Linx

Un système de génération d'anaphores dans les textes d'assemblage

Leïla Kosseim, Guy Lapalme

Citer ce document / Cite this document :

Kosseim Leïla, Lapalme Guy. Un système de génération d'anaphores dans les textes d'assemblage. In: Linx, hors-série n°4, 1991. Texte et ordinateur. Les Mutations du Lire-Ecrire. pp. 215-232;

doi : 10.3406/linx.1991.1199

http://www.persee.fr/doc/linx_0246-8743_1991_hos_4_1_1199

Document généré le 01/06/2016

Leila Kosseim et Guy Lapalme
Université de Montréal

Un système de génération d'anaphores dans les textes d'assemblage

AFIN DE PRODUIRE un texte de qualité, un écrivain doit constamment prendre des décisions, par exemple, quelle information inclure dans le texte et quelle perspective choisir, quelle structure textuelle et quel ton utiliser, comment atteindre une cohésion textuelle, quels mots choisir... Dans un générateur automatique, la prise de ces décisions est assez complexe; c'est pourquoi les travaux de recherche se spécialisent généralement dans un domaine particulier de la production.

Persée creative commons

1 Introduction

Afin de produire un texte de qualité, un écrivain doit constamment prendre des décisions, par exemple, quelle information inclure dans le texte et quelle perspective choisir, quelle structure textuelle et quel ton utiliser, comment atteindre une cohésion textuelle, quels mots choisir... Dans un générateur automatique, la prise de ces décisions est assez complexe; c'est pourquoi les travaux de recherche se spécialisent généralement dans un domaine particulier de la production.

Le travail présenté ici porte sur le choix de procédés anaphoriques. Les procédés anaphoriques sont des procédés linguistiques utilisés principalement pour atteindre une cohésion textuelle [Tut92a, Tut92b]. En génération automatique de texte, l'introduction d'anaphores permet de produire des textes plus naturels. Par exemple, le texte :

(1) Marie est partie en vacances. Elle a d'abord visité l'Espagne seule, ensuite elle est allée en France avec sa cousine.

est plus naturel que

(2) *Marie* est partie en vacances. *Marie* a d'abord visité l'Espagne seule, ensuite *Marie* est allée en France avec sa cousine.

Le système informatique présenté dans cet article est basé sur l'étude linguistique de Tutin [Tut92a, Tut92b] qui décrit un ensemble de contraintes régissant l'introduction naturelle d'anaphores grammaticales et lexicales dans les textes d'assemblage. Il constitue une composante

tactique : il ne traite pas la planification du texte (détermination du message à transmettre), mais plutôt la réalisation linguistique, c'est-à-dire comment exprimer ce message en langue naturelle [McK85, DMZ90, Dan91].

La section 2 présente les aspects linguistiques sur lesquels le système est basé. Les anaphores traitées sont décrites, mais les contraintes régissant leur introduction ne sont qu'indirectement présentées à la section 4 (pour plus de détails linguistiques, se référer à [Tut92a, Tut92b]). Le reste de l'article décrit le système informatique. À la section 3, nous montrons une particularité des textes d'assemblage : le dynamisme des objets et sa représentation à l'aide du dictionnaire de concepts. Les étapes de la génération et en particulier la réalisation des procédés anaphoriques sont ensuite détaillées à la section 4 à l'aide d'un exemple de texte généré par le système.

2 Aspects linguistiques

En utilisant les termes de Tutin [Tut92a, Tut92b], on peut définir une anaphore comme suit :

Étant donné deux éléments textuels A et B, B est une anaphore de A si B est situé après A dans la proposition et si l'interprétation référentielle et/ou sémantique de B dépend de celle de A. On appelle alors A l'antécédent immédiat de B.

Il existe une interprétation référentielle entre deux éléments textuels s'ils ont le même référent. Il y a interprétation sémantique entre deux éléments textuels s'ils sont reliés par leur sens. Il existe plusieurs procédés anaphoriques ; cependant, tous ne sont pas toujours appropriés. En vérifiant les contraintes d'introduction d'anaphores développées par Tutin [Tut92a, Tut92b], le système produit de façon naturelle les anaphores grammaticales et lexicales suivantes :

• Les anaphores grammaticales :

Les anaphores grammaticales utilisent des lexèmes de classes fermées (de déterminants ou de pronoms) ou utilisent l'absence de marque linguistique [HH76].

– L'ellipse du complément d'objet direct (C.O.D.) :

L'ellipse consiste en une absence de marque linguistique. Il s'agit d'un procédé co-référentiel, car le C.O.D. déjà introduit et l'élément linguistique vide ont le même référent. Dans

(3) Mélanger *la sauce et le bouillon*, chauffer Ø[1] et servir Ø.

l'antécédent immédiat de Ø est *la sauce et le bouillon*..

- La pronominalisation :

La pronominalisation, par exemple l'emploi du *il* dans :

(4) Le client est arrivé à l'avance. *Il* attend dans la salle.

---

1 Le symbole Ø sert à désigner l'ellipse.

est le procédé le plus étudié en génération automatique. On retrouve par exemple les travaux informatiques de Dale [Dal92] et Danlos [Dan91] et l'étude linguistique de Namer [Nam89].

• Les anaphores lexicales :

Les anaphores lexicales produisent un groupe nominal défini et mettent en jeu des lexèmes de classes lexicales ouvertes (les adjectifs, les noms et les adverbes) [HH76].

– Le dérivé sémantique de type résultat :

Ce type de dérivé sémantique (un des plus fréquents dans les textes d'assemblage) permet de relier des éléments textuels par l'intermédiaire de la relation *action-résultat typique*. Par exemple,

(5) *Battre* les œufs et le sucre. Ajouter le reste des ingrédients *au mélange.*

Les relations sémantiques de type résultat correspondent à la fonction lexicale S_{res} du Dictionnaire Explicatif et Combinatoire (DEC) de Mel'cuk [Mel84].

– L'hyperonymation :

L'hyperonymation permet de faire référence à plusieurs objets (concrets ou abstraits) venant d'une même classe générique, à l'aide d'un terme décrivant cette classe. Par exemple,

(6) Préparer la viande, *les pommes de terre et les carottes* pour la cuisson. Ensuite, faire bouillir *les légumes.*

– La dénomination de base :

La dénomination de base utilise un terme générique, comme l'hyperonymation, mais pour identifier un seul objet. Elle consiste à introduire un objet par un **nom de base**, c'est-à-dire, le terme le plus naturel plutôt que le plus spécifique [Rei90b, Rei90a]. Dans

(7) Juliette a acheté *une Porsche. La voiture* est encore chez le concessionnaire.

la voiture est une dénomination de base de *une Porsche.*

– La répétition partielle :

La répétition partielle s'applique aux référents décrits par un groupe nominal composé non-figé. Ce procédé consiste à utiliser ce même groupe nominal, auquel un certain nombre de qualificatifs ont été enlevés. Le groupe nominal résultant est donc une abréviation de l'original. Par exemple,

(8) Éplucher *3 petites pommes.* Ajouter *les pommes* au reste des fruits.

– La répétition exacte :

La répétition exacte consiste à introduire le référent par un groupe nominal défini en répétant sa description complète. Par exemple,

(9) Nadia parlait à *un jeune skieur* au chalet. *Le jeune skieur* était un champion mondial.

– La coordination complexe :

Tout comme l'hyperonymation, ce procédé permet de faire référence à plusieurs objets. Cette fois-ci, différentes anaphores lexicales sont

appliquées sur des sous-ensembles de ces objets. Les résultats sont ensuite combinés en un seul groupe nominal. Par exemple, dans

(10) Jean est revenu de la bibliothèque avec *un roman policier, un dictionnaire et deux disques compacts. Les livres et les disques* doivent être rendus la semaine prochaine.

il y a coordination d'une hyperonymation (*les livres*) et d'une répétition partielle (*les disques*).

D'autres travaux ont aussi traité la génération automatique d'anaphores (ou expressions référentielles). Parmi ceux-ci, le plus connu et plus près du nôtre est le système Epicure de Dale [Dal92].

Le système Epicure est un système qui génère des expressions référentielles dans les recettes de cuisines. Ce système étant très près du nôtre, nous en discutons brièvement ici. Pour introduire une expression référentielle, Epicure essaie d'abord d'utiliser un pronom. La pronominalisation est basée sur la distinction entre le focus local et le focus global du texte. Le focus local des propositions précédentes ainsi que la pile d'espace de focus de Grosz [GS86] sont pris en considération pour déterminer si un pronom est adéquat. Comme décrit plus loin, notre système tient aussi compte du focus local des propositions précédentes, mais pas de la pile d'espace de focus.

Si un pronom n'est pas adéquat, un groupe nominal défini est produit en utilisant une « description minimale permettant de distinguer ». Epicure génère, parmi les groupes nominaux définis, des anaphores en « one », comme dans :

(11) Cut *the white onions* into rings and chop *the red ones.*

Notre système diffère de celui de Dale, car il est axé vers l'introduction de l'anaphore la plus naturelle et non l'introduction d'une « description minimale permettant de distinguer ». Aussi, notre système génère le texte en français, par conséquent l'introduction des anaphores est régie par des règles différentes et de plus, les anaphores traitées sont celles du français. Les anaphores dans ces deux langues sont différentes ; par exemple, il n'existe pas d'anaphores en « one » en français. L'équivalent de 11 est

(12) Couper *les oignons blancs* en rondelles et hacher *les Ø rouges.*

qui contient une ellipse.

3 Le dictionnaire de concepts

3.1 L'utilité du dictionnaire de concepts

Dans les textes d'assemblage, on retrouve trois entités : l'**agent** qui manipule les **objets** par l'intermédiaire d'**instruments**. Nous appelons ces trois entités les **participants** de l'assemblage. Dans les recettes de cuisine, l'agent (le cuisinier) est implicitement identifié au lecteur du texte, les objets sont les ingrédients et les instruments sont les ustensiles nécessaires

à la recette. Une importante caractéristique des textes d'assemblage est l'aspect dynamique des objets. En effet, on distingue trois types de dynamisme :

• Le changement d'état, comme dans :

(13) Faire fondre le beurre.[1]

Dans cet exemple, le beurre change de l'état solide à l'état liquide. Le choix des termes que le système utilisera une fois que le beurre est liquide sera différent que lorsqu'il était solide. En effet, il serait maintenant préférable de générer :

(14) *Verser* le beurre sur le fromage en crème.

plutôt que

(15) *Mettre* le beurre sur le fromage en crème.

Le choix du verbe *verser* dépend de l'état courant de l'ingrédient à verser. Le générateur doit donc tenir compte de l'état des objets, pour produire les termes les plus appropriés.

• La création et la destruction d'objets, comme dans :

(16) Séparer le jaune et le blanc des œufs.

Ici, les œufs sont détruits, et deux nouveaux ingrédients, les blancs et les jaunes, sont créés dynamiquement. Une référence aux œufs dans la suite du texte serait maintenant ambiguë.

• La création et la destruction de perspectives, comme dans

(17) Couper la viande.

Après l'énonciation de cette phrase, la viande peut être vue sous une nouvelle perspective. On peut donc y référer par sa matière, *la viande*, ou par sa nouvelle forme *les morceaux de viande*.

Pour produire un texte de qualité, un générateur de textes d'assemblage doit donc tenir compte de ces trois types de dynamisme des objets. Pour ce faire, notre système tient à jour une liste des objets et leur état courant à un instant donné de la recette. Cette liste est emmagasinée dans la structure *État*. Après la génération de chaque élément d'une proposition, *État* est mis à jour pour refléter les changements d'état, les créations et destructions d'objets et de perspectives qui se sont produits. Pendant la production de chaque proposition, cette structure est consultée pour pouvoir choisir les termes les plus appropriés à l'état courant des objets (par exemple le choix de *verser* plutôt que *mettre*). Notons que *État* est équivalent au « Working Set » de Epicure, une structure représentant l'état courant des objets. Cependant, *État* est moins détaillé que le « Working Set ». Notre système comprend deux dictionnaires distincts : un lexique et un dictionnaire de concepts[2]. Le lexique contient l'information des entrées lexicales (les mots) particulières à la langue cible (ici, le français), tandis

---

1 Les exemples sont tirés du domaine des recettes de cuisine, mais des exemples équivalents peuvent être trouvés dans d'autres textes d'assemblage.

2 Comme le souligne Zock dans [Zoc91], un concept peut correspondre à plusieurs réalisations lexicales, et inversement.

que le dictionnaire de concepts sert à représenter le dynamisme des objets. Ce dernier dictionnaire est formé d'une taxonomie de deux classes de concepts : les **actions**, et les **participants**, tout comme dans le « Upper Model » de Penman [Man83] qui comprend les classes d'actions et d'objets. Les classes sont décrites par des propriétés, et sont organisées de façon hiérarchique où les propriétés d'une classe sont héritées par ses sous-classes. Les propriétés décrivant une action indiquent l'état que certains participants doivent satisfaire avant et après que l'action ait lieu, tandis que les propriétés des participants indiquent leur état de défaut.

Une partie des taxonomies d'actions et de participants culinaires est illustrée à la figure 1 (page de droite). Aux feuilles de la hiérarchie des participants se trouvent des instances particulières de ces concepts : les participants du monde réel. Par convention, un terme désignant un concept se termine par « _c » (par exemple `lapin_c`), et une instance par un numéro (par exemple `lapin1`).

3.2 Les propriétés des actions

Les propriétés des actions sont de deux types : les préconditions et les postconditions.

• Les préconditions définissent les conditions que les participants doivent satisfaire pour qu'une action puisse avoir lieu. Par exemple, pour verser (le concept `verser_c`) un ingrédient, celui-ci doit être liquide, tandis que pour l'étendre (`étendre_c`), l'ingrédient doit être crémeux. Les préconditions d'une action sont hérités des ancêtres de l'action, par exemple pour hacher (`hacher_c`) ou trancher (`trancher_c`) un ingrédient, celui-ci doit être solide : une précondition héritée de l'action `couper_c`.

• Les postconditions indiquent les résultats de l'action. Ces résultats représentent le dynamisme des objets en indiquant les changements d'état qu'une action implique sur un objet, les créations et les destructions d'objets et de perspectives. Par exemple, après l'application de l'action `tailler_c` sur un ingrédient, il y a création de perspective car l'ingrédient est maintenant sous la forme de *morceaux* : une postcondition héritée de l'action `couper_c`. De son côté, l'action `hacher_c` redéfinit la postcondition de son parent, car son résultat typique est un *hachis*, plutôt que des *morceaux*.

3.3 Les propriétés des participants

Les propriétés des participants spécifient leur état de défaut. Toute modification de l'état standard d'un participant particulier est reflétée dans la structure dynamique *État*. Par exemple, le dictionnaire conceptuel spécifie que le concept `beurre_c` par défaut est sous l'état solide. Si après

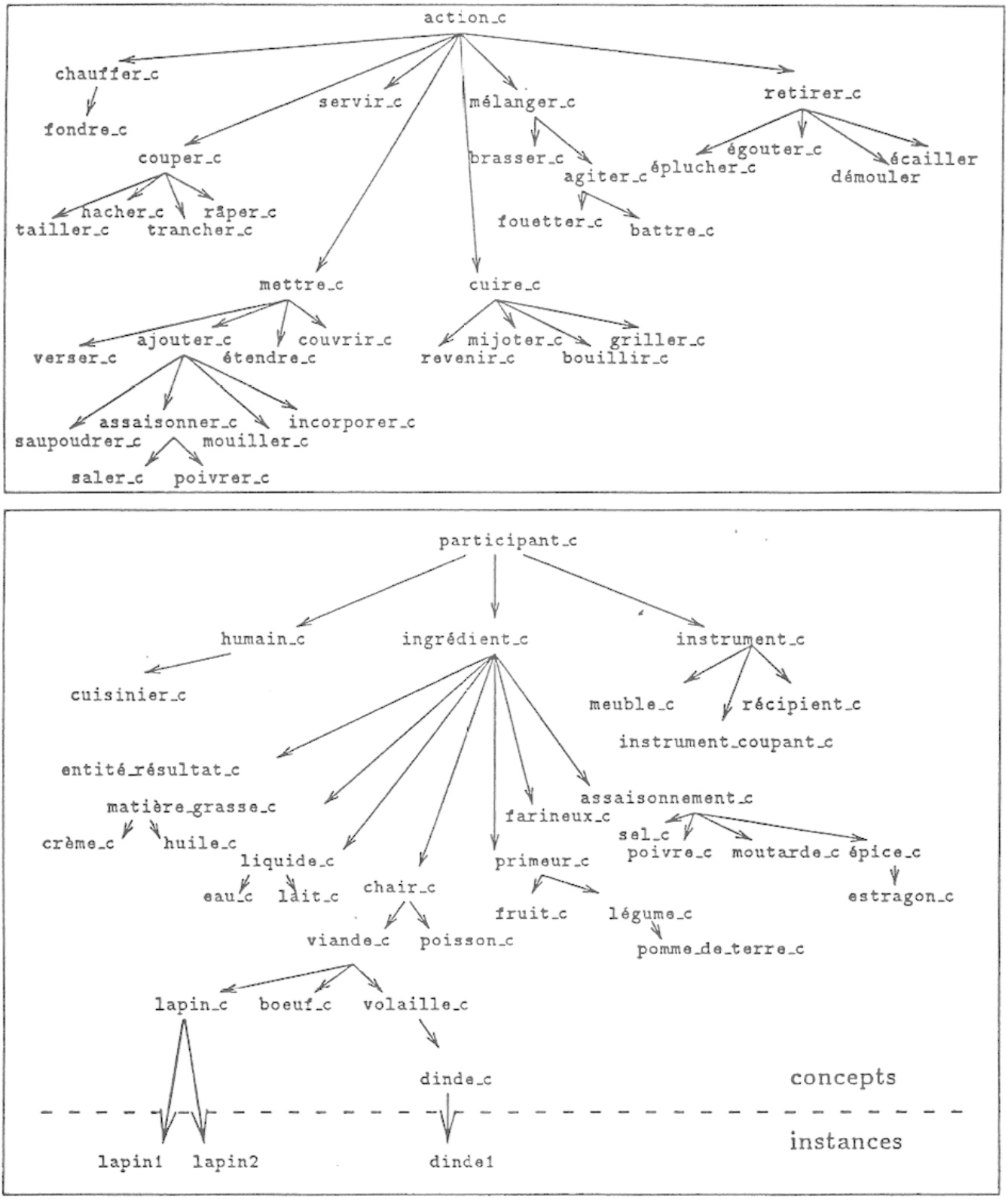

Figure 1 : Taxonomies d'actions et de participants culinaires

manipulations, `beurre1`, une instance particulière de `beurre_c`, change d'état, ceci est emmagasiné dans la structure *État*. Les propriétés sont aussi héritées du participant parent. Par exemple, tous les sous-concepts de `chair_c` héritent de l'état solide de ce concept.

3.4 Les avantages du dictionnaire conceptuel

Les principaux avantages qu'apporte le dictionnaire conceptuel à notre système sont les suivants :

• Il permet de représenter le dynamisme des objets et donc d'effectuer des inférences simples sur les connaissances du monde. Dans notre système, ceci est possible grâce aux post- et préconditions des actions et l'état de défaut des participants qui y sont emmagasinés. Le dictionnaire permet, par exemple, d'inférer qu'après avoir chauffé un objet, celui devient chaud.

• La séparation des connaissances en deux dictionnaires distincts, le dictionnaire de concepts et le lexique, permet de représenter et manipuler les connaissances du monde indépendamment de la langue utilisée.

4 La génération du texte

4.1 La représentation conceptuelle initiale

L'entrée du générateur (un exemple est illustré à la section 4.3) indique le message à transmettre. Il comprend deux parties:

• Une description de l'état initial de chaque participant et l'identité des instances d'action primitives. Comme l'indique la figure 1, il existe 7 actions primitives (`chauffer_c`, `couper_c`, `servir_c`, `mélanger_c`, `cuire_c`, `mettre_c` et `retirer_c`).

• Une description conceptuelle de chaque action de la recette. En effet, chaque proposition du texte final est réalisée à partir d'une description conceptuelle. Ceci n'implique pas une correspondance directe entre la description conceptuelle d'une action et une proposition dans le texte final, comme nous verrons plus loin, certaines actions peuvent être réalisées à l'intérieur de groupes nominaux (GNs). Cette représentation initiale diffère de celle des systèmes Epicure [Dal92] et EDGE [Caw90] qui, à partir d'une description de haut niveau, génèrent de façon arborescente un plan de discours.

4.2 La production du texte

Pour produire le texte final, le générateur construit une proposition à la fois. La production s'effectue d'abord au niveau conceptuel, ensuite au niveau lexical et suit les étapes illustrées à la figure 2 (page de droite).

Avant de choisir les termes à utiliser pour se référer à un concept particulier, le générateur organise le texte "en surface". Rappelons que le système est une composante tactique et n'effectue pas la planification du texte. Le travail au niveau conceptuel modifie l'entrée conceptuelle pour

tenir compte des actions réalisées à l'intérieur de groupes nominaux et du regroupement de certaines actions.

Une fois le travail au niveau conceptuel effectué, le choix des mots et des procédés anaphoriques est effectué. Ce travail s'effectue en trois étapes : la spécialisation des verbes, la génération des anaphores dans les GNs et la réalisation finale.

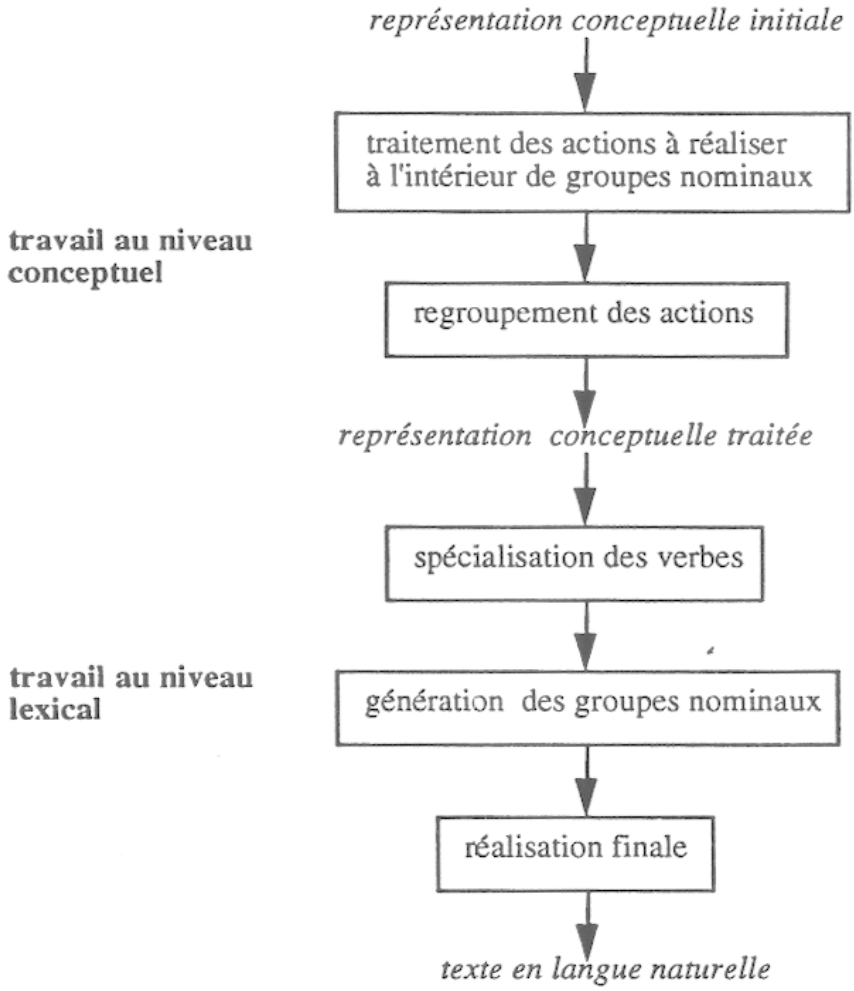

Figure 2 : Étapes de la génération

4.2.1 Les actions réalisées à l'intérieur de groupes nominaux

Certaines actions de préparation sont réalisées dans les textes d'assemblage à l'intérieur de GNs, plutôt que par une proposition à part entière. Par exemple, on retrouvera aussi bien

(18) *Râper les carottes.* Les ajouter au reste des ingrédients.

que

(19) Ajouter *les carottes râpées* au reste des ingrédients.

Dans le premier cas, l'action de « `râper_c carotte1` » est réalisée par une proposition, tandis que dans le second cas, elle est réalisée à l'intérieur d'un GN. Le système n'ayant aucun critère pour déterminer de quelle

façon une action de préparation doit être réalisée, nous supposons que le planificateur l'indique au système.

4.2.2 Le regroupement des actions

Les actions consécutives ayant le même concept d'action ou le même objet sont regroupées en une seule action. Ceci évite de produire des propositions "parallèles" comme

(20) Éplucher les carottes. Râper les carottes.

Dans ce cas, le système produira :

(21) Éplucher et râper les carottes.

4.2.3 La spécialisation des verbes

Pour introduire le verbe le plus approprié, le système effectue une recherche en profondeur dans la hiérarchie des concepts d'action (*cf* la figure 1, p. 221). Pour sélectionner un concept, ses préconditions doivent être satisfaites, c'est-à-dire que l'état courant des participants doit être approprié à cette action. Une fois un concept d'action sélectionné, ses postconditions sont immédiatement appliquées pour refléter le nouvel état des objets. Le concept d'action choisi est ensuite lexicalisé en consultant le lexique.

4.2.4 La génération des anaphores dans les GNs

Pour générer les GNs, les contraintes développées par Tutin sont vérifiées pour déterminer le procédé anaphorique le plus approprié au contexte présent. Nous distinguons entre **l'introduction, la première mention** et **les mentions ultérieures** d'une instance (par exemple `lapin1`). Ceci est en accord avec les travaux de Dale, qui distingue aussi la première mention et les mentions ultérieures.

L'introduction d'un ingrédient s'effectue dans la liste des ingrédients de la recette, avant les instructions, l'introduction d'un instrument s'effectue dans les instructions, au moment où ils sont nécessaires. À l'introduction d'une instance, aucune anaphore ne peut être produite simplement parce qu'il y a pas encore d'antécédent.

Ensuite, à **la première mention** dans le texte, une instance ne pourra pas être mentionnée par une ellipse du C.O.D., un pronom ni un dérivé sémantique de type résultat.

En **mentions ultérieures**, une instance pourra profiter de tous les procédés anaphoriques si leurs contraintes sont satisfaites.

Le but de l'article est de présenter l'aspect informatique du système, c'est pourquoi nous ne justifierons pas les contraintes linguistiques gérant l'introduction des anaphores ni de leur ordre de préférence. Nous présentons ici la réalisation des procédés anaphoriques introduits à la section 2. Pour illustrer le choix de procédés, nous nous référons à la recette de Lapin à la moutarde générée par le système. La figure 3 (page de droite) illustre cette recette ainsi que les procédés sélectionnés par le système pour la production des GNs.

Recette générée automatiquement

Lapin à la moutarde

1 **lapin**$_1$
3 c. à soupe de **moutarde de Dijon**$_1$
pommes de terre$_1$
50 ml d' **huile de maïs**$_1$
100 ml de **lait**$_1$
crème$_1$
estragon$_1$
sel$_1$
poivre$_1$

Couper **la viande**$_2$ et mettre la **moutarde**$_3$ sur **les morceaux**$_4$. Chauffer l'**huile**$_3$ dans **une casserole**$_1$, **y**$_5$ faire revenir **la viande**$_2$ de tous côtés et mettre **l'huile, le lait et les assaisonnements**$_6$ dans **la casserole**$_7$. **Y**$_5$ mettre **le couvercle**$_8$ et cuire **la viande, les assaisonnements, l'huile et le lait**$_9$ à feu très lent. Mettre **la crème**$_7$, chauffer **la viande, les assaisonnements, l'huile, la crème et le lait**$_{10}$ et servir **Ø**$_{11}$ avec **les pommes de terre cuites**$_{12}$.

$_1$Introduction
$_2$Dénomination de base
$_3$Répétition partielle
$_4$Dérivé sémantique de type résultat
(Le lecteur attentif remarquera qu'ici un pronom aurait dû être produit, cependant, pour illustrer la génération d'un dérivé sémantique de type résultat, nous avons donné priorité à ce dernier procédé par rapport à la pronominalisation.)
$_5$Pronominalisation
$_6$Coordination complexe (2 répétitions exactes et 1 hyperonymation)
$_7$Répétition exacte
$_8$Introduction (d'une partie d'un objet déjà introduit)
$_9$Coordination complexe
(1 dénomination de base, 1 hyperonymation, 1 répétition partielle et 1 répétition exacte)
$_{10}$Coordination complexe
(1 dénomination de base, 1 hyperonymation, 1 répétition partielle et 2 répétitions exactes)
$_{11}$Ellipse
$_{12}$Groupe nominal avec participe passé (action réalisée à l'intérieur d'un GN)

Figure 3 : Procédés sélectionnés pour la recette du Lapin à la moutarde
(Pour faciliter la lecture, les GNs sont en caractères gras et les indices indiquent les procédés anaphoriques sélectionnés par le système.)

4.2.4.1 Anaphores de première mention

Si le GN à introduire doit contenir une action (*cf* la section 4.2.1), le participe passé correspondant à cette action est généré, l'instance est lexicalisée et le GN comprenant ces deux éléments est produit (le GN12 de la figure 3 en est un exemple). Sinon, les procédés anaphoriques de première mention sont tentés dans l'ordre suivant

• L'hyperonymation :

Pour produire une hyperonymation de plusieurs instances, les conditions suivantes doivent être respectées :

– Les instances à introduire doivent avoir un hyperonyme en commun. Ceci est vérifié en remontant dans la hiérarchie du dictionnaire de concepts (cf la figure 1) en cherchant un ancêtre commun à toutes les instances pour lequel il existe une réalisation lexicale. Pour éviter de produire des hyperonymes trop généraux, la recherche dans la hiérarchie conceptuelle ne remonte qu'un nombre limité de niveaux.

– Aucune ambiguïté ne sera créée, c'est-à-dire qu'aucune autre instance participant à la recette ne doit descendre de l'ancêtre commun. Ceci évite les ambiguïtés comme :

(22) Couper *les pommes, les poires* et les pêches en cube, ensuite ajouter *les fruits* au mélange à gâteau.

où *les fruits* ne se réfèrent qu'aux pommes et aux poires.

Si ces deux conditions sont satisfaites, une réalisation lexicale[1] est trouvée pour l'ancêtre commun. Par exemple, dans le texte de la figure 3, le système a hyperonymé les instances (`moutarde1, estragon1, sel1, poivre1`) par *assaisonnement* dans la coordination du GN 10, car ces quatre instances ont `assaisonnement_c` comme concept commun, et aucun autre descendant de ce concept ne participe à la recette.

• La coordination complexe :

Pour produire une coordination complexe de plusieurs instances, les instances ayant un hyperonyme en commun sont hyperonymées et les procédés anaphoriques de première mention sont appliqués récursivement sur le reste des instances. Un exemple de coordination complexe sera décrit en détail plus loin.

• La dénomination de base :

Pour produire une dénomination de base :

– L'instance doit être un descendant d'un concept de base. Une recherche ascendante dans le dictionnaire conceptuel est effectuée (un nombre limité de niveaux) jusqu'à ce qu'un concept de base (un concept correspondant à un nom de base[2]) soit trouvé.

---

1 Les lexicalisations de concepts, c'est-à-dire le lien entre le dictionnaire de concepts et le lexique, sont emmagasinées dans le système.

2 Le système considère que le trait *de base* est une caractéristique conceptuelle plutôt que lexicale.

– Aucune ambiguïté ne sera créée. Cette contrainte est vérifiée si aucune autre instance participant à la recette n'est un descendant du concept de base.

Si ces contraintes sont respectées, alors le concept de base est lexicalisé.

Dans le Lapin à la moutarde, il y a une dénomination de base pour mentionner l'instance `lapin1` (indice 2 de la figure 3), car `viande_c` est considéré un concept de base, tandis que `lapin_c` ne l'est pas.

• La répétition partielle

Un GN composé seulement d'un déterminant et de la lexicalisation de l'instance est produit si :

– L'instance à mentionner est qualifiée, comme dans *les oignons blancs*.

– Si l'élimination des attributs ne cause pas d'ambiguïté, c'est-à-dire, lorsque les attributs enlevés sont nécessaires pour identifier distinctement l'instance référée.

Dans la recette du Lapin à la moutarde, le GN *la moutarde* a comme antécédent *moutarde de Dijon*. Cette répétition partielle n'est pas ambiguë, car la recette ne nécessite qu'un seul type de moutarde.

• La répétition exacte :

Une répétition exacte est produite en première mention, si aucun des procédés ci-dessus ne peut être appliqué. Ici, le GN contient un déterminant, la lexicalisation de l'instance et tous les attributs de cette instance. C'est donc le procédé produisant la description la plus complète. La recette de la figure 3 comprend les répétitions exactes: *la casserole* et *la crème*, aucune d'elles n'ayant des attributs.

4.2.4.2 Anaphores de mentions ultérieures

En mention ultérieure, l'algorithme tente les procédés suivant: l'ellipse, la pronominalisation, le dérivé sémantique de type résultat et si aucun de ces procédés ne peut être utilisé, les procédés de première mention sont tentés.

• L'ellipse :

L'ellipse consiste à produire un GN vide si:

– Les instances à mentionner ne descendent pas du même concept. En effet, si les instances ont un concept commun, un pronom est préféré à l'ellipse.

– L'antécédent est situé dans le même paragraphe. Ceci permet de le retracer facilement durant la lecture.

– Finalement, les instances à introduire forment le focus local de la proposition précédente. Pour des raisons pratiques, le système tient compte du focus local, cependant une introduction plus naturelle devrait considérer le focus global du texte. En effet, l'antécédent de l'ellipse est identifié aux instances les plus saillantes du texte, plutôt qu'aux instances sur lesquelles portait l'action précédente. Dans les textes d'assemblage, le focus local est

généralement identifié à l'objet sur lequel l'action a lieu et est réalisé par le C.O.D. d'une proposition.

Pour mentionner le focus local de l'avant dernière proposition du Lapin à la moutarde, le système a produit une ellipse à la dernière proposition. En effet, les GNs 10 et 11 ont les mêmes référents.

• **La pronominalisation :**

Ce procédé introduit un pronom suivant les traits morphologiques de l'antécédent (et non de l'instance à mentionner) et suivant son rôle syntaxique. Ainsi on aura :

(23a) Couper le veau en dés. Faire frire *les morceaux* avant de *les* mettre au four.

(23b) Couper le veau en dés. Faire frire *les morceaux* avant d'*y* ajouter les champignons.

L'exemple 23a illustre que le pronom *(les)* a les traits morphologiques de genre et de nombre de son antécédent *(les morceaux)* plutôt que ceux de l'instance qu'il introduit (`veau1`). Aussi un pronom nominal *(les)* est produit plutôt qu'un pronom locatif comme dans 23b.

Pour pronominaliser une (ou plusieurs) instance :

– L'antécédent doit être situé dans le même paragraphe.

– Aucune ambiguïté ne doit être créée. Le système vérifie les ambiguïtés, grâce à trois méthodes : les contraintes de focus, les restrictions de sélection et les restrictions morphologiques.

* Les contraintes de focus : Si l'antécédent est le focus local de la proposition précédente, il sera directement retracé par le lecteur, et aucune ambiguïté ne sera crée. Si tel n'est pas le cas, les restrictions de sélection sont vérifiées.

* Les restrictions de sélection : Si aucune autre instance mentionnée à la proposition précédente peut participer à l'action courante, alors le pronom ne sera pas ambigu. Par exemple, le pronom dans :

(24) Verser le vin sur *le bœuf*, puis *le* faire revenir.

n'est pas ambigu car on ne peut pas faire revenir le vin. Pour vérifier cette contrainte, le système vérifie qu'aucune autre instance de la proposition précédente ne satisfait les préconditions de l'action courante.

* Les restrictions morphologiques : Finalement, si les restrictions de sélection ne suffisent pas, les restrictions morphologiques vérifient que l'antécédent est identifiable par le genre et le nombre du pronom.

Deux pronoms ont été produits dans le Lapin à la moutarde, les deux sont des pronoms locatifs.

• **Le dérivé sémantique de type résultat :**

Pour introduire une (ou plusieurs) instance par un dérivé sémantique de type résultat :

– L'instance doit être le focus local de la proposition précédente ou celle d'avant. Ceci est nécessaire, car plus le résultat typique a été créé il y a longtemps, plus il est difficile de retracer l'antécédent du dérivé.

– L'instance doit avoir un dérivé sémantique de type résultat. Pour vérifier ceci, le dictionnaire de concepts est consulté pour identifier le résultat typique (une postcondition) de la dernière action effectuée sur l'instance. Si ce résultat typique se lexicalise dans la langue cible, l'anaphore est appliquée.

Si une action ayant le même résultat typique a été appliquée sur une autre instance, une ambiguïté est possible. Par exemple dans,

(25) Couper les champignons de Paris. Couper la viande. Faire frire *les morceaux.*

l'antécédent de *les morceaux* peut tout aussi bien être *les champignons de Paris* ou *la viande.* Dans ces cas d'ambiguïté, le système précise l'instance référée en y appliquant les procédés de première mention. Par exemple, si l'antécédent de *les morceaux* dans 25 est *les champignons de Paris* et la répétition partielle est sélectionnée pour les identifier, le générateur produira :

(26) Couper les champignons de Paris. Couper la viande. Faire frire les morceaux de champignons.

À la figure 3, l'instance `lapin1` est mentionnée par *les mor*ceaux (indice 4), car elle vient juste d'être coupée et aucun autre ingrédient n'a été coupé.

4.2.4.3 Exemple

Pour illustrer le choix des procédés anaphoriques, nous allons suivre la génération du GN *la viande, les assaisonnements, l'huile, la crème et le lait* (10), pour faire référence aux instances `(lapin1, moutarde1, huile1, crème1, lait1, estragon1, sel1, poivre1)`. Seuls les procédés de première mention sont considérés, car ces instances n'ont pas été mentionnés auparavant. Le procédé d'hyperonymation est donc considéré, mais rejeté, car il n'existe pas d'hyperonyme commun à toutes les instances. La coordination complexe est alors considérée. Les instances `(moutarde1, estragon1, sel1, poivre1)` sont hyperonymées par *les assaisonnements*, et les procédés de première mention sont appliqués récursivement sur chaque instance restant, c'est-à-dire `lapin1, huile1, crème1` et `lait1`. Pour toutes ces instances, l'hyperonymation et la coordination complexe sont rejetées car une seule instance doit être mentionnée. La dénomination de base est sélectionnée pour `lapin1` car le concept de base `viande_c` est trouvé dans la hiérarchie conceptuelle. Pour les autres instances, la dénomination de base est rejetée car aucun concept de base n'est trouvé. La répétition partielle est alors tentée. Seule `huile1` a un attribut la qualifiant (*de maïs*) et son élimination ne causera pas d'ambiguïté, car la recette ne requiert pas d'autre sorte d'huile. Pour mentionner les deux dernières instances `crème1` et `lait1`, la répétition exacte est donc sélectionnée. Tous ces choix sont ensuite combinés pour former le GN *la viande, les assaisonnements, l'huile, la crème et le lait.*

4.2.5 La réalisation finale

La réalisation finale combine les résultats des deux étapes précédentes : les verbes spécialisés (section 4.2.3) et les GNs produits (section 4.2.4) pour former la proposition finale. Le système effectue à cette dernière étape les élisions nécessaires, l'ordonnancement des mots dans les propositions et l'édition finale du texte (mises en majuscules...).

4.3 Implantation

Avec ses quelques 2500 lignes de Prolog (incluant 700 lignes pour les dictionnaires), le générateur a pris environ 2 secondes pour produire la recette du Lapin à la moutarde sur une station de travail Sparc. Le système se base sur la Grammaire à Clauses Définies (DCG), qui gère la construction des propositions et sur la grammaire de traits de Boyer [Boy87], qui facilite la manipulation et l'unification de traits syntaxiques et sémantiques.

L'entrée du générateur qui a permis de produire le Lapin à la moutarde est illustrée à la figure 4 (page de droite). Comme décrite à la section 4.1, l'entrée conceptuelle initiale comprend deux parties : la description de l'état initial des participants et une liste décrivant chaque action de la recette. La représentation initiale est assez détaillée et proche des propositions finales, pour permettre au système de se concentrer sur la génération des anaphores. Nous travaillons présentement sur cet aspect du système afin de fournir au générateur une représentation initiale plus abstraite. Le but est de n'inclure dans la représentation initiale que l'information sur le contenu du message à transmettre et d'éliminer l'information sur la forme syntaxique des propositions et des phrases.

5 Conclusion

Nous avons présenté, dans cet article, un générateur de texte axé sur la production d'anaphores. Le système génère actuellement des recettes de cuisine, mais peut être adapté à un autre type d'assemblage[1] (par exemple, des instructions de bricolage ou de couture) car les règles d'introduction d'anaphores ne sont pas spécifiques aux recettes. Nous avons présenté le processus de génération et la réalisation des procédés anaphoriques. Le système est basé sur la séparation des informations conceptuelles et lexicales en deux dictionnaires distincts. L'important rôle joué par le dictionnaire de concepts a été souligné: il permet au système d'inférer le dynamisme des objets à partir des connaissances qu'il contient et ce au niveau conceptuel plutôt que lexical.

1 en incorporant un dictionnaire approprié.

```
% partie 1: description de l'état initial des participants

% participant(Type, Instance, Concept, Attributs).
participant(ingredient, lapin1, lapin_c, [quantite: 1]).
participant(ingredient, moutarde1, moutarde_c,
    [quantite: 3, unite: c_a_soupe_c, attributs: 'Dijon']).
                                .
                                .
                                .
participant(ingredient, poivre1, poivre_c, []).

% action(Instance, Concept).
action(couper1, couper_c).
                                .
                                .
                                .
action(servir1, servir_c).

% partie 2: description des actions de la recette
actions
    ([[[action(principale,
    [action: couper1, objet: lapin1]),

    action(principale,
        [action: mettre1, objet: moutarde1,
        lieu: [valeur:lapin1, prep:par_dessus_c]])],
                                .
                                .
                                .
    action(principale,
        [action: chauffer2,
        objet:(lapin1, moutarde1, huile1, creme1, lait1,
                estragon1, sel1, poivre1)]),

    action(principale,
        [action: servir1,
        objet:(lapin1, moutarde1, huile1, creme1, lait1,
                estragon1, sel1, poivre1)]),
         acc: pomme_de_terre1])]]]).
```

Figure 4 : Entrée conceptuelle du Lapin à la moutarde

Remerciements

Nous nous devons de remercier tout d'abord Agnès Tutin pour ses constantes discussions et son appui, et sans qui cette recherche n'aurait pas vue le jour. Nous tenons aussi à remercier les membres de l'équipe SCRIPTUM de L'Université de Montréal et en particulier Michel Gagnon, Massimo Fasciano et Richard Kittredge dont les nombreux commentaires ont permis d'améliorer ce texte.

Ce travail a été réalisé grâce à l'aide financière du Conseil de Recherche en Sciences Naturelles et en Génie du Canada.

Références

[Boy87] M. Boyer. « Towards Functional Logic Grammars ». Dans P. St-Dizier, éditeur, *Proceedings of the Second International Workshop on Natural Language Understanding and Logic programming*, pages 46-62, Vancouver, 1987.

[Caw90] A. Cawsey. « Generating Explanatory Discourse ». Dans Dale et al. éditeurs, *Current Research in Natural Language Generation*, chapitre 4, pages 75-101. Academic Press, 1990.

[Dal92] R. Dale. « Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes ». ACL-MIT Press Series in Natural Language Processing. The MIT Press, 1992.

[Dan91] L. Danlos. « Génération automatique de textes en langue naturelle ». Dans ce volume même, p. 197-214.

[DMZ90] R. Dale, C. Mellish et M. Zock, éditeurs. *Current Research in Natural Language Generation*. Cognitive Science Series. Academic Press, 1990.

[GS86] B. J. Grosz et C. L. Sidner. « Attention, Intentions, and The Structure of Discourse ». *Computational Linguistics*, 12(3):175-204, juillet-septembre 1986.

[HH76] M. A. K. Halliday et R. Hasan. *Cohesion in English*. Quirk, R., Longman: London, 1976.

[Man83] W. C. Mann. « An overview of the Penman text generation system ». Dans *Proceedings of the National Conference on Artificial Intelligence*, pages 261-265. American Association for Artificial Intelligence, 1983.

[McK85] K. R. McKeown. « Discourse strategies for generating natural-language text ». *Artificial Intelligence*, 27:1-41, 1985.

[Mel84] I. Mel'cuk. *Dictionnaire explicatif et combinatoire du français contemporain*. Les Presses de l'Université de Montréal, 1984.

[Nam89] F. Namer. « Subject erasing and pronominalization in Italian text generation ». *EACL-Manchester*, 1989.

[Rei90a] E. B. Reiter. « Generating Description that Exploit a User's Domain Knowledge ». Dans Dale et al. éditeurs, *Current Research in Natural Language Generation*, chapitre 10, pages 257-285. Academic Press, 1990.

[Rei90b] E. B. Reiter. « A New Model for Lexical Choice for Open-Class Words ». *Proceedings of the Fifth International Workshop on Natural Language*, pages 23-30, juin 1990.

[Tut92a] A. Tutin. *Étude des anaphores grammaticales et lexicales pour la génération automatique de textes de procédures*. Thèse de doctorat, Département de linguistique et philologie, Université de Montréal, 1992.

[Tut92b] A. Tutin. « Lexical choice in context: generating procedural texts ». Dans *Proceedings of the fifteenth International Conference on Computational Linguistics*, volume II, pages 763-769, Nantes, août 1992. COLING.

[Zoc91] M. Zock. « La génération interactive de langage: comment visualiser le passage de l'idée à la phrase ». Dans ce volume-même, p. 249-267.