation d'une page Web, le robot analyse syntaxiquement cette page au moyen d'un analyseur

syntaxique (*parser*) programmé à cet effet et retient du code HTML/XML, uniquement la structure des formulaires :

- le nom de la passerelle CGI appelée,
- le type de la méthode HTTP employée (GET ou POST),
- et pour chaque variable associée à une zone de saisie, une liste déroulante ou une série de boutons radio : son nom, son type, sa (ses) valeur(s) par défaut et les informations textuelles affichées sur la page Web.

Cette analyse syntaxique est illustrée ci-dessous sur la page *http://voyages.sncf.fr/reservation*, par le détail de la transcription d'une variable de sélection d'un item dans une liste déroulante :

```
<select name="choix_localite_arrivee" size="1"
   onChange="input_localite_arrivee.value=''">
  <option>Choisissez votre localité
  <option>Aeroport Ch de G
  ...
  <option>Tours
</select>
```

cette analyse est transcrite en :

```
<select>
 <name> choix_localite_depart
 <type> "check"
 <display> "Aeroport Ch de G"
 ...
 <display> "Tours"
 <texte> "Choisissez votre localité"
</select>
```

En analysant les formulaires, nous définissons les informations mises en jeu et ainsi cernons mieux le service proposé par le serveur que nous analysons; puis nous accédons aux pages envoyées par le serveur en réponse à un formulaire.

Un formulaire est composé d'une série de zones de saisie (d'items à sélectionner dans des listes déroulantes ou de séries de boutons radio), chaque zone ou item étant représenté par une variable dont le nom apparaît explicitement dans le code HTML/XML. Renseigner un formulaire correspond à l'instanciation de chacune de ces variables dont nous devons interpréter le sens (par l'analyse de son nom et des informations textuelles qui l'entourent), par une valeur qui peut être prise parmi celles affichées par défaut, en respectant des contraintes associées à l'ontologie globale, comme par exemples :la localité d'arrivée doit être différente de la localité de départ; la date de départ doit être antérieure à la date d'arrivée.

2.2 L'analyseur d'informations textuelles

Cet analyseur est appelé pour chaque page Web et appliqué sur :

- les titres et entêtes des pages Web,
- les textes de présentation des pages,
- **le nom des variables d'entrées des formulaires**,
- **les brèves explications accompagnant ces variables**.

L'analyseur de texte que nous présentons a été conçu à l'origine pour travailler rapidement sur des phrases grammaticalement simples (cad ne comportant pas trop de propositions imbriquées), pour enrichir automatiquement un dictionnaire bilingue (français-anglais) en comparant les mêmes textes traduits en français et en anglais, cet enrichissement prenant en compte le contexte des traductions effectuées. Comme cela sera expliqué plus avant, cet analyseur d'informations textuelles, surtout dans sa phase d'analyse syntaxique, n'a pas la prétention d'être au niveau d'outils beaucoup plus complets [1], [25]. Mais son atout majeur est sa rapidité.

Nous l'avons adapté à l'analyse de brèves informations textuelles qui nous intéressent dans les pages Web, à savoir, celles qui entourent les zones de saisie, en profitant du fait que les mots du dictionnaire bilingue qu'il utilise sont classés suivant un treillis d'hyperonymes. Un effet de bord intéressant est que cet outil peut également analyser des pages anglophones.

Cet outil enchaîne de manière distincte les trois premières phases classiques du traitement de la langue naturelle que sont les phases lexicale, syntaxique et sémantique. L'analyse syntaxique repose sur un système expert fonctionnant en chaînage avant. La quatrième phase pragmatique, non encore traitée, sera juste esquissée à l'aide de quelques exemples.

La phase lexicale

L'analyse lexicale permet de retrouver les mots du texte dans un dictionnaire. Elle exploite:

- un dictionnaire bilingue (français-anglais) de mots

  Exemple de traduction :
  ```
  descendre & verbe & 053
    bring_down/objet.0x0~action_de_déplacer ...
  ```
- un dictionnaire bilingue (français-anglais) de locutions

  Voici un exemple de traduction d'une locution :
  ```
  parler de [Qch] à [Qn]
  → tell [Qn] about [Qch] ~propos
  ```
- différentes tables de conjugaison (Bescherelle, table des verbes irréguliers anglais)

La phase syntaxique

L'analyse syntaxique permet d'analyser la syntaxe du texte. Elle exploite:

- une base de règles grammaticales dans chacune des deux langues :

  Voici un extrait des règles grammaticales décrivant les déterminants
  (Détermi, Loc–adj, pa_passé, prép–loc, et Groupe–n signifiant respectivement

déterminant, locution adjectivale, participe passé, préposition locative, et groupe nominal) :

Détermi:	adjectif	———	———
Détermi:	Loc–adj	———	———
Détermi:	pa_passé	———	———
Détermi:	pa_passé	prép–loc	nom–commun
Détermi:	pa_passé	prép–loc	nom–de–lieu
Détermi:	pa_passé	Préposition	Groupe–n
Détermi:	...		
Gr_dét:	Détermin	———	———
Gr_dét:	Détermin	Déterminant	———
Gr_dét:	Détermin	Conjonction	Déterminant
Gr_dét:	...		

– un moteur de système expert en chaînage avant

La phase sémantique

L'analyse sémantique exploite le dictionnaire bilingue où chaque mot est sémantiquement classifié sous un ou plusieurs concepts (les opérateurs =, \, /, ... sont détaillés plus loin), ces concepts formant un treillis de classes de substantifs.
Exemple pour le nom "départ" :

départ nom-m — :
leaving=action_de_départ\humain
departure=action_de_départ\moyen_de_transport
start=commencement\action_sportive
start=commencement\media

Exemple pour le verbe "descendre" :
descendre verbe 053:
bring_down/objet.0x0˜action_de_déplacer
take_down/objet.0x0˜action_de_déplacer
get_out.i\être/moyen_de_transport˜déplacement
fly_down\oiseau˜déplacement
shoot_down\humain/être˜attaque
go_down\soleil
go_down\humain/lieu˜déplacement
come_down\humain/lieu˜déplacement
dismount\humain/cheval
descend˜déplacement
flow_down\cours_eau
fall\température
fall\ombre

Cet analyseur n'a aucunement la prétention de rivaliser avec les analyseurs syntaxiques sophistiqués du français qui ont nécessité des années d'investissement humain, (par exemple ITS-2 pour ne citer que celui-ci [25]), mais il est adapté aux besoins de notre approche et comporte les limites suivantes :

– La taille des ressources est limitée, celles-ci étant intégralement chargées en mémoire centrale.
– Un moteur classique de système expert est utilisé en chaînage avant lors de la phase d'analyse syntaxique.
– Les informations textuelles analysées doivent être lapidaires car :
  – le nombre de mots par phrase est limité (une cinquantaine au plus),
  – les énumérations sont limitées à trois éléments,
  – le nombre de compléments imbriqués est restreint.

L'analyse syntaxique est réalisée sur des informations textuelles brèves, et n'opère qu'une recherche de "mots clés" pour les textes complexes. Cet analyseur a l'avantage d'être rapide ce qui permet de l'employer en temps réel lors de l'analyse des pages Web.

2.2.1 Utilisation de locutions spécifiques au domaine

Les locutions recouvrent des suites de mots auxquels sont attribuées une ou plusieurs natures grammaticales. Elles permettent l'analyse rapide de structures complexes, sans que le moteur ait à posséder des règles grammaticales trop nombreuses ou trop complexes. Ces locutions sont souvent propres au domaine exploré.

Par exemple :
à bas prix *Locution–complément*
→ ce service, **à bas prix**, ...

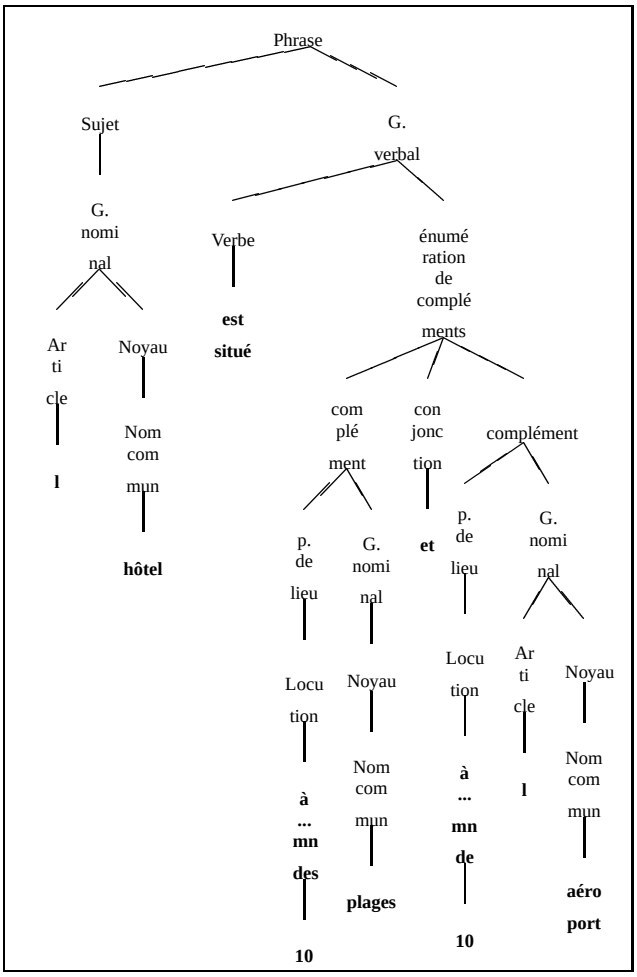
Un certains nombre de difficultés sont à gérer :

- Les locutions verbales doivent pouvoir être conjuguées :
*avoir froid *Locution–verbale* → Elle **a eu froid***
- Plusieurs natures grammaticales peuvent être attribuées :
ajuster [Qch] à [Qch]
Locution.–verbale+Locution–objet+L.–complément*
- Elles peuvent gérer d'un pronom comme objet :
faire [numer] enfants à [Qn]
Locution–verbale+Locution–objet*
→ *Il **lui a fait trois enfants***

Les locutions suivantes sont utilisées dans la phrase d'exemple :
"se situer" : localisation
"à *numéral* mn des" : préposition de lieu
"à *numéral* mn de" : préposition de lieu
nous permettant de générer pour la phrase

"**l'hôtel est situé à 10 mn des plages et à 10 mn de l'aéroport**"

l'arbre syntaxique suivant :



2.2.2 Utilisation d'un treillis de concepts pour hiérarchiser les substantifs du français

Dans l'outil d'origine d'enrichissement de dictionnaire bilingue, les méthodes d'apprentissage mises en jeu nécessitent l'emploi d'un treillis mettant en relation de subsomption les concepts sous-tendant les substantifs du français, pour essayer de gérer au mieux le fondamental et incontournable problème de la polysémie. La pauvreté des ouvrages de références (l'édification de dictionnaires à forte sémantique représente un travail colossal [21] et souvent décourageant), devrait nous insciter à étudier également l'emploi de techniques d'apprentissage automatique pour assister l'enrichissement de notre corpus sémantique, dans la lignée de quelques travaux d'analyse de dictionnaires monolingues [24] ou bilingues [12] [3]. En attendant, nous avons créé un treillis hiérarchisant 700 concepts (en nous inspirant partiellement de ceux du thésaurus Larousse [19]), soit par relation de subsomption (généralisation / spécialisation), soit exceptionnellement par la relation " est une partie de ". Il est important de remarquer que d'une part, cette hiérarchie de concepts est relativement naïve, (nous nous sommes focalisés uniquement sur les concepts permettant de discriminer les traductions principales des principaux noms, verbes et adjectifs), et que d'autre part nous avons édifié un treillis d'hyperonymes et non pas reproduit un thésaurus (où par exemple dans celui de Larousse, le substantif "émigré" sera associé, entre autres notions, à celle d'expatriation et non pas référencé comme une spécialisation d'un être humain).
Ce treillis est réemployé dans l'analyseur des informations textuelles pour cerner les différents contextes sémantiques des informations textuelles apparaissant dans les pages Web.

2.2.3 Liste des opérateurs sémantiques

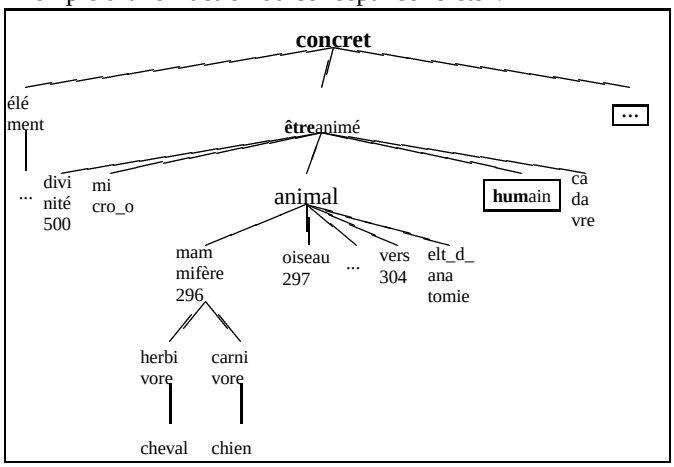
Pour les noms, les opérateurs sémantiques possibles sont les suivants :

Opérateur	Sémantique	Entrée	traduction annotée
=	est un	singe	donkey=mammifère
(	est un groupe de	grands-parents	grandparents(parents
<	est une partie de	accoudoir	armrest<meuble
>	a les propriétés de	glace	ice>froid
\	se rapporte à	accès	access\lieu
¯	implique	meurtre	murder¯mort

Pour les verbes, les opérateurs sémantiques possibles sont les suivants :

Opérateur	Sémantique	Entrée	traduction annotée
=	hypernonyme	crawler	crawl=nager
¯	réalise	choisir	choose¯action_sélection
\	nature du sujet	descendre	fly_down\oiseau
/	nature de l'objet	descendre	get_out.i\être /moyen_de_transport
*	contexte d'usage	élever	raise/enfant*américain
.i	verbe intransitif	couler	sink.i\bateau
.t	verbe avec COD	tromper	dupe.t\humain/humain
.0x0	pour la tr. anglaise phrasal-verb/adver.	découvrir	find_out.0x0 ¯action_de_détection
.00x	pour la tr. anglaise phrasal-verb/prépo.	regarder	look_at.00x ¯action_visuelle

Exemple d'une fraction du concept "concrets":



2.2.4 Application à l'analyse des noms des variables d'entrée

Chaque nom de variable est scindé en composants sémantiques (lexèmes), par rapport aux éventuels _, et aux éventuelles transitions entre majuscules et minuscules. Si un composant est référencé comme substantif dans le dictionnaire français/anglais, nous pouvons déterminer les concepts de plus en plus généraux auxquels il appartient, sous la forme : composant_sémantique=plus_petit_concept_généralisant <...<concept_universel

Exemple de séries de concepts généralisant les lexèmes de variables d'entrée :

choix_jour_aller	choix=action_sélection<action _recherche<tâche<... jour<événement_temporel<... aller=voyage<aventure<...
choix_localite_depart	choix=action_selection<... localité=agglomération<lieu_bâti... départ=action_de_départ<action _physique<action<...

Les correspondances entre les concepts du treillis d'hyperonymes et ceux de l'ontologie globale sont effectuées par la recherche de similitudes :

une même dénomination :	jour → jour
un lexème significatif :	événement_temporel → temporel
une synonymie :	voyage → trajet

Ce qui nous permet d'établir la table de correspondances suivante :

	concept du treillis	concept de l'OG
choix	=action_selection	
jour	=événement_temporel	temporel
aller	=voyage	trajet_aller
localité	=agglomération	ville
départ	=action_départ	s'applique à date, ...

Cette mise en correspondance, insuffisante, va être améliorée dans les futures versions de notre prototype.

2.2.5 Application de l'analyseur aux informations textuelles

L'analyse de

"Choisissez votre localité de départ :"

produit l'arbre syntaxique suivant :

```
{Phrase
 {Enumération-de-propositions
  {Proposition
   {Groupe-verbal
    {Verbe [verbe: choisir
            ~action_sélection<action_recherche<...],
     Enumération-d-objets
     {Objet
      {Groupe-nominal
       {Article [adjectif-possessif: votre],
        Noyau-avec-relatives
        {Noyau-nominal
         {Nom-commun [nom-féminin: ville
                      =agglomération<lieu_bâti<...],
          Enumération-de-compléments-de-noms
          {Complément-de-nom
          {[préposition-nominale: de],
           Noyau-nominal
           {Nom-commun [nom-masculin: départ
                        =action_de_départ<action<...])
}}}}}}}}}}
```

Ce qui produit l'analyse sémantique suivante :

action_sélection sur une **agglomération**
relative à une **action_de_départ**
qui devient, après association aux concepts correspondants dans l'ontologie globale :
action de sélection sur une **ville de départ**

2.2.6 Production d'une signature sémantique exploitée par la base de connaissances

Un fichier décrivant le nom des variables de saisie et leur signification sémantique probable est créé pour être exploité lors de la création de l'ontologie locale dans la base de connaissances. Il exploite également les instances associées aux variables (par exemple les choix proposés dans les listes déroulantes).
Dans l'exemple, le lexème *choix* est identifié comme signifiant simplement que le champ correspondant est à renseigner par l'utilisateur (ce qui est le cas par défaut).

choix_jour_aller	jour<temporel trajet_aller
choix_localite_depart	ville_départ<ville *texte: Choisissez votre localité*
choix_localite_arrivee	ville_arrivée<ville *texte: Choisissez votre localité*
...	

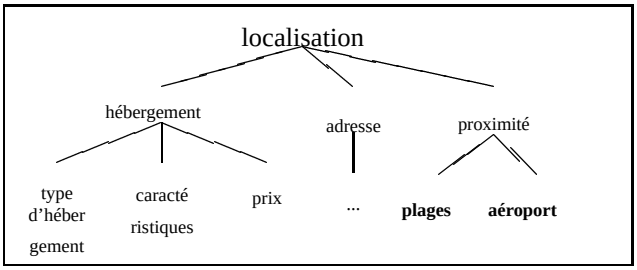
2.2.7 Fonctionnalités d'apprentissage prévues

Nous prévoyons d'intégrer des mécanismes d'apprentissage qui permettent de faire évoluer les ontologies locales et l'ontologie globale.

Par exemple, à partir des informations sémantiques associées aux locutions :

- le verbe "se situer" a comme objet une indication de localisation,
- la locution/préposition "à *numéral* mn de" introduit une indication de lieu.

L'ontologie locale concernant cet hôtel pourra être complétée pour faire apparaître sa proximité par rapport aux plages et à l'aéroport :



3 La modélisation des connaissances

Les connaissances extraites des différents sites sont stockées dans des agents informationnels. Dans l'état actuel de notre travail, l'ensemble des agents informationnels sont regroupés dans une seule base de connaissances.

Notre système est basé sur la représentation sémantique du domaine d'application et des sources d'information. Pour cela, nous utilisons des ontologies (dont la définition est donnée dans la section 3.1.1) qui permettent de décrire le domaine d'application (ontologies globales) et les sites correspondant à ce domaine (ontologies locales). Une ontologie est créée pour chacun des sites et décrit la structure des pages qui le composent, les informations accessibles sur le site, et les accès aux bases de données à travers des pages formulaires.

3.1 Les Ontologies

3.1.1 Définition

Parmi toutes les définitions du terme *ontologie* données dans la littérature, la plus citée est celle de Gruber [15] : *Une ontologie est une spécification explicite d'une conceptualisation*, c'est-à-dire une description d'une partie du "monde" en termes de concepts et de relations entre ces concepts.

L'ontologie a un rôle clé dans la représentation et l'utilisation des connaissances. Elle fournit une définition cohérente et non ambiguë du vocabulaire utilisé pour représenter la connaissance, mais elle ne se limite pas à une simple liste de termes; elle doit aussi fournir l'interprétation sémantique de ces termes.

Guarino [16] distingue quatre type d'ontologies, classées selon leur dépendance à une tâche ou un point de vue particulier. Les ontologies *"top-level"* décrivent des concepts très généraux, indépendants d'un problème ou d'un domaine particulier et sont donc réutilisables dans différents domaines. Les ontologies *du domaine* et les ontologies *de tâche* décrivent respectivement un domaine ou une tâche particulière. Les ontologies *applicatives* décrivent des concepts dépendants d'un domaine particulier pour une tâche particulière.

C'est à ce dernier type d'ontologie que nous nous intéressons. En ce sens, les ontologies sont dépendantes d'un domaine d'étude, d'une application spécifique de ce domaine, et des méthodes de résolution de problèmes que nous utiliserons. Leur rôle n'est pas d'être réutilisées pour d'autres applications, mais de faciliter un raisonnement particulier, dans notre cas l'extraction et l'exploitation d'informations sur le Web.
La construction d'une ontologie passe par les phases suivantes :

- délimiter le domaine d'intérêt et le niveau d'abstraction pour le décrire,
- définir le vocabulaire spécifique aux connaissances du domaine, c'est-à-dire un ensemble de termes, d'assertions et de contraintes sémantiques concernant ce domaine,
- modéliser les connaissances en termes de taxonomie de concepts et d'individus, de relations entre ceux-ci, et de contraintes et règles d'inférence.

3.1.2 Le formalisme de représentation

Tout comme le choix du type d'ontologie, le choix du formalisme de représentation des ontologies doit être guidé par la nature des connaissances à représenter et par le raisonnement à effectuer sur ces connaissances. Le formalisme utilisé doit permettre de représenter efficacement la connaissance, c'est-à-dire définir, structurer et classer les concepts relatifs au domaine, décrire les propriétés qui les caractérisent ainsi que les relations sémantiques qui existent entre eux. Il doit aussi permettre d'exploiter cette connaissance et de la faire évoluer. Notre choix a été guidé par le type d'informations à représenter, le type de raisonnement à effectuer et le type de requêtes disponibles.

Dans les différentes approches concernant l'intégration de connaissances, plusieurs formalismes à base de connaissances ont été utilisés; parmi les plus importants on peut citer la représentation des connaissances par objets [10], les graphes conceptuels [8] et les logiques de description [22]. Plusieurs études ont été menées pour comparer les fonctionnalités de ces différents formalismes (par exemple [5] et [9]).

Dans le cadre de notre application nous nous sommes orientés vers les logiques de description (DL), ou logiques terminologiques, qui sont des formalismes de représentation de connaissances, issues du système KL-ONE, développé par Brachman [6], lui-même issu des travaux relatifs à la logique du premier ordre, les réseaux sémantiques et les langages de frames. Il existe toute une gamme de DL, (e.g. CLASSIC [23], LOOM [18], BACK [17]), parmi lesquelles le point délicat était de trouver un compromis entre la puissance d'expression et la complexité des mécanismes d'inférence. Nous avons choisi CLASSIC qui offre une expressivité assez res-

treinte quoi que suffisante pour notre application, mais des algorithmes d'inférence complets et de complexité polynomiale.

Les composants de base des logiques de description sont les concepts, les rôles et les individus qui correspondent respectivement aux classes, relations et instances (ou objets) des langages à objets. Un concept est une description des propriétés communes à une collection d'individus; un individu est une entité particulière, instance du concept; un rôle est une relation binaire entre deux individus.

Les DL offrent un langage terminologique (T-Box) permettant de décrire les concepts et les rôles, et un langage assertionnel (A-Box) permettant de décrire les règles et contraintes qui s'appliquent aux concepts ainsi que les instances des concepts.

Les principaux mécanismes d'inférence offerts par les DL sont basés sur la relation de *subsomption*; La relation de subsomption est une relation d'ordre partiel qui permet d'organiser les concepts sous forme hiérarchique (on parle alors de taxinomie de concepts). Elle est également à la base des inférences telles que la *classification* qui permet de déterminer l'emplacement d'un concept ou d'une instance dans une hiérarchie, et la *détection d'incohérences*.

3.2 Construction de la base de connaissances

La base de connaissance qui permet de gérer les agents informationnels est décrite en CLASSIC. Nous représentons les méta concepts de cette base dans la figure 1 à l'aide du formalisme UML sous forme de trois paquetages :

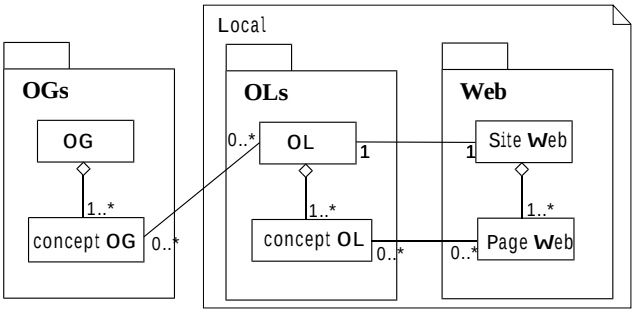


FIG. 1 – *Méta-concepts de la base de connaissances*

- le paquetage OGs contient les ontologies globales (par exemple l'ontologie globale des voyages). Chaque concept d'une ontologie globale sait dans quelles ontologies locales il peut être trouvé.
- le paquetage OLs contient les concepts des ontologies locales, définies à partir des ontologies globales et des pages web. Une ontologie locale correspond à la projection d'une ontologie globale sur un site Web. Chaque concept d'une ontologie locale sait dans quelles pages Web il peut être trouvé et de quelle ontologie globale il relève.
- le paquetage Web contient les informations concernant les pages web.

Toute la connaissance concernant un serveur est "encapsulée" localement dans un agent informationnel. Dans la version actuelle de notre prototype, les agents sont regroupés dans une même base de connaissances.

3.2.1 Les concepts principaux de la Base de Connaissances

Les concepts principaux de la base de connaissance (figure 2) sont CONCEPT-OG, CONCEPT-OL, SERVEUR-WEB et PAGE-WEB.

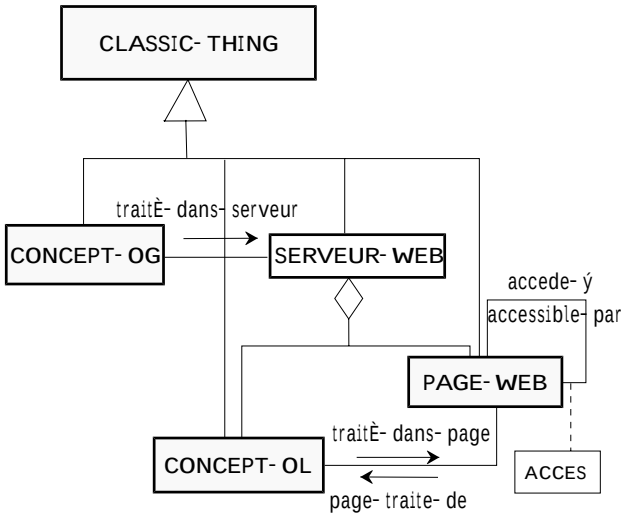


FIG. 2 – *Concepts principaux de la BC*

Le concept CONCEPT-OG subsume tous les concepts des ontologies globales utilisées pour chaque domaine. Tous les concepts décrits dans les ontologies globales le sont dans le cadre du Web. Nous admettrons donc, dans le contexte (spécialisé) de notre base de connaissances que PAYS, MOIS, VILLE, ... "sont-des" CONCEPT-OG.
Le rôle *traité-dans-serveur* entre les concepts CONCEPT-OG et SERVEUR-WEB permettra de connaître, à partir de l'ontologie globale, dans quel serveur, et donc dans quelle ontologie locale le concept peut être trouvé.
Le concept SERVEUR-WEB correspond aux serveurs du web se rapportant à l'ontologie globale, Chaque SERVEUR-WEB contient des PAGE-WEB. Le rôle *page-traite-de* (inv. *traité-dans-page*) entre les concepts CONCEPT-OL et PAGE-WEB permet de savoir dans quelle page du serveur a été rencontré le concept de l'ontologie locale.

3.2.2 L'ontologie Globale Voyage

L'ontologie globale utilisée dans notre application est celle des voyages, dont un extrait est représenté dans la figure 3. Elle est construite manuellement et décrit les connaissances de base nécessaires à la compréhension du domaine.

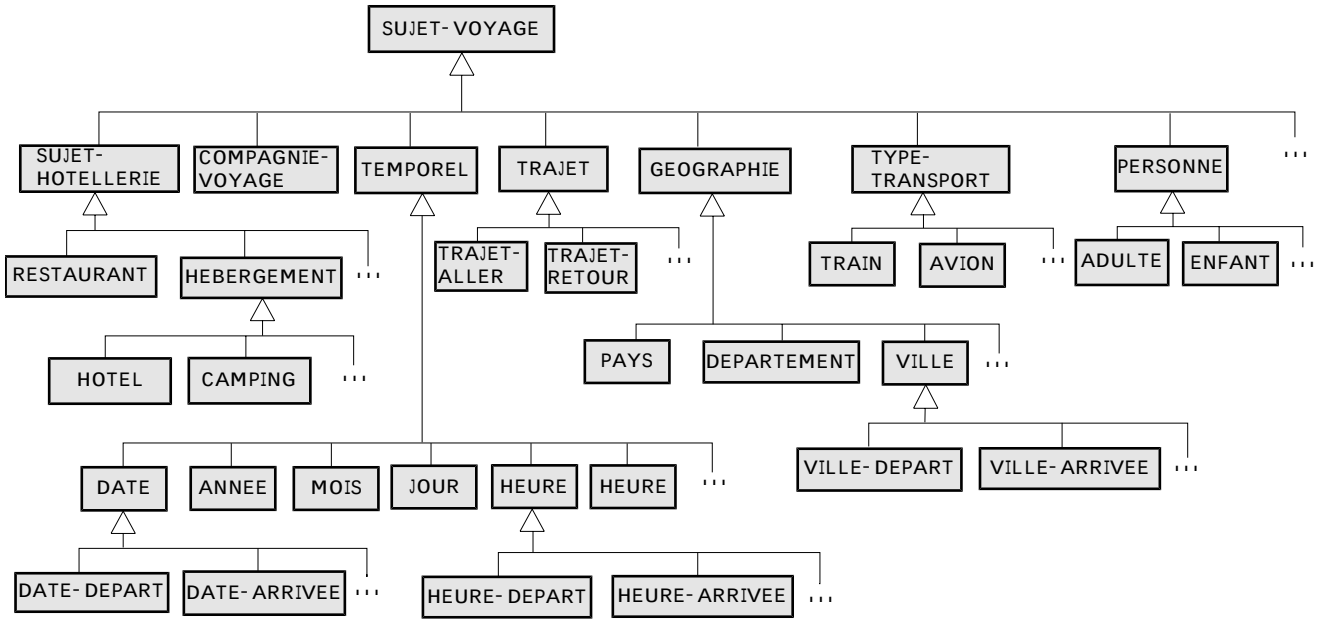


FIG. 3 – *Extrait de l'Ontologie Globale des voyages*

Nous rappelons que ce sont des connaissances spécifiques au domaine et à l'application. Par exemple, la modélisation de VILLE, DATE, PERSONNE comme une spécialisation de SUJET-VOYAGE, qui est lui même une spécialisation de CONCEPT-OG, est spécifique aux raisonnements utilisés par notre recherche d'informations sur le web. De la même manière, la spécialisation de VILLE en VILLE-DEPART et VILLE-ARRIVEE n'a un sens que dans le domaine des voyages.

3.3 Représentation des serveurs Web

3.3.1 Les informations représentées

Les types d'informations représentées localement sont les suivants :

- Des concepts et individus, sous-ensembles de l'ontologie globale correspondante, qui constituent l'ontologie locale.
- Des individus, instances de concepts de l'OG appris sur les pages (en général des instances rencontrées dans les parties "<Option>" des formulaires, par exemple des noms de villes, des dates ...).
- Des individus, instances de concepts spécifiques aux constructions locales :
  - Les instances de serveurs web.
  - Les instances de pages web, avec leurs relations et attributs : url; accès direct ou non; les contraintes d'accès; les enchaînements entre pages.
- les fonctions permettant de connaître les pages dans lesquelles les concepts ou instances de la requête sont traités.

3.3.2 Création des Ontologies Locales

La construction des ontologies locales se fait à partir des résultats fournis par l'analyseur d'informations textuelles, c'est-à-dire les concepts de l'OG reconnus dans les pages, les signatures sémantiques des pages concernant les variables d'entrée des formulaires et les informations textuelles associées. Les concepts d'une ontologie locale sont toujours un sous-ensemble des concepts de l'ontologie globale, seuls des individus peuvent être rajoutés.

Pour faciliter l'inférence, une *ontologie minimale* est calculée. Elle correspond aux concepts trouvés dans les pages, auxquels on ajoute les plus petits généralisants connus (PPGC) dans la liste de ces concepts, pris deux à deux, dont voici l'algorithme :

Entrée une liste de concepts
Sortie: concepts de l'ontologie locale correspondante.
Début
(liste ← **liste-de-concepts**(liste-de-termes)
OL ← liste
Tant-que liste ≠ ∅ **Faire**
Pour tout A ∈ liste **Faire**
Pour tout B ∈ {liste-B} **Faire**
liste ← liste ∪ **ppgc**(A,B)
OL ← OL ∪ liste
FinPour
liste ← liste - A
FinPour
FinTant-que
Fin

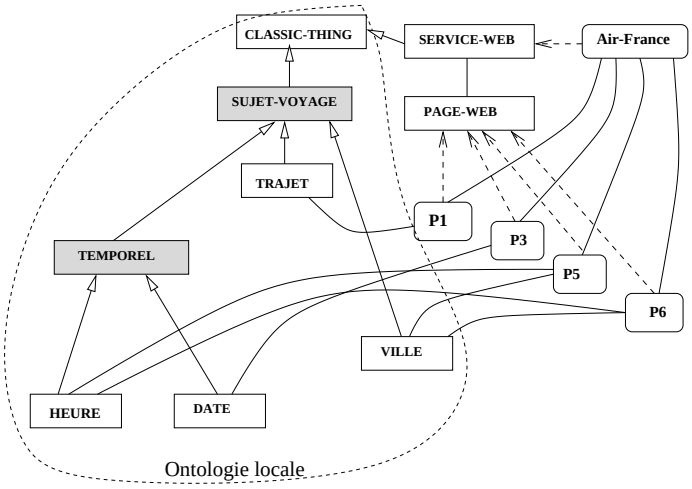


FIG. 4 – *Représentation locale d'un serveur*

Le calcul du PPGC

Pour deux concepts A et B, la fonction **ppgc**(A,B) renvoie une liste de concepts (C_0, ... C_n) vérifiant les conditions suivantes :
Pour tout i tel que $0 \leq i \leq n$,

- C_i subsume A,
- C_i subsume B,
- Il n'existe pas de concept connu subsumant de A et B qui soit subsumé par C_i.

Ce qui correspond à :

ppgc(A,B) =
(concepts les plus spécifiques
((A ∪ ancêtres (A)) ∩ (B ∪ ancêtres (B))

Il est à noter que cette fonction ne construit pas de nouveau concept, mais effectue seulement une recherche parmi les concepts connus.

Exemple de construction de la taxinomie d'une OL

La figure 4 présente un exemple de création d'une ontologie locale à partir des pages du serveur Air-France, dans lesquelles nous avons trouvé les termes suivants :
page1 : trajet, ...; page3 : date, ...; page 5 : ville, heure, ...; page 6 : ville, heure, ... Les concepts grisés sont ceux qui ont été rajoutés par le calcul du PPGC.

Modélisation des fonctions d'accès aux bases de données :

Il s'agit de modéliser la fonction d'accès une page qui n'est pas accessible directement par l'utilisateur mais uniquement à travers une page formulaire. Par exemple, l' accès à la page des horaires de la SNCF demande :

- la page formulaire précédente,
- obligatoirement : ville de départ, ville d'arrivée, date de départ, heure de départ,
- optionnellement : heure de départ, via ...

Pour créer l'ontologie locale correspondant à cette page, il faut pouvoir y accéder en remplissant le formulaire correspondant, ce qui inclut les étapes suivantes :

- créer l'OL correspondant à la page formulaire, et contenant toutes les instances données en option,
- créer une instance d'objet web, respectant les règles liées aux contraintes. Dans le cas des horaires :
  - ville de départ différente de ville d'arrivée,
  - date de départ inférieure ou égale à date d'arrivée,
  - heure de départ inférieure à heure d'arrivée.

  Les instances son choisies de manière aléatoire dans l'OL si elle en contient, dans l'OG sinon.
- soumettre cet individu web au navigateur pour accéder à la page "horaires".

Nous obtenons ainsi les mécanismes permettant à l'utilisateur de faire des requêtes accédant aux BDs.
Par exemple, pour une page de consultations d'horaires, la fonction trouvée dans la page est

(Date départ, heure départ, ville départ, ville arrivée) → une vue de la BD des horaires.

La requête :"Horaires des trains de Montpellier à Paris le 21 janvier" renvoie la page suivante :

Montpellier	14 : 06	Paris Gare de Lyon	18 : 28	TGV
Montpellier	15 : 43	Paris Gare de Lyon	20 : 10	TGV
...				

Exemple d'individus

La figure 5 est un diagramme d'instances, représentant un extrait des individus de l'ontologie locale du serveur SNCF, destiné à montrer le fonctionnement des accès aux informations pour un serveur donné.

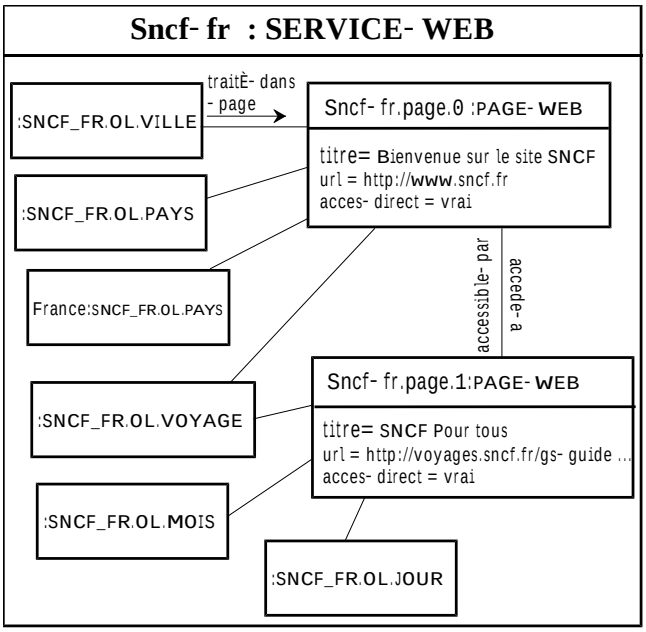


FIG. 5 – *Exemple d'instanciation de pages*

4 Conclusion et état d'avancement

Nous avons retenu pour le projet Chimère une approche incrémentale consistant à développer un prototype "minimal" qui permet d'intégrer des informations issues de sites web et de montrer la faisabilité de l'approche retenue. Ce prototype sera progressivement développé et complété par l'ajout de fonctionnalités. Actuellement, ce prototype importe des pages HTML/XML, les analyse syntaxiquement pour en extraire les structures relatives aux formulaires. A partir de ces structures, il analyse sémantiquement les noms des variables, s'ils sont explicites, et les brèves informations textuelles les accompagnant. Cette analyse nous permet de renseigner automatiquement les formulaires, en identifiant les saisies à opérer à l'aide de l'ontologie globale.
Un site type exploitable est celui de la SNCF; tous ne le seront pas (à cause de noms de variables abscons, d'informations affichées sous forme d'images ou par l'intermédiaire d'applets java).
Un point délicat est la correspondance entre l'analyse sémantique résultant du traitement de la langue naturelle effectué, et les concepts de l'ontologie globale, plus spécifiques que ceux exprimés dans notre classification des substantifs. Nous insistons néanmoins sur le fait que cette analyse est indispensable, par exemple la variable *choix_nombre_type_voyageur_6* apparaissant dans le formulaire de la SNCF, ne pouvant être correctement interprétée que par l'analyse du texte associé : *Adultes (+ de 17 ans)*.
Les prochaines étapes vont concerner la prise en compte de l'aspect dynamique des ontologies (requêtes d'accès aux bases de données sur les serveurs), ainsi que les fonctionnalités d'apprentissage et de traitement des requêtes.

Remerciements : Nous tenons à remercier Thierry Bouron, Gilles Deflandres et Philippe Porretta du CNET pour leur collaboration très fructueuse sur ce projet.

Références

[1] A. Abeillé. *Une grammaire lexicalisée d'Arbres Adjoints pour le français : application à l'analyse automatique*. PhD thesis, Université Paris 7, 1991.

[2] S. Abiteboul, P. Buneman et D. Suciu. *Data on the Web*. Morgan Kaufmann, 1999.

[3] A.Michiels. An experiment in translation selection and word sense discrimination. english department. Technical report, University of Liège, Liège, Belgium, 1996.

[4] Y. Arens, C.A. Knoblock et C. Hsu. *Query Processing in the SIMS Information Mediator*. Austin Tate, AAI Press, Menlo Park, CA, 1996.

[5] B. Biébow et G. Chaty. A comparison between conceptuals graphs and kl-one. In *Proceedings of 1st ICCS*, LNAI 669, Springer-Verlag, Québec, CA, 1993.

[6] R.J. Brachman et J.G. Schmolze. An overview of the KL-ONE Knowledge Representation System. *Cognitive Science*, 9:171–216, 1985.

[7] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman et J. Widom. The Tsimmis project: Integration of heterogeneous information sources. In *Proceedings of IPSJ*, Tokyo, Japan, Octobre 1994.

[8] M. Chein et M.-L. Mugnier. Conceptual graphs : Fundamental notions. *Revue d'Intelligence Atificielle*, 6(4):365–406, 1992.

[9] P. Coupey et C. Faron. Toward correspondances between conceptual graphs and description logics. In *Actes 6th ICCS*, Montpellier, France, 1998.

[10] J. Euzenat. Représentation de connaissance par objets. In Ducournau, Euzenat, Masini et Napoli, éditeurs, *Langages et modèles à objets : état et perspectives de la recherche*. INRIA, Rocquencourt, 1998.

[11] D. Fensel, S. Decker, M. Erdmann et R. Studer. How to Enable Intelligent Access to the WWW. In *Proceedings of KAW'98*, Alberta, Canada, Avril 1998.

[12] Fontenelle. *Turning a Bilingual Dictionary into a Lexical-semantic Database*. PhD thesis, University of Liège, 1995.

[13] F. Goasdoué et C. ReynaudD. Modeling Information Sources for Information Integration. In Dieter Fensel et Rudi Studer, éditeurs, *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management, EKAW'99*. Lecture Notes in AI 1621, Springer Verlag, 1999.

[14] A. Goñi, E. Mena et A. Illarramendi. Querying Heterogeneous and Distributed Data Reposititories using Ontologies. In *7th European-Japanese Conference on Information Modelling and Knowledge Bases (IMKB'97)*, Toulouse, Mai 1997.

[15] T. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5:19–220, 1993.

[16] N. Guarino. Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In M. T. Pazienza, éditeur, *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, pages 139–170. Springer Verlag, 1984.

[17] T. Hoppe, C. Kindermann, J. Quantz, A. Schmiedel et M. Fischer. BACK V5 Tutorial & Manual. KIT-Report 100. Technical report, Department of Computer Science, Technische Universität Berlin, Berlin, Germany, Mars 1993.

[18] ISX Corporation. LOOM *User's Guide for Loom version 1.4*, Août 1991.

[19] Thésaurus Larousse. Larousse, 1997.

[20] A.Y. Levy, A. Rajaraman et J.J. Ordille. Querying Heterogeneous Information Sources Using Sourc Descriptions. In *Proceedings of the 22nd International conference on Very Large Databases, VLDB96*, Bombay, India, Septembre 1996.

[21] Igor A. Mel'cuk, André Clas et Alain Polguère. *Introduction à la lexicologie explicative et combinatoire*. Editions Duculot AUPELF UREF.

[22] A. Napoli. *Une introduction aux logiques de description*. Rapport de Recherche n^o 3314, INRIA Lorraine, 1997.

[23] L.A. Resnick, A. Borgida, R.J. Brachman, D.L. McGuinness et P.F. Patel-Schneider. CLASSIC *Description and Reference Manual for the* COMMON LISP *Implementation: Version 2.3*. AI Principles Research Department, AT&T Bell Laboratories, 1995.

[24] S. Richardson, L. Vanderwende et W. Dolan. Combining dictionary-based and example-based methods for natural language analysis. In *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation*, Kyoto, Japan, 1993.

[25] Eric Wehrli. *Description du système ITS-2*. Université de Genève, Septembre 1993.