



**HAL**  
open science

## Un éditeur pour le dictionnaire explicatif et combinatoire du français contemporain

Gilles Sérasset

► **To cite this version:**

Gilles Sérasset. Un éditeur pour le dictionnaire explicatif et combinatoire du français contemporain. Journées lexique du PRC-CHM, 1996, Grenoble, France. pp.131-138. hal-00966352

**HAL Id: hal-00966352**

**<https://hal.science/hal-00966352v1>**

Submitted on 26 Mar 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Un éditeur pour le dictionnaire explicatif et combinatoire du français contemporain

Gilles Sérasset

GETA-CLIPS-IMAG (UJF & CNRS)

BP 53

38041 Grenoble Cedex 9

Tél. : 04.76.51.43.80 - Fax : 04.76.51.44.05

Courriel : Gilles.Serasset@imag.fr

---

## Résumé

Dans le domaine de l'ingénierie linguistique et de la connaissance, le problème des ressources lexicales et linguistiques s'est toujours posé. Néanmoins, l'avancée des techniques du Traitement Automatique des Langues Naturelles (TALN) l'a rendu plus sensible. Il nous faut maintenant pouvoir répondre à des besoins importants en terme de quantité, de qualité et de complexité. La complexité et la diversité des informations requises augmente avec les exigences des outils de TALN ainsi qu'avec le développement de nouvelles applications (humaines ou machinales). Si la récupération (semi)automatique d'information lexicale est une piste, elle ne pourra remplacer la création manuelle de dictionnaires.

Nous nous sommes donc intéressé à la construction d'outils pour lexicographes et lexicologues. Afin d'avoir une bonne compréhension des problèmes qui se posent, nous avons décidé d'informatiser un dictionnaire complexe, contenant de nombreuses informations structurées, le dictionnaire explicatif et combinatoire du français contemporain (DEC). Le DEC étant un travail de lexicologie, il ne s'agit donc pas à proprement parler d'un dictionnaire, mais plutôt d'un ensemble d'entrées destinées à illustrer une théorie linguistique. Ce ne sont donc pas les données que l'on va informatiser, mais le processus de rédaction de ces données.

Nous décrivons dans cet article une première version de l'éditeur spécialisé du dictionnaire informatique et combinatoire du français contemporain DECID. Nous verrons l'ensemble de ses fonctionnalités actuelles et discuterons des travaux futurs dans le cadre plus général d'un poste de travail du lexicographe.

---

## 1. 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'čuk ([Mel'čuk, Arbatchesky-Jumarie, Dagenais & al. 1988, Mel'čuk, Arbatchesky-Jumarie, Eltnisky & al. 1984, Mel'čuk, Arbatchesky-Jumarie, Iordanskaja & al. 1992, Mel'čuk, Clas & Polguère 1995]). Ce projet est actuellement soutenu en tant qu'action de recherche concertée du réseau Lexicologie Terminologie et Traduction de l'AUPELF-UREF et, plus récemment, en tant que coopération franco-québécoise en ingénierie linguistique (Ministère des affaires étrangères).

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-CLIPS ([Sérasset 1994]).

Après avoir discuté des objectifs et moyens du projet NADIA-DEC, nous montrons dans cet article une première version de l'éditeur spécialisé du dictionnaire informatique et combinatoire

du français contemporain. Nous verrons l'ensemble de ses fonctionnalités actuelles et discuterons des travaux futurs dans le cadre plus général d'un poste de travail du lexicographe.

## 2. Aperçu du projet

### 2.1. Objectifs

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

1. création d'un éditeur informatique spécialisé pour le DEC,
2. réalisation d'un mécanisme d'import des données existantes, actuellement au format R.T.F. (Rich Text Format),
3. réalisation d'un module d'export vers différents formats utilisables informatiquement (SGML/TEI, format LISP...) et pour la publication papier (R.T.F., M.I.F....).
4. intégrer dictionnaire et éditeur à un système de gestion de données lexicales générique (SUBLIM). Les vérifications des différentes contraintes sur les données seront vérifiées à ce niveau,

Les points 1, 2 et 3 sont en cours de réalisation et sont assez avancés. Le point 4, à plus longue échéance, en est à ses débuts.

### 2.2. Originalités

D'autres projets ayant pour but l'informatisation du DEC sont en cours, notamment à l'université de Montréal ([Mel'čuk & al. 1995], chap. 4) par Alain Polguère. Ces projets utilisent des approches différentes :

- Pour Alain Polguère, l'informatisation du DEC est un moyen de disposer de données pour des traitements automatiques. Ainsi, seules les données complètement formalisables sont retenues lors de cette informatisation. De plus, certaines de ces données sont légèrement modifiées par rapport à la version imprimée et d'autres sont ajoutées.
- le projet NADIA/DEC vise l'informatisation du processus de production du DEC, il doit donc prendre en compte l'ensemble des informations présentes dans le DEC, et ce, quel que soit leur degré de formalisation.

Les deux approches ne sont pas antinomiques et nous travaillons actuellement à l'adaptation de l'éditeur DECID pour la génération (au moins partielles) d'entrées lexicales utilisables par Alain Polguère. Nous envisageons aussi d'augmenter les fonctionnalités de l'éditeur actuel, afin d'intégrer dans le processus de création d'entrées du DEC, la saisie de données propres au formalisme utilisé à l'université de Montréal.

### 3. L'éditeur spécialisé DECID

#### 3.1. Les éléments d'interface

L'interface a été conçue afin de simplifier au maximum le travail du rédacteur, tout en respectant les recommandations d'Apple pour l'interface du Macintosh. De plus, l'interface est à la fois une interface de navigation et une interface d'édition.

##### Les menus



Les dictionnaires peuvent être gérés à la discrétion de l'utilisateur. Il est possible d'utiliser un seul fichier dictionnaire, comme d'utiliser un fichier dictionnaire par champs lexical ou par vocable.

Afin de permettre à l'utilisateur de gérer un fichier contenant des notes personnelles, des résultats de recherches dans des bases terminologiques, des résultats de recherche dans des corpus ou des fichiers provenant d'autres éditeurs (notamment comme l'éditeur Édico, développé par Alain Polguère), nous avons rajouté à l'éditeur DECID des fonctionnalités d'édition générales de fichiers textes. L'utilisateur peut donc créer, ouvrir et sauvegarder des fichiers textes.

L'utilisateur peut choisir le format de sauvegarde du dictionnaire, parmi Decid (format standard), Édico (non implémenté actuellement) ou R.T.F. (Rich Text Format) pour obtenir une version imprimable du dictionnaire.

Comme toutes les applications sur Macintosh, Decid comprend un menu *édition* où l'on trouve les éléments standards de l'interface Macintosh, *annuler, couper, copier, coller, effacer* et *tout sélectionner*. Ces fonctionnalités sont implémentées sur les différents objets que l'on est amené à manipuler avec DECID.



Le menu création permet au lexicographe de créer de nouveaux éléments

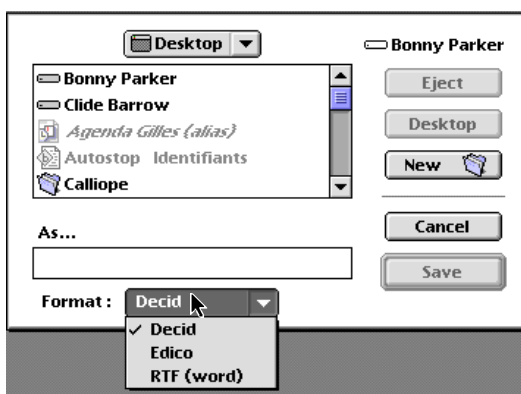
d'information dans un dictionnaire ou dans une lexie. Lorsque la fenêtre courante est une fenêtre de dictionnaire, le lexicographe peut créer de nouvelles lexies. Lorsque la fenêtre courante est une fenêtre de lexie, le lexicographe peut créer une nouvelle fonction lexicale pour cette lexie ou une nouvelle valeur pour la fonction lexicale sélectionnée.

Les menus *police* et *style* permettent de changer l'apparence d'un élément d'information des lexies. Il permet aussi de changer la police d'un élément de texte dans un fichier texte.

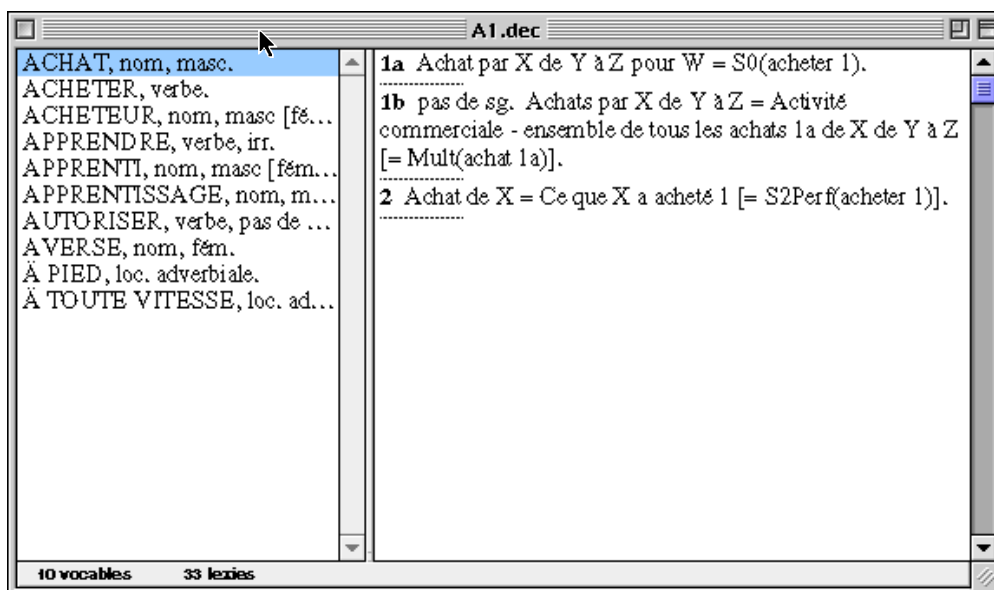
Cette possibilité permet l'édition d'entrée du DEC dans des langues non basées sur le script roman. Tous les éléments de DECID sont conçus pour rendre possible l'édition du DEC pour des langues comme le russe ou le japonais.

##### Fenêtre de dictionnaire

Dès que l'on crée ou que l'on ouvre un dictionnaire, la fenêtre principale du dictionnaire apparaît. Elle se décompose en deux parties. La première (liste de gauche) contient l'ensemble



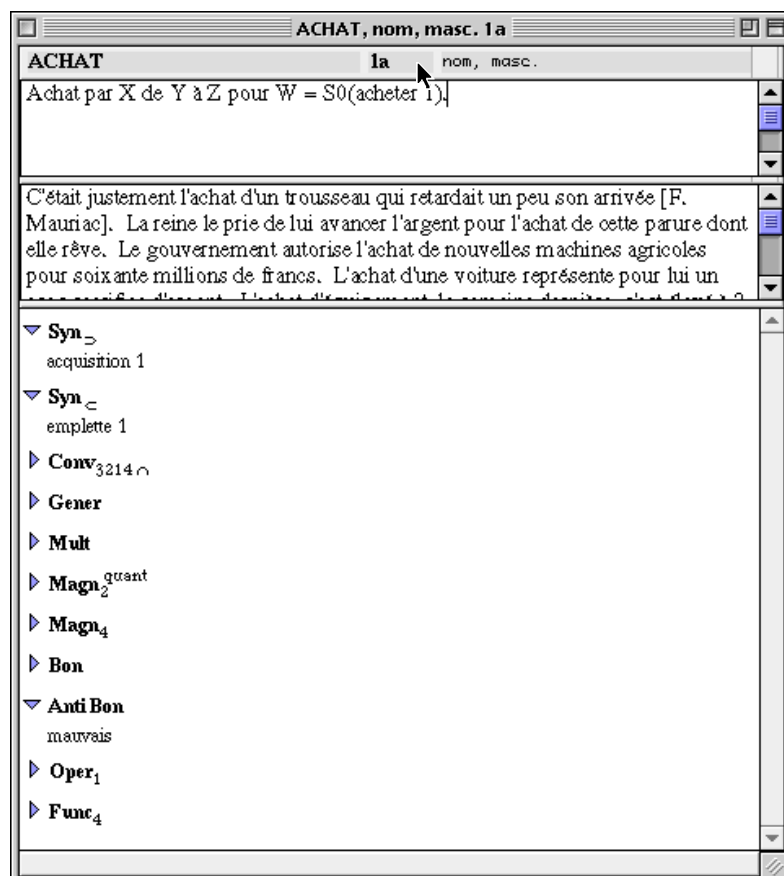
des vocables définis dans ce dictionnaire. Lorsque l'on sélectionne un (ou plusieurs) vocables dans cette liste, les lexies correspondantes apparaissent dans la liste de droite.



Ces lexies apparaissent sous forme d'un numéro de sens (lorsqu'il y en a) suivi d'un résumé. C'est ce résumé qui est utilisé dans la version papier comme tableau synoptique d'un vocable.

Comme ces résumés ne sont pas obligatoirement identiques à la définition de la lexie, il est possible dans cette fenêtre de les modifier. Pour cela on passe en mode édition en appuyant sur la touche Entrée du pavé numérique. On en sort en validant les modifications par un nouvel appui sur la même touche. Par défaut, la définition est recopiée comme résumé.

En double-cliquant sur un résumé, on ouvre la fenêtre de la lexie correspondante.



### Fenêtre de lexie

La fenêtre de lexie comporte quatre parties principales. De haut en bas, on remarque les informations graphiques et morphologiques, puis la définition, les exemples et enfin la liste des fonctions lexicales.

Dans cette fenêtre, chaque élément est éditable. On peut ainsi à tout moment changer la graphie, la définition, le numéro de lexie ou toute autre information.

### 3.2. Éditer une lexie

#### Informations graphiques et morphologiques

Cette information est décomposée en trois éléments d'interface distincts.

Le premier est un texte éditable contenant la graphie. Si l'on souhaite distinguer deux vocables ayant une même graphie et une même morphologie, on rajoutera simplement un numéro à la graphie. Lors de l'export en word, ce numéro apparaîtra de manière adéquate (cette manière peut dépendre du style que l'on choisira pour la forme papier). Ainsi, si l'on créé une graphie **bleu1**, le vocable BLEU<sup>1</sup> sera créé à l'export.

Le second est un texte éditable contenant le numéro d'homonyme. Lors de l'édition, il n'y a pas de vérification de la validité de ce numéro.

Le troisième est un texte éditable contenant les informations morphologiques sur la lexie.

Les vocables n'ont pas d'existence propres dans l'éditeur DECID, ils sont créés à la volée, suivant l'ensemble des lexies présentes dans la base. Un vocable est identifié par sa graphie et sa morphologie. Ainsi, deux lexies ayant une même graphie et une même morphologie seront réunies automatiquement dans un même vocable. Si l'on souhaite séparer dans deux vocables différents des lexies ayant une même graphie et une même morphologie, on doit ajouter des numéros distincts à chacune des graphies. Si deux lexies ont une même graphie et des informations morphologiques légèrement différentes, mais que l'on souhaite les réunir dans un même vocable, on le peut, en mettant en italiques les éléments de l'information morphologique qui ne devront pas être pris en compte pour discriminer le vocable.

Par exemple, le vocable BLEU<sup>1</sup> (DEC III, p. 159), comporte des lexies adjectivale et des lexies nominale. Comme aucune information morphologique n'est utilisée pour les distinguer, on mettra l'ensemble des informations morphologiques en italique. De la même manière, l'entrée FUREUR, nom fém. comprend des lexies où l'on précise pas de pl. (Fureur 1a., DEC III, p. 217) ou pas de sg. (Fureur 1b., DEC III, p218). Ces compléments d'information morphologique seront indiqués en italique pour ne pas intervenir lors de la détermination du vocable.

Lorsqu'une partie d'information morphologique est indiquée sans italique, elle sera (par définition) commune à l'ensemble des lexies d'un vocable.

### *Définition*

La définition est contenue dans un champs de texte éditable. Elle se modifie de manière classique.

### *Exemples*

Les exemples sont contenus dans un champs de texte éditable. Ils se modifient de manière classique.

### *Fonctions lexicales*

Les fonctions lexicales sont présentées sous forme de liste. À l'ouverture de la fenêtre de lexie, on ne voit que les fonctions lexicales, mais pas leurs valeurs. On peut obtenir leurs valeurs en cliquant sur le petit triangle situé à coté de leur nom. L'interface utilisée est identique à celle utilisée dans les vues en liste des fichiers et dossiers sous le finder du Macintosh (Système 7.0 et ultérieurs).

Il est possible d'éditer les valeurs. Pour cela, il suffit de sélectionner la valeur à éditer, puis d'appuyer sur la touche *entrée* du pavé numérique. L'édition se fait ensuite de manière classique. On termine et valide l'édition en appuyant une nouvelle fois sur la touche *entrée* du pavé numérique.

Les noms de fonctions lexicales sont eux aussi éditables. Comme une fonction lexicale est un objet complexe et non une simple chaîne de caractères, l'éditeur utilise une méthode de saisie (voir les détails plus bas) qui simplifie grandement le travail du rédacteur. Pour entrer dans le mode d'édition, il suffit de sélectionner la fonction à éditer, puis d'appuyer sur la touche *entrée* du pavé numérique (un curseur clignotant apparaît alors). L'édition se fait ensuite grâce à la méthode de saisie. On termine et valide l'édition en appuyant une nouvelle fois sur la touche *entrée* du pavé numérique.

### *Méthode de saisie des fonctions lexicales*

L'édition des noms de fonctions lexicales est un travail ennuyeux sous un texteur. En effet, ces noms de fonctions apparaissent avec des indices et des exposants numériques ou alphabétiques. De plus, ces fonctions doivent être triées selon un ordre précis.

L'éditeur simplifie la création et la gestion des fonctions lexicales en prenant en charge la gestion des attributs typographiques pouvant servir dans un nom de fonction et en s'occupant du tri.

Pour cela, l'éditeur gère un objet logique représentant un nom de fonction lexicale. Il sait l'afficher, le sauver, le restaurer, l'exporter au format word...

Pour éditer une fonction lexicale, nous aurions pu, en utilisant des éléments d'interface standard, créer un formulaire de saisie d'une fonction lexicale. Mais ce type d'interface par formulaire n'est pas adaptée à nos objectifs, et sera rejetée par l'utilisateur. Nous avons donc préféré une interface plus WYSIWYG (what you see is what you get), et plus habituelle pour le rédacteur.

Decid aide le rédacteur en lui fournissant une méthode de saisie qui permet de créer des fonctions lexicales en un minimum de manipulation. De plus, le rédacteur créé logiquement une fonction lexicale, il n'a pas à se soucier de la manière dont les informations sont affichées.

Ainsi, la création de la fonction lexicale **Instr** se fait simplement en appuyant successivement sur les touches **i**, **n**, **s**, **t** et **r**, sans même se soucier des majuscules ou des minuscules.

Lorsqu'une fonction lexicale a des numéro d'actants, comme **Real<sub>1</sub>**, il suffit de taper l'actant après la fonction lexicale en question. Ainsi, le lexicographe se contente de taper **r**, **e**, **a**, **l** et **1** pour obtenir la fonction lexicale **Real<sub>1</sub>**. Le rédacteur peut aussi rajouter des actants à une fonction existant déjà. Dans ce cas, il positionne son curseur à la suite de la fonction et tape les actants désirés.

Pour obtenir une fonction ayant un glissement de sens, comme **Syn<sub>∩</sub>**, le rédacteur se contente d'appuyer sur la touche **+** pour obtenir un sens plus large, **-** pour obtenir un sens plus étroit et **\*** pour obtenir une intersection de sens, pour supprimer un glissement de sens, il appuiera sur la touche **/**. Le rédacteur peut aussi modifier le glissement de sens d'une fonction existant déjà. Dans ce cas, il positionne son curseur à la suite de la fonction et tape sur la touche désirée.

Lorsqu'une fonction lexicale a un sémantisme, comme **Magn<sup>quant</sup>**, le rédacteur le spécifie en tapant le sémantisme, précédé du caractère **'**. Ainsi, pour obtenir la fonction précédente, le rédacteur appuie sur **m**, **a**, **g**, **n**, **'**, **q**, **u**, **a**, **n** et **t**. Le rédacteur peut aussi modifier le sémantisme d'une fonction existant déjà. Dans ce cas, il positionne son curseur à la suite de la fonction et tape le caractère **'** suivi du sémantisme souhaité.

Lorsqu'une fonction lexicale a une facette, comme **Magn<sub>[communiquer]</sub>**, le rédacteur le spécifie en tapant la facette entre crochets. Ainsi, il obtient la fonction précédente en tapant : **m**, **a**, **g**, **n**, **[**, **c**, **o**, **m**, **m**, **u**, **n**, **i**, **q**, **u**, **e**, **r**, **]**. Le rédacteur peut aussi ajouter ou modifier la facette d'une fonction existant déjà. Dans ce cas, il positionne son curseur à la suite de la fonction et tape la facette souhaitée entre crochets.

Chacune de ces information peut être ajoutée à la même fonction lexicale, sans contrainte sur l'ordre dans lequel on ajoute ces informations.

- ▶  $S_{instr}$
- ▶  $S_{loc}$
- ▶  $Oper_{2}^{II}$  ["garder"]
- ▶  $Oper$
- ▶  $Caus Oper$
- ▶  $A_1 Caus Oper$
- ▶  $Caus_1 Oper$
- ▶  $Oper$  ["garder"]

La composition de fonction se fait simplement en tapant les fonction les unes à la suite des autres. On peut ainsi très facilement créer les fonctions lexicales reprises dans la figure ci-contre.

#### *Limitations actuelles*

Actuellement, l'interface de création des fonctions lexicales ne permet que la création de fonctions standards simples ou composées. Elle ne permet pas encore la création de fonctions lexicales non-standard ou complexes. Par contre, le noyau de représentation d'une fonction lexicale, ainsi que le module d'affichage sont suffisamment avancés pour gérer ce type de fonction. Ainsi, il est possible de récupérer et de visualiser des fonctions lexicales non-standard ou complexes à partir des

fichier word du dictionnaire publié.

- ▶  $Syn_{\cap}$
- ▶  $Caus Conv_{21}$
- ▶  $S_{0<}$
- ▶  $S_0$  (A. le métier ou la profession Y de Z en travaillant sous la direction de Z)
- ▶  $S_{1<}$
- ▶  $S_1$  (A. par soi-même)
- ▶  $S_{2<}$
- ▶  $S_3$
- ▶  $Magn$
- ▶  $S_1$  (Result +  $Magn_2$ )
- ▶  $Ver$
- ▶  $Anti Ver$
- ▶  $Result$
- ▶ par manque de pratique de Y,  $Fin Result$

Fonctions lexicales récupérées automatiquement

## 4. Discussions et perspectives

L'éditeur présenté ci-dessus, n'est qu'une première étape vers un outil véritablement utilisables par les lexicographes souhaitant éditer un dictionnaire explicatif et combinatoire. Pour cet objectif, il nous faudra notamment le doter d'un système de vérification de contraintes de cohérence et l'intégrer à des outils plus généraux utilisés par tout lexicographe (recherches de collocations dans un corpus, outils d'acquisition d'information...).

Il nous a néanmoins permis de mieux comprendre les besoins des lexicographes en terme d'outils. Certes, l'intégration à des outils plus généraux est importante. Mais de très nombreuses fonctionnalités plus simples, que nous n'envisagions pas a priori, sont tout aussi importante. On nous a ainsi demandé des filtres permettant d'extraire différentes informations, telle qu'une liste de toutes les abréviations utilisées dans le dictionnaire (afin de détecter des incohérences dans les notations de différents lexicographes).

Les requêtes étant très nombreuses, il nous apparaît qu'un outil pour lexicographe doit fournir un langage de description de filtres permettant vérifications et extractions de données



diverses. Un éditeur tel que DECID ne doit donc pas trop cacher la structure interne du dictionnaire, afin d'être suffisamment adaptable à différents usages. Les versions ultérieures contiendront donc un langage de filtrage permettant au linguiste de spécifier des extractions spéciales et des règles de cohérence.

Une autre difficulté provient de la nature même du dictionnaire explicatif et combinatoire qui n'est pas un travail de lexicographie, mais un travail de lexicologie. La structure des entrées est donc en permanente évolution. On peut ainsi observer des différences dans les informations des quatre volumes actuellement publiés. Cette constante évolution de la structure logique (qui apparaît dans une moindre mesure dans d'autres projets de dictionnaires) rend impossible la maintenance d'un outil ad-hoc.

Nous souhaitons donc maintenant orienter nos développements vers l'intégration des aspects d'interface dans le cadre d'un outil générique de gestion de bases de données lexicales multilingues, tel que nous l'avons défini dans [Sérasset 1994]. Une telle approche nous permettra de construire une plate-forme générique où l'on pourra paramétrer à la fois les structures de données utilisées mais aussi les différentes "vues" (interfaces) de ces structures.

L'intégration d'autres outils externes se fera par la définition d'interfaces de programmation. Nous explorerons aussi les possibilités d'utilisation d'architectures comme OpenDoc™, qui permet l'intégration simple de composants divers travaillant en collaboration. Cet aspect fera l'objet de projets ultérieurs.

## 6. Bibliographie

Mel'čuk I., Arbatchesky-Jumarie N., Elnitsky L., Iordanskaja L. & Lessard A. (1984) *DEC : Dictionnaire explicatif et combinatoire du français contemporain, recherches lexicosémantiques I*. Presses de l'université de Montréal, Montréal (Québec), Canada, 172 p.

Mel'čuk I., Arbatchesky-Jumarie N., Dagenais L., Elnitsky L., Iordanskaja L., Lefebvre M.-N. & al. (1988) *DEC : Dictionnaire explicatif et combinatoire du français contemporain, recherches lexicosémantiques II*. Presses de l'université de Montréal, Montréal (Québec), Canada, 332 p.

Mel'čuk I., Arbatchesky-Jumarie N., Iordanskaja L. & Mantha S. (1992) *DEC : Dictionnaire explicatif et combinatoire du français contemporain, recherches lexicosémantiques III*. Presses de l'université de Montréal, Montréal (Québec), Canada, 323 p.

Mel'čuk I., Clas A. & Polguere A. (1995) *Introduction à la lexicologie explicative et combinatoire*. Universités francophones et champs linguistiques, AUPELF-UREF et Duculot, Louvain-la-Neuve, 256 p.

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