

Université Paris 7-Denis Diderot

Serge Fleury

POLAS FRITAS

**Prototypes Oriented Language
Has Freed Us**

*La Programmation à Prototypes, un outil pour une
linguistique expérimentale : mise en oeuvre de
représentations évolutives des connaissances pour
le traitement automatique du langage naturel*

Thèse

présentée pour l'obtention
du titre de Docteur d'Université
Spécialité : Linguistique

Soutenue devant la Commission d'Examen
composée de

MM.	Bernard Victorri	(Rapporteur)
	Pierre Cadiot	(Rapporteur)
	François-Xavier Testard-Vaillant	(Directeur)
	Benoît Habert	
	Marcel Cori	
	Violaine Prince	

Introduction

- **Mise en œuvre d'un dispositif expérimental de TALN**
- **Un cadre de représentation particulier : la PàP**
- **Définir des représentations évolutives pour modéliser les faits de langue**
- **Un méta-niveau d'analyse : le retour du linguiste**

Origine et choix

○ **Nécessité de représenter la mouvance ?**

- La langue évolue en permanence (Biber 1993)
- La description linguistique doit être ajustée pour tenir compte de nouvelles informations

○ **Pourquoi associer des savoirs aux mots?**

- Il est illusoire de traiter globalement les problèmes de la langue
- Etude au niveau des mots des régularités et des redondances

Quels savoirs pour les mots ?

○ Peut-on prévoir les fonctionnements lexicaux ?

- Un examen des comportements des mots révèle des variations qu'il semble difficile de fixer dans des structures de représentation statiques

En médecine coronarienne, longue marque le degré dans la séquence nominale "une occlusion longue"

Dans d'autres sous-langages, cet adjectif n'est pas typé pour marquer le degré

○ Imprévisibilité des comportements lexicaux

artère et infarctus entrent dans des relations de localisation : deux réalisations distinctes

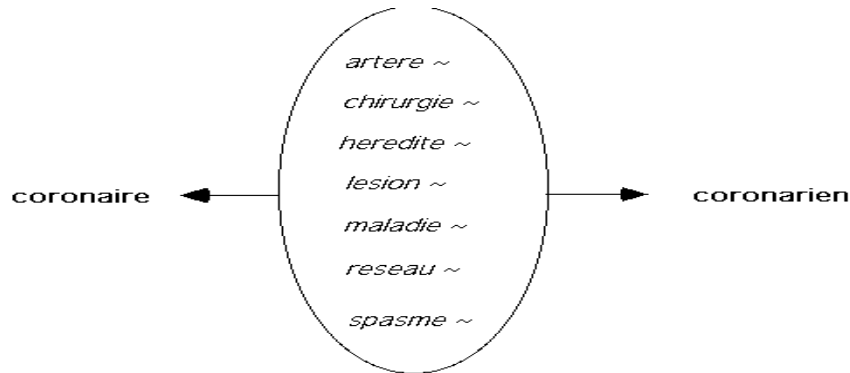
artère {coronaire circonflexe diagonal...}

infarctus {anterieur inferieur apical }

○ Imprévisibilité des contraintes sur les combinaisons d'arbres

- Proximités de contextes entre adjectifs...

coronaire et coronarien partagent des contextes

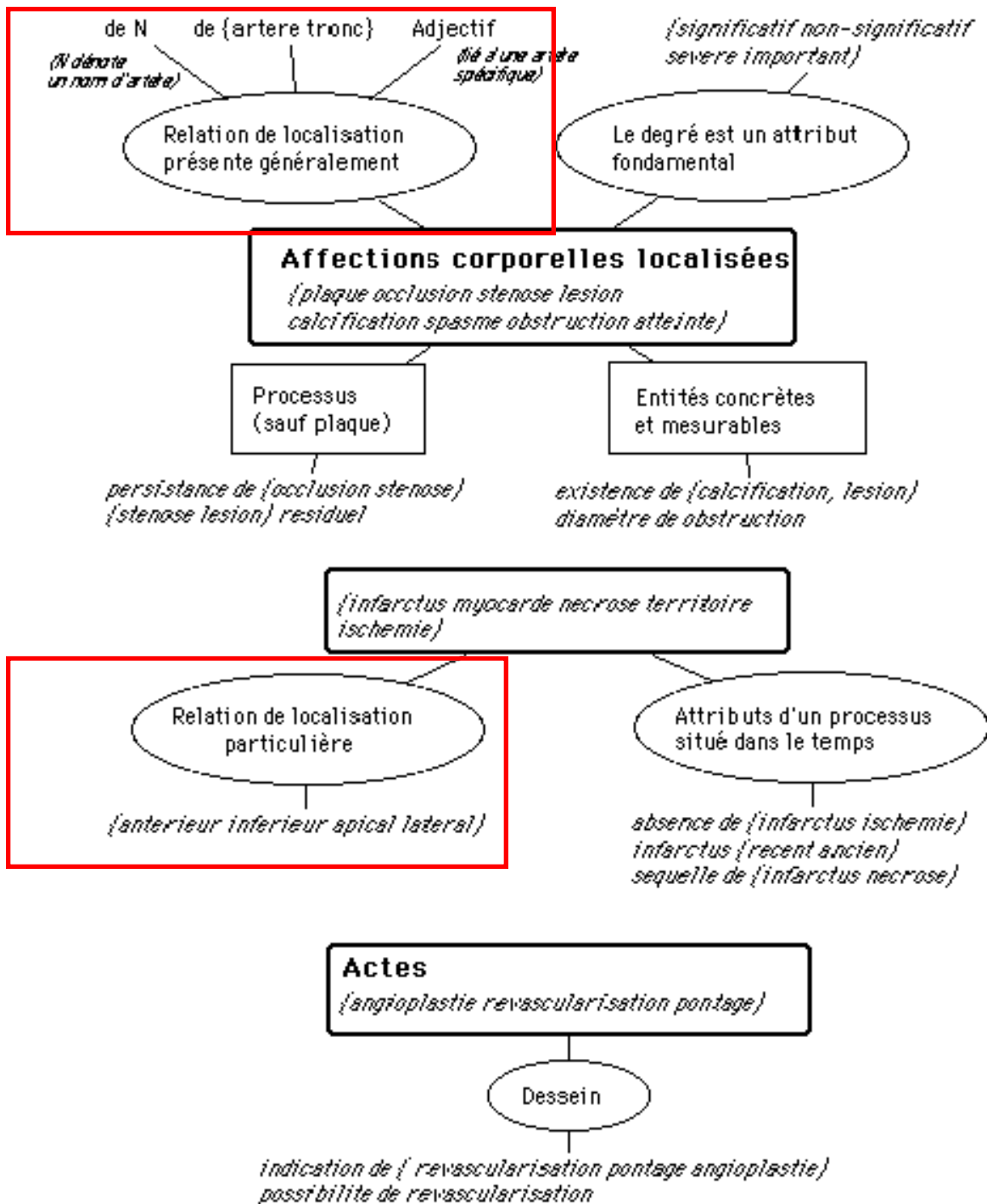


- Mais combinaisons divergentes

coronarien est associé à des adjectifs évaluatifs {severe, significatif, important}

coronaire ne l'est pas

- Diversité des combinaisons d'arbres réalisées sur une même (sous) famille de mots
- La relation de localisation est soit généralement présente, soit présente sous une forme particulière, soit non présente



Un Problème

○ (1) Représenter les mots et leurs comportements

- Les savoirs associés aux mots peuvent bouger et remettre en cause des représentations construites à un moment donné
- Les savoirs généraux que l'on peut associer aux mots ne sont pas toujours pertinents (Biber 1993)
- Il n'est donc pas satisfaisant de se contenter d'un modèle apriorique pour construire une représentation des comportements des unités lexicales

○ (2) Classer les mots suivant leurs comportements

- Ne pas prédéterminer de manière figée ni les structures définies pour cette représentation ni leurs classements

"La construction d'une hiérarchie est un processus incrémental" et une hiérarchie "évolue et s'améliore en fonction des résultats obtenus jusqu'à ce qu'une certaine forme de stabilité soit atteinte" (Haton & al. 1991)

Un Corpus

- **Hypothèse retenue : il y a peu de sens à vouloir faire de l'acquisition sémantique en dehors d'un sous-langage**
- **Comment attacher des connaissances aux unités lexicales ?**
 - Acquisition de connaissances par apprentissage
 - Par affinements successifs
- **Solution retenue : extraction de savoirs à partir de corpus**
 - Repérer les arbres élémentaires de dépendance entre mots (relations opérateur-opérandes) et les contraintes sur les combinaisons de ces arbres
 - Corpus MENELAS (Zweigenbaum & al. 1995)
 - Unité thématique : les maladies coronariennes

Un Outil Informatique

- **Les représentations des mots et de leurs comportements doivent pouvoir évoluer**
 - Les outils de représentation doivent permettre que les représentations des mots et leurs classements évoluent

- **Quel outil pour représenter la mouvance ?**
 - La PàP

- **La PàP : Programmation à Prototypes**
 - Construction progressive d'entités informatiques suivant les connaissances dont on dispose
 - Si de nouvelles connaissances sont mises au jour, on affine le processus de représentation déjà amorcé sans avoir à reconstruire entièrement de nouvelles structures

Plan

○ **Partie 1 : Acquisition de savoirs en corpus**

- Extraction des fonctionnements lexicaux : Lexter, Zellig
- Limites de l'extraction

○ **Partie 2 : Cadre de représentation**

- PàP : Programmation A Prototypes
- Le langage Self

○ **Partie 3 : Représentation et classement des mots**

- Construction inductive des prototypes de mots et d'arbres
- Classements des prototypes

○ **Partie 4 : Résultats construits**

○ **Conclusion et perspectives**

Extraction des fonctionnements lexicaux

○ Point de départ : le corpus MENELAS

- 70000 mots sur les maladies coronariennes

○ Des outils d'extraction

- AlethIPGN : un extracteur de SN (développé par GSI-ERLI dans le cadre du projet Eureka GRAAL)
- Lexter : un extracteur de candidat terme (Direction des Etudes et Recherches EDF)
- Zellig : une chaîne de recyclage (Habert & al. 1996)

○ A partir d'arbres d'analyse

1. Simplifier les arbres d'analyse
2. Extraire les arbres "élémentaires" ou minimaux
3. Caractériser les combinaisons de ces arbres

○ Travaux récents en acquisition automatique de classes de mots

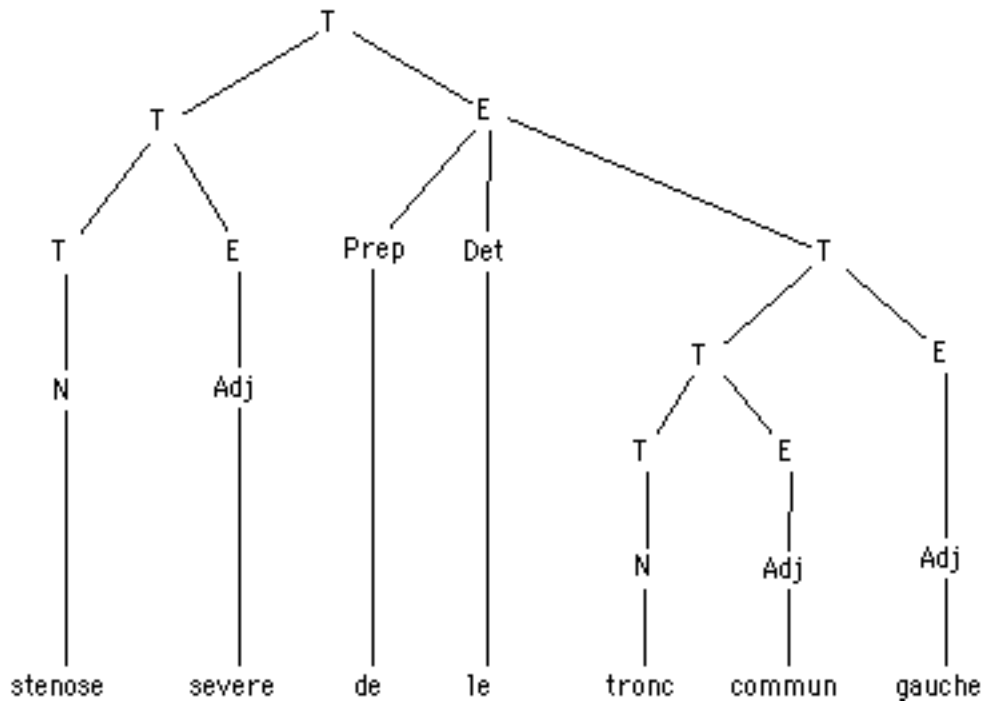
- (Church & Hank 1990), (Hindle 1990), (Smadja 1993), (Resnik 1993), (Grefenstette 1994), (Grishman & Sterling 1994)

LEXTER (Bourigault 93)

- **Outil d'acquisition terminologique**
- **En entrée : textes techniques arbitrairement longs**
- **Recherche de groupes terminologiques maximaux**
- **Structuration**

○ Une séquence nominale = une Tête (T) + une Expansion (E) (récurivement)

[T [T stenose] [E severe]][E de le [T tronc][E commun]][E gauche]



○ Approche endogène

- Deux découpages pour une séquence

(1) [angine de poitrine] instable

(2) angine de [poitrine instable]

- Recherche dans le corpus de situations non ambiguës

angine de poitrine existe

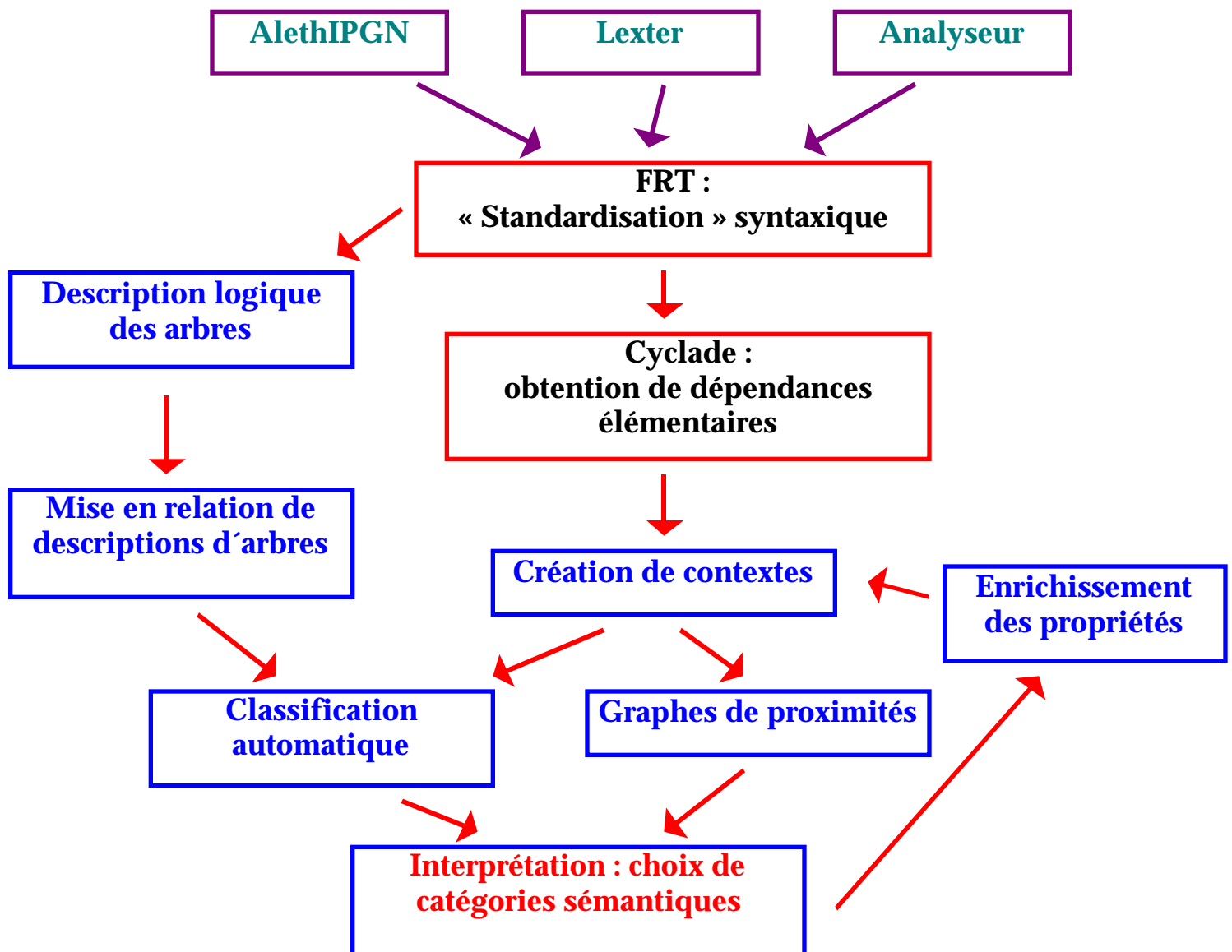
poitrine instable n'existe pas

- Sélection de (1)

ZELLIG (Habert & al. 95)

o But

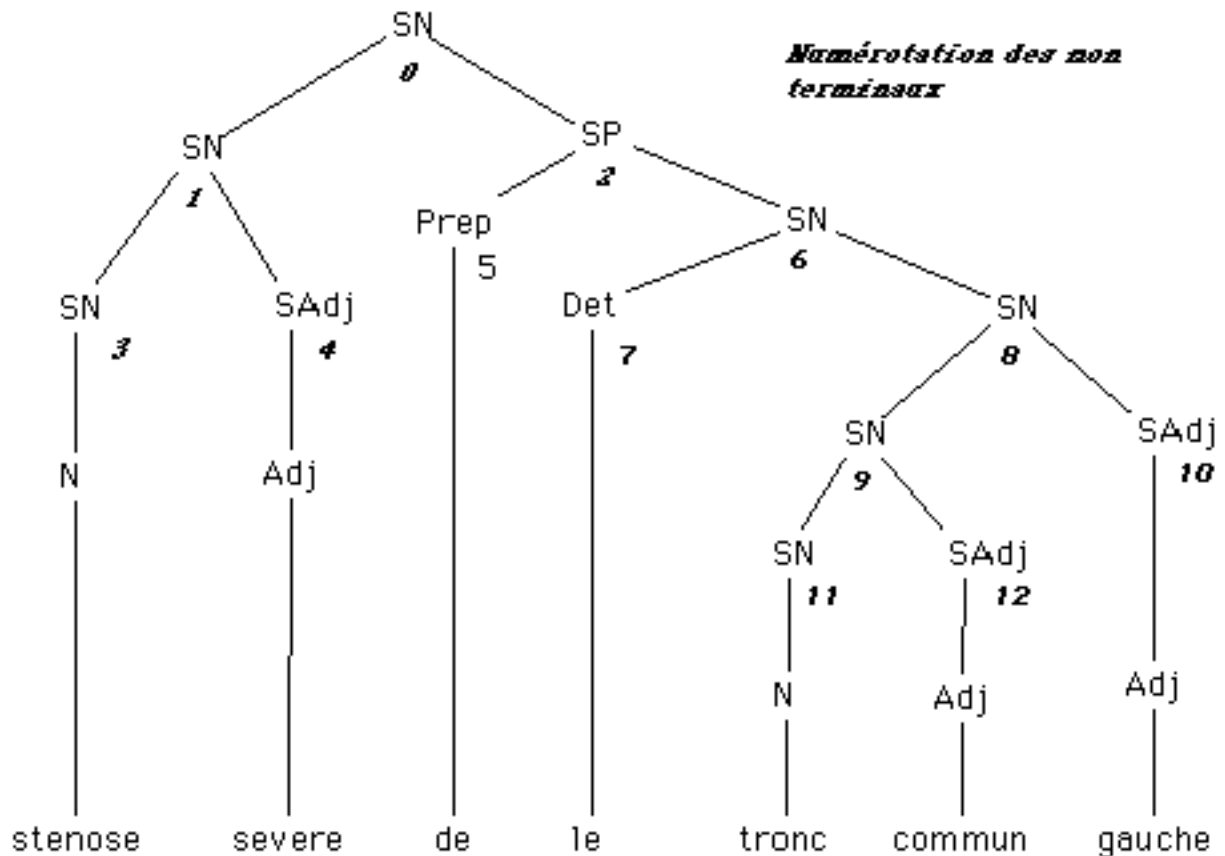
- Simplifier les arbres d'analyses fournis par un analyseur, ici LEXTER
- Mettre en évidence les relations élémentaires de dépendance entre mots pleins



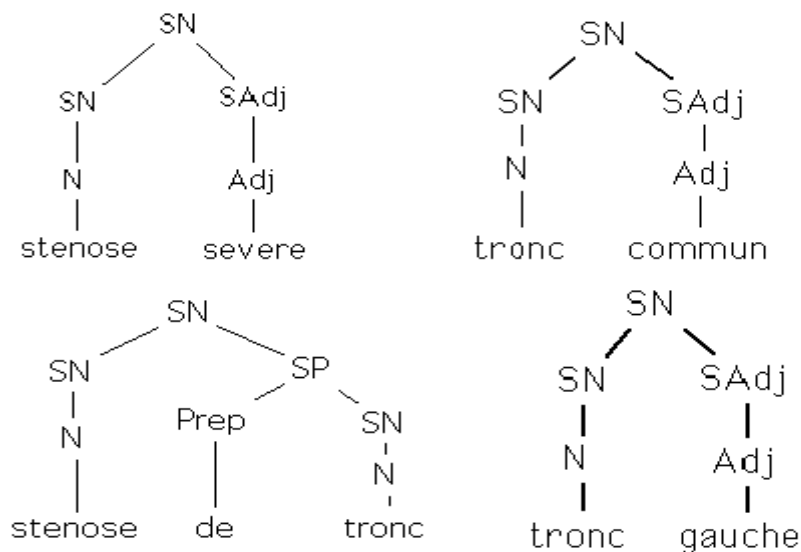
○ Simplification d'arbres d'analyse

- (1) Transformation en un arbre syntagmatique via le transducteur FRT (Habert & al. 1996)

=> Normalisation en arbre binaire Tête/(Modifieur ou Argument)



- (2) Le programme Cyclade (Habert & Nazarenko 96) est ensuite chargé de déterminer les arbres élémentaires via un filtrage de quasi-arbres (Habert & Folch 96)
- La déconstruction récursive met à jour des arbres élémentaires qui ne sont pas forcément directement présents dans l'arbre de départ
- Les arbres élémentaires mis au jour sont les suivants :



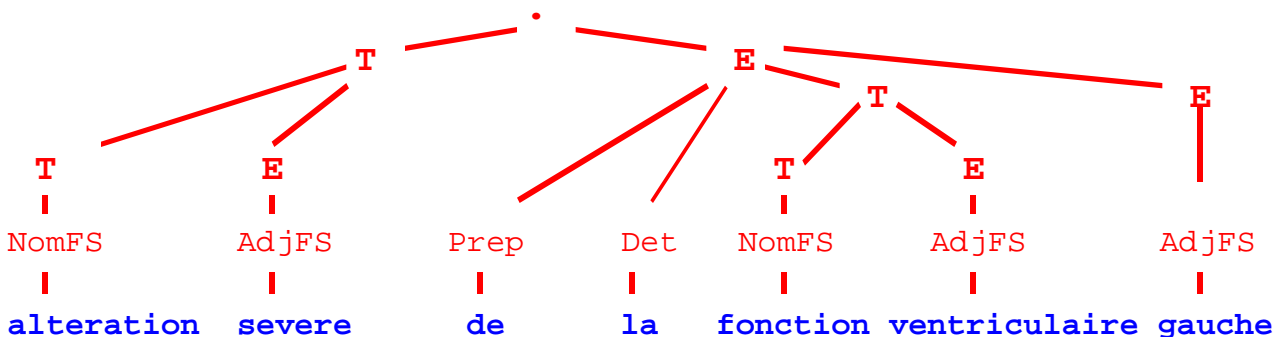
"Les dépendances élémentaires ainsi définies n'ont pas forcément de réalisation effective dans le corpus mais ils correspondent à des relations de dépendance vérifiées dans les arbres d'analyse, si l'on passe par une représentation logique de ces arbres et de ces dépendances élémentaires" (Habert & al. 1996)

Du corpus aux arbres minimaux

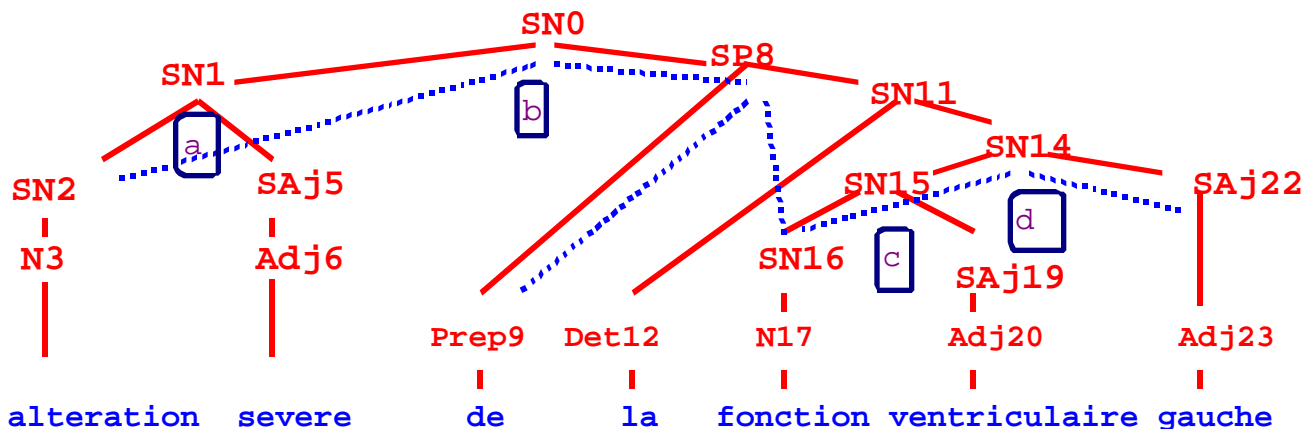
○ Extrait du corpus

Patient âgé de 70 ans, diabétique, qui a présenté il y a un an une douleur thoracique nocturne probablement en rapport avec un infarctus antéro-septal. Il est toujours symptomatique sous la forme d'un angor d'effort qu'il a lui-même négligé, avec semble-t-il plusieurs épisodes de préchordialgies de repos. La coronographie met en évidence des lésions bitronculaires. L'occlusion de l'IVA est responsable d'une hypokinésie antérieure. Une sténose serrée, diagonale et circonflexe est responsable de l'angor d'effort.

○ Arbre de Lexter



○ Après transformation dans Zellig



"Limites" de l'extraction

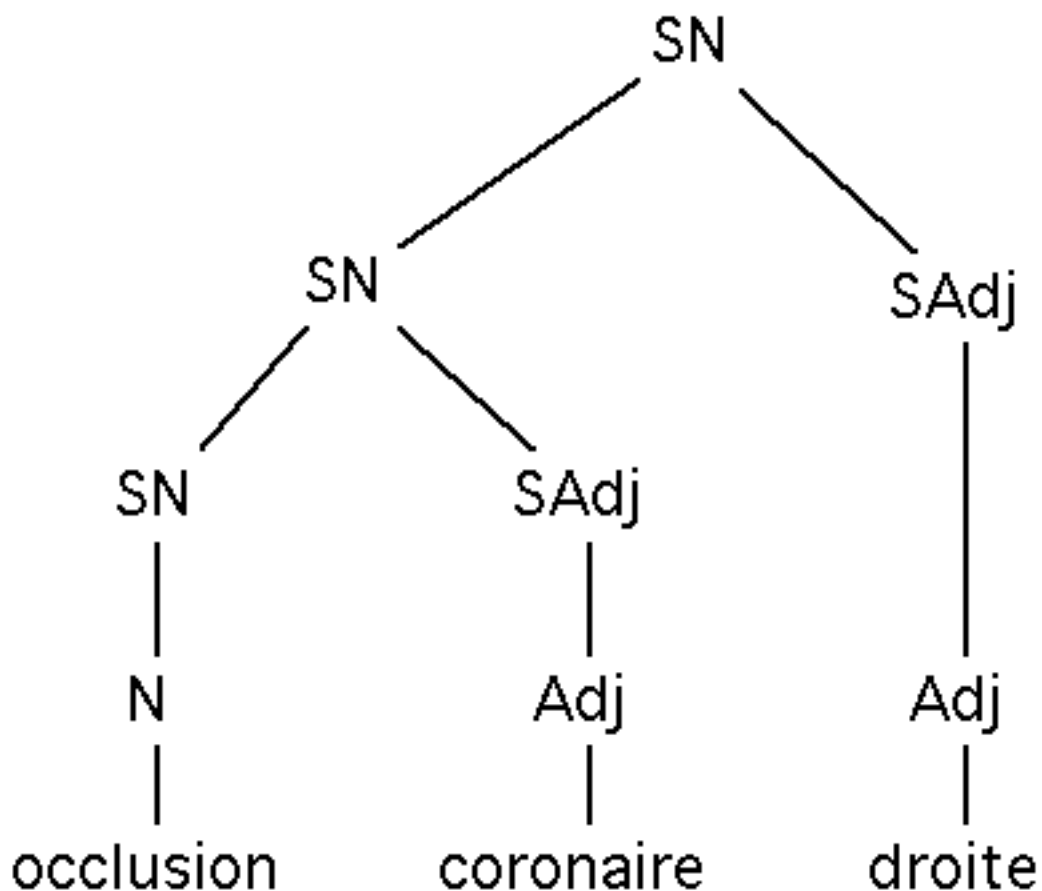
○ Les arbres d'analyse produisent des arbres "élémentaires" non pertinents

- On a les séquences :

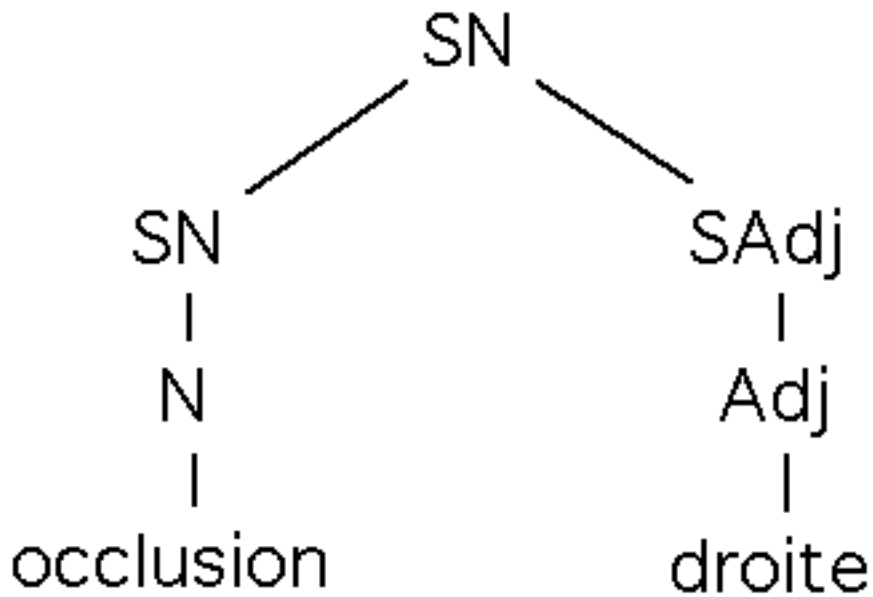
N coronaire droite et N coronaire

(où N n'est pas un nom d'artère)

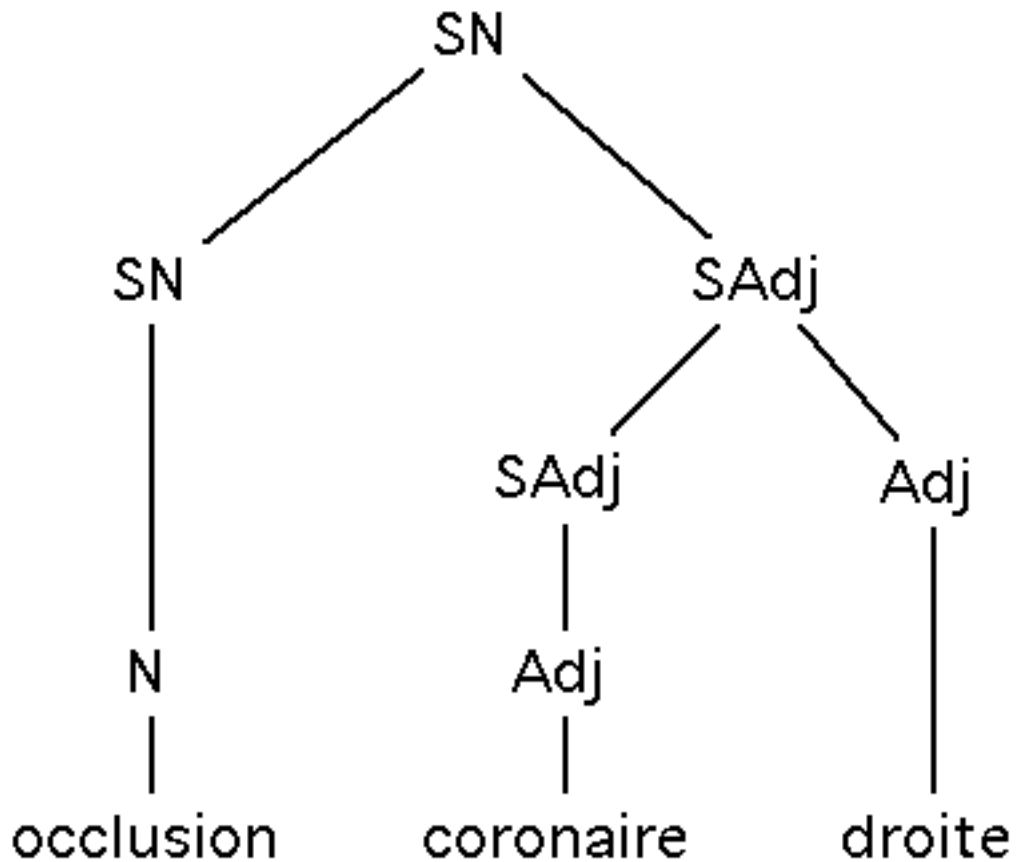
- Analyse proposée par LEXTER



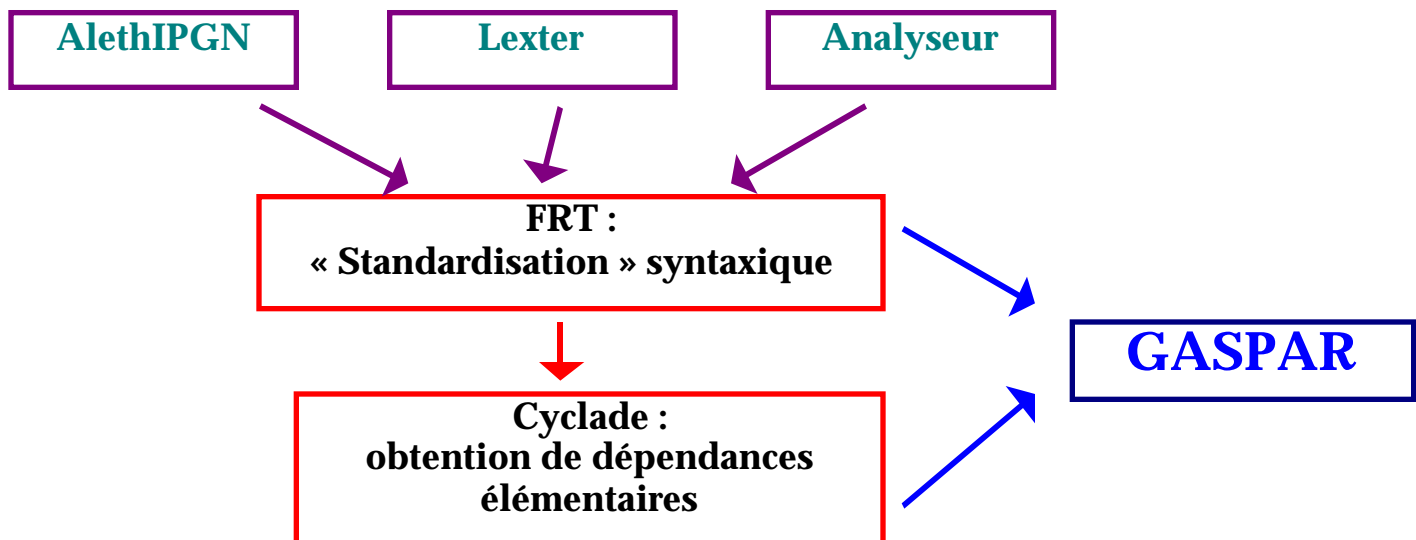
- Arbre "élémentaire" construit par CYCLADE



- En fait `droite` modifie l'adjectif dénotant une artère



Du corpus jaillit un réseau de prototypes



- On dispose en amont d'arbres associés à des mots

LEXTER => FRT => Arbres standardisés
=> CYCLADE => Arbres minimaux

- Savoirs éclatés, peu de généralisation
- Ces savoirs ne sont pas figés : ils pourront évoluer
 - Relatifs à un état particulier du travail d'extraction sur corpus

Définitions préliminaires

- On appelle **arbre minimal** ou **élémentaire** un arbre associé à une entrée lexicale traduisant une relation binaire de dépendance entre mots pleins
- On appelle **arbre d'analyse** un arbre associé à un arbre élémentaire : cet arbre d'analyse est obtenu par des opérations effectuées à partir de l'arbre élémentaire (adjonction, substitution...)
- On appelle **prototype de mot** l'objet informatique défini pour représenter un mot
- On appelle **prototype d'arbre élémentaire** l'objet informatique défini pour représenter un arbre élémentaire
- On appelle **prototype d'arbre d'analyse** l'objet informatique défini pour représenter un arbre d'analyse

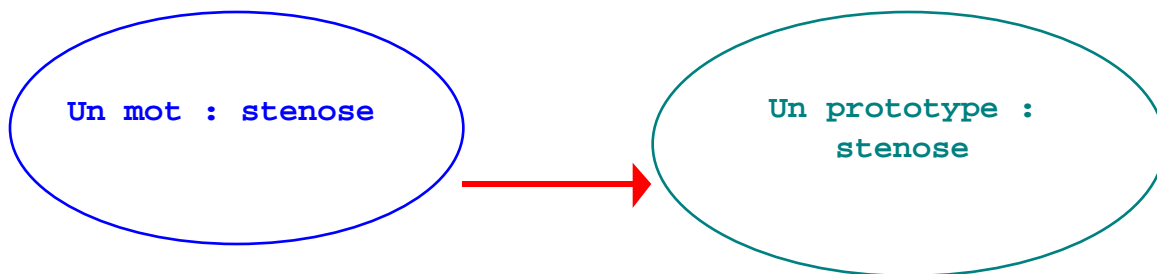
Esquisse d'une démarche de représentation

○ CYCLADE révèle des proximités (comportementales) entre formes

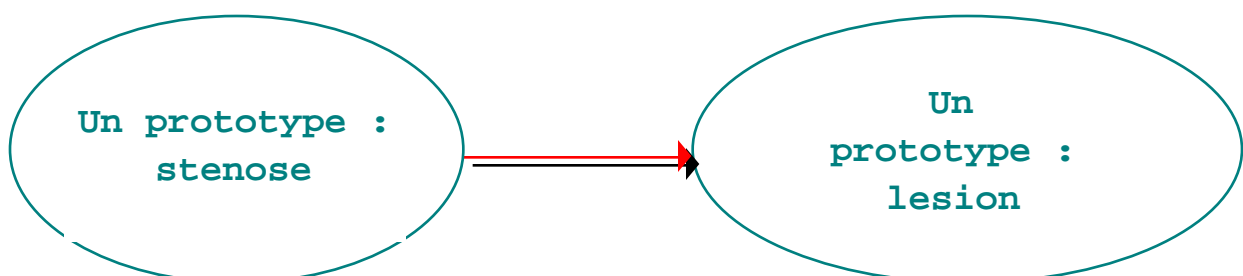
- stenose et lesion appartiennent à la même catégorie
- Contructions partagées par stenose et lesion

○ Utilisation des savoirs particuliers à un mot pour représenter une catégorie

- Un mot -> une représentation informatique

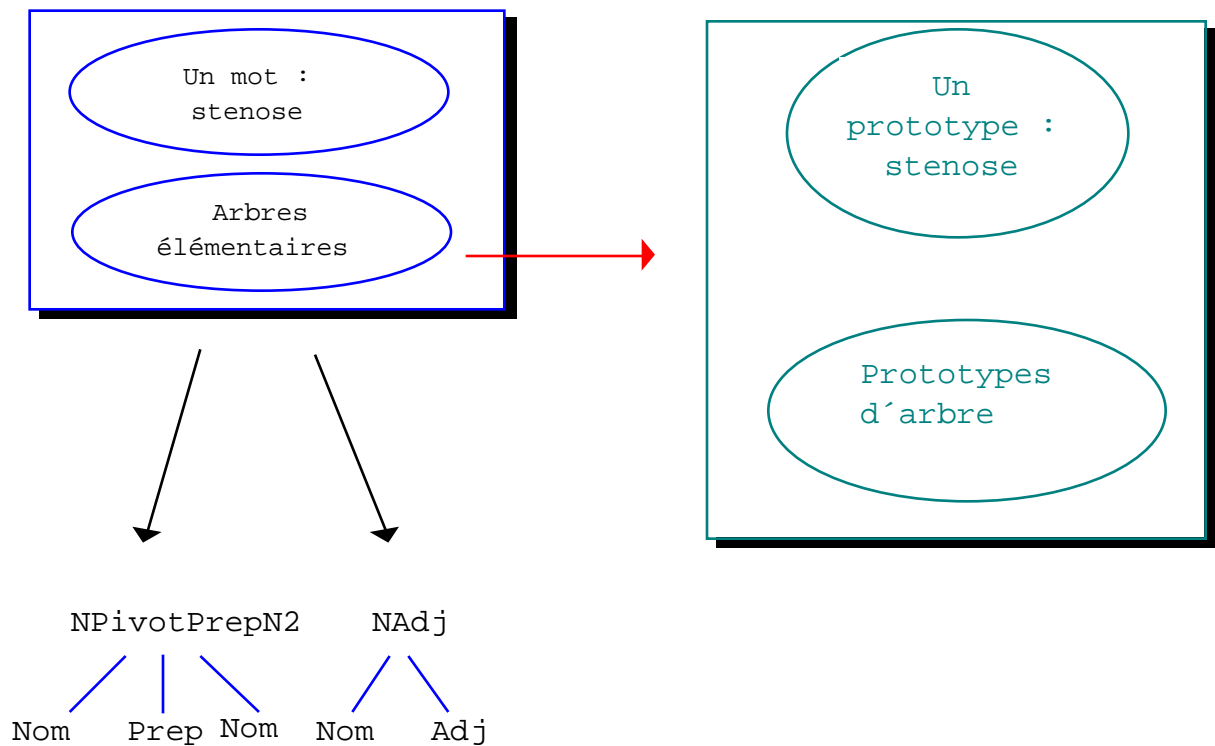


○ Représentation d'unités semblables par copie et ajustements



○ Représentation des comportements lexicaux

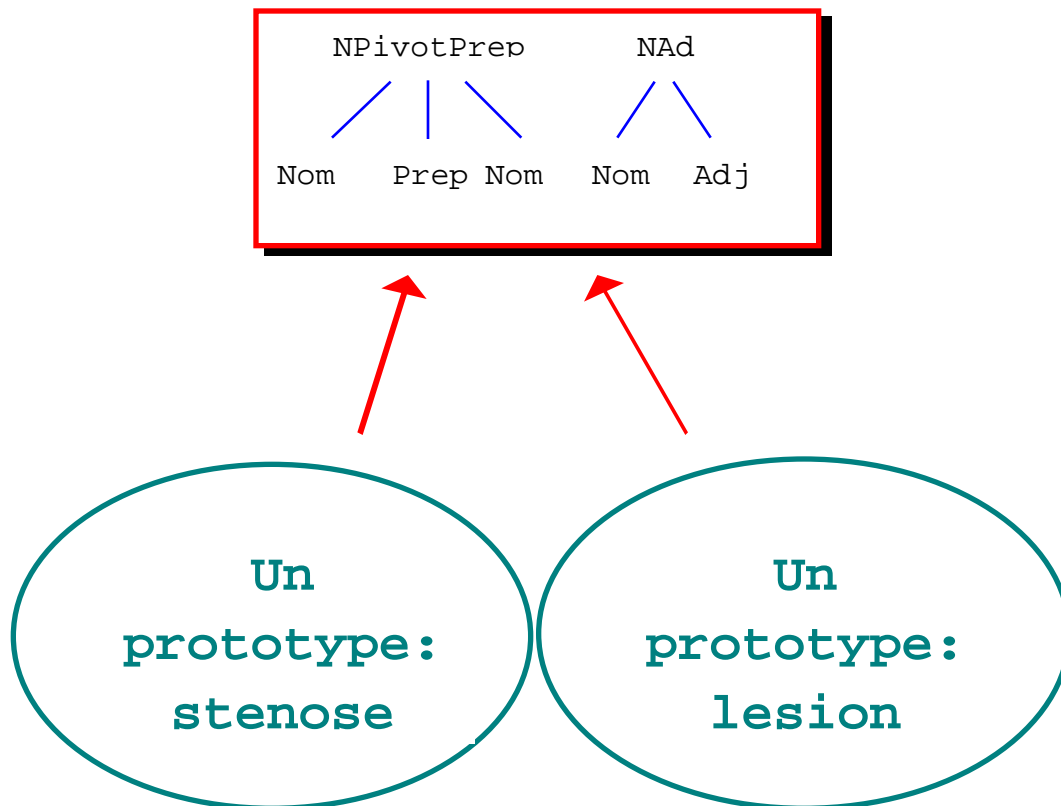
- On commence par construire les structures pour représenter les arbres associés à *stenose*



- Les prototypes d'arbres déjà construits pour *stenose* et partagés par *lesion* seront associés au prototype représentant *lesion*

○ Partage des comportements communs

Prototypes d'arbres partagés



- Les mots stenose et lesion partagent des arbres élémentaires
- On construit un pôle de savoir partagé

Que fait la PàP ?

- **Représentation des connaissances peu hiérarchisées**
- **Représentation d'un domaine en utilisant un élément particulier du domaine**
- **Représentation d'autres éléments par clonage et ajustements**
- **Partage local d'informations via la délégation : mise en place d'héritages locaux de comportements partagés**

PàP : Programmation A Prototypes

- **Programmation à Prototypes (Lieberman 86, The Self Group 87-95, Blascheck 1994, Smith 1995 Projet Newton)**
- **Prototypes vs. Instances de Classes**
 - **PàO**

Un objet est une instance de classe

Nouveau type d'objet -> nouveau type de classe

Modifier la structure ou les comportements d'un objet -> modifier la classe
 - **PàP**

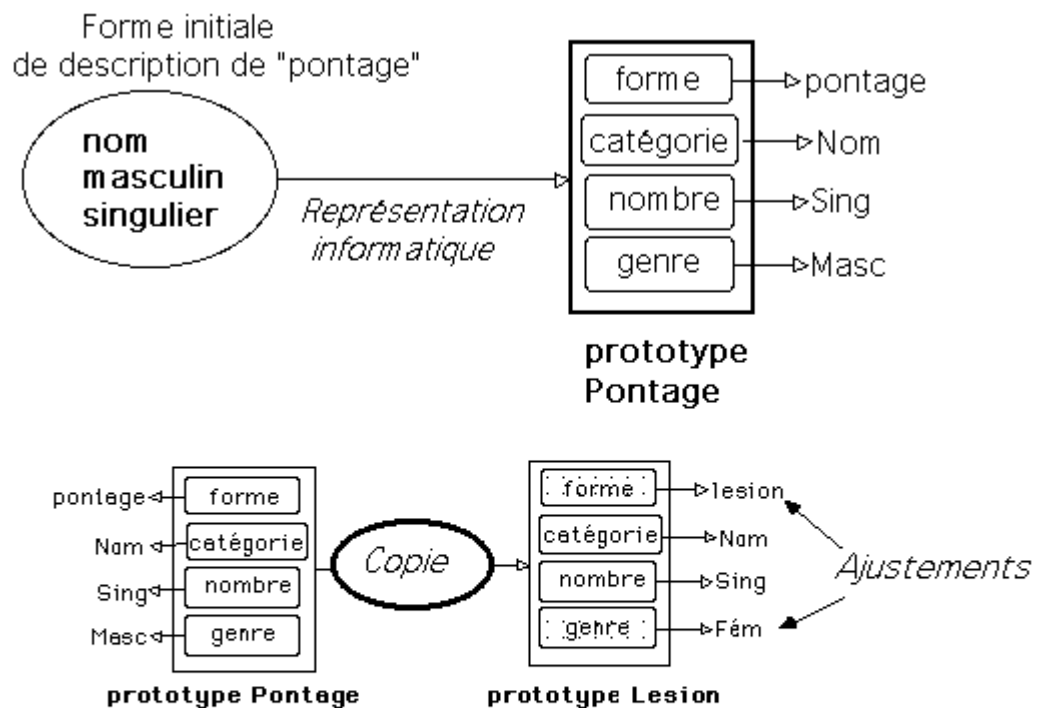
Nouvel objet par copie d'un objet existant

Modification de la copie sans altérer l'objet initial

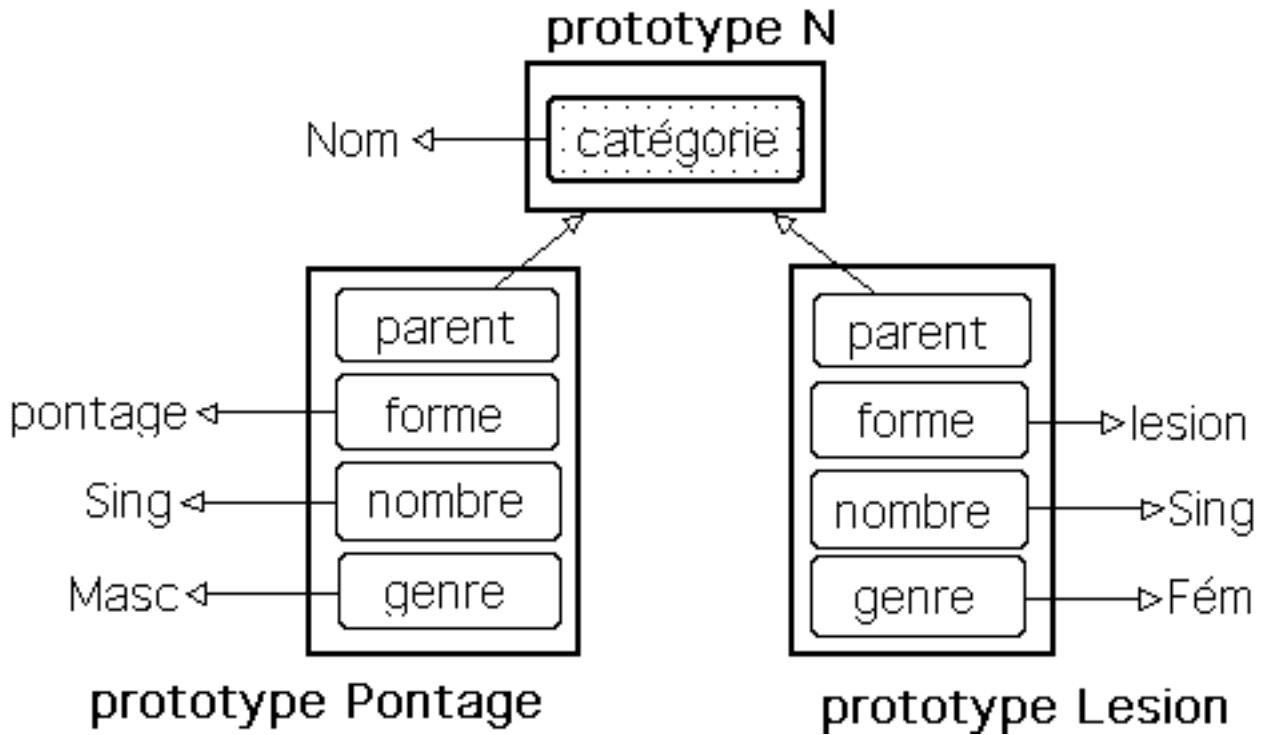
○ Une entité peut être représentée par un prototype

- Construction d'une représentation conceptuelle d'un domaine par la connaissance d'un élément particulier de ce domaine dans un contexte donné

○ Représentation d'une entité similaire par clonage et différenciation



o Partage des structures et des comportements communs par **délégation**



Self

(Sun Microsystems Lab.)

○ Langage à Prototypes

- Héritage multiple
- Héritage dynamique
- Conçu en 1986 par David Ungar & Randall Smith

○ "Self : the power of Simplicity" OOPSLA'87

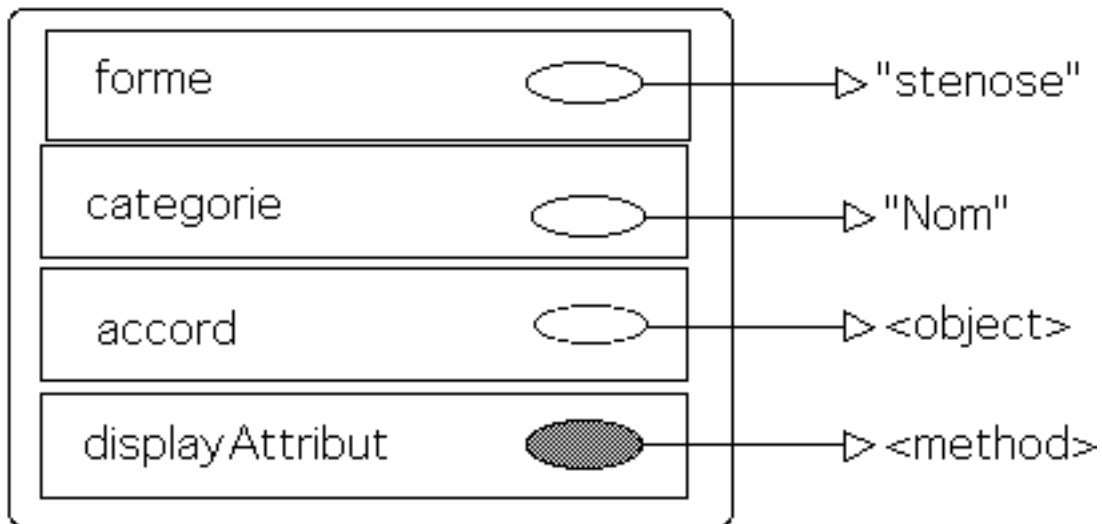
- 1ère implémentation à Stanford 1987
- Dernière version : Self-4.0, juillet 1995

Objets et attributs Self

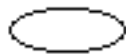
- **Attribut = méthode ou données**
- **Les contenus des attributs peuvent évoluer**
- **Ajout(s) ou retrait(s) dynamique(s) d'attributs**

=> utile pour construire des représentations par affinements successifs

prototype Stenose



Attribut : méthode

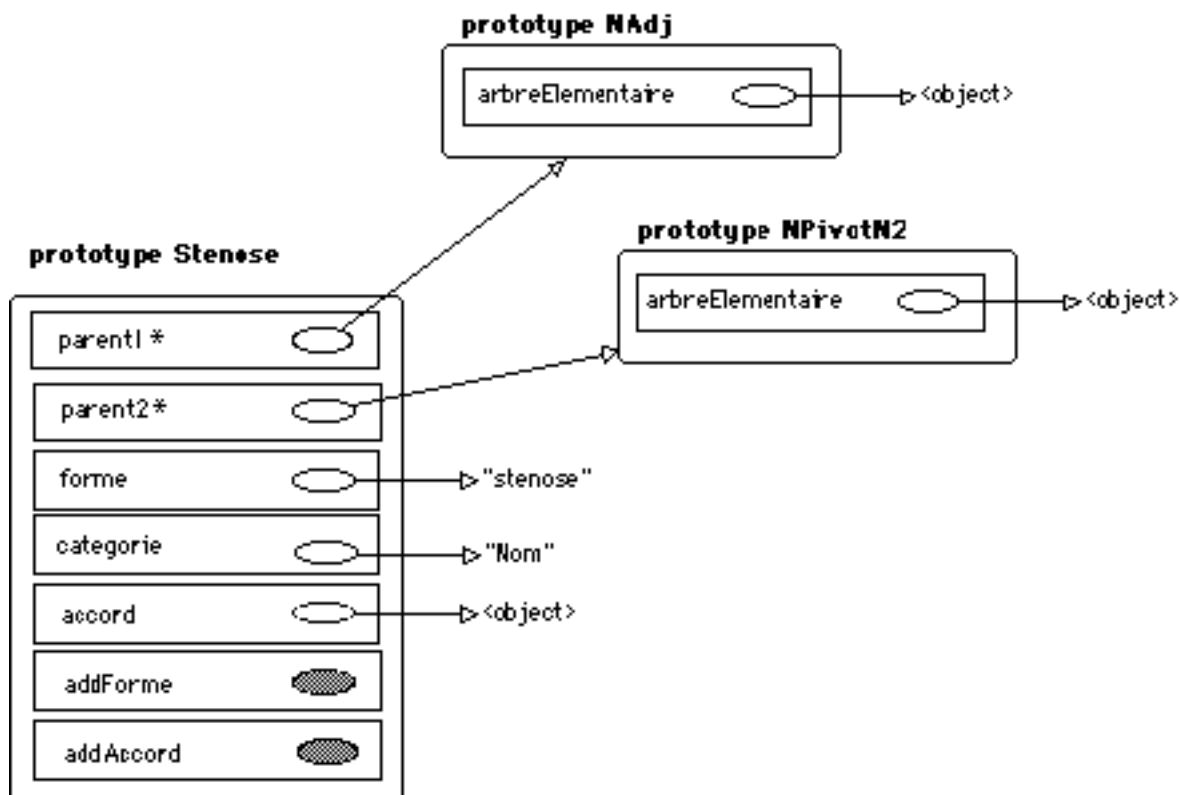


Attribut : Valeur

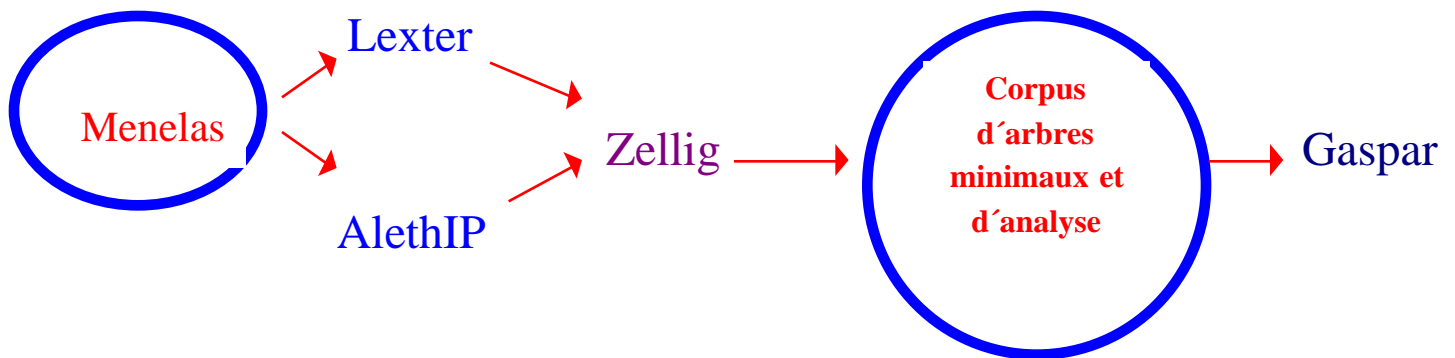
Héritage et Self

La délégation, une vision dynamique de l'héritage

- Attributs **parent** : délégation d'attributs vers d'autres objets
- Un parent commun à plusieurs prototypes est appelé un **objet traits**
- Les traits portent les comportements partagés



Construction inductive des savoirs



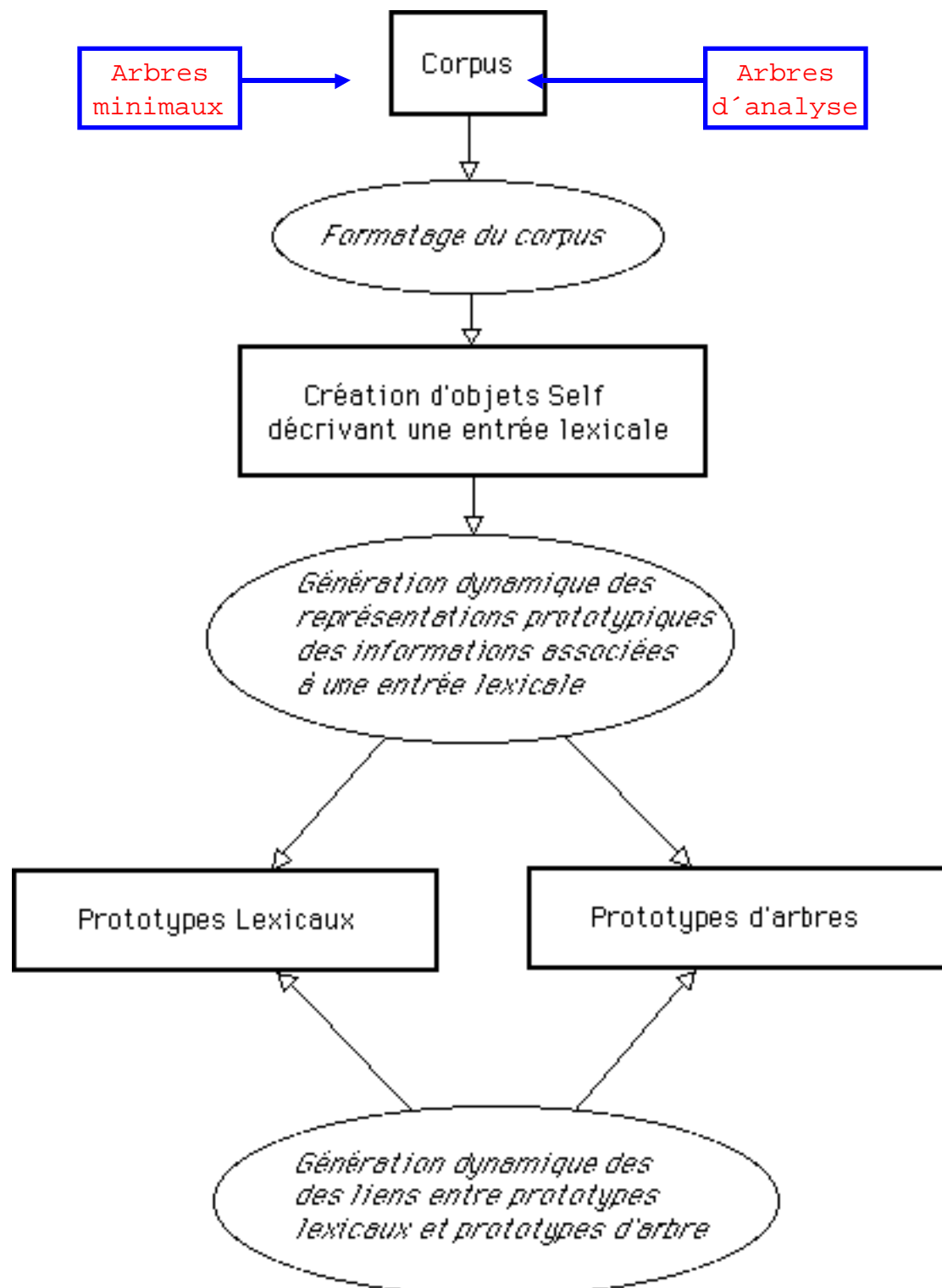
○ Représenter

- Faciliter la représentation des connaissances associés aux mots via la PàP
- Apprendre aux prototypes lexicaux leurs comportements à partir d'un corpus
- Le avoir attaché au mot n'est pas encodé à la main; il est affecté dynamiquement et automatiquement au prototype qui le représente

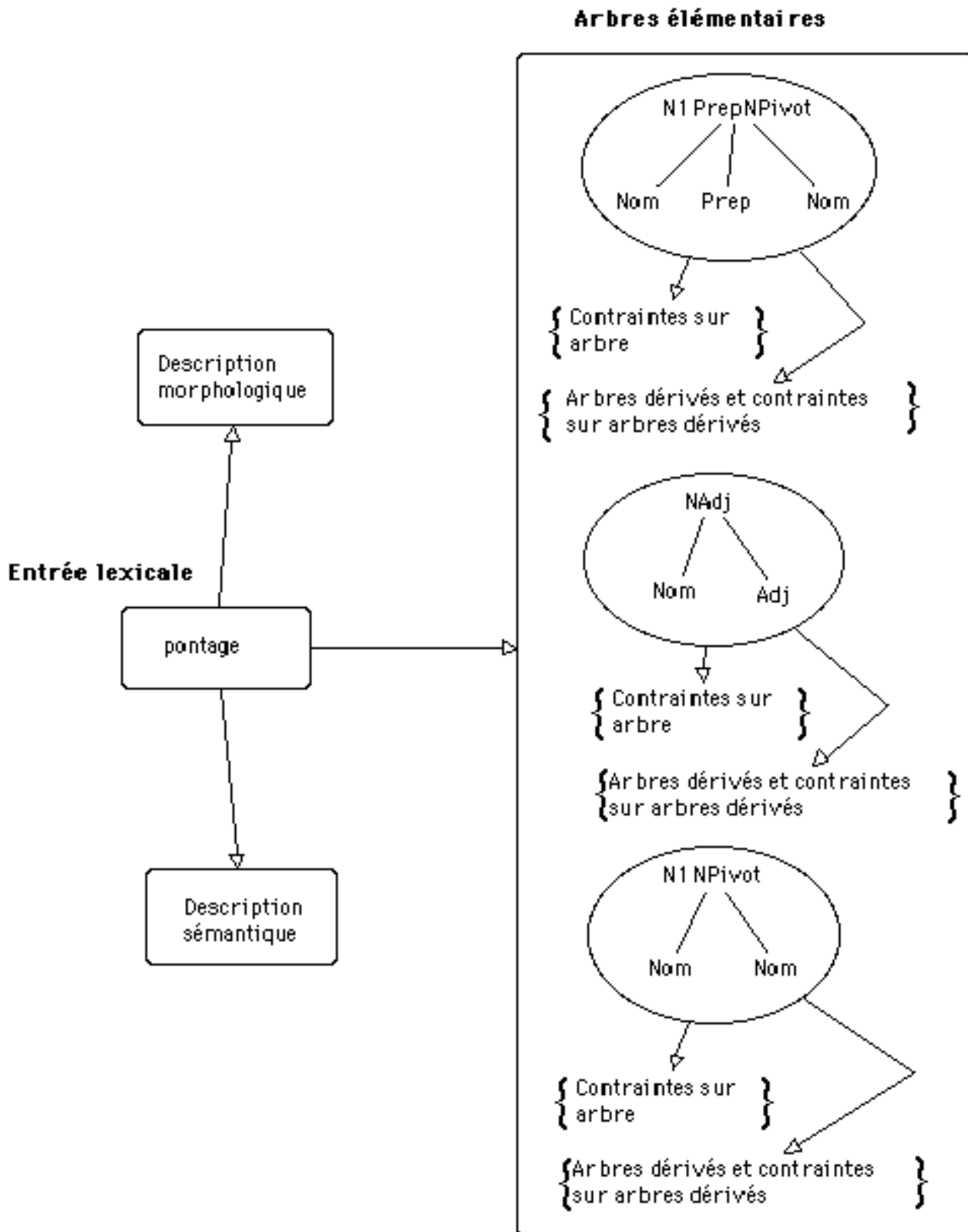
○ Classer

- Classer les représentations construites suivant leurs comportements

Représentation dynamique des unités lexicales

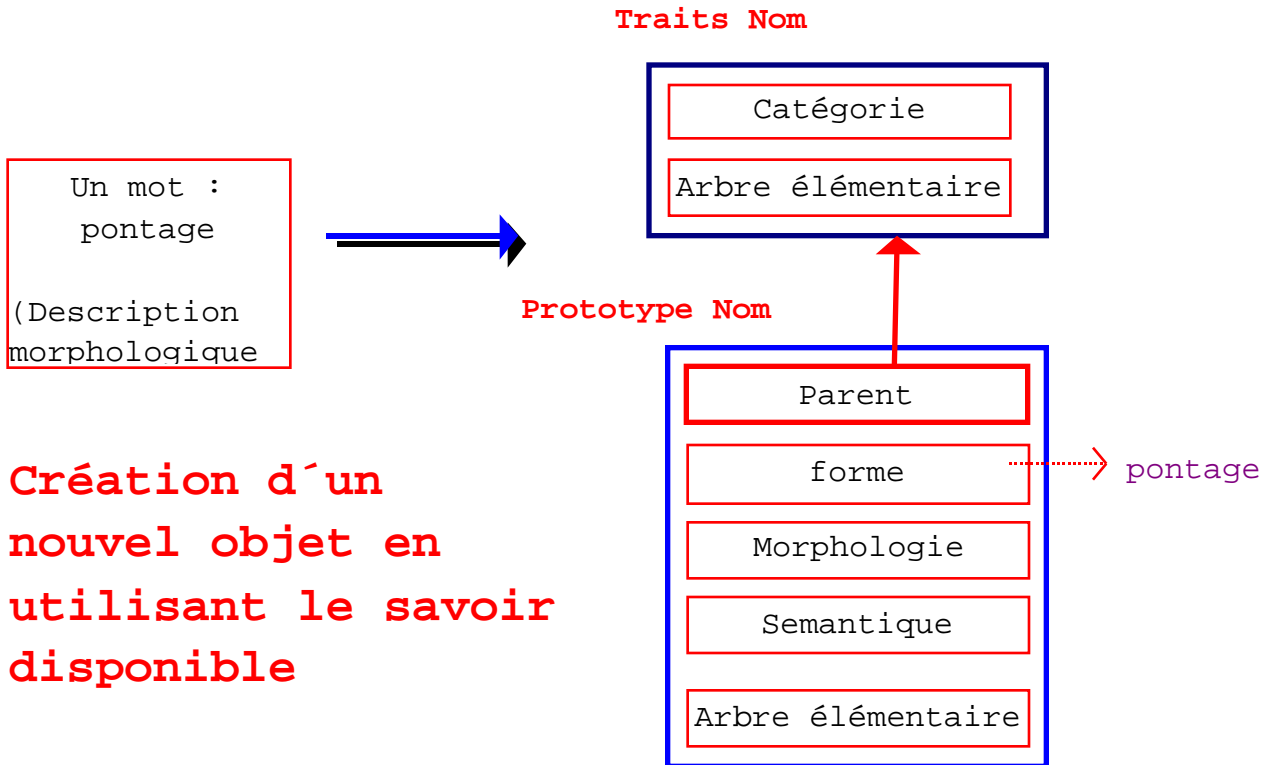


○ Utilisation des résultats d'analyses syntaxiques pour apprendre aux mots

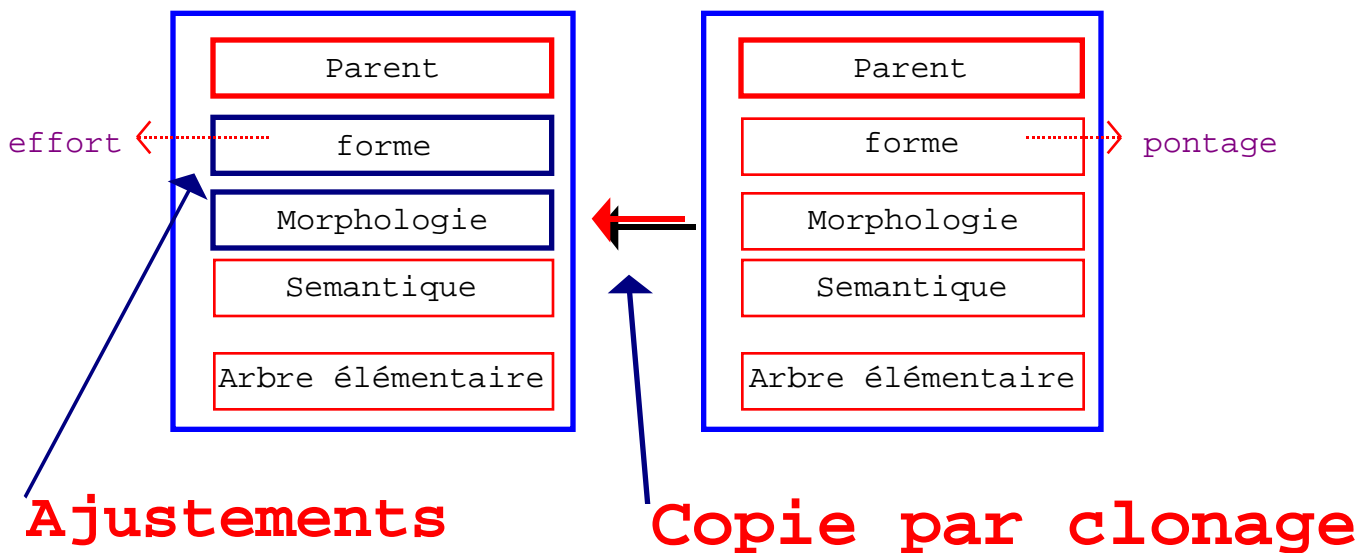


○ Création automatique des prototypes de mot

Génération à la volée (automatique) de prototypes par création explicite d'un nouvel objet



Génération à la volée (automatique) de prototypes par clonage et ajustements

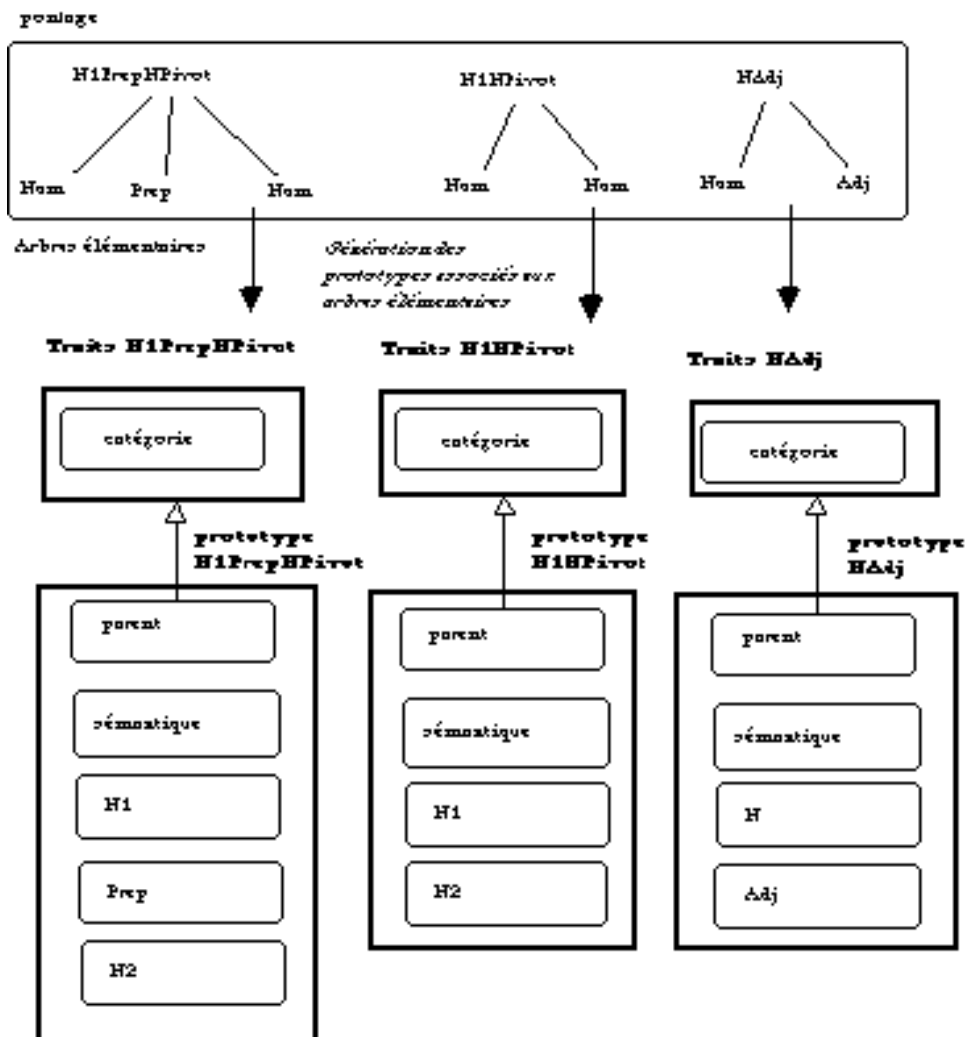


Représentation dynamique des contraintes syntaxiques

o Génération automatique des prototypes d'arbres "élémentaires"

- Création explicite d'un nouvel objet

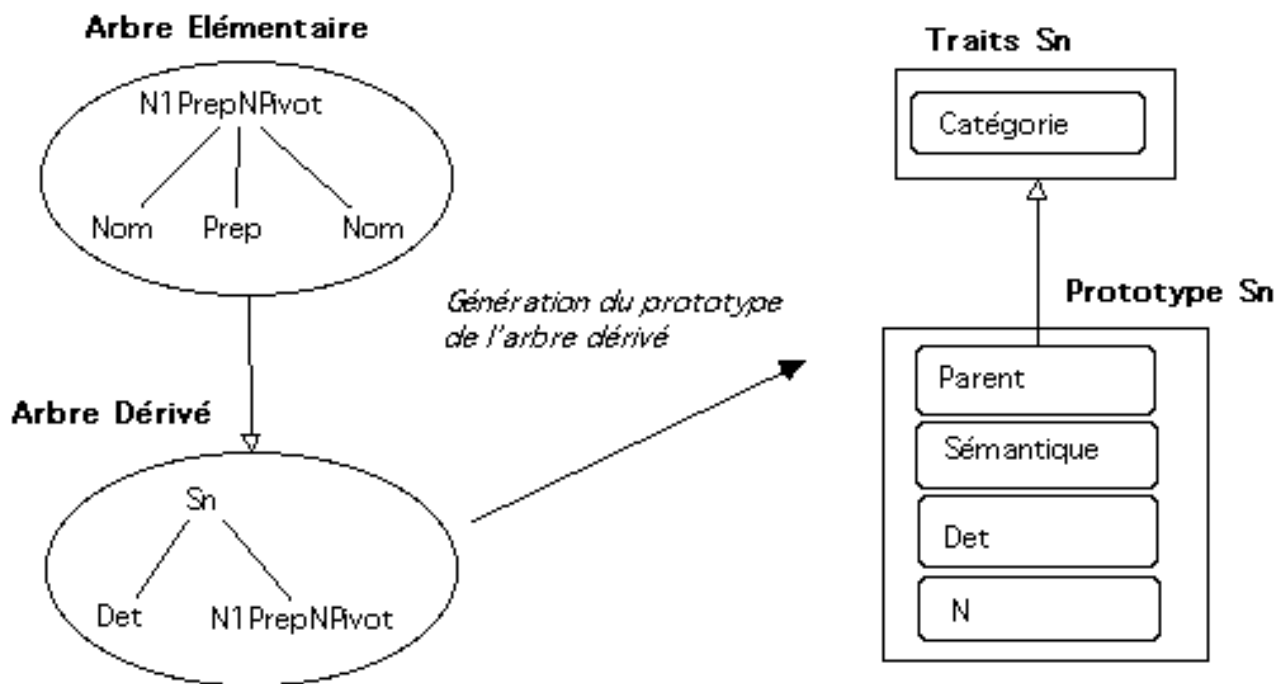
Si la catégorie syntaxique tête d'un arbre "élémentaire" n'est pas encore représentée, on définit les objets qui vont la représenter avec les attributs adéquats



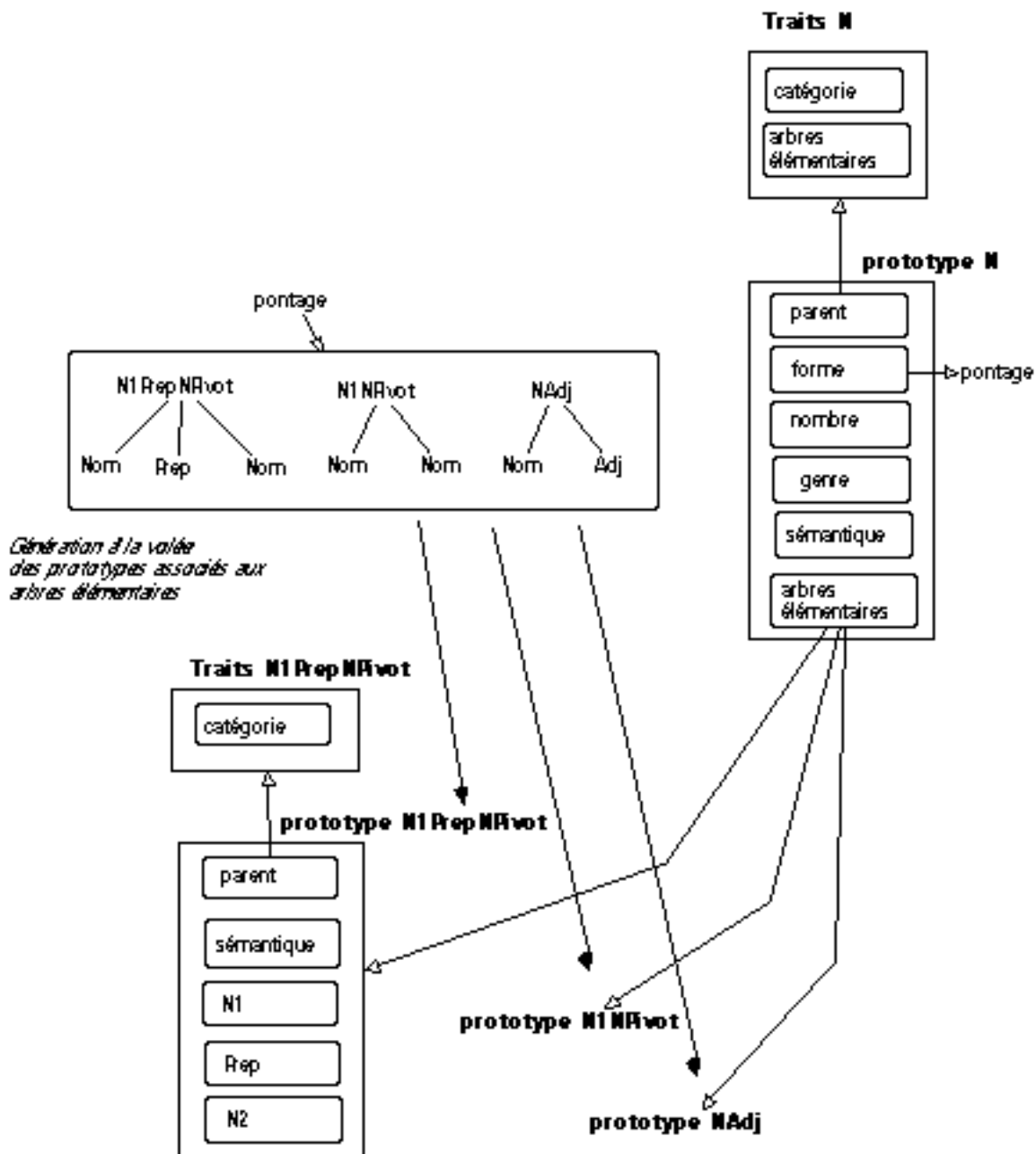
○ Génération automatique des prototypes d'arbres d'analyse

- Création explicite d'un nouvel objet

Si la catégorie syntaxique tête d'un arbre d'analyse n'est pas encore représentée, on définit les objets qui vont la représenter avec les attributs adéquats

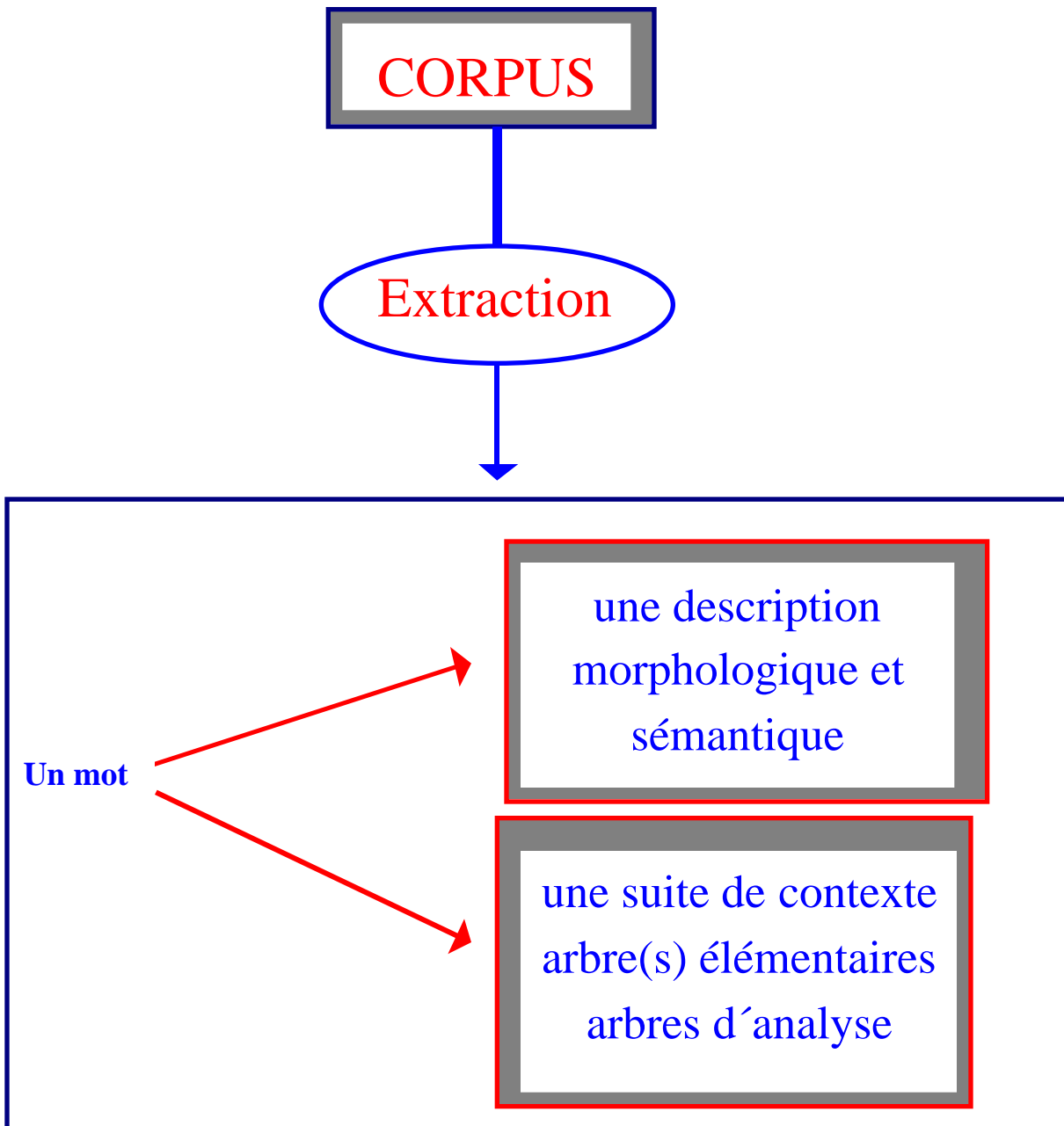


- Mise en place des liens entre prototypes lexicaux et prototypes d'arbres "élémentaires"
- Mise en place des liens entre prototypes d'arbres

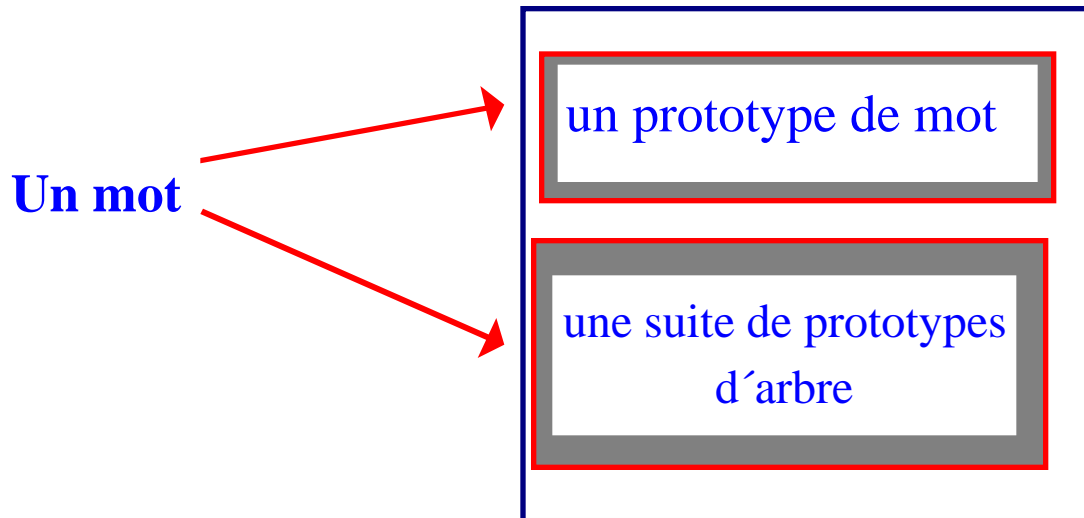


Représentation évolutive des mots

○ Avant Phase de génération



○ Après Phase de génération



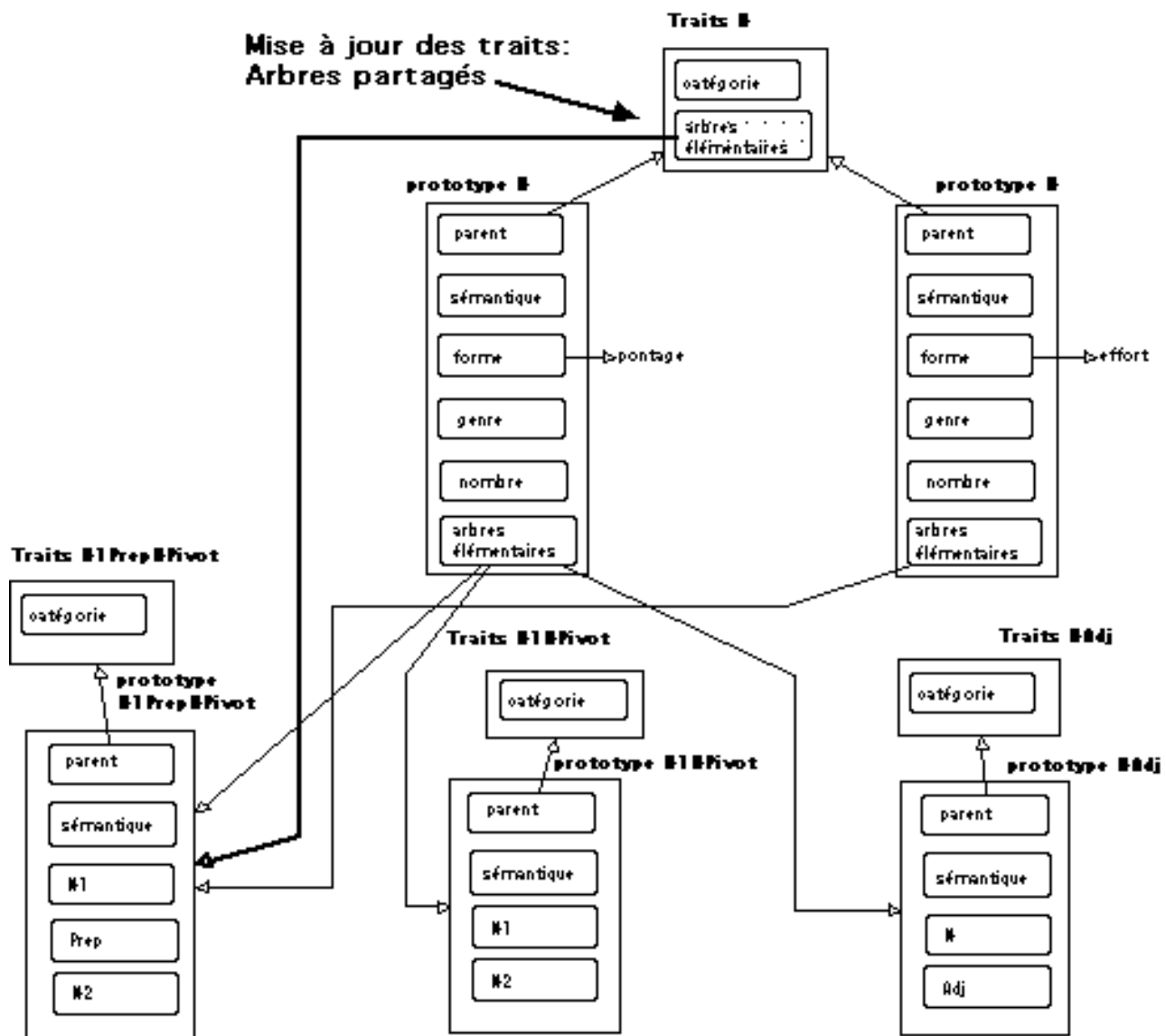
Représentation sous déterminée des mots : sémantique, morphologique...

C'est l'examen des contextes (i.e les arbres associés) qui doit permettre de **construire un sens possible** ou **tracer une piste de sens**

=> **Affinements possibles de manière dynamique et à tout moment des objets construits**

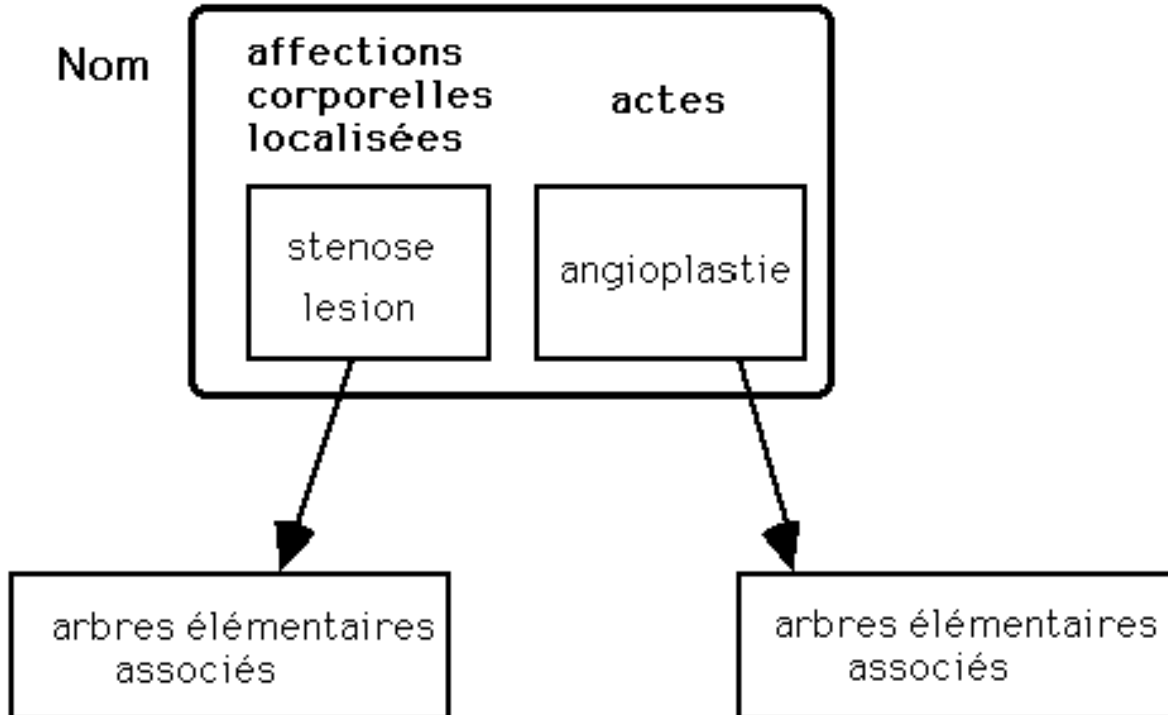
Classement des prototypes

- Mise à jour automatique des traits : héritage local des comportements partagés



Un premier classement des prototypes suivant leurs comportements syntaxiques

- stenose et lesion partagent des comportements
- angioplastie entre dans des constructions du type "indication de angioplastie"
- Avec les noms stenose, lesion et angioplastie on obtient le mini réseau suivant



Classement des mots (prototypes de mots) suivant leurs comportements (prototypes d'arbres associés)

Retour sur Self et l'héritage

Se déplacer dans un réseau de prototypes

(Comment sélectionner un comportement associé à un mot)

○ Self permet l'héritage multiple

○ Self permet l'héritage dynamique

- Le lien entre un objet et son parent peut être un attribut d'assignation
- Modification dynamique de délégation

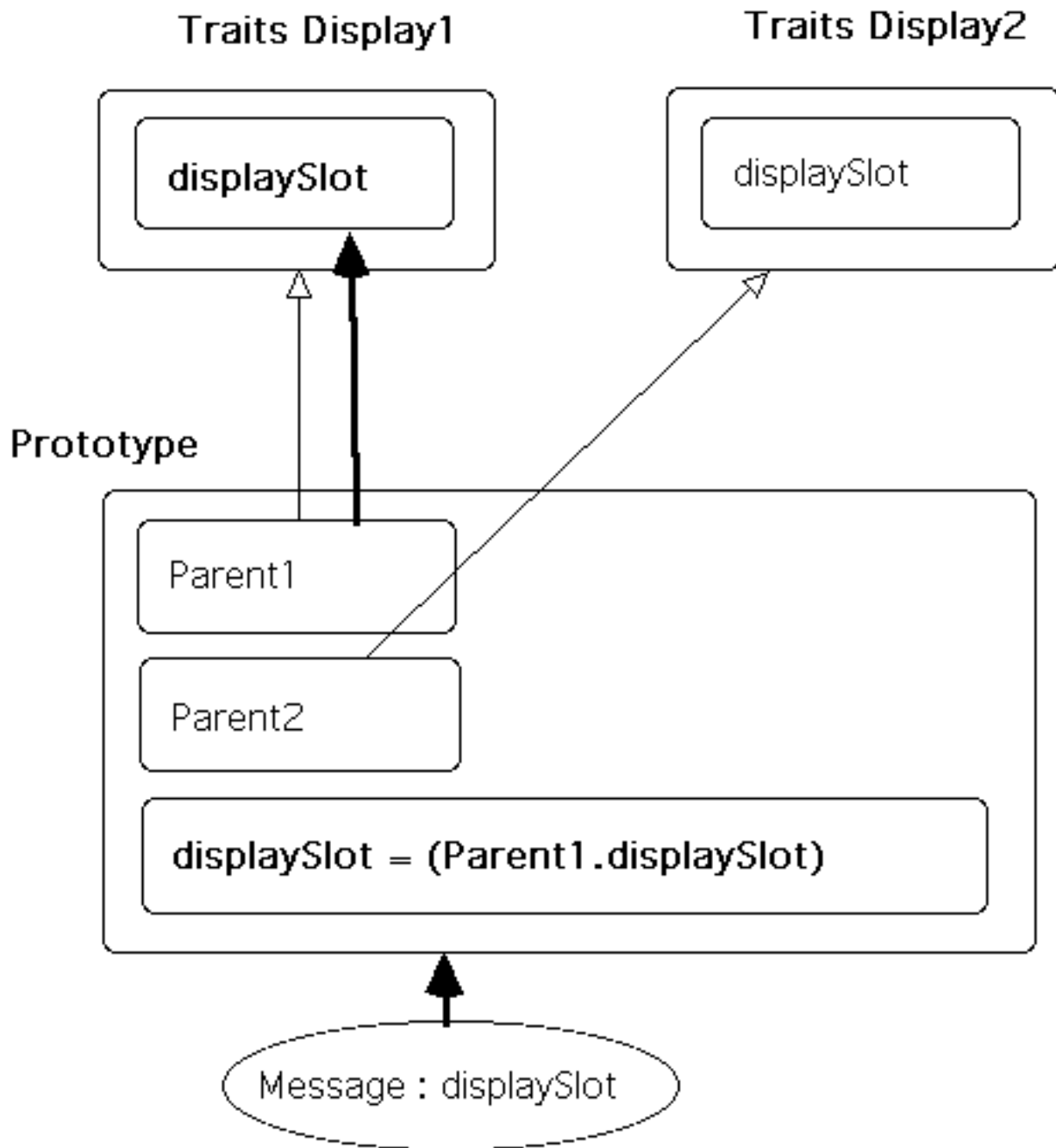
-> ajout ou modification de délégation

○ Héritage non-ordonné

- Pas de rang de priorité dans l'héritage
- Génération des messages d'erreur si un ou plusieurs attributs de parents distincts portent les mêmes noms
- Si un objet et l'un de ses ancêtres définissent un attribut avec le même nom, c'est l'attribut de l'objet qui sera pris en compte

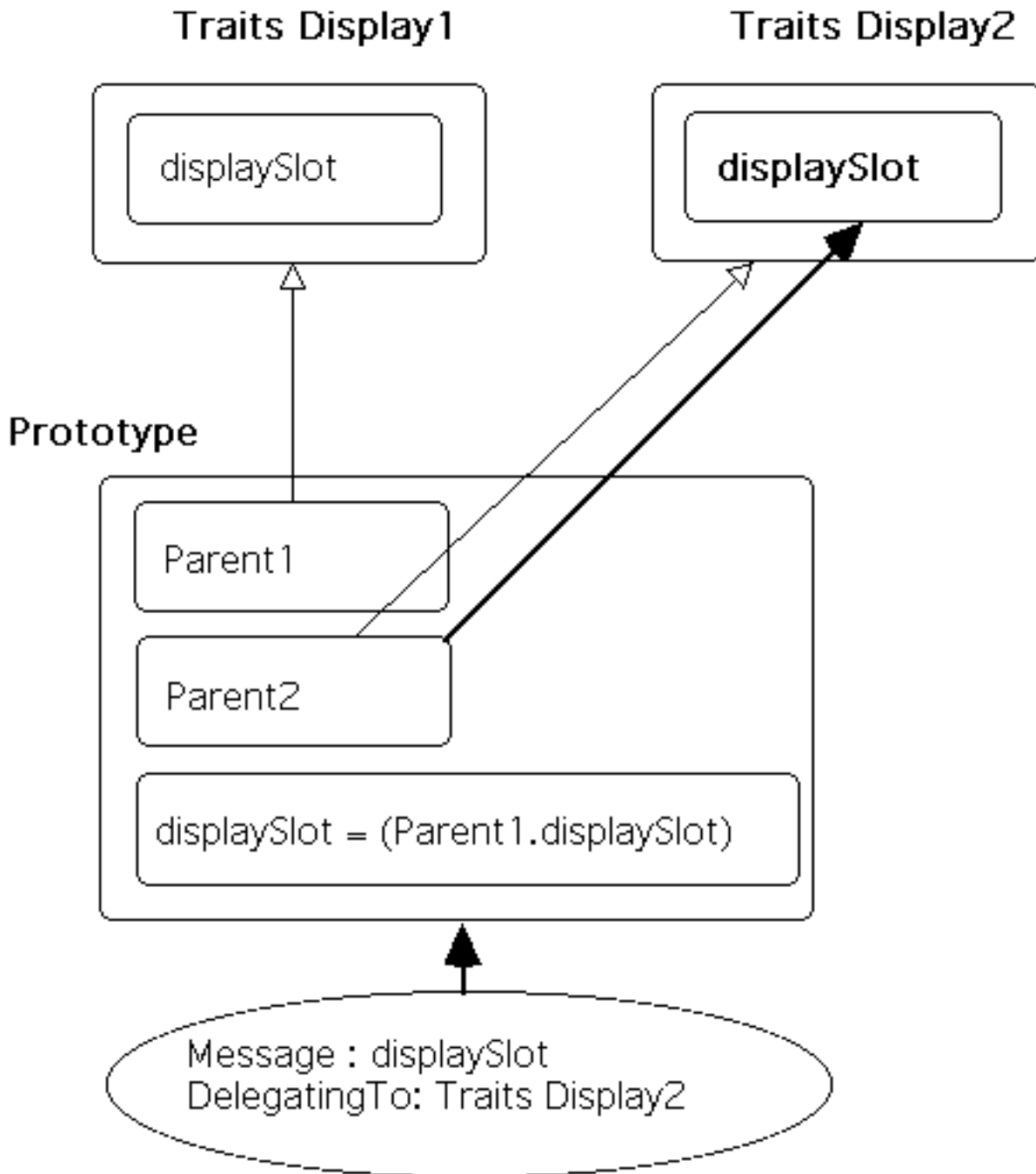
Renvoi dirigé de messages

o Receveur implicite



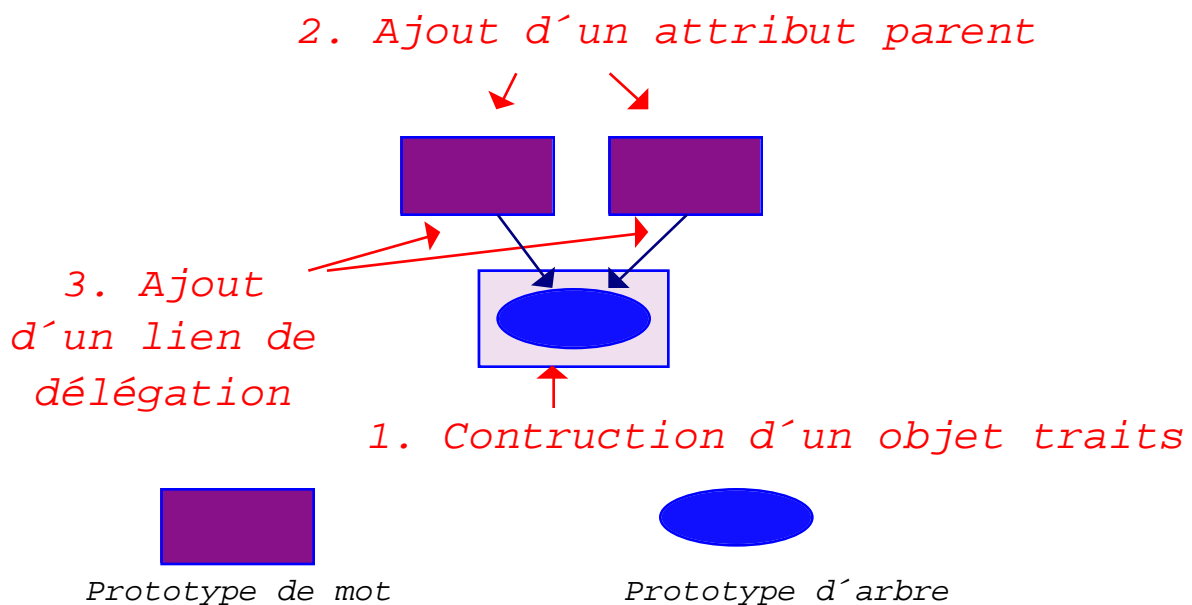
o Receveur explicite

'message' sendTo: Objet DelegatingTo:
parentChoisi



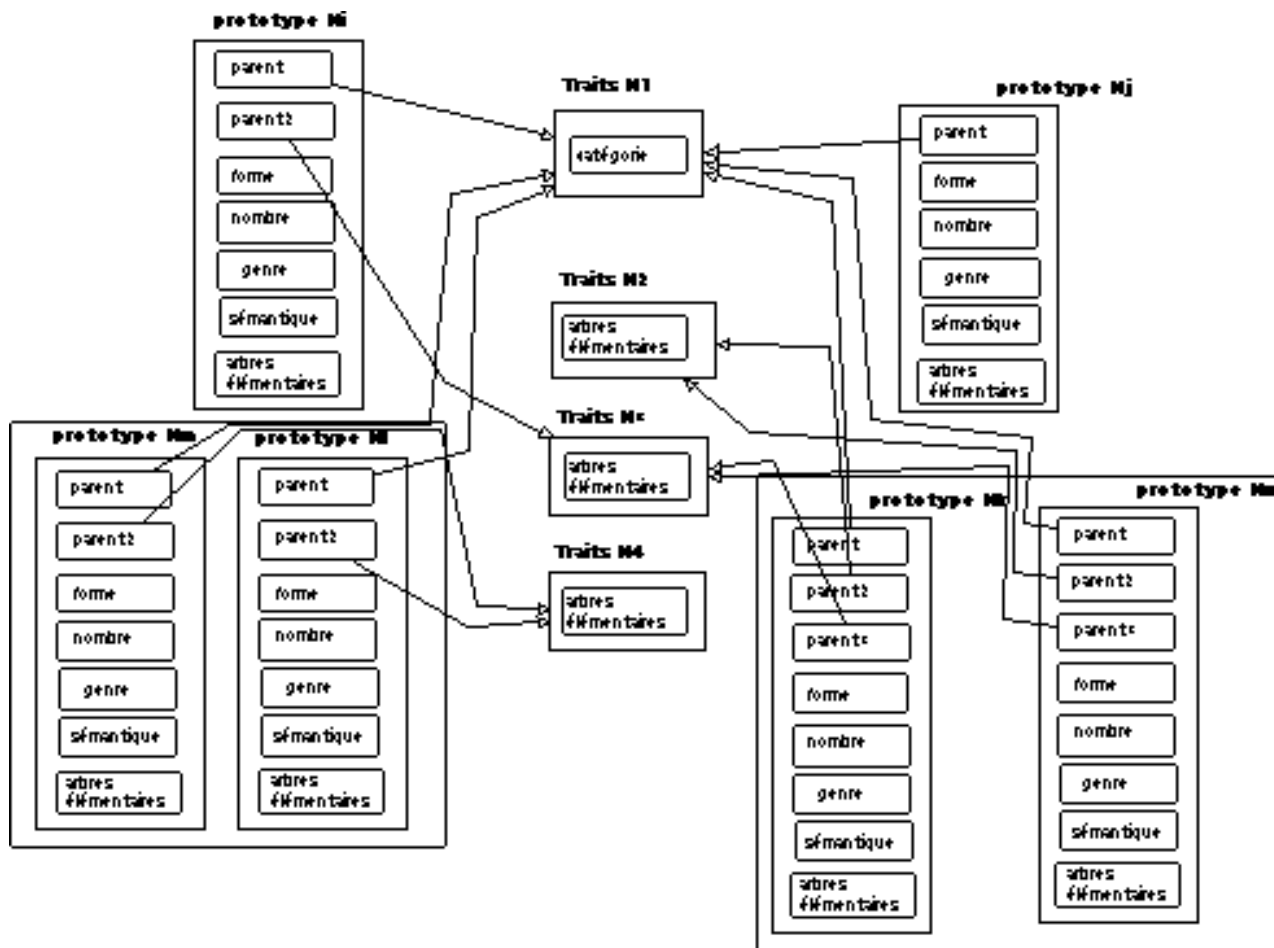
Classification évolutive

- (1) Recherche sur tous les mots d'une même catégorie des comportements partagés (i.e. détermination des arbres propres à chacun d'eux)
- (2) Recherche des arbres élémentaires communs à deux prototypes (i.e. détermination des arbres propres à chacun d'eux)
- Examen des comportements partagés par les mots (deux à deux)



○ (3) Recherche sur des sous-familles de mots d'une même catégorie des comportements partagés et construction automatique d'un pôle de comportements partagés

- Examen des comportements partagés sur des sous-familles de mots d'une famille catégorielle



○ (4) Recherche sur des sous-familles d'arbres élémentaires d'une même catégorie des comportements partagés (arbres d'analyse)

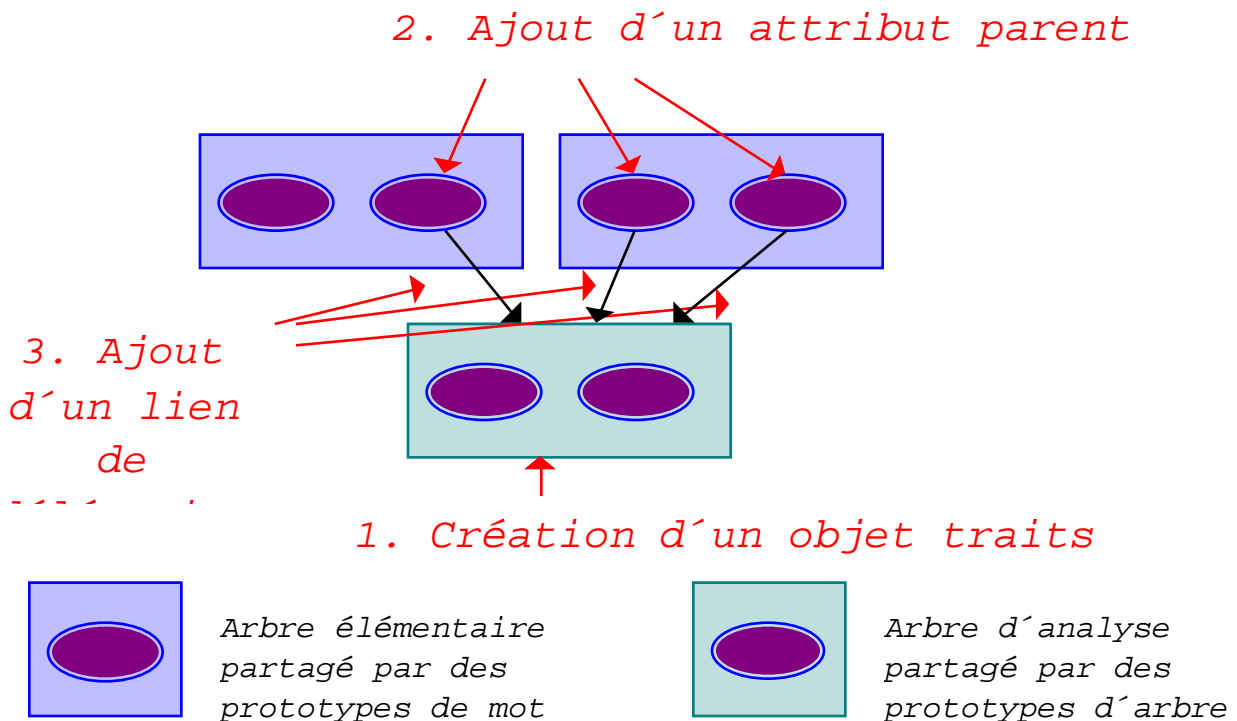
et construction d'un pôle de comportements partagés

- Examen des comportements partagés par les arbres élémentaires

effort et myocarde entrent dans des séquences N1 Prep N2 en position N2

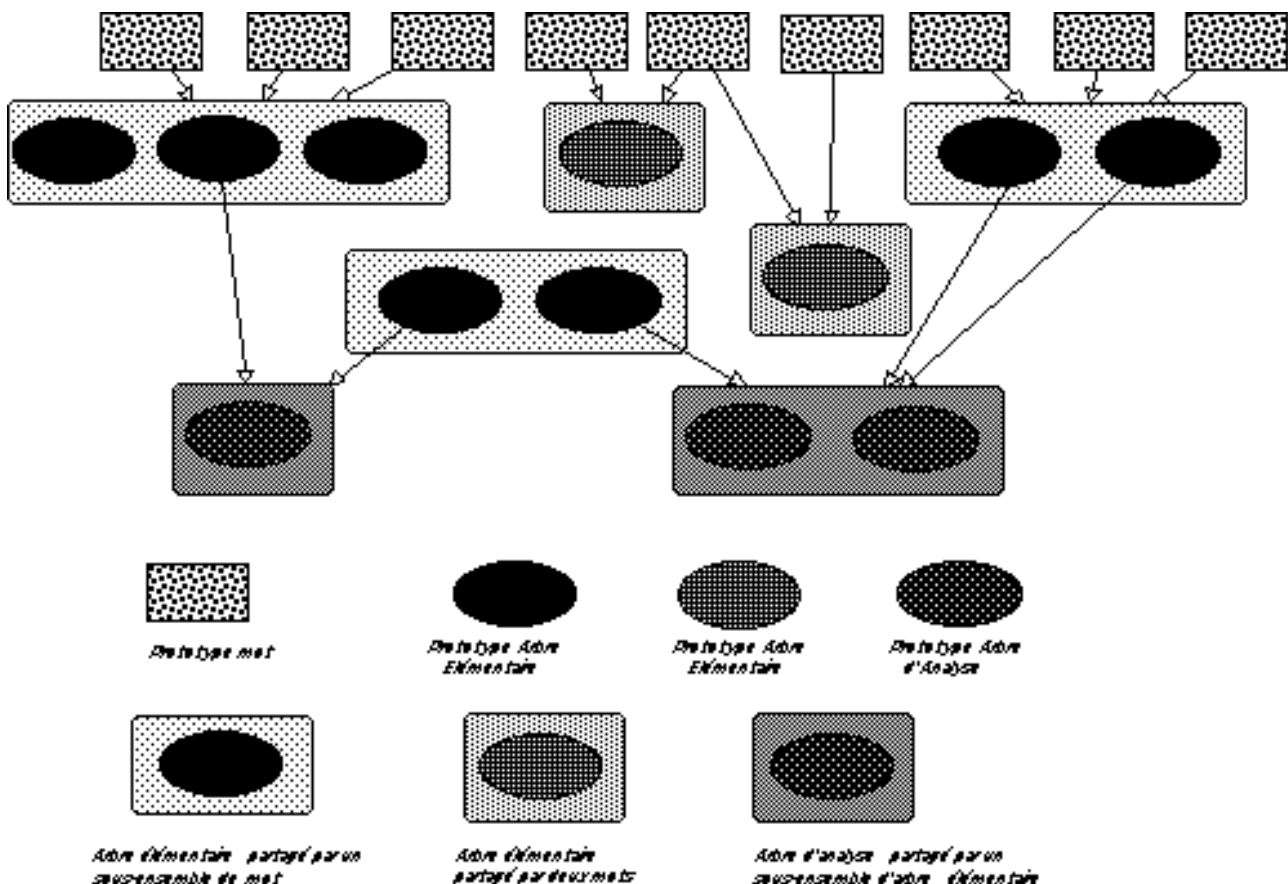
Différence de comportements de ces deux mots dans des arbres d'analyse :

effort ne peut pas y être modifié, myocarde l'est toujours par un déterminant



Mise en place de réseaux de prototypes : autant de réseaux à interpréter

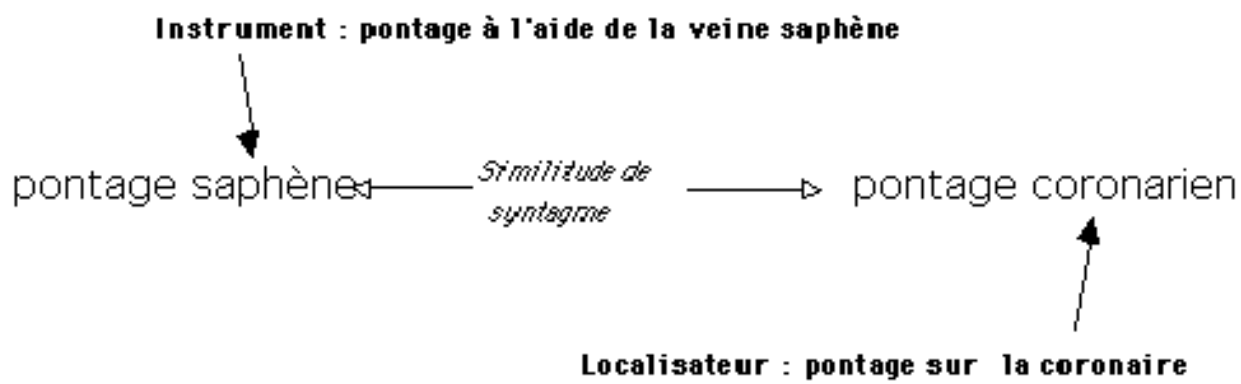
- Des regards multiples et croisés sur les savoirs représentés
- Ces liens multiples constituent autant de pistes de sens à interpréter



Amorçages d'interprétation

○ La syntaxe pour dégrossir le classement

- Rapprochement de certaines unités lexicales suivant certains comportements syntaxiques

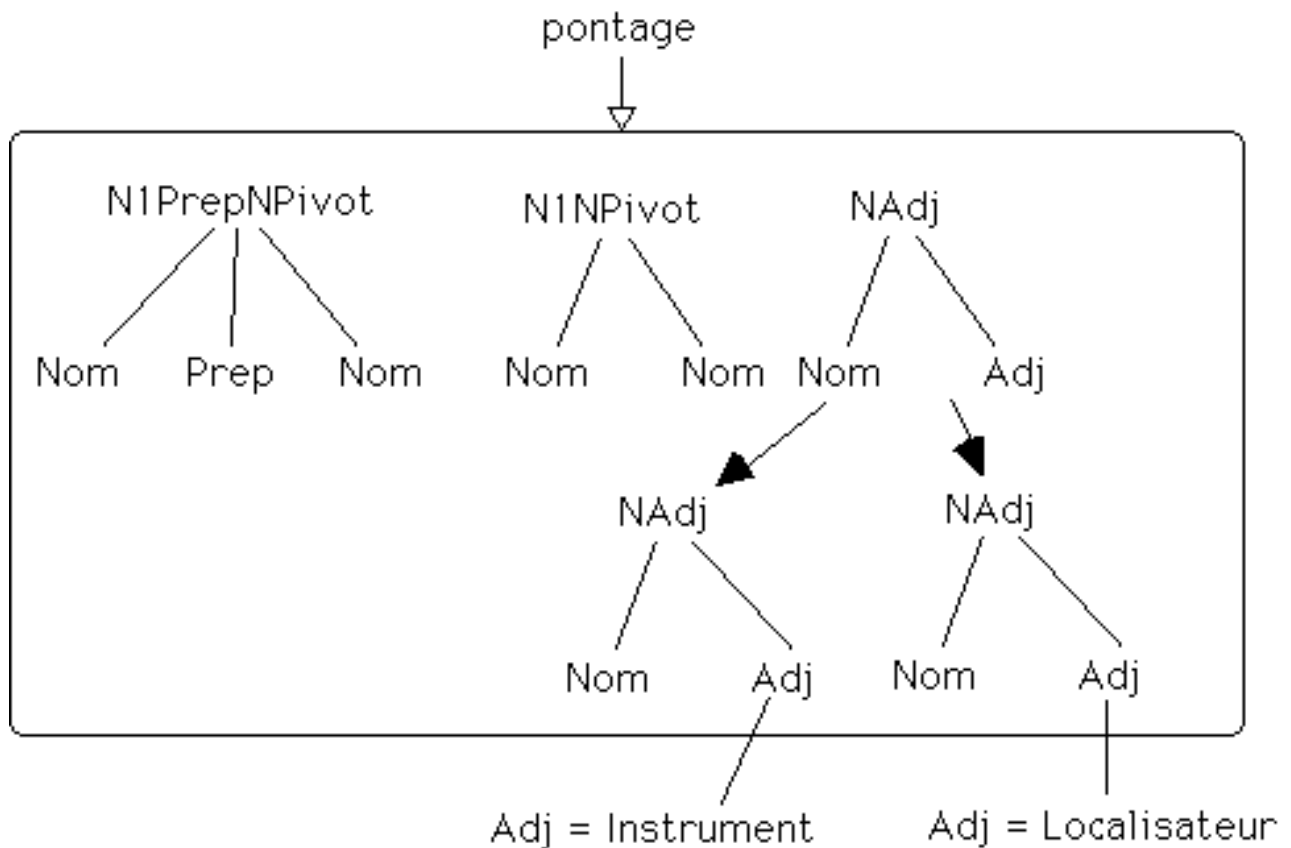


○ La syntaxe est incapable à elle seule de délimiter des classes de mots reflétant une notion

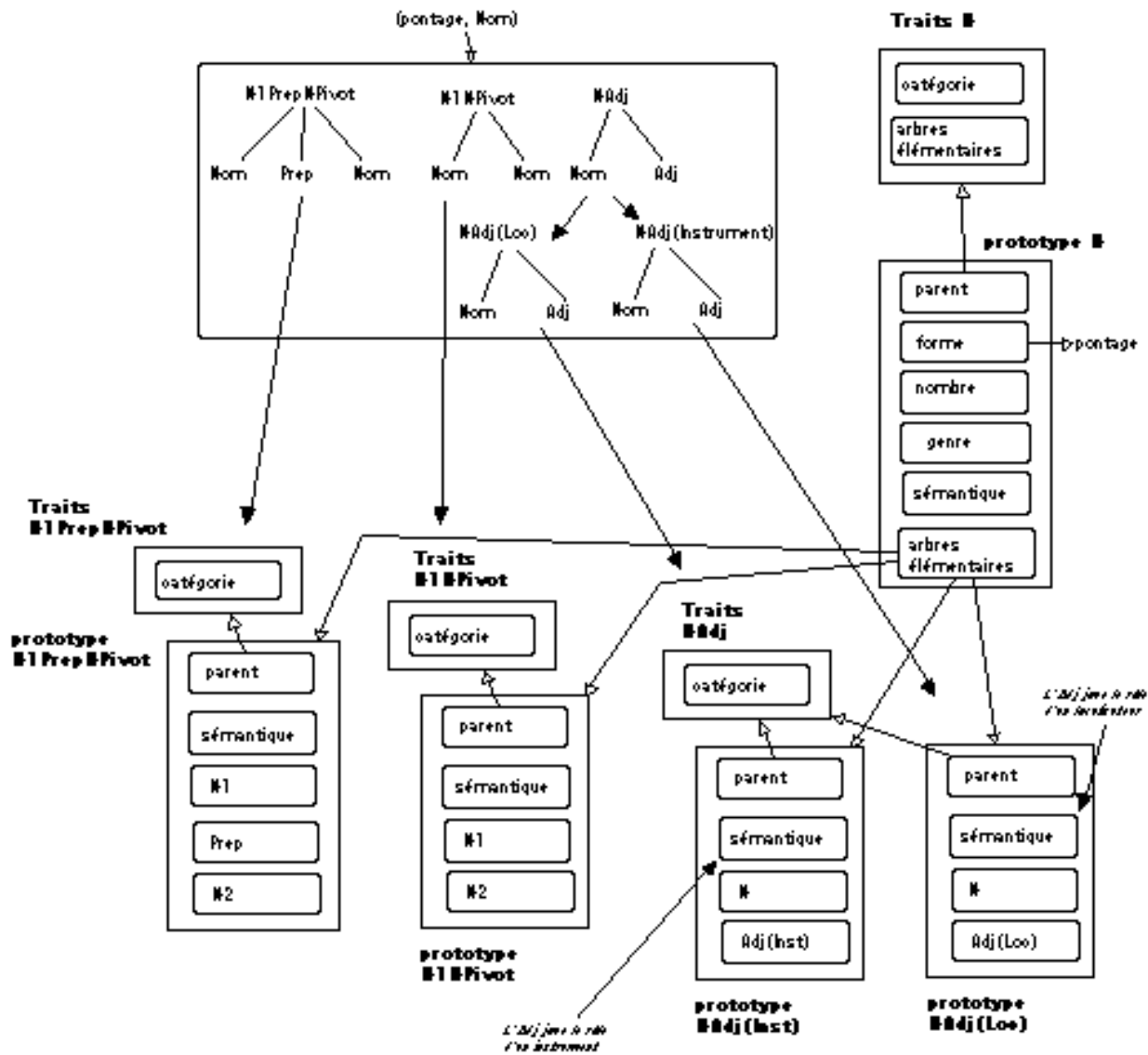
- "La syntaxe comme marche-pied de l'acquisition des connaissances" : (Habert & Nazarenko 1996)

○ Affinement de la représentation en contraignant les arbres élémentaires

- Affiner les savoirs initiaux....



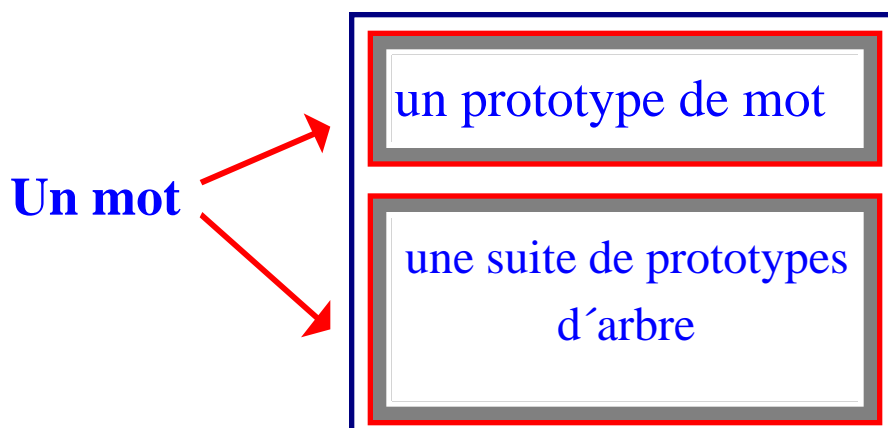
- Pour affiner les contraintes à construire



Des réseaux de hiérarchies évolutives

○ Avant Phases de classement

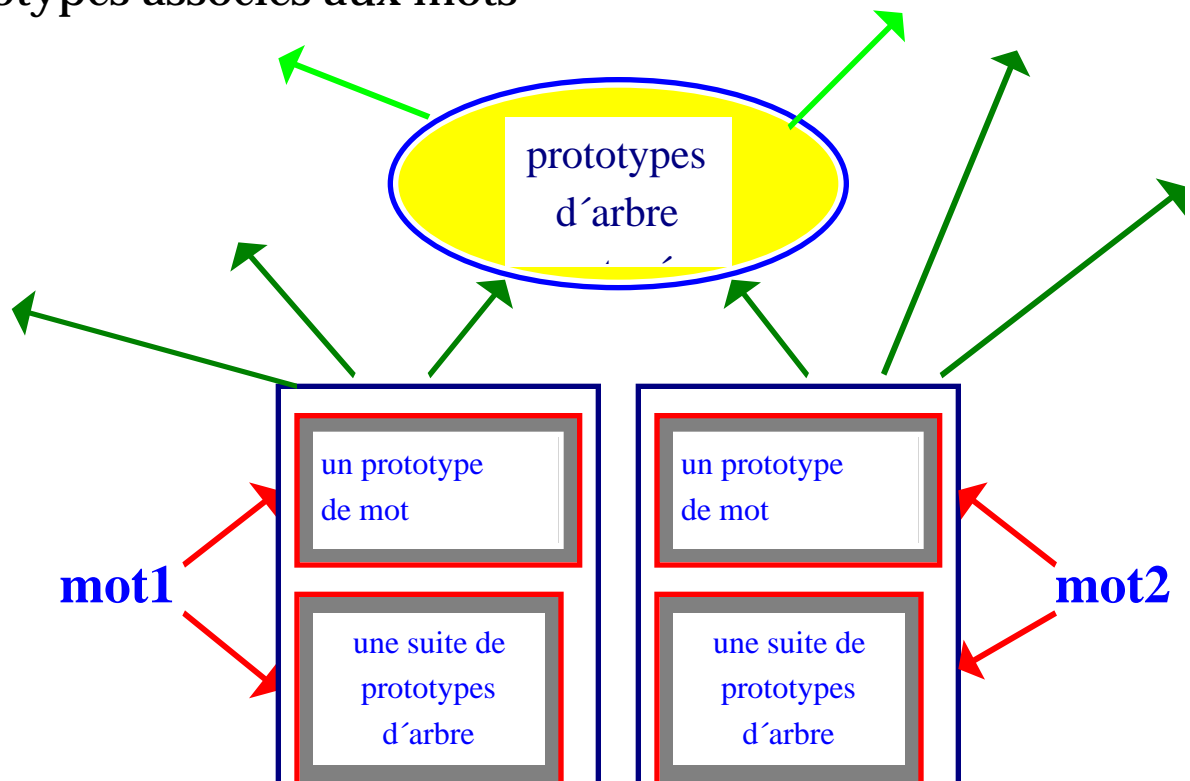
Un mot → un micro-réseau de prototypes (mots + arbres)



○ Après Phases de classement

→ mise en place de liens entre les prototypes associés aux mots

micro-réseaux de



Une démarche interprétative contrôlée et en spirale

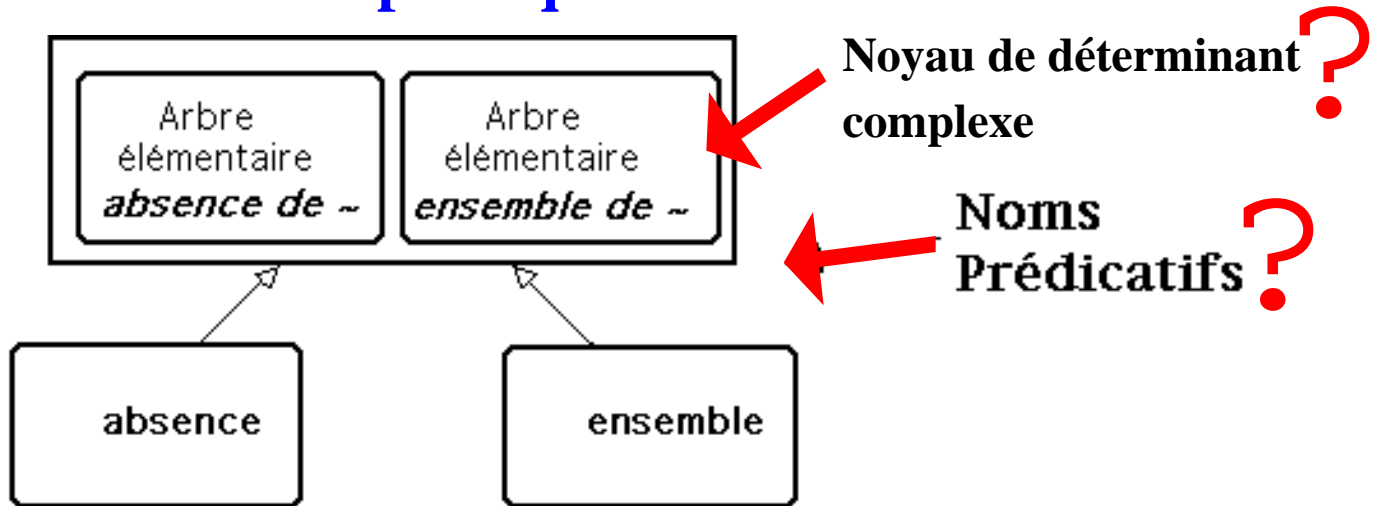
○ Une construction inductive

- 1. Extraction d'arbres sur corpus
- 2. Représentation des mots et des arbres
- 3. Un premier classement

○ Créer des pôles multiples de comportements partagés, les abstraire et les organiser

Le retour du linguiste

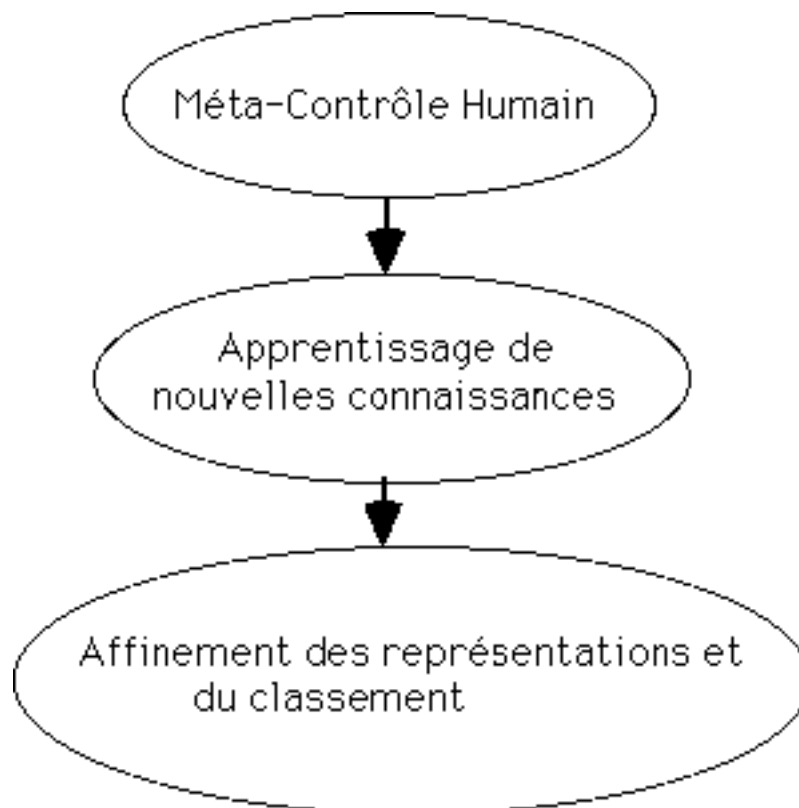
- **Interpréter les regroupements : intervention manuelle pour qualifier les résultats**



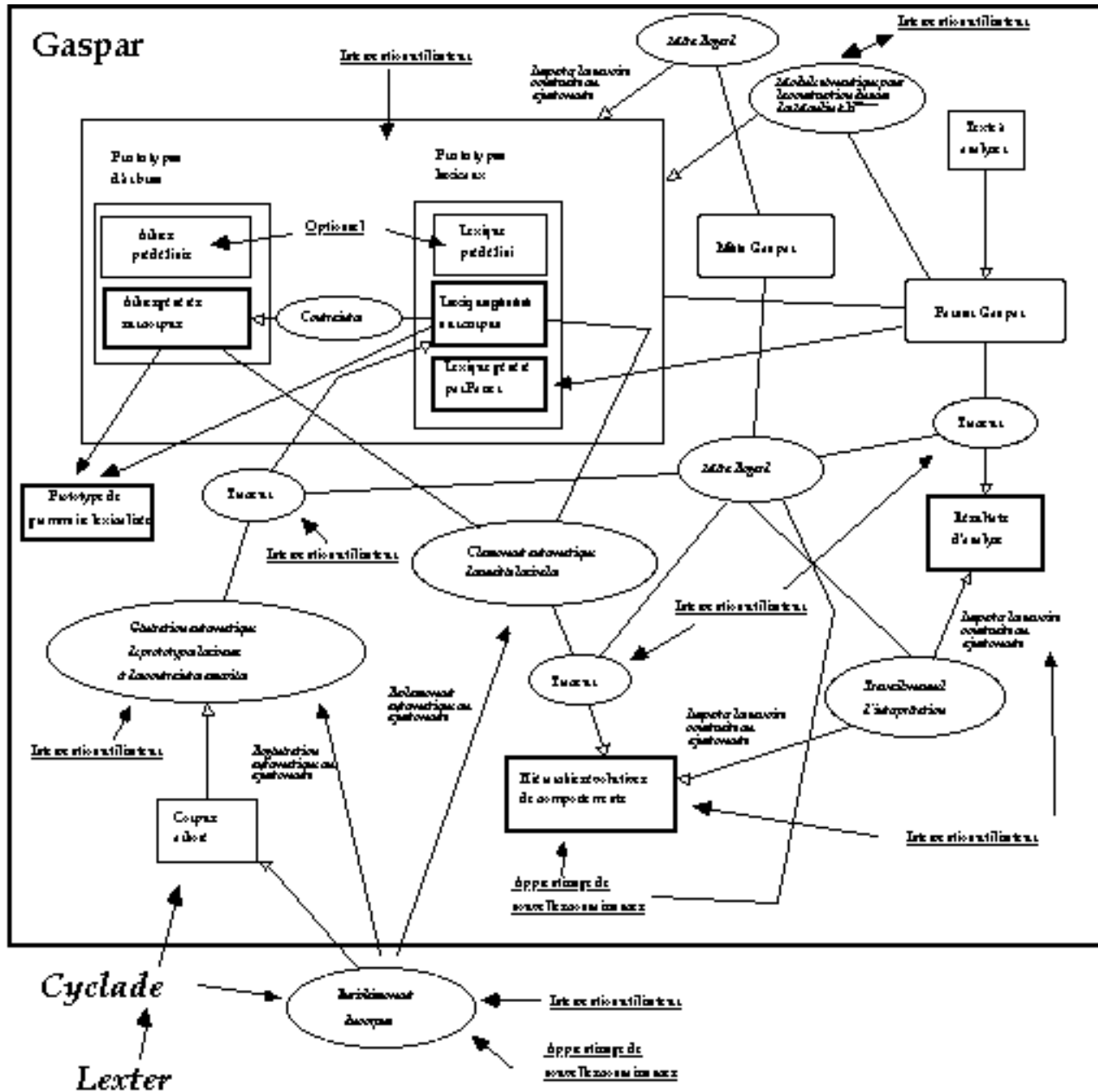
- **Projeter les résultats intermédiaires, réitérer la représentation en affinant les objets à construire**
- **L'utilisateur constitue à lui tout seul une "entité" de méta-contrôle indispensable pour évaluer les résultats construits**
 - C'est à lui d'interpréter et d'évaluer toutes les mises à jour réalisées
 - Il peut lui même être à l'origine de modifications sur les savoirs représentés

Du méta-regard à l'automodification et processus de contrôle

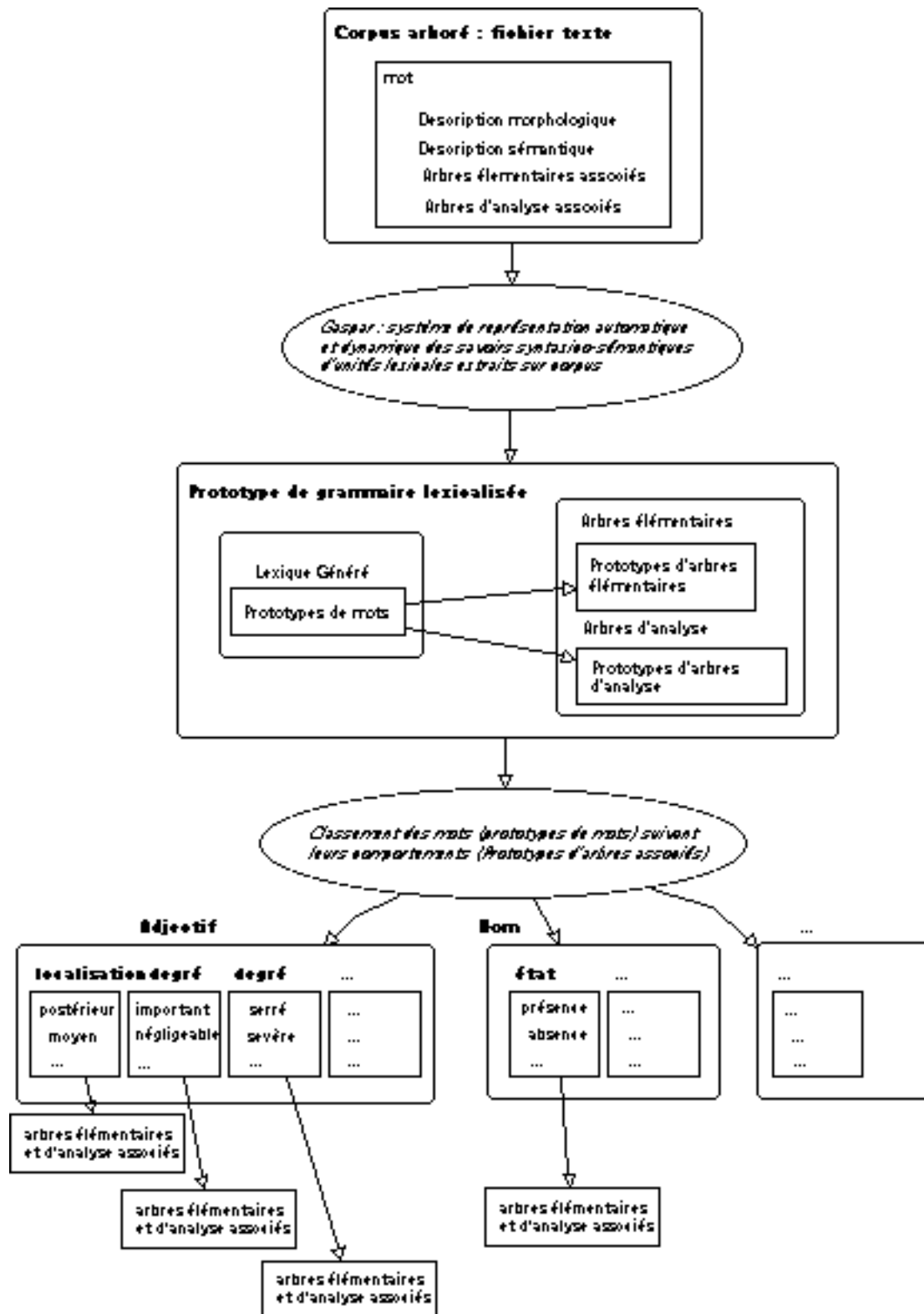
- Il est impératif de disposer de processus de traçage de toutes les opérations
 - Mise en place de processus de traçage qui présentent les différents états pertinents des traitements réalisés et des résultats construits à chaque étape
- **Méta-Contrôle Humain**



Gaspar : ce qui est construit



Gaspar : classements des mots

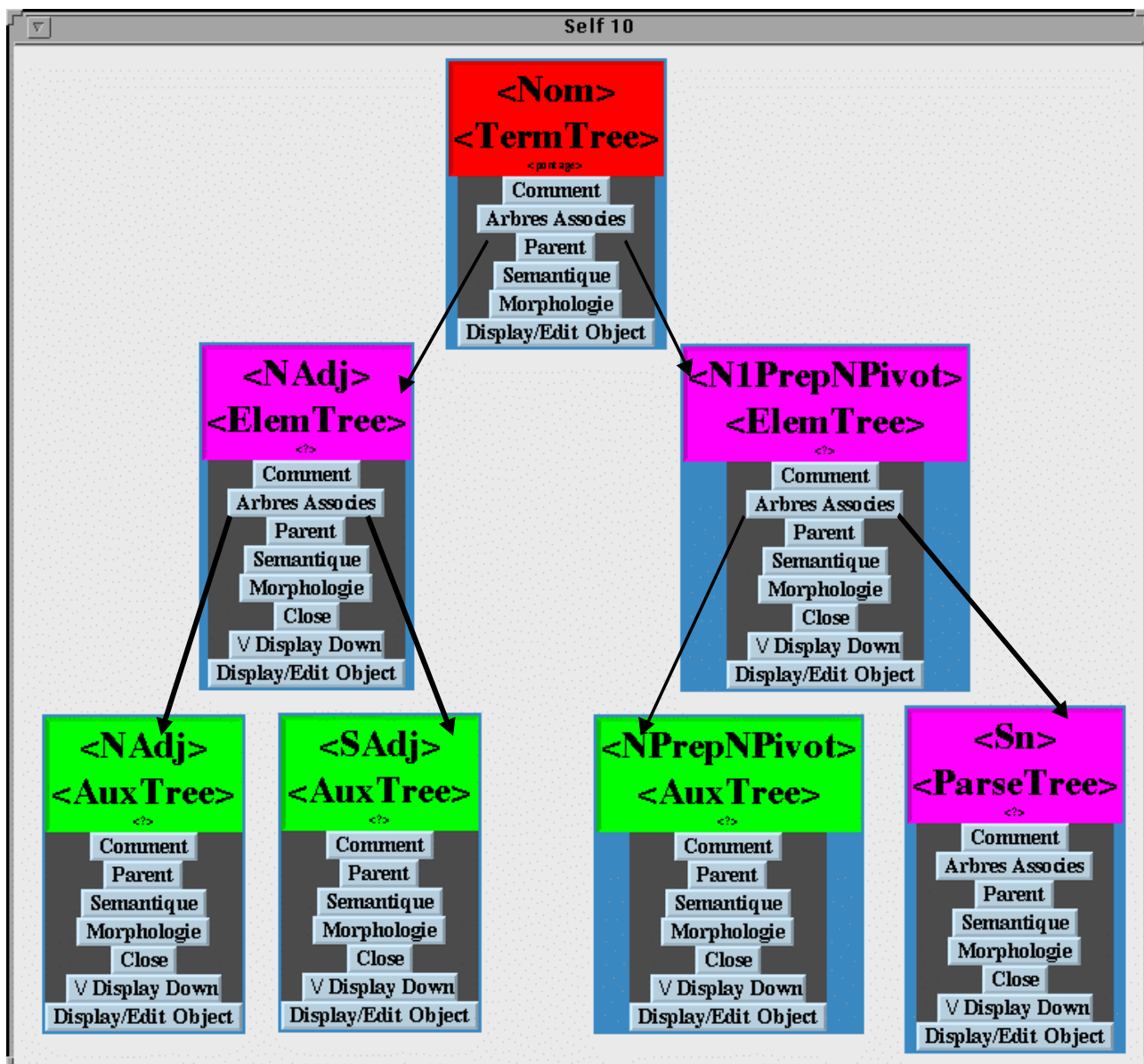


Gaspar : résultats construits

o Corpus de test

- 50 mots auxquels sont attachés une centaine d'arbres élémentaires et d'analyse

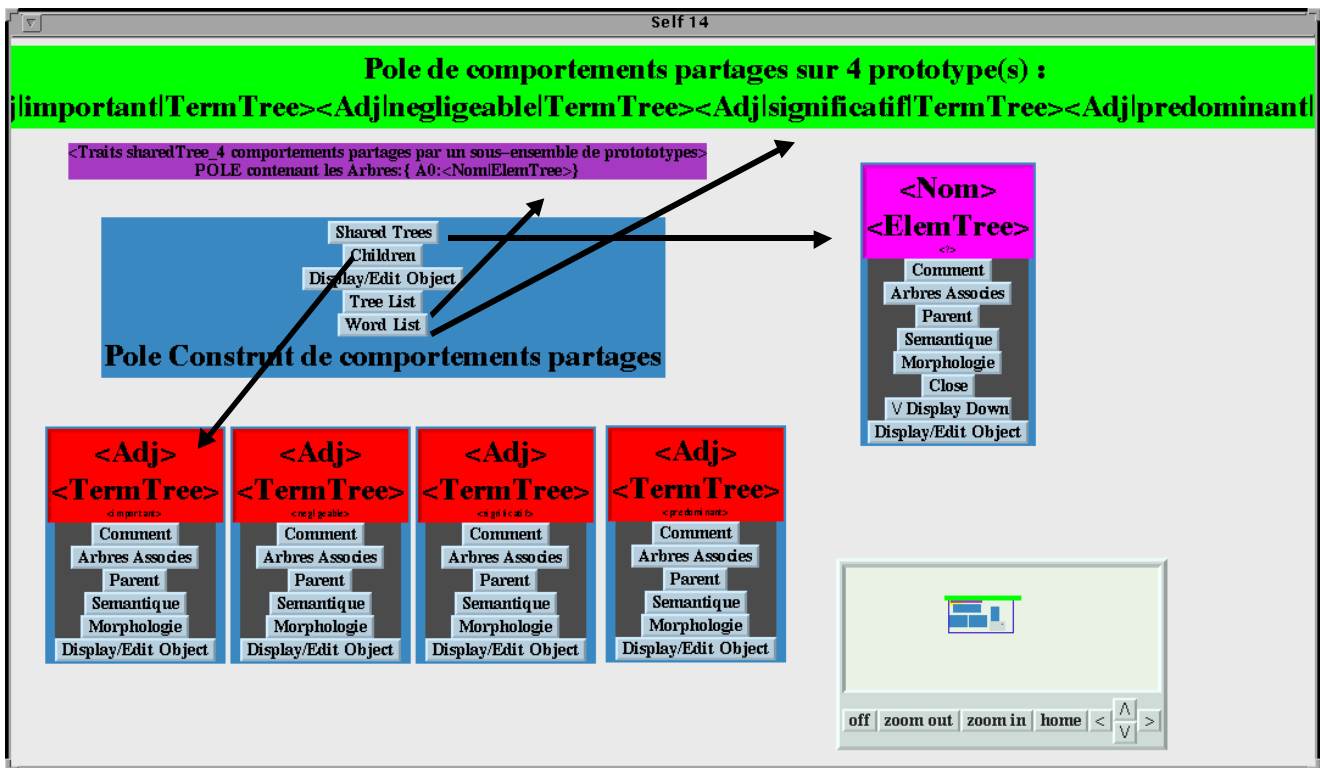
o Trace graphique des prototypes construits associés au mot pontage



- Trace graphique d'un pôle de comportements partagés construit sur notre corpus regroupant les adjectifs :

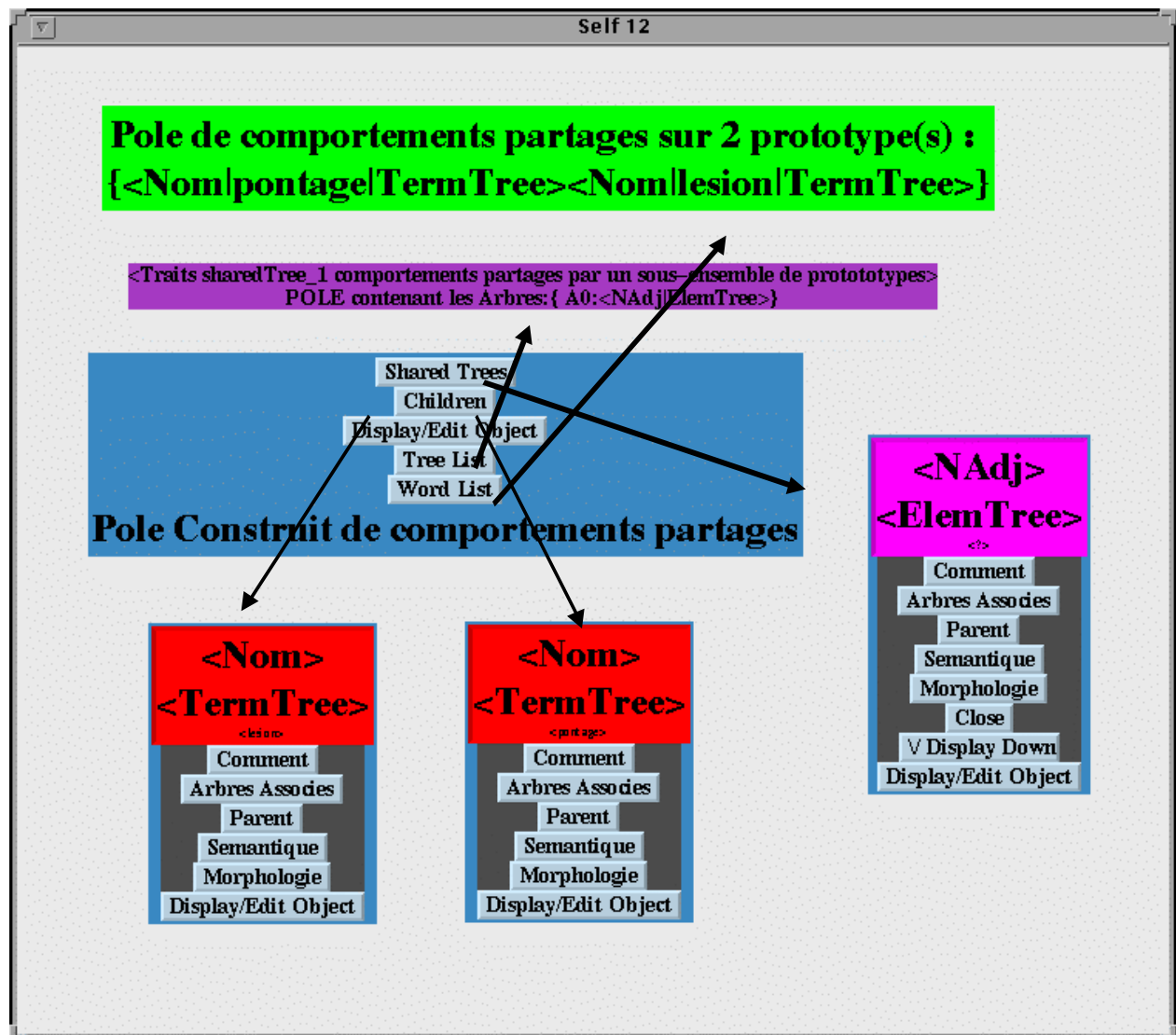
important negligeeable
significatif predominant

=> Classe de mots sémantiquement homogènes



○ Trace graphique d'un pôle de comportement construit sur deux mots :

« pontage » & « lesion »



Travail sur de « gros corpus »

○ Travail sur des séquences NAdj

○ Corpus via Lexter

- A partir de 8754 séquences nominales
- 586 mots (des noms) extraits
- 1413 arbres élémentaires associés de type :

Sn -> Nom Adj

Sn -> Adj Nom

Sn -> Adj XX

Sn -> XX Adj

Sn -> Pp(a/é/r) Nom

Sn -> Nom Pp(a/é/r)).

- Cette première sélection a donc consisté à ne garder que les arbres binaires portant les feuilles Nom/XX et Adj/Pp(a/é/r)

○ Corpus via AlethIP

- A partir de 7766 séquences nominales
- 718 mots (des noms) extraits
- 1680 arbres élémentaires associés de même type que précédemment

○ Résultats Lexter

- Création de prototypes de toutes pièces (une dizaine)
- Création de plus de 2000 prototypes par copie et ajustements
- Examen des partages de dépendances élémentaires : création de 56 pôles

Pôle n°1

Arbre élémentaire partagé associé à ce pôle : PAr(Sn) : Sn -> Nom
Adj(forme: gauche)

Liste des mots partageant ce type d'arbre : occipital bras
aisselle epaule

Pôle n°2

Arbre élémentaire partagé associé à ce pôle : PAr(Sn) : Sn -> Nom
Adj(forme: ponderal)

Liste des mots partageant ce type d'arbre : excès surcharge

Pôle n°3

Arbre élémentaire partagé associé à ce pôle : PAr(Sn) : Sn -> Nom
Adj(forme: dernier)

Liste des mots partageant ce type d'arbre : octobre juillet juin
mai mars avril

Pôle n°4

Arbre élémentaire partagé associé à ce pôle : PAr(Sn) : Sn -> Nom
Adj(forme: clinique)

Liste des mots partageant ce type d'arbre : besoin tableau

Pôle n°5

Arbre élémentaire partagé associé à ce pôle : PAr(Sn) : Sn -> Nom
Adj(forme: ventriculaire)

Liste des mots partageant ce type d'arbre : capture fonction
conduction

Résultats sur Lexter : « Fonction »

Self 10

off | zoom out | zoom in | home | < | ^ | v | >

Gaspar Object!

<SN>
<ElemTree>

- Comment
- Arbres Associes
- Parent
- Semantique
- Morphologie
- Close
- ∨ Display Down
- Display/Edit Object

<Nom>
<TermTree>
<FONCTION>

- Comment
- Arbres Associes
- Parent
- Semantique
- Morphologie
- Display/Edit Object

Display/Edit Object

Comment

Semantique:<Nom|FONCTION|TermTree>

<Classe>
(classe: ?)

►a list('Adj', '{forme:=VENTRICULAIRE}')

Shared Trees

- Children
- Display/Edit Object
- Tree List
- Word List

Objets Traits <Pole de comportements partages>

Shared Trees

- Children
- Display/Edit Object
- Tree List
- Word List

Objets Traits <Pole de comportements partages>

Shared Trees

- Children
- Display/Edit Object
- Tree List
- Word List

Objets Traits <Pole de comportements partages>

Display/Edit Object

Comment

Morphologie:<Nom|FONCTION|TermTree>

<Accord>

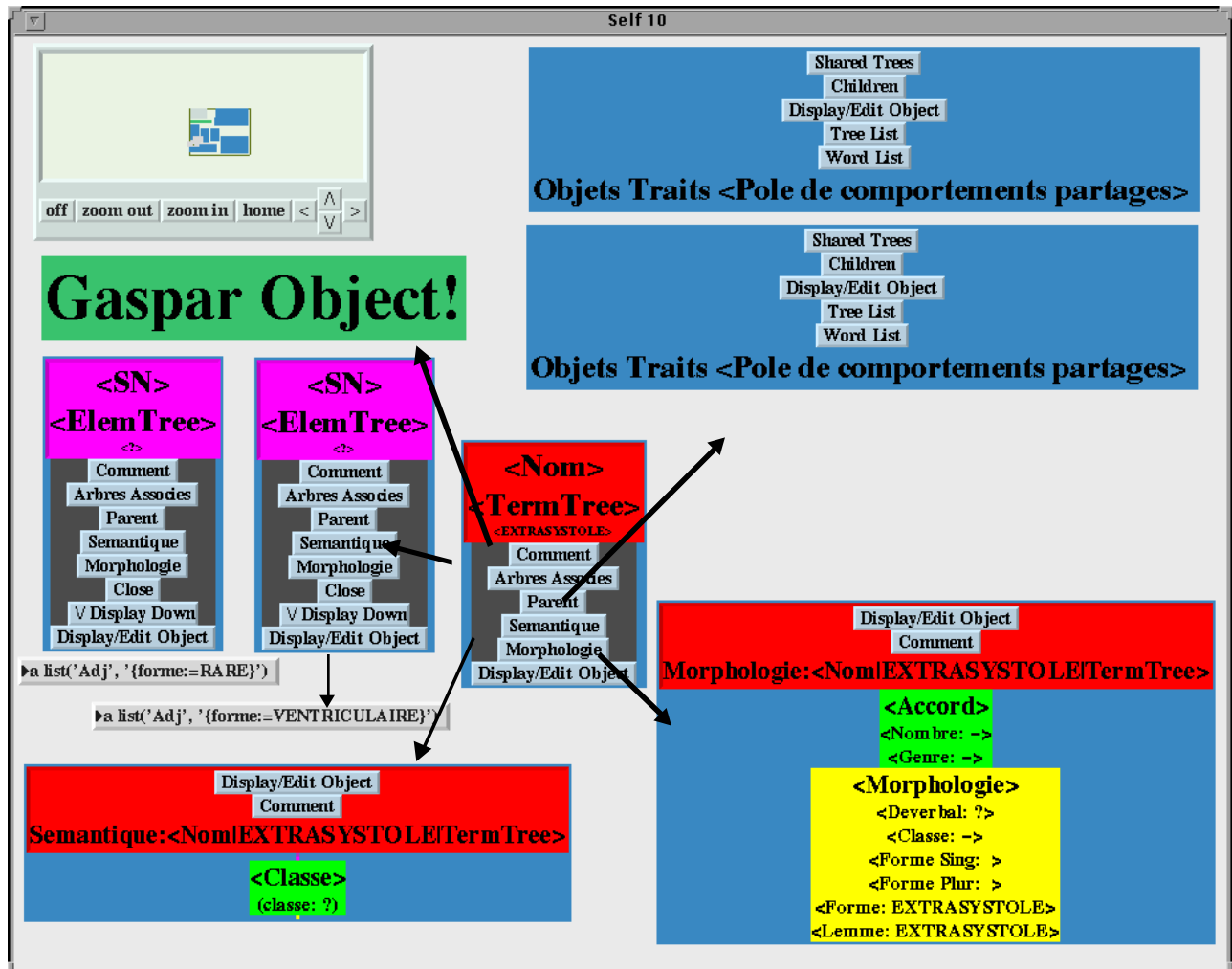
<Nombre: ->

<Genre: ->

<Morphologie>

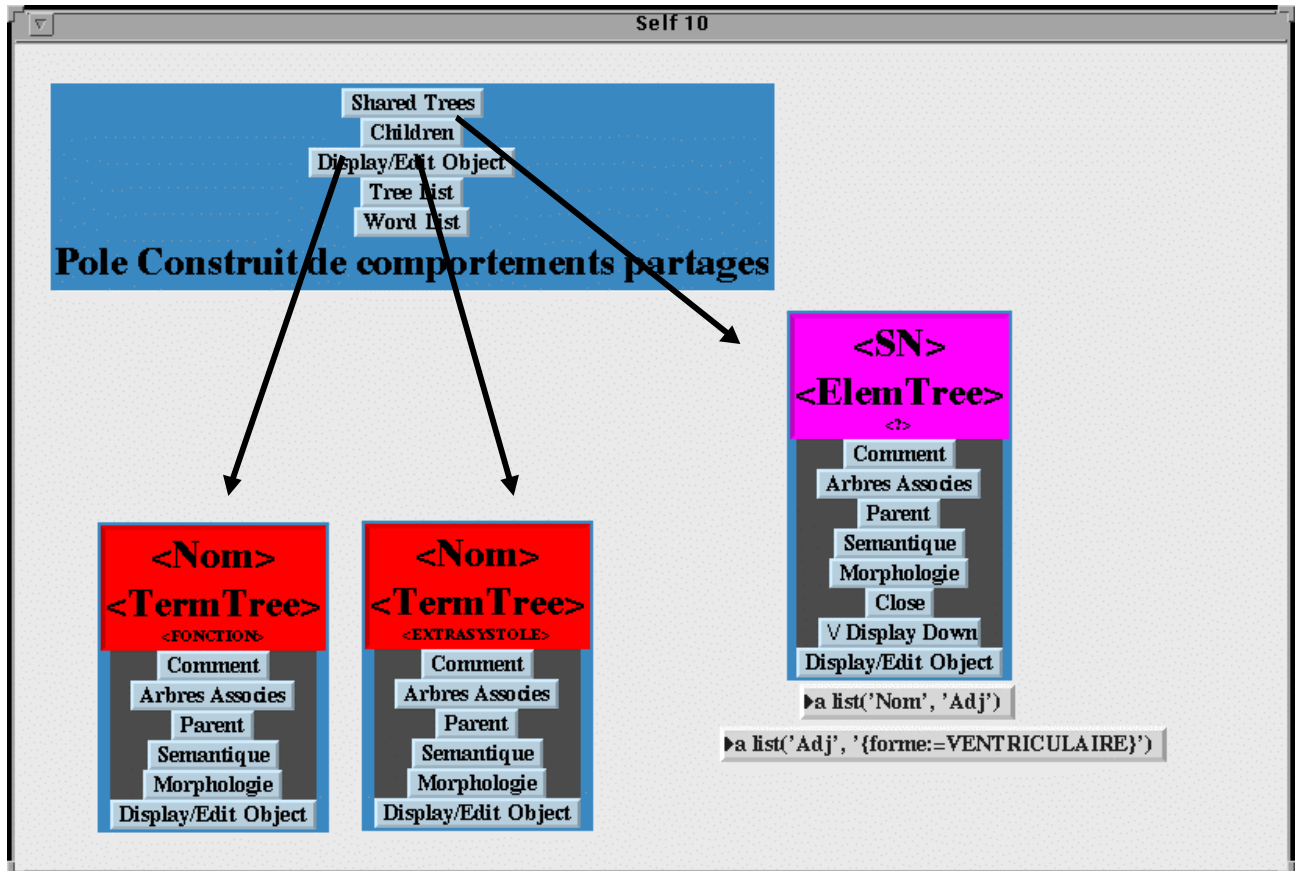
- <Deverbal: ?>
- <Classe: ->
- <Forme Sing: >
- <Forme Plur: >
- <Forme: FONCTION>
- <Lemme: FONCTION>

Résultats sur Lexter : « Extrasystole »



Résultats sur Lexter : Partage

« Extrasystole » & « Fonction »



○ Résultats AlethIP

- Création de prototypes de toutes pièces (une dizaine)
- Création de plus de 2500 prototypes par copie et ajustements
- Examen des partages de dépendances élémentaires : création de 57 pôles

Pôle n°1

Arbre élémentaire partagé associé à ce Pôle : PAr(Sn) -> Sn Nom
Adj(Forme: renal)

Liste des mots partageant ce type d'arbre : FONCTION,
ARTERIOGRAPHIE

Pôle n°2

Arbre élémentaire partagé associé à ce Pôle : PAr(Sn) -> Sn Nom
Adj(Forme: pulmonaire)

Liste des mots partageant ce type d'arbre : RESISTANCE, CHAMP,
COEUR, SOMMET, PARENCHYME, SURINFECTION, PLAGE,
TUBERCULOSE, VASCULARISATION

Pôle n°3

Arbre élémentaire partagé associé à ce Pôle : PAr(Sn) -> Sn Nom
Adj(Forme: gauche)

Liste des mots partageant ce type d'arbre : LOBECTOMIE, EPAULE,
SUPERIEUR, COUDE, GALOP, AISSELLE, BORD, INTERNE

Pôle n°4

Arbre élémentaire partagé associé à ce Pôle : PAr(Sn) -> Sn Nom
Adj(Forme: bon)

Liste des mots partageant ce type d'arbre : PRONOSTIC,
FONCTIONNEMENT, SUIVI, FRACTION

Pôle n°5

Arbre élémentaire partagé associé à ce Pôle : PAr(Sn) -> Sn Nom
Adj(Forme: nouveau)

Liste des mots partageant ce type d'arbre : REDUCTION,
RECOMMANDATION, MOLECULE

Résultats sur AlethIP : « Fonction »

Self 10

off zoom out zoom in home < >

Gaspar Object!

Shared Trees
Children
Display/Edit Object
Tree List
Word List

Objets Traits <Pole de comportements partagés>

Shared Trees
Children
Display/Edit Object
Tree List
Word List

Objets Traits <Pole de comportements partagés>

Display/Edit Object
Comment
Morphologie: <Nom|FONCTION|TermTree>

<Accord>
<Nombre: ->
<Genre: ->

<Morphologie>
<Deverbal: ?>
<Classe: ->
<Forme Sing: >
<Forme Plur: >
<Forme: FONCTION>
<Lemme: FONCTION>

<SN>
<ElemTree>
<FONCTION>
Comment
Arbres Associes
Parent
Semantique
Morphologie
Close
V Display Down
Display/Edit Object
list('Adj', {forme:=RENAL})

<Nom>
<TermTree>
<FONCTION>
Comment
Arbres Associes
Parent
Semantique
Morphologie
Display/Edit Object

Display/Edit Object
Comment
Semantique: <Nom|FONCTION|TermTree>

<Classe>
(classe: ?)

Résultats sur AlethIP : « Extrasystole »

Self 10

off zoom out zoom in home < ^ v >

Gaspar Object!

Shared Trees
Children
Display/Edit Object
Tree List
Word List
Objets Traits <Pole de comportements partagés>

Shared Trees
Children
Display/Edit Object
Tree List
Word List
Objets Traits <Pole de comportements partagés>

Shared Trees
Children
Display/Edit Object
Tree List
Word List
Objets Traits <Pole de comportements partagés>

<SN>
<Elem Tree>
Comment
Arbres Associes
Parent
Semantique
Morphologie
Close
v Display Down
Display/Edit Object
▶ a list('Adj', '{forme:=ISOLE}')

<SN>
<Elem Tree>
Comment
Arbres Associes
Parent
Semantique
Morphologie
Close
v Display Down
Display/Edit Object
▶ a list('Adj', '{forme:=VENTRICULAIRE}')

<Nom>
<Term Tree>
<EXTRASYSTOLE>
Comment
Arbres Associes
Parent
Semantique
Morphologie
Display/Edit Object

Display/Edit Object
Comment
Morphologie: <Nom|EXTRASYSTOLE|TermTree>
<Accord>
<Nombre: ->
<Genre: ->
<Morphologie>
<Deverbal: ?>
<Classe: ->
<Forme Sing: v>
<Forme Plur: v>
<Forme: EXTRASYSTOLE>
<Lemme: EXTRASYSTOLE>

Display/Edit Object
Comment
Semantique: <Nom|EXTRASYSTOLE|TermTree>
<Classe>
(classe: ?)

Le travail sur les gros corpus et les limites de Self

- **Lenteur des traitements : plus de 30 heures de traitement sur corpus Lexter**
- **Self et le système sous-jacent manquent de ressources pour manipuler une grande quantité d'objets**
- **Les processus sont prêts : seule la "mémoire" leur manque pour exprimer et produire des pistes de sens**

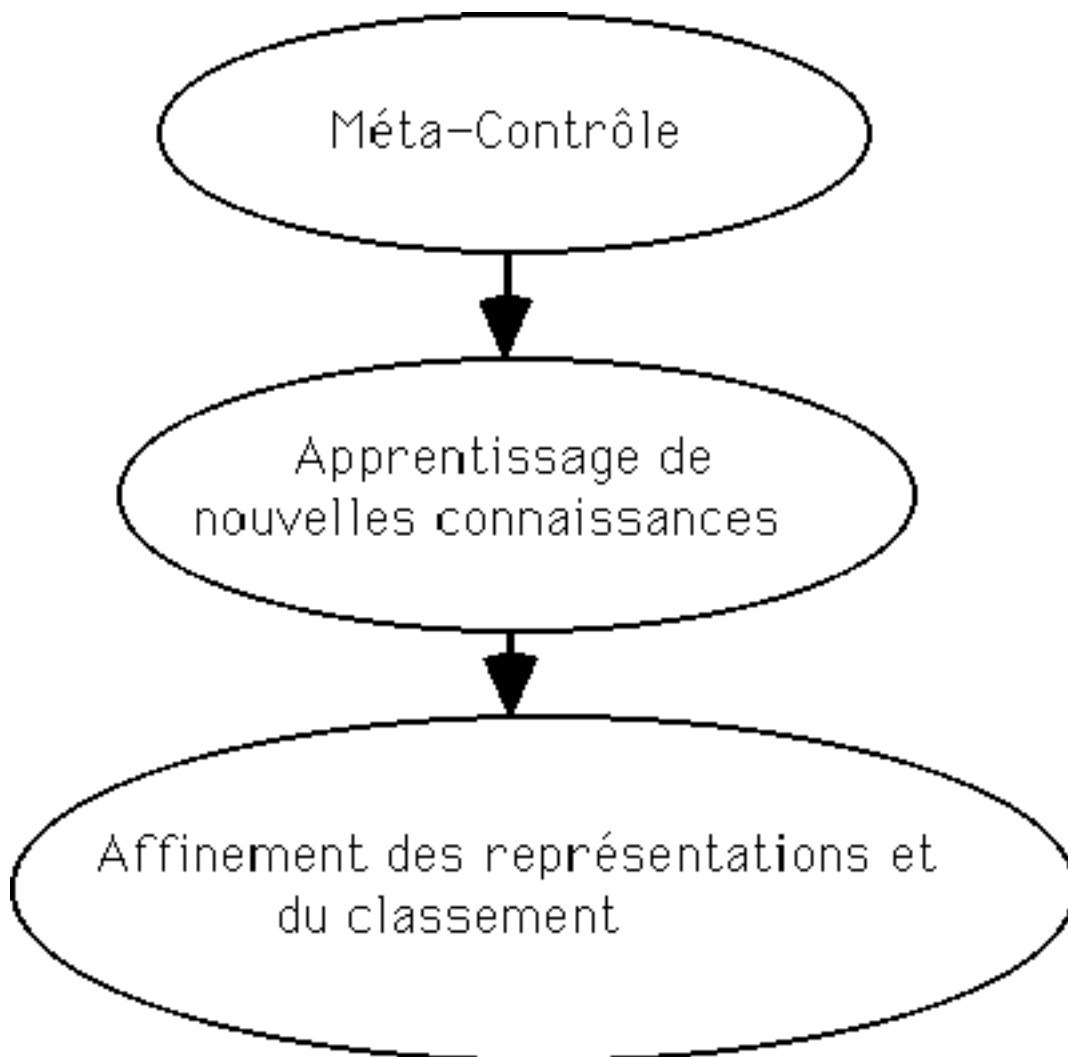
"Prolog est né d'un pari : créer un langage de très haut niveau, même inefficace au sens des informaticiens de l'époque. L'efficacité consistait alors à faire exécuter très rapidement par une machine des programmes écrits laborieusement. Le pari était donc de pouvoir écrire rapidement des programmes, quitte à ce que la machine les exécute laborieusement." Colmerauer, préface à Giannesini & al. "Prolog", InterEditions, 1985.

Conclusion

- **La PàP : une solution satisfaisante pour la mise en place de processus de représentation capables de rendre compte des aspects mouvants du domaine décrit**
- **Une approche inductive**
 - Représentations à partir de savoirs extraits d'un corpus
 - Projections des résultats
 - Ajustements des représentations
 - Ajustement du classement
- **Classer les prototypes suivant leurs comportements linguistiques**
 - Arbres élémentaires et arbres d'analyse
- **Construire une grammaire d'arbres**
 - Approche d'une grammaire de sous-langage
- **Phases intermédiaires de travail manuel (Mikheev & Moens 1994)**

Perspectives

- Un méta-niveau pour le contrôle des représentations et des héritages mis en place
 - Des outils de **contrôle** et de **traçage** pour évaluer les résultats construits



Mettre en place la génération de nouveaux savoirs

- **Création automatique de pôles de comportements**
- **Affiner les processus autodescriptifs des objets**
- **Des règles pour la génération des nouveaux objets**
 - Méta-Règle n°1 :
Si un prototype d'arbre a x composants
S'il n'existe pas de moule d'arbre à x composants
Alors créer ce moule en s'appuyant sur ce prototype
 - Méta-Règle n°2 :
Si un Objet Ae (Arbre élémentaire) a x composants.
S'il n'existe pas de moule d'arbre à x composants.
Alors créer ce moule en s'appuyant sur Ae.
- **Créer un dictionnaire d'erreurs**