La base de données

Introduction

Ce chapitre est destiné à présenter l'implémentation des différents concepts et relations du domaine des noms propres sous la forme d'une base de données relationnelle, que nous avons appelée Prolexbase.

Pour créer notre base de données des noms propres, nous avons utilisé la méthode Merise. Dans la première partie, nous allons présenter quelques notions de base sur cette méthode. Ensuite, nous décrirons le modèle conceptuel de données et le modèle logique de données que nous avons mis en place pour les noms propres.

5.1 La méthode Merise

Développée en France en 1978, la méthode Merise (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise) [Matheron, 1998] propose une démarche pour analyser et concevoir un système d'information. Dans cette partie, nous allons présenter brièvement deux étapes de cette méthode : le modèle conceptuel de données et le modèle logique de données.

5.1.1 Le modèle conceptuel de données

Le modèle conceptuel de données (MCD) permet de décrire les objets de la réalité et les dépendances ou associations entre ces objets. Un MCD aboutit à la création d'un schéma d'entité/association (E/A).

Les entités

Une entité est définie comme un objet concret ou abstrait du monde réel. Dans le modèle E/A, on représente une entité sous la forme d'un rectangle (figure 5.1) dans lequel on inscrit le nom de l'entité.

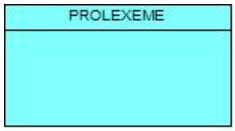

Fig. 5.1 – Représentation d'une entité.

Chaque entité peut posséder un ou plusieurs attributs, dont on devra préciser le type (Date, Entier, Booléen, Texte, etc.). Pour pouvoir identifier chaque occurrence d'une entité de manière unique, il faudra obligatoirement désigner parmi ses différents attributs un attribut ou un ensemble d'attributs qui jouera le rôle d'identifiant ou de clé primaire. Il arrive souvent que l'on rajoute un attribut fictif (un numéro) qui servira de clé primaire. Nous avons associé à chaque identifiant le type ID dans nos schémas E/A (figure 5.2).

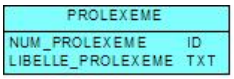

FIG. 5.2 – Représentation des attributs.

Les associations

Les associations sont des liens qui unissent les entités du modèle. Elles apparaissent dans un schéma E/A sous la forme d'un ovale (figure 5.3). On associe à chaque entité d'une association une cardinalité qui précise si une entité peut participer dans l'association zéro, une ou plusieurs fois.

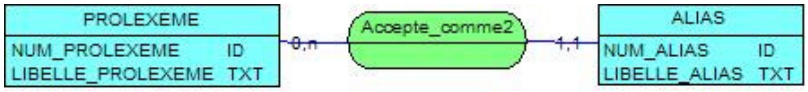

FIG. 5.3 – Représentation d'une association.

Dans la figure 5.3, les entités *PROLEXEME* et *ALIAS* sont reliées par l'association *Accepte_ comme2*. La cardinalité (0,n) indique qu'un prolexème accepte au minimum zéro alias et au maximum plusieurs alias. La cardinalité (1,1) précise qu'un alias correspond à un seul et unique prolexème.

5.1.2 Le modèle relationnel de données

Le modèle relationnel de données (MLD) correspond à une traduction des entités et des associations du MCD sous la forme de relations. Les principales règles de passage d'un MCD vers MLD sont les suivantes :

- Règle 1 : une entité du MCD se transforme en relation. Ses propriétés deviennent des attributs. La clé primaire de la relation sera représentée par l'identifiant.
- Règle 2 : soit R une association de type un-à-plusieurs reliant deux entités E1 et E2 (une occurrence de E1 peut être en relation avec au maximum une occurrence de E2 et une occurrence de E2 peut être en relation avec plusieurs occurrences de E1). R ne devient pas une relation. L'identifiant de E2 et les éventuelles propriétés de R sont rajoutés dans la relation E1.
- Règle 3 : soit R une association de type plusieurs-à-plusieurs reliant deux entités E1 et E2 (plusieurs occurrences de E1 peuvent être en relation avec plusieurs occurrences de E2 et plusieurs occurrences de E2 peuvent être en relation avec plusieurs occurrences de E1). R devient une relation et ses éventuelles propriétés seront des attributs. Les identifiants de E1 et E2 deviennent les clés primaires de R.

En appliquant ces règles sur la figure 5.3, nous obtenons le modèle relationnel suivant :

PROLEXEME (NUM_PROLEXEME, LIBELLE_PROLEXEME)
ALIAS (NUM_ALIAS, LIBELLE_ALIAS, NUM_PROLEXEME)

qu'il est possible de représenter sous la forme d'un schéma relationnel (figure 5.4).

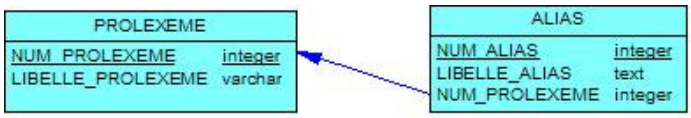

FIG. 5.4 – Schéma relationnel.

5.2 Modèle conceptuel de données

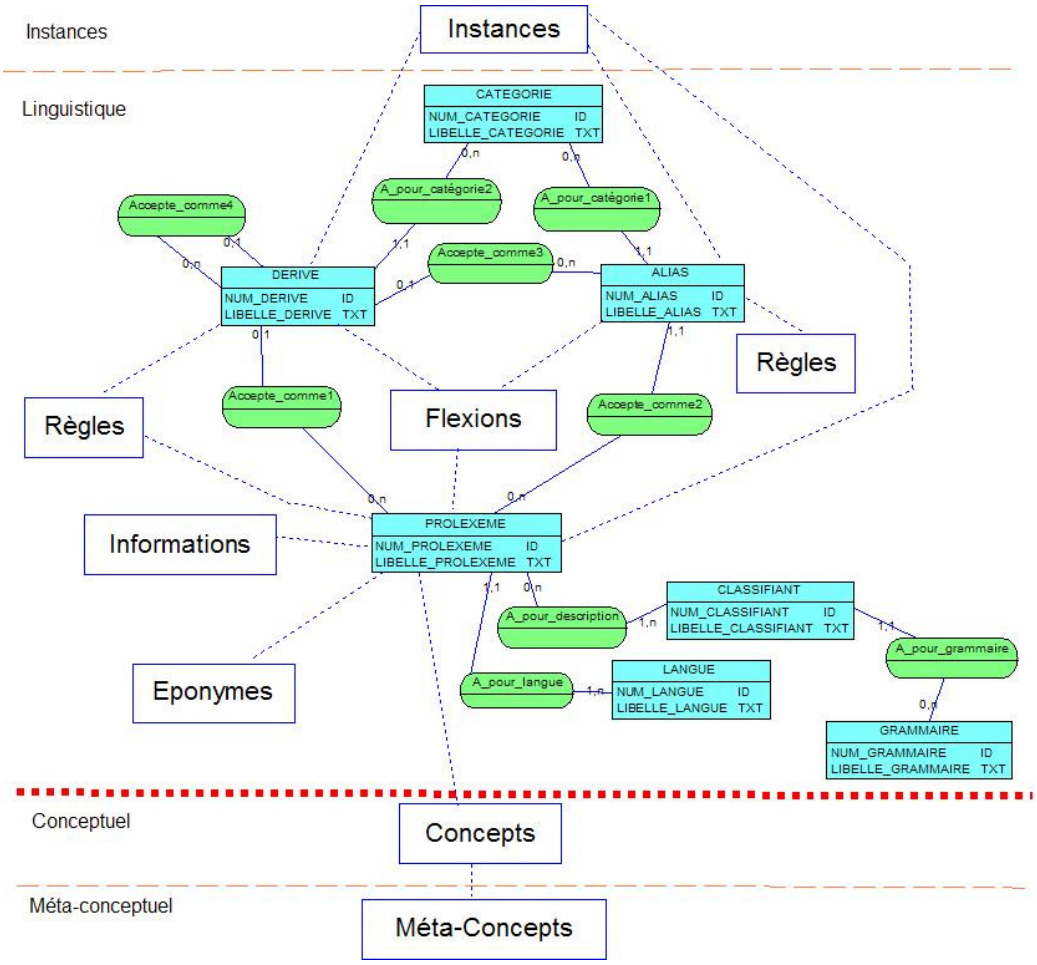

FIG. 5.5 – Le modèle conceptuel de données.

Nous avons établi notre MCD des noms propres à partir des différents concepts et relations définis dans le chapitre 2 et le chapitre 3. La figure 5.5 présente une version simplifiée de notre MCD (voir figure A.3 de l'annexe A page 151 pour un MCD complet).

Notre MCD peut être regroupé en quatre niveaux (méta-conceptuel, conceptuel, linguistique et instances) et comprend au total 28 entités et 41 associations.

Nous avons créé une entité pour chaque concept du domaine des noms propres (prolexème, alias, dérivé, etc.). L'entité *DERIVE* permet de stocker les dérivés de prolexème, d'alias ou d'autres dérivés (dans le cas du serbe). Nous avons associé à chaque alias, à travers la relation *A_pour_catégorie1*, une catégorie qui précise s'il s'agit d'une variante de caractères, d'une abréviation, d'acronymes ou sigles, d'une transcription, d'un synonyme diastratique ou d'un synonyme diatopique. Nous avons aussi associé à chaque dérivé, à travers la relation *A_pour_catégorie2*, une catégorie qui indique s'il s'agit d'un nom relationnel, d'un préfixe, d'un adjectif relationnel ou possessif. Les expansions classifiantes sont stockées dans l'entité *CLASSIFIANT* et chaque classifiant sera en relation avec une description (entité *GRAMMAIRE*) sous forme de grammaire locale, lien vers EuroWordNet ou Framenet (voir section 3.3.4). La relation *A_pour_langue* permet d'associer à chaque prolexème une langue.

5.2.1 Le niveau conceptuel

La partie conceptuelle (figure 5.6) est formée de quatre entités et cinq relations. L'entité *PIVOT* permet de stocker les noms propres conceptuels. La relation *Concept* associe à chaque nom propre conceptuel un ou plusieurs prolexèmes. On retrouve dans ce niveau la relation de méronymie, de synonymie et d'accessibilité.

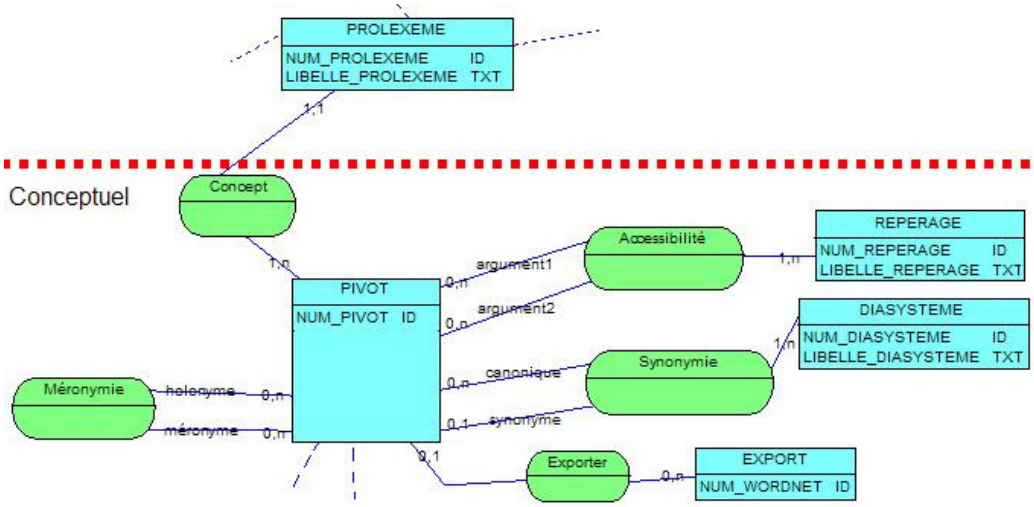

FIG. 5.6 – Le niveau conceptuel.

Un nom propre conceptuel sera en relation de synonymie avec un autre nom propre conceptuel suivant un diasystème (entité *DIASYSTEME*), qui peut être diachronique, diastratique ou diaphasique. La figure 5.7 présente la liste de repérages pour la relation d'accessibilité. Pour une relation de synonymie, nous avons imposé que chaque pivot peut être la forme canonique de plusieurs autres pivots et que chaque pivot peut être le synonyme d'une seule forme canonique.

L'entité *EXPORT* sert à relier les noms propres conceptuels de notre dictionnaire vers d'autres bases de données lexicales ou vers des encyclopédies. Des liens vers l'encyclopédie Wikipédia[1] et vers EuroWordNet ont été envisagés.

[1]Wikipédia est une encyclopédie gratuite accessible à l'adresse suivante : `http://www.wikipedia.org/`.

Repérage	**Exemple**
Capitale	Paris est la capitale de la France
Créateur	Auguste Rodin est le sculpteur du Penseur
Dirigeant non politique	Ray Norda est le patron de Novell
Dirigeant politique	Jacques Chirac est le président de la République française
Élève	Platon est l'élève de Socrate
Fondateur	Dardanos est le fondateur mythique de Troie
Héritier	Charles, prince de Galles, héritier du Royaume-Uni
Locataire	Jacques Chirac est le locataire de l'Elysée
Parent	Aaron est le frère de Moïse
Siège	Le Bureau Veritas a son siège à Paris
...	

FIG. 5.7 – Les repérages.

Le lien vers l'encyclopédie Wikipédia n'est pas conservé dans Prolexbase. Ce lien est généré dynamiquement sur le site de consultation en concaténant le code iso de la langue de consultation (fr, en, etc.), une url (wikipedia.org/wiki/Special:Search/) et le prolexème sélectionné par le visiteur. Pour le nom propre *France*, on produit ainsi le lien suivant :

http://fr.wikipedia.org/wiki/Special:Search/France

La génération automatique des liens vers l'encyclopédie Wikipédia présente un inconvénient. Tous les liens générés automatiquement n'ont pas été testés, l'interface de consultation peut par conséquent produire des liens qui n'existent pas, car certains articles ne sont pas présent dans cette encyclopédie, ou des liens vers un mauvais article. Pour éviter les liens incorrects, il faudrait vérifier manuellement chaque lien et les conserver dans la base de données. Il s'agit d'une tâche extrêmement longue. Par manque de temps, nous avons décidé de générer automatiquement les liens vers l'encyclopédie Wikipédia. Cette encyclopédie est en cours de développement : un lien incorrect aujourd'hui pourrait devenir correct le jour suivant.

Le lien vers la base lexicale EuroWordNet est conservé dans notre base de données grâce à l'entité *EXPORT*. Si le nom propre conceptuel existe dans EuroWordNet, son numéro ILI (Inter-Lingual-Index) apparaîtra dans l'entité *EXPORT*. Par exemple, on associera au nom propre *Paris* le numéro d'ILI *0558236n* (figure 5.8).

entity
location
region
area, country
center, middle, heart
seat
capital
national capital
Paris, City of Light, French capital, capital of France

FIG. 5.8 – Le nom propre *Paris* dans EuroWordNet.

5.2.2 Le niveau méta-conceptuel

La partie méta-conceptuelle (figure 5.9) comprend deux entités et quatre associations. L'entité *EXISTENCE* contient trois occurrences : historique, fictif et religieux. Nous avons regroupé les types et les supertypes dans une seule entité (*TYPE*), afin de pouvoir associer à un nom propre conceptuel un supertype, si l'on n'a pas d'information sur son type. Cela nous permet d'insérer dans notre dictionnaire des noms propres qui ont été trouvés par des systèmes de reconnaissance automatique de noms propres.

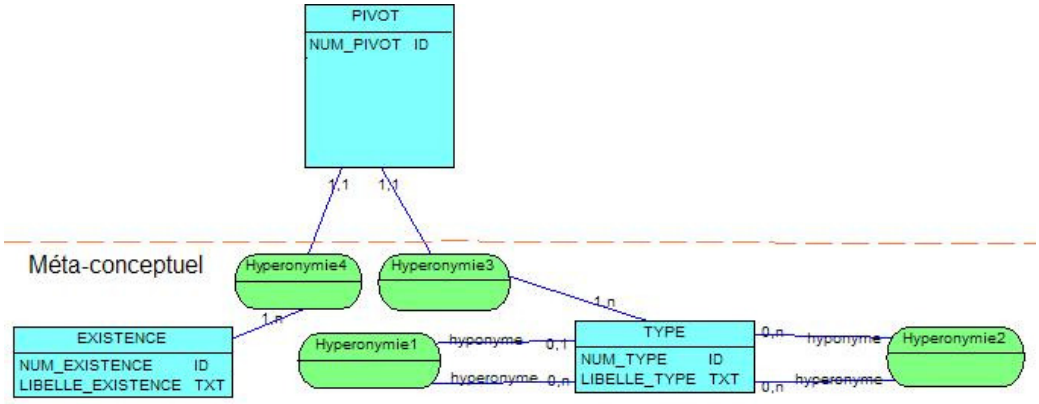

FIG. 5.9 – Le niveau méta-conceptuel.

5.2.3 L'éponymie

L'éponymie (figure 5.10) regroupe les entités *IDIOME*, *TERMINOLOGIE* et *ANTONOMASE*.

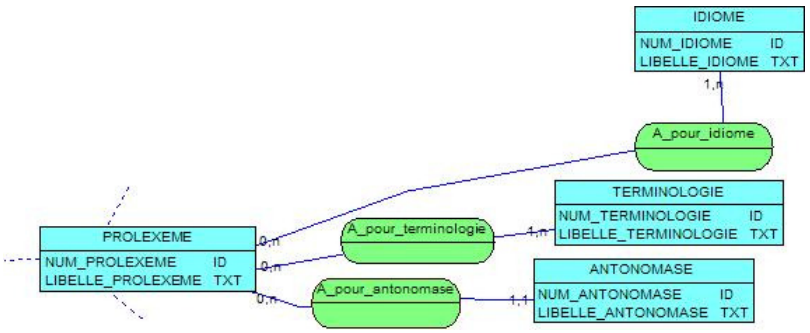

FIG. 5.10 – L'éponymie.

5.2.4 Les règles

L'entité *ALIASISATION* (figure 5.11) permet de stocker les règles de création d'alias à partir d'un prolexème. L'entité *DERIVATION* permet de stocker les règles de création de dérivés à partir d'un prolexème, d'un alias ou d'un dérivé.

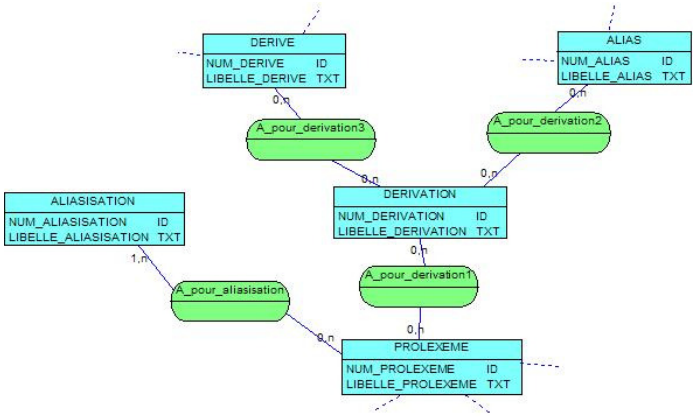

FIG. 5.11 – Les règles.

5.2.5 Les autres informations

Les informations supplémentaires (figure 5.12) sont formées de cinq entités et de cinq associations.

L'association *A_pour_statistique* permet d'associer à chaque prolexème des informations relatives à ses fréquences d'apparition (attribut *POIDS*) au sein d'un corpus donné (attribut *LIBELLE_STATISTIQUE*). Il peut s'agir, par exemple, d'étudier les fréquences d'apparition de noms propres sur quelques années d'un corpus journalistique. Certains noms propres apparaissant durant une année donnée pourront ne plus réapparaître quelques années plus tard. Cette étude statistique peut prendre en compte les différentes formes d'un même prolexème (ses alias, ses dérivés et leurs formes fléchies).

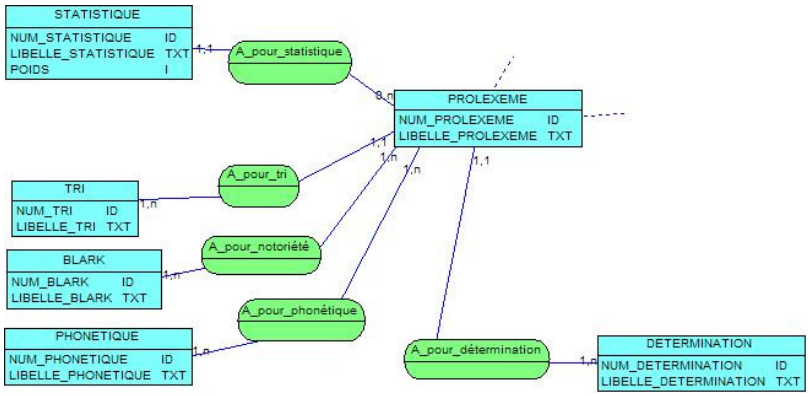

FIG. 5.12 – Les informations.

S'il est normal pour le simple mortel que nous sommes de ne pas posséder d'entrée dans les dictionnaires de noms propres, on peut parfois se demander pourquoi certains chanteurs ou chanteuses de variété française (*Johnny Hallyday*, etc.), actrices chinoises (*Michelle Yeoh*, etc.) ou autres célébrités ne figurent pas dans ces dictionnaires. On pourra aussi s'étonner que des villes telles que, par exemple, *Sainte-Enimie*, qui est le chef-lieu de canton de la *Lozère* comprenant à peine moins de 600 habitants et dont la majorité des Français ignore même l'existence, puisse apparaître dans le dictionnaire, alors que des villes de Russie, de Chine, et d'autres pays ayant une population nettement supérieure n'y figurent pas. Comme

le fait remarquer [Leroy, 1994], le choix d'inclure ou non un nom propre dans un dictionnaire repose essentiellement sur un critère de notoriété :

> *[Les dictionnaires] recueillent en effet uniquement les noms propres notoires : loin de recenser tous les noms propres, comme les dictionnaires de langue visent à recenser l'ensemble du lexique général, ils ne contiennent que les noms propres dont le référent a une certaine notoriété.*

Cette notoriété dépend beaucoup de différents facteurs extralinguistiques, tels que la culture, la région, la période considérée, etc. La figure 5.13 présente les indicateurs que nous avons utilisés. Ces indicateurs pourront servir à définir des dictionnaires de base dans une langue donnée, selon l'idée de [Cucchiarini et al., 2000] (BLARK pour *Basic LAnguage Ressources Kit*). Notons qu'un même nom propre pourra recevoir plusieurs indicateurs Blark. Par exemple, le prolexème français *Paris* correspond aux lignes 1, 2, 4, 5, 7 et 8 de la figure 5.13.

Sources des données françaises		Indicateur BLARK
INSEE (consultation Internet de juin 2005)	Les cinquante-sept villes françaises comportant plus de dix mille habitants	NATIONAL
Base de données Géopolis (consultation Internet de juin 2005)	Les quarante villes de l'Union Européenne comportant plus d'un million d'habitants	EUROPEEN
	Les soixante-quatorze principales villes du monde comportant plus de trois millions neuf cent mille habitants	INTERNATIONAL
Prolex (travaux antérieurs)	Toutes les villes françaises	DETAIL
	Les départements et les régions françaises	NATIONAL
	Hydronymes mondiaux	DETAIL
	Les pays de l'ONU, leurs capitales	INTERNATIONAL
	Tous les pays, régions et capitales mondiales	DETAIL
Dictionnaire Larousse du Collège, édition 2004	Extraction des noms propres en entrée d'un article	NATIONAL

FIG. 5.13 – Les indicateurs actuellement utilisés pour le BLARK.

La relation *A_pour_détermination* permet de spécifier si le prolexème comporte ou non un déterminant. On trouve la détermination dans de nombreuses langues, comme le français, l'anglais, l'allemand, etc. On constate que dans certaines langues, comme le serbe, ce phénomène n'existe pas. Un nom propre se construisant dans une langue avec une détermination peut apparaître dans une autre langue sans détermination. C'est le cas du nom propre *Spanien* en allemand qui devra être traduit en français par *l'Espagne*. Ce phénomène devra être pris en compte dans le cadre de la traduction automatique.

La phonétique permet de proposer une transcription d'un nom propre lorsque celui-ci ne possède pas de traduction et qu'il appartient à un autre système d'écriture. C'est le cas pour le prénom Paul qui se transcrira par :

- *Pol* en serbe alphabet latin
- *Пол* en serbe alphabet cyrillique
- *Поль* en russe

Il existe des prolexèmes ayant plusieurs prononciations différentes. Par exemple, les parisiens prononcent [*mɛts*] pour la ville de Metz alors que les Lorrains prononcent [*mɛs*].

La relation *A_pour_tri* donne des informations sur la façon de trier les noms propres polylexicaux. Nous avons attribué à chaque prolexème de notre dictionnaire un numéro qui correspond au début du cycle de tri de celui-ci. Par exemple, on associera au nom propre polylexical *mer des Philippines*, classé dans les dictionnaires sous la lettre P, le numéro de tri 3. Il devra être trié comme le mot polylexical *Philippines mer des*. C'est une simplification du modèle que nous avons présenté dans [Tran et al., 2005].

5.2.6 Les flexions

Un code flexionnel est attribué à chaque prolexème, alias et dérivé (figure 5.14). Pour la flexion des noms monolexicaux français, nous avons décidé d'utiliser les codes flexionnels du DELA [Courtois, 1992] [Paumier, 2006]. Une liste des codes flexionnels des noms monolexicaux (entité *FLEXION*), utilisée dans Prolexbase, est donnée en annexe B.

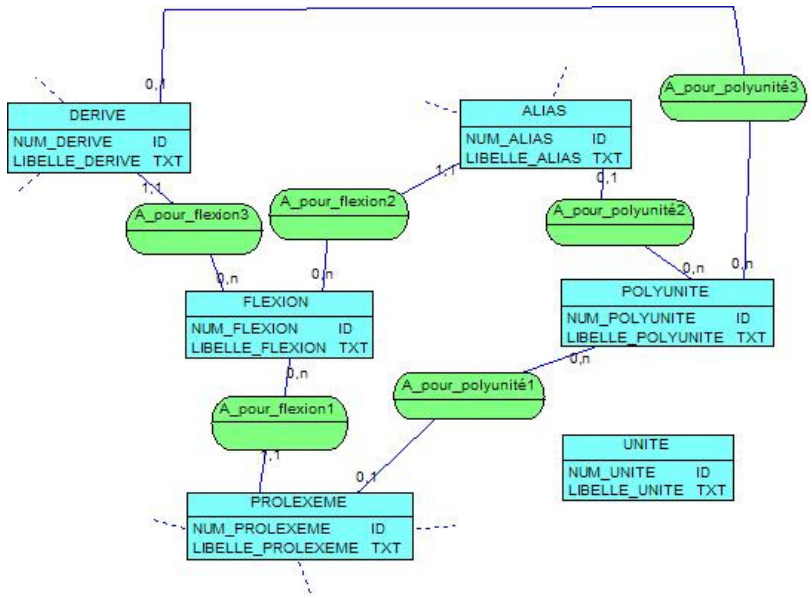

FIG. 5.14 – La flexion.

Nous avons prévu d'utiliser les codes flexionnels inspirés de [Savary, 2006] pour les noms propres polylexicaux. Chaque nom propre polylexical sera en relation avec une polyunité (entité POLYUNITE) qui correspond à une concaténation d'unités *UNITE*. Par exemple, nous associerons au nom relationnel *Antiguais-et-Barbudien*, dont le prolexème est *Antigua-et-Barbuda* :

– un code flexionnel pour les deux unités : $Antiguais.N61 : ms$, $Barbudien.N41 : ms$.
– un graphe de flexion (figure 5.15).

5.2.7 Les instances

L'entité *INSTANCE* (figure 5.16) regroupe l'ensemble des formes fléchies des prolexèmes, des alias et des dérivés. Selon la langue, à travers l'association *A_pour_morphologie*, nous indiquons pour chaque instance des informations morphologiques :

– CLASSE : nom, adjectif, etc.

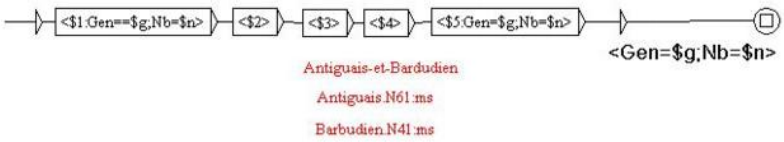

FIG. 5.15 – Le graphe de flexion d'*Antillais-et-Barbudien*.

– GENRE : masculin, féminin, etc.
– CAS : nominatif, accusatif, etc.
– NOMBRE : singulier, pluriel, etc.

Cette entité est utilisée pour les recherches de noms propres à partir de leurs formes fléchies à travers une interface web de consultation (voir section 7.2). Le visiteur rentre un nom propre fléchi et l'interface lui affiche le prolexème correspondant avec ses informations.

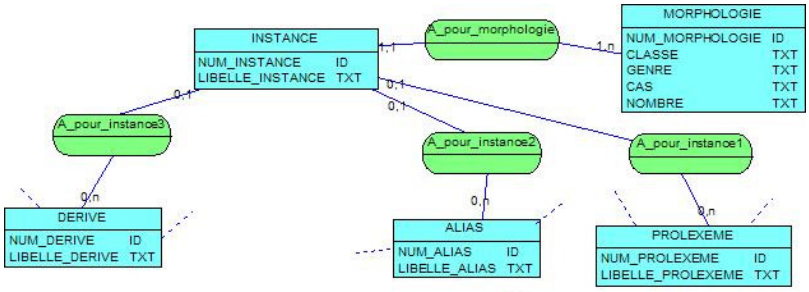

FIG. 5.16 – Les instances.

Les instances pour le français sont générées en utilisant le module *Inflect* d'Unitex [Paumier, 2003]. Le programme accepte en entrée une liste de mots au format DELA (voir chapitre 2). Voici un exemple d'un fichier que nous fournissons au module *Inflect* :

Onusien,N41+48226+48226+20394+22

Sur cette ligne, *Onusien* est le lemme et *N41* le code flexionnel ; les quatre traits qui suivent sont les numéros respectifs du pivot et du prolexème, puis éventuellement de l'alias et du dérivé.

Le programme fournit pour cette ligne le résultat suivant :

Onusiennes,Onusien.N+48226+48226+20394+22:fp
Onusiens,Onusien.N+48226+48226+20394+22:mp
Onusienne,Onusien.N+48226+48226+20394+22:fs
Onusien,Onusien.N+48226+48226+20394+22:ms

En récupérant ces informations et d'autres informations contenues dans la base, nous pouvons créer automatiquement les occurrences de l'entité *INSTANCE*.

Pour certaines langues, cette génération des instances risque de coûter cher en espace mémoire en raison d'un nombre important de cas, de genres et de nombres. Nous avons décidé que chaque langue développera sa propre stratégie pour générer ses instances. Chaque langue pourra soit stocker les formes fléchies dans cette entité, soit utiliser des systèmes morphologiques externes.

5.3 Modèle logique de données

Pour pouvoir créer les tables de notre base de données, nous avons traduit notre MCD des noms propres en MLD en appliquant les trois règles présentées dans la première partie. En traduisant directement notre MCD, nous aurions un modèle physique de données qui pourrait présenter les inconvénients suivants :

- La plupart des SGBD (MySQL, Access, etc.) possèdent une limitation sur la taille des tables. Si le nombre de langues et de données que l'on souhaite intégrer à notre base de données devenait assez grand, la taille des tables *PROLEXEME*, *ALIAS*, *DERIVE* et *INSTANCES* risquerait de dépasser cette taille limite.
- Certaines requêtes SQL, comme la recherche de données, la mise à jour, etc., risqueraient d'être longues.
- Selon les langues, certaines entités, associations ou propriétés ne seront jamais utilisées. En français, les associations *Accepte_comme4* et *A_pour_derivation3* ne seront pas utilisées, car les prolexèmes ne possèdent pas de dérivés de dérivés[2]. La propriété *CAS* de l'entité *MORPHOLOGIE* n'est pas utile pour le français. L'entité *DETERMINATION* et l'association *A_pour_détermination* ne seront jamais utilisées pour le serbe, car dans cette langue les groupes nominaux ne possèdent pas de déterminant.

A cause de ces diverses raisons, nous avons décidé de séparer chaque langue de notre dictionnaire afin de mettre en place pour chacune une structure plus adaptée. Notre modèle logique de données final comprendra deux parties : une partie commune aux langues traitées (figure 5.17) et une partie spécifique à chaque langue (figure 5.18). Le niveau méta-conceptuel et le niveau conceptuel appartiennent à la partie commune aux langues et seront reliés à la partie spécifique d'une langue donnée par la relation *Concept*.

Un schéma relationnel de données pour le français et un pour le serbe sont donnés en annexe A (voir page 151).

Conclusion

Nous avons utilisé la méthode Merise pour définir un MCD qui s'applique pour toutes les langues traitées. La traduction de ce MCD en MLD soulève un certain nombre de problèmes : limitation sur la taille des tables, rapidité des requêtes SQL, absence ou présence de tables spécifiques à certaines langues. A cause de ces différentes raisons, nous avons décidé de transformer ce MCD en un MLD comprenant deux parties : une partie commune aux langues et une partie particulière à chaque langue.

Le principal objectif de nos travaux est de développer des ressources linguistiques multilingues sur les noms propres qui puissent être mises à disposition de la communauté des chercheurs en TAL. Il nous paraît donc indispensable de développer un format d'exportation de notre base de données. Ce format devra être indépendant des systèmes d'exploitation et compatible avec la plupart des outils existants. Nous présenterons dans le chapitre suivant le modèle XML d'exportation de Prolexbase.

[2]En effet, comme nous l'avons expliqué au chapitre 3, les dérivés d'un prolexème sont des synonymes de celui-ci à une transformation près. Le dérivé *pasteurisation* est un dérivé du dérivé *pasteurisé*, mais ceux-ci ne figurent pas dans Prolexbase.

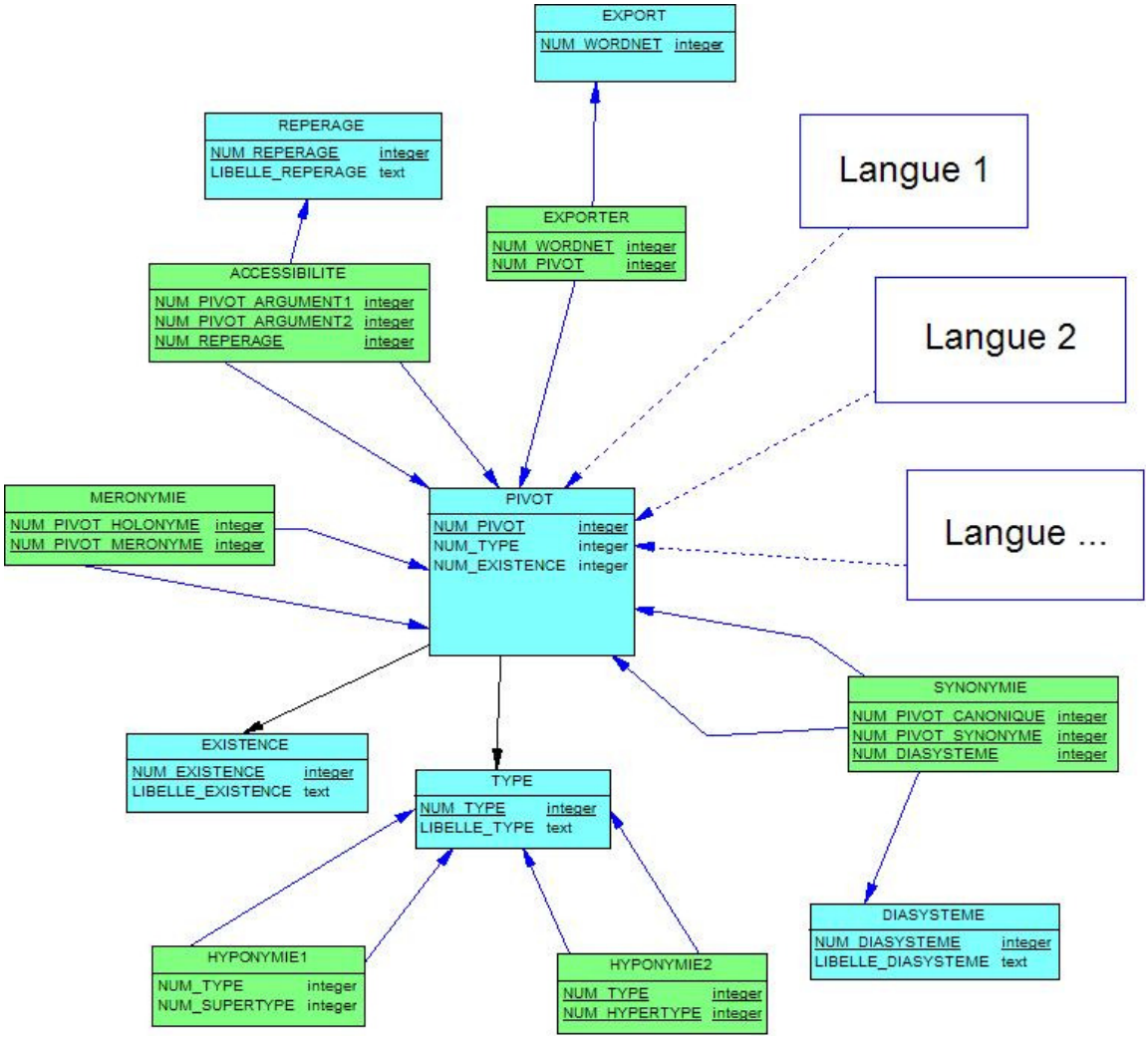

FIG. 5.17 – Le modèle relationnel de données : partie commune.

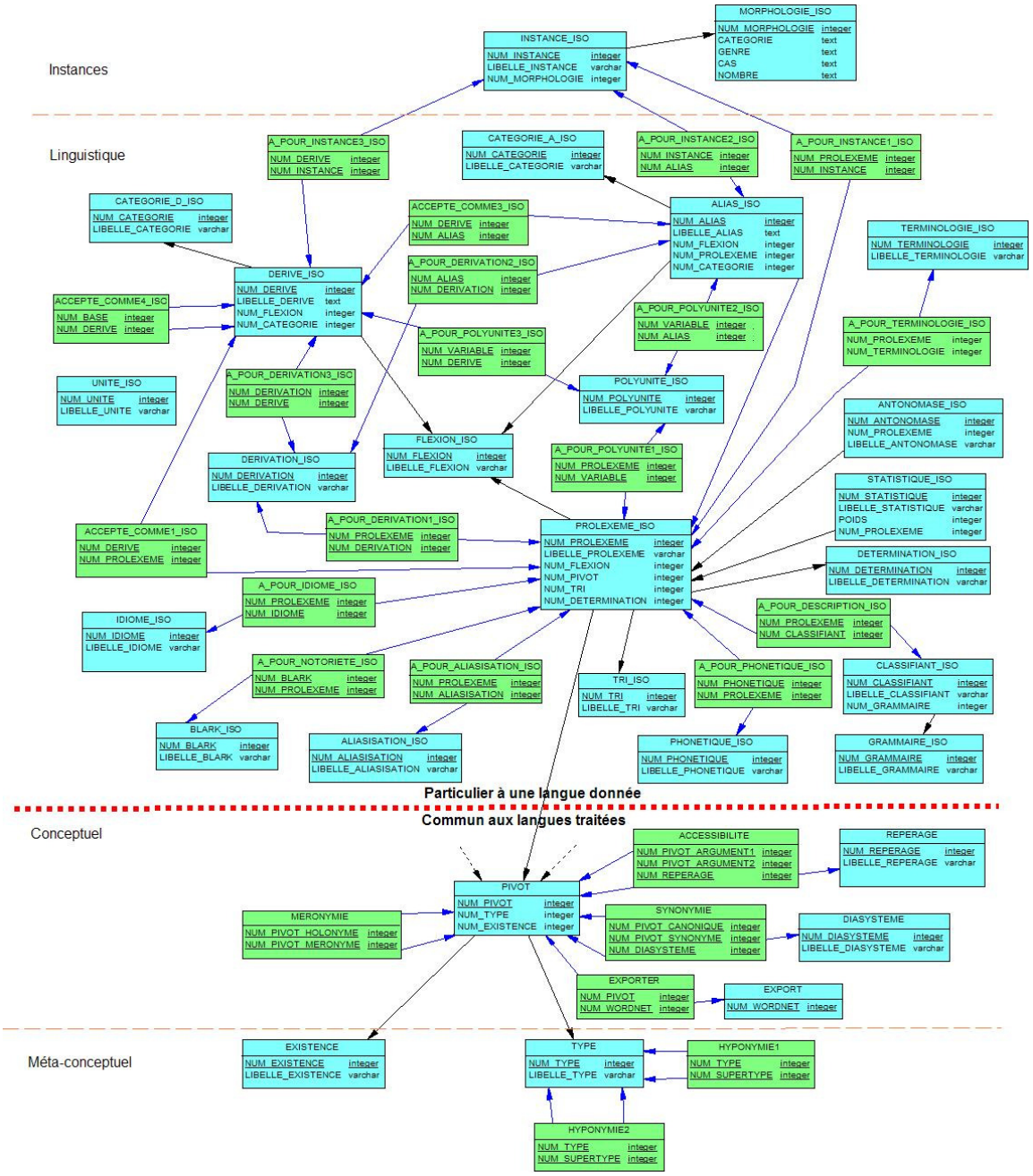

FIG. 5.18 – Le modèle relationnel de données : partie spécifique.

Chapitre 6

Exportation des données

Introduction

Dans l'article [Bouchou et al., 2005], nous avons proposé une version XML de notre base de données. Ce chapitre est consacrée à présenter une version respectant la norme TMF. Nous allons dans un premier temps présenter les caractéristiques de deux standards pour le partage et l'échange de données sur la Toile : la TEI et la TMF. Ensuite, nous présenterons le modèle d'échange et d'exportation de données pour Prolexbase que nous avons défini en nous inspirant de la TMF. Enfin, nous décrirons notre contribution au projet Outilex.

6.1 État de l'art

6.1.1 TEI

Apparue officiellement en 1988, la TEI[1] (Text Encoding Initiative) est née du besoin de normalisation du balisage de textes électroniques. L'objectif du projet est de développer un format indépendant des systèmes ou des logiciels informatiques, simple et facile pour les utilisateurs, permettant le partage et l'échange de données textuelles. Ce format doit être assez riche pour permettre à des chercheurs, provenant de différents domaines et de divers pays, de baliser leurs textes électroniques. Le projet est soutenu par de nombreuses institutions et associations, telles que l'Association for Computational Linguistics, l'Association for Literary and Linguistic Computing, le Social Sciences and Humanities Research Council du Canada, la Commission européenne, etc.

La DTD[2] de la TEI repose sur une architecture formée d'un ensemble de trois modules :

- *core tag sets* : il s'agit d'un ensemble de balises obligatoires pour toute DTD TEI (en-tête, paragraphe, divisions, etc.).
- *base tag sets* : ce module contient un ensemble de balises de base spécifique pour six catégories de textes : prose, poésie en vers, œuvre théâtrale, transcription du discours, dictionnaire et base terminologique. La figure 6.1 présente un exemple des balises TEI utilisées pour coder l'article *abandon* d'un dictionnaire.
- *additional tag sets* : il fournit un ensemble de balises additionnelles qui peut être utilisé pour n'importe quel type de textes. On trouve, par exemple, des balises pour repérer les noms de personnes (<*persName*>, etc.), de lieux (<*placeName*>, etc.) et d'organisations (<*orgName*>, etc.). La figure 6.2 donne un exemple avec la balise

[1] est accessible sur le site www.tei-c.org

[2] Le schéma est également défini en XML-Schema ou RELAX-NG

<*persName*> ; l'attribut *key* permet d'identifier le nom propre de façon unique dans un texte.

a.ban.don 1 /@"b&nd@n/ v [T1] 1 to leave completely and for ever; desert: The sailors abandoned the burning ship. 2 ... **abandon** 2 n [U] the state when one's feelings and actions are uncontrolled; freedom from control: The people were so excited that they jumped and shouted with abandon / in gay abandon. [LDOCE]

```
<superEntry>
  <form>
    <orth>abandon</orth>
    <hyph>a|ban|don</hyph>
    <pron>@"b&amp;nd@n</pron>
  </form>
  <entry n="1">
    <gramGrp>
      <pos>v</pos>
      <subc>T1</subc>
    </gramGrp>
    <sense n="1">
      <def>to leave completely and for ever ... </def>
      <!-- ... -->
    </sense>
    <sense n="2"> <!-- ... --> </sense>
  </entry>
  <entry n="2">
    <gramGrp>
      <pos>n</pos>
      <subc>U</subc>
    </gramGrp>
    <def>the state when one's feelings and actions are
      uncontrolled; freedom from control</def>
    <!-- ... -->
  </entry>
</superEntry>
```

FIG. 6.1 – Exemple de balise pour les dictionnaires.

```
<persName key="FDR1">
  <foreName>Franklin</foreName>
  <foreName>Delano</foreName>
  <surname>Roosevelt</surname>
</persName>
```

FIG. 6.2 – Exemple avec la balise <*persName*>.

La DTD proposée par la TEI pour les dictionnaires est destinée à la création de dictionnaires pour les humains. Les concepteurs de la TEI ont rencontré de nombreux problèmes lors de l'élaboration de cette DTD [Ide and Véronis, 1996]. Il était difficile de proposer une DTD qui soit à la fois assez générale pour décrire tous les dictionnaires et assez précise pour faire ressortir la spécificité de chaque dictionnaire.

Le chapitre 13[3] sur les bases terminologiques est devenu obsolète en raison de la sortie de la norme ISO 16642 ou TMF (Terminological Markup Framework) :

> *Since its first publication, this chapter has been rendered obsolete in several respects, chiefly as a result of the publication of ISO 12200, and a variant of it*

[3]sur le site www.tei-c.org consulté le 16/06/2006

(TBX) which has been recently adopted by LISA, the Localisation Industry Standard Association. Work is currently ongoing in the ISO community to define a generic platform for terminological markup (ISO CD 16642, TMF : Terminological Markup Framework), in the light of which it is anticipated that the recommendations of the present chapter will be substantially revised.

6.1.2 TMF

Le Terminological Markup Framework (TMF) ou la norme ISO 16642 [Romary, 2002] [Romary and Van Campenhoudt, 2001] propose un standard de représentation pour des données terminologiques multilingues en XML. Il définit un ensemble de contraintes que chaque langage de description de données terminologiques (TML) doit suivre. Chaque TML se caractérise par :

- un méta-modèle : il s'agit d'un squelette (figure 6.3) décrivant la structure de tout TML. Chaque entrée (TE) correspond à un ou plusieurs termes (TS) dans plusieurs langues (LS).
- un lien vers des catégories de données de la norme ISO 12620. Par exemple, la propriété *Content* précise le type du contenu, *DCName* un nom, etc.

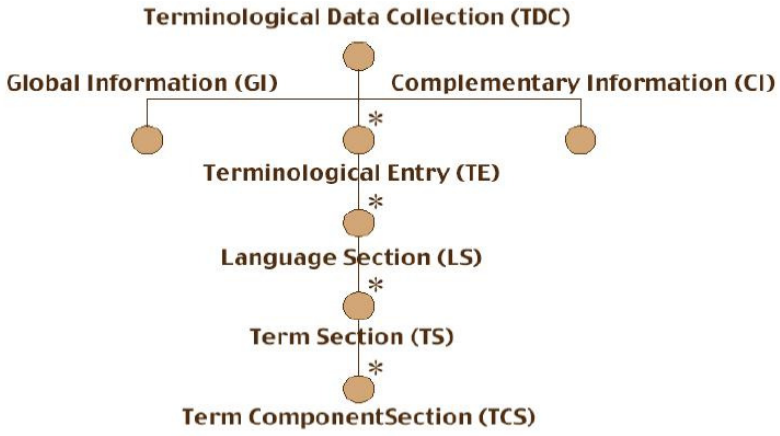

FIG. 6.3 – Le méta-modèle du TMF.

GMT (Generic Mapping Tool) est un outil permettant de représenter la structure du TMF sous le format XML. La figure 6.4 donne un exemple de représentation GMT. La représentation GMT repose sur deux balises :

- <struct> : permettant de définir chaque niveau du squelette structurel.
- <feat> : permettant de préciser le trait d'un nœud.

6.1.3 Discussion

La TEI propose un format de balisage figé qui n'est pas adapté pour modéliser notre base de données avec ses différents niveaux (niveau indépendant de la langue et niveau dépendant de la langue). Le balisage pour décrire les dictionnaires de la TEI peut être intéressant pour créer des dictionnaires monolingues pour les humains. Ce balisage ne permet pas de modéliser le niveau interlangue et nos relations sémantiques. Pour les bases terminologiques, la TEI recommande l'utilisation de la TMF.

```
<struct type="TE">
   <feat type="id">ID67</feat>
   <feat type="subjectField">manufacturing</feat>
   <feat type="definition">A value between 0 and 1 used in ...</feat>
   <struct type="LS">
      <feat type="lang">en</feat>
      <struct type="TS">
         <feat type="term">alpha smoothing factor</feat>
         <feat type="termType">fullForm</feat>
      </struct>
   </struct>
   </struct>
</struct>
```

FIG. 6.4 – Exemple de représentation avec le Generic Mapping Tool.

La TMF propose une architecture en deux niveaux : niveau dépendant de la langue et niveau indépendant de la langue, plus proche de notre modèle. Cependant, il manque une profondeur dans le niveau linguistique (indispensable pour la traduction des noms propres) et une modélisation pour des relations qui ne dépendent pas de la langue (synonymie, méronymie et accessibilité).

Comme la GMT propose un format de balisage assez flexible permettant de modéliser facilement n'importe quelle structure de données grâce à deux balises (*<struct>* et *<feat>*), nous pourrons l'adapter à notre architecture.

6.2 Le modèle XML de Prolexbase

Dans cette partie, nous allons présenter le format des fichiers de requête XML et le format des fichiers d'exportation de la base.

6.2.1 Fichier de requête XML

Toute application souhaitant extraire des données de Prolexbase doit envoyer une requête au format XML. Voici les balises qui doivent apparaître :

- *Request* : contient la structure de la requête.
- *Libelle* : nom propre sur lequel on souhaite faire une recherche.
- *RequestLanguage* : langue et le système d'écriture dans laquelle est écrit le nom propre que l'on recherche.
- *ProperName* : regroupe les informations que l'on souhaite avoir sur le nom propre :
  1. *Prolexeme* : spécifie si l'on souhaite récupérer le prolexème du nom propre que l'on recherche.
  2. *Type* : spécifie si l'on souhaite avoir le type du nom propre.
  3. *Existence* : spécifie si l'on souhaite avoir l'existence du nom propre.
  4. *Alias* : si l'on souhaite avoir tous les alias du prolexème.
  5. *Derivative* : si l'on souhaite avoir tous les dérivés du prolexème.
- *Lemmas* : regroupe les informations sur la catégorie et la classe du prolexème, de chaque alias et chaque dérivé.

```
<?xml version='1.0' encoding='ISO-8859-1' standalone='no' ?>
<!DOCTYPE Request SYSTEM "../requetes/Requete_DTD.dtd">
<Request>
      <Libelle >Organisation des nations unies</Libelle>
      <RequestLanguage>fr</RequestLanguage>
      <ProperName>
            <Prolexeme  status='ON'/>
            <Type status='ON'/>
            <Existence status='ON'/>
            <Alias status='ON' />
            <Derivative status='OFF' />
      </ProperName>
      <Lemmas>
            <Lemma status='ON'/>
            <Pos status='ON' />
            <Category status='ON' />
      </Lemmas>
      <Inflexions>
            <Form status='ON'/>
            <Gender status='ON' />
            <Number status='ON' />
      </Inflexions>
      <AnswerLanguage>
            <Language>fr</Language>
      </AnswerLanguage>
</Request>
```

FIG. 6.5 – Requête XML.

1. *Lemma* : libellé du prolexème, de l'alias ou du dérivé.
2. *Pos* : classe du prolexème, de l'alias ou du dérivé.
3. *Category* : catégorie du prolexème, de l'alias ou du dérivé.

– *Inflexions* : précise si l'on souhaite avoir les formes fléchies.

1. *Form* : libellé de la forme fléchie.
2. *Gender* : genre de la forme fléchie.
3. *Number* : nombre de la forme fléchie.

– *AnswerLanguage* : regroupe les langues dans lesquelles on souhaite faire une recherche.

1. *Language* : langue dans laquelle s'effectue la recherche.

La figure 6.5 présente un exemple de requête avec le nom propre *Organisation des nations unies*. La valeur *ON* de l'attribut *statut* de la balise *Alias* précise que le résultat doit faire apparaître les alias. La valeur *OFF* de l'attribut *Derivative* indique que les dérivés ne doivent pas apparaître dans le fichier résultat.

6.2.2 Fichier d'exportation XML

Pour créer le modèle XML d'exportation de notre base de données, nous nous sommes inspirés de la représentation GMT de la TMF, car l'utilisation d'une norme garantit la portabilité et la compatibilité de notre format avec la plupart des systèmes existants. De plus, cette norme permet de modéliser facilement n'importe quelle structure de données en se basant sur deux balises. La figure 6.6 présente le modèle sous la forme d'une arborescence.

Voici la liste des balises pouvant apparaître dans ce fichier :

– *Prolex* : contient les entrées.

– *Pivot* : Il s'agit d'une entrée qui ne dépend pas de la langue et qui correspond à la TE du méta-modèle de la TMF. Cette balise comprend les attributs suivants :

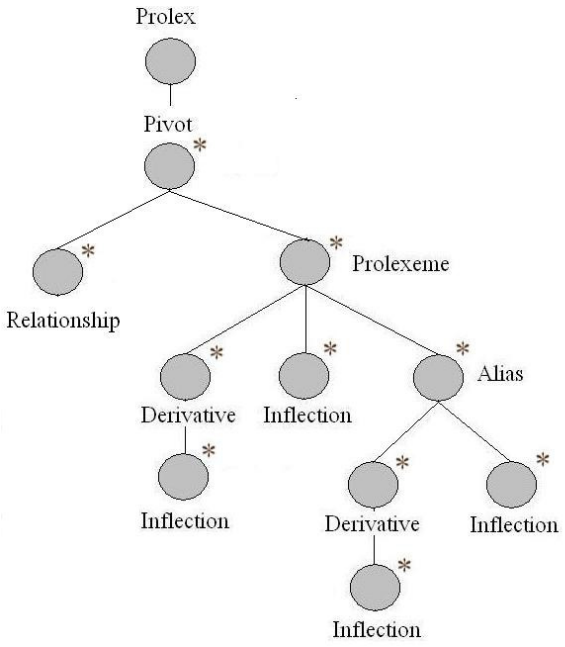

FIG. 6.6 – Le modèle de Prolexbase.

1. *type* : indique type du pivot.
2. *existence* : indique existence du pivot.
3. *identifier* : précise le numéro du pivot.

– *Relationship* : précise les relations du pivot avec d'autres pivots. On y trouve les attributs :

1. *relation* : nom de la relation (méronymie, synonymie ou accessibilité).
2. *identifier* : indique le numéro du pivot en relation.
3. *argument* : précise quel est l'argument 1 et l'argument 2 de la relation. Soit deux pivots P_1 et P_2 et R une relation entre deux arguments telle que R(P_1, P_2). Dans le cadre d'une relation de synonymie, P_1 est appelé le synonyme et P_2 est appelé le canonique. Pour une relation de méronymie, P_1 est appelé le méronyme et P_2 est appelé l'holonyme. Dans une relation d'accessibilité, P_1 est appelé l'argument 1 et P_2 l'argument 2.
4. *context* : précise le diasystème pour une relation de synonymie, le repérage pour une relation d'accessibilité.

– *Prolexeme* : Cette balise correspond à une entrée dans une langue donnée (équivalente à la LS du méta-modèle de la TMF). Elle accepte les attributs suivants :

1. *language* : indique la langue à laquelle appartient le prolexème.
2. *lemma* : libellé du prolexème.
3. *pos* : donne la catégorie grammaticale du prolexème.
4. *category* : précise qu'il s'agit d'un nom propre.

– *Alias* : regroupe les alias du prolexème. On trouve les attributs suivants :

1. *lemma* : lemme de l'alias.

2. *pos* : donne la catégorie grammaticale de l'alias.
3. *category* : précise la catégorie de l'alias (acronyme ou sigle, abréviation, etc.).

– *Derivative* : regroupe les dérivés du prolexème ou des alias. On trouve les attributs suivants :
1. *lemma* : lemme du dérivé.
2. *pos* : donne la catégorie grammaticale du dérivé.
3. *category* : précise la catégorie du dérivé (nom relationnel, adjectif relationnel, préfixe, etc.).

– *Inflection* : contient une forme fléchie pouvant provenir soit d'un dérivé, soit du prolexème, soit d'un alias. On trouve les attributs suivants :
1. *form* : libellé de la forme fléchie.
2. *gender* : donne le genre de la forme fléchie.
3. *number* : contient le nombre (singulier, pluriel, etc.).
4. *case* : pour les langues casuelles.

La figure 6.7 donne le résultat de la requête XML de la figure 6.5. Un exemple complet avec le prolexème *États-Unis d'Amérique* est donné en annexe C. La figure 6.8 donne un exemple de relation de méronymie (la *France* est un méronyme de l'*Europe*) et de relation d'accessibilité (*Paris* est la capitale de la *France*).

6.3 Implémentation

Nous avons encadré un stage d'une étudiante de Master qui a développé une interface en PHP permettant à chaque visiteur ou à chaque application de soumettre des requêtes à notre base de données.

La figure 6.9 présente l'architecture globale permettant d'exporter nos données. Le visiteur précise dans une interface web les données qu'il souhaite extraire de Prolexbase. L'interface génère à l'aide d'un module de construction de requête un fichier de requête en format XML (voir section 6.2.1) qui est envoyé à un module de traitement de requête. Ce module interroge la base de données et renvoie le résultat à l'interface sous la forme d'un fichier XML (voir section 6.2.2). Toute application peut directement soumettre un fichier de requête XML au module de traitement de requêtes et obtenir un fichier XML résultat.

6.4 Une contribution effective

Avant la conception de ce format d'exportation, nous avions contribué au Projet Outilex par la création des dictionnaires Prolex-Toponymes et Prolex-PaysCapitales. Ces dictionnaires ont été réalisés en extrayant des toponymes français de notre base de données sous le format DELAF (voir la section 2.1 page 32).

Le dictionnaire Prolex-Toponymes comprend 9 225 entrées, dont 2 110 toponymes, 3 415 gentilés, 3 407 adjectifs toponymiques, 12 préfixes toponymiques et 281 hydronymes. Voici un extrait de ce dictionnaire :

Lyon,.N+PR+DetZ+Toponyme+Ville:ms:fs
lyonnais,lyonnais.A+Toponyme+Ville:ms:mp
Lyonnais,Lyonnais.N+PR+Hum+Toponyme+Ville:ms:mp
Mékong,.N+PR+Hydronyme:ms
Vallée des Rois,.N+PR+Toponyme+Region:fs

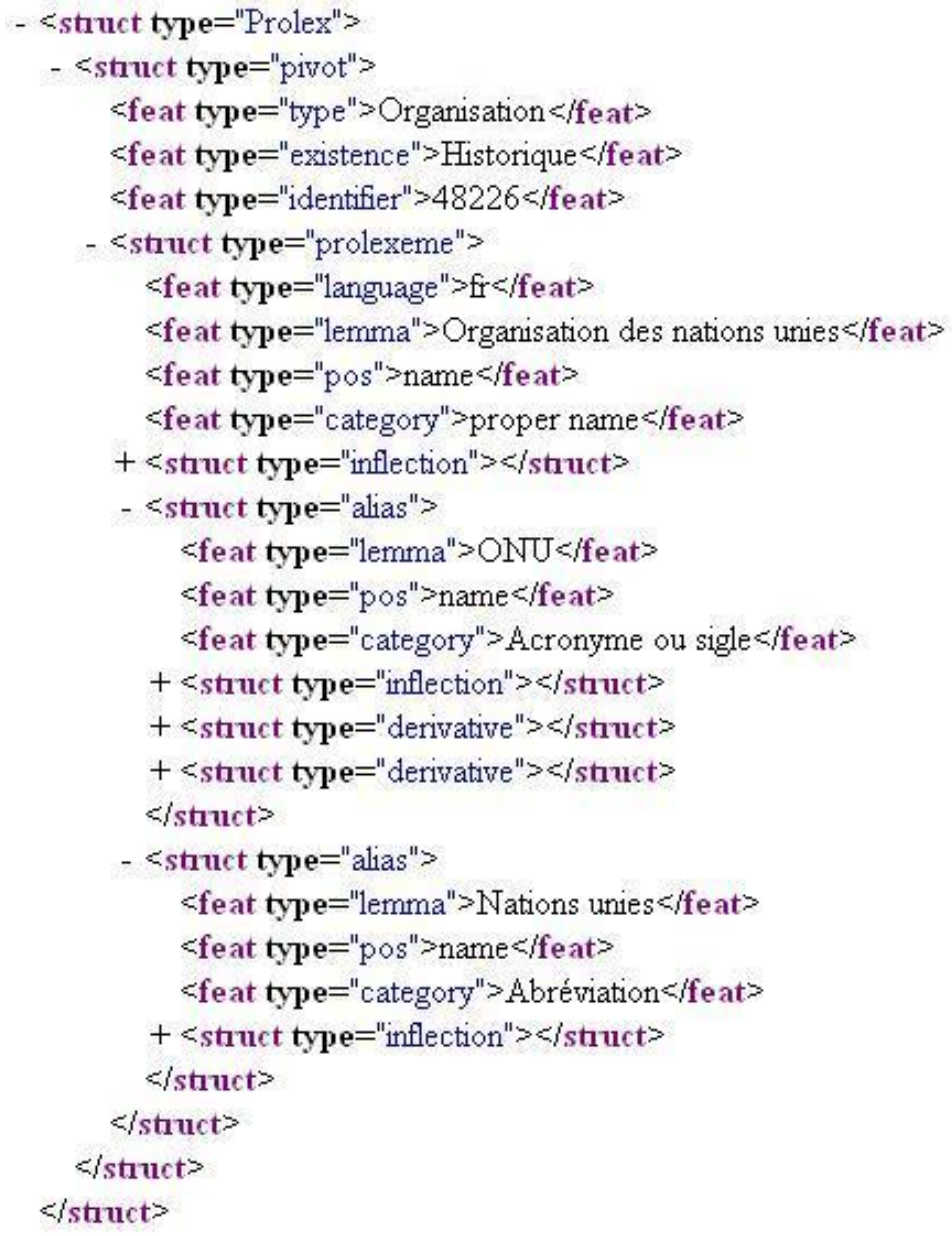

FIG. 6.7 – Résultat d'une requête XML.

```
<struct type="Prolex">
    <struct type="pivot">
        <feat type="type">Country</feat>
        <feat type="existence">Historical</feat>
        <feat type="identifier">27</feat>
        <struct type="relationship">
            <feat type="relation">meronymy</feat>
            <feat type="identifier">47947</feat>
            <feat type="argument">arg1</feat>
        </struct>
        <struct type="relationship">
            <feat type="relation">accessibility</feat>
            <feat type="identifier">38558</feat>
            <feat type="argument">arg2</feat>
            <feat type="context">capital</feat>
        </struct>
        <struct type="prolexeme">
            <feat type="language">fr</feat>
            <feat type="lemma">France</feat>
            ...
        </struct>
    </struct>
    <struct type="pivot">
        <feat type="type">Supranational</feat>
        <feat type="existence">Historical</feat>
        <feat type="identifier">47947</feat>
        <struct type="prolexeme">
            <feat type="language">fr</feat>
            <feat type="lemma">Europe</feat>
            ...
        </struct>
    </struct>
    <struct type="pivot">
        <feat type="type">City</feat>
        <feat type="existence">Historique</feat>
        <feat type="identifier">38558</feat>
        <struct type="prolexeme">
            <feat type="language">fr</feat>
            <feat type="lemma">Paris</feat>
            ...
        </struct>
    </struct>
</struct>
```

FIG. 6.8 – Exemple de relations.

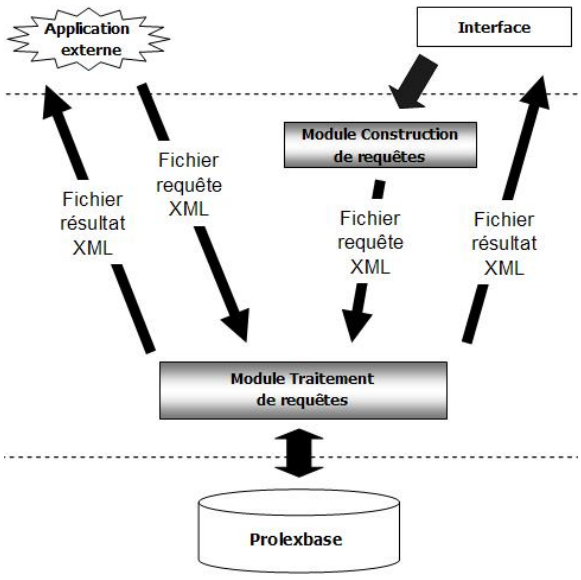

FIG. 6.9 – Architecture d'exportation de Prolexbase.

La figure 6.10 présente la liste des traits utilisés dans le dictionnaire Prolex-Toponymes.

+PR	Les noms propres
+Hum	Les humains (les gentilés)
+Toponyme	Les toponymes et les gentilés
+Ville	Les villes et les gentilés
+Region	Les régions et les gentilés
+Pays	Les pays (indépendants) et les gentilés
+Hydronyme	Les hydronymes
+DetZ	Les noms sans déterminant

FIG. 6.10 – Traits du dictionnaire Prolex-Toponymes.

Le dictionnaire Prolex-PaysCapitales regroupe les 191 pays indépendants avec leur capitale, les gentilés et les adjectifs toponymiques. Nous avons aussi ajouté dans ce dictionnaire les régions qui sont assimilées à des pays, comme par exemple le Royaume-Uni, les émirats, etc. Il comporte 3 092 entrées, dont 592 toponymes, 1 250 gentilés, 1 240 adjectifs toponymiques et 10 préfixes toponymiques. Voici quelques entrées de ce dictionnaire :

Athènes,.N+PR+DetZ+Toponyme+Ville+IsoGR:ms:fs
athénien,athénien.A+Toponyme+Ville+IsoGR:ms
Roumains,Roumain.N+PR+Hum+Toponyme+Pays+IsoRO:mp
Roumanie,.N+PR+Toponyme+Pays+IsoRO:fs
Suède,.N+PR+Toponyme+Pays+IsoSE:fs
suédois,suédois.A+Toponyme+Pays+IsoSE:ms:mp

Nous avons précisé pour chaque entrée le code ISO de son pays. Par exemple, *IsoRO* pour la *Roumanie*, *IsoSE* pour la *Suède*, etc.

Ces deux dictionnaires sont gratuitement mis à disposition des utilisateurs d'Unitex et téléchargeables à l'adresse suivante : `http://tln.li.univ-tours.fr/Tln_Unitex.html`.

Ils sont stockés dans des fichiers enregistrés au format Unicode Little-Endian ou UTF-16, qui est le format des fichiers utilisés par Unitex [Paumier, 2006].

Conclusion

Il est possible de formuler une requête en format XML pour rechercher des noms propres dans Prolexbase. Le résultat des requêtes est un fichier XML dans le format GMT de la TMF.