Informatisation du Dictionnaire Explicatif et Combinatoire : le projet NADIA-DEC

Gilles Sérasset

Informatisation du Dictionnaire Explicatif et Combinatoire : le projet NADIA-DEC

Gilles Sérasset
GETA-CLIPS-IMAG (UJF & CNRS)
BP 53
F-38041 Grenoble Cedex 9
Tél. : (+33) 76 51 43 80
Fax. : (+33) 76 51 44 05
e-mail : Gilles.Serasset@imag.fr

Introduction

Depuis novembre 1994, le GETA-CLIPS (Université de Grenoble) et le GRESLET (Université de Montréal) se sont associés pour mettre en œuvre un projet visant l'informatisation du Dictionnaire Explicatif et Combinatoire développé par Igor Mel'č uk ([Mel'č uk et al. 1984, Mel'č uk et al. 1988, Mel'č uk et al. 1992, Mel'č uk et al. 1995]). Ce projet est actuellement soutenu en tant qu'action de recherche concertée du réseau Lexicologie Terminologie et Traduction de l'AUPELF-UREF.

Ce projet a deux intérêts majeurs : (a) mettre à disposition des lexicologues, des outils simplifiant la création d'articles du Dictionnaire Explicatif et Combinatoire et (b) mettre à l'épreuve un système universel de gestion de bases lexicales multilingues (SUBLIM/NADIA) développé au GETA ([Sérasset 1994]).

Dans cet article, nous présentons les objectifs du projet NADIA-DEC. Nous montrons ensuite les principes de la plate-forme informatique (SUBLIM) définie au GETA, puis nous présentons la structure (relativement complexe) d'un article du DEC en montrant notamment comment on peut la coder informatiquement grâce au langage spécialisé DARLING du système SUBLIM.

Nous terminerons cet article en donnant l'état d'avancement du projet les perspectives qu'il ouvre pour une recherche future.

Présentation du projet

Le projet NADIA-DEC est basé sur les travaux de définition d'un système universel de bases de données lexicales multilingues au laboratoire GETA-CLIPS et sur les travaux de définition du dictionnaire explicatif et combinatoire du français contemporain au laboratoire GRESLET.

Ce projet vise la création d'une version informatisée du dictionnaire explicatif et combinatoire du français contemporain. Cette version devra contenir l'ensemble des informations présentes dans le DEC sous une forme aussi structurée que possible. Elle s'appuie donc sur le système de gestion de bases lexicales multilingues SUBLIM défini au GETA-CLIPS.

Ce projet répond à plusieurs motivations de la part de chacun des partenaires. D'une part, il permet de tester le système SUBLIM en l'utilisant pour un dictionnaire mettant en œuvre des structures complexes. D'autre part, les informations contenues dans le dictionnaire explicatif combinatoire présentent une richesse que l'on ne trouve dans aucun autre dictionnaire informatisé. Enfin, la mise en œuvre de ce projet suppose la création d'outils informatiques simplifiant la gestion d'un tel dictionnaire.

Pour atteindre ces différents objectifs, nous souhaitons non seulement informatiser une version du DEC, mais aussi informatiser sa chaîne de production afin de faire en sorte que le DEC existe d'abord sous forme informatique puis sous une forme imprimable. La méthodologie du projet est données par la figure 1.

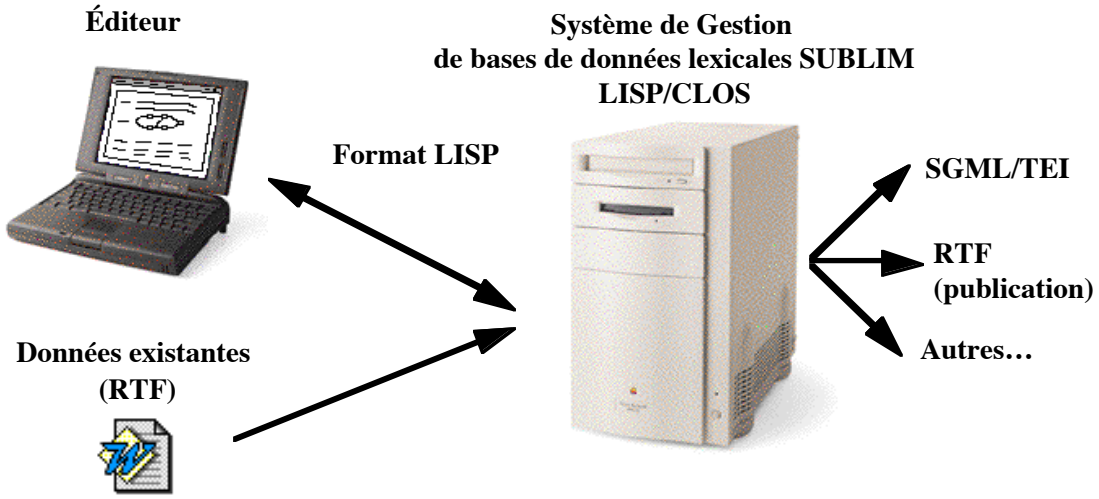

Figure 1 : méthodologie du projet Nadia-DEC

Ainsi, le travail à effectuer se décompose en différentes étapes :

- création d'un éditeur informatique spécialisé pour le DEC,
- création (en utilisant le système SUBLIM) d'un système de gestion des données lexicales. Les vérifications des différentes contraintes sur les données seront vérifiées à ce niveau,
- réalisation d'un mécanisme d'import des données existantes, actuellement au format RTF (Rich Text Format),
- réalisation d'un module d'export vers différents formats utilisables informatiquement (SGML/TEI, format LISP…) et pour la publication papier (RTF, MIF…).

Cet article présente la structure informatique utilisées à la fois pour l'éditeur et pour le système central de gestion des données lexicales.

Le projet SUBLIM

L'étude détaillée des travaux menés dans le cadre de la construction de systèmes de bases lexicales multilingues ou monolingues informatisées nous a permis de faire les constatations suivantes :

- De nombreux projets existent, qui présentent des particularités intéressantes. Il est donc vain de proposer un nouveau système qui ne puisse prendre en charge les méthodes de ces divers projets.
- Les projets multilingues existants figent en général l'approche lexicale que l'on peut utiliser pour réaliser le multilinguisme. Ainsi, le projet Esprit Multilex (par exemple)

ne permet l'utilisation que d'une approche bilingue et ne peut donc satisfaire les lexicologues tentés par des approches interlingues.
- Les projets existants offrent parfois au lexicologue la possibilité de personnaliser les structures linguistiques utilisées dans le lexique, mais de manière insuffisante. Ainsi, si, dans le projet MULTILEX, il est possible de définir ses propres traits linguistiques, il est impossible de s'affranchir de l'utilisation d'une structure de traits typés pour coder les informations lexicales.

Partant de ces constatations, [Sérasset 1994] propose un système informatique de gestion de bases lexicale multilingue qui réponde à ces différentes critiques. Ainsi, ce système peut satisfaire les lexicologues souhaitant coder des informations lexicales complexes utilisant des structures informatiques diverses (arbres, automates, graphes, structures de traits, ensembles…) dans le cadre de bases lexicales monolingues ou multilingues, basées sur les approches par dictionnaires de transfert ou par dictionnaires interlingues.

Pour atteindre cet objectif, le système SUBLIM demande au lexicographe de définir la structure linguistique du dictionnaire qu'il souhaite informatiser par deux langages ayant des objectifs différents :
- Dalex (Définition de l'Architecture LEXicale) permet de définir les dictionnaires mis en œuvre dans la base lexicale et leurs liens. Il permet, entre autres, de définir l'approche lexicale utilisées (interlingue ou par transfert).
- Darling (Définition de l'ARchitecture LINGuistique) permet, pour chaque dictionnaire, de définir les structures informatiques utilisées dans les entrées. Le lexicographe peut utiliser de nombreuses structures prédéfinies comme les arbres, graphes, automates, structures de traits…

Nous ne donnerons pas dans cet article plus de détails sur ces langages. Le lecteur en trouvera néanmoins des exemples d'utilisation dans la suite de cet article.

Le dictionnaire explicatif et combinatoire du Français contemporain

Une unité de ce dictionnaire (lexème) est un sens de mot ou de locution. Cette unité lexicale est associée à une unité morphologique, à une définition, à d'éventuelles connotations, à un régime, à des exemples, et à des fonctions lexico-sémantiques. Nous lui affectons de plus un numéro de sens qui l'identifie parmi les différents sens d'une entrée.

Un lexème peut aisément être codé comme une structure de traits :

```
(def-linguistic-class lexème
  (feature-structure
    ((numéro          numéro)
     (UMorph          UMorph)
     (définition      définition)
     (connotations    connotations)
     (régime          régime)
     (exemples        exemples)
     (lexico-sem-fns lex-sem-fns))
  ))
```

Une unité morphologique comprend une forme graphique et des informations morphologiques. Elle peut être reliée à plusieurs lexèmes qui lui sont associés de manière arborescente :

CŒUR, nom, masc.

I.1 a.	Organe principal de la circulation sanguine d'une personne… [*le cœur de Jean*]
1 b.	Organe principal de la circulation sanguine d'un animal… [*le cœur de lion*]
2.	Produit alimentaire … [*le cœur de veau*]
3.	Partie de la poitrine d'une personne … [*Il a serré son fils sur son cœur*]
4 a.	Organe imaginaire des sentiments … [*Le cœur espère toujours*]
4 b.	Organe imaginaire de l'intuition … [*Son cœur le lui dit*]
5 a.	… propriété de la personnalité … [*un cœur de glace*]
5 b.	Personne possédant le cœur I.5a [*Vous devez la vie à un noble cœur, à un homme vaillant*]
II.1 a.	Partie principale d'une unité fonctionnelle… [*le cœur du bateau*]
1 b.	Élément principal [*le cœur du problème*]
2 a.	Partie centrale d'un espace… [*le cœur du royaume*]
3.	Objet… ayant la forme du cœur I.1a [*un cœur en papier*]
4.	Une des quatre couleurs 2 des cartes à jouer… [*l'as de cœur*]
III.	Organe imaginaire des nausées … [*Cette senteur lui tournait le cœur*]

Aussi, nous définirons une unité morphologique comme un arbre portant des informations morphologiques à la racine et des lexèmes sur les feuilles.

```
(def-linguistic-class UMorph
  (tree :root   Morphological-information
        :leaves lexème))
```

L'information morphologique associée à la racine de cet arbre ne comporte qu'une graphie, une catégorie, un genre et un nombre.

```
(def-linguistic-class Morphological-information
  (feature-structure
     ((graph     string)
      (catégorie cat)
      (genre     gnr)
      (nombre    nbr))))

(def-linguistic-class cat
  (one-of (nom verbe adjectif adverbe… )))
(def-linguistic-class gnr
  (one-of (masculin féminin)))
(def-linguistic-class nbr
  (one-of (singulier pluriel)))
```

Une définition du DEC n'est pas une simple chaîne de caractères :

I.1a. *Cœur de X* = Organe principal de la circulation sanguine d'une personne X qui se trouve dans la partie centrale du corps II.1d de X et qu'on représente symboliquement comme ayant la forme ♡.

Mis à part le fait que l'on y trouve une image, on peut remarquer qu'elle se compose de deux parties principales. La première (indiquée en italiques) présente un usage du lexème dans une locution où les différents arguments du prédicat représenté sont indiqués sous forme de variable. La seconde est une explicitation du sens du lexème, elle réutilise les variables de la première partie. On remarque aussi qu'elle fait référence à des lexèmes définis par ailleurs dans le dictionnaire (corps II.1d).

Nous simplifierons cette structure en la décomposant simplement en deux chaînes de caractères, l'une contenant la forme du prédicat, l'autre contenant sa définition :

```
(def-linguistic-class définition
  (feature-structure
    ((prédicat  string)
     (explicite string))))
```

Après cette partie de définition, on trouve éventuellement une partie consacrée aux connotations :

Connotations

1) Cœur I.1a est le siège des sentiments [voir CŒUR I.4a].
2) Cœur I.1a est le siège de l'intuition [voir CŒUR I.4b].
3) Cœur I.1a qui bat 1 représente la vie [voir les phrasèmes correspondants dans CŒUR I.1a].

Cette partie se présente comme une liste de connotations. Chacune est donnée sous forme de chaîne de caractères faisant référence à au moins un lexème. Il est donc intéressant, dans une version informatique de ce dictionnaire, de conserver à la fois la connotation sous forme de chaîne de caractères et sous forme d'un ensemble de liens vers d'autres lexèmes :

```
(def-linguistic-class connotations
  (set-of connotation))
(def-linguistic-class connotation
  (feature-structure
    ((texte    string)
     (réfère-à (set-of ((link :target lexème)))))))
```

À la suite de ces éventuelles connotations, on trouve le régime du prédicat. Ce régime donne les informations sur les différentes réalisations syntaxiques des arguments du prédicat. Le régime est donné sous forme de tableau dont les colonnes correspondent aux arguments et les lignes aux différentes réalisations. Certaines combinaisons ainsi établies étant non valides, on en reprend ensuite la liste, en indiquant leur impossibilité. On reprend aussi un certain nombre de ces combinaisons pour en donner des exemples (l'exemple suivant est tiré de l'entrée "enseigner 1") :

1. *X enseigne Y à Z* = X, censé avoir la qualification professionnelle dans le domaine Y, cause que Z apprenne III.1b Y en transmettant, méthodiquement et dans un cadre officiel, à Z des connaissances (portant sur) Y ou des techniques (portant sur) Y [≠ CausConv_{21}(*apprendre III.1b*)].

Régime

1 = X	2 = Y	3 = Z
1. N	1. N 2. *à* V_{inf}	1. *à* N 2. **rare** N

1) $C_{2.2}$ sans $C_{3.1}$,2) $C_2 + C_{3.2}$} : impossible

$C_1 + C_2$: *Pierre enseigne la grammaire <la couture >/ à faire cela*

$C_1 + C_2 + C_3$: *Pierre enseigne la grammaire à ses élèves*

La structure correspondant à cette partie est beaucoup plus compliquée que celle des parties précédentes. En effet, cette présentation n'est que le reflet, imprimable, d'une structure complexe où l'on retrouve l'ensemble des combinaisons possibles de réalisations d'arguments. On peut donc représenter cette partie de deux manières :

- en restant proche de sa forme papier. On a alors un tableau et une liste des combinaisons impossibles.
- en représentant cette structure de manière plus abstraite. On peut ainsi la représenter par un automate dont chaque chemin forme une combinaison valide.

Si l'on choisit la seconde solution, le régime donné en exemple sera représenté par l'automate donné en figure 2.

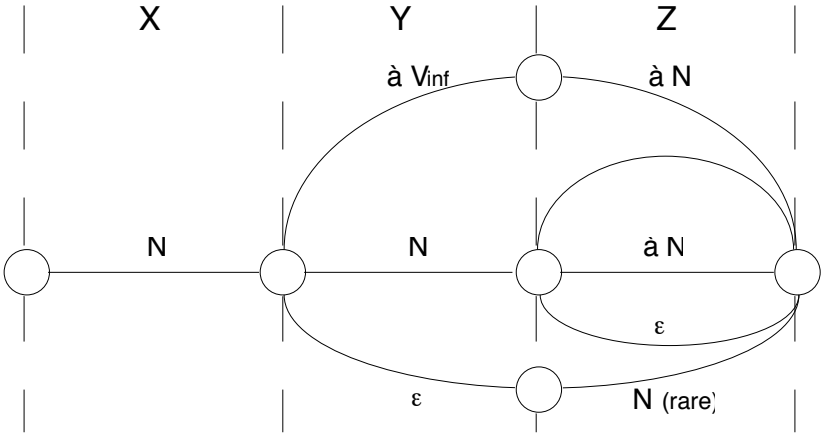

Figure 2 : Régime d'enseigner 1, sous forme d'automate

Pour exprimer cette solution, nous utiliserons la structure logique d'automate prédéfinie dans le système SUBLIM où l'on peut définir des contraintes sur les classes acceptables en décoration des différents éléments de l'automate (arcs, nœuds ou états, états initiaux, états finals).

Ainsi, la structure du régime s'exprimera sous forme d'une structure de traits regroupant l'automate des régimes, l'ordre d'apparition des arguments et l'ensemble des exemples :

```
(def-linguistic-class régime
  (feature-structure
    ((automate       automate-régime)
     (argument-order (list-of (string)))
     (exemples       exemples-régime))))
(def-linguistic-class automate-régime
  (automaton :arcs réalisation-argument))
(def-linguistic-class exemples-régime
  (set-of ((feature-structure
             ((réalisations (list-of (string)))
              (exemple      string))))))
```

La partie la plus importante de ce dictionnaire réside dans l'ensemble des fonctions lexicales du lexème. Leur meilleur définition est donnée, en première partie du DEC, par l'auteur, Igor Mel'č uk :

> Les fonctions lexicales (FL) présentent l'ensemble de la cooccurrence lexicale restreinte intéressant le lexème considéré. Elles constituent une innovation lexicographique qui permet de décrire d'une façon systématique un vaste ensemble de locutions plus ou moins figées qui ne sont quand même pas des expressions idiomatiques *stricto sensu.* Il s'agit, par exemple, des locutions comme une FERME intention, une résistance ACHARNÉE, un argument DE POIDS, un bruit INFERNAL, un désir ARDENT, une envie FOLLE, une règle STRICTE, une vérité INCONTESTABLE, où des adjectifs bien spécifiques doivent être employés avec les différents noms pour exprimer la même idée d'intensification. Comme autre exemple de locution de ce type, on peut citer les expressions DONNER une leçon, FAIRE un pas, COMMETTRE un crime, PORTER une accusation, etc., où des verbes sémantiquement vides (ou presque vides) différents doivent être choisis en fonction du nom d'action pour lier le nom d'agent en tant que sujet grammatical au nom d'action en tant que complément d'objet direct.

L'écriture générale d'une FL est de la forme : **f**(X) = Y, où **f** est la FL, X est son argument (un lexème ou bien une locution), et Y est la valeur de la FL **f** pour cet argument, c'est à dire l'ensemble des expressions linguistiques qui peuvent exprimer le sens ou le rôle syntaxique donné (noté par **f**) auprès de X.

Comme ce dictionnaire est imprimé, les expressions linguistiques sont données sous une forme linéaire :

MÉPRIS, nom, masc.

I. Attitude émotionnelle défavorable… [*le mépris pour ce corrupteur*]

[…]

Fonctions lexicales

$Caus_3Func_1$	:	engendrer [ART s chez N] [*La familiarité engendre le mépris*]
$Caus_{(3)}Func_1$	:	apprendre, inculquer [ART s à N] [*Jean inculque à ses étudiants le mépris de l'hypocrisie ; Son attitude partiale envers ses employés apprend à ces derniers le mépris de leur chef*]
$Caus_{(2/3)}Func_1$	:	inspirer [ART s à N] [*Cet événement inspire aux travailleurs le mépris de leur patron ; L'argent inspirait à ce philosophe un tel mépris qu'il a donné son héritage à son frère ; L'hypocrisie de Jean leur inspirait un profond mépris*]

Mais la structure interne de ces expressions linguistiques est un arbre syntaxique donnant la construction de ces expressions linguistiques et de l'argument X pour réaliser la fonction **f**. Ainsi, la structure interne de $Caus_3Func_1$(Mépris I) est l'arbre donné ci-dessous :

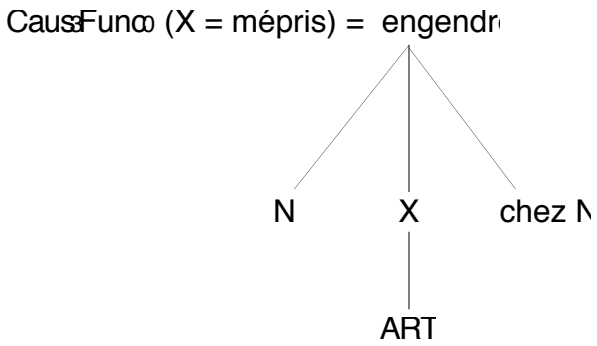

Figure 3 : Structure interne d'une expression linguistique, valeur de fonction lexicale

Une fonction lexicale représente donc un lien entre un lexème et une expression linguistique complexe comportant d'autres lexèmes. Aussi, la valeur de ces fonctions lexicales peut être représentée comme un ensemble d'arbres dont certains nœuds sont des variables, et d'autres sont des lexèmes.

Il faut aussi représenter les fonctions lexicales. En effet, s'il y a un nombre limité de fonctions lexicales de base, on trouvera des fonctions composées dans les différents articles de dictionnaire.

Prenons un exemple : les fonctions $\mathbf{Oper_1}$, $\mathbf{Oper_2}$… ont pour valeur les verbes sémantiquement vides qui prennent le nom du premier, deuxième… actant comme sujet grammatical et C_0 (leur argument) comme complément d'objet principal :

$\mathbf{Oper_1}$(*attention*) = *faire*
$\mathbf{Oper_2}$(*attention*) = *attirer*
$\mathbf{Oper_1}$(*conseil*) = *donner*

Oper$_2$(*conseil*) = *recevoir*
Oper$_1$(*aide*) = *prêter, accorder*
Oper$_2$(*aide*) = *recevoir*

La fonction Caus représente la notion : "faire en sorte que quelque chose ait lieu". Elle s'emploie le plus souvent en combinaison avec d'autres fonctions. Ainsi, si **Oper$_1$**(*désespoir*) = *éprouver, ressentir, avoir*, **CausOper$_1$**(*désespoir*) représente "faire en sorte que quelqu'un éprouve du désespoir". Donc **CausOper$_1$**(*désespoir*) = *pousser, réduire* [*qqn au désespoir*], *jeter* [*qqn dans le désespoir*], *frapper* [*qqn de désespoir*].

Il n'est donc pas possible de représenter chaque fonction lexicale comme un attribut dans une structure, puisque la possibilité de composition entraîne toute une combinatoire des fonctions lexicales. Nous les représenterons donc par la structure logique de base *function* prédéfinie dans le système SUBLIM, où l'on peut contraindre les classes acceptables pour les différents éléments de fonction (nom, arguments, valeur). Ainsi, la structure correspondant aux fonctions lexicales peut s'exprimer comme suit :

```
(def-linguistic-class lex-sem-fns
  (set-of (lex-sem-fn)))
(def-linguistic-class lex-sem-fn
  (function :label nom-FL
            :arguments FL-arg
            :value expression-linguistique))
```

Pour représenter la composition de fonctions, on peut autoriser l'utilisation d'une fonction lexicale en argument d'une fonction lexicale. Néanmoins, la valeur de la fonction intermédiaire (si elle existe) n'est pas nécessairement pertinente. Seule les noms des fonctions composées sont porteurs d'information. Aussi, le plus simple est d'autoriser une valeur complexe comme nom de la fonction. Nous définirons donc un label de fonction comme une liste (ordonnée) de noms de fonctions de base.

```
(def-linguistic-class nom-FL
  (list-of (nom-FL-base)))
```

Le nom d'une fonction de base est donné par un identificateur de la fonction (une chaîne de caractères) et par un éventuel numéro de l'actant sur lequel elle opère :

```
(def-linguistic-class nom-FL-base
  (feature-structure
    ((fonction string)
     (actant   integer))))
```

L'argument de la fonction est un lexème. Le fait d'indiquer cet argument est redondant puisque cette fonction est définie à l'intérieur de la structure du lexème (il sera néanmoins représenté afin de simplifier les mécanismes d'extraction d'information).

```
(def-linguistic-class FL-arg lexème)
```

L'expression linguistique valeur de la fonction est représentée sous forme d'arbre (comme nous l'avons indiqué plus haut). Les nœuds de cet arbre sont soit des lexèmes, soit des variables. Pour simplifier, nous noterons les variables comme des chaînes de caractères.

```
(def-linguistic-class expression-linguistique
  (tree :nodes (one-of (lexème string))))
```

La définition de la structure linguistique du DEC, même simplifiée, illustre parfaitement le besoin ressenti par les linguistes de pouvoir mélanger différentes structures logiques dans une seule et même structure linguistique. Le fait de proposer différentes structures logiques permet au linguiste de manipuler des concepts proches de ceux utilisés dans sa

théorie. Cela permet de simplifier le travail du linguiste en lui permettant de rester à un niveau d'abstraction très utile lorsqu'il souhaite implémenter une théorie complexe.

Conclusion et perspectives

L'étape de formalisation de la structure du dictionnaire explicatif et combinatoire est nécessaire pour son informatisation. On remarque que ce dictionnaire met en œuvre de nombreuses structures informatiques de base (arbres, listes, structures de traits, etc.) et nécessite donc un environnement permettant de gérer un tel ensemble de structures hétérogènes. Cette constatation nous conforte dans l'idée que les futurs systèmes de gestion de bases lexicales devront être à même de gérer de telles structures.

Le projet NADIA-DEC va permettre différentes études :

- Multilinguisation du DEC : le DEC est un ensemble de dictionnaires monolingues, mais il se base sur une théorie qui s'applique à l'ensemble des langues étudiées jusqu'alors. Il semble donc souhaitable de mettre en correspondance les différents lexèmes des différentes langues disponibles, afin d'étudier les problèmes éventuels d'une multilinguisation du DEC,
- Vérification automatique de cohérence : pour la sortie de chaque volume du DEC, les auteurs doivent vérifier de nombreuses contraintes de cohérence sur les différents articles. Certaines de ces contraintes (comme la vérification de la numérotation des lexèmes, de certaines contraintes de formatage…) peuvent être prises en charge par le gestionnaire de bases lexicales. Nous étudierons les contraintes à vérifier et un environnement permettant de les spécifier et de les vérifier.
- Récupération des informations existantes : l'ensemble des informations déjà publiées n'est disponible que sous forme de fichiers de traitement de texte. Nous souhaitons étudier les problèmes d'import de ces entrées existantes, en les généralisant afin de définir un environnement d'import de différentes sources existantes dans une base lexicale.

En résumé, le projet NADIA-DEC ne vise pas seulement le développement d'une version informatique du DEC, mais permet aussi, de mettre en pratique, sur un cas réel et exigeant les idées de généricité développées au GETA dans le cadre du système SUBLIM. Il permettra de plus de développer d'autres aspects de recherche dans le domaine de la lexicographie (multilinguisation du DEC) et dans le domaine de l'informatique linguistique (développement d'environnements de manipulation de données linguistiques, etc.).

Pour terminer, je tiens a remercier l'AUPELF-UREF pour son soutient a cette action de recherche partagée entre le GETA-CLIPS (institut IMAG, université Joseph Fourier - Grenoble I) et le GRESLET (département de linguistique, université de Montréal)

Bibliographie

Mel'č uk I. et al. (1988). *DEC : Dictionnaire explicatif et combinatoire du français contemporain, recherches lexico-sémantiques II*, Montréal(Quebec), Canada, Presses de l'université de Montréal.

Mel'č uk I. et al. (1984). *DEC : Dictionnaire explicatif et combinatoire du français contemporain, recherches lexico-sémantiques I*, Montréal(Quebec), Canada, Presses de l'université de Montréal.

Mel'č uk I. et al. (1992). *DEC : Dictionnaire explicatif et combinatoire du français contemporain, recherches lexico-sémantiques III*, Montréal(Quebec), Canada, Presses de l'université de Montréal.

Mel'č uk I., Clas A. et Polguere A. (1995). *Introduction a la lexicologie explicative et combinatoire*, AUPELF et Hachette (sous presse).

Sérasset G. (1994). *SUBLIM système universel de bases lexicales multilingues et NADIA : sa spécialisation aux bases lexicales interlingues par acceptions*, Thèse nouveau doctorat, Université Joseph Fourier-Grenoble 1 : 194 p.