Navigation textuelle : représentation des textes et des connaissances

Textual Navigation : Texts and Knowledge Representation

Javier Couto* — Jean-Luc Minel**

* *Universidad de la República – Facultad de Ingeniería*
INCO
J. Herrera y Reissig 565, Montevideo, Uruguay
jcouto@fing.edu.uy

** *MoDyCo, UMR7114, CNRS*
Université Paris X
200, avenue de la République, France
Jean-Luc.Minel@u-paris10.fr

RÉSUMÉ. *Nous présentons tout d'abord notre conception de la navigation textuelle conçue comme un processus cognitif qui convoque des connaissances qui sont propres à la finalité de la navigation. Nous formulons l'hypothèse que ces connaissances peuvent être, en partie, modélisées sous une forme déclarative avec le langage SEXTANT que nous décrivons. Enfin, nous présentons plusieurs expérimentations qui utilisent la plate-forme NaviTexte dans laquelle le langage SEXTANT est implémenté.*

ABSTRACT. *In this paper, we present our approach to text navigation conceived like a cognitive process, which exploits navigation specific knowledge. We draw up the hypothesis that such knowledge can be designed in a declarative way with our language SEXTANT. Finally, several experimentations are described.*

MOTS-CLÉS : *navigation textuelle assistée, modélisation des connaissances de navigation, modèle de représentation de texte.*

KEYWORDS *Assisted navigation of texts, navigation knowledge management, text model.*

TAL. Volume 47 – n° 2/2006, pages 225 à 254

1. Conceptions de la navigation textuelle

Le terme de navigation textuelle reçoit de multiples interprétations. La plus commune renvoie inévitablement au processus mis en œuvre par les outils de navigation utilisés pour circuler dans les documents hypertextes ; c'est-à-dire la possibilité d'activer un lien pour déplacer le point de lecture, ce déplacement pouvant être intra ou intertextuel. Avant de présenter notre propre conception, nous allons rappeler les principales caractéristiques de la navigation hypertexte.

1.1. La navigation hypertexte

Plusieurs points caractérisent la navigation hypertexte quelle fut effectivement mise en en œuvre à l'origine. Tout d'abord, l'activation du lien est en quelque sorte « aveugle » ou « non assistée », plus précisément aucune signalétique (en dehors d'un titre ou de l'adresse Url qui est en général peu significative) ou instructions de navigation ne sont associées au lien. Deuxièmement, l'orientation de la navigation n'est pas indiquée explicitement : le lecteur ne sait pas si le déplacement se fait vers l'amont ou vers l'aval[1] du texte lu. Dans certains systèmes, l'affichage d'une carte représentant l'ensemble du site et la localisation du point de lecture sont utilisés pour résoudre en partie ce problème (Danielson, 2002). Enfin et surtout, les liens sont placés dans le corps même du texte, mélangeant ainsi le texte et les connaissances qui spécifient la navigation dans ce texte. Cela limite les possibilités d'adapter les parcours de lecture à un lecteur car tous les parcours possibles sont potentiellement définis indépendamment du lecteur. En d'autres termes, aucune information ou connaissances complexes ne peuvent être associées à la navigation.

Malgré le succès incontestable de l'hypertexte, les avis sur l'effectivité de son utilisation comme support à la lecture sont partagés. Par exemple, certaines critiques mentionnent le phénomène de désorientation cognitive (Elm *et* Woods, 1985) (Edwards *et* Hardman, 1989) (Cotte, 2004) lorsqu'un utilisateur navigue dans un environnement hypertexte. Même si ce phénomène est généralement utilisé comme mécanisme de narration par les écrivains, il est particulièrement perturbateur lorsque les textes sont de type informatif ou argumentatif, entre autres. Plusieurs approches ont proposé de résoudre en partie ces problèmes en adaptant les techniques d'hypertextes. On peut les distinguer suivant deux modes : adaptatif ou dynamique.

L'approche de l'hypertexte adaptatif (Mathe *et* Chen, 1994) (Brusilovsky, 1994 et 1996) cherche à pallier le problème de l'adaptabilité à un utilisateur présente dans l'approche traditionnelle de l'hypertexte. L'objectif principal de cette approche est d'adapter un système hypertexte aux besoins spécifiques d'un utilisateur. Notons que la modélisation de l'utilisateur est un point clé de cette approche. Il existe deux voies d'adaptation : *présentationnelle* et *navigationnelle*. La première a trait à l'adaptation des nœuds afin de modifier la manière dont l'information est affichée, tandis que la seconde concerne les liens qui sont présentés à l'utilisateur.

[1] L'orientation n'a de signification que dans le cas d'une navigation intratextuelle.

Tandis que l'hypertexte adaptatif conserve la notion de lien tel qu'il existe dans l'approche traditionnelle de l'hypertexte, l'hypertexte dynamique change la notion même de lien. Dans cette approche, les liens sont créés à l'exécution, au lieu d'être calculé préalablement (*liens pré-calculés*), ou à travers des modifications ou sélections dans des ensembles existants de liens (*liens adaptatifs*) (Bodner *et* Chignell, 1999). Une possibilité dans cette approche consiste à calculer les liens selon des relations prédéfinies ou selon des critères de similarité entre les textes. Dans ce cas, un lien n'est pas défini comme un pointeur d'un nœud hypertexte vers un autre, mais comme une requête visant un nœud. Ces requêtes peuvent prendre en considération l'historique de navigation, le profil de l'utilisateur, etc.

Cette notion de requête sur un nœud est une source d'inspiration pour notre démarche, mais plutôt que de considérer des éléments liés au lecteur, nous allons exploiter les différentes connaissances linguistiques (au sens large du terme) présentes dans les textes. Cette exploitation exige une étape d'annotation du texte, résultat d'un traitement linguistique finalisé.

1.2. *Une approche fondée sur la modélisation des connaissances*

Notre conception de la navigation textuelle se démarque ainsi de la navigation hypertextuelle car nous considérons que circuler ou naviguer dans un texte est l'expression d'un processus cognitif qui convoque des connaissances qui sont propres à la finalité de la navigation (Minel, 2002 et 2003 ; Couto et Minel, 2004 ; Couto, 2006). Ainsi, comme nous l'illustrerons en présentant différentes expérimentations, un documentaliste qui doit écrire un résumé d'un texte (Endres-Niggemeyer B. *et al.*, 1995) ne navigue pas de la même façon qu'un lecteur intéressé par l'évolution des sentiments d'un des personnages d'un roman (Mathieu, 2005) ou qu'un linguiste qui explore les annotations placées par un système automatique (Pery-Woodley, 2005). Ainsi, le fait qu'un texte soit maintenant numérisé et qu'il soit présenté au lecteur sur un écran peut être considéré, de notre point de vue, comme une nouvelle mutation qui place le lecteur devant de nouvelles possibilités qui restent à explorer :

« Le texte [...] offre en effet une richesse sémiotique particulière, qui fournit de multiples objets d'interprétation et de multiples pistes d'actions [...] les lecteurs n'ont pas la même démarche envers l'objet ni la même définition de cet objet, ils ne « voient » pas la même chose. » (Souchier *et al.*, 2003)

Nous formulons l'hypothèse que la démarche du lecteur, ou plus exactement d'une catégorie de lecteurs, peut d'une part, s'appuyer sur la présence de marques discursives et d'annotations dans le texte et, d'autre part, sur des connaissances qui exploitent celles-ci. De plus, ces connaissances sont susceptibles d'être modélisées sous une forme déclarative. En conséquence, nous proposons le langage SEXTANT[2] pour exprimer ces connaissances. Autrement dit, nous considérons qu'il

[2] Par analogie avec les navigateurs du XVIII[e] siècle qui ont parcouru le monde en s'orientant sur les mers avec un sextant.

ne suffit pas de créer des liens mais qu'il est nécessaire d'expliciter l'opération de navigation et que, de plus, ce processus doit être mis en œuvre par un « expert » capable d'encoder ces connaissances.

Une différence capitale entre notre conception de la navigation textuelle et la navigation hypertextuelle tient au statut du texte et de la modélisation des connaissances de navigation. Dans le cas de l'hypertexte, la visualisation du texte est unique et les connaissances de navigation sont encodées dans le texte même et en font partie. Ces connaissances ne sont donc pas explicitement modélisées mais plutôt dispersées dans le texte. Dans notre approche, nous considérons qu'un texte peut être visualisé de différentes manières, appelées vue du texte, et que chaque vue peut donner lieu à différentes manières de naviguer dans le texte. De plus, les connaissances navigationnelles sont modélisées indépendamment de l'objet texte et gérées dans une base de connaissances appelées « module de navigation ».

Du point de vue du lecteur, cette différence de conception entraîne que c'est celui-ci qui active les connaissances d'interprétation (Kintsch, 1998 ; Baccino, 2004) et qu'il peut interagir en choisissant la vue du texte et la voie de navigation qui lui semble la plus appropriée pour sa tâche de lecture. Il est néanmoins évident que cette interaction est actuellement très limitée, car la navigation proposée reste dans les limites posées par le concepteur des modules de navigation. En ce sens, il serait peut-être plus précis de parler de « navigation textuelle assistée ».

Notre approche systématique de la navigation textuelle nécessite quatre éléments : i) une représentation du texte pouvant décrire différents phénomènes linguistiques ; ii) la possibilité d'isoler les connaissances de visualisation et de navigation ; iii) un agent (une personne, une équipe d'experts, un système, etc.) capable d'encoder ces connaissances ; iv) un système qui interprète ces connaissances. Le point iii) sera abordé dans la discussion sur l'évaluation des expérimentations (*cf.* 5.4).

2. Représentation du texte

C'est le type de traitement à effectuer qui vient déterminer habituellement la représentation la plus convenable selon des critères à choisir : rapport coût/performance, souplesse, exhaustivité, simplicité, etc. En conséquence, il existe des représentations d'un texte qui le conceptualisent comme une séquence de caractères, jusqu'à des représentations fortement imprégnées par des approches linguistiques, où la représentation de texte permet de modéliser certains phénomènes discursifs. Dans cette optique, ces logiciels de traitement automatique de textes fonctionnent de manière semblable à certains « traducteurs » (Aho *et al.*, 1986) : ils traitent un texte source, construisant une représentation intermédiaire, puis ils génèrent un texte résultat. Bien évidemment, les trois représentations (*source*, *intermédiaire* et *résultat*) ne sont pas nécessairement différentes pour tous les systèmes de traitement de textes, et il existe des cas où la représentation source

coïncide avec la représentation intermédiaire, où cette dernière correspond aussi à la représentation du résultat.

2.1. *Nature des objets à manipuler*

Les représentations structurées de texte sont parmi les plus utilisées actuellement. Elles adoptent généralement une approche hiérarchique où des aspects syntaxiques et discursifs se mélangent souvent. La représentation générique proposée par la Text Encoding Initiative (TEI), et plus spécifiquement la TEI Lite est emblématique de ce type d'approche. Celle-ci propose des éléments de division du texte tels que le paragraphe *« p »*, ou les génériques *« div »*, qui peuvent s'emboîter (« div1 », « div2 », …, « div8 »). Ces éléments de division peuvent être typés et comporter un identifiant, utilisé postérieurement pour créer des liens entre les éléments.

Certains auteurs (Webber *et al.*, 2003), (Wolf *et* Gibson, 2005) ont critiqué le choix de représentations arborescentes pour modéliser les phénomènes discursifs. Par exemple, dans (Wolf *et* Gibson, 2005), les auteurs montrent comment des structures de graphe sont nécessaires afin de représenter la cohérence discursive. Ils présentent un exemple de texte analysé à l'aide de l'ensemble des relations de cohérence développées dans (Hobbs, 1985) et (Kehler, 2002) et montrent que la structure de cohérence discursive correspond à un graphe. D'autres exemples (*e.g.* les relations anaphoriques) montrent qu'il existe des phénomènes discursifs non représentables avec des structures hiérarchiques et qui doivent faire appel à des structures de graphe.

2.2. *Une représentation hybride*

Les insuffisances des représentations présentées précédemment ont motivé la définition d'une représentation des textes spécifique à la navigation textuelle (Couto, 2006) qui s'inspire à la fois des propositions de (Crispino, 2003) et de celles du modèle TEI Lite (TEI), tout en se fixant les objectifs suivants : le premier objectif vise à ne pas restreindre le type d'unités textuelles qui composent un texte à un ensemble prédéterminé ; le deuxième objectif consiste à offrir à la fois une organisation hiérarchique des unités textuelles et une autre permettant d'exprimer des relations non hiérarchiques ; le troisième objectif est de considérer les titres comme des unités textuelles ; le dernier objectif vise à offrir la possibilité que toute unité textuelle, y compris les titres et les relations non hiérarchiques, soit susceptible d'avoir un nombre non limité d'annotations de nature quelconque.

Un texte est ainsi représenté comme une hiérarchie d'unités textuelles de base, permettant la définition d'unités plus complexes susceptibles de ne pas suivre la hiérarchie établie. Toutes les unités de base sont typées, ce qui offre une souplesse non négligeable car au lieu d'avoir, par exemple, une unité textuelle *section* ou *paragraphe*, nous disposons d'une unité textuelle générique pouvant être d'un type quelconque qui peut éventuellement être instancié en un type *section* ou *paragraphe*. Il ne s'agit pas ici d'une nuance mais d'une décision conceptuelle d'après laquelle

l'utilisateur[3] n'est pas restreint par un jeu prédéfini d'unités. Au contraire, celui-ci peut définir ces types d'unités à volonté.

Le texte est donc composé des trois éléments : son *Titre*, un ensemble de relations non hiérarchiques regroupées dans une unité nommée *Tête*, et une hiérarchie d'unités textuelles formulée dans une unité nommée *Corps*. Précisons qu'une DTD (Couto, 2006) spécifie l'agencement de ces différents éléments.

2.2.1. *Le Titre*

Pourquoi concéder un statut spécial aux titres au lieu, par exemple, de confier à l'utilisateur le soin de définir des unités textuelles de type « Titre » ? Les titres pouvant faire partie des relations entre constituants textuels (par exemple une relation rhétorique), leur légitimité comme une entité allant au-delà du simple attribut d'un constituant textuel est justifiée. De ce fait, un titre est composé d'unités textuelles de base dont l'organisation dépendra de l'analyse faite par l'encodeur.

Malgré une diversité d'études sur les titres de presse (Ho-Dac *et al.*, 2004), il en existe peu sur leur rôle dans des documents longs structurés hiérarchiquement en sections et sous-sections titrées. Intuitivement, même si les titres surpassent le simple rôle d'attribut, il est cependant certain que ceux-ci sont généralement reliés à un constituant textuel, celui-ci constituant leur empan. Le titre est donc une unité textuelle particulière formée, au sens le plus général, par une hiérarchie d'unités textuelles de base, et étant toujours affectée à une unité textuelle de base. Ces unités textuelles de base sont présentées par la suite.

2.2.2. *Le Corps*

Dans le *Corps*, l'élément de base de notre modèle est l'*Unité Textuelle* (UT) typée, ce qui permet d'incorporer de nouveaux éléments textuels de manière simple. Ces principes d'annotation sur lequel s'appuie la plate-forme de navigation textuelle NaviTexte sont classiquement ceux proposés par les standards tels que ceux de la TEI. Concrètement, dans le *Corps*, une UT est balisée, avec la balise <Chaine>, et des attributs, en nombre illimité, peuvent lui être attribués. Chaque UT est typée et possède optionnellement un rang. Le type peut aussi bien dénoter la fonction structurelle de l'unité en question, sa caractéristique syntaxique, sa fonction discursive. On peut remarquer que ce type d'annotation laisse une marge de liberté très grande, notamment dans la répartition des valeurs d'annotation entre le type de l'UT et les attributs de cette UT (Couto *et al.*, 2005).

2.2.2.1. Granularité des UT et notion d'héritage d'annotations

Examinons un fragment de texte (*cf.* tableau 1), emprunté à (Couto *et al.*, 2004) et encodé par nos soins afin d'être utilisé par NaviTexte (*cf.* section 4). Le diagramme correspondant à l'analyse effectuée pour ce fragment est affiché dans la

[3] L'utilisateur pouvant être, entre autre, un système automatique ou un opérateur humain.

figure 1, où les UT ont été numérotées, selon un parcours en profondeur de l'arborescence, pour simplifier leur mention.

L'importance quantitative de l'investissement étranger est cependant moins significative de l'impact des firmes multinationales que le type de secteurs où elles se localisent. En Côte-d'Ivoire les firmes contrôlent pratiquement l'ensemble de l'industrie produisant pour le marché interne. Au contraire, l'accès à ce dernier leur est interdit dans la plupart des branches en Corée du Sud. Cette situation a des conséquences décisives, particulièrement sur trois variables stratégiques du processus de développement : l'allocation des ressources, le modèle de consommation et l'intégration en amont de l'activité industrielle.

Tableau 1. *Exemple de texte à représenter.*

Plusieurs points sont à signaler dans l'exemple précédent. Le premier correspond à la granularité des UT : notons qu'il n'est pas nécessaire d'encoder toutes les UT de manière identique. Ainsi, l'UT numéro 2 (*cf.* figure 1) est encodée comme une phrase, tandis que les trois autres UT de type « Phrase » sont composées d'UT filles. Le deuxième point correspond à deux notions importantes : l'*héritage d'annotations* et la *synthèse de chaînes lexicales*. Les annotations d'une UT peuvent être héritées ou bien par les UT descendantes (*héritage descendant*), ou bien par les UT ascendantes (*héritage ascendant*), ou bien dans les deux cas.[4] Ce *mode d'héritage* est choisi parmi les valeurs suivantes : *aucun*, *ascendant*, *descendant* et *bidirectionnel*.

Dans l'exemple, l'UT numéro 10 est une phrase annotée comme une phrase démonstrative (*ES* signifiant « Étiquette Sémantique » et *Dém.* la valeur « Démonstration »), composée d'un segment et de trois syntagmes nominaux. Chacune de ces quatre UT filles, en tant que constituants textuels individuels, ne constitue pas une « Démonstration ». Par exemple, le syntagme nominal « l'allocation des ressources » n'est pas une « Démonstration ». En conséquence l'annotation de la phrase ne devrait pas être propagée à celles-ci. C'est-à-dire qu'il ne devrait pas y avoir d'héritage descendant. De même pour l'héritage ascendant, car le fait d'avoir une phrase étiquetée en tant que « Démonstration » ne nous permet pas d'inférer que le paragraphe parent peut lui aussi être étiqueté comme « Démonstration ».

[4] Les notions d'héritage ascendant et descendant d'annotations ont été préférées aux termes classiques utilisés dans les grammaires attribuées qui sont *attributs hérités* et *attributs synthétisés* (Aho *et al.*, 1986).

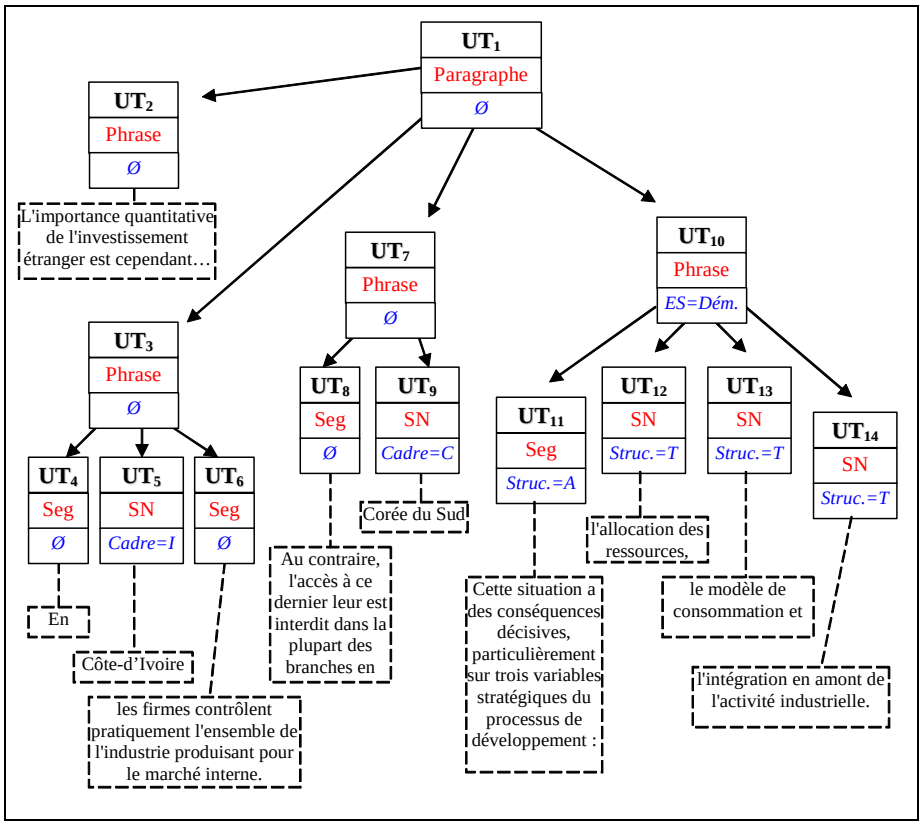

Figure 1. *Diagramme d'UT pour le texte du tableau 1*.

2.2.3. *La Tête*

Néanmoins le type de délimitation des unités présenté dans la section précédente est insuffisant pour traiter certains phénomènes linguistiques tels que la discontinuité ou le recouvrement. Plusieurs solutions ont été proposées qui reposent généralement sur les fonctionnalités offertes par X-Link et XPointer. Mais la généricité et la relative complexité de ces approches associées à l'absence d'outils d'édition sophistiqués rendent leur utilisation plutôt difficile dans le cadre du Traitement Automatique du Langage (TAL). C'est pour répondre à ce besoin, l'annotation des structures complexes rencontrées en TAL, que quatre structures, (déclarées dans la *Tête*), ont été définies (Couto, 2006). Ces quatre structures sont nommées *Ensemble*, *Séquence*, *Référence* et *Graphe* et elles permettent de déclarer de nouveaux éléments composés d'unités textuelles du *Corps* du texte. De plus, pour chacune de ces structures, des opérations de visualisation et de navigation prédéfinies sont en cours de développement.

Un *Ensemble* déclare un ensemble non ordonné d'UT pour lesquelles existe, du point de vue de l'annotateur, une relation d'équivalence. Par exemple, des UT avec des étiquettes morphosyntaxiques mais qui expriment un même thème peuvent être regroupées dans un *Ensemble*.

Une *Référence* décrit une relation orientée entre deux UT et une opération de navigation prédéfinie est associée à cet objet. Cette opération va du référé au référent. Typiquement une *Référence* permet de représenter le lien entre une anaphore et son référent discursif. D'autre part, il est possible de déclarer plusieurs références ayant la même UT référée, ce qui permet de lier toutes les anaphores à leur référent. La différence avec la déclaration d'une *Séquence* (*cf.* ci-dessous*)* tient au fait que les UT qui référent n'ont pas de liens déclarés entre elles. Un deuxième exemple est celui de la représentation des relations rhétoriques entre un noyau et un satellite proposées par la RST (Thompson *et* Mann, 1988). Un objet de type *Référence* est un moyen de représenter ce type de relation discursive.

Une *Séquence* est une suite ordonnée d'UT à laquelle l'annotateur attribue une cohésion. Les cadres thématiques (Porhiel, 2003) constituent un premier exemple de l'intérêt de cette structure puisqu'elle permet de déclarer les introducteurs de cadre comme appartenant à une même unité : *« Les introducteurs thématiques constituent une classe cohésive dont les éléments, de nature abstraite, sont morphologiquement des prépositions (*en ce qui concerne, pour ce qui est de, à propos de, sur, *etc.) et des anaphores résomptives (*à ce sujet, à ce propos*). […] Les introducteurs ont aussi pour fonction de séquencer explicitement des parties d'un texte : ils attirent l'attention sur un référent et le rendent saillant par rapport à d'autres choix possibles ; ils organisent l'information dans un texte […] »* (Couto *et al.*, 2004). En effet, la portée de ces marques, syntaxiquement non intégrées à l'énoncé où elles figurent matériellement, généralement en position initiale, peut s'étendre sur plusieurs phrases (voire paragraphes, pour certaines d'entre elles). Une *Séquence* composée par les différents introducteurs des cadres thématiques présents dans le texte permet ainsi de représenter la fonction cohésive de ces marques.

Le deuxième exemple d'utilisation d'un objet *Séquence* concerne les chaînes de référence. Une chaîne de référence est constituée par l'ensemble des syntagmes nominaux qui réfèrent à un même objet. Ainsi, dans un article de presse (*Le Figaro*, le 16 juillet 2004) sur l'amnistie fiscale[5], on trouve pour référer à « La taxe sur les fonds rapatriés en France », dix-sept corrélats linguistiques qui réfèrent au même référent dont par exemple « La taxe sur les fonds rapatriés en France », « une taxe sur les fonds placés à l'étranger et rapatriés en France », « une telle mesure », « elle », etc. La déclaration d'une *Séquence* composée de toutes ces unités textuelles, et qui concrétise la chaîne de référence lexicale, permet d'offrir au lecteur un parcours entre ces éléments en utilisant la même opération de navigation décrite précédemment.

[5] Ce texte fait partie des textes recueillis et analysés par Lita Lundquist dans le cadre du projet NaviLire.

Comme l'illustrent ces deux exemples, l'objet *Séquence* qui combine une structure avec une opération de visualisation et une opération de navigation offre les moyens de traiter simplement des phénomènes linguistiques très fréquents.

Le dernier type d'objet, *Graphe,* est utilisé pour construire des relations multiples entre des UT. Cette structure correspond exactement à la notion de graphe en mathématiques (Berge, 1958) : on distingue les *sommets* du graphe, dans le cas présent ce sont des UT déclarées dans le *corps*, et les *arcs* orientés entre ces sommets, qui représentent les relations entre les UT. Cette structure permet de représenter des phénomènes linguistiques complexes comme par exemple l'imbrication de différents cadres de discours (Charolles, 1997), les différentes pistes de cohérence qui structurent un texte (Lundquist, 1990), ou encore les expressions des sentiments d'un personnage dans un roman (Mathieu, 2005).

2.3. *Constat critique de la représentation proposée*

Toute représentation de texte est en même temps une prise de position et un choix dont il convient de mesurer les conséquences. Celle proposée ici, bien que raisonnablement indépendante d'un type d'application déterminée, est focalisée sur la navigation textuelle, qui se fonde sur l'existence dans les textes d'informations servant de guide à la lecture. Un des aspects positifs de cette représentation est de permettre de typer librement les UT, ce qui constitue un bon compromis entre flexibilité et expressivité. Certaines unités textuelles habituelles dans les textes, telles que les tables ou les énumérations, peuvent être en conséquence encodées. De plus, la possibilité de construire de nouvelles unités textuelles ne suivant pas une hiérarchie textuelle définie offre les moyens de traiter assez aisément des phénomènes linguistiques très fréquents. Un deuxième aspect concerne la lisibilité de la représentation proposée. Ainsi, la représentation dans le *Corps* du texte reste accessible et compréhensible par un lecteur humain et manipulable par des outils standard capables de visualiser des textes annotés dans le format XML. Nous nous distinguons ainsi des propositions de représentation, qui offrent des possibilités beaucoup plus puissantes et génériques (Bird *et* Liberman, 2001), comme par exemple le codage des variantes d'un texte, mais dont la transformation du texte nécessite l'usage d'outils spécialisés pour le rendre lisible par un lecteur humain.

Néanmoins, la représentation que nous proposons n'est pas exempte de critiques. La première concerne le fait que la sémantique des constituants textuels (par exemple : « cette UT est une table ») est indépendante du texte (*i.e.* la sémantique des UT n'est pas guidée par un constituant textuel déterminé, mais elle doit être correctement décodée à partir des types et annotations des différentes UT). C'est le problème classique de trouver le bon compromis entre spécificité et généralité. La deuxième critique vient du fait que les nouvelles UT construites dans la *Tête* en utilisant les constructeurs *Ensemble*, *Séquence*, *Référence* et *Graphe*, risquent, d'une part, de devenir très complexes et, d'autre part, d'introduire des erreurs ou une certaine redondance dans la représentation. A priori, la représentation n'offre pas de mécanismes de contrôle de ce type d'erreurs. La troisième critique concerne les

éléments iconographiques et certains éléments de mise en forme tels que la numérotation des pages ou les entêtes : ils ne sont pas pris en considération par la représentation, bien qu'ils soient habituellement présents dans les textes. Enfin, la représentation proposée suppose que des outils de repérage automatique des structures sont développés en amont afin d'annoter les textes ou que ces annotations sont placées manuellement (*cf.* section 5).

3. Le langage SEXTANT

Afin de pouvoir isoler les connaissances visuelles et navigationnelles, il est nécessaire de définir un formalisme pour représenter celles-ci car, faute de formalisation, il serait impossible, pour les différents agents encodeurs, de partager ces connaissances et, pour un système interpréteur, de les interpréter correctement. La navigation textuelle se fonde sur l'hypothèse selon laquelle il existe dans les textes des informations qui servent, à un utilisateur, de guide à la lecture. Les informations guidant cette lecture correspondent aux résultats d'un ensemble de traitements automatiques ou semi-automatiques d'un texte, qui se traduisent en la réécriture de ce texte selon la représentation proposée ci-dessus. Du point de vue de cette représentation, cela peut signifier plusieurs choses. Un traitement de découpage automatique en propositions (Wonsever, 2004) peut construire la hiérarchie dans le *Corps* du texte, en utilisant des UT de base. Des traitements comme ceux effectués dans le projet RÉGAL (Ferret *et al.,* 2001) (Couto *et al.,* 2004) peuvent d'une part, annoter différentes UT du *Corps* et, d'autre part, créer de nouvelles UT dans la *Tête* afin de représenter des structures telles que les cadres thématiques ou les cadres organisationnels.

Le fait de pouvoir afficher un texte de différentes manières, et que chaque manière (*vue du texte*) comporte des indications précises sur les différentes options d'affichage (*opérations de visualisation*) et sur les interactions que l'utilisateur peut effectuer (*opérations de navigation*) constitue l'épine dorsale de notre approche. De plus, une vue d'un texte n'en montre pas nécessairement tous les constituants ; il peut s'agir d'une vue partielle se focalisant sur certains aspects spécifiques ou phénomènes présents dans celui-ci. Cela constitue, en quelque sorte, la vue d'un filtrage du texte. La gestion de la coordination entre différentes vues d'un même texte peut être définie par l'encodeur en exprimant des *opérations de coordination.*

Le langage SEXTANT a pour finalité d'offrir des fonctionnalités à la fois suffisamment génériques tout en proposant une sémantique qui se focalise sur l'essentiel du processus de visualisation et de navigation coordonnée dans les textes, à l'inverse de langages de transformation ou de programmation comme, par exemple, XSLT (EXtensible Stylesheet Language). Dans sa version actuelle, le langage SEXTANT est de type déclaratif et propose des opérations prédéfinies de visualisation et de navigation. La sémantique formelle du langage est intégralement décrite dans (Couto 2006) où toutes les opérations sont définies dans un format de « règles » inspiré de la déduction naturelle en logique.

3.1. *Les vues d'un texte*

Afin de présenter une approche systématique des différentes vues, nous proposons une classification selon leur *type* et leur *contenu*. Les types possibles sont : *linéaire*, *arborescent* et *graphe* tandis que les contenus possibles sont : les *chaînes lexicales* et les *annotations*. Il en résulte qu'il existe six combinaisons possibles. Cependant, une vue de type linéaire où le contenu correspond aux annotations ne semble pas particulièrement utile, et elle ne sera pas prise en considération. Certes, d'autres types de vues que ceux présentés ici sont envisageables, comme les vues basées sur la technique *« Focus + Context »* (Lamping *et* Rao, 1996) (Dieberger *et* Russell, 2002), par exemple ; ou d'autres plutôt *ad hoc* comme la vue « docball » (Crestani *et al.*, 2002), qui montre la structure hiérarchique d'un document. Néanmoins, le choix des types *linéaire*, *arborescent* et *graphe* correspond à la représentation de texte proposée, et constitue, de notre point de vue, un bon point de départ, pouvant s'enrichir des propositions et des développements postérieurs.

3.1.1. *Vue linéaire*

La vue la plus simple d'un texte est la vue *linéaire*, où celui-ci est représenté comme une suite de chaînes de caractères, cette suite étant le résultat d'un parcours en profondeur dans la hiérarchie d'UT de base présentes dans le *Corps* du texte. Notons que si une UT comporte un titre, celui-ci est affiché en premier, selon une chaîne calculée également à partir d'un parcours en profondeur dans les UT constituant le titre. Étant donné la nature plate de la représentation, il n'est pas possible de visualiser les unités textuelles complexes pouvant être exprimées dans la *Tête* du texte. Pour cette vue, il est important d'indiquer les éléments textuels (types d'UT) constituant des *séparateurs* visuels.

3.1.2. *Vue arborescente*

Un deuxième type de vue montre le contenu du texte, c'est-à-dire les UT de base déclarées dans le *Corps* du texte, de manière arborescente, cette arborescence reflétant la hiérarchie d'UT existante. La nature hiérarchique de cette vue ne permet pas de représenter les unités textuelles complexes exprimées dans la *Tête* du texte. Selon le contenu, les libellés à afficher seront les chaînes lexicales des UT ou les annotations de celles-ci. La figure 2 montre un exemple de vue de type arborescent dont le contenu visualise les annotations utilisées dans le cadre du projet RÉGAL (Couto *et al.*, 2004). Notons que dans cette figure, toute UT comporte ou bien une annotation (dont la valeur s'affiche en utilisant un jeu de couleurs selon une carte de couleurs définie par l'utilisateur) ou bien aucune.

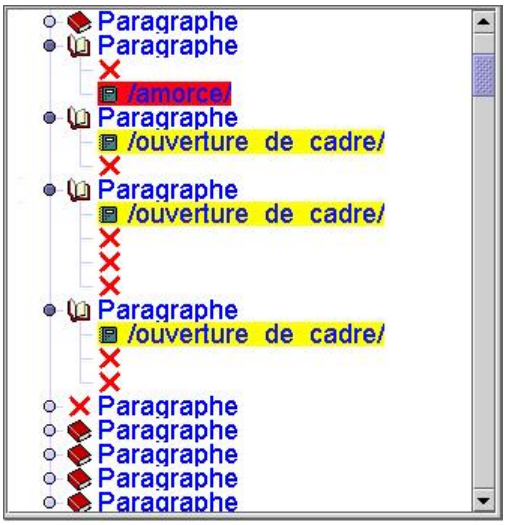

Figure 2. *Exemple d'une vue de type arborescent (extrait de Couto* et al.*, 2004).*

3.1.3. *Vue graphe*

Ce type de vue affiche toutes les UT d'un texte, en montrant les rapports entre elles. Ce rapport est déterminé à partir du rapport hiérarchique exprimé dans le *Corps* du texte, et à partir des unités textuelles complexes exprimées dans la *Tête* du texte. En conséquence, c'est cette vue qui montre la structuration globale du texte. De même que pour la vue précédente, les libellés à afficher seront les chaînes lexicales des UT ou les annotations de celles-ci. Notons, d'une part, que la vue de type arborescent est un cas particulier de cette vue. D'autre part, la vue graphe peut devenir très complexe à visualiser et la navigation s'avérer difficile[6].

3.2. *Modules de connaissances et descriptions de vue*

Les éléments constitutifs d'une vue sont spécifiés dans une *description de vue*. Plusieurs descriptions de vues peuvent être rassemblées dans une entité cohérente, d'après l'encodeur des connaissances, nommée *module de connaissances*. Nous pouvons concevoir la création d'une vue comme l'application d'une description de vue à un texte déterminé. Par analogie, l'application d'un module de connaissances à un texte implique la création d'un ensemble de vues. En conséquence, toute vue est

[6] On peut trouver à l'adresse http ://www.visualcomplexity.com/vc/index.cfm, un ensemble d'exemples de complexité de visualisation lorsque l'espace d'information est vaste. Il s'agit de représentations graphiques des réseaux de connaissances, et parmi ces exemples, on en trouve plusieurs correspondant à des vues de graphes en deux dimensions, similaires au type de vue proposé ici.

liée à un texte, à une description de vue et, indirectement, à un module de connaissances. Une description de vue est identifiée dans le module par son nom. Afin de la définir, l'encodeur doit indiquer :

- le type de vue selon les types de vues disponibles : linéaire, arborescent et graphe ;

- le contenu de la vue selon les contenus disponibles : chaînes lexicales et annotations ;

- ses paramètres, selon le type de représentation ;

- ses contraintes de création (*i.e.* des conditions d'appartenance à la vue, à vérifier par les unités textuelles du texte) ;

- un ensemble d'opérations de visualisation ;

- un ensemble d'opérations de navigation ;

- un ensemble d'opérations de coordination.

Le fait de pouvoir créer des vues partielles d'un texte introduit le besoin de contraintes qui vont être exprimées à l'aide du langage de conditions.

3.3. *Le langage de conditions*

Le langage de conditions constitue une partie importante du langage SEXTANT. Par exemple, on formule une condition pour exprimer des contraintes d'appartenance d'une UT à une vue, pour indiquer les UT sur lesquelles une mise en relief s'applique, ou bien pour préciser la cible et la source dans la description d'une opération de navigation. Le langage de conditions est composé de *conditions simples*, de *conditions d'existence sur les éléments des UT* et de *conditions sur la hiérarchie.*

existeAnnotations : *teste si l'ensemble d'annotations d'une UT n'est pas vide ;* existeChaîneLexicale *: teste si la chaîne lexicale d'une UT est définie ;* existeTitre *: teste si le titre d'une UT n'est pas vide ;* existeParent *: teste si une UT a une UT parent ;* existeFils : *teste si la suite d'UT filles d'une UT n'est pas vide.*

Tableau 2. *Opérateurs d'existence sur les éléments des UT.*

Les *conditions simples* portent sur les attributs et sur les annotations des UT. Pour ce type de conditions, nous utiliserons une notation proche de la notion de patron. On définit un opérateur *UT* comportant cinq opérandes qui correspondent aux propriétés suivantes d'une UT : le *Type*, le *Numéro*, le *Rang*, les *Annotations* et la *chaîne lexicale*. Avec les trois premiers opérandes on dénote des contraintes d'égalité, d'inégalité, d'ordre (inférieur et supérieur), de préfixe, de suffixe et de recherche de sous-chaîne. De même pour le cinquième opérande. Le quatrième

opérande est utilisé pour indiquer l'existence ou non-existence d'annotations, que ce soit un nom d'annotation, une valeur ou un couple nom d'annotation – valeur.

estParent *: teste si une UT est le parent dans la hiérarchie d'UT d'une UT décrite en utilisant une condition simple ;* estFils *: teste si une UT est le fils dans la hiérarchie d'UT d'une UT décrite en utilisant une condition simple ;* estFrère *: teste si une UT est le frère dans la hiérarchie d'UT d'une UT décrite en utilisant une condition simple ;* estAscendant *: teste si une UT est l'ascendant dans la hiérarchie d'UT d'une UT décrite en utilisant une condition simple ;* estDescendant *: teste si une UT est le descendant dans la hiérarchie d'UT d'une UT décrite en utilisant une condition simple ;* contientDansTitre *: teste si une UT contient dans les UT du titre une UT décrite en utilisant une condition simple ;* estDansTitreDe *: teste si une UT appartient aux UT du titre d'une UT décrite en utilisant une condition simple.*

Tableau 3. *Opérateurs portant sur le rapport hiérarchique des UT***.**

Pour les *conditions d'existence* d'UT, un opérateur sans arguments est défini pour chaque élément (*cf.* tableau 2.). Pour les conditions où se joue le rapport entre les UT dans la hiérarchie, des opérateurs unaires spécifiques sont définis. Ces opérateurs prennent comme argument une condition simple. Le tableau 3 montre les opérateurs définis pour tester des conditions sur le rapport hiérarchique des UT. Il convient de noter que ces opérateurs permettent de parcourir intégralement, à partir d'une UT, l'arbre qui représente la structure du texte (Couto, 2006). Les conditions peuvent se combiner en utilisant les opérateurs classiques *OU*, *ET* et *NON*, de la logique. Le tableau 4 présente un exemple d'expression du langage qui exprime la condition suivante : « Les UT de type *SN* comportant une annotation de nom *Référent discursif*, tel qu'il existe dans les ascendants une UT de type *Paragraphe* qui ne comporte pas une annotation de nom *Étiquette Sémantique* et de valeur *Conclusion* ».

En résumé, le langage de conditions permet d'exprimer pour une UT quelconque, d'une part, des conditions sur les annotations de cette UT, et, d'autre part, des conditions sur la position de cette UT dans la structure hiérarchique du texte.

UT(Type = *SN*, *,*,{(*Référent discursif*, *)},*) ET estDescendant(UT(Type = *Paragraphe*,*,*,{ ¬∃(*Étiquette Sémantique*, *Conclusion*)},*))

Tableau 4. *Exemple d'utilisation du langage de conditions*

3.4. *Les opérations de SEXTANT*

3.4.1. *Les opérations de visualisation*

Les connaissances de visualisation modélisées sont considérées ou bien comme des opérations de transformation qui s'appliquent sur une UT ou bien comme des opérations d'affichage d'information relative à une UT. Elles s'inspirent des propositions de (Couto, 2002) et peuvent se diviser en deux types principaux : des opérations de *mise en relief* et des opérations d'*aide contextuelle*. Les opérations de mise en relief peuvent, d'une part, changer l'apparence sur l'écran de la chaîne textuelle (colorisation, taille de la police, etc.), et, d'autre part, permettent de modifier la chaîne textuelle, en ajoutant des informations lexicales. Une opération basique de mise en relief est définie par : i) l'élément textuel à mettre en relief ; ii) une liste d'opérations de transformation.

Les opérations d'aide contextuelle utilisent des techniques de type *infobulle* pour transmettre à l'utilisateur des informations sur les UT du texte. Une opération d'aide contextuelle est définie par : i) le type d'aide contextuelle ; ii) l'élément textuel qui déclenche l'opération ; ii) l'aide à afficher. Nous avons défini deux types d'aide contextuelle : l'*affichage d'infobulles* et l'*affichage dynamique de vues*. L'UT déclenchant l'opération est signalée en exprimant la condition qu'elle doit vérifier. L'aide à afficher dépend du type d'aide contextuelle. Dans le cas d'affichage d'infobulles, il s'agit d'une *chaîne lexicale*. Dans le cas d'affichage dynamique de vues, il s'agit d'une *description de vue* ou du *nom* d'une description de vue existante dans le module.

3.4.2. *Les opérations de navigation*

La navigation est conceptualisée comme une opération reliant une UT *source* avec une UT *cible*. La manière dont ces deux UT sont liées est fonction de quatre paramètres : i) la condition à vérifier par l'UT source ; ii) la condition à vérifier par l'UT cible ; iii) le type d'opération de navigation ; iv) la relation existant entre l'UT source et l'UT cible.

IF (condition UT_{source})
THEN : DO SELECT CRITERIA (Orientation, Ordre)
WHERE {(condition UT_{cible})
AND
(Relation (UT_{source}, UT_{cible})
} ;
: DO SHOW (Libellé de l'Opération) ;

Tableau 5. *Opération de navigation générique.*

Une opération de navigation est définie comme une opération qui cherche l'UT cible à partir de l'UT source, en vérifiant une condition déterminée (exprimée dans

le langage de conditions présenté ci-dessus) et en suivant l'orientation et l'ordre spécifiés par le type d'opération (*cf.* tableau 5). La source est définie en utilisant une condition sur les UT. Implicitement, une opération de navigation est *disponible* pour une UT déterminée si celle-ci vérifie la condition exprimée par la source. La cible est déterminée à partir de deux paramètres : une condition à vérifier pour l'UT cible et le type d'opération de navigation qui combine orientation et ordre dans la recherche. Une fois la source déterminée, plusieurs UT peuvent vérifier la condition de la cible, et c'est le type d'opération qui indique laquelle choisir parmi celles-ci. Chaque opération est donc typée avec une valeur qui appartient à l'ensemble {*premier, dernier, suivant[i], précédent[i]*}, *i* étant un nombre entier positif. Ces valeurs spécifient d'une part l'orientation, c'est-à-dire dans quel sens (avant ou après l'UT source) doit être effectué la recherche de l'UT cible, et, d'autre part, le référentiel, absolu (*premier, dernier),* ou relatif (*suivant[i], précédent[i]*), par rapport à la source. Dans le cas d'un référencement relatif, l'index *i* permet de spécifier le rang de la cible recherchée. Par ailleurs il est possible de spécifier que la recherche de la cible ne se fait pas en suivant l'ordre des UT dans le texte (ordre narratif), mais un ordre (déclaré par l'intermédiaire d'un objet *Séquence*) imposé par le concepteur du *module de connaissances.* Cette fonctionnalité du langage est utilisée dans l'expérimentation de lecture chronologique des sentiments d'un personnage d'un roman (Mathieu, 2005).

Dans sa première version, la puissance d'expression du langage était limitée par la nécessité d'exprimer de manière absolue les conditions sur les valeurs des attributs des UT. Cette limitation avait par exemple pour conséquence l'obligation d'écrire une opération de navigation différente pour naviguer entre chaque anaphore et son référent discursif. Dernièrement, nous avons enrichi le langage de conditions par la possibilité d'exprimer des relations entre les valeurs des attributs des UT de la source et de la cible, ce qui entraîne qu'une seule opération de navigation suffit pour traiter la navigation évoquée ci-dessus. Par exemple, il suffit d'indiquer que la source et la cible portent sur le même référent discursif, sans avoir besoin d'indiquer explicitement le référent discursif sur lequel elles doivent porter.

3.4.3. *Les opérations de coordination*

Puisqu'un même texte peut avoir différentes vues, il est important de pouvoir coordonner celles-ci, c'est-à-dire que le fait de changer l'UT active d'une vue (que ce soit par la sélection explicite d'une UT ou par l'exécution d'une opération de navigation), implique le changement de l'UT active d'autres vues.

Supposons que nous ayons une vue de type linéaire et une autre de type carte sur cette vue linéaire. Si un changement de l'UT active dans la première vue se répercute automatiquement sur la deuxième, nous avons toujours, grâce à la deuxième vue, un repère visuel du positionnement global de l'UT. Néanmoins, si nous avons une vue partielle d'un texte (*i.e.* une vue contenant seulement un sous-ensemble de toutes les UT du texte), il peut être intéressant de ne pas faire automatiquement la coordination avec une vue totale du même texte. L'utilisateur

pourrait souhaiter parcourir (naviguer) la vue partielle (par exemple la vue d'un extrait), vouloir mettre en contexte une UT dans la vue totale, et demander manuellement la coordination. Notons que si la coordination est automatique, le système interpréteur est chargé de la réaliser, tandis que si elle est manuelle, le système doit fournir des mécanismes pour sélectionner les opérations de coordination à exécuter. Nous travaillons actuellement à la formalisation des opérations de coordination.

4. La plate-forme NaviTexte

Afin de vérifier la validité et l'intérêt de nos hypothèses concernant la navigation textuelle nous avons entrepris la réalisation d'une plate-forme logicielle, nommée NaviTexte, dédiée à la navigation textuelle Cette implémentation comporte deux aspects. Tout d'abord, l'implémentation informatique de la représentation des textes proposée dans la section 2 et celle de SEXTANT, le langage de modélisation des connaissances décrit dans la section 3. Puis, la conception et le développement d'un environnement capable de traiter les textes, d'interpréter le langage SEXTANT et de gérer l'interaction avec l'utilisateur.

4.1. *Architecture conceptuelle et logicielle*

D'un point de vue conceptuel, la plate-forme NaviTexte est constituée par trois sous-systèmes principaux. Le *système de gestion des textes*, le *système de gestion des modules*, et le *système de gestion des interactions*.

Le *système de gestion des textes* se charge d'offrir toutes les fonctionnalités concernant la représentation informatique du texte décrite ci-dessus. Rappelons que d'après celle-ci, les textes sont encodés sous forme de fichiers XML, selon une DTD définie. Le chargement d'un fichier par le système entraîne la création en mémoire d'une instance du texte selon une représentation interne. Cette représentation prend en considération (*i.e.* interprète) des éléments XML ayant une sémantique particulière vis-à-vis du texte, tels que les constructeurs déclarés dans la partie *Tête* (*cf.* section 2.2.3). De même pour le traitement des titres. C'est ce sous-système qui offre les fonctions nécessaires afin de parcourir la structure d'un texte : hiérarchie d'UT dans le corps, les UT « construites » dans la partie *tête*, les titres, etc.

Le *système de gestion des modules* permet de charger les modules de connaissances en mémoire et de manipuler ceux-ci. Pour ce faire, une représentation des modules en mémoire a été définie équivalente à celle d'un arbre. La représentation en mémoire tient compte des différents éléments du langage et c'est ce sous-système qui s'occupe, entre autres, d'interpréter le langage de connaissances, y compris le langage de conditions.

Quant au *système de gestion des interactions*, celui-ci se charge de tout ce qui concerne l'interaction avec l'utilisateur, et agit comme une interface entre les deux

sous-systèmes précédents. À titre d'exemple, ce sous-système est chargé d'offrir les différentes opérations de navigation sur une UT (lorsque l'utilisateur clique sur celle-ci), de capturer le choix de l'utilisateur, de faire exécuter l'opération, d'appliquer le résultat de cette exécution (*i.e.* changer éventuellement d'UT active, modifier l'historique de navigation…), etc.

À partir de l'architecture conceptuelle présentée, plusieurs architectures logicielles sont possibles. Nous avons opté (Couto, 2006) pour une architecture du type Model-View-Controller (MVC) (Burbeck, 1987). Par ailleurs, afin d'optimiser les traitements plusieurs aspects ont été pris en compte. Le premier concerne les structures de données. Bien que les textes et les modules des connaissances soient encodés sous forme de fichiers XML, lorsque ceux-ci sont chargés en mémoire, le système analyse le fichier et construit l'arbre DOM correspondant ainsi que plusieurs dictionnaires en vue d'optimiser les recherches dans l'arbre.

Un deuxième critère d'optimisation concerne les opérations de navigation. Étant donné que ni le texte ni le module de connaissances ne changent après que la vue est créée (ou bien si ceux-ci changent, ces changements ne sont pas répercutés sur les vues de manière dynamique tant que l'utilisateur ne redemande pas le chargement du module), l'ensemble du parcours reste figé au moment de la création d'une vue. Il en résulte que le système pourrait calculer, avant de présenter les vues à l'utilisateur, le résultat de l'application de toutes les opérations disponibles pour chaque UT appartenant à la vue. Néanmoins, ce calcul peut devenir complexe et potentiellement non nécessaire car il se peut que l'utilisateur n'exécute qu'un nombre très réduit de toutes les combinaisons possibles. Afin d'optimiser ce calcul, un système de trace de résultats, s'inspirant des techniques de programmation dites paresseuses (*lazy)*, a été mis en place.

4.2. *Interactions avec l'utilisateur*

La figure 3 illustre le modèle général de l'écran principal de NaviTexte. L'écran est divisé en trois panneaux, une barre de menus, une barre d'outils et une barre d'état. Le panneau placé en haut à gauche visualise une vue condensée du texte où les éléments correspondent aux annotations. Le panneau placé en bas à gauche visualise l'information sur les textes et des modules de connaissances ouverts, ainsi que l'information des vues créées. Le panneau placé à droite contient les différentes vues des textes accessibles via des onglets. Une barre d'outils propose les opérations principales de la plate-forme.

Dans une vue déterminée, une UT peut avoir trois états vis-à-vis de la navigation textuelle : i) l'état d'une UT sans annotations et, en conséquence, sans opérations de navigation déclenchables ; ii) l'état d'une UT avec annotations mais qui ne vérifie la condition d'aucune des opérations de navigation décrites dans la description de vue correspondant à la vue ; iii) l'état d'une UT avec opérations de navigation déclenchables (*i.e.* une UT pour laquelle au moins une condition des sources des

opérations de navigation se vérifie). La *souris sémantique* est une option configurable différenciant sous la forme d'icônes, ces trois états possibles. Ainsi, au survol de la souris, lorsque l'utilisateur se positionne sur l'objet représentant une UT, le caret de la souris change de forme selon l'état de l'UT.

De la même manière, il existe trois états possibles pour les opérations de navigation : i) soit l'opération n'est pas exécutable, c'est-à-dire que bien que l'UT vérifie la condition de la source de l'opération, il n'existe pas d'UT vérifiant la condition de la cible ; ii) soit l'opération est exécutable et elle n'a pas été exécutée ; iii) soit l'opération est exécutable et elle a été déjà exécutée.

Lorsque l'utilisateur clique sur une UT ayant des opérations de navigation déclenchables (*i.e.* une UT qui vérifie la condition d'au moins une source de toutes les opérations de navigation définies pour la vue), un menu lui est offert, affichant les opérations de navigation associés à l'UT actuelle. La mise en forme de chaque option correspondant à une opération de navigation indique son état. Dans l'exemple de la figure 3, la première option, libellée « Lire_Thématique », est grisée car elle n'est pas exécutable. La deuxième et la quatrième options, libellées respectivement « Lire_Récapitulation » et « Lire_Conclusion », s'affichent en gras car elles sont exécutables et n'ont jamais été exécutées. Enfin, la troisième option, libellée « Lire_Argumentation » et affichée dans une police de style normal, renvoie à une opération de navigation exécutable qui a déjà été exécutée.

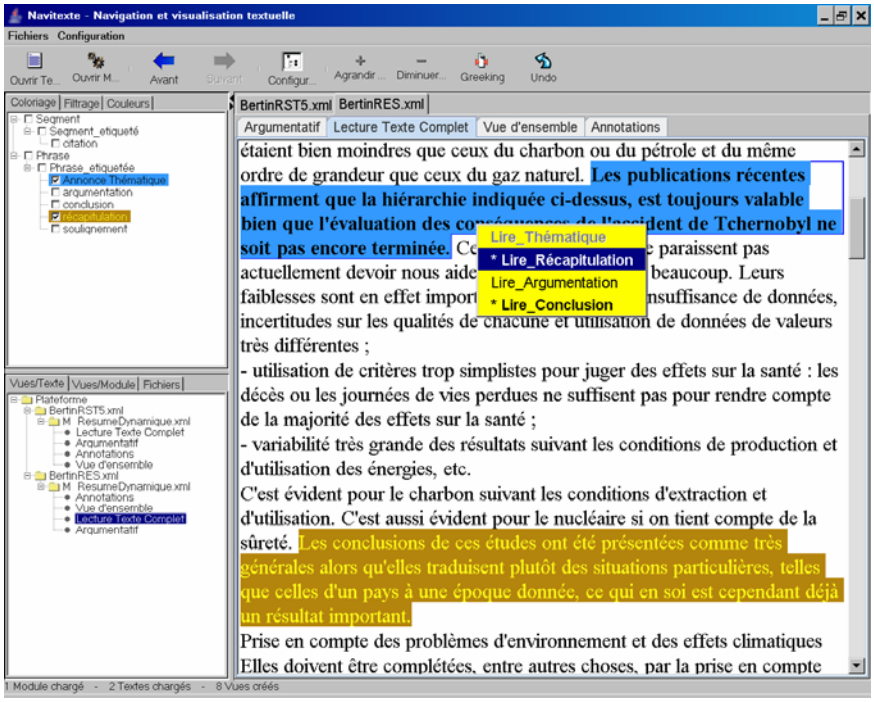

Figure 3. *Différents états des opérations de navigation (extrait de Couto, 2006).*

Rappelons que différentes vues graphiques d'un texte peuvent être visualisées. Celles-ci sont classifiées selon leur type et leur contenu. Les types possibles sont : linéaire (texte plat), arborescente et graphe tandis que les contenus possibles sont : les chaînes lexicales et les annotations. De plus, il existe des vues condensées et des vues de type « *carte* ». La technique de « *greeking* » est l'une des techniques utilisées en visualisation d'informations pour afficher un texte dans un format réduit. L'utilisation de cette technique permet de visualiser un texte selon une vue de type « carte » et d'avoir une vision globale du texte. Enfin, en ce qui concerne l'aide contextuelle, une option configurable, de type affichage d'infobulles a été implémentée.

5. Expérimentations et évaluation

Afin de vérifier la validité de nos hypothèses nous avons entrepris plusieurs expérimentations dont une, le projet *NaviLire* (*cf.* 5.2.), a donné lieu à une application pédagogique. Les trois expérimentations que nous présentons ci-dessous nous ont aussi permis d'évaluer la pertinence du modèle de représentation des textes proposé, la puissance d'expression du langage SEXTANT et la validité des choix concernant la gestion des interactions avec l'utilisateur. Ces expérimentations recouvrent des finalités très différentes, qui vont du parcours de lecture comme procédé de résumé automatique, à l'apprentissage du français comme langue seconde, ainsi qu'à l'exploration des sentiments éprouvés par un des personnages dans un roman. Nous les présentons ci-dessous avant de synthétiser quelques éléments d'évaluation sur la plate-forme.

5.1. *Une alternative au résumé automatique : le parcours de lecture*

Un grand nombre de systèmes de résumé automatique ont été proposés ces dernières années (Mani, 2001 ; Minel, 2003). Tous ces systèmes, fondés sur le principe de l'extraction de phrases, sont confrontés à deux problèmes intrinsèques au procédé d'extraction : d'une part à la rupture de la cohésion textuelle, comme par exemple la présence d'anaphores sans leur référent discursif, d'autre part à l'adaptation du résumé aux besoins spécifiques d'un lecteur. Jusqu'à présent ces problèmes n'ont pas reçu de solutions totalement satisfaisantes. Une autre approche consiste à considérer le processus de résumé comme un cheminement - plus exactement un parcours de lecture (Minel, 2002) - dans le texte source, qui soit propre au lecteur. Ainsi, plutôt que de construire des fragments textuels, nous proposons des parcours de lecture spécifiques. Les spécifications des parcours de lecture sont fondées sur les propositions de (Endres-Niggemeyer *et al.*, 1995) et sur les observations d'usage effectuées dans le cadre de l'évaluation du système de résumé SERAPHIN (Minel *et al.*, 1997) et de la plate-forme (Filtext Minel *et al.*, 2001). Ces travaux tendent à montrer qu'un « résumeur » humain professionnel d'une part s'intéresse, à certaines catégories discursives qu'ils repèrent en exploitant à la fois l'organisation du texte et des marques lexicales et, d'autre part, navigue

dans les textes en appliquant des heuristiques acquises par expérience. Par exemple, commencer par lire la conclusion, puis rechercher dans l'introduction des syntagmes nominaux qui figurent dans la conclusion.

Par conséquent, un parcours de lecture spécifie le type de catégorie discursive que cherche un lecteur (par exemple, une conclusion, une définition, une argumentation, une hypothèse, etc.) et l'enchaînement dans lequel les segments qui énoncent linguistiquement ces catégories (en général les phrases) doivent lui être présentés. La mise en œuvre de ces parcours de lecture présuppose le repérage de ces catégories discursives et l'annotation du texte source. Pour ce faire, nous avons utilisé la plate-forme ContextO (Crispino, 2003) mais d'autres systèmes d'annotation automatique pourraient être utilisés, par exemple Linguastream (Bilhaut, 2003) ou Gate (Cunningham *et al*., 2002). Il convient par ailleurs de noter que ces plates-formes n'offrent pas de fonctionnalités qui permettent de programmer des opérations de navigation.

Un exemple de parcours de lecture orienté est présenté dans la figure 3. Le point de lecture initial est positionné sur la première UT (phrase) annotée « Annonce thématique » (les différentes valeurs sémantiques annotant le texte correspondent aux résultats de plusieurs travaux, voir (Minel *et., al.* 2001) pour plus de détails). À partir de ce point, quatre opérations de navigation sont proposées. Une opération qui déplacera le point de lecture vers l' « Annonce Thématique » suivante, les trois autres opérations proposant respectivement de se déplacer vers la première « Conclusion », la première « Récapitulation », la première « Argumentation ». Ainsi, dans la continuité de sa lecture du texte, le lecteur se voit proposer, par une signalétique spécifique, des parcours spécifiques sans rupture de la cohésion textuelle puisqu'il peut voir à tout instant le texte complet, ce qui lui permet entre autres d'assurer la continuité référentielle (Battistelli *et* Minel, 2006).

5.2. *Enseignement du français en langue seconde*

L'utilisation de la navigation textuelle à des fins pédagogiques au travers du projet *NaviLire*, est le fruit d'une collaboration entre Lita Lundquist et nous-mêmes ; en conséquence, nous reprenons brièvement ci-dessous les principaux concepts et résultats exposés dans les articles cités. Des explications plus détaillées sur les hypothèses linguistiques sous-jacentes et sur les interactions proposées par le logiciel *NaviLire* peuvent être consultées respectivement dans (Lundquist, 90) et (Couto *et al.*, 2005, Lundquist *et al.*, 2006).

Par le procédé, par lequel le lecteur apprend à naviguer dans un texte en suivant ses différentes pistes de cohérence – basées sur la référence, sur la prédication et sur les connecteurs – nous nous attaquons à des problèmes cognitifs cruciaux pour lire, comprendre et interpréter correctement un texte, ainsi que pour apprendre par les textes. Le premier problème consiste à identifier les référents discursifs d'un texte et à établir les relations correctes entre les SN qui y réfèrent. Le second problème cognitif consiste à identifier le « où veut en venir l'émetteur » du texte. Cette

orientation – expressive, argumentative, ou autre – a été qualifiée de « programme d'interprétation » (Lundquist, 1990), étant donné qu'il s'agit d'une orientation marquée dès le début du texte. Cette identification de l'orientation, apportée entre autres par les prédications, est primordiale pour un déchiffrage correct de la cohérence sémantique et pragmatique du texte. Finalement, les connecteurs soulignent les relations rhétoriques à établir entre des propositions ou autres séquences du texte, ce qui contribue, évidemment, de manière essentielle à établir les relations nécessaires pour construire la représentation mentale correcte du texte, c'est-à-dire, de son contenu et de son acte illocutoire prédominant (tels informer, persuader, convaincre, narrer, décrire, etc.).

Dans le cadre du projet *NaviLire*, pour naviguer dans l'objet texte, nous avons isolé des unités textuelles qui permettent de spécifier des opérations de navigation, ce qui équivaut à établir des liens de cohérence entre des unités de même nature. Comme les éléments textuels appartiennent à des types différents, la navigation permet d'une part de suivre des pistes de cohérence différentes dans un même texte, et, d'autre part, d'en identifier les réalisations linguistiques dans une langue donnée (ici, et pour le moment, le français). Plutôt que de manipuler des structures textuelles hiérarchiques (Couto *et* Minel, 2004), nous distinguons ici des pistes parallèles de marques textuelles qui chacune contribue à un type particulier de cohérence.

Jusqu'à présent, *NaviLire*[7] a été mis en pratique auprès d'étudiants danois de quatrième année d'études dans le cursus de « Language and Communication » à « The Copenhagen Business School ». Un « pilote » a permis de réaliser une première expérience auprès de quatorze étudiants, divisés en deux groupes, les « Papiristes » qui lisent le texte en utilisant les méthodes traditionnellement utilisées dans ce cursus, et les « NaviListes » qui lisent le même texte avec *NaviLire*.

	Nombre de questions	Pourcentage
Performances des « Navilistes » supérieures au « Papiristes »	14	40
Performances des « Navilistes » identiques au « Papiristes »	16	45,7
Performances des « Navilistes » inférieures au « Papiristes »	5	14,3
Total	35	100

Tableau 6. *Comparaison des performances entre « NaviListes » et « Papiristes »*

[7] Depuis septembre 2006, le logiciel connaît une diffusion publique sous la forme d'un CD qui accompagne l'ouvrage de (L. Lundquist, 2006).

Les premiers résultats (*cf.* tableau 6), fondés sur les réponses d'un questionnaire composé de 40 questions, dont 35 sur le contenu du texte, (voir Lundquist *et al.*, 2006 pour le détail de l'expérience) montrent que les « Navilistes » ont une performance (mesurée par le nombre de bonnes réponses aux questions) de compréhension du texte qui est supérieure au « Papiristes » pour 14 questions, identique pour 16 autres questions et inférieure pour 5 questions. Une deuxième expérimentation sur une promotion complète d'étudiants et sur une année scolaire pleine est en cours.

5.3. *En relisant Madame Bovary*

Cette expérimentation, visant la lecture du roman « Madame Bovary » (Gustave Flaubert) en suivant les sentiments des personnages, a été développée en collaboration avec Yannick Mathieu du Laboratoire de Linguistique Formelle (UMR 7110, CNRS – Université Paris 7). L'annotation des sentiments dans le roman est fondée sur un lexique d'environ 1 200 termes dénotant sentiments, émotions et états psychologiques sous forme de verbes comme *aimer*, *effrayer*, etc., substantifs comme *amour*, *colère*, etc., et adjectifs comme *amoureux*, *jaloux*, etc. Ces termes sont regroupés en 38 classes sémantiques : *amour*, *amusement*, *déception*, *déprime*, *indifférence*, *peur*, etc. (Mathieu, 2000). De plus, il existe trois catégories de termes : *négatif*, *positif* et *neutre*. Il existe également des relations de *sens*, d'*intensité* et d'*antonymie* entre les classes sémantiques, lesquelles peuvent se représenter en utilisant des graphes.

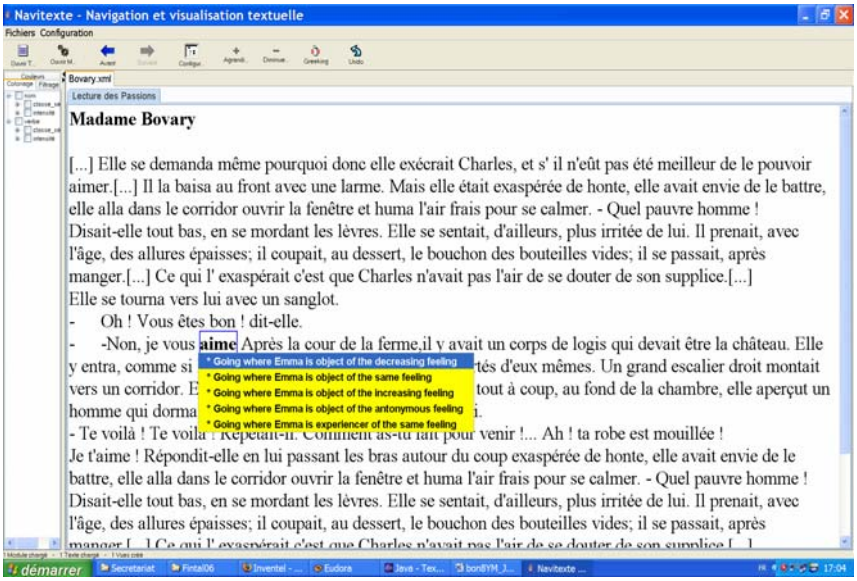

Figure 4. *Exemple de pistes de lecture pour lire Madame Bovary (extrait de Mathieu, 2006).*

L'orientation des arcs dans les graphes dépend de l'intensité du sentiment expérimenté. Trois concepts participent aux relations : la notion d'*intensifieur*, la notion d'*expériencer* et celle *d'objet*. L'*intensifieur* est une propriété dont la valeur par défaut est *neutre*, qui sert à dénoter, en affectant sa valeur à *fort*, que dans une même classe sémantique il existe des termes dont l'intensité est plus marquée. Par exemple, pour les termes *exaspérer* et *irriter*, appartenant à la même classe sémantique, *l'intensifieur* du premier terme est affecté à *fort* tandis que celui du deuxième est affecté à *neutre*. L'*expériencer* est l'individu qui éprouve un sentiment et l'*objet* est l'objet que concerne ce sentiment. En guise d'exemple, dans la phrase « Paul effraye Marie », l'*expériencer* est Paul et *l'objet* est Marie.

Une UT générique de type « Marque Sentiment » est utilisée, regroupant les trois catégories syntaxiques possibles. Les UT annotées dans le roman contiennent les informations suivantes indiquées sous forme d'annotations : catégorie grammaticale, classe sémantique, *intensifieur*, *expériencer* et *objet*. L'annotation du texte est réalisée semi-automatiquement en s'appuyant d'une part sur le logiciel Syntex (Bourigault, 2002) et d'autre part, sur les ressources linguistiques de (Mathieu, 2000, 2005). Une étape manuelle corrige certaines erreurs et résous les problèmes de co-référence. Actuellement, cinq pistes de lecture existent (*cf.* figure 4), pour chaque personnage principal du roman (Emma, Rodolphe, Charles et Léon).

5.4. *Évaluation critique*

Ces trois expérimentations nous ont permis d'évaluer le modèle de représentation du texte, le langage SEXTANT et les interfaces de navigation qui sont proposées aux utilisateurs. En ce qui concerne le modèle de représentation du texte, les résultats confirment nos hypothèses : les différentes relations convoquées, syntagmatiques, syntaxiques, sémantiques et discursives ont pu être facilement représentées tout en conservant une bonne lisibilité. Il convient néanmoins de pondérer ce constat en remarquant que ces trois expérimentations ne convoquent pas de représentations intratextuelles complexes. En ce qui concerne le langage SEXTANT, les résultats sont là aussi satisfaisants, puisque toutes les opérations de navigation demandées par les experts ont pu être exprimées. Néanmoins, certaines limitations du langage de conditions sur lequel s'appuie le langage SEXTANT sont apparues. Notamment, il n'est pas possible d'exprimer une condition sur la position relative d'une UT cible dans le texte, autrement qu'en faisant référence à l'UT source. Si aucune des expérimentations n'a nécessité l'expression de telles conditions, il n'en reste pas moins que ceci pourrait constituer une limitation pour certains développements futurs. Quant à la gestion des interactions, le menu actuel de choix des opérations de navigation satisfait les besoins essentiels, mais sa position dans le texte ne peut pas être paramétrée, il se positionne toujours sur l'UT source, ce qui rend la lecture quelquefois difficile.

Ces expérimentations nécessitent l'expression d'opérations de navigation qui convoquent des expertises très différentes. L'expérimentation *NaviLire* a bénéficié de la longue expérience pédagogique et des réflexions théoriques de L. Lundquist

sur l'enseignement de la linguistique textuelle. Nous disposions ainsi d'une documentation abondante (exemples d'exercices sur support imprimé, ouvrages publiés) que nous avons aisément transcrite sous la forme d'opérations de navigation. De même, les structures textuelles ont été balisées (manuellement) par L. Lundquist à partir de ces mêmes exercices. Pour l'expérimentation sur le résumé, nous avons déjà mentionné les travaux sur lesquels nous nous sommes appuyés et il faut simplement remarquer que les structures textuelles sont repérées automatiquement par un système dédié au filtrage d'informations. En revanche, les opérations de navigation, même si elles sont issues de divers travaux ont été spécifiées empiriquement et une évaluation en usage reste à faire. Enfin, l'expérimentation sur la navigation dans le roman « Madame Bovary » est la plus prospective mais elle a le mérite de montrer que des relations de types syntaxiques et ontologiques peuvent être représentées dans le formalisme proposé par NaviTexte.

6. Conclusion

Nous avons montré que la navigation dans un texte, processus cognitif complexe, peut être modélisée à l'aide du langage SEXTANT. Les expérimentations que nous avons conduites sont assez hétérogènes, ce qui est pour nous une preuve de la souplesse de la plate-forme logicielle NaviTexte. Néanmoins, ces expérimentations nous ont confrontés à de nouveaux besoins. D'une part, nous cherchons à modéliser des phénomènes discursifs qui mettent en jeu des relations d'ordre comme par exemple l'ordre chronologique (Battistelli *et al.*, 2006), d'autre part, nous travaillons actuellement à des extensions du langage et à leur implémentation dans NaviTexte afin de pouvoir exprimer des opérations de navigation sur ces structures discursives. Enfin, la création d'applications concrètes pour des cas d'utilisation réels, alimente des discussions quant à une méthodologie d'acquisition et d'expression de connaissances. Soulignons qu'il est important de définir les moyens pour coordonner l'information encodée dans les textes avec les vues et les opérations définies dans les modules de connaissances. Il existe également des problèmes concernant le niveau de détail des connaissances dans les textes, la granularité des UT, la nomenclature, la manière dont les opérations de navigation peuvent être définies, etc. Un travail futur consistera à proposer des « directives d'encodage de connaissances », tant dans les textes que dans les modules. Cela entraînera, sûrement, un enrichissement du langage SEXTANT et de nouvelles versions de la plate-forme logicielle NaviTexte.

Remerciements

NaviLire a reçu un soutien financier de l'Ambassade de France au Danemark. *NaviTexte* est soutenu actuellement par le programme ECOS-Sud U05H01.

7. Bibliographie

Aho A.V., Sethi R., Ullman J.D., *Compilers : Principles, Techniques and Tools*, Addison Wesley, Massachusetts, USA, 1986.

Baccino T., *La lecture électronique*, Grenoble, Presses Universitaires de Grenoble, 2004.

Battistelli D., Minel J.-L., « Les systèmes de résumé automatique : comment assurer une continuité référentielle dans la lecture des textes », in G. Sabah (Ed.), *Compréhension des langues et interaction*, p. 295-330, 2006.

Battistelli D., Minel J.-L., Schwerr S., « Représentation des expressions calendaires : vers une application à la lecture des biographies», *TAL 47/2*, 25 p.

Berge C., *Théorie des Graphes*, Paris, Dunod, 1958.

Bilhaut F., « The Linguastream Platfom », *Proceedings of the 19th Spanish Society for Natural Language Processing Conference (SEPLN)*, Alcalá de Henares, Spain, p. 339-340, 2003.

Bird S., Liberman M., « A Formal Framework for Linguistic Annotation», *Speech Communication*, n° 33, p. 23-60, 2001.

Bodner R., Chignell M., « Dynamic hypertext : querying and linking », *ACM Computing Surveys*, vol 31, n° 4,p. 120-132, 1999.

Bourigault D., « Upery : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus », *TALN 2002*, Nancy, p. 75-84, 2002.

Brusilovsky P., « Adaptive Hypermedia : An attempt to analyse and generalise », *UM'94 Fourth International Conference on User Modelling*, p. 80-91, 1994.

Brusilovsky P., « Methods and techniques of adaptive hypermedia », *User Modeling and User-Adapted Interaction*,vol 6, *2-3*, p. 87-129, 1996.

Burbeck S., « Applications Programming in Smalltalk-80 (TM) : How to use Model-View-Controller (MVC) », http ://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html.

Charolles M., « L'encadrement du discours - Univers, champs, domaines et espace », *Cahier de recherche linguistique, LANDISCO*,vol 6, Université Nancy 2, p. 1-73, 1997.

Cotte D., « Leurres, ruses, désorientation dans les écrits de réseau : la métis à l'écran. », *Communication et langages*, n° 139, p. 63-74, 2004.

Couto J., « ContextO, Los sistemas de exploracion contextual de cara al usuario», Mémoire de Master, Université de la République, Uruguay, 2001.

Couto J., Modélisation des connaissances pour une navigation textuelle assistée. La plate-forme logicielle NaviTexte. Thèse de doctorat, Université Paris-Sorbonne, 2006.

Couto J., Minel J.-L, « Outils dynamiques de fouilles textuelles », *Actes de RIAO*, Avignon, p. 420-430, 2004.

Couto J., Ferret O., Grau B., Hernandez N., Jackiewicz A., Minel J.-L., Porhiel S., « RÉGAL, un système pour la visualisation sélective de documents. », *Revue d'Intelligence Artificielle*, Hermès, p. 481-514, 2004.

Couto J., Lundquist L., Minel J.-L., « Naviguer pour apprendre», *EIAH 2005*, Montpellier, p. 45-56, 2005.

Couto J., Minel J.-L, « SEXTANT, un langage de modélisation des connaissances pour la navigation textuel», *ISDD'06*, Caen, p. 80-90, 2006.

Crestani F., de la Fuente P., Vegas J., « Experimenting with graphical user interface structured document retrieval », *SIGIR'02*, Tampere, Finlande, 2002.

Crispino G., Conception et réalisation d'un système informatique d'exploration contextuelle. Conception d'un langage de spécification de connaissances linguistiques, Thèse de doctorat, Université Paris-Sorbonne, Paris, 2003.

Cunningham H., Maynard D., Bontcheva K. & al., « GATE : A Framework and Graphical Development Environment for Robust NLP Tools and Applications, *ACL'02*, ACM Press, Philadelphie, Pennsylvanie, p. 168-175, 2002.

Danielson D.R., « Web navigation and the behavorial effects of constantly visible maps », *Interacting with Computers*, 14, 2002, p. 601-618, 2002.

Dieberger A., Russell D.M., « Exploratory navigation in large multimedia documents using Context Lenses », *35th Hawaii International Conference on System Sciences*, Hawaii, 2002.

Edwards D.M., Hardman L., « Lost in hyperspace : cognitive mapping and navigation in a hypertext environment», in R. McAleese (Ed.), *Hypertext : Theory and Practice*. Oxford, England : Intellect Books, p. 105-125, 1989.

Elm W.C., Woods D.D., « Getting lost : A Case Study in Interface Design. », *Proceeding of the Human Factors Society*, p. 927-931, 1985.

Endres-Niggemeyer B., Maier E., Sigel A.. «How to implement a naturalistic model of abstracting : four core working steps of an expert abstractor», *Information Processing et Management*, 31(5), p. 631-674, 1995.

Ferret O., Grau B., Minel J.-L. , Porhiel S., « Repérage de structures thématiques dans des textes », *TALN 2001*, Tours, p. 163-172, 2001.

Ho-Dac M., Jacques M.-P., Rebeyrolle J., « Sur la fonction discursive des titres », *in S. Porhiel et D. Klingler (Eds.), L'unité texte*, p. 125-152, 2004.

Hobbs J.R., « On the coherence and structure of discourse », *CSLI Technical Report 85-37*, Stanford, CA, USA, 1985.

Kehler A., *Coherence, reference, and the theory of grammar*, Stanford, CA, CSLI Publications, 2002.

Kintsch W., *Comprehension. A Paradigm for Cognition*, Cambridge, Cambridge University Press, 1998/2003, 1998.

Lamping J., Rao R., « The Hyperbolic Browser : A Focus + Context technique for visualizing large hierarchies », in *Readings in Information Visualization : Using Vision to Think*, Morgan Kaufmann Publlishers, p. 382-408, 1996.

Lundquist L., *L'analyse textuelle. Méthode, Exercices*, Copenhagen, Nordisk Forlag, 1990.

Lundquist L., *Tekstkompetence på fremmedsprog*, Copenhagen, Forlaget samfundslitteratur, 2006.

Lundquist L., J.L. Minel, Couto J., « NaviLire, Teaching French by Navigating in Texts», *IPMU'2006*, Paris, 2006.

Mani I., *Automatic Summarization*, Amsterdam, John Benjamins Publishing Company, 2001.

Mathe N., Chen J., « A User-Centered Approach to Adaptive Hypertext based on an Information Relevance Model », *4th International Conference on User Modeling (UM'94)*, Hyannis, MA,. p. 107-114, 1994

Mathieu, Y. Y., *Les verbes de sentiments. De l'analyse linguistique au traitement automatique*, Paris, CNRS éditions, 2000.

Mathieu, Y. Y., « Annotations of Emotions and Feelings in Texts », In Conference on *Affective Computing and intelligent Interaction (ACII2005)*, Beijing, Springer Lecture Notes in Computer Science, p. 350-357, 2005**.**

Minel , J-L., Nugier, S., Piat, G., « How to appreciate the Quality of Automatic Text Summarization ». *Workshop Intelligent Scalable Text Summarization, EACL 97*, Madrid, p. 25-30, 1997.

Minel J.-L., Cartier E., Crispino G., Desclés J.-P., Ben Hazez S., Jackiewicz A., « Résumé automatique par filtrage sémantique d'informations dans des textes, Présentation de la plate-forme FilText », *Technique et Science Informatiques*, n° 3, Paris, Hermès, p. 369-396, 2001.

Minel J.-L., Filtrage sémantique de textes. Problèmes, conception et réalisation d'une plate-forme informatique. Habilitation à diriger des recherches, Université Paris-Sorbonne, 2002.

Minel J.-L., *Filtrage sémantique. Du résumé à la fouille de textes*. Paris Hermès, 2003.

Pery-Woodley M.-P., «Discours, corpus, traitements automatiques», in A. Condamines (Ed.), *Sémantique et Corpus*. Hermès, p. 177-205, 2005.

Porhiel S., « Les introducteurs de cadre thématique », *Cahiers de Lexicologie*, vol. 83, n°2, p. 1-36, 2003.

Salton G., Wong A., Yang C., « A vector space model for information retrieval », *Communications of the ACM*, vol 18, n° 11, p. 613–620, 1975.

Souchier E., Jeanneret Y., Le Marec J., *Lire, écrire, récrire : objets signes et pratiques des médias informatisés*, Paris, Direction du Livre et de la Lecture, BPI , 2003.

Text Encoding Initiative, http ://www.tei-c.org

Thompson S., Mann W., «Rhetorical structure theory, a framework for the analysis of texts», *IPRA Papers in Pragmatics*, p. 79-105, 1988.

Webber B., Knott A., Stone M., Joshi A., « Anaphora and Discourse Structure », *Computational Linguistics, vol 29, n°4*, p. 545-588, 2003.

Wolf G., Gibson A., « Representing Discourse Coherence : A Corpus-Based Study », *Computational Linguistics, vol. 31, n0. 2*, p. 249-288, 2005.

Wonsever D., Repérage automatique des propositions par exploration contextuelle, Thèse de doctorat, Université Paris-Sorbonne, Paris, 2004.