Hypertextualisation automatique de documents techniques

Fabrice Papy

Université Paris 8 – Saint-Denis

THESE

Présentée en vue d'obtenir le Doctorat en **Information et Communication**

par

Fabrice PAPY

HYPERTEXTUALISATION AUTOMATIQUE DE DOCUMENTS TECHNIQUES

Soutenue le 8 février 1995

devant le jury composé de :

Monsieur Jean-Pierre BALPE	Directeur
Monsieur Richard BOUCHÉ	Rapporteur
Monsieur Christian FLUHR	Rapporteur
Monsieur Jean-Louis VULDY	

Remerciements

Mes remerciements et ma profonde reconnaissance vont à toutes celles et à tous ceux qui ont contribué à ce que cette thèse devienne réalité :

- à Monsieur le Professeur Jean-Pierre Balpe qui a accepté d'être mon directeur de thèse,

- à Monsieur Le Professeur Richard Bouché et Monsieur Le Professeur Christian Fluhr pour leur participation à mon jury,

- à Monsieur Jean-Louis Vuldy sans qui, rien n'eût été possible,

- à Véronique Jolly, Michel Boin, et Patrick Mouttapa pour leur relecture attentive,

- à toute ma famille pour leur soutien moral,

- à André, Gilles, Gérard, Poukham, Florence, Isabelle, Julio, Imad, Cristina, Régis, Sylvie, Frédéric, Catherine, Mazhar, Arash pour leur indéfectible amitié.

Je voudrais exprimer ici toute ma reconnaissance à mon professeur d'Aïkido, Madame Claudie Bélus, pour tout ce qu'elle m'a apporté, tout ce qu'elle m'aide à comprendre et pour ses qualités humaines, dont est si empreint son enseignement.

Introduction

Peut–on produire automatiquement, et de manière industrielle des réseaux hypertextes ?

Les systèmes hypertextes / hypermédias commercialisés ont été à l'origine de démonstrations et de prototypes qui ont, certes, mis en évidence l'intêrèt de l'accès intuitif à l'information propre au concept de l'hypertexte, mais qui n'ont pas conduit à la réalisation de réseaux véritablement opérationnels. L'absence d'une méthodologie de conception est une cause majeure à cette dimension artisanale de l'hypertexte. Les problèmes relatifs à la maintenance des réseaux, à leur évolutivité ou bien à la capacité des systèmes à répondre précisément aux besoins d'information des lecteurs ont certainement empêché la percée de l'hypertexte dans un environnement industriel.

La création automatique de réseaux hypertextes ou hypertextualisation automatique, revêt davantage l'apparence d'un processus empirique que d'une réelle méthodologie de conception mais son objectif est bien d'atteindre cette dimension industrielle. L'hypertextualisation est un processus distinct de l'hypertexte, dont la problématique ne se réduit pas uniquement à la granularisation de l'information et à la pose automatique de liens et qui implique doublement l'hypertexte, du point de vue de l'organisation des réseaux automatiquement générés et du point de vue de leur consultation par les lecteurs.

Entreprendre l'hypertextualisation, c'est avoir fait le choix de l'hypertexte, le choix d'une manière différente d'appréhender la consultation de l'information, dans le chapitre 1 nous dresserons un état de l'art consacré au concept de l'hypertexte. Nous nous intéresserons à l'aspect historique, aux définitions, aux problèmes de conception, de maintenance et de navigation qu'il ne manque pas de soulever.

Fournir aux lecteurs une aide intelligente à la navigation, offrir des possibilités de recherche d'information aussi puissantes que celles des systèmes de gestion de bases de données ont orienté de nombreuses recherches vers la coopération entre d'une part l'Intelligence Artificielle et l'hypertexte et d'autre part les bases de données et l'hypertexte. Le chapitre 2 nous permettra de faire le point sur cette coopération.

L'hypertextualisation repose sur la réutilisation de documents linéaires déjà existants pour créer dynamiquement les composantes de base des futurs réseaux hypertextes. Nous présenterons dans le chapitre 3 les notions de structure logique et physique des documents sur lesquelles s'appuie l'automatisme de création. Nous décrirons plus précisément dans le chapitre 4 les mécanismes d'extraction des nœuds et de reconnaissance des liens explicites de type références croisées.

La production automatique de noeuds et de liens alimente un fonds hypertextuel qui va croître rapidement et qui risque de devenir anarchique si les opérations d'insertion, de modification et de suppression portant sur ces éléments ne sont pas parfaitement gérées. Préserver la cohérence des données est une fonction que les systèmes de gestion de bases de données savent assumer efficacement. Le chapitre 5 décrit le couplage de l'hypertextualisation avec un

système de gestion de bases de données relationnelles pour répondre aux exigences de l'actualisation des réseaux hypertextes.

Enfin, l'implémentation du processus d'hypertextualisation automatique est proposé dans le chapitre 6.

Chapitre 1

Hypertexte et Hypermédia

1.1 Genèse d'un concept

L'Hypertexte au même titre que les Bases de Données et l'Intelligence Artificielle se situe comme une tentative originale de représentation de l'information. Cette représentation, contrairement aux deux autres domaines, ne s'appuie pas sur une démarche rigoureuse et méthodologique. En effet, l'Hypertexte ne propose ni modélisation du monde réel ni organisation logique de la connaissance, il introduit ainsi une rupture dans la démarche algorithmique et arborescente : les possibilités d'association sont vastes, la rapidité d'accès assurée et la progression dans l'information personnalisée.

Alors que les définitions de l'I.A. et des Bases de Données sont claires et laissent peu de libertés quant à l'interprétation, celle de l'Hypertexte en revanche, restent plus difficiles à cerner.

On trouve dans l'ensemble des travaux spécialisés dans le domaine de l'Hypertexte des tentatives de définition qui tournent autour :

- des éléments de base qui composent l'hypertexte et des objets qu'il manipule,

- des spécificités qu'un système hypertexte doit offrir (qui du même coup permettent de qualifier d'hypertexte ledit système),
- de comparaison avec les graphes,
- des avantages qu'il offre par rapport à la consultation essentiellement séquentielle des documents papier,
- des différences qui le distinguent des autres techniques de traitement de l'information (un hypertexte n'est pas un SGBD ni un système de recherche documentaire et moins encore un système de gestion de fichiers, etc...),

Cette difficulté à trouver une définition précise à l'hypertexte vient certainement du fait, que le terme lui-même se rattache à une réalité en perpétuelle mouvance.

Alors qu'un regard sur l'histoire de l'hypertexte aurait pu être le moyen de stabiliser sa définition (éclairée en cela par les travaux de Bush, Engelbart et Nelson) les incessantes réalisations d'hypertextes introduisent constamment de nouveaux éléments qui perturbent cette définition.

En effet, l'hyperdocument, l'hypermédia, le document multimédia interactif, les systèmes, produits et logiciels hypertextes, le livre électronique, l'aide en ligne hypertexte, les systèmes auteur, etc... dont on pressent bien le rapport avec le concept d'hypertexte, sont autant de notions rapportées qui justifient la perpétuelle révision des définitions proposées.

1.1.1 Des origines de l'Hypertexte

Trois visionnaires : Vannevar Bush, Douglas Engelbart et Theodor Nelson sont reconnus mondialement comme les pionniers de tous les travaux sur l'hypertexte **[Daniel 90] [Nielsen 89] [Conklin 87].** Alors que Bush confronté à des problèmes de documentation scientifique imaginait un système fonctionnant comme le cerveau humain et lançait les bases de ce qu'un tel système devait être, Douglas Engelbart en réalisait le premier prototype et Theodor Nelson baptisait ce nouveau mode de représentation de l'information.

1.1.1.1 La version Vannevar Bush de l'Hypertexte

L'origine reconnue (le terme hypertexte n'existait cependant pas encore à cette époque !) de l'hypertexte remonte à Vannevar Bush (alors conseiller scientifique du Président F.D. Roosevelt) qui s'interrogeait sur la manière de "mécaniser" les systèmes de documentation scientifique devant les difficultés qu'il rencontrait pour manipuler un volume d'information de plus en plus important. Il imagine à l'époque, le système Memex qui imite les capacités associatives de l'esprit humain où possibilité est donnée d'explorer et d'annoter de l'information textuelle ou graphique (la citation ci-dessous est emprunté à **[Nielsen 89]**, extrait de l'article de V.Bush, "As we may think", paru en Juillet 1945 dans Atlantic Monthly, pp 101-108) :

> *"...The human mind (...) operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain. Selection by association, rather than indexing, may yet be mechanized. Man cannot hope fully to duplicate this mental process artificially, but he certainly ought to be able to learn from it. One cannot hope to equal the speed and flexibility with which the mind follows an associative trail, but it should be possible to beat the mind decisively in regard to the permanence and clarity of the items resurrected from storage..."*

Compte tenu des médias de restitution de l'époque (papier, microfiches, photos...), V. Bush n'a pu proposer de prototype fonctionnel du système de consultation idéal que représentait son système Memex. Du reste, il admettait que seuls d'importants progrès technologiques pouvaient rendre son projet réalisable.

1.1.1.2 La version Douglas Engelbart de l'Hypertexte

C'est moins de 20 ans plus tard que Douglas Engelbart proposait une implémentation des idées de Vannevar Bush. Le système NLS (oN Line System) était né. Plus qu'une simple implémentation, Engelbart voulait montrer qu'une assistance de l'être humain par l'ordinateur en matière de manipulation de l'information pouvait aller vers l'amélioration des capacités intellectuelles de l'utilisateur. Il décrit NLS comme une réalisation expérimentale d'un outil répondant aux besoins de ses activités quotidiennes.

Dans ce système se retrouve stocké tout son environnement de travail ; plans, programmes, rapports, documents, bibliographies, etc... qui lui permettent de créer, concevoir, planifier, corriger par l'intermédiaire d'un poste de travail informatique.

1.1.1.3 La version Theodor Nelson de l'Hypertexte

Theodor Nelson, alors lancé à cette même époque dans le projet Xanadu, invente les termes *hypertexte* et *hypermédia*.

Xanadu est une sorte d'environnement de publication où des millions de personnes de par le monde pourraient se connecter à travers l'écrit, le son, l'image et coopérer dans leurs créations. Il y autorise certaines fonctionnalités telles que :

- avoir accès à l'historique d'un document particulier,

- pouvoir lier de façon explicite un document à ceux avec lesquels il entretient des rapports,
- pouvoir gérer dans un même et unique environnement réseau tous les documents,
- respecter le copyright et les royalties dans le cas de travail coopératif,
- etc...

1.1.2 Des visions complémentaires

De récents travaux tendent à préciser le fonctionnement des hypertextes, à élargir leurs champs d'action ou à leurs donner une réalité particulière. Nous présentons quelques aspects originaux (livre électronique, hypermédia, hyperdocument...) gravitant autour du concept hypertexte.

1.1.2.1 Livre électronique

Les travaux qui décrivent les systèmes à base de livre électronique **[Savoy 88] [Pasquier 89]** cherchent à montrer la spécificité du livre électronique en le distinguant fonctionnellement de l'hypertexte.

Ainsi, J. Savoy appréhende l'hypertexte comme une alternative électronique permettant de rompre avec la linéarité des livres traditionnels. Il garde principalement à l'esprit l'essence même des travaux de V. Bush selon laquelle la structure linéaire et statique de l'écrit ne correspond pas aux processus mentaux humains.

Il implémente le livre électronique EBOOK3 dédié à l'animation de textes par des traitements numériques, qui gère texte et image en permettant

également l'exécution de programmes informatiques (pour présenter des résultats numériques, exercer les connaissances acquises par le lecteur, construire des graphiques complexes) dans les domaines des sciences telles que la recherche opérationnelle, l'économétrie, les statistiques, la physique, la chimie, etc...

Dans ce projet, J. Savoy parle constamment de livres électroniques et non d'hypertextes. Il considère son système comme hypertexte dans le sens où celui-ci permet une exploration non séquentielle des documents (du reste cela représente pour lui, **LA** caractéristique fondamentale de l'hypertexte).

J. Savoy distingue l'hypertexte du livre électronique dans le sens où ce dernier met en évidence les capacités d'apprentissage (et d'enseignement) du livre. Un livre est destiné à fournir de la connaissance, à l'expliciter au moyen d'exemples et à vérifier sa bonne assimilation par le lecteur (J. Savoy cite souvent à cette occasion l'étudiant) au moyen d'exercices particuliers. C'est ce qu'il a reproduit dans EBOOK3 en mettant à la disposition d'utilisateurs en situation d'apprentissage des connaissances brutes auxquelles il associe des exemples guidés, des exercices et leurs corrections avec un haut niveau d'interaction (cela est corroboré par le fait qu'EBOOK3 s'applique particulièrement bien aux disciplines scientifiques où on retrouve le triangle connaissances, exercices, corrections).

Dans cet article J. Savoy prédispose le livre électronique à une utilisation de type Enseignement Assisté par Ordinateur.

1.1.2.2 Document Multimedia interactif

L'ouvrage de C.J Anderson et M.K. Veljkov **[Anderson 90]** est un guide pratique pour la création de documents multimédia interactif. Dans cet ouvrage,

les auteurs insistent sur l'aspect interactif qui représente la capacité de l'utilisateur à interagir avec la technologie. Selon eux, le multimédia interactif se distingue du multimédia et de l'hypermédia (qu'ils assimilent à des *presentation manager* sophistiquées), dans le sens où l'essentiel des efforts lors de la conception d'un produit multimédia interactif, est mis dans l'amélioration constante de l'interactivité pouvant exister entre l'utilisateur et les moyens technologiques qui lui sont proposés.

La caractéristique de l'interactivité réside dans :

> *"…not just sitting and watching, but seeing, hearing, thinking and doing…".*

En revanche, dans la description des moyens matériels et logiciels indispensable à la réalisation de produit multimédia interactif, les auteurs précisent l'obligation de disposer de fonctionnalités hypertextes :

> *"…Hypertext functions can dynamically link on-screen elements - video, animation, text or graphics - to additional information. The information that you might link to, can also include video, graphics, animation and/or text. The uniqueness of Hypertext is its ability to link this information together, thus creating a gigantic knowledge navigation system. Hypertext features are becoming more important all the time. However, many developers see them as fluff, and not necessery for creating a well designed interactive multimedia course or presentation. With the increasing amounts of information and knowledge being used in interactive multimedia, Hypertext features are becoming standard…"*

Ils mettent effectivement en évidence la capacité de l'hypertexte à lier des informations de natures hétérogènes mais ne lui accordent pas le contrôle pertinent de cette navigation par rapport à l'utilisateur.

1.1.2.3 Hypermédia

Le terme d'hypermédia est une généralisation du concept d'hypertexte lorsque celui-ci est pris dans le sens restrictif de l'écrit non-linéaire. En effet, et

principalement pour des raisons historiques (les premiers produits hypertextes étaient essentiellement composés de textes), bon nombre d'auteurs **[Dachelet 90]** **[Frisse 88a]** ont défini le concept d'hypertexte en termes d'ensemble de textes ou de morceaux de textes reliés entre eux par des éléments textuels (un lien peut exister entre un point dans un texte et une référence bibliographique, entre un mot et un commentaire ou un résumé, ou bien encore entre l'auteur du texte et sa biographie...).

Dès que les nœuds du réseau hypertexte ne se limitent plus à contenir du texte mais contiennent des graphiques, des schémas, des séquences audio ou vidéo, on parle d'hypermédia.

[Halasz 88] :

> *"...Hypermedia is a style of building systems for information representation and management around a network of multi-media nodes connected together by typed links. Such systems have recently become quite popular due to their potential for aiding in the organization and manipulation of irregularly structured information in application ranging from legal research to software engineering..."*

[Balpe 90] :

> *"...Un hypermédia est un ensemble d'informations appartenant à plusieurs types de médias (texte, son, image, logiciels) pouvant être lu (écouté, lu) suivant de multiples parcours de lectures, en utilisant également la possibilité du multi-fenêtrage. Ce qui différencie essentiellement l'hypermédia de l'hypertexte n'est ainsi que la nature symbolique des codages d'information utilisés. Un hypermédia n'est rien d'autre qu'un hypertexte gérant des textes supportés par des médias divers..."*

Actuellement, les termes Hypertexte et Hypermédia tendent à désigner les mêmes choses et sont en ce sens de plus en plus utilisés comme synonymes.

[Smith 88] :

> *"...Hypertext is an approach to information management in which data is stored in a network of nodes connected by links. Nodes can contain text, graphics, audio, video, as well as source code or other forms of data. The nodes, and in some systems the network itself, are meant to be viewed throught an interactive browse and manipulated throught a structure editor..."*

[Nielsen 89] :

> *"...Since many of the current systems actually also include the possibility for working with graphics and various other media, some people prefer using the term hypermedia, to stress the multimedia aspects of their system. Personally, I would like to keep using the traditionnal term "hypertext" for all systems since there does not seem to be any reason to reserve a special term for text-only systems. Therefore I tend to use the two terms hypertext and hypermedia interchangeably with preference to sticking to hypertext ..."*

Dans le cadre de cette thèse, nous utiliserons les termes Hypermédia et Hypertexte comme synonymes.

1.1.2.4 Hyperdocument

Le terme *hyperdocument* peut être génériquement défini comme la description de l'information représentée à un instant donné sous une forme hypertexte. Certains auteurs lui associent certaines particularités :

[Brown 89] :

> *"...A body of information stored in a form suitable for hypertext processing is called a hyperdocument. There at the moment no accepted standards for hyperdocuments, and each hypertext system has its own representation. Nevertheless the underlying principles are common in that the hyperdocument normally consists of a directed graph..."*

[Balpe 90] :

> *"...Les principales caractéristiques des hypertextes sont celles qui définissent les hyperdocuments : - ils sont constitués d'un ensemble d'informations enregistrées, uniquement de nature textuelle, au sein desquelles les parcours de lecture peuvent être multiples - ils utilisent le multifenêtrage : plusieurs documents lus pouvant s'inscrire simultanément sur un même espace-écran, - ils permettent une certaine interactivité : la possibilité est offerte au lecteur*

de faire ses choix de parcours localement, c'est à dire à la différence des parcours de bases de données par exemple, non pas au départ de la demande qui est alors traitée globalement par l'ordinateur, mais en fonction de la lecture qu'il est en train de faire. En ce sens, ils se distinguent fondamentalement du produit que certains auteurs appellent livre électronique ..."

Globalement on peut définir un hyperdocument comme tout document électronique accessible de manière non-linéaire grâce à un cheminement au moyen de liens **[Sandvad 89].**

1.2 Une définition fédérée

Une définition précise de l'hypertexte ne peut s'affranchir des nombreuses réalisations de systèmes hypertextes/hypermédia qui sont apparus ces dernières années (Knowledge Management System, Notecards, Guide, Intermédia, Multicard...).

Plus que la matérialisation des idées des pionniers, ces applications informatiques ont sans nul doute apporté leur part de créativité au concept d'hypertexte **[Campbell 88]**.

Car au-delà d'une description physique de l'hypertexte en termes de nœuds et de liens, les systèmes hypertextes ont montré l'utilisation qui pouvait être faite de la technologie (terminaux graphiques, périphériques audio et vidéo, mémoires de masse à accès rapide,...) pour une représentation aisée de l'information associée à une manipulation intuitive au moyen d'interfaces conviviales. Dans cet objectif, des solutions locales, n'engageant que les développeurs de systèmes hypertextes, ont été envisagées pour répondre aux problèmes pratiques du contrôle des versions, de la navigation ou de la coopération dans les hypertextes.

Ces problèmes (et leurs solutions) sont autant de facteurs qui participent à une meilleure définition de l'hypertexte.

1.2.1 Nœuds et liens

Les chercheurs s'accordent pour définir physiquement l'hypertexte comme un réseau de nœuds et de liens. Les nœuds ou grains d'informations sont les réceptacles de l'information. Ils sont considérés par l'utilisateur qui les consulte (ou par l'auteur qui les crée) comme des entités sémantiques distinctes et "autonomes". L'information qui y est stockée peut être de nature différente (texte, image, vidéo, son). Les liens sont principalement de deux types **[Conklin 87] [Daniel 90]** :

- les liens de références qui traduisent les relations sémantiques entre deux nœuds d'informations. Ce sont des relations orientées entre un nœud (ou une partie d'un nœud) et un autre nœud. Ils s'expriment en termes d'*ancre de départ* et d'*ancre d'arrivée*.

- les liens hiérarchiques sont destinés à organiser logiquement l'ensemble des nœuds. Ces liens en donnant une référence unique à des familles de nœuds permet de les organiser et de les classifier. Ils servent par exemple à établir des plans et des tables des matières ou à obtenir une vue partielle du réseau.

1.2.2 Système auteur et système lecteur

Avant l'apparition des premiers outils de création automatique d'hypertextes, toure information présentée sous forme hypertexte devait être préparée manuellement.

L'élaboration d'un hyperdocument nécessite alors des outils spécialisés permettant de décrire la structure du réseau, de créer les nœuds et de poser les liens. L'information elle-même pouvant revêtir des formes diverses, le système

hypertexte doit alors disposer d'éditeurs spécialisés pour la créer ou pour l'importer à partir d'autres environnements.

Tout réseau pouvant évoluer, les opérations de destruction, de modification doivent être autorisées. Dans les premiers systèmes hypertextes, ces fonctionnalités spécifiques à la situation d'auteur, ne pouvaient être obtenues qu'en adjoignant au système hypertexte initial (limité à la navigation) une *boîte à outils Auteur*. Plus récemment, les systèmes hypertextes permettent à l'utilisateur de définir son profil (auteur ou lecteur) et lui fournissent alors les outils adaptés à ce profil. Un utilisateur déclaré lecteur sera limité à des opérations d'exploration (les rares possibilités de création concernent les annotations ou commentaires que l'utilisateur peut éventuellement associer aux nœuds qu'il visite).

1.2.3 Navigation et Recherche

Les systèmes hypertextes permettent un accès facile à l'information. Pour cela ils proposent deux méthodes : la navigation et la recherche.

La navigation consiste à visualiser le contenu des nœuds au fur et à mesure des liens activés. Le lecteur peut ainsi explorer le réseau en suivant des chemins prévus par les auteurs (ou par d'autres lecteurs) ou en suivant librement d'autres parcours virtuels au gré des liens qu'il déclenche.

Cette navigation peut se faire directement en passant du contenu d'un nœud au contenu d'un autre nœud en activant des liens, mais elle peut aussi s'effectuer en utilisant la représentation cartographique de l'hypertexte que certains systèmes proposent (Notecards, MacWeb, etc...).

Une navigation par la structure est aussi possible et permet ainsi le cheminement inverse du parcours (*backtracking)*, l'accès au nœud précédent ou au

nœud suivant, ou bien encore un accès direct à un nœud précédemment visité grâce à l'utilisation d'un historique de parcours.

La recherche concerne essentiellement la recherche en texte intégral d'une chaîne de caractères cible (Hypercard, Toolbook...). Une telle recherche est facilitée par différentes options (que l'on retrouve aussi dans les traitements de textes) :

a) mot entier. Par exemple si le mot *texte* est le terme à trouver et si l'option "mot entier" est active alors le système ne proposera pas les mots *hypertexte*, *prétexte*, et *télétexte*,

b) minuscule/majuscule. Cette option introduit ou enlève la distinction entre les formes minuscule et majuscule d'un même mot. Le mot *document* pourra être considéré identique à *DOCUMENT* ou à *Document*,

c) caractères spéciaux. Certains caractères (habituellement le point d'interrogation et l'astérisque) autorisent des rapprochements paronymiques. Ainsi *p?rte* permettra de retrouver aussi bien *porte* que *perte* alors que *form*t* proposera *format*, *forment* ou *formaient.*

Ces options de recherche bien que cumulables entre elles demeurent cependant inopérantes dès qu'il s'agit de rechercher des expressions textuelles complexes (qui tendent à devenir alors des requêtes de bases de données) ou bien des éléments de nature non textuelle.

Pour **[Brown 88b],** la commande de recherche (*Find Command*) représente un type de lien implicite et non-structuré, dans le sens où elle réalise des déplacements vers n'importe quel point du document (il l'assimile à un genre de *goto*) alors que les liens posés par l'auteur sont structurés et implicites. La commande de recherche correspond à "rechercher la prochaine occurence" (lien

de type 1:1) et se différencie fondamentalement de "rechercher toutes les occurences" (lien de type 1:N).

1.2.4 Les versions d'un hypertexte

Un hypertexte est susceptible d'évoluer au gré des opérations de modification, de suppression et d'insertion de nœuds et de liens. Cette mise à jour perpétuelle de l'hypertexte pose le problème concret de savoir si l'hypertexte doit seulement garder l'image la plus récente de l'information qu'il représente ou au contraire s'il doit "garder en mémoire" les modifications successives de cette information. Le *contrôle de version* (ou *versioning*), inspiré des techniques du Génie Logiciel, est destiné à gérer les configurations des différentes versions des modules constituant un programme **[Ambriola 90]** :

Afin de pouvoir répondre à la mise à jour des nœuds et des liens dans un réseau hypertexte, l'essentiel du contrôle des versions propre aux programmes, a été adapté au monde de l'hypertexte **[Halasz 88] [Campbell 88]**. Malgré les similitudes qui ne manquent pas d'exister entre le "réseau de modules" et le réseau hypertexte, certaines spécificités de ce dernier soulève les problèmes suivants **[Osterbye 92]** :

- l'immuabilité des versions. Si un nœud dispose de plusieurs versions dont chacune correspond à un instantané de l'information, faut-il autoriser ou non les annotations ou l'insertion de nouveaux liens anachroniques ? Etant donné que les différentes versions d'un nœud (à l'exclusion de la dernière) sont considérées comme "gelées", il n'est pas possible d'ajouter de nouveaux liens. Or, il faudrait autoriser l'insertion d'un nouveau type de lien (de nature contextuelle) qui renseignerait sur le fait que la version du nœud est gelé.

- les versions de liens. Peut-on envisager la gestion des versions de liens de la même manière que celle des nœuds? Ne pas tenir compte d'une évolution des liens équivaudrait à perdre l'historique des liaisons entre les nœuds (si un nœud A relie un nœud B et si la destination du nœud A est changée en faveur du nœud C, l'ancienne destination du nœud A est irrémédiablement perdue). Mais inversement, une gestion de version uniquement orientée sur les liens, introduira immanquablement des problèmes d'ambiguïté (si, dans l'exemple précédent, on garde la liaison entre le nœud A et le nœud B, on se retrouve en situation de concurrence entre le lien menant vers B et le lien menant vers C). Une solution intermédiaire consiste à envisager le contrôle des versions de liens par le contrôle des versions de nœuds (le nouveau lien du nœud A vers le nœud C entrainera la création du nœud A'. Le nœud A, "gelé", gardera le lien vers le nœud, tandis que A' supportera le nouveau lien vers C),

- les versions de structure. Pouvoir revenir à l'état du réseau correspondant à l'état d'une des versions d'un nœud entraîne la nécessité de garder la trace des évolutions globales de l'hypertexte. Ce contrôle des versions de structure permettrait, à partir d'une version locale de nœud ou de lien, de passer aux versions de la structure du réseau hypertexte global (et inversement).

- la création de version. Accumuler les versions locales (des nœuds et des liens) ou globales (de l'ensemble de l'hypertexte afin de préserver la notion de contexte) accentue le phénomène de charge cognitive dont souffre l'utilisateur d'hypertexte,

- La sélection d'éléments. La création de liens sur des versions d'un même nœud augmente aussi la charge cognitive car il s'agit de spécifier de manière univoque ces nouveaux liens.

1.2.5 Hypertexte coopératif

Comme Ted Nelson a tenté de le montrer avec son projet Xanadu, les systèmes hypertextes représentent un moyen particulièrement approprié au travail coopératif.

Certains chercheurs **[Garg 88b]** ont repris cette idée en s'attachant à définir les mécanismes de base permettant le travail coopératif (écriture simultanée de parties par un collectif d'auteurs sur un même sujet et intégration de ces parties au sein d'un même réseau hypertexte). Ces mécanismes reposent essentiellement sur l'agrégation de nœuds hypertextes et s'appuient sur la notion d'objets composés et de types de données abstraites propres aux langages de programmation orientés objet.

D'autres chercheurs **[Cybulski 92]** ont montré de manière plus pratique, à travers un environnement spécialement étudié pour le génie logiciel (HyperCASE), comment l'hypertexte pouvait être utilisé dans le cadre du travail coopératif.

1.2.6 Modélisation des systèmes hypertextes

Malgré les difficultés à couvrir précisément la globalité du concept d'hypertexte, certains auteurs **[Garg 88a] [Richard 90]** ont cherché à formaliser les réseaux hypertextes. Cette tentative de modélisation part de l'observation des systèmes les plus répandus et tend à donner une description mathématique des concepts de bases communs à ces différents systèmes dans le but de décrire théoriquement leur évolution.

Ce sont principalement les aspects statiques (définition précise de la nature des objets informatifs atomiques et des types des liens qui existent entre ces

éléments atomiques) du réseau hypertexte qui ont été modélisés. Cette modélisation s'appuie essentiellement sur les opérateurs ensemblistes de l'algèbre relationnelle (généralisation, agrégation, composition, spécialisation) qui conduisent à la construction d'objets complexes.

La notion d'abstraction s'avère particulièrement intéressante pour répondre aux problèmes soulevés par :

- la pertinence de l'information ou comment cibler l'information à consulter en s'inspirant des vues des bases de données ?
- les unités d'information composées. Le lecteur d'un hypertexte en sélectionnant les nœuds du réseau qui lui paraissent appropriés se compose sa propre structure de l'hypertexte. Ceci représente un cas particulier d'agrégation,
- le travail collaboratif,
- le versioning.

Les aspects plus dynamiques de l'hypertexte (relatifs à la présentation de l'information ou à la façon dont elle est consultée) ont fait l'objet de travaux théoriques (s'appuyant sur les réseaux de Petri) et ont conduit à la réalisation du modèle Trellis **[Stotts 88] [Stotts 89].** Ces travaux tentent de donner un formalisme mathématique aux caractéristiques des systèmes hypertextes existants par rapport à :

- la concurrence et à la synchronisation. Plusieurs éléments d'un hypertexte peuvent être actifs simultanément. Le modèle Trellis autorise des activités concurrentes au sein de l'hypertexte et veille à ce qu'elles soient synchronisées.
- une sémantique d'exploration programmable. Les auteurs peuvent décrire les éléments qui peuvent apparaître en

même temps en fonction du chemin de consultation. Des profils d'utilisateur peuvent être alors déclarés et donner lieu à l'exécution de scénarios particuliers.

- l'accessibilité des nœuds. Il est possible de savoir si tous les nœuds du réseau peuvent être visités à partir d'un parcours de consultation ou si des nœuds demeurent inaccessibles à partir de certains points initiaux.

1.2.7 Hypertexte et Maintenance

L'absence de modélisation des réseaux hypertextes rend difficile l'élaboration d'une méthodologie de conception et de réalisation (autre qu'empirique) de ces réseaux. Cette absence de méthodologie resurgit aussi bien lors de la création de nouveaux réseaux que lors de la modification de réseaux existants.

Dans ce domaine de l'information électronique traitée manuellement, **[Brown 88]** estime qu'une centaine d'heures auteur sont nécessaires pour créer une heure d'information électronique purement textuelle, ce ratio pouvant s'élever à 1 pour 1000 dans le cas de média différents. Bien qu'aucune analyse publiée ne vienne confirmer ce rapport pour la création manuelle de réseaux hypertextes, on peut croire que cela reste vrai dès qu'il s'agit de réaliser des réseaux hypertextes où la présentation de l'information doit être soignée.

Par la mise à jour physique du réseau (insertion, suppression, modification de nœuds et de liens), c'est l'esprit dans lequel le réseau a été conçu que devra perpétuer le responsable de la maintenance. Si celui-ci se trouve être une autre personne que l'auteur, le réseau de ce dernier risque de changer fondamentalement et au pire de devenir incohérent, accentuant ainsi les

phénomènes de désorientation (car l'équipe de maintenance n'a pas trace des intentions originales de l'auteur).

Alors que les efforts des développeurs tendent à étoffer les systèmes auteurs d'outils de navigation toujours plus puissants, aucun outil de test et de validation n'est développé pour assurer efficacement la maintenance des versions d'un réseau. Le manque d'outils de recherche simple qui permettraient par exemple de retrouver "tout ce qui est relié à ce nœud" ou "tous les types de liens utilisés dans le réseau". Ces outils se révèlent vite indispensables pour les auteurs et les équipes de maintenance dès qu'il s'agit de faire évoluer un réseau.

1.4 Finalité de l'Hypertexte

Alors que les bases de données ont favorisé les techniques d'interrogation et les systèmes experts, les techniques de déduction et d'induction, les hypertextes ont mis en évidence la navigation libre à travers l'information et s'est imposé comme un moyen original d'appréhender l'information en permettant de réaliser des associations entre informations, associations qui ne s'expliquent pas nécessairement par des raisons déductives, de cause à effet ou de tout autre nature logique mais pour des raisons purement intuitives.

Nous avons vu à ce propos qu'une définition précise de l'hypertexte est difficile à formuler tant les notions qui s'y rapportent sont diverses. Le développement de nouveaux systèmes hypertextes et l'amélioration continuelle de ceux existant déjà en contribuant à enrichir le concept, le maintiennent dans un flou artistique.

Mais c'est plus en terme de finalité que se comprend l'hypertexte plutôt que par l'énonciation de ses caractères essentiels et de ses qualités propres. En effet, on ne peut oublier que l'hypertexte permet avant tout de mettre les capacités de calcul et de présentation d'un ordinateur au service de l'information non structurée en réalisant des associations entre des éléments de nature différente, associations conduites par l'intelligence ou l'intuition de l'utilisateur.

Puissance des calculateurs, mémoires de masse, écrans haute définition, réseaux de communication haut débit sont autant de termes qui confirment la dimension technologique de l'hypertexte mise au service de disciplines aussi différentes que la philosophie, la psychologie, les langues, la médecine, ou l'informatique. Cette pluridisciplinarité des sources d'information que manipule l'hypertexte confirme aussi que le développement de l'hypertexte tient au désir de

supporter l'associativité inhérente aux situations de recherche ou d'élaboration d'informations.

L'utilisateur (ou lecteur) d'hypertexte dispose désormais d'un outil interactif qui lui permet de naviguer dans une base d'informations, de fouiller en profondeur une information (dont il a l'intuition de l'intérêt) ou bien de passer au sujet connexe traité et ce en variant ses chemins de lecture.

Alors que **[Dachelet 90]** situait l'hypertexte

> *"...à la croisées des chemins, celui des traitements de textes, des bases de données et celui des disciplines cognitives (psychologie et linguistique). C'est à dire à l'intersection du document, de l'information et de la connaissance..."*

on ne peut nier que **[Camacho 91]** est plus proche de la réalité quand ils écrivent que :

> *"...L'hypertexte est à son niveau le plus simple, un système de gestion de données qui permet d'associer des écrans par des liens. A son niveau le plus sophistiqué, c'est un environnement logiciel pour un travail collaboratif, pour la communication, pour l'acquisition et la représentation de connaissances. Les produits hypertextes s'inspirent de la capacité humaine d'engranger puis de rechercher l'information par des liens référentiels, pour un accès rapide et intuitif..."*

La représentation de l'information dans les domaines des bases de données et des systèmes experts a été rendu possible grâce à une modélisation et une structuration des données extrêmement poussées. Ces systèmes demandent aux utilisateurs de bien connaître les structures des données et à bien savoir les manipuler avant de pouvoir extraire les informations utiles à partir des données stockées.

Les systèmes hypertextes en revanche n'obligent pas les lecteurs à connaître la structure des données représentées. La consultation (plus précisément la navigation) se fait directement par l'information elle-même. Du point de vue de

la réalisation des réseaux hypertextes, la structuration des données relève principalement de la nature des informations manipulées (texte, graphiques, son, etc...) plutôt que d'une obligation d'ordre méthodologique. Nous verrons à ce sujet que cette carence en matière de modélisation influe immanquablement sur le lecteur (en matière de navigation et errances éventuelles) et sur l'auteur (en ce qui concerne la maintenance et l'évolutivité des réseaux hypertextes créés).

1.5 Hypertexte et Utilisateur

L'hypertexte représente pour de nombreux chercheurs **[Van Dam 88]** un nouveau moyen de consultation de l'information puisqu'il autorise et réalise de manière déconcertante des associations arbitraires. Ce nouveau mode de consultation n'en soulève pas moins de sérieux problèmes d'ordre cognitif et méthodologique. Bien que conscient des indéniables potentialités des systèmes hypertextes comme moyen de représentation de l'information, **[Mc Knight 88]** suggère que l'adhésion des utilisateurs à un tel moyen sera complétement asujetti à "l'utilisabilité" des hypertextes. La lecture sur écran, le comportement de l'utilisateur, les interfaces des systèmes et produits (présentation de l'information, périphériques d'entrée et de visualisation) et la navigation sont autant de facteurs qui détermineront l'adhésion ou le refus de l'utilisateur.

1.5.1 Utilisateur et Lecture électronique

Les expériences conduites par les psychologues ont cherché à vérifier si la lecture d'un texte sur un périphérique de visualisation électronique était "équivalente" à la lecture de ce même texte sur un support papier. Dans la plupart des expériences, les psychologues ont tenté d'établir la réalité de cette "équivalence" en exprimant un rapport entre vitesse et erreurs de lecture en fonction du support utilisé.

L'expérience menée par **[Hansen 88]** et reprenant les résultats de **[Wilkinson 88]** tentait d'évaluer la vitesse et la qualité de lecture d'un texte sur un périphérique de visualisation et à les comparer avec le même texte imprimé. Des scientifiques dont une des activités consiste à commenter des publications proposées par des éditeurs, composaient l'essentiel de l'échantillon. Il s'agissait de

déterminer si la qualité de ces commentaires changeaient selon le mode de distribution (papier ou électronique) de ces articles. Dans les textes soumis à ces scientifiques, des erreurs volontaires avaient été glissées (fautes de frappe, mots manquants ou répétés, inversion de lignes).

L'expérience a révélé que la lecture sur écran était de 30 à 40 % moins rapide que sur papier.

L'expérience conduite par **[Wright 83]** cherchait à vérifier ces résultats en les étendant au travail d'écriture mais en s'attachant à faire varier les équipements informatiques ; micro-ordinateur utilisé comme poste de travail autonome ou comme terminal relié à un calculateur central, et stations de travail dotées d'écrans graphiques sur lesquelles les tailles des fenêtres pouvaient être modifiées jusqu'à permettre l'affichage d'une page complète.

Tout en confirmant les résultats de Hansen, les résultats de Wright ont mis en évidence l'importance de la taille des fenêtres de visualisation en ce qui concerne la recherche d'une information dans un texte, le réarrangement de lignes de texte désorganisées et le processus d'écriture. Sur ce dernier point, la qualité d'écriture sur une station de travail (avec une fenêtre d'écriture maximale) est comparable à celle du papier.

Bien que la qualité des périphériques de visualisation tendent à s'améliorer, notamment grâce aux dispositifs anti-aliesing visant à une meilleure définition du contour des lettres, la fatigue visuelle s'installe très vite et détériore la qualité de lecture sur ce type de support.

Le lecteur d'hypertexte se trouve alors confronté au paradoxe de pouvoir naviguer librement et intuitivement dans un espace de connaissances dont il ne connaît pas toujours l'étendue et d'être incapable de bénéficier durablement de

cette connaissance. L'alternative réside alors dans la production d'outils que le lecteur d'hypertexte pourrait utiliser dans le but d'optimiser sa navigation vers l'information qu'il recherche et de l'extraire sans trop lire **[Nielsen 89]**.

1.5.2 Méthodologie de conception et Ingénierie de l'hypertexte

Les auteurs **[Brown 88]** **[Van Dam 88]** qui regrettent et dénoncent la difficulté avec laquelle percent les applications hypermédia, attribuent cette situation à l'absence de concepteurs hypermédia (*hypermedia designer, professional designers*), seuls maîtres d'œuvres capables de donner à l'information la dimension "hyper" adaptée à la puissance des outils développés à ce jour.

Mais plus que d'individus disposant d'un savoir-faire hypertexte/hypermédia, c'est véritablement de l'absence d'une méthodologie de conception d'hypertextes éprouvée dont souffre le monde de l'hypertexte.

Cette absence de méthodologie a d'ailleurs conduit les auteurs d'hypertextes à faire appel à une vaste panoplie d'expériences relatives à la conception d'interfaces, à la recherche d'information, à la documentation en ligne, ou bien au génie logiciel.

Sans méthodologie unificatrice, c'est la créativité subjective des auteurs qui est sollicitée lors de l'élaboration d'un produit hypertexte et porte de lourdes conséquences en terme d'utilisation et de maintenance.

Alors que Jacob Nielsen constate dans son ouvrage **[Nielsen 89]** :

> "*...Just as the films are not made by putting a camera in the front row of a theater, the best hypertexts are not made from text that was originally written for the linear medium. But in the real world we have to respect that large amounts of extremely useful text already exist and can be converted to hypertext much more cheaply than the same information could written...*"

On pourrait rétorquer que les méthodes existent aussi bien au cinéma qu'à la télévision pour porter à l'écran (petit ou grand) une pièce de théâtre, un roman, ou un témoignage. On parle alors d'adaptation et les nombreux romans qui ont fait d'énormes succès cinématographiques sont là pour infirmer cette assertion.

En matière d'hypertextes, ce constat est similaire. Les principales réalisations de réseaux hypertextes ont été faites à partir d'informations initialement prévues pour une représentation sous forme papier.

Les travaux de **[Glushko 88],** visant à porter sur un CD-ROM l'encyclopédie "Engineering Data Compendium" en offrant une forme de consultation hypertexte, en est un cas pratique. Ils soulignent que la compréhension de la structure logique et physique d'un document est indispensable dans le cadre d'un projet important d'hypertexte (tel que leur encyclopédie qui correspond à 3000 pages et à 2000 illustrations pour un coût global de 250 000 dollars). La compréhension des liaisons qui existent entre la structure logique et la structure physique d'un document est un pré-requis indispensable pour définir d'une part la taille et la structure des nœuds et décider quels liens devront être créés.

De la même manière, les travaux de **[Raymond 88]** sur l'hypertexte de "L'Oxford English Dictionary" (12 volumes représentant 250 000 entrées soit 42 millions de mots) confirme la nécessité de tenir compte du support initial (et de l'usage qui en était fait) pour rendre la fragmentation possible et pour imaginer l'utilisation hypertextuelle.

Alors qu'il est possible de faire le tour des raisons qui poussent l'auteur à créer un nouveau nœud au sein du réseau hypertexte (par rapport à la taille d'un écran, d'une page, par rapport à une idée, à la nature de l'information, etc...), il est

beaucoup plus difficile d'expliquer les raisons qui le poussent à déclarer des liens entre les différents nœuds d'un réseau. La pose des liens étant une opération essentiellement manuelle, l'auteur reste seul juge de la pertinence et de la nature du lien qu'il déclare en fonction de l'information qu'il manipule et du type de lecture qu'il préconise pour le lecteur occasionnel. A ce sujet, nous avons vu **[cf. 1.2.1]** que les réseaux utilisaient principalement des liens références et des liens hiérarchiques. Imaginons alors l'usage que l'auteur d'un hypertexte pourrait faire de tels liens. Après avoir fragmenté l'information en autant d'unités que nécessaires, il devra les organiser logiquement en familles distinctes (nœuds participant à l'introduction, à la conclusion, à la bibliographie, aux citations, etc...) puis décider quelles liaisons sémantiques peuvent exister entre les nœuds d'une même famille. L'auteur, lorsqu'il est amené à poser les liens de références peut choisir de les poser en privilégiant l'information contenue dans les nœuds ou bien en s'inspirant du profil du lecteur qu'il imagine. Il pourra ainsi créer un lien entre un nœud et un autre parce que ce dernier développe un point simplement abordé dans le premier nœud. De la même façon, c'est le niveau ou la qualité du lecteur (l'auteur peut juger les connaissances du lecteur insuffisantes pour comprendre l'information contenue dans un nœud) qui pourra justifier de la pose du lien de référence. Quoiqu'il en soit, l'auteur a toute liberté, lors de la pose des liens, de considérer tantôt le premier point de vue, tantôt le second et parfois même, les deux simultanément.

Envisager une typologie des liens reviendrait à connaître exhaustivement l'ensemble des types de liens pouvant exister entre des nœuds d'information et à les adapter en fonction du profil des utilisateurs. Or les travaux de **[Garg 88a] [Richard 90] [cf. 1.2.6]** ont montré que si une typologie des liens pouvait exister, elle n'était exprimable que par le biais des opérateurs ensemblistes génériques permettant la création et la décomposition d'objets complexes.

On pressent bien que les "bonnes" réalisations de réseaux hypertextes passent par le savoir-faire des auteurs aguerris aux capacités des systèmes hypertextes qui sont alors à même de donner à l'information une véritable dimension hypertexte. Mais cette personnalisation des réseaux lorsqu'ils sont créés par des auteurs inexpérimentés soulève d'une part les problèmes de désorientation qui ne manque pas d'affecter l'utilisateur et d'autre part la difficulté de la maintenance des réseaux et fait ainsi renaître avec plus de force la nécessité d'une méthodologie de conception.

1.5.3 Utilisateur et Désorientation

Le phénomène de désorientation dans l'espace hypertexte est présenté comme étant le problème de la navigation au sein d'un hypertexte, or il s'agit de le relativiser compte tenu du fait qu'il n'apparaît que dans les hypertextes de taille importante et qu'il dépend de l'expérience de l'auteur en matière de réalisation de réseaux hypertextes.

Une stratégie habituellement utilisée dans les systèmes hypertextes consiste à proposer à l'utilisateur une carte du réseau afin qu'il puisse, à partir de la vision globale de la structure de l'hypertexte, plus facilement se repérer et éviter ainsi de se perdre. Les systèmes tels que *NoteCards*, *Intermedia*, *StorySpace*, *Extended Books*, et *Neptune* proposent des *browsers* graphiques qui, telle la carte routière, permettent de se situer dans un espace. Un browser peut être global ou local selon qu'il représente l'ensemble des nœuds et des liens du réseau hypertexte, ou uniquement les liens rattachés à un nœud particulier **[Halasz 88] [Conklin 87]**.

[Brown 88a] [Brown 88b] [Brown 89] a cherché à montrer que si la représentation cartographique de la structure du réseau hypertexte pouvait

effectivement faciliter la navigation de l'utilisateur et limiter les dangers de la désorientation, elle n'en constituait pas pour autant la seule solution.

Il prend pour cela une analogie avec la programmation structurée des années 60 qui s'est efforcée d'éliminer les instructions de branchement inconditionnel (programmation sans *goto*) dans le but d'améliorer la lecture des programmes et leur maintenabilité. Les solutions de cette époque n'ont pas consisté à créer des outils destinés à fournir une carte des programmes indiquant tous les débranchements relatifs à des goto mais à porter sur des techniques de programmation les évitant (récursivité, modularisation des programmes, etc...).

Ces techniques de programmation en minimisant l'usage des *goto* (voire même en les éliminant totalement) ne remettaient pas pour autant en cause leur présence à d'autres niveaux de la machine (les binaires des exécutables contiennent en effet des sauts inconditionnels ou *jumps* que l'on peut assimiler à des *goto*!). En appliquant ce principe aux hypertextes et en ne donnant pas à l'utilisateur la perception de la structure physique du réseau, la notion de carte devient alors inutile. L'alternative consiste à ce moment à donner au réseau hypertexte l'apparence d'une structure hiérarchique simple (analogie est faite avec la colonne vertébrale) d'où peuvent partir les liens.

Le système *Guide* de Office Workstation Limited est selon lui, l'exemple le plus probant d'une telle approche.

En plus des procédés techniques destinés à pallier au problème de désorientation (on peut citer le backtracking et l'historique de navigation), les chercheurs **[Brown 88a] [Nielsen 90]** se sont penchés sur le comportement des utilisateurs pour expliquer le phénomène de désorientation. Ils ont mis en évidence trois niveaux de problèmes liés à :

- la navigation dans l'espace logiciel : le lecteur ne comprend pas les facilités de navigation que lui offre le système hypertexte (les communautés susceptibles d'utiliser un système hypertexte peuvent être aussi variées que les disciplines dont l'hypertexte représente l'information et elles n'ont pas nécessairement l'*intuition* du fonctionnement des outils hypertextes),

- la navigation dans l'espace conceptuel : le lecteur ne comprend pas comment l'auteur a organisé l'information pour la représenter sous une forme hypertextuelle (par exemple comment il a défini la hiérarchie et comment il l'a enrichi par les liens supplémentaires). La perception qu'il peut avoir de la manière dont l'information est organisée peut alors entrer en collision avec le modèle de conception de l'auteur et ainsi expliquer des activations de liens inadéquates.

- la navigation dans le texte : le lecteur se perd de la même manière qu'un automobiliste se perd en prenant successivement des routes à droite et à gauche en espérant reprendre son itinéraire initial. Le lecteur d'hypertexte, sans cesse sollicité par des liens à activer, se laisse tenter par des digressions, qui lui font perdre le fil de sa navigation de départ. Les multiples décrochements qu'il a effectués finissent par le laisser à un point du réseau qui ne correspond plus à sa recherche première.

Plus précisément, **[Foss 89]** a mené une étude empirique visant à déterminer les conditions de désorientation du lecteur (taille du réseau, stratégie

d'apprentissage et de gestion d'objectifs, etc...) en dégageant les symptômes observables (nombre de fenêtres ouvertes simultanément, retour systématique sur les mêmes nœuds, fermeture rapide des fenêtres, etc...).

Ce constat ne doit pas cependant conduire à considérer la désorientation comme l'inconvénient majeur à une consultation hypertextuelle. En effet, la désorientation se retrouve tout autant dans le cas d'un livre dont la bibliographie intéresse le lecteur. Si celui-ci consulte les ouvrages relatifs à cette bibliographie et récursivement les bibliographies de ces ouvrages, le phénomène de désorientation par rapport à l'espace de connaissances s'installe de la même manière.

Enfin, au problème de désorientation, vient s'ajouter un phénomène de surcharge cognitive (*cognitive overhead*) **[Astofi 93]** qui provient de l'incapacité du lecteur à retenir l'essentiel de l'information liée à un nœud parcouru ou de ses difficultés à identifier rapidement la nature des liens qu'il a activés.

[Conklin 87] :

> *"...cognitive overhead : the additional effort and concentration necessary to maintain several tasks or trails at one time..."*

[Osterbye 92] :

> *"...the amount of mental effort that must be devoted to the non-task-related aspects of doing something..."*

1.6 Texte et Hypertexte

[McKnight 88] après avoir évalué trois systèmes hypertextes (Guide, HyperTIES et HyperCard), montre que de nombreux efforts (par les concepteurs des systèmes et les auteurs) sont encore à faire pour que l'hypertexte puisse être considéré un jour comme une alternative valable au document papier. En effet, celui-ci bénéficie d'incontestables habitudes et conventions d'écriture (et de lecture) qui en rendent l'usage facile et efficace. Les nombreuses expériences en psychologie cognitive **[cf. 1.5.1]** ont montré que la lecture d'un document sur le support papier restait meilleure à la lecture de ce même document sur écran (contextuellement, ce résultat est obtenu en demandant aux lecteurs de trouver dans un document de référence les réponses à certaines questions. Ce document de référence étant présenté alternativement sous forme papier et sur écran). Ainsi, c'est en terme de stratégies de lecture et de manipulation des documents que doivent se développer les hypertextes, afin qu'ils offrent aux lecteurs des méthodes de lecture aussi adaptées aux dispositifs de visualisation électronique que celles qu'ils utilisent déjà pour la lecture des documents papier.

Néanmoins, nous ne pouvons nier que l'hypertexte se présente comme un système d'information original permettant à tout lecteur de se déplacer intuitivement dans un espace documentaire théoriquement illimité.

Bien que les pionniers, Bush, Engelbart et Nelson n'aient pas traité de l'hypertexte en terme de taille, il est clair que les projets qu'ils souhaitaient voir aboutir, sous-entendaient des volumes colossaux d'informations, hétérogènes et potentiellement sans limites. Les systèmes qu'ils avaient imaginés devaient être capable de maintenir la rapidité des associations quelque soit la quantité d'informations stockée. Sur ce point, les récentes réalisations de systèmes

hypertextes ont confirmé que l'efficacité de leur fonctionnement (et l'aisance d'utilisation) était indépendante de toute considération quantitative. En effet, plus que dans les volumes eux-mêmes, c'est dans la méthodologie à mettre en œuvre pour présenter des volumes importants d'information sous forme de réseaux hypertextes que réside la pierre d'achoppement **[Frisse 88a]**.

Parallèlement à cette situation, on voit se dessiner une impossibilité à gérer manuellement les documentations papier volumineuses telles que les encyclopédies générales, médicales et techniques, les documentations techniques des centrales nucléaires ou des matériels militaires (documentations composées de centaines de milliers de pages) **[Ventura 88]**.

Dans ces domaines particuliers, bien que l'usage des systèmes électroniques d'édition performants soit largement répandu et qu'ils s'avèrent extrêmement efficaces en matière de représentation et de mise en forme des documents, ils se montrent en revanche inadaptés pour assurer la mise à jour et faciliter une utilisation en-ligne de la documentation. C'est ainsi que les temps de mise à jour, de réédition et de distribution des documents, augmentent les dangers d'une information incomplète, inexacte et périmée. A cela vient s'ajouter les problèmes de manutention d'une documentation purement papier qui devient rapidement intransportable et difficilement stockable.

En effet, les documentations et manuels techniques qui accompagnent systématiquement tout matériel sont non seulement destinés à en décrire le fonctionnement (et le mode d'emploi) mais aussi à permettre sa réparation en cas de défaillance. Or la complexité de certains équipements (voitures, avions, centrales nucléaires, etc...) introduit naturellement une complexité dans la documentation qui peut rapidement devenir inutilisable (les schémas par exemple

deviennent de plus en plus complexes, la maintenance d'un appareil peut faire appel à une multitude d'autres documents, etc...).

La documentation électronique s'impose alors comme une solution à ces problèmes de volume, d'inexactitude et de complexité. Bien que les systèmes de bases de données documentaires offrent un maximum de garanties dès que les aspects mise à jour, stockage et accès à l'information sont envisagés, ils impliquent cependant, une démarche rigoureuse dans l'expression des requêtes de consultation et exclue une navigation intuitive.

Chapitre 2

Hypertexte, Bases de Données et Intelligence Artificielle

2.1 Synergie des techniques

Avec l'hypertexte, nous avons pu constater que la formalisation est difficile à établir et la conception relève moins d'une véritable méthodologie que du savoir-faire des auteurs rompus à la technologie hypermédia. Il n'y a pas davantage de véritable "traitement électronique de l'information" ; les systèmes hypertextes ne synthétisent ni ne déduisent de nouvelles informations ou de nouveaux faits. Ils réalisent rapidement au gré des sélections du lecteur, l'enchaînement de connaissances distribuées en unités d'information. Ils n'attachent à cette navigation d'autre signification que celle que le lecteur aura pu y mettre.

Laisser le lecteur libre d'ouvrir son propre chemin de connaissances à travers l'information a entraîné des problèmes de désorientation et de surcharge cognitive **[cf. 1.6],** mais aussi d'insatisfaction quant aux réponses attendues. Ces problèmes dissimulent cependant une fabuleuse liberté d'exploration inconnue du monde des bases de données ; l'utilisateur n'a nulle obligation d'exprimer clairement ses attentes pour être en mesure de les traduire explicitement en un quelconque langage de manipulation. Sa quête d'information ne passe plus par le

filtre de la structure mais directement par l'information elle-même. En cela, cette démarche s'oppose totalement à celle adoptée dans les systèmes de gestion de bases de données. Dans ces derniers, c'est grâce à la structure générique des données que la cohérence des informations est vérifiée, et c'est toujours grâce à la structure que le système peut répondre exhaustivement aux questions de l'utilisateur. Nous décrivons dans ce chapitre quelques tentatives de rapprochement entre bases de données et hypertexte destinées à apporter aux hypertextes la puissance d'extraction d'informations factuelles propres aux SGBD.

Par ailleurs, chercher à améliorer la navigation dans les réseaux hypertextes passe par une aide à la navigation "intelligente" qui permettrait de réduire considérablement les risques de désorienation. C'est par le biais de l'Intelligence Artificielle que l'on peut envisager des systèmes hypertextes adaptatifs qui seraient capables de reconnaître le profil de l'utilisateur et de s'adapter à ses besoins d'information. L'application de l'I.A. au monde de l'hypertexte s'est concentré d'une part sur l'interface des systèmes hypertextes et d'autre part sur l'organisation sémantique des informations. Nous présenterons quelques travaux qui ébauchent l'assistance idéale aux lecteurs dans le but de les mener dans un espace hypertextuel sémantiquement plus proche de leurs préoccupations.

2.2 Bases de données et hypertexte

Les systèmes de gestion de bases de données ont cherché à reproduire le fonctionnement des entités évoluant dans un certain monde réel (très souvent celui de l'entreprise) grâce aux différents modèles de données élaborées. En permettant de comprendre et de décrire les entités et les relations existant entre ces entités, les modèles conceptuels ont rendu possible les implémentations. Les premières conceptualisations relationnelles ont laissé apparaître des différences entre la perception que l'utilisateur pouvait avoir du monde réel dans lequel il évoluait et la représentation informatique de ce même monde réel. Ces différences, suffisantes pour troubler et égarer l'utilisateur final, sont les conséquences des carences fonctionnelles du modèle relationnel et de la forte influence des contraintes liées à l'implémentation physique. Mais c'est moins les problèmes d'organisation des données que l'incapacité du modèle relationnel à exprimer avec exactitude la sémantique des données qui explique ces différences entre perception de l'utilisateur et réalité conceptuelle. Les nombreux modèles de données qui ont été proposées visaient à aplanir ces différences.

On peut désormais admettre que les modèles sémantiques sont capables d'enrichir le modèle relationnel afin que les utilisateurs disposent d'une représentation logique du monde réel proche de leur perception. Mais, bien que les modèles sémantiques soient appropriés pour exprimer finement des fonctionnements, ils s'avèrent inadaptés pour décrire des comportements. Car c'est de comportement dont il s'agit quand l'utilisateur cherche à intervenir dans le monde réel (de son entreprise). On peut se demander, même dans le cas où l'utilisateur dispose d'une représentation logique en phase avec sa perception du monde réel, s'il y a équivalence entre intervention physique et manipulation de la représentation logique.

Nous pouvons mettre cela en évidence en prenant l'exemple d'une entreprise qui fabrique en grandes quantités, à la demande de ses clients des bijoux fantaisie en résine ou en métal. Lorsque le client a créé le prototype du bijoux, le fabricant réalise à partir de ce prototype les moules qui permettront la production industrielle. Dès que les moules sont terminés, le processus de commande se déroule de la manière suivante : le client demande la production d'une certaine quantité de pièces qui amène le fabricant à lui établir un bon de commande. En fonction des impératifs de production, le fabricant honorera la commande en une ou plusieurs livraisons (le fabricant remet alors à son client, à chaque livraison un bon de livraison qui lui permettra d'établir sa facture).

Supposons qu'après avoir passé sa commande de pièces, le client s'aperçoive qu'il lui faut revoir à la hausse les quantités commandées. Il avertit le fabricant qui accroît alors la production de pièces qu'il livrera en une ou plusieurs fois. A l'issue de la production complète, le nombre de pièces livrées et facturées se trouve être supérieur au nombre de pièces initialement commandées.

Informatisées et modélisées en tenant compte des obligations comptables et des nécessités de gestion de production, les relations clients/fournisseur vont être modifiées. En effet, en raison de contraintes référentielles entre quantités commandées et quantités livrées assujetties à des questions de mise en production, aucune application informatique n'acceptera d'éditer des bons de livraisons (et a fortiori des factures) pour des pièces non commandées. Il n'y aura d'autre choix que la saisie d'une nouvelle commande. Le travail administratif du client s'en trouvera compliqué puisqu'une même commande sera "éclatée" en plusieurs bons de commande.

La modélisation et l'informatisation, en offrant au fabricant plus de rigueur dans sa gestion commerciale le contraignent à adopter de nouveaux

comportement qui, dans ce cas précis réduisent considérablement sa souplesse de fonctionnement et son adaptabilité aux demandes de ses clients.

A travers cet exemple, on constate que les entités logiques existant dans ce système d'information dédié à la gestion commerciale d'une entreprise sont équivalentes aux entités physiques du monde réel de cette même entreprise (on retrouve les notions de client, de commandes, de bons de livraisons, de factures, de quantités, etc...). En revanche, le comportement de l'utilisateur diffère selon qu'il manipule les entités logiques du système d'information ou qu'il interagit avec les entités physiques correspondantes du monde réel. Cette différence provient essentiellement des relations existant entre les entités physiques et mises en évidence par la modélisation. Ces relations entre entités physiques se traduisent par des contraintes de dépendances qui, en maintenant la cohérence des entités entre elles, assurent l'unité du système d'information. L'individu, en raison de la fonction qu'il occupe au sein de l'entreprise ne perçoit pas toujours l'intégralité des relations entre les entités. Néanmoins, cette vision réductrice qu'il peut avoir de l'entreprise ne l'empêche pas de jouer son rôle.

Avec un système d'information modélisé, où toutes les relations sont décrites, l'individu (qui devient alors utilisateur) doit s'imprégner de toutes les réalités conceptuelles relatives à l'entreprise, même si celles-ci ne le concernent pas fonctionnellement. Bien que la fonction de l'individu au sein de l'entreprise reste la même, son comportement par rapport à une tâche donnée, variera selon qu'il manipulera les entités logiques du système d'information ou interagira avec les entités physiques de l'entreprise.

Les langages de manipulation des systèmes relationnels occultent totalement cet aspect et contraignent l'utilisateur à connaître non seulement la structure générique du schéma relationnel de la base de données (la métabase)

mais aussi à identifier, dans les schémas de chacune des relations, les entités et leurs propriétés. A cette seule condition, il lui sera possible d'interagir efficacement avec le système de gestion de bases de données. Car, même s'il bénéficie de la non-procéduralité des SGBD relationnels, qui dispense l'utilisateur de décrire les chemins d'accès aux données (le Comment), l'utilisateur se doit d'indiquer les résultats qui l'intéressent (le Quoi) et les éléments de structure qui lui permettront de les obtenir (le Où).

Inversement, les SGBD en s'appuyant sur la structure générique des bases de données offrent toutes les garanties de maintien de la cohérence et de l'intégrité des données. Appliquer aux réseaux hypertextes ces principes d'évolutivité et d'intégrité de l'information revient à extraire des réseaux une structure des données plus riche que les nœuds et les liens élémentaires.

2.2.1 L'hypertexte comme nouveau modèle de recherche d'information

Les réalisations de systèmes hypertextes ont montré leurs capacités à mettre en œuvre des mécanismes de navigation qui deviennent néanmoins insuffisants dès qu'il s'agit de retrouver des données factuelles.

C'est génèralement sur les modèles booléen, probabiliste, linguistique ou que sont conçus les systèmes de recherche d'information **[Van Rijsbergen 71] [Faloutos 85] [Puthuff 78].** Dans son article, **[Agosti 89]** montre que l'hypertexte, bien que ne proposant pas de tels modèles, s'apparente à un véritable système de recherche d'information au même titre que les systèmes de gestion de bases de données.

Selon lui, ce sont les composantes de base des systèmes qui déterminent leur appartenance au modèle de recherche d'information. Les systèmes

hypertextes peuvent ainsi être considérés comme des instances d'un nouveau modèle de recherche d'information car, comme pour les SGBD, ils sont composés de deux parties essentielles.

Dans les SGBD, ces parties essentielles sont :

- l'ensemble des documents. Dans un SGBD, ces documents sont présentés en structure verticale de sorte que tous les documents sont potentiellement égaux et pertinents par rapport à la requête.

- les données auxiliaires (ou paradata). Ce sont les données associées à chaque document et qui représentent la sémantique de l'information. Les données auxiliaires sont utilisées pour choisir et retrouver les documents pendant le processus d'interrogation.

Dans les systèmes hypertextes, ces deux parties essentielles existent mais sont implémentées différemment :

- la collection de documents. Les documents (ou parties de) d'un hypertexte sont structurés en *n* dimensions.

- le réseau de liens qui relient les documents ou les parties de document à la collection. Ce réseau est le moyen de connecter sémantiquement ou structurellement les nœuds entre eux.

Cette implémentation, différente selon les systèmes, induit directement le mode de consultation et de recherche d'informations.

Dans les systèmes hypertextes, cette recherche est assimilée à une navigation dans l'hyperdocument. Si un lien n'existe pas entre des documents qui sont cependant sémantiquement liés, ils ne peuvent être rapprochés par l'utilisateur lors de l'exploration. Le seul moyen de retrouver cette relation est

l'usage de la recherche de texte intégral, de mots-clés ou d'attributs qui sont susceptibles d'être partagés **[cf. 1.2.3]**. Dans l'hypertexte, tous les documents ne sont pas sur le même plan, c'est l'utilisateur qui, au gré de l'exploration du réseau, active, sous sa responsabilité, les liaisons éventuelles entre les unités d'information.

Les liens des réseaux hypertextes jouent le même rôle que les données auxiliaires des systèmes de recherche d'information traditionnels. Mais, dans ces derniers, le système répond à la demande de l'utilisateur par un ensemble de documents non arrangés. L'utilisateur est alors obligé de considérer chaque document avec la même importance.

Pour développer les systèmes hypertextes comme de nouveaux systèmes de recherches d'information, l'auteur suggère de s'orienter vers la représentation de la sémantique des documents. Selon lui, le problème de la représentation de la connaissance est central au domaine de la recherche d'information, il n'a pas été abordé dans les premières générations de systèmes hypertextes où l'important était de résoudre les problèmes liés à la nature des informations manipulées.

D'autres travaux tels que ceux de **[Lucarella 90] [McLeod 88]** confirment cette approche de l'hypertexte comme système de recherche d'information en proposant des modèles qui offrent à l'utilisateur les deux modes d'exploration (*browsing*) et de recherche (*searching*).

2.2.2 Hypertexte et Bases de Données Relationnelles

De nombreux travaux ont amorcé la convergence du monde des bases de données relationnelles et du monde de l'hypertexte. L'idée commune à ces travaux est de donner aux SGBD relationnels (parce que leur puissance et leur efficacité

ont été vérifiées au travers des nombreux systèmes développés ; DB2, ORACLE, SABRINA, etc...) un mode de consultation hypertextuelle et d'associer aux systèmes hypertextes un moyen de gérer la cohérence de l'information en tentant d'en extraire une structure suffisamment riche.

[Schütt 90] traite du système HyperBase qui se compose d'un système de bases de données (SyBase), chargé du stockage des données hypertextes et de la synchronisation des accès concurrents, et d'un moteur hypermédia assurant la manipulation des données stockées et la description de l'interface des applications liées à la sémantique des objets manipulés.

[Clifton 88] présente un modèle de document et son langage de manipulation qui fournissent l'intégrité, le partage, la sécurité et la recherche efficace des documents et qui supportent des vues hypertextes (sous forme de nœuds et de liens) de ces mêmes documents dans un environnement application–serveur.

[Beeri 90] présente un langage logique de requêtes pour les systèmes hypertextes qui permet de passer d'une navigation s'appuyant sur les attributs des nœuds et de leur sémantique à une interrogation formelle par la structure. Ce mode d'interrogation pourrait soulager les lecteurs des dangers de la surcharge cognitive consécutive à l'exploration de grands réseaux hypertextes. Pour que le langage d'interrogation puisse répondre aux requêtes structurelles dans l'hypertexte, les auteurs proposent un modèle de données adapté qui associe à chaque nœud et lien du réseau des attributs et des propositions.

Inversement, l'article de **[Hara 91]** présente le moyen d'extraire de réseaux hypertextes existants une structure globale qui soit exploitable par un modèle relationnel ou Entité-Association. La méthode d'extraction repose sur un modèle

de clusterisation de relation, ACE (*Aggregation Clustering with Exception*) utilisant les agrégats et les exceptions. Ce modèle permet d'une part de remédier aux problèmes de maintenance (et de coûts) des réseaux hypertextes et d'autre part d'améliorer la navigation en rajoutant la dimension structurelle. L'idée consiste à considérer un réseau hypertexte comme un graphe orienté et défini de la manière suivante : G = (V, E) où V représente l'ensemble des nœuds et E l'ensemble des liens. Il s'agit d'obtenir à partir de ce graphe d'entrée, deux graphes de sortie ;

- un *graphe aggrégation* défini par $G_a = (V_a, E_a)$ qui est une simplification du graphe initial,
- un *graphe d'exception* défini par G_X qui représente la différence entre la graphe initial G et le graphe aggrégation G_a. La combinaison du graphe d'exception G_X et du graphe aggrégation G_a permet d'obtenir le graphe initial. Le graphe G_X se compose de liens inclusifs (ce sont des liens qui existent dans le graphe G_a mais pas dans le graphe G) et de liens exclusifs (ce sont les liens qui n'existent pas entre deux nœuds du graphe initial G mais qui ont été créés dans le graphe G_a).

L'intérêt de la méthode ACE réside dans la simplification et la réorganisation du réseau initial et conduit non seulement à réduire la taille globale du réseau, mais aussi le nombre de nœuds et de liens.

2.2.3 Hypertexte et Bases de données Orienté Objet

Les bases de données orienté objet gèrent des données de nature textuelle et graphique et modélisent les situations complexes du monde réel faisant appel à ces données (génie logiciel, CAO, cartographie,etc...). Mais ces systèmes, comme les SGBD relationnelles ne disposent pas d'interface permettant une consultation simple et intuitive de l'information.

En revanche, les systèmes hypertextes offrent une telle caractéristique en proposant la consultation de données de même nature que celles gérées dans les BDOO. Il manque cependant aux systèmes hypertextes des mécanismes qui permettent de déclarer et de gèrer la structure globale des documents et de leurs liaisons.

L'article de **[Wang 91]** présente InterSect ; un système hypertexte générique conçu à partir d'une base de données orientée objet. InterSect rassemble en une seule application le moyen de définir (et de manipuler) des structures de documents complexes et le moyen de naviguer dans le flux de données. Les problèmes relatifs à la désorientation et à la surcharge cognitive (inhérente à la création, à la gestion et à la sélection des liens de navigation) se trouvent résolus par les outils de représentation de données structurelles des SGBDOO.

Dans leur article, les auteurs expliquent que les systèmes hypertextes existants (NodeCards, HyperCard, HyperTIES, Intermédia, etc...) n'ont pas apporté de solutions aux points suivants :

- peu de sémantique dans le modèle des données. Dans certains systèmes, on peut connaître le nombre de liens existant entre les nœuds du réseau mais il est impossible d'obtenir cette même information pour un type de lien précis.
- les nœuds de haut niveau ne modélisent pas correctement les relations hiérarchiques entre les nœuds,
- pas de schéma conceptuel des nœuds et des liens,
- pas de définition de type pour les nœuds et les liens.

A travers InterSect_DM (InterSect Data Model), les auteurs défendent la nécessité de disposer d'un modèle de données dédié aux systèmes hypertextes

supportant une meilleure représentation de la sémantique des données qui permettra alors de :

- rechercher tout type de document,
- explorer toute instance de documents et passer d'une instance à une autre quelle soit ou non de même type,
- fournir un haut niveau de description de la structure des documents (autre les liens organisationnels de base) et des relations entre les documents,
- assister l'utilisateur,
- vérifier l'intégrité des instances des documents,
- assurer le versioning,
- rendre l'applicatif indépendant des données,
- partager les données entre plusieurs utilisateurs.

2.3 Hypertexte et Intelligence Artificielle

Quelques travaux ont amorcé des tentatives de rapprochement entre Intelligence Artificielle et hypertexte. Quatre axes préfigurent la contribution de l'I.A. à l'hypertexte **[Kesselman 88] [Marchionini 88]** :

1) la compréhension de la stratégie globale mise en œuvre par les utilisateurs dans le processus général de recherche d'information. En identifiant les éléments tactiques de cette stratégie, tenter de les appliquer aux systèmes hypertextes.

2) la représentation des connaissances dans les systèmes hypermédia. Chercher à mettre en évidence des méthodes précises de recherche de l'information peut être vain. La solution réside dans une représentation des connaissances proche de la représentation humaine.

2) les systèmes d'aide auto-adaptatifs. Doter les systèmes hypertextes d'une aide "intelligente", qui entraînerait une modification dynamique de l'interface afin qu'elle soit à même d'améliorer et de faciliter la recherche d'information.

4) Le traitement du langage naturel ; en introduisant davantage de sémantique dans les réseaux hypertextes au moyen d'index de termes pertinents automatiquement constitués. La structure relativement pauvre des réseaux hypertextes renseigne peu sur l'information manipulée (contrairement à ce qui se produit dans les bases de données), et disposer de points d'entrées sémantiques orienterait la navigation et améliorerait les recherches d'information de l'utilisateur-lecteur. Dans le chapitre 2, nous traiterons des possibilités d'indexation automatique dans les réseaux hypertextes automatiquement générés **[cf. 2.4].**

2.3.1 Dynabook : le livre dynamique

Dynabook **[Weyer 82]** rassemble les travaux du Learning Research Group du Xerox Palo Alto Research Centers sur la conception de livres électroniques dynamiques à partir de l'observation des comportements de recherche. Dynabook expérimente, les effets des variations de l'interface d'un système d'information électronique sur la stratégie de recherche globale des utilisateurs.

Les travaux sur Dynabook se basent sur le fait que les systèmes d'information électroniques actuels sont considérés comme une alternative au support papier. Mais la compréhension générale sur la manière dont tout individu recherche de l'information étant limitée, ces systèmes sont sous-utilisés et souvent mal conçus. Ils n'explorent pas les aspects tels que l'accès partagé à des versions de documents mis à jour, la présentation personnalisée d'informations complexes, l'aide active dans la recherche, etc... et se limitent à améliorer le côté "tourne-page" du support papier.

Le prototype du livre dynamique Dynabook (conçu à partir d'un manuel d'histoire), a permis d'étudier les pré-requis nécessaires à la conception et à l'évaluation des systèmes d'informations électroniques, et a conduit les chercheurs à s'interroger sur la qualité de l'environnement des systèmes par rapport à l'aide active qu'ils fournissent aux utilisateurs. Dynabook les a aidés à mieux comprendre les tactiques locales de recherche d'information dans le cadre de la stratégie globale de l'utilisateur. Ils ont élaboré un modèle du processus de recherche qui prend en compte l'environnement du "chercheur", ses habitudes de recherche, les questions qu'il est amené à se poser, la réutilisation d'une information trouvée, le type d'information qu'il manipule, les interfaces qu'il utilise. En définitive, les études menées sur le livre dynamique ont mis évidence certains problèmes relatifs à la manière dont le livre dynamique peut aider

l'utilisateur à mieux exprimer ses demandes d'information, à relier des informations entre elles ou bien encore à synthétiser un ensemble de réponses. Sur un plan plus opératoire, les travaux de Weyer montre, sans proposer de solution, qu'il y aurait nécessité à adapter l'interface utilisateur du livre dynamique en fonction du domaine de connaissances traité.

2.3.2 GKBS : Un système hypermédia à base de connaissances

L'article de **[Shen 89]** présente un système hypermédia "intelligent" à base de connaissances ; le *Graphic Knowledge Base System*. Dans cet article, les auteurs expriment l'idée qu'un système hypermédia serait plus performant (dans le sens où il pourrait mieux répondre aux besoins informationnels des utilisateurs) s'il permettait de représenter la connaissance sous une forme qui se rapproche le plus de celle utilisée par l'être humain. Selon les auteurs, l'être humain se représente la connaissance sous forme d'images et de flux de textes mémorisés durablement au moyen de réseaux. Le GKBS associe à un logiciel hypermédia une méthodologie pour représenter et organiser la connaissance sous cette forme.

Dans cet article, les chercheurs ont placé l'essentiel de leur travail sur l'amélioration "technique" de la représentation des connaissances dans les systèmes hypermédia traditionnels plutôt que sur l'identification et la reproduction des mécanismes utilisés par tout individu en situation de recherche d'information.

Ce sont les points suivants, jugés insuffisamment développés dans les systèmes hypermédia traditionnels pour manipuler de la connaissance, qui ont été améliorés dans le GKBS :

- le manque de richesse dans la représentation des objets. Pour qu'une machine puisse traiter de la connaissance, elle a besoin de reconnaître génériquement les différents types

d'objets afin d'éviter des opérations secondaires qui visent à reconnaître les objets en interrogeant spécifiquement les attributs des objets en question. Les systèmes hypermédias standards n'ont pas développé cet aspect typologie des objets de sorte que l'utilisateur doit s'imprégner davantage de la nature de l'objet qu'il manipule accentuant ainsi le phénomène de surcharge cognitive.

- le manque de richesse dans la représentation des liens. Les systèmes hypermédias standards offrent peu de types de liens pré définis parmi les divers types d'objets susceptibles d'être représentés. Or dans le cas d'applications à domaine spécifique, de nombreux types de lien capables de porter des attributs différents sont indispensables.

- Inexistence de procédures opératoires associées aux objets et aux liens. La plupart des systèmes hypermédias sont généralement utilisés uniquement comme simple système de recherche ou comme système de préparation de documents. Ils proposent aux lecteurs des outils de navigation dans un réseau pré défini mais ne permettent pas d'associer des procédures opératoires aux nœuds et aux liens. Dans ces conditions, un système "intelligent" qui doit tenir compte des attributs des objets et des liens ne peut être implémenté. Les auteurs préconisent de pouvoir associer aux objets et aux liens non seulement du texte mais aussi des "procédures exécutables".

2.3.3 Ergolab

Ergolab **[Camacho 91]** est un atelier logiciel, mis au point par Le Centre National d'Etudes en Télécommunications (CNET), et destiné à l'élaboration d'un outil d'aide interactive et évoluée pour le maquettage d'interface homme-machine. A l'origine du projet, il s'agissait de fournir aux concepteurs d'interface une aide en ligne souple et puissante permettant de les assister dans la manipulation des

logiciels de conception d'interfaces ainsi que dans l'activité cognitive liée à une telle conception.

En associant des approches hypertexte (pour l'aisance de la navigation) et système expert (pour l'introduction du raisonnement) à l'aide en ligne, les concepteurs d'Ergolab ont essayé de substituer à une documentation lourde et passive une documentation interactive et adaptable. L'idée principale était de gérer les points d'entrée dans l'hypertexte en fonction du profil de l'utilisateur et du contexte de la demande d'assistance.

Peu à peu, l'atelier Ergolab est passé de l'état d'outil d'aide à la conception d'interfaces homme-machine à l'état d'outil d'accompagnement et de méthodologie pour l'usage des réseaux hypertextes. En couplant hypertexte et système expert, Ergolab est devenu un début de solution au besoin d'aide en ligne pertinente et contextuelle demandée par l'auteur et le lecteur.

Pour l'auteur, c'est une aide "intelligente" à la conception qui lui permet de répondre aux questions suivantes :

- Quelles sont les voies de navigation à proposer ?
- Comment choisir les termes pertinents qui commanderont les débranchements ?
- Comment prévoir le type d'information à fournir à un instant ou à un endroit donné ?
- Comment apprécier l'accroissement de la connaissance du lecteur ?
- etc...

Pour le lecteur, c'est une aide "intelligente" à la navigation qui permet de :

- reconnaître le contexte de la demande d'information,
- prendre en compte l'acquis du lecteur et son évolution (acquis par rapport à l'outil et acquis par rapport à sa recherche d'information et aux connaissances préalables indispensables pour accéder à sa requête),
- offrir des simulations de fonctionnement,
- etc...

2.4 Indexation et recherche documentaire

La conséquence directe de la production automatique de réseaux hypertextes est l'accroissement du volume d'information que les lecteurs devront appréhender lors de leurs futures recherches. Cette masse d'information automatiquement générée fait apparaître avec plus de vigueur les problèmes de désorientation et de surcharge cognitive liés à la consultation des hypertextes.

A l'issue de la phase de génération, les quelques possibilités de navigation disponibles : liens de type références croisées, recherche d'occurrences de chaînes de caractères, tables des matières motrices, *backtracking*, consultation du nœud précédent ou du nœud suivant, vont représenter autant de risques d'errance pour le lecteur. Alors que les éléments moteurs des tables des matières pourront l'aider à s'introduire dans le réseau hypertexte, les recherches d'occurrences de mots et les références croisées vont tout en lui permettant de naviguer, risquer aussi de le perdre.

On sait désormais que la recherche d'occurrences de mots est loin d'être un mode de navigation fiable ; la polysémie des mots qui va immanquablement conduire à un changement de contexte difficilement prévisible, les problèmes de formes fléchies (conjugaison des verbes, accord en genre et en nombre des adjectifs, pluriel spécifique des noms) et la représentativité élevée de certains mots (qui en réduit d'autant leur pertinence) sont autant de difficultés qui vont rendre ce mode de recherche extrêmement hasardeux. Bien que l'approche par coocurrences améliore considérablement la recherche rudimentaire de termes simples, elle soulève d'autres problèmes relatifs à l'appréciation de la proximité entre les termes concernés, à la définition du champ de recherche ou bien à l'ordre des occurrences.

De la même manière, les références croisées extraites par le processus d'hypertextualisation, mettent en évidence la logique de l'auteur qui n'est pas nécessairement la logique que désira suivre le lecteur. Plus encore, comme les références croisées s'appuient sur la dimension structurelle des documents, elles pourront être codifiées (donc non explicites) et obligeront non seulement les lecteurs à activer les liens références mais aussi à s'imprégner de l'information contenue dans les nœuds et de décider si celle–ci correspond effectivement à leur recherche initiale.

Il serait donc intéressant d'offrir aux utilisateurs un moyen "intelligent" de naviguer dans les réseaux hypertextes, qui limiterait les risques de désorientation et de surcharge cognitive. Il est évident que cette approche intelligente passe par l'information, où plus exactement par la connaissance contenue dans les nœuds. Il ne s'agit plus d'offrir au lecteur de nouveaux outils de consultation s'appuyant encore sur la structure des réseaux, mais qui, au contraire lui permettront d'accéder à la sémantique de l'information.

Cette approche qui devra s'adapter à la généricité du processus d'hypertextualisation par rapport aux documents traités, exclut une constitution préalable de thésauri, bases de connaissances ou réseaux sémantiques spécialisés et destinés à des domaines de connaissances précis.

Les programmes d'indexation utilisés dans le domaine de la documentation automatique cherchent à retrouver dans un texte, par une analyse morphologique, lexicale et syntaxique, les éléments de sens caractéristiques. L'indexation de ces termes fournit alors une liste qui peut être considérée comme une représentation du textes, bien plus petite mais suffisamment pertinente pour donner une bonne idée du contenu sémantique du texte.

En associant à chaque nœud des réseaux automatiquement générés, un sous–ensemble de termes pertinents (en l'occurrence des termes techniques spécifiques), on pourrait offrir aux lecteurs, par le biais du mécanisme de recherche d'occurrences adapté à ces expressions pertinentes et non plus à des chaînes de caractères quelconques, un moyen de se déplacer sémantiquement dans le fonds d'information.

Les techniques de traitement du langage naturel issues de recherches au croisement de l'Intelligence Artificielle et de la Linguistique sont principalement utilisées dans ce but **[Jones 86] [Ermine 92] [Bassano 86]** :

- pour améliorer l'indexation de documents et en conséquence la productivité des recherches documentaires,
- pour rendre les systèmes documentaires plus conviviaux, facilitant l'interrogation par des non-professionnels.

2.5 Indexation manuelle ou indexation automatique

L'indexation permet d'extraire d'un document, le thème principal, exprimé au travers de notions décrites par des mots ou de groupes nominaux. L'indexation tout en réduisant l'information offre la possibilité de classer et de rechercher aisément des documents pour un thème donné.

Dans ce but, l'indexation doit résoudre les problèmes d'ordre sémantique posés par le passage du concept à sa représentation par des mots, et d'ordre linguistique posés par les mots eux–mêmes ; synonymies, polysémies, formes flexionnelles, évolution du langage, etc...

L'indexation doit, en définitive, fournir une description pertinente des documents. Cette pertinence repose sur trois critères **[Chartron 89]** :

- l'exhaustivité des termes extraits. Ce critère caractérise la capacité à signaler toutes les notions importantes abordées dans un document.
- la spécificité ou capacité à garder une information précise et non généralisée.
- le degré d'ambiguïté des termes. Il s'agit de réduire ici la polysémie des termes d'indexation.

Jusqu'à l'apparition des premiers outils d'indexation automatique, cette opération était réalisée entièrement à la main.

2.5.1 Indexation manuelle

Ce mode d'indexation a fait l'objet d'une norme **[ISO 85]**, dans laquelle des procédés sont recommandés pour l'analyse des documents, la détermination de leur contenu et la sélection des termes d'indexation. Ces procédés s'appuient sur trois phases :

1) l'examen du document et la définition de son contenu,

2) l'identification et la sélection des notions principales du contenu,

2) le choix des termes d'indexation (pouvant être composés de plus d'un mot).

On pourrait penser que le processus manuel (souvent conduit par des documentalistes) représente la panacée en matière d'indexation. L'article de **[Chartron 89]** montre qu'un tel processus peut aussi conduire à des "effets de bord" :

- effet de généralisation. Certains termes généraux sont introduits de façon quasi systématique par les indexeurs dans le but de traduire le contenu des documents pour des demandeurs potentiels travaillant dans des disciplines variées.

- effet de variabilité dans les descripteurs utilisés. L'indexation d'un même document effectuée par des indexeurs différents débouche rarement sur l'utilisation de descripteurs identiques.

- effet de décalage entre le thésaurus créé et le contenu des documents provenant des délais de mise à jour manuelle. Cet effet est accentué dans des domaines évoluant très rapidement.

2.5.2 Hypertextualisation automatique et indexation manuelle

Associer au processus d'hypertextualisation automatique une indexation manuelle effectuée par des indexeurs qualifiés augmenterait considérablement les délais de mise à disposition des réseaux hypertextes générés. Plus encore, la pertinence des termes retenus reste totalement assujettie à la bonne compréhension de la langue utilisée dans les documents et aux domaines de connaissances qui sont abordés. Une solution, déjà appliquée dans de nombreuses communications scientifiques, consiste à demander aux auteurs eux–mêmes, de sélectionner quelques mots ou expressions clefs reflétant le contenu de leurs documents.

On peut cependant émettre des réserves quant à valeur de l'exhaustivité dont font preuve les auteurs lorsqu'ils établissent les termes pertinents. On peut en effet se demander si les termes retenus tiennent compte du niveau de connaissances des lecteurs. C'est tout le problème de l'exhaustivité interne au domaine traité et de l'exhaustivité externe relative aux notions connexes évoquées dans les documents. La spécificité ou la généricité des expressions clefs proposées influera directement l'utilisateur néophyte qui pourra éprouver des difficultés à les rattacher à leurs domaines d'appartenance.

2.5.3 Indexation automatique

Les techniques élaborées dans le traitement automatique de textes **[Bonnet 91] [Bonnet 89]** (traducteurs, phonétiseurs, correcteurs, etc...) ont permis le développement d'outils d'indexation automatique basés sur trois types de méthodes ; par assignation, statistiques ou syntaxiques.

2.5.3.1 Méthode par assignation

Cette méthode consiste à retenir les mots pertinents d'un document par comparaison à un thésaurus préétabli qui contient l'ensemble des termes décrivant toutes les notions d'un domaine précis de connaissances.

2.5.3.2 Méthodes statistiques

Ces méthodes d'analyse se fondent sur certaines caractéristiques statistiques des documents, comme la fréquence d'apparition de certains mots, ou de certains groupes de mots (co–occurences, ou collocation). L'hypothèse de départ de ces programmes statistiques repose sur le fait que les termes ou les co–occurences des termes sont jugés représentatifs d'un document dès l'instant où ils apparaissent fréquemment.

Lors de l'extraction automatique des termes, est associé une valeur numérique ou poids du terme qui va représenter l'importance d'un terme par rapport aux documents où il apparaît. Des fonctions statistiques permettent d'établir par exemple, le facteur de pertinence d'un terme **[Faloutos 85] [Salton 91] [Gilloux 91] [Bonnet 92]** :

- $FREQ_{ik}$: représente la fréquence d'un terme *k* dans un document *i*.
- Fréquence inverse du document : $\log N - \log(DOCFREQ_k) + 1$, où $DOCFREQ_k$ est le nombre de documents qui contient le terme *k*, et *N* est le nombre total de documents. Cette fonction rend compte de la capacité d'un terme à constituer un bon index.
- Pertinence d'un terme : $FREQ_{ik} * TERMREL_k$, où $TERMREL_k = (r_k/(R - r_k)) / (s_k/(I - s_k))$
  R est le nombre total de documents pertinents, r_k le nombre de documents pertinents qui contient le terme *k*, *I* est le

nombre total de documents jugés non pertinents et s_k est le nombre de documents jugés non pertinents et contenant le terme *k*.

De telles fonctions statistiques servent à la mise en œuvre de techniques dites de "*relevance feedback*" **[Dachelet 90]** qui améliore la recherche d'information : les résultats d'une première recherche sont utilisés automatiquement pour reformuler la requête en accroissant les poids des termes de la requête qui sont présents dans les documents retrouvés jugés pertinents par l'utilisateur, et à l'inverse, en diminuant les poids des termes de la requête qui sont également présents dans les documents non pertinents retrouvés.

Ces traitements statistiques supposent que les mots (pris au sens d'une séquence de lettres compris entre deux séparateurs) ont une valeur linguistique suffisamment stable pour que les calculs auxquels ils sont soumis, autorisent diverses déductions sémantiques. Les nouveaux outils de traitement automatique du langage (traduction, phonétisation, etc...) s'appuient sur une conception plus contextuelle et dynamique des langues **[Balpe 93]**. Ils recherchent la diminution du "bruit" linguistique engendré par les mots seuls, au profit de formes linguistiques plus élaborées (mots composés, expressions figées, unités terminologiques candidates, suite figée d'unités lexicales).

2.5.3.3 Méthodes syntaxiques

L'analyse automatique de textes par méthodes syntaxiques passe par quatre étapes **[Silberztein 93]** :

1) l'analyse typographique qui permet d'identifier les caractères séparateurs dans la phrase et faciliter le découpage en mots (par exemple l'espace, le retour à la

ligne vont identifier les mots simples, les tirets vont permettre de détecter certains mots composés, etc...),

2) l'analyse morphologique qui ramène chaque forme fléchie des éléments du texte à leur forme canonique (lemmatisation).

2) l'analyse lexicale. La forme canonique des mots obtenus après l'analyse morphologique représente une clef d'index qui renvoie aux différentes unités lexicales dans un dictionnaire syntaxique. On conduit à distinguer trois classes de mots : les mots simples (séquence de lettres comprises entre deux séparateurs consécutifs ; espace, virgule, point, deux points, point virgule, etc...), les mots composés (séquences connexes qui incluent au moins deux mots simples (et donc au moins un séparateur) et les expressions figés (syntagmes nominaux plus complexes que les mots composés mais qui ne sont pas des séquences obligatoirement connexes de mots simples).

4) l'analyse syntaxico–sémantique qui tente de lever les ambiguïtés lexicales en tenant compte du contexte des mots dans les phrases. Cette désambiguïsation fonctionne grâce à des règles de grammaire dites locales.

L'analyse morphosyntaxique de documents est d'autant plus intéressante qu'elle est adaptable à des domaines de connaissances quelconques. L'apport d'un telle approche est immédiat pour l'hypertextualisation automatique, puisqu'elle offre, par la constitution d'index pertinents la possibilité aux lecteurs d'hypertextes de naviguer sémantiquement dans les réseaux automatiquement créés. Des couplages de ce type ont déjà été envisagés dans les systèmes experts **[Cavazza 92] [Faribault 84]**, les réseaux sémantiques **[Gilloux 91] [Gilloux 89]**, et les réseaux de neurones **[Biennier 90] [Biennier 89]**.

2.6 Hypertextualisation et indexation automatique

L'extraction automatique de termes pertinents à partir de textes portant sur des domaines de connaissances quelconques est actuellement possible. Des logiciels tels que Lexter, Termino, Intex et Lexinet ont été conçus en ce sens. Afin de démontrer qu'une indexation automatique de "termes pertinents" est adaptable à l'hypertextualisation automatique, nous supposerons dans la suite de ce chapitre, que l'usage d'un de ces logiciels nous est acquis. Nous obtiendrons, à partir de chaque document hypertextualisé, et ce pour chaque nœud généré, une liste de termes. Nous avons vu **[cf. 5.6]** que l'actualisation des réseaux hypertextes pouvait être confiée à un SGBD relationnel, nous montrerons que les termes associés à chaque nœud du réseau peuvent être, eux–aussi soumis à une normalisation relationnelle qui servira non seulement à la mise à jour des termes mais aussi à la préparation des chemins de lecture.

2.6.1 Intex

Le système INTEX **[Silberztein 93]**, développé au Laboratoire d'Automatique Documentaire et Linguistique (LADL) de l'université de Paris 7 rassemble des dictionnaires électroniques (DELAS, DELAC, DELAP) et des programmes d'analyse lexicale. Ce système s'appuie sur des techniques informatiques d'automates finis (Recursive Transition Networks) et constitue un outil de base pour tout traitement linguistique précis. Ces traitements linguistiques sont l'indexation automatique, la recherche de structure morpho–syntaxique donnée ou la vérification orthographique et grammaticale. En ce qui concerne l'indexation automatique, le système INTEX permet d'identifier automatiquement les mots simples, les mots composés et les expressions figées **[cf. 6.2.3.3]**.

2.6.2 Lexter

Le logiciel d'extraction de terminologie LEXTER **[Bourigault 92]** **[Bourigault 94]**, réalisé à la Direction des Etudes et Recherches d'Electricité de France permet d'extraire à partir de corpus de textes rédigés en français une liste d'unités terminologiques candidates, c'est à dire susceptibles de désigner les concepts spécifiques du domaine couvert par les corpus. Dans une première étape consacrée à l'analyse morphologique des textes, LEXTER produit des groupes nominaux grâce à des règles linguistiques de détection de frontières. Dans une seconde étape, LEXTER s'appuie sur une base de règles linguistiques de décomposition pour extraire de ces groupes nominaux des sous–groupes que leur structure grammaticale et leur place dans le groupe désignent comme de bons candidats terminologiques.

2.6.3 Termino

Termino **[David 90]** est un progiciel de dépouillement terminologique assisté par ordinateur conçu et réalisé par le groupe de Recherche et Développement en Linguistique Computationnelle (RDLC) du centre d'Analyse de Textes par Ordinateur de l'Université du Québec à Montréal. Termino effectue une analyse morphosyntaxique des textes qui lui sont soumis, à partir desquels il repère automatiquement des "synapsies" (unités polylexicales dont la structuration relève de la syntaxe et qui occupent la position noyau d'un groupe nominal). C'est dans l'ensemble de ces synapsies que sont retrouvés la grande majorité des termes.

2.6.4 Lexinet

Le logiciel Lexinet **[Chartron 89]** est prioritairement un gestionnaire de lexique. Fondé sur des procédés statistiques et combinatoires, il permet de construire ces lexiques dans les domaines scientifiques et techniques à partir du vocabulaire repéré dans les corpus de documents de type "résumé". Lexinet repose sur des algorithmes flexibles réalisés pour effectuer des traitements linguistiques élémentaires : normalisation des termes par l'étude des suffixes, détection des mots composés par des études statistiques de proximité de chaînes de caractères, synonymie des termes, etc...

2.6.5 Spirit

Le système S.P.I.R.I.T. (Système Syntaxique et Probabiliste d'Indexation et de Recherche d'Informations Textuelles) **[Fluhr 89] [Fluhr 84] [Fluhr 77] [CISI 82]** développé par la société SYSTEX à partir de travaux menés à l'INSTN (CEA) est basé sur un modèle linguistique et statistique. Ce système destiné à l'interrogation de bases de données textuelles quelconques repose sur la comparaison requête / textes afin d'extraire une sous-base en rapport avec le sujet de la requête. Ce système offre la possibilité d'exploiter des liens hypertextes dynamiques **[Fluhr 90]**. Ainsi, la requête n'est plus véritablement une question mais un ensemble de mots décrivant le thème de la recherche. Ces mots sont ensuite soumis au traitement linguistique appliqué lors de l'indexation des textes. L'utilisateur peut alors naviguer sémantiquement dans les textes en utilisant des mots ou des couples de mots liés par des relations de dépendance.

2.7 Eléments essentiels de la synergie

Les travaux présentés qui associent des fonctionnalités hypertextes aux SGBD, aux systèmes à base de connaissances et aux systèmes documentaires soulignent l'apport original de l'approche hypertexte en matière de recherche d'informations.

Nous avons vu qu'une convergence se dessinait entre le monde des bases de données et celui de l'Intelligence Artificielle par le biais du traitement du langage naturel. Les schémas des bases de données relationnelles, se sont complexifiés avec les modèles sémantiques, compliquant davantage la tâche des utilisateurs lors de l'élaboration des requêtes. Il s'est avéré que les langages formels d'interrogation (SQL, ALPHA, LAGRIF, etc...), adaptés pour manipuler des schémas relationnels simples devenaient rapidement inopérants pour traduire la sémantique des données. De nombreux travaux ont montré qu'une interrogation des bases de données en langage naturel était inéluctable d'une part afin de compenser cette situation et d'aider l'utilisateur à poser des questions qu'il était incapable de formuler au moyen d'un langage de requêtes, d'autre part afin de remédier à la crédibilité des réponses données par le SGBD à l'utilisateur en fournissant un contrôle sémantique sur la requête de l'utilisateur qui consiste à reformuler sa requête formelle en langage naturel.

De la même manière, les hypertextes proposent une alternative au mode formel d'interrogation par la structure. Sans chercher à déduire des faits, l'hypertexte amplifie les capacités intellectuelles de l'Homme en proposant des mécanismes associatifs de recherche d'information. Ainsi, les méthodes d'analyse linguistique et statistique utilisées dans les systèmes de recherche documentaire peuvent être mises à profit dans les systèmes hypertexte pour pré-sélectionner les

informations pertinentes et orienter sémantiquement la navigation dans les réseaux.

Enfin, la pauvreté des réseaux hypertextes du point de vue de la structure influe directement sur les capacités des systèmes hypertextes à maintenir la cohérence des réseaux et l'intégrité de l'information.

On voit donc s'esquisser une nouvelle approche dans la gestion de l'information électronique qui, au lieu de privilégier unilatéralement la démarche bases de données, Intelligence Artificielle ou hypertexte conduit plutôt à fusionner ce que chacun des trois mondes offre de mieux à l'utilisateur en matière de recherche et de gestion de l'information.

Chapitre 3

L’hypertextualisation par la structure des documents

3.1 Approche structurelle

C'est à partir de documents techniques existant sous format électronique et initialement destinés à l'impression que nous envisageons de créer automatiquement des réseaux hypertextes. Cette création automatique exige de distinguer au sein des documents les futurs composants élémentaires des réseaux : les nœuds. Le problème majeur consiste alors à déterminer l'unité de ces nœuds.

Bien que la définition précise de ce que doit être un nœud hypertexte demeure floue, il est généralement admis que son contenu peut être de nature hétérogène (texte, image, son, vidéo) et que l'information qui y est présente doit être relativement "autonome". Très souvent, Le contenu d'un nœud est assimilé à une idée.

Déterminer précisément l'unité d'un nœud et donc son homogénéité oblige à nous intéresser directement à la signification de l'information que l'on manipule. Alors que le processus qui consiste à faire migrer une information d'un contexte textuel à un contexte hypertextuel en la granularisant peut paraître trivial

pour un opérateur humain (pour autant qu'il soit familiarisé avec le domaine de connaissances de l'information), il devient extrêmement complexe, voire même impossible dès qu'il s'agit de le faire exécuter par un ordinateur. Ni les travaux menés en informatique linguistique, ni les nombreuses réalisations destinées au traitement du langage naturel ne laissent entrevoir dans un proche avenir une possibilité de rendre les ordinateurs en mesure de "comprendre" toutes les informations contenues dans un texte, quelque soit le domaine de connaissances et a fortiori, de justifier d'un point de vue sémantique sa granularisation (la problèmatique de la compréhension de textes n'est pas triviale, nous orientons le lecteur vers **[Ehrlich 93] [Cavazza 92] [Moch 91]**, pour une introduction à une telle problématique).

Pour les puristes, un bon réseau hypertexte contient de l'information créée directement sous forme de nœuds, respectant les impératifs d'autonomie sémantique et de taille, mais qui saurait tirer partie, au moyen de liens, des possibilités de navigation associative. Or, on ne peut ignorer l'existence de l'important volume d'informations d'ores et déjà disponible sur support électronique qu'il est impensable de vouloir ressaisir sous prétexte que son organisation initiale est inadéquate à une représentation hypertexte. Les outils de conversion dont sont dotés les systèmes hypertextes / hypermédias, et qui permettent d'importer des textes et des graphiques créés sous différents formats n'offrent pas une réponse satisfaisante à cette situation. Bien qu'ils évitent une ressaisie massive de l'information, ils contraignent néanmoins l'auteur à de nombreuses opérations manuelles et ne le dispensent pas d'une refonte logico-sémantique des informations pour les adapter à l'hypertexte.

Ainsi, entre l'idéale mais irréalisable granularisation à partir de l'analyse sémantique des documents électroniques existant et le mépris complet de cette

source d'information en raison de sa structuration initiale inadaptée à un usage hypertexte, existe un compromis qui s'appuie sur le découpage des documents et tenant compte de leur structure physique, reflet de la structure logique.

La structure logique d'un document concerne l'ensemble des éléments logiques de type chapitres, titres, paragraphes, énumérations, etc... utilisés lors de la rédaction. Ces éléments délimitent les parties d'un même document et jouent un rôle important dans l'organisation générale du texte.

La structure physique quant à elle, rassemble sur un support papier ou cathodique les caractéristiques typographiques et de mise en page des éléments logiques. Ces caractéristiques qui ont trait directement à la présentation du document, mettent en évidence au moyen de procédés graphiques (fontes, interlignage, cadrage, etc...) les différents éléments logiques utilisés par l'auteur. En cela, la structure physique est dédiée à une vision lecteur et permet à ce dernier une meilleure "prise en main" du document en l'aidant à accéder rapidement et directement à des zones d'information pertinentes. Elle représente une aide à la lecture, une possibilité de sélection visuelle de l'information à appréhender.

Certains documents en raison même de leur nature, ne font pas l'objet d'une structuration logique très poussée. Cela apparaît dans leur présentation et affecte la structure physique qui sera, de manière équivalente, peu élaborée. Ces documents tels que romans, chroniques, articles de presse, récits historiques ou poèmes utilisent en effet peu d'éléments logiques (chapitres, paragraphes) et la représentation graphique, minimale, est de peu d'aide pour le lecteur. Plus encore, ces documents, par les thèmes qu'ils abordent portent en eux une très forte séquentialité qui devient le gage de la cohérence de l'écrit. L'absence de structure physique suffisante et le poids de la chronologie dans le texte, limitant les possibilités de granularisation, rendent ces documents "littéraires" inadaptés à

l'hypertextualisation automatique. Conduire un processus d'hypertextualisation automatique sur de tels documents réduirait totalement les systèmes hypertextes à fonctionner comme de simples "tourne-pages" électroniques assujettis à l'indispensable linéarité des nœuds créés et rendrait inexploitables les mécanismes de navigation.

En revanche, les documents d'origine scientifique et technique traitent d'informations beaucoup moins sensibles à la dimension chronologique et font usage d'une plus grande variété d'entités logiques (titres, chapitres, sections, sous-sections, paragraphes, énumérations, formules, graphiques, etc...) qui déterminent des représentations graphiques plus précises sur lesquelles l'hypertextualisation peut s'appuyer. Dans notre cas, les documents que l'on se propose de traiter sont essentiellement techniques et se rapportent à différents projets relatifs au monde de l'électricité.

Ils concernent soit un état de l'art sur des domaines de connaissances précis (Isolement des lignes électriques aériennes à haute tension, Dossiers des Systèmes Elémentaires relatifs aux centrales nucléaires, guide d'expertise des transformateurs électriques, etc...), soit un descriptif de logiciels industriels (Code de mécanique Aster fonctionnant sur super calculateur CRAY, Plant Design Management System en C.A.O d'ingénierie). Ces documents sont toujours le fait de collectifs d'auteurs rédigeant leurs documents au moyen de logiciels de traitement de textes fonctionnant sur micro-ordinateur. C'est dans le contexte de rédaction de documents destinés au support papier que s'intègre l'hypertextualisation automatique.

En ce sens, nous ne sommes pas en situation de "documents structurés" car il n'est ici ni question de modèles de documents associés à des modèles de traitement et de présentation, ni d'éditeurs structurés. Les notions de structure

logique et physique que nous présentions plus haut sont, dans notre contexte d'étude étroitement mêlées. Nous considérons que, compte tenu de la nature technique des documents rédigés, des règles de présentation adoptées et des outils de rédaction utilisés, il est possible de déduire les différentes entités logiques présentes dans les documents et de permettre une fragmentation cohérente de l'information dans l'optique d'un portage automatique vers un réseau hypertexte.

3.2 Déterminer les nœuds

La conception des réseaux hypertextes de manière automatique ou non, soulève le délicat problème de la détermination des unités d'information manipulées par les systèmes hypertextes.

Bon nombre de chercheurs estiment que l'information manipulée par les systèmes hypertextes doit être, dès l'origine, façonnée pour une utilisation hypertexte. C'est une des raisons pour lesquelles ces systèmes intégrent des éditeurs de textes et de graphiques permettant de créer directement l'information dans les réseaux. Ces systèmes offrent aussi des outils d'importation qui rendent possible la récupération d'informations électroniques existantes mais au prix de lourdes opérations manuelles.

Nous verrons que quelques tentatives ont été lancées pour produire automatiquement des réseaux hypertextes à partir de ces informations électroniques, tentatives d'automatisation qui sont globalement restées à un stade expérimental en raison même de l'origine des informations. En effet, on a souvent reproché à celles-ci, puisqu'elles étaient saisies à partir de logiciels de traitement de textes, d'être trop linéarisées et de ne pouvoir être réutilisées dans les réseaux hypertextes qu'au prix d'une refonte totale de l'information. Cette refonte ne pouvant être menée qu'au terme d'une analyse sémantique effectuée par un être humain.

Alors que les chercheurs s'accordent pour considérer les nœuds d'un réseau hypertexte comme un réceptacle d'informations de nature hétérogène, ils restent beaucoup plus évasif quant à la manière d'appréhender sémantiquement l'unité des futurs nœuds :

[Camacho 91] :

> *"...les nœuds peuvent contenir plusieurs types de médias. Les éléments d'informations, les nœuds, ont un sens, une cohérence interne. Cela peut être un fichier, une page écran ou simplement un terme..."*

[Balpe 90] :

> *"...Définir concrètement ces unités d'information, définir de ce qui va être la part d'information insécable est une première approche analytique indispensable lors de la construction d'un hyperdocument..."*

[Nelson 90] :

> *"...Hypertext is non–sequential writing : a directed graph, were each node contains some amount of text or other information..."*

[Nielsen 89] :

> *"...Nodes are the fundamental unit of hypertext, but there is no agreement as to what really constitutes a node..."*

Cette vision conceptuelle, tend à considérer le nœud comme une entité sémantiquement autonome. Le nœud, souvent assimilé à la notion d'idée, doit comprendre l'essentiel des informations lui permettant d'assurer sa cohérence interne. Mais cette obligation d'autonomie sémantique exige une grande prudence car elle ne doit pas conduire à un isolement des nœuds dans le réseau hypertexte. En effet, en cherchant à trop appuyer la signature sémantique des nœuds, afin que, sémantiquement parlant, ils se suffisent à eux–mêmes, le risque de fermeture grandit de sorte que l'auteur éprouvera rapidement des difficultés à poser des liens entre les nœuds. Ce danger de fermeture sémantique menace de la même manière le lecteur du point de vue de la signification de ses parcours de lecture. L'exploration de nœuds sémantiquement trop autonomes pourra compliquer les tentatives de synthèse des informations glanées dans les différents nœuds.

Alors que cette idée de cohérence interne des nœuds paraît applicable lors de la conception de réseaux hypertextes où l'information est spécifiquement créée et préparée pour une telle représentation, on peut s'interroger sur la facilité à produire systématiquement des unités d'information à partir de leur autonomie sémantique en s'appuyant sur l'analyse sémantique d'un fonds informationnel déjà existant.

Prenons pour exemple, la note de renvoi, souvent utilisée dans les documents papier. Hormis le fait qu'on lui associe une représentation graphique particulière (notes en bas de page ou en fin de chapitre avec une fonte différente de celle du texte principal), on ne peut nier qu'elle apporte des informations supplémentaires mais néanmoins différentes de celles exprimées dans le texte principal.

Supposons que le texte de la note de renvoi soit "*De nombreux traitements de textes intègrent désormais le WYSIWYG qui est un acronyme anglais signifiant What You See Is What You Get que l'on peut traduire littéralement en français par Tel Ecran, Tel Ecrit"*, et qu'il soit rattaché à un paragraphe traitant de l'évolution des logiciels de traitement de textes.

L'auteur communique au lecteur une information périphérique qu'il estime indispensable mais à laquelle il n'associe pas le même "poids" sémantique par rapport au thème qu'il développe dans le texte principal. On constate dès lors, que la note de renvoi dispose d'une cohérence interne mais reste extrêmement dépendante de l'information à laquelle elle est rattachée. Et aussi forte que puisse être cette dépendance, elle ne justifie pas, pour l'auteur, de fusionner l'information de la note de renvoi au texte principal.

Transposer cette situation au contexte hypertexte, reviendrait à créer deux nœuds et un lien de complémentarité entre eux. Dans le premier nœud serait inséré le texte principal alors que le second recevrait le texte "complémentaire" de la note de renvoi. Un lecteur qui parcourrait le réseau hypertexte en suivant les liens auteur, pourrait passer du nœud contenant le texte principal au nœud complémentaire, en appréciant la complémentarité des deux sources d'information. En revanche, si sa navigation privilégie une exploration par la recherche d'occurrences de chaînes de caractères, il pourra se retrouver dans le nœud complémentaire tout en ignorant l'existence d'un lien privilégié avec un autre nœud (tous les systèmes hypertextes ne proposent pas une cartographie détaillée des réseaux!). Il ne pourra nier ni la pertinence du nœud visité ni sa cohérence intrinsèque, mais il aura certainement des difficultés à l'intégrer dans sa logique de parcours et lui donner un sens.

Une solution consisterait à fusionner le texte principal et le texte de la note de renvoi dans le même nœud, mais elle soulève la question de l'ingérence dans la logique de l'auteur.

Alors que l'on pourrait penser que l'analyse sémantique est le facteur décisif dans la granularisation de l'information, dans les faits, l'auteur doit aussi tenir compte des contraintes de représentation électronique de l'information et adapter la taille des nœuds aux impératifs d'implémentation des logiciels hypertextes :

[Akscyn 88] :

> *"...Units of information are displayed on per window. (Systems vary in the number, size and arrangement of windows they permit)...The hypermedia node KMS is the frame, a screen-sized workspace upon which the user can place text, graphics and image items...KMS fixes the size of a frame to a width of 1132 pixels and a height of 805 pixels...The main reason we limit the*

size of a frame is to reduce reliance on scrolling, which is an inefficient way to navigate ina database..."

[Halasz 88] :

"...A notecard is an electronic generalization of the 3x5 paper notecard. Each notecard contains an arbitrary amount of some editable substance such as piece of text, a structured drawing, a bitmap image..."

[Nanard 91] :

"...En fait, ce qui caractérise vraiment un hypertexte, c'est donc la finalité du contenu des nœuds. Leur contenu est avant tout destiné à l'homme. Par conséquent, ce qui caractérise le contenu d'un nœud dans un hypertexte est d'avoir une représentation perceptible par l'homme...Un nœud reprèsente une information. Il doit correspondre à un atome dans l'hypertexte. Or la granularité dépend du point de vue d'observation (...) Des règles de lisibilité au sens le plus large suggèrent naturellement de limiter la taille pour préserver son unité de sens..."

Certes, il est important de rechercher la cohérence interne des nœuds du réseau hypertexte et on peut imaginer de prime abord que seule l'analyse sémantique de l'information peut prétendre à de bons résultats. Cette démarche, aussi performante et efficace qu'elle paraîsse, a aussi ses limites. En effet, elle exige que le concepteur du réseau hypertexte possède le pré-requis suffisant lui permettant de comprendre le domaine de connaissances d'où est tirée l'information qu'il doit fragmenter pour en faire un hypertexte. La citation de Jeff Conklin **[Conklin 87]** résume assez bien ce constat :

"...But this sizing is completely at the discretion of the hypertext writer and the process of determining how to modularize a document into nodes is an art, because its impact on the reader is not well understood...Hypertext invites the writer to modularize ideas into units in a way that allows an individual idea to be referenced elsewhere, and alternative successors of a unit to be offered to the reader (...) But the writer must also reckon with the fact that a hypertext node, unlike a textual paragraph, tends to be a strict unit which dœs not blend seamlessly with its neighbors (...) But the boundaries around nodes are always discrete and require sometimes difficult judgements about how to cleave the subject matter into suitable chunks..."

L'opération qui consiste à déterminer des unités d'informations intrinsèquement cohérentes d'un point de vue sémantique peut être menée par un autre procédé que l'analyse sémantique. Cela dépend du type d'informations à traiter et de la manière dont elles ont été structurées. Car il est possible d'émuler une fragmentation sémantique de l'information en s'appuyant uniquement sur la structure logique des documents.

La manière dont les informations sont structurées dans les documents d'origine scientifique et technique laisse entrevoir une équivalence dans la manière dont ces mêmes informations seront représentées au sein du réseau hypertexte. Prenons l'exemple d'un manuel de référence du programmeur du système d'exploitation UNIX. Ce manuel contient le descriptif complet des commandes et des appels systèmes que le programmeur peut utiliser dans les applications qu'il développe. Chacune des commandes est bâtie sur le même schéma ; on trouve le nom de la commande, un synopsis, une description de la commande, les autres commandes auxquelles elle est associée et les diagnostics relatifs à l'exécution de la commande. La structure même du manuel, indépendamment de toute analyse sémantique pourra être utilisée pour déterminer les unités d'information. Dans le réseau hypertexte correspondant à ce manuel, nous trouverons autant de nœuds qu'il y a de commandes. Chacun des nœuds reprendra les informations du schéma que nous avons donné plus haut.

3.3 Structure des documents

L'information électronique que nous nous proposons de traiter a été produite à partir d'applications de traitement de textes et réapparaît sous la forme de documents destinés à l'impression. La notion de document introduit certes la notion d'information mais aussi la notion de vecteur, grâce auquel l'information sera transmise.

[Covannon 93] :

> *"...The simplest document has two components, the content and its presentation...Other terms for these two components are information and context, content and structure, and vocabulary and grammar..."*

[Quint 91] :

> *"...On peut appeler document le contenu de ce que l'on produit, distribue, utilise ou garde lors d'un processus de communication écrite ou électronique..."*

[Moreau 88] :

> *"...Un document au sens bureautique est un ensemble de caractères respectant un certain formalisme (note interne, lettre, demande de congès, etc...)..."*

[Timbal 84] :

> *"...Le texte est un produit intellectuel reliant entre elles de manière cohérente des idées. De même étymologie que tissu ou textile, le texte résulte donc de la sélection et de l'entrecroissement original des idées de son auteur. Inscrit sur un support, le texte devient document. Le document est donc un texte plus son support..."*

Ces définitions paraissent cependant incomplètes si la notion d'usage n'y est pas associée. En effet, une facture de téléphone, un magazine informatique ou une convocation au tribunal sont tous des documents contenant de l'information et

présentés sur support papier. Mais, chacun d'entre eux a une finalité propre et la manière dont l'information y est représentée renseigne immédiatement le lecteur sur le type du document et son utilisation.

La représentation graphique des zones d'information sur le support physique va permettre au lecteur d'identifier visuellement le type de document auquel il a affaire et d'en déduire les informations qu'il pourra en retirer. La présentation d'une feuille extraite d'un annuaire téléphonique, de la page d'un quotidien ou d'une lettre va déterminer de la part du lecteur, une démarche cognitive adaptée à la spécificité ou à la généricité du type de document.

La représentation graphique des informations dans un annuaire téléphonique est un cas flagrant de type spécifique puisque le lecteur pourra en déduire rapidement que le document contient les nom, prénom, adresse et numéro des abonnés du téléphone. La présentation de la page d'un quotidien concerne un type de document plus générique. A partir uniquement de cette présentation génèrale, il pourra être difficile de déduire le type du journal (politique, économique, spécialisé, etc...). et plus encore à identifier le type d'information dont traite la page (politique intérieure et étrangère, vie sociale, économie, etc...). En ce qui concerne les documents de type lettre, il est évident que la généricité du type est encore plus importante et l'identification précise de l'objet de la lettre simplement en tenant compte de sa représentation graphique générale, beaucoup plus difficile.

Ces derniers types génériques possèdent, eux-aussi leur représentation graphique spécifique que le lecteur pourra mettre à profit dans ses stratégies visuelles d'identification du document et d'extraction d'informations pertinentes.

Pour le document de type quotidien, la sélection visuelle de l'en-tête de la feuille de journal permettra au lecteur de reconnaître le nom du journal (et d'en déduire la tendance), et le repérage des manchettes (toujours en gros caractères) l'aidera à retrouver les différentes rubriques de la page.

Cette stratégie visuelle intervient également sur les documents de type lettre. La zone expéditeur par exemple, traditionnellement située en haut et à gauche de la lettre permettra au lecteur de situer le "contexte" de la lettre.

Malgré la diversité des types de document existant les éléments atomiques logiques utilisés sont systématiquement les mêmes. On retrouve ainsi les entités logiques titres, sous–titres, chapitres, sections, sous-sections, paragraphes, sous–paragraphes, énumérations, en-têtes, notes de renvoi, etc...**[Furuta 87a] [Furuta 89b] [Furuta 89d]**.

Les différentes entités logiques, combinées, composent la structure logique et donnent une signature particulière à chaque type de documents. Cette structure logique est une structure abstraite totalement indépendante de la représentation graphique du texte.

Ainsi un roman de Georges Pérec, un article de politique intérieure du journal *Le Monde* et un rapport technique possèdent leur propre structure logique. Observé sous l'angle des entités logiques utilisées, la structure logique du roman est assez similaire à celle de l'article de journal puisqu'elle se compose, l'une et l'autre, essentiellement de paragraphes (et accessoirement de chapitres). En revanche, le rapport technique pourra faire appel à toutes les entités logiques dont nous avons donné, plus haut quelques exemples .

D'un point de vue graphique, le roman se distingue facilement d'un article de journal. Cela tient aux règles de présentation adoptées pour chacune des

structures logiques. Ces règles de présentation affectent la taille, l'épaisseur et le style des différentes polices de caractères utilisées ainsi que la mise en page des documents (espacement vertical et horizontal). L'article de journal adopte souvent le multi-colonnage et des tailles de fontes réduites. La structure physique rassemble donc, pour un document donné, l'ensemble des attributs de présentation affectés aux différentes entités logiques présentes dans le document et contribue à lui donner une plus grande lisibilité.

L'utilisation de la structure physique comme moyen de granulariser l'information n'est possible que dans les documents en possédant effectivement une et a fortiori, une logique, variée. Les documents "littéraires" tels que romans, articles de journaux ou lettres ne font pas appel, comme nous l'avons vu, à une grande diversité d'entités logiques. Dans ces documents, c'est la séquentialité du raisonnement et des idées qui est importante. Cette séquentialité s'exprime au travers d'un nombre restreint d'entités logiques (le paragraphe en l'occurrence). En ce sens, ces documents sont de mauvais candidats à l'hypertextualisation. L'hypertexte fait ressortir une organisation non linéaire pour pouvoir accéder directement aux unités du document, les documents littéraires en raison de leur linéarité, ne sont pas adaptés à cette forme de représentation.

[Harmon 89] dans un article consacré à l'évolution des publications scientifiques et techniques, montre que les publications scientifiques, s'organisent historiquement en trois catégories principales ; l'*experimental paper*, le *developmental paper* et le *theorical paper*. Ces trois catégories s'articulent autour d'une même structure logique générique appelée structure topique (*topical structure*) qui comprend plusieurs entités logiques différemment développées en fonction de la nature de la publication. Ces entités sont l'en-tête, le résumé, l'introduction, les détails de l'expérimentation ou les méthodes utilisées, les

résultats obtenus, les commentaires sur les résultats, la conclusion, les remerciements et la bibliographie.

Dans le même esprit, deux normes issues de l'ISO (*International Organization for Standardization*) **[ISO 82] [ISO 86a]** confèrent aux documents techniques et scientifiques, aux thèses et aux documents assimilés une "présentation" particulière.

[ISO 82] :

> *"...La présente Norme internationale détermine la manière génèrale dont les rapports scientifiques et techniques doivent être présentés. Elle donne les règles relatives aux articles pour lesquels une procédure uniforme facilitera l'échange d'information, soit en favorisant la compéhension du lecteur, soit en facilitant le développement des rapports dans un système d'information..."*

[ISO 86a] :

> *"...la présente Norme internationale établit les règles pour une représentation rationnelle des thèses et documents assimilés, quelle que soit la discipline..."*

Corollaire à ces normes, la norme **[ISO 78]** précise la manière de numéroter les divisions et subdivisions dans les documents écrits afin de "*...mettre en évidence la succession et l'importance des différentes divisions et subdivisions, ainsi que leurs différentes relations...*", "*...simplifier la recherche de certains passages du texte et permettre de citer des parties de ce texte...*" et "*...faciliter les citations à l'intérieur de l'écrit lui–même..."*

On peut remarquer ici, que les deux premières normes disent traiter de présentation alors qu'en fait, il faut comprendre qu'elles s'attachent à définir le mode d'organisation logique des documents. La présentation, en rapport direct avec la structure physique a pour objet les caractéristiques typographiques et de mise en page du document. C'est dans cette acception que nous employons le

terme présentation dans cette thèse. Par rapport à cette définition, la présentation dont traite les normes se limite à l'insertion de lignes blanches qui en séparant les diverses entités logiques, aèrent le texte et le rendent plus lisible.

Ces normes mettent aussi en évidence le fait que les documents scientifiques et techniques traitent de sujets dont la structuration en éléments distincts est plus aisée que dans les documents littéraires. Si on reprend la structure topique des publications scientifiques décrite par Mark Harmon, on constate que la fragmentation du document dans un processus d'hypertextualisation pourra s'inspirer directement des composants logiques de cette structure topique.

Ces normes, dont il faut bien se rendre compte qu'elles sont génériques à tous les documents scientifiques et techniques, insistent sur la possibilité de créer des divisions et des subdivisions dans un même document, c'est à dire qu'il est possible de distinguer des entités logiques relativement autonomes les unes des autres. Une présentation spécifique associée à chacune des entités logiques va permettre de les distinguer plus rapidement sans avoir besoin de recourir à la lecture.

Nous considérerons dans cette thèse, que les attributs typographiques et de mise en page d'un document technique et scientifique relèvent moins des conditions économiques et sociales de l'édition (par exemple les formats et le style éditorial imposés par certaines collections) que de la volonté de l'auteur d'assister le lecteur dans la compréhension du document et lui faciliter le repérage des zones d'informations pertinentes. Cette idée n'est pas nouvelle et quelques chercheurs l'ont déjà exprimée **[Tazi 89] [Southall 88] [Virbel 87]**. Ils considèrent que le marquage visuel contribue au sens du texte à tel point que l'identification de certaines unités logiques est réalisable au travers de leurs attributs de présentation.

Dans le domaine de l'OCR (*Optical Character Recognition*), de nombreuses techniques de reconnaissance de la structure logique des documents imprimés s'inspirent de cette constatation en utilisant les propriétés typographiques du texte pour en déduire les entités logiques **[Ingold 90] [Ingold 87] [Derrien 89]**.

Dans le cadre de l'hypertextualisation automatique présentée dans cette thèse, nous considérons que les documents techniques et scientifiques, parce qu'ils sont sujets à une décomposition structurelle poussée, sont constitués, de par la matière qu'ils traitent, d'une juxtaposition de blocs d'informations distincts, relativement autonomes les uns des autres d'un point de vue sémantique.

Les outils d'édition et de formatage de textes **[Quint 87]**, mettent désormais à la disposition des auteurs un large éventail de possibilités typographiques permettant entre autres, de mettre en valeur leur documents. Plus que de la mise en valeur, le formatage des textes, comme le soulignent Jacques Virbel et Saïd Tazi, contribue au sens en caractérisant et identifiant le texte formaté. Ainsi, les titres des chapitres commencent toujours au début d'une nouvelle page. Très souvent centrés, ils utilisent un corps de police nettement supérieur à celui utilisé dans les paragraphes. Les différentes sections d'un même chapitre possèdent eux–aussi un formatage particulier ; ils utilisent un corps plus petit que celui du titre du chapitre mais supérieur à celui des paragraphes, le texte de la section est souvent graissé et cadré à gauche. La figure suivante illuste notre propos :

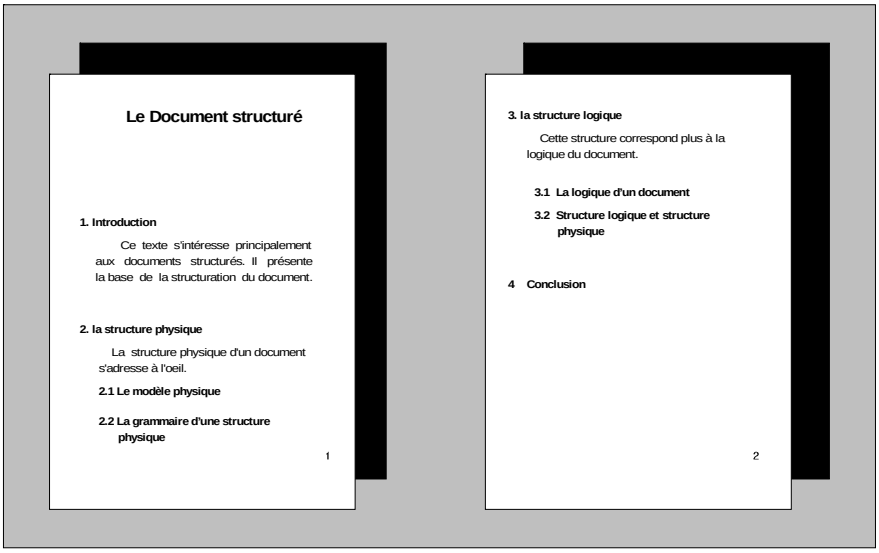

Figure 6 : Le formatage des unités logiques d'un document

Dans le but de rendre possible une hypertextualisation automatique, nous appliquons le mécanisme inverse à celui pratiqué dans la rédaction de documents. Nous relevons les caractéristiques typographiques et de mise en page du texte qui nous permettent alors de déterminer les unités logiques manipulées. En fonction du niveau logique, nous décidons ou non de la fragmentation. C'est expérimentalement ce que nous avons tenté de faire avec certains documents techniques produits à la Direction des Etudes et Recherches d'Electricité De France. Nous décrivons plus précisément la fragmentation dans le chapitre 5.

3.4 La documentation technique à l'Electricité De France

Les documents techniques candidats à l'hypertextualisation automatique se rattachent à différents projets en cours de développement à la Direction des Etudes et Recherches d'Electricité De France :

- Le Code de Mécanique Aster© est un code de calcul fonctionnant sur mainframe CRAY YMP. Ce code est un outil destiné à la modélisation numérique en Mécanique des Structures (Mécanique vibratoire, Thermique, Mécanique statique, Thermo–mécanique, Mécanique non linéaire). Il est destiné aussi bien à des activités de Recherche/Développement, que pour des études industrielles de composants de matériels, de machines de production et de transport d'électricité. Ce code fait l'objet d'une documentation importante destinée aux utilisateurs (un manuel de référence où sont formulés les phénomènes modélisés, un manuel d'utilisation où le langage de commandes est décrit, et un manuel de validation qui contient des dossiers de test dans tous les domaines de modélisation du code), aux développeurs (un descriptif informatique des algorithmes utilisés) et aux partenaires du projet (un dossier Administration traite du plan Qualité Logiciel et du Cahier des Charges des versions) **[cf. A.1]**. Ce code de mécanique est en perpétuelle évolution, les mises à jour des programmes et de la documentation interviennent une fois par semaine.

- Le Dossier des Systèmes Elémentaires (DSE) destiné à la conception des centrales nucléaires. Ce dossier rassemble des documents d'étude servant de recueil aux caractéristiques fonctionnelles nécessaires à la passation des marchés de fourniture de matériels et à la réalisation des circuits électriques. Il sert de base à la préparation des essais et regroupe de ce fait des informations sur le fonctionnement du système élémentaire considéré : son

rôle, les caractéristiques et descriptions des matériels constitutifs, les conditions de fonctionnement, le contrôle–commande, la sûreté, les principes d'entretien et d'exploitation, etc...

- Le comportement diélectrique des isolations dans l'air. L'objectif du projet est de rassembler et d'organiser la connaissance relative aux isolations qui interviennent dans le dimensionnement des lignes et des postes de manière à fournir à des non-spécialistes des directives de dimensionnement diélectrique qu'ils auront à appliquer autant pour EDF que pour des clients étrangers, (notions de base, démarches utilisées, outils du dimensionnement diélectrique dans l'air des lignes et des postes). Cette connaissance est aussi destinée à fournir à des spécialistes un état de connaissances en matière de coordination d'isolement dans l'air aussi exhaustif et actualisé que possible (paramètres qui interviennent dans le dimensionnement des lignes, de l'appareillage des postes, des transformateurs et des câbles) **[cf. A.2]**. Une quinzaine d'auteurs participe à la rédaction des documents traitant de ce domaine.

- Le logiciel de Conception Assisté par Ordinateur ; *Plant Design Management System.* Ce logiciel prévu à l'origine pour concevoir des réseaux de tuyauteries a été modifié pour répondre aux besoins des bureaux d'étude de la Direction de l'Equipement d'EDF œuvrant pour la construction des centrales nucléaires. La documentation initiale de PDMS se présentant davantage comme un manuel de référence, l'adaptation du logiciel à la représentation des structures de Génie-Civil et de réseaux de charpentes a été prétexte à l'écriture d'un véritable manuel d'utilisation intégrant toutes les fonctionnalités du logiciel de CAO.

Les documents rédigés dans le cadre de ces différents projets sont le fait de nombreux auteurs. Ceux–ci utilisent un logiciel de traitement de textes (principalement Word de Microsoft®) fonctionnant sur micro–ordinateurs. Est confié à chaque auteur une feuille de style qui permet d'homogénéiser la présentation de ses documents. Cette feuille de style **[Boin 90] [Sabot 92]** contient un ensemble de propriétés typographiques destinées à être appliquées aux différents élements textuels que l'on rencontre dans les documents techniques (titres, sections, paragraphes, formules; tableaux, dessins, bibliographie, etc...) **[cf. A.3]**. Elle fait ressortir l'intérêt d'une structuration physique des documents techniques en termes de lisibilité et de lecture fonctionnelle. Une description plus précise de ces documents est donnée en **[cf. 5.6]**.

Compte tenu du contexte rédactionnel, aucune contrainte ne peut être exercée sur l'auteur tant du point de vue de l'organisation logique de ses documents que de leur présentation. L'auteur reste totalement libre d'organiser logiquement ses documents, d'y appliquer les styles comme il souhaite, ou de modifier les caractéristiques typographiques des styles définis. Aucun lien, aucune dépendance n'existe a priori entre les éléments logiques du document et les attributs de formatage. C'est l'auteur qui jugera pertinent d'associer telle ou telle représentation graphique à telle ou telle entité logique de son document. Les styles définis ne font que mettre en évidence d'une part les habitudes d'écriture des auteurs de documents techniques et d'autre part l'organisation logique de ces documents (au sens des normes). Ces styles présument que les choix de présentation adoptées pour chacune des entités logiques convergent avec les pré–supposés de l'auteur en matière de présentation de titres, sections, paragraphes, etc...

L'idée sous-jacente à l'élaboration de cette feuille de style était une simulation du fonctionnement d'un système d'édition de documents structurés à partir de logiciels de traitement de textes fonctionnellement non structurés. Cette simulation ne pouvait, bien sûr s'appuyer que sur une démarche volontaire des auteurs. C'est le souci de convaincre les auteurs d'une logique de présentation facilement adaptable à l'organisation logique d'un document technique qui a guidé le choix des styles disponibles et l'élaboration de leurs caractéristiques typographiques.

3.5 Documents structurés

Par rapport au contexte d'écriture que nous venons de décrire, nous ne nous trouvons pas en situation de "documents structurés", car il n'est ici ni question d'éditeurs structurés, de modèles logiques de documents ou de normes. Steven Newcomb dans son article **[Newcomb 91]** définit précisément la notion de documents structurés :

> *"...Structured documents are so named because the hierarchical and sequential structure of the various kinds of information they contain is made explicit by identifying tags. Each tag associates a "generic identifier" - the name of the kind of thing being tagged (e.g., "subsection") - with the data surrounded by a start tag and end tag of the same generic identifier..."*

Dans les documents structurés, on met en évidence les documents sous l'angle de leur structure logique indépendamment de leur représentation physique **[Quint 87]**. On s'intéresse à définir des modèles de documents qui vont permettre par la suite, de produire des instances structurellement conformes du point de vue logique et manipulables au moyen des entités logiques les composant.

Les normes SGML (*Standard Generalized Markup Language* **[ISO 86b]**) et ODA (*Open Document Architecture* **[ISO 89]**) sont souvent associées à la documentation structurée. En effet, ces normes visent à décrire principalement la structure logique de modèles de documents (ODA intègre aussi la dimension physique alors que SGML fait appel à une autre norme DSSSL ; *Document Style Semantics and Specification Language*, pour assurer, entre autres choses, le formatage des documents).

La raison d'être de ces normes provient de la multiplication des systèmes de production de documents et des formats propriétaires de représentation qui ont énormément complexifié l'échange de documents sous leur forme électronique.

Ces deux normes ont été mises en place pour résoudre ce problème en proposant des modes de représentation de documents homogénéisés qui faciliteraient les échanges entre systèmes différents.

La norme SGML est d'origine américaine. Elle dérive de GML (*Generalized Markup Language*) d'IBM. La norme SGML est un langage de description de structures génériques. A partir de la définition d'une structure générique, SGML permet de dériver automatiquement un langage de marquage qui est un sous–produit de la définition des structures génériques. Le marquage est de type déclaratif. Les marques (ou *tags*, ajoutés au texte même du document) délimitent et identifient les éléments de la structure et leur associent des attributs. Elles ne portent pas d'indication de traitement. Les traitements ne sont pas définis dans les documents eux–mêmes, mais par les applications qui les associent aux marques fournies par SGML. C'est ce qui permet, par exemple aux formateurs ou à des systèmes documentaires de travailler sur la même description du document. La norme SGML définit donc un méta–langage pour la déclaration des structures des documents et des différents types d'éléments qui les composent. Ces déclarations sont appelées DTD (*Document Type Definition*) et les noms des types d'éléments sont appelés identificateurs génériques.

La norme SGML est devenue la référence incontournable en matière d'échange de documents électronique depuis qu'elle a été adoptée par le Département Américain de la Défense (*Department of Defense*), dans le cadre du programme CALS (*Computer aided Acquisition and Logistics Support*), destiné à gérer et à contrôler les volumes colossaux de documents relatifs à la conception, au développement, à la fabrication, au déploiement et à la maintenance des systèmes d'armement. L'initiative CALS est en fait une méta–DTD qui permet au département américain d'exploiter les documents de ses fournisseurs dans des

systèmes de CAO, d'édition électronique ou de bases de données. Une des conséquences de ce programme a été le développement massif de systèmes d'édition structurés (Grif, FrameBuilder, Edidoc, Author/Editor, etc...) et d'analyseurs SGML (Sgmls, MarkIt, OmniMark, SGML Hammer, Balise, etc...).

De plus, la norme SGML a eu un impact considérable sur les langages de représentation de documents hypermédia. La norme HyTime **[ISO 92] [Newcomb 91] [Covannon 93]** et le langage de représentation HTML (*HyperText Markup Language*) **[Berners-Lee 93]** utilisé dans le *World Wide Web* sont, en effet l'un et l'autre, basés sur la norme SGML. La norme HyTime permet de représenter la structure des documents hypermédias tout en rendant possible l'interopérabilité des hyperdocuments entre eux. L'initiative World Wide Web est un projet visant la création d'un réseau mondial de liens hypertextes. Le langage HTML, qui en revanche n'est pas normalisé, permet d'attendre les informations formatées, réparties sur la planéte au moyen des applications World Wide Web existantes et respectant le format HTML.

La norme ODA est, elle d'origine européenne. Cette norme, contrairement à la norme SGML prend en compte la structure physique des documents en plus de leur structure logique et de leur contenu. ODA permet ainsi de décrire :

- des documents formatés, qui peuvent seulement être affichés ou imprimés selon les intentions de l'émetteur,
- des documents révisables non formatés, qui sont destinés à être édités puis formatés par le récepteur,
- des documents révisables et formatés qui peuvent êre soit imprimés tels qu'ils sont, soit édités et reformatés par le récepteur.

En fonction du type de document manipulé (formaté, révisable non formaté, révisable et formaté), ODA distinguera des structures logiques et physiques qui pourront être spécifiques ou génériques. En plus de ces structures, la norme ODA comporte des profils de documents, qui fournissent des informations sur le traitement des documents (structures et types de contenu utilisés, jeux de caractères, polices, etc...). Enfin, des profils d'applications ont été définis et s'appliquent à des sous-ensembles de la norme :

- Q111 : permet d'échanger des documents principalement textuels en provenance / à destination de systèmes de traitements de texte fonctionnant sur micro–ordinateurs.

- Q112 : c'est un sur-ensemble du profil Q111 et est destiné aux documents incorporant du texte, des graphiques et des images photographiques.

- Q113 : équivalent à Q112 mais orienté vers les systèmes de Publication Assistée par Ordinateur,

  Q123 : destiné au systèmes de messagerie à vocation textuelle.

La norme ODA, bien que plus complète que la norme SGML n'a pas bénéficié du phénomène porteur que provoque les grands projets comme l'initiative CALS. Les applications (éditeurs structurés, convertisseurs) sont encore rares et celles qui existent déjà ou qui sont en cours de développement favorisent nettement la norme SGML. C'est un état de fait qui laisse peu d'illusion quant au choix définitif de l'industrie de l'édition électronique en matière de norme. Plus encore, l'utilisation de SGML dans les langages de description de documents hypermédias (HyTime, HTML) est un signe avant coureur, sinon une preuve d'une telle orientation.

Il est évident que la démarche suggérée par l'édition structurée tant du point de vue des normes que des applications utilisées, rendrait aisée l'hypertextualisation automatique de documents techniques. En revanche, c'est une démarche extrêmement lourde et inadaptée au monde du traitement de textes sur micro–ordinateur, puisqu'elle demande d'une part une modélisation poussée des documents et des traitements associés, et d'autre part l'utilisation de systèmes d'édition complexes qui exigent une grande puissance de calcul (la plupart des éditeurs structurés fonctionnent en effet sur des stations de travail de type scientifique). Un compromis a été trouvé et consiste à produire un fichier balisé SGML à partir de documents provenant de logiciel de traitement de textes (par exemple, le produit *FastTag* d'Avalanche Systems permet ce type de conversion à partir de documents Word ou WordPerfect). Cette solution reste cependant très contraignante pusqu'elle demande à l'utilisateur de définir au préalable une DTD spécifique sur laquelle s'appuiera l'outil de conversion pour produire l'instance du document SGML.

Dans un proche avenir, tout laisse à supposer que les systèmes d'édition structurée pourraient devenir une solution de remplacement définitive à l'hypertextualisation automatique. Ces systèmes intègrent de plus en plus des fonctionnalités hypertextes (principalement en ce qui concerne la pose de liens) et peuvent être utilisés comme *viewer* dès que leur est associé un modèle de traitement destiné à la consultation de l'information sur écran **[Vuldy 93] [Quint 92] [Gay 89]**.

En attendant, les documents au format SGML sont souvent utilisés dans les tentatives d'hypertextualisation automatique. L'étiquetage des entités logiques facilitent considérablement la détermination des unités d'informations du réseau hypertexte.

Le lecteur pourra trouver des informations complémentaires sur la norme SGML dans **[Bradley 92] [Goldfarb 91] [Role 91] [Barron 89] [Howgate 87] [Smith 86]** et sur ODA dans **[Appelt 93a] [Appelt 93b] [Peterlongo 89] [Quint 91] [Appelt 88].** Une comparaison fonctionnelle des deux normes est disponible dans **[Afnor 91].**

3.6 Les expériences d'hypertextualisation automatique

Bien que la transposition vers un réseau hypertexte des informations prévues à l'origine pour des documents papier représente une aberration pour de nombreux chercheurs, néanmoins plusieurs expériences ont été menées en ce sens. les stratégies mises en œuvre dans ces expériences d'hypertextualisation automatique s'appuient sur :

- la structure logique des documents. Le principe consiste à déterminer des zones d'informations par le balisage (au sens SGML) du contenu des documents. L'exemple le plus marquant en terme de volumes est certainement l'hypertextualisation de l'*Oxford English Dictionary*, **[Raymond 88]**. Bien que la conversion de l'OED sous forme hypertexte se veut automatique, il faut savoir que tout le pseudo-balisage SGML identifiant les entrées du dictionnaire et leurs caractéristiques principales a été réalisé à la main. De plus, dans cette expérience, on ne peut guère parler d'une démarche particulière pour déterminer les unités d'informations puisque la structure globale de l'OED a été reproduite telle quelle dans l'hypertexte ; à chaque entrée correspond un nœud. Une autre démarche consiste à créer automatiquement un réseau hypertexte à partir de documents stockés dans une base de données relationnelles **[Giordano 92] [Rada 91].** Dans cette base de données, ce sont les entités logiques constituant les documents qui sont représentées. Comme dans l'exemple précédent, il faut savoir distinguer ici le processus automatique du processus manuel. La conversion des entités logiques stockées dans les tables relationnelles vers le réseau est automatique. En revanche la décomposition du document en entités logiques représentables dans les tables relationnelles a été effectuée manuellement par l'auteur.

- la structure physique des documents. L'article de **[Furuta 89c]** met en évidence l'utilisation des commandes de formatage de **troff** pour déterminer les futurs nœuds du réseau hypertexte. Ces commandes insérées par l'auteur sont syntaxiquement différentes du reste du texte et précèdent la séquence de caractères qu'elles doivent formater. C'est cette approche que nous avons adoptée dans notre processus d'hypertextualisation automatique, mais appliquée aux systèmes de traitement de textes interactifs. Avec ces systèmes, l'auteur n'a plus besoin d'introduire ces commandes de formatage dans le corps du texte, ce sont des raccourcis claviers ou des sélections par menu qui lui permettent d'obtenir le même formatage et évitent ainsi de "polluer" le contenu du document avec des informations de contrôle d'impression.

- l'analyse morpho-syntaxique des éléments structurels **[Frisse 88a] [Frisse 88b] [Furuta 89c]**. Cette approche s'appuie sur la reconnaissance de repères syntaxiques pour déduire des éléments structurels, qui seront à leur tour utilisés pour déterminer les unités d'information du réseau hypertexte correspondant. La numérotation des chapitres et sections, les chaînes de caractères en lettres majuscules, les lignes blanches sont autant d'indices qui vont concourir à l'identification des entités logiques de type titre, section, sous-section, etc...Cette approche, bien sûr, n'est valable que pour des documents syntaxiquement parfaits. Espérer généraliser une telle démarche à des documents variés demanderait un pré–traitement (dont on voit mal comment il pourrait être autrement que manuel), visant à déterminer les éléments structurels syntaxiquement homogènes qui organisent logiquement les documents.

On pressent bien que l'approche qui prend en compte la structure logique des documents dans l'hypertextualisation automatique est certainement la plus efficace. Mais elle exige que tous les maillons de la chaîne d'édition de documents

structurés soient présents ; modèle de documents, modèle de traitements, éditeurs, etc... et paradoxalement, nous avons vu dans la section précédente que si une telle chaîne existait, elle rendrait totalement obsolète toute tentative d'hypertextualisation automatique.

En général, les travaux de création automatique d'hypertextes que nous avons vus, sont loin de vérifier le caractère automatique du processus d'hypertextualisation et n'offrent pas de réponse satisfaisante quant à la détermination des unités d'information et à l'estimation de leur cohérence interne.

On peut constater aussi la quasi-inexistence de liens dans les réseaux automatiquement générés. En dehors des liens structurels (nœud suivant, nœud précédent, nœud table des matières), le lien de type occurrence d'une chaîne de caractères est parfois disponible. La mise en évidence de cette chaîne de caractères cible se fait au moyen de balises **[Raymond 88] [Giordano 92]** ou de commandes de formatage physique **[Rada 91]**.

De plus, on peut remarquer dans toutes ces expériences d'hypertextualisation que chaque réseau hypertexte généré provient d'un seul document source. Dans aucune des approches n'est ébauchée la possibilité de générer un réseau à partir de plusieurs documents.

L'hypertextualisation automatique que nous proposons dans cette thèse, tient compte d'une production continue de documents à intégrer au fur et mesure dans le réseau hypertexte. Cette réalité nous a conduit à mettre au point une méthode empirique qui nous permette dans un premier temps de réaliser cette incrémentation et dans un second temps de remédier aux problèmes d'actualisation du réseau hypertexte.

Chapitre 4

Le processus d'hypertextualisation automatique

4.1 Un format pivot : le Rich Text Format

L'ensemble des documents techniques relatifs aux différents projets exposés dans cette thèse **[cf. 3.4]**, ont été rédigés à partir d'applications de traitement de textes fonctionnant sur micro-ordinateur (Word MacOS / DOS / Windows). Les documents destinés à l'hypertextualisation n'ont pas été récupérés sous le format natif des logiciels de traitement de textes mais sous le format d'échange R.T.F (Rich Text Format) défini par MicroSoft **[Microsoft 90]** et laissé dans le domaine public.

Ce standard a été défini pour faciliter l'échange de textes et de graphiques formatés provenant des nombreuses applications existantes (Works; Word, Excel, FrameMaker, WordPerfect, etc...) et fonctionnant sous des environnements différents (MSDOS, OS/2, Windows, MacOS, UNIX, NextOS, etc...).

Un fichier RTF est constitué de texte littéral et d'informations de contrôle, appelées "mots de contrôle" apparaissent dans des "groupes". Ce sont les caractères de la table ASCII 7 bits (*American Standard Code for Information Interchange*) qui sont utilisés pour coder ces informations de contrôle.

Un mot de contrôle est une commande utilisée pour piloter le périphérique d'impression (fonte, gras, italique, etc...) ou pour gérer le document dans le logiciel de traitement de textes (marges, index, table des matières, signets, etc...).

Les groupes de mot de contrôle permettent de représenter aussi bien la "structure physique" du document que sa "structure logique". Dans la "structure physique", on retrouve toutes les propriétés de mise en forme du document (marges, orientations, pagination, etc...), des paragraphes (alignement, interlignage, espacement avant et après, etc...) et des caractères (police, couleur, taille, espacement, etc...). Dans la "structure logique", sont explicitement définis les notes de bas de page, les annotations, les en–têtes, les pieds de page, les entrées de l'index et de la table des matières. Aux groupes décrivant les entités logiques, sont associées les commandes de formatage physique.

L'existence de la dimension logique et physique au sein d'un même fichier R.T.F. pourrait suggérer que ce standard est une alternative valable aux normes SGML et ODA **[cf. 3.5]** qui, ont été conçues, elles aussi, pour simplifier les échanges de documents entre systèmes différents. En effet, nous avons vu que la norme SGML ne pouvait représenter que la structure logique des documents, et qu'en aucune façon elle ne pouvait décrire leur formatage, ceci étant laissé à la norme DSSSL. La norme ODA, capable d'exprimer la structure logique et physique du document est extrêmement complexe à mettre en œuvre et manque d'implémentations.

En fait, les quelques distinctions logiques exprimées par des groupes spécifiques du standard R.T.F. ne font que traduire les fonctionnalités du logiciel de traitement de texte et non pas la logique du document. Alors que les normes SGML et ODA peuvent traduire toutes les entités logiques susceptibles d'être

rencontrées dans un document, dans un fichier R.T.F., il est impossible autrement que par le formatage de distinguer d'autres entités logiques que le paragraphe (dans le sens, bloc de texte). Les normes SGML et ODA intègrent la notion de modèle de document qui permet d'identifier et d'organiser génériquement les entités logiques entre elles et de faciliter ultérieurement leur manipulation.

Le format RTF ne fait que coder la représentation graphique d'un document destiné à un périphérique d'impression. Les deux normes internationales sont beaucoup plus généralistes et peuvent associer différents traitements aux documents (transformation de la structure logique, chargement ou extraction des bases de données, etc...).

4.2 Extraction des nœuds

Nous avons vu que les rédacteurs des documents techniques pouvaient utiliser une feuille de styles leur permettant d'associer à chaque type d'entité logique, le même formatage. Cette feuille de style rassemble toutes les caractéristiques d'un "modèle physique" du document.

A condition que cette stylisation soit faite à bon escient, on obtient une correspondance directe et univoque entre style utilisé et entité logique formatée. Il devient donc possible de manipuler les entités logiques seulement par l'image de leur représentation physique. Les documents techniques structurés que nous traitons correspondent schématiquement à une grammaire "logique" **[cf. tableau 1]** et vérifient également une grammaire "physique" **[cf. tableau 2]** :

Document technique ::= Titre du document **ET** Texte1 **ET** Section1
Titre du document ::= *chaîne de caractères*
Texte1 ::= Sommaire **OU** Résumé
Sommaire ::= *chaîne de caractères*
Résumé ::= *chaîne de caractères*
Section1 ::= Titre Texte Section1 **OU** Titre Texte Section2
Section2 ::= Titre Texte Section1 **OU** Titre Texte Section2 **OU** Titre Texte Section3
Section3 ::= Titre Texte Section1 **OU** Titre Texte Section2 **OU** Titre Texte Section3 **OU** Titre Texte Section4
Section4 ::= Titre Texte Section1 **OU** Titre Texte Section2 **OU** Titre Texte Section3 **OU** Titre Texte Section4
Titre ::= *chaîne de caractères*
Texte ::= *chaîne de caractères*

Tableau 1 : Organisation logique d’un document

Document technique ::= Titre du document **ET** Texte **ET** Section1
Section1 ::= TitreSection1 Texte Section1 **OU** TitreSection1 Texte Section2
Section2 ::= TitreSection2 Texte Section1 **OU** TitreSection2 Texte Section2 **OU** TitreSection2 Texte Section3
Section3 ::= TitreSection3 Texte Section1 **OU** TitreSection3 Texte Section2 **OU** TitreSection3 Texte Section3 **OU** TitreSection3 Texte Section4
Section4 ::= TitreSection4 Texte Section1 **OU** TitreSection4 Texte Section2 **OU** TitreSection4 Texte Section3 **OU** TitreSection4 Texte Section4
Titre du document ::= <Police **Times**><Taille **18 point**> <**gras**> <**centré**>
TitreSection1 ::= <Police **Times**> <Taille **14 point**> <**gras**> <**indentation gauche** 1,4 cm> <**aligné à gauche**> <**bordure inférieure simple**>
TitreSection2 ::= <Police **Times**> <Taille **12 point**> <**gras**> <**indentation 1ère ligne** -1,4 cm> <**indentation gauche** 1,4 cm> <**aligné à gauche**>
TitreSection3 ::= <Police **Times**> <Taille **10 point**> <**gras**> <**indentation 1ère ligne** -1,4 cm> <**indentation gauche** 1,4 cm> <**aligné à gauche**>
TitreSection4 ::= <Police **Times**> <Taille **10 point**> <**gras**> <**indentation 1ère ligne** -2,1 cm> <**indentation gauche 2,**1 cm> <**aligné à gauche**>
Texte ::= <**indentation 1ère ligne** 0,7cm> <justifié> <**Police** Times> <**Taille** 10 point>

Tableau 2 : Formatage des entités logiques

C'est à partir de ce formatage et de l'affectation des styles aux différents paragraphes que les unités d'information du futur réseau hypertexte seront déterminées. A chaque document, candidat à l'hypertextualisation, est associé des "règles de découpage" s'appuyant sur les styles utilisés dans le document. Ces règles traduisent en fait les entités logiques qui vont déterminer la granularisation des documents.

Par exemple, un document pourra être totalement inséré dans un seul nœud si le découpage ne prévoit qu'une seule règle s'appuyant uniquement sur le style du titre de document (dans le tableau 2, ce style correspond à "Titre du document"). On pourra faire varier cette granularisation en adaptant les règles de découpage aux différentes entités logiques de type section présentes dans le document (TitreSection1, TitreSection2, TitreSection3, etc...).

Ce processus de granularisation est adaptable à tout type de documents (potentiellement granularisables) dont les entités logiques ont été formatées au moyen d'une feuille de style.

4.3 Repérage des liens références

Une fois les nœuds créés, il s'agit de repérer les liens explicites (ou structurels) qui existent entre eux. Ces liens explicites sont les références croisées intégrées dans le texte par l'auteur afin d'orienter les lecteurs vers une autre partie du document en cours de consultation ou vers un autre document susceptible de leur fournir de plus amples informations un point particulier.

La norme **[ISO 78]** décrit un système de numérotation des divisions et subdivisions dans les documents écrits. Ce système de numérotation met en évidence l'importance des différentes entités du document, et facilite la recherche de certains passages à l'intérieur du document. Dans ce but, la norme préconise l'utilisation des chiffres arabes et conseille une numérotation continue pour chacun des niveaux du document. Cette normalisation de la numérotation influence directement la manière dont les auteurs doivent noter les références dans leurs documents. Ces références doivent inclure la numérotation complète du niveau où se trouve l'information citée par l'auteur (par exemple les expressions de la forme ; voir 4.2, voir 1.1.2.2, [voir 10.2.1], etc...).

Dans les faits, cette normalisation des références dans le texte n'est pas rigoureusement suivie par les auteurs qui adoptent souvent une codification hybride représentant un compromis entre les habitudes d'écriture et la nécessité de fournir une référence précise. Le tableau 3 donne quelques exemples de références croisées utilisées dans des documentations de logiciels. Ces références croisées s'expriment au travers de syntaxes différentes qui restent cependant homogènes dans une même documentation.

Documentation	Exemples de Références croisées	Remarques
La documentation du SGBD relationnel ORACLE pour station de travail SUN (release 7.0.13) est composée d'une dizaine de manuels (Server Concepts Manual, Server Utilities User's Guide, Server Administrator's Guide, etc...)	...See the "Upgrade and Migration Paths" section... ...see the *ORACLE7 Server Migration Guide*... ...refer to "Tunable System Parameters" in the *Operation/System Administration Guide*...	Les références relatives à un chapitre ou à une section du même manuel sont écrites entre guillements. Les références aux autres ouvrages de la documentation sont écrites en italique. Les références à un chapitre ou une section font appel aux deux conventions.
La Documentation du Système d'exploitation Unix System V/386 release 4 éditée par Prentice Hall et Masson (1991)comprend plusieurs manuels (Guide de l'administrateur système, Guide du programmeur; Guide d'évolution, Guide d'Administration et d'utilisation du réseau, etc...)	"Etendre votre réseau", "Installation de TCP/IP", "La sécurité dans un environnement TCP/IP", "Dépannage", *Manuel de référence de l'administrateur système*, etc...	Le texte placé entre guillements ("") reprend le titre d'une section ou dans chapitre. Le texte des références croisées vers d'autres manuels est là aussi placé en italique.
Le guide l'utilisateur de l'environnement graphique d'exploitation Windows de Microsoft comporte une quinzaine de chapitres (Le gestionnaire de programme, Le gestionnaire de fichiers, Le Gestionnaire d'impression, etc...)	«Le gestionnaire de fichiers », «Le Panneau de Configuration», «Libérer de la mémoire », etc...	Le texte placé entre les symboles («) et (») reprend le titre d'une section ou d'un chapitre

Tableau 3 : Documents techniques et leurs références croisées

En ce qui concerne les documents techniques relatifs aux projets présentés en **[cf. 3.4]**, les références croisées obéissent à une syntaxe adaptée à l'organisation des documentations. Celles–ci se composent de manuels, de

fascicules et de documents codifiés **[cf. 5.6]**. A partir de cette codification, une grammaire des références a été mise au point pour repérer précisément un document, une section, un paragraphe, une figure ou une équation mathématique **[cf. Tableau 4]**.

Référence Croisée	Signification
[U2.10.03]	la totalité du document U2.10.03 est référencée (3ème document du 10ème fascicule du 2ème volume du manuel U)
[R5.01.01 §4.2]	référence à la section 4.2 du document R5.01.01
[fig 2.2.3 – a]	référence à la figure 2.2.3 – a du document courant
[10]	référence à la 10ème référence bibliographique du document courant
[§4.1.2]	référence à la section 4.1.2 du document courant

Tableau 4 : Codification et références croisées

Au–delà des différences syntaxiques que l'on peut constater dans ces exemples, on remarque la nécessité de fournir au lecteur toutes les informations indispensables qui vont lui permettre de consulter le manuel, le chapitre ou la section référencée. Automatiser l'extraction des références croisées sous–entendait s'adapter à une multiplicité de syntaxes de références croisées. Le programme d'analyse permettant cette extraction s'appuie sur une description générique de la grammaire des références croisées **[Tremblay 85] [Dechamp 88] [Noyelle 88].** Cette description est faite en utilisant la convention Backus Naur (BNF) qui permet de décrire facilement les constructions autorisées par les grammaires.

4.4 Grammaire de références croisées

La grammaire permettant d'identifier les références croisées dans les documents relatifs aux projets développés à l'Electricité De France est décrite en annexe **[cf. A.6]**. Cette grammaire est adaptable pour prendre en compte les références croisées des documentations données en exemple dans le tableau 3.

Cette grammaire s'exprime dans une notation BNF et utilise des règles de production sous la forme :

PARTIE GAUCHE *constructeur* **PARTIE DROITE**

où

- **PARTIE GAUCHE** est un symbole terminal ou non terminal,
- *constructeur* est le symbole "=",
- **PARTIE DROITE** peut être elle–même composée de plusieurs règles (elles sont alors séparées par le symbole "|"). Chaque règle de la partie droite comprend une suite de terminaux et/ou de non terminaux.

Par exemple, la grammaire qui permettrait de retrouver une référence de la forme **[U.01.10]** (10ème document du 1er fascicule du manuel Utilisateur) s'exprimerait de la manière suivante :

```
Reference =       CrochetOuvrant CorpsRef1
                  CrochetFermant
CorpsRef1 =       Manuel Point Partie Point
                  Document
Manuel =          "U" | "A" | "C"
Point =           "."
```

```
Partie =            Nombre
Document =          Nombre
Nombre =            Digit Nombre
Digit =             "0" | "1" | "2".|...|"9"
CrochetOuvrant =    "["
Crochet Fermant =   "]"
```

Elle donnerait comme résultat :

```
<Ref1>[U.01.10]</Ref1>
<Manuel>U</Manuel>
<Partie>01</Partie>
<Document>10</Document>
```

La grammaire qui permettrait d'extraire les expressions de la forme **(voir Chapitre 1)** s'écrirait :

```
Référence =         Parenthese CorpsReference
                    Parenthese
Parenthese =        "(" | ")"
CorpsReference =    Voir Espace Chapitre Espace
                    Nombre
Chapitre =          Titre
Voir =              "Voir" | "voir" | "cf." | ...
Titre =             "Chapitre" | "chapitre" | "Chap."
                    | "chap." | ...
....
```

et donnerait ces résultats :

```
<Référence>(voir Chapitre 1)</Référence>
<Chapitre>Chapitre</Chapitre>
<Nombre>1</Nombre>
```

La grammaire définie pour extraire les références croisées des documents EDF **[cf. A.6]** est beaucoup plus complexe. Elle permet de repérer les références croisées vers les entités logiques de type chapitres, sous–chapitre, section et sous–

section, mais aussi vers les éléments bibliographiques, les formules mathématiques et les graphiques que ceux–ci fassent partie du document courant ou de documents externes.

En voici quelques exemples :

[§4]	référence au chapitre 4 du document courant
[§4.1.2]	référence à la section 4.1.2 du présent document
[4]	référence au 4ème élément bibliographique du présent document
[U2.00.00 §4.2]	référence au sous–chapitre 4.2 du document U2.00.00
[R5.01.01 fig 2.2.3-b]	référence à la figure 2.2.3-b du document R5.01.01
[D8.01.03 éq 1-3]	référence à l'équation 1-3 du document D8.01.03

Si on reprend la description de la grammaire donnée en annexe, les éléments non–terminaux *RefBibExt*, *RefTitExt_1*, *RefTitExt_2*, etc... permettent de repérer les différentes formes de références croisées :

RefBibExt	éléments bibliographiques externes au document courant
RefBibInt	éléments bibliographiques du document courant (de la forme [4])
RefTitExt_1, ..., RefTitExt_12	éléments logiques de documents externes (de la forme [U2.00.00 §4.2])
RefForExt_1, ..., RefForExt_8	formules mathématiques de documents externes
ReFigExt1_1, ..., RefFigExt_9	illustrations des documents externes

etc...

Les différentes variations que l'on peut observer sur *RefTitExt*, *RefForExt*, *RefTitInt*, *RefFigExt*, *RefFigInt*, etc... permettent de s'adapter à

l'insertion de caractères d'espacement supplémentaires entre les différents éléments des références croisées. A titre d'exemple, le non–terminal *RefTitInt* qui comporte huit variations pourra aussi bien repérer [§4.2.1], que [§ 4 . 2.1], que [§ 4 . 2.1], etc...

4.5 Conclusion

Au–delà de la création dynamique des réseaux hypertextes à partir de documents conçus linéairement, qui conduit à mettre en évidence les grains d'informations autonomes et les liens structurels, on ne peut cacher que l'efficacité du processus d'hypertextualisation dépend, pour une grande part, de la bonne volonté des auteurs. Eux seuls, lors de la rédaction des documents pourront garantir d'une part la justesse et la pertinence de la granularisation, et d'autre part la véritable dimension hypertextuelle de leurs documents. En effet, une conception logique *mal pensée* de leurs documents entraînera inévitablement une fragmentation inappropriée, et l'inexistence de références croisées réduira les réseaux hypertextes automatiquement générés à un usage de type "tourne–page".

Les éditeurs structurés peuvent, dans une certaine mesure, remédier à cette situation, moins d'ailleurs par les techniques qu'ils utilisent que par la démarche de modélisation qu'ils imposent. En effet, la création de documents à partir de ces éditeurs contraint les auteurs à une phase de réflexion pendant laquelle ils vont concevoir l'organisation des différentes entités logiques au travers d'un modèle qui leur permettra, par la suite, de produire des documents homogènes. On peut présumer qu'un tel effort de modélisation profitera directement à la manière dont les auteurs adapteront l'information aux différentes entités logiques présentes dans leurs documents, et par relation de cause à effet, à la pertinence de la granularisation lors du processus d'hypertextualisation.

On commence à voir apparaître dans les récentes versions de logiciels de traitement de textes sur micro–ordinateurs (Word de Microsoft), la notion de "modèle de document" qui, bien sûr, est encore loin de cette même notion induite par les éditeurs structurés, puisqu'il s'agit essentiellement de modéliser la couche

présentation d'un même type de document. On obtient néanmoins, de la part des auteurs, cet effet de réflexion sur l'information présente dans le document et sur la manière de l'organiser, même si cette organisation porte d'abord sur la structure physique.

Dès l'instant où la structuration physique des documents reflète un effort de structuration logique de la part de l'auteur, et si celui–ci a conscience d'intégrer ses documents au sein d'un ensemble beaucoup plus vaste, dont il peut tirer profit au moyen des références croisées, alors l'hypertextualisation automatique révèle qu'un document linéaire, de type technique, possède intrinsèquement des potentialités hypertextuelles. Les documentations de logiciels (systèmes d'exploitation, SGBD, environnements de programmation, etc..), souvent composés de plusieurs manuels, sont un exemple flagrant d'une organisation utilisant intensivement les références croisées. En effet, ces documentations sont conçus pour répondre à un besoin d'information ponctuel de la part des lecteurs / utilisateurs sur des questions relatives à des procédures, des descriptifs de commandes, de fonctions ou de messages d'erreur. Ces questions s'insèrent souvent dans un contexte de connaissances avec lequel doit être familiarisé le lecteur. Les références vers d'autres sections, chapitres ou éventuellement manuels permettent de synthétiser ce contexte de connaissances nécessaire à la bonne compréhension de l'information délivrée, références dont le lecteur pourra user en fonction de ses carences techniques.

Les potentialités hypertextuelles présentes dans ce type de documentation peuvent apporter une réponse intéressante à la carence méthodologique en matière de conception des réseaux hypertextes.

Par ailleurs, le processus d'hypertextualisation automatique tel qu'il fonctionne actuellement, traite exclusivement les objets de type texte. Bien que les

objets de type graphique soient reconnus lors de l'hypertextualisation automatique, la pléthore de formats graphiques (étroitement dépendant de l'environnement logiciel et matériel) rendait improbable la récupération des dessins par les systèmes hypertextes. La normalisation du format graphique CGM (*Computer Graphic Metafile*) sera certainement une réponse à cette situation.

En revanche, l'hypertextualisation automatique n'apporte de réponse ni aux problèmes posées par la taille des nœuds et la nécessité de les fragmenter ni à la consultation d'informations sur écran cathodique. Sur ce dernier point, on sait désormais que la représentation de l'information sur support électronique obéit à des contraintes totalement différentes de celles imposées par le support papier.

Enfin, on pourrait reprocher aux réseaux hypertextes automatiquement générés la pauvreté des liens mis à la disposition des lecteurs. Il est vrai que les liens traduisant la séquentialé du document et les liens structurels des références croisées donnent au réseau hypertexte une connotation de *réseau minimal.* Mais c'est en terme d'évolutivité qu'il faut percevoir ce réseau minimal qui ne demandera qu'à s'enrichir des parcours de lecture des utilisateurs.

Chapitre 5

Hypertextualisation et Actualisation
Apport du modèle relationnel

5.1 Actualisation des documents hypertextualisés

Le chapitre 2 nous a permis d'aborder les points de convergence qui existaient d'une part, entre le monde de l'hypertexte et celui des bases de données et d'autre part, entre l'hypertexte et l'Intelligence Artificielle. Nous avons vu que les travaux de rapprochement entre l'hypertexte et les bases de données tendaient essentiellement à améliorer l'aspect navigation / exploration dans les réseaux hypertextes en leur associant de véritables méthodes de recherche **[cf. 2.2.1]**. Pour ce faire, les travaux engagés cherchaient à montrer qu'il était possible d'extraire des réseaux hypertextes une structure suffisamment riche qui permettrait alors une interrogation de type bases de données.

Par ailleurs, le processus d'hypertextualisation automatique **[cf. 4.1]** conduit à une importante production de réseaux hypertextes. L'hypertextualisation crée un hyperdocument à partir d'un document électronique structuré, et l'itération d'un tel processus mène à la création d'un fonds hypertextuel ; le "méta–réseau", constitué d'un ensemble d'hyperdocuments potentiellement reliés. Comme l'information contenue dans le méta–réseau est amenée à évoluer plus ou moins rapidement, un des problèmes majeurs consiste à propager cette évolution dans le

méta–réseau, principalement quand cette mise à jour concerne les nœuds ou les liens d'un hyperdocument. En effet, une modification d'un hyperdocument qui entraîne l'insertion d'un nouveau nœud ou l'élimination d'un lien explicite influe davantage sur la cohérence du méta–réseau que la simple modification d"information dans un nœud (comme par exemple l'insertion ou la suppression de phrases).

Deux solutions sont envisageables pour remédier à ce problème : la première consiste à regénérer totalement le méta–réseau hypertexte à la moindre modification locale d'un document, la seconde répercute uniquement dans le méta–réseau la modification du document en ne mettant à jour que l'hyperdocument correspondant. En raison de la complexité d'une part à gérer les accès concurrents sur un réseau hypertexte **[Campbell 88],** ces solutions excluent un fonctionnement de l'hypertexte dans un environnement distribué.

La regénération totale de l'ensemble des hyperdocuments résout totalement et simplement le problème des mises à jour mais ne peut être envisagée pour un méta–réseau de taille importante. Recomposer systématiquement l'intégralité du méta–réseau à la moindre modification ou insertion d'hyperdocument entraîne des problèmes de temps de traitement et de stockage. D'une part, à chaque insertion de nouvel hyperdocument, les temps de régénération vont s'accroître. D'autre part, dans le cas d'informations évoluant rapidement et nécessitant des mises à jour fréquentes (documentation de logiciels par exemple), le méta–réseau risque d'être le plus souvent en état de modification (et donc impossible à consulter). Enfin, des problèmes de stockage apparaissent puisque une regénération globale exige d'archiver tous les documents électroniques qui ont participé à la création des hyperdocuments dans le méta–réseau.

La seconde solution élimine les problèmes de temps de traitement et de stockage et est adaptée à un méta–réseau de taille quelconque. L'idée consiste à représenter sous forme relationnelle l'infra-structure des hyperdocuments et de la gérer au moyen d'un moteur de bases de données relationnelles. On voit ici que cette solution exige une plus grande technicité puisqu'elle introduit un outil supplémentaire dans la chaîne d'hypertextualisation et qu'elle demande une normalisation de la structure des hyperdocuments.

Nous montrons dans la suite de ce chapitre que le modèle relationnel défini par E.F. Codd dans les années 70 **[Codd 70]** nous permet de normaliser la structure des documents hypertextualisés ce qui rend possible la mise à jour locale d'un hyperdocument dans le méta–réseau.

5.2 Modèle relationnel versus modèle orienté objet

Les types de données que les réseaux hypertextes sont amenés à manipuler, peuvent être assimilés à ceux manipulés par les systèmes de gestion de bases de données orienté objet **[cf. 2.2.3]**. Ce constat conduit à nous interroger sur la pertinence du choix d'un système relationnel pour gérer l'actualisation des réseaux hypertextes.

En effet, et malgré des capacités puissantes de représentation et de manipulation de l'information, les SGBD relationnelles traditionnels ont d'abord été conçus pour manipuler des données de gestion de type élémentaire (les tables relationnelles contiennent exclusivement des valeurs alphanumériques) et souffrent de limitations majeures dès qu'il s'agit de manipuler des données plus complexes. Les domaines liés au traitement de l'image, à la C.A.O., à la cartographie apportent un ensemble de caractéristiques spécifiques que les SGBD relationnelles peuvent difficilement supportées **[Vidal 93] [Houbart 87]**. Ces caractéristiques concernent bien sûr les types de données (tableaux, cartes, graphiques, images, sons etc...), mais aussi les volumes (alors que pour des données de type texte, l'unité utilisé est le kilo–octet, on parle plus fréquemment de méga–octet pour des données de type image) et les opérations à appliquer à ces données (l'affichage à l'écran d'objets en 2 ou 3 dimensions fait appel à de nombreuses primitives de traçage).

Les SGBDOO se montrent mieux adaptés à gérer des données généralisées tout en continuant à assurer les fonctions de SGBD traditionnels (langage de manipulation ensembliste, gestion de la concurrence, intégrité, etc...). Les SGBDOO montrent la nécessité de définir de nouveaux modèles capables de saisir un objet complexe dans sa totalité sans le décomposer en plusieurs tuples

comme l'impose dans le modèle relationnel, la normalisation des schémas de relations **[Cheiney 89]**.

Dans notre problématique, c'est davantage la structure des réseaux hypertextes automatiquement générés que nous cherchons à gérer plutôt que les données elles–même (dont la manipulation reste du ressort du système hypertexte). Cette structure, qui s'appuie sur la numérotation adoptée par l'auteur dans un même document et sur les titres des différentes entités logiques composant ses documents **[cf. 3.4]**, fait appel exclusivement à des données de type alphanumérique. L'usage d'une numérotation et du titre permet alors d'identifier et d'adresser, au sein d'un même document, chacune des entités logiques composites. On obtient alors une vision simplifiée du document **[cf. 4.6]** qu'un SGBD relationnelles peut représenter et gérer facilement.

5.3 Représentation relationnelle de documents textuels

Pendant longtemps, les bases de données de type relationnel ont été destinées à des types de données précis, parfaitement structurés et ayant une sémantique précise **[Codd 70] [Jault 86] [Miranda 86a] [Miranda 86b] [Miranda 91] [Gardarin 87b] [Delobel 82]**. L'essentiel de la technologie des bases de données a été orientée vers la gestion de ces données de type élémentaire en négligeant les autres classes de données plus générales (mais complexes comme le texte) en raison des difficultés à identifier leurs marques de structure. Néanmoins, plusieurs chercheurs ont montré que les SGBD, prévus pour gérer des données structurées, pouvaient être avantageusement utilisés pour représenter de l'information textuelle au prix de quelques améliorations.

Les travaux de **[Stonebraker 86] [Clifton 88]** ont montré que les systèmes relationnels pouvaient en effet traiter des documents textuels. En intégrant les documents, ils montrent que les SGBD relationnels sont non seulement en mesure de gérer des données moins structurées mais qu'ils peuvent leur apporter des services inconnus jusque là des systèmes traditionnels d'édition de textes tels que le contrôle des concurrences, le contrôle des accès, la reprise sur panne, les possibilités d'indexation et la manipulation des documents au moyen d'un langage de requête. Cette intégration de données textuelles passe par la création de nouveaux mécanismes : chaîne de caractères à longueur variable, relations ordonnées, opérations sur les sous-chaînes, opérations de concaténation. Ces travaux ont surtout souligné la possibilité de manipuler les documents de la même manière que des données plus structurées. Ils n'introduisent pas de processus de normalisation dans la représentation du document puisque l'essentiel des travaux repose sur la représentation de l'objet atomique qu'est le mot et des opérateurs d'agrégation qui permettent de retrouver la dimension du document.

Le modèle relationnel tel qu'il a été défini par Codd est davantage un modèle orienté "données" que "liaisons" entre les données. Avec le modèle relationnel, c'est l'identification des éléments moteurs d'un monde réel qui est recherché. Bien qu'au travers des dépendances fonctionnelles, la notion de liaisons sémantiques soit présente dans le modèle relationnel, elle n'apparaît que dans le processus de normalisation des relations (c'est à cette étape que s'exprime la sémantique entre les différents attributs). Une fois la normalisation effectuée, l'interprétation sémantique des différentes relations devient difficile car aucune sémantique interrelation n'est exprimée dans le modèle relationnel.

Alors que les travaux de Stonbraker et de Clifton tendaient à monter que, fonctionnellement les SGBD relationnels existants pouvaient souffrir la comparaison avec les systèmes de gestion documentaire **[BasisPlus 90] [Heaulme 90]**, au contraire l'article de **[Heather 89]** montre que le modèle relationnel, d'un point de vue conceptuel, en raison de ses insuffisances à exprimer la sémantique entre les données (les associations) est inadapté pour représenter les documents textuels. L'article de Heather souligne que la représentation relationnelle d'un document textuel basée sur le mot conduit à perdre la richesse des associations existant entre les différentes entités subtiles du document textuel.

Une compagnie aérienne, une société de chemin de fer, une coopérative viticole, etc... se composent d'entités significatives (telles que les avions, les pilotes et les destinations pour les compagnies aériennes, les caves, les exploitants pour les coopératives viticoles, etc...) alors que les documents textuels se composent d'entités dont la signature est plus imprécise, c'est davantage la sémantique des associations qui est importante. De ce point de vue les modèles sémantiques de Chen (E-R) **[Chen 76]** et de Codd (RM/T) **[Codd 79]** sont plus appropriés pour représenter les documents textuels.

En effet, un document se compose d'entités de type section, sous-section, paragraphe, sous-paragraphe et note de bas de page. Des liens de type "référence croisée" permettent de relier une ou plusieurs occurrences d'une entité section, sous-section, paragraphe ou sous-paragraphe à une autre occurrence d'entité de type section, sous-section, etc...

De plus, les possibilités d'agrégation et de spécialisation utilisées dans les modèles sémantiques soulèvent le délicat problème de l'unité d'information manipulée à un instant donné. Dans le modèle relationnel, grâce à la première forme normale, cette notion d'unité est ramenée à la notion d'attribut atomique (un employé est représenté par les attributs nom, prénom, salaire, âge, un avion par les attributs type, capacité, vitesse etc...).

En ce qui concerne les documents textuels, cette notion d'unité est beaucoup plus difficile à établir. Les informations inhérentes à certains domaines (musée, médecine, politique, etc...) ne peuvent pas être totalement réduites à une représentation sous forme de champs ou d'attributs dont le contenu et l'usage sont parfaitement déterminés. Souvent, des informations pertinentes sont exprimées dans des zones texte où il est difficile de déterminer la notion d'unité (on peut par exemple associer à une sculpture ou à une peinture des éléments stables comme le nom de l'artiste, l'époque de la création, le matériau utilisé, le nom de l'œuvre, etc... mais les informations relatives au contexte de création, les rapports que l'œuvre entretient avec d'autres créations sont autant d'informations pertinentes dont la structuration est difficile).

Avec la variété des documents textuels (œuvres littéraires, articles de journaux, lettres commerciales, rapports, etc...), il est hasardeux de définir une notion d'unité qui soit indépendante de la nature du document textuel et de l'usage qui en est fait (dans le cas de correspondances commerciales, si on ramène

arbitrairement l'unité à la notion de paragraphe, on peut s'interroger sur la pertinence à "désunifier" le paragraphe consacré à la zone expéditeur, des paragraphes relatifs au corps de la lettre).

Ces différents travaux nous permettent de constater que le modèle relationnel initial est fonctionnellement trop pauvre pour gérer les entités des documents textuels dont la notion d'unité est floue et dont la sémantique est principalement exprimée dans les associations entre les entités. De ce point de vue, les modèles sémantiques corrigent les insuffisances du modèle relationnel mais leur implémentation reste plus difficile.

Mais nous avons vu que, dès que la notion d'unité est précisée (par exemple, le mot chez Stonebraker), le modèle relationnel pouvait représenter un document textuel indépendamment de la richesse sémantique des associations. Le processus d'hypertextualisation automatique permet justement de déterminer les différentes unités structurelles d'un document textuel. Une fois définies et normalisées, ces différentes unités textuelles peuvent alors être représentées sous forme de tables relationnelles. Grâce à cette représentation, nous proposons une vision "administratrice" du méta–réseau hypertexte qui permet de vérifier la cohérence de chacune des unités structurelles des documents hypertextualisés par rapport à l'ensemble du méta–réseau. Le schéma suivant résume l'actualisation locale d'un hyperdocument au moyen d'un système de gestion de bases de données relationnelles qui manipule la représentation structurelle du méta–réseau.

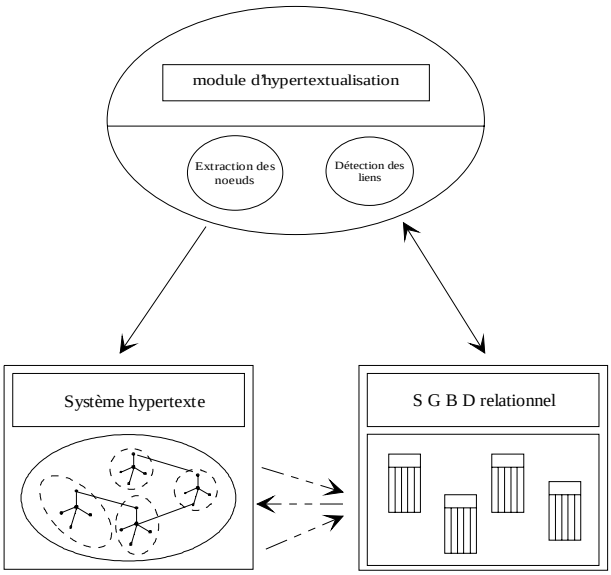

figure 7 : Le processus d'actualisation du méta-réseau hypertexte

Du point de vue de l'actualisation locale des hyperdocuments d'un méta–réseau hypertexte, le modèle relationnel tel qu'il a été défini par Codd se montre suffisamment évolué pour maintenir la cohérence de chacun des hyperdocuments. En normalisant les documents hypertextualisés et les références croisées, il est possible de suivre l'évolution de chaque hyperdocument et de répondre aux questions suivantes :

- Quelles sont les entités structurelles (les nœuds) d'un hyperdocument donné ?
- Pour un hyperdocument donné, quels sont les hyperdocuments auquel il est rattaché ?
- Quels sont les hyperdocuments pour lesquels les références croisées aboutissent à des documents n'existant pas encore ?
- Quel sera l'impact de la suppression de l'hyperdocument *x* ?
- etc...

5.4 Schéma relationnel d'un réseau hypertexte

Nous présentons la normalisation des documents techniques électroniques hypertextualisés grâce au processus décrit dans le chapitre 4. Cette normalisation s'applique sur les entités structurelles (sections, sous-sections, etc...) mises en évidence par l'automate d'hypertextualisation et sur les références croisées extraites des documents grâce à l'analyseur lexical de références croisées. La normalisation de ces documents repose sur le postulat selon lequel chaque nœud d'un hyperdocument est identifiable de manière unique par rapport à son hyperdocument d'appartenance et à tous les autres hyperdocuments générés à un instant donné.

La représentation relationnelle des données sous-entend l'existence de clefs, permettant de distinguer de manière unique un n–uplet. Les documents techniques dont nous proposons d'extraire les tables relationnelles répondent à cette condition. En effet, ces documents techniques relatifs au domaine de l'électricité disposent du point de vue intra-documentaire de structures logique et physique clairement déterminées ; les composants logiques tels que les chapitres, les sections de différents niveaux, les paragraphes, etc... sont identifiées par une numérotation et une structure physique précise.

Chaque document est rattaché génériquement à un titre, un auteur, un service, un projet et une date de création qui, conjointement utilisés, peuvent servir de clef primaire et assurent une identification du document. Néanmoins, les documents d'un même projet sont regroupés sous un plan de classement, permettant dans le cadre de ce projet de les identifier précisément. Ce plan de classement s'articule hiérarchiquement en 4 niveaux **[Boin 92] [Sabot 92]** :

1) un projet distinct (par exemple ; Isolement des lignes électriques à haute tension, Code de mécanique Aster©, Guide d'expertise des transformateurs électriques, Dossier des Systèmes Elémentaires des centrales nucléaires, etc...),

2) domaine de connaissance ; chaque documentation des projets comprend plusieurs manuels décrivant des domaines de connaissances particuliers. La documentation du Code de Mécanique Aster© comprend les manuels suivants : Descriptif, Utilisation, Administration, Références, Validation, etc...

3) chaque domaine de connaissances s'articule en plusieurs parties. Le manuel Utilisation du projet Aster© contient les parties : accès au code, commandes, résultats, messages d'erreur, etc...

4) le document qui correspond à l'élément terminal de cette hiérarchie. Chaque partie d'un domaine de connaissances est constitué de plusieurs documents ayant trait à un même sujet. Dans une même partie, un numéro identifie de manière unique un document et le distingue des autres documents de la même partie.

La documentation relative au projet de Coordination de L'isolement des lignes électriques à haute tension nous fournit un exemple concret d'une telle représentation. Un manuel de cette documentation est entièrement destiné au domaine de connaissances relatif aux Contraintes de Tension. Ce manuel se compose de plusieurs parties, elles-mêmes constituées de plusieurs documents (cf. Tableau 1).

C1.01	**Types et Catégories des contraintes de tension**
	C1.01.00 : Types des contraintes de tension
	C1.01.01 : Catégories des contraintes de tension
C1.05	**Principale caractéristiques des surtensions**
	C1.05.01 : Surtensions Temporaires ou Dynamiques
	C1.05.02 : Surtensions Transitoires à Front Lent
	C1.50.03 : Surtensions Transitoires à Front Raide
	C1.50.04 : Surtensions Transitoires à Front Très Raide
C1.20	**Surtensions Temporaires dues à ...**
	C1.20.01 : l'effet Ferranti
	C1.20.02 : une résonance série
	C1.20.03 : l'auto excitation des alternateurs
C1.25	**Surtensions temporaires dues à la ferrorésonance**
	C1.25.00 : Qu'est-ce que la férrorésonance ?
C1.60	**Surtensions à front raide et très raide dues à la foudre**
	C1.60.01 : Surtensions sur coup de foudre sur les câbles de phases
	C1.60.02 : Surtensions consécutives à un amorçage en retour
	C1.60.03 : Surtensions induites par un coup de foudre proche de la ligne
	C1.60.04 : Propagation des surtensions de foudre vers les postes

Tableau 1 : Projet Coordination de l'Isolement - Manuel des Contraintes (C1)

Du point de vue intra–documentaire, chaque document possède la même structure logique, sur laquelle s'appuie le processus d'hypertextualisation automatique. Un document contient une en–tête et un pied de première page normalisés où sont renseignés :

- le projet auquel est rattaché le document,
- la version du document,
- le titre du document,
- l'auteur du document et le service auquel il est rattaché,
- la clef du document (conforme au plan de classement),
- le nombre de pages du document et la date de création.

Le corps des documents est formé de paragraphes organisés en sections d'au maximum 4 niveaux comme les consignes dactylographiques et de mise en page le préconisent. A chaque niveau est associée une numérotation et un titre. Par exemple, le document C1.30.01 relatif à "L'enclenchement et au ré-enclenchement de ligne à vide" se décompose en plusieurs niveaux comme le montre le tableau 2.

1. Description du phénomène
    1.1 Le phénomène: approche simplifiée
        1.1.1 Conditions initiales: disjoncteur ouvert
        1.1.2 Régime transitoire : fermeture du disjoncteur
            1.1.2.1 Cas de ligne à vide sans tension résiduelle
            1.1.2.2 Cas de ligne avec tension résiduelle: ré-enclenchement
        1.1.3 Régime permanent final: disjoncteur fermé
        1.1.4 Surtension transitoire totale
2 Caractérisation des surtensions
    2.1 Surtensions phase-terre en bout de ligne ouverte
    2.2 Distribution des niveaux de surtension le long de la ligne
    2.3 Surtensions phase–phase
3 Paramètres d'influence des surtensions.

Tableau 2 : Niveaux de structure du document C1.30.01

5.4.1 Attributs du schéma relationnel

Pour représenter un document respectant le plan de classement précédent, sous forme de tables relationnelles, nous allons définir les attributs et leurs domaines associés :

Code_Manuel : Code du manuel (U pour Utilisation, V pour Validation, C pour Contrainte etc...)

Lib_Manuel : Intitulé du Manuel (chaîne de caractères)

Part_Num : Partie dans le manuel (entier)

Lib_Partie : Intitulé de la Partie (chaîne de caractères)

Doc_Num : Numéro du document (entier)

Lib_Doc : Titre du document (chaîne de caractères)

Section_Num : numérotation des sections (au maximum 4 entiers séparés par des points)

Lib_Section : Titre de la section (chaîne de caractères)

Ref_Manuel : références croisées pour le manuel

Ref_Partie : référence croisée pour la partie

Ref_Doc : référence croisées pour le document

Ref_Sec : référence croisée vers une section d'un autre document.

Lib_Réf : libellé complet de la référence (chaîne de caractères)

Les attributs **Ref_Manuel**, **Ref_Partie, Ref_Doc** et **Ref_Sec** permettent d'exprimer les références croisées. Ces attributs sont de domaines compatibles deux à deux avec les attributs **Code_Manuel**, **Part_Num**, **Doc_Num** et **Section_Num**.

L'attribut **Code_Manuel** est la clef qui permet de distinguer chaque manuel. L'attribut **Part_Num** est une clef qui identifie de manière unique une partie dans un manuel. L'attribut **Doc_Num** est la clef qui permet de retrouver un document à l'intérieur d'une partie d'un manuel. Enfin, l'attribut **Section_Num** permet de distinguer les différentes sections d'un même document.

Un document et les références croisées qu'il contient peut s'exprimer sous la forme d'une relation universelle en première forme normale :

Document (Code_Manuel, Lib_Manuel, Part_Num, Lib_Partie, Doc_Num, Lib_Doc, Section_Num, Lib_Section, Ref_Manuel, Ref_Partie, Ref_Doc, Ref_Sec, Lib_Ref)

En raison des redondances importantes qu'introduit cette relation universelle, une décomposition s'avère indispensable. Cette décomposition conduit à découper la relation universelle en plusieurs autres relations. La redondance d'information est alors considérablement réduite puisqu'elle est ramenée à la répétition des attributs identifiant les n–uplets (cette répétition minimale est cependant indispensable car elle traduit les contraintes d'intégrité référentielles). La décomposition qui repose sur les formes normales, met en évidence les dépendances fonctionnelles élémentaires suivantes :

(Code_Manuel) ⟶ Lib_Manuel

(Code_Manuel, Part_Num) ⟶ Lib_Partie

(Code_Manuel, Part_Num, Doc_Num) ⟶ Lib_Doc

(Code_Manuel, Part_Num, Doc_Num, Sect_Num) ⟶ Lib_Section

Ces dépendances fonctionnelles nous conduisent à créer les relations suivantes (les attributs en italique identifient les clefs primaires) :

Manuel (*Code_Manuel*, Lib_Manuel)

Partie (*Part_Num, Code_Manuel*, Lib_Partie)

Document (*Doc_Num*, *Part_Num, Code_Manuel*, Lib_Doc)

Section (*Section_Num, Doc_Num, Part_Num, Code_Manuel*, Lib_Section)

RefCroisées (*Ref_Doc, Ref_Partie, Ref_Manuel, Ref_Sec*, Lib_Ref)

SectionRef (*Code_Manuel, Part_Num, Doc_Num, Section_Num, Ref_Manuel, Ref_Partie, Ref_Doc, Ref_Sec*)

La relation SectionRef est une relation dynamique qui traduit la présence de références croisées dans les sections des documents. Elle exprime le fait que chaque section peut référencer plusieurs autres documents (lien de type 1:N).

5.4.2 Extension réduite des relations

Nous proposons quelques extensions des relations précédemment définies à partir desquelles nous exprimerons des requêtes SQL visant à administrer les hyperdocuments du méta–réseau.

Manuel	
Code_Manuel	**Lib_Manuel**
C1	Contraintes de Tension

Partie		
Part_Num	Code_Manuel	Lib_Partie
20	C1	Surtensions temporaires
40	C1	Catégories des tensions
60	C1	Surtensions à front raide

Document			
Doc_Num	Part_Num	Code_Manuel	Lib_Doc
01	20	C1	Surtensions consécutives à un amorçage en retour
02	40	C1	Catégories des contraintes de tensions
05	60	C1	Une résonance série

Section				
Section_ Num	Doc_ Num	Part_ Num	Code_ Manuel	Lib_Section
1	01	20	C1	Surtensions consécutives à un amorçage en retour
1.1	01	20	C1	Description du phénomène
1.2	01	20	C1	Le phénomène : approche simplifiée
2	01	20	C1	Caractérisation des surtensions
2.1	01	20	C1	Surtension phase-terre en bout de ligne ouverte
2.2	01	20	C1	Distribution des niveaux de surtension le long de la ligne
3	01	20	C1	Paramètres d'influence des surtensions
1	02	40	C1	Catégories des contraintes de tensions
1.1	02	40	C1	Tension permanente d'un réseau triphasé
1.2	02	40	C1	Tension assignée d'un réseau triphasé
1.3	02	40	C1	Tension la plus élevée d'un réseau triphasé U_m
2	02	40	C1	Surtensions
3	02	40	C1	Tensions Continues
1	05	60	C1	Une résonance série
2	05	60	C1	Description du phénomène
3	05	60	C1	Phénomène linéaire
3.1	05	60	C1	Paramètres d'influence
3.2	05	60	C1	Moyens de réduction

RefCroisées				
Ref_Sec	Ref_Doc	Ref_Part	Ref_Manuel	Lib_Ref
1	02	40	C1	[C1.40.02 1]
(∅)*	00	70	C1	[C1.70.00]
1	05	60	C1	[C1.60.05 1]
(∅)*	01	20	C1	[C1.20.01]
3.2	05	60	C1	[C1.60.05 3.2]
1.1	01	20	C1	[C1.20.01 1.1]
2.1	01	20	C1	[C1.20.01 2.1]
3.1.1	02	70	C1	[C1.70.02 3.1.1]

*Les références croisées exprimées sans le numéro de section (valeur nulle ∅ dans l'attribut Ref_Sec) indiquent que le document entier est référencé.

SectionRef							
Section_ Num	Doc_ Num	Part_ Num	Code_ Man.	Ref_ Sec	Ref_ Doc	Ref_ Part	Ref_ Man.
1.2	01	20	C1	1	02	40	C1
1.2	01	20	C1	(∅)	00	70	C1
1.3	02	40	C1	1	05	60	C1
1.3	02	40	C1	(∅)	01	20	C1
2	02	40	C1	3.2	05	60	C1
3	05	60	C1	1.1	01	20	C1
3.1	05	60	C1	2.1	01	20	C1
3.2	05	60	C1	3.1.1	02	70	C1

La figure suivante résume les liaisons inter-documentaires dont nous venons de donner les extensions relationnelles. Les documents sont représentés par des rectangles dans lesquels on a réduit l'extension des documents à la numérotation des niveaux de structure. Chaque document (par le biais des différents élements de structure) est relié à d'autres documents. Ce sont les zones

délimitées par une ligne pointillée qui contiennent la codification précise des documents référencés. Par exemple, nous voyons que la section 1.2.1 du document C1.20.01 référence intégralement les documents C1.50.02 et C1.70.00, alors que la section 3.2 de ce même document contient une référence plus fine vers la section 1.2.1 du document C1.20.01.

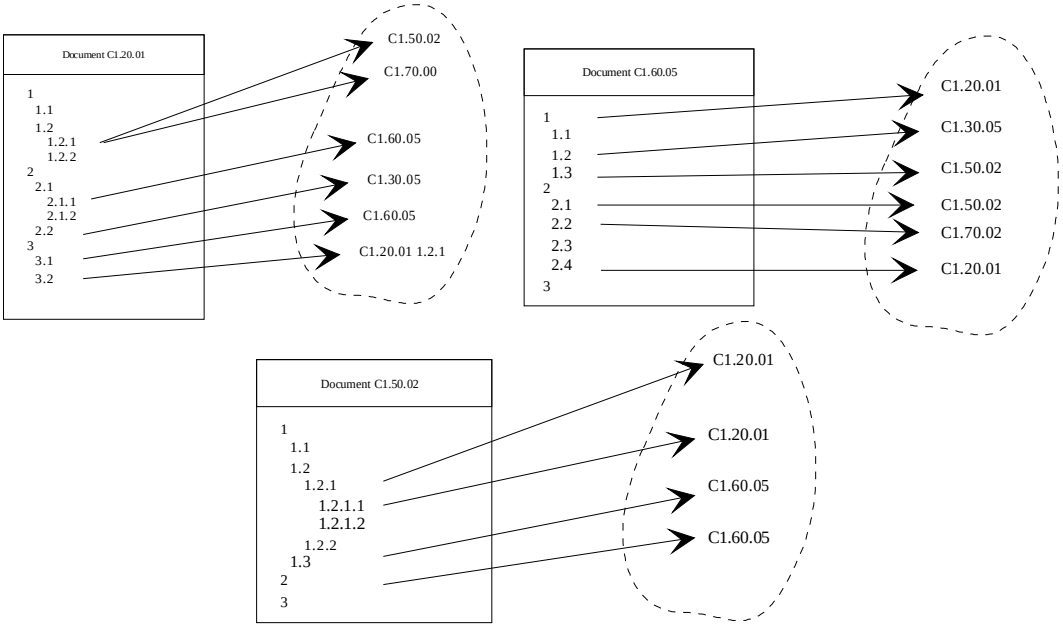

Figure 8 : quelques documents et les références croisées du Manuel Contraintes de Tension (projet Isolement des lignes électriques)

5.4.3 Administration des relations

Comme nous l'avons dit plus haut, l'intérêt d'une telle représentation relationnelle réside dans la gestion de la cohérence des nœuds et des liens des documents hypertextualisés. Une interrogation du schéma relationnel au moyen, par exemple, de requête SQL (Structured Query Language) permet de suivre

l'évolution globale du méta–réseau au fur et à mesure des opérations de mises à jour des hyperdocuments. Nous proposons quelques requêtes d'administration :

1) Quels sont les éléments structurels du document C1_20_01 ?

```
SELECT    Section_Num, Lib_Section
FROM      Section
WHERE     Doc_Num = '01' AND
          Part_Num = '20' AND
          Code_Manuel = 'C1'
```

Avec les opérateurs de l'algèbre relationnelle **[cf. 5.3]**, cette requête s'exprimerait de la manière suivante :

```
R1 = SELECT( Section, Code_Manuel = 'C1' ET Part_Num = '20' ET
             Doc_Num = '10' )

Résultat= PROJECT( R1, Section_Num, Lib_Section )
```

La relation Résultat donnerait alors :

Résultat	
Section_Num	Lib_Section
1	Surtensions consécutives à un amorçage en retour
1.1	Description du phénomène
1.2	Le phénomène : approche simplifiée
2	Caractérisation des surtensions
2.1	Surtension phase-terre en bout de ligne ouverte
2.2	Distribution des niveaux de surtension le long de la ligne
3	Paramètres d'influence des surtensions

2) Quels sont les documents qui traitent des "surtensions à front raide" ?

```
SELECT    Lib_Doc
FROM      Document
WHERE     Part_Num    IN
          ( SELECT    Part_Num
                      FROM      Partie
                      WHERE     Lib_Partie = 'surtension à front raide' )
          AND Code_Manuel = 'C1'
```

Avec les opérateurs relationnels, on obtiendrait :

R1 = JOIN(Partie, Section) /* Jointure naturelle entre les attributs Part_Num, et Doc_Num des relations Partie et Section */

R2 = SELECT (R1, Lib_Partie = 'Surtensions à front raide')

R3 = JOIN(R2, Document) /* Jointure naturelle entre les attributs Code_Manuel, Part_Num et Doc_Num */

Résultat = PROJECT(R3, Lib_Doc)

L'extension de la relation Résultat serait :

Résultat
Lib_Doc
Une résonance série

3) Quelles sont les documents cités en références croisées qui n'existent pas encore ?

```
SELECT    Ref_Manuel, Ref_Part, Ref_Doc
FROM      SectionRef
WHERE     Ref_Manuel, Ref_Part, Ref_Doc ARE NOT IN
          (SELECT    Code_Manuel, Part_Num, Doc_Num
          FROM       Document
          WHERE      Ref_Manuel = Code_Manuel
                     AND
                     Ref_Part = Part_Num
                     AND
                     Ref_Doc = Doc_Num )
```

La requête sous forme d'opérateurs relationnels :

```
R1 = PROJECT( SectionRef, Ref_Manuel, Ref_Part, Ref_Doc )

R2 = PROJECT( Document, Code_Manuel, Part_Num, Doc_Num )

R3 = DIFFERENCE( R1, R2 )  /* On retrouve dans R3 les n–uplets de R1 qui
                              n'apparaissent pas dans R2 */

Résultat = PROJECT( R3, Ref_Manuel, Ref_Part, Ref_Doc )
```

Résultat		
Ref_Manuel	Ref_Part	Ref_Doc
C1	70	00
C1	70	02

4) Quels sont les documents qui contiennent des références croisées?

```
SELECT    Code_Manuel, Part_Num,.Doc_Num, Lib_Doc
FROM      Document
WHERE     Code_Manuel, Part_Num, Doc_Num IN
          (SELECT Code_Manuel, Part_Num, Doc_Num
          FROM      SectionRef
          WHERE     Code_Manuel = Document.Code_Manuel
                    AND
                    Part_Num = Document.Part_Num
                    AND
                    Ref.Doc_Num = Document.Doc_Num)
```

Avec les opérateurs relationnels, on aurait :

R1 = PROJECT(Section, Code_Manuel, Part_Num, Doc_Num)

R2 = JOIN(R1, Document)

Résultat = PROJECT(R2, Code_Manuel, Part_Num, Doc_Num, Lib_Doc)

Résultat			
Code_Manuel	Part_Num	Doc_Num	Lib_Doc
C1	20	01	Surtensions consécutives à un amorçage en retour
C1	40	02	Catégories des contraintes de tensions
C1	60	05	Une résonance série

5) Pour chaque document, quels sont les documents cités en référence croisée ?

```
SELECT    Code_Manuel, Part_Num, Doc_Num, Lib_Doc
          Ref_Manuel, Ref_Part, Ref_Doc
FROM      SectionRef, Document
WHERE     SectionRef.Code_Manuel = Document.Code_Manuel  AND
          SectionRef.Part_Num = Document.Part_Num        AND
          SectionRef.Doc_Num = Document.Doc_Num
```

A l'aide des opérateurs relationnels, cette requête s'exprimerait :

R1 = PROJECT(SectionRef, Ref_Doc, Ref_Part, Ref_Manuel)

R2 = PROJECT(Document, Doc_Num, Part_Num, Code_Manuel)

R3 = DIFFERENCE(R1, R2)

R4 = DIFFERENCE(R1, R3)

R5 = JOIN(SectionRef, R4)

R6 = JOIN(Document, R5)

Résultat = PROJECT(R6, Code_Manuel, Part_Num, Doc_Num, Lib_Doc,
Ref_Manuel, Ref_Part, Ref_Doc)

Résultat						
Code_ Manuel	Part_ Num	Doc_ Num	Lib_Doc.	Ref_ Man.	Ref_ Part	Ref_ Doc
C1	20	01	Surtensions consécutives à un amorçage en retour	C1	50	02
C1	20	01	Surtensions consécutives à un amorçage en retour	C1	60	05
C1	20	01	Surtensions consécutives à un amorçage en retour	C1	20	01
C1	50	02	Catégories des contraintes de tensions	C1	20	01
C1	50	02	Catégories des contraintes de tensions	C1	60	05
C1	60	05	Une résonance série	C1	20	01
C1	60	05	Une résonance série	C1	50	02

6) Quels sont les documents où le document C1.20.01 est cité en référence croisée ?

```
SELECT    Code_Manuel, Part_Num, Doc_Num, Lib_Doc
FROM      Document
WHERE     Code_Manuel, Part_Num, Doc_Num IN
          (SELECT Code_Manuel, Part_Num, Doc_Num
          FROM      SectionRef
          WHERE     Ref_Manuel = 'C1'   AND
                    Ref_Part = '20'     AND
                    Ref_Doc = '01')
```

Avec les opérateurs de l'algèbre relationnelle, nous aurions :

R1 = SELECT(SectionRef, Ref_Doc = '01' ET Ref_Part = '20' et Ref_Manuel = 'C1')

R2 = PROJECT(R1, Code_Manuel, Part_Num, Doc_Num)

R3 = JOIN(R2, Document)

Résultat = PROJECT(R3, Code_Manuel, Part_Num, Doc_Num, Lib_Doc)

Résultat			
Code_Manuel	Part_Num	Doc_Num	Lib_Doc.
C1	20	01	Surtensions consécutives à un amorçage en retour
C1	50	02	Catégories des contraintes de tensions
C1	60	05	Une résonance série

5.5 Validité des parcours de lecture

La navigation associative d'un lecteur au sein d'un réseau hypertexte se traduit par la construction d'un parcours de consultation. Celui-ci se bâtit au fil des liens que le lecteur active dans les nœuds visités ou au moyen des commandes de l'interface qu'il utilise (nœud précédent, nœud suivant, backtracking, recherche de chaînes de caractères, historique, etc...). Les nœuds visités finissent alors par représenter une recherche d'information sur un thème précis. Les parcours de lecture une fois épurés, réorganisés et identifiés peuvent être mis à la disposition d'autres lecteurs.

L'actualisation des réseaux hypertextes, et principalement la suppression des nœuds dans le méta–réseau, affecte directement les parcours de lecture déclarés. En modélisant sous forme relationnelle les lecteurs et leurs parcours de lecture, nous proposons un moyen de détecter les parcours de lecteur rendus invalides par les opérations de mise à jour. La représentation relationnelle des lecteurs et de leurs chemins de consultation s'appuie sur le schéma relationnel déjà défini **[cf. 5.6.1]**.

De cette manière, il sera possible de répondre aux questions suivantes :

- Quels sont les parcours créés par les lecteurs ?
- Quels sont les parcours qui traitent de "Surtension" ?
- Quels parcours seront concernés par la suppression du document C1.20.01 ?
- Quels sont les parcours empruntant des nœuds dont le contenu a été modifié après la création de ces parcours ?
- Quels sont les nœuds les plus visités ?

- etc...

5.5.1 Schéma relationnel des parcours de lecture

Nous complétons le schéma relationnel des documents hypertextualisés au moyen des attributs suivants :

Num_Lecteur : Numéro identifiant un utilisateur (entier)

Nom : Nom de l'utilisateur (texte)

Prénom : Prénom de l'utilisateur (texte)

Service : Service auquel appartient l'utilisateur (texte)

Code_Parc : Mnémonique identifiant le parcours de l'utilisateur (texte)

Desc_Parc : Descriptif du parcours (texte)

Date_Parc : Date de création du parcours (date)

Le schéma de la relation universelle décrivant les lecteurs et leurs parcours de lecture est le suivant :

Parcours (Num_lecteur, Nom, Prénom, Service, Code_Parc, Desc_Parc, Date_Parc, Code_Manuel, Part_Num, Doc_Num, Section_Num)

Les importantes redondances qu'introduit cette relation nous conduit à la décomposer en trois relations répondant aux dépendances fonctionnelles suivantes :

(Num_Lecteur) $\longrightarrow$ Nom, Prénom, Service

(Num_Lecteur, Code_Parc) $\longrightarrow$ Desc_Parc, Date_Parc

(Num_Lecteur, Code_Parc, Code_Manuel, Part_Num, Doc_Num, Section_Num) ⟶ *nœuds empruntés par chaque lecteur*

A partir de ces dépendances fonctionnelles, nous créons les relations suivantes :

Lecteur (*Num_Lecteur*, Nom, Prénom, Service)

Parcours (*Num_Lecteur*, *Code_Parc*, Desc_Parc, Date_Parc)

NœudsDesParcours (*Num_Lecteur*, *Code_Parc*, *Code_Manuel*[1], *Part_Num*[2], *Doc_Num*[3], *Section_Num*[4])

[1] Attribut défini dans la relation Manuel

[2] Attribut défini dans la relation Partie

[3] Attribut défini dans la relation Document

[4] Attribut défini dans la relation Section

Ces trois relations traduisent le fait que plusieurs utilisateurs peuvent être déclarés dans le méta–réseau et qu'ils peuvent créer individuellement plusieurs parcours de lecture.

5.5.2 Extension des relations supplémentaires

Lecteur			
Num_Lecteur	Nom	Prénom	Service
100	Pierre	Henri	IMA
250	Paul	Marc	IPN
420	Jacques	Antoine	ERMEL
490	Paul	Philippe	IMA

Parcours			
Num_Lecteur	Code_Parc	Desc_Parc	Date_Parc
100	Influence	Recherche sur les paramètres d'influence des surtensions	28/11/92
100	Triphasé	Phénomènes de surtension sur réseau triphasé	27/05/93
420	Phénomène	Phénomène touchant à la surtension des réseaux triphasés	05/07/93
490	Surtension	Informations générales sur la surtension des réseaux	12/07/93

NœudsDesParcours					
Num_Lecteur	Code_ Parc	Code_ Manuel	Part_ Num	Doc_ Num	Section_ Num
100	Influence	C1	20	01	3
100	Influence	C1	40	02	3.1
100	Influence	C1	40	02	3
100	Triphasé	C1	40	02	1.1
100	Triphasé	C1	40	02	1.2
100	Triphasé	C1	40	02	1.3
420	Phénomène	C1	20	01	1.1
420	Phénomène	C1	20	01	1.2
420	Phénomène	C1	60	05	2
420	Phénomène	C1	60	05	3
490	Surtension	C1	20	01	1
490	Surtension	C1	20	01	2.1
490	Surtension	C1	20	01	3
490	Surtension	C1	40	02	1.1
490	Surtension	C1	40	02	2

5.5.3 Administration des parcours

La représentation relationnelle présentée ci-dessus permet non seulement de suivre la répercussion d'une suppression de nœuds sur les parcours mais aussi d'administrer plus génériquement les lecteurs et leurs parcours. Nous donnons quelques exemples de requêtes SQL d'administration de parcours :

1) Quels sont les chemins créés par l'utilisateur Pierre ?

```
SELECT    Code_Parc, Desc_Parc
FROM      Parcours
WHERE     Num_Lecteur IN
          (
          SELECT    Num_Lecteur
          FROM      Lecteur
          WHERE     Nom = 'Pierre'
          )
```

En utilisant les opérateurs relationnels, nous aurions :

R1 = JOIN(Lecteur, Parcours)

R2 = SELECT(R1, Nom = 'Pierre')

Résultat = PROJECT(R2, Code_Parc, Desc_Parc)

Résultat	
Code_Parc	Desc_Parc
Influence	Recherche sur les paramètres d'influence des surtensions
Triphasé	Phénomènes de surtension sur réseau triphasé

2) Quels utilisateurs n'ont pas créé de parcours ?

```
SELECT    Nom, Prenom
FROM      Lecteur
WHERE     Num_Lecteur IS NOT IN
          (
          SELECT    Num_Lecteur
          FROM      Parcours
          WHERE     Num_Lecteur = Lecteur.Num_Lecteur
          )
```

En utilisant les opérateurs relationnels, nous aurions :

R1 = PROJECT(Lecteur, Num_Lecteur)

R2 = PROJECT(Parcours, Num_Lecteur)

R3 = DIFFERENCE(R1, R2)

R4 = JOIN(R3, Lecteur)

Résultat = PROJECT(R4, Nom, Prenom)

Résultat	
Nom	Prenom
Paul	Marc

3) Quels utilisateurs ont créé des parcours empruntant les nœuds du document C1.20.01 ?

```
SELECT    Nom, Prenom, Code_Parc
FROM      Lecteur, Parcours
WHERE     Num_Lecteur IN
          (
          SELECT    Num_Lecteur
          FROM      Parcours
          WHERE     Num_Lecteur, Code_Parc IN
                    (
                    SELECT    Num_Lecteur, Code_Parc
                    FROM      NœudsDesParcours
                    WHERE     Code_Manuel = 'C1'
                              AND
                              Part_Num = '20'
                              AND
                              Doc_Num = '01'
                    )
          )
```

En utilisant les opérateurs relationnels, nous obtiendrons :

R1 = SELECT(NœudsDesParcours, Code_Manuel = 'C1' ET Part_Num = '20' ET Doc_Num = '01')

R2 = JOIN(Parcours, R1)

R3 = JOIN(Lecteur, R2)

Résultat = PROJECT(R3, Nom, Prenom, Code_Parc)

Résultat		
Nom	Prenom	Code_Parc
Pierre	Henri	Influence
Jacques	Antoine	Phénomène
Paul	Philippe	Surtension

4) Quels sont les titres des nœuds empruntés par le parcours Triphasé de l'utilisateur Pierre ?

```
SELECT    Lib_Section
FROM      Section
WHERE     Section_Num, Doc_Num, Part_Num, Code_Manuel IN
          (
          SELECT    Section_Num, Doc_Num, Part_Num,
                    Code_Manuel
          FROM      NœudsDesParcours
          WHERE     Num_Lecteur, Code_Parc IN
                    (
                    SELECT    Num_Lecteur
                    FROM      Lecteur
                    WHERE     Nom = 'Pierre'
                    )
                    AND
                    Code_Parc = 'Triphasé'
          )
```

En utilisant les opérateurs relationnels, nous obtiendrons :

R1 = JOIN(NœudsDesParcours, Lecteur)

R2 = SELECT(R1, Nom = 'Pierre' ET Code_Parc = 'Triphasé')

R3 = JOIN(R2, Section)

Résultat = PROJECT(R3, Lib_Section)

Résultat
Lib_Section
Tension permanente d'un réseau triphasé
Tension assignée d'un réseau triphasé
Tension la plus élevée d'un réseau triphasé U_m

5.6 Index et normalisation relationnelle

L'actualisation des réseaux hypertextes **[cf. 4.1]** repose sur la normalisation relationnelle des nœuds et des liens. Dans un premier temps, cette normalisation a conduit à la création de six relations : Manuel, Partie, Document, Section, RefCroisées et SectionRef. Dans un second temps, trois relations supplémentaires ; Lecteur, Parcours et NœudDesParcours ont été déclarées pour assurer la validité des parcours de lecture des utilisateurs. Nous proposons de compléter ce schéma relationnel par une dernière relation qui accueillera les termes associés aux différents nœuds. Cette relation devra refléter le fait qu'un nœud puisse contenir 0 ou plusieurs termes, et qu'un même terme puisse être présent dans plusieurs nœuds (lien maillé de type N:M).

La relation *Expressions* contiendra les attributs suivants :

Code_Manuel : Code du Manuel (clef étrangère de la relation Manuel)

Part_Num : Partie dans le manuel (clef étrangère de la relation Partie)

Doc_Num : Numéro du document (clef étrangère de la relation Document)

Section_Num : numérotation des sections (clef étrangère de la relation Section)

Terme : groupe nominal (chaîne de caractères)

Le quintuplet (*Code_Manuel, Part_Num, Doc_Num, Section_Num, Terme*) forme la clef primaire de la relation Expressions.

Nous avons vu que l'administration des réseaux et des parcours de lecture pouvait être menée au moyen de requêtes SQL **[cf. 4.6.3]**. En s'appuyant sur le schéma relationnel défini, nous allons voir que la navigation peut débuter de la même manière :

1) Quels sont les termes associés au document C1.20.01 ?

```
SELECT  Terme
FROM    Expressions
WHERE   Doc_Num = '01' AND
        Part_Num = '20' AND
        Code_Manuel = 'C1'
```

En utilisant les opérateurs relationnels, nous aurions :

```
R1 = SELECT(  Expressions, Code_Manuel = 'C1' ET Part_Num = '20' ET
              Doc_Num = '10' )

Résultat = PROJECT( R1, Terme )
```

2) Quels sont les documents dont les termes associés contiennent le mot "surtension" ?

```
SELECT  Code_Manuel, Part_Num, Doc_Num, Lib_Doc
FROM    Document
WHERE   Code_Manuel, Part_Num, Doc_Num IN
        ( SELECT    Code_Manuel, Part_Num, Doc_Num
          FROM      Expressions
          WHERE     Code_Manuel = Expressions.Code_Manuel
                    AND
                    Part_Num = Expressions.Part_Num
                    AND
                    Doc_Num = Expressions.Doc_Num
                    AND
                    Terme LIKE '%surtension%'
        )
```

Avec les opérateurs relationnels, nous obtenons :

```
R1 = SELECT( Expressions, Terme = '%surtension%' )

/* '%surtension%' permet de retrouver le mot surtension, quelque soit sa
     position dans les différentes valeurs que peut prendre l'attribut Terme */

R2 = JOIN( R1, Document )

Résultat = PROJECT( R2, Code_Manuel, Part_Num, Doc_Num, Lib_Doc )
```

3) Quels sont les parcours de lecture où le mot "surtension" apparaît au moins cinq fois dans l'ensemble des termes associés aux nœuds visités ?

```
SELECT  Code_Parc, Num_Lecteur
FROM    NœudsDesParcours
WHERE   Code_Manuel, Part_Num, Doc_Num, Section_Num IN
        ( SELECT    Code_Manuel, Part_Num, Doc_Num,
                    Section_Num
          FROM      Expressions
          WHERE     Code_Manuel = NœudsDesParcours.Code_Manuel
                    AND
                    Part_Num = NœudsDesParcours.Part_Num
                    AND
                    Doc_Num = NœudsDesParcours.Doc_Num
                    AND
                    Section_Num = NœudsDesParcours.Section_Num
                    AND
                    Terme LIKE '%surtension%'
        )
GROUP BY Code_Manuel, Part_Num, Doc_Num, Section_Num
HAVING COUNT( * ) >= 5
```

Les focntions statistiques (somme, moyenne, minimum, maximum, comptage) n'existant pas dans l'algèbre relationnelle, cette requête n'est que partiellement exprimable :

R1 = SELECT(Expressions, Terme = "%surtension%")

R2 = JOIN(R1, NœudsDesParcours)

R3 = *cette relation utilise la fonction SOMME pour faire apparaître les n–uplets pour lesquels la condition : fréquence du mot "surtension" >= 5 pour un même parcours*

Résultat = PROJECT(R3, Num_Lecteur, Code_Parc)

4) Quels sont les termes associés au parcours de lecture "Triphasé" ?

```
SELECT  Terme
FROM    Expressions
WHERE   Code_Manuel, Part_Num, Doc_Num, Section_Num IN
        ( SELECT    Code_Manuel, Part_Num, Doc_Num, Section_Num
          FROM      NœudsDesParcours
          WHERE     Code_Manuel = NœudsDesParcours.Code_Manuel
                    AND
                    Part_Num = NœudsDesParcours.Part_Num
                    AND
                    Doc_Num = NœudsDesParcours.Doc_Num
                    AND
                    Section_Num = NœudsDesParcours.Section_Num
                    AND
                    Code_Parc = 'Triphasé'
        )
```

Avec les opérateurs relationnels, nous obtenons :

```
R1 = SELECT( NœudsDesParcours, Code_Parc = "Triphasé" )

R2 = PROJECT( R1, Code_Manuel, Part_Num, Doc_Num, Section_Num )

R3 = JOIN( R2, Expressions )

Résultat = PROJECT( R3, Terme )
```

5.7 Conclusion

La modélisation des documents hypertextualisés que nous venons de présenter demeure encore théorique. Son implémentation dépend d'une part du choix d'un SGBD relationnel existant (Oracle, Sybase, Access, DbaseIV, Paradox, etc...) et d'autre part de la modification de la chaîne actuelle d'hypertextualisation pour permettre l'intégration du SGBD. Ces modifications concernent principalement la description formelle du contenu des entités structurelles significatives (les en–têtes et les premières pages) qui permettra d'enrichir automatiquement les tables relationnelles **[cf. A.4]**. Dans ce but, l'analyseur lexical initialement paramétré pour extraire les références croisées des documents sera réutilisé avec de nouveaux paramètres **[cf. A.5]**.

Le chapitre 2 nous a permis de constater que l'essentiel des travaux décrivant un rapprochement entre bases de données et hypertexte, visaient à renforcer la navigation dans les réseaux hypertextes au moyen de mécanismes de manipulation de données propres aux SGBD. C'est essentiellement la puissance des SGBD en matière de non–procéduralité et d'extraction de données factuelles qui était recherchée.

Dans notre démarche, nous nous sommes davantage inspirés de l'aspect purement gestion de données des SGBD (relative à l'indépendance logique/physique et à l'intégrité), que nous avons utilisé pour faciliter la maintenance du réseau.

La solution d'actualisation locale des hyperdocuments que nous venons d'expliciter repose essentiellement sur l'administration de la cohérence des nœuds et des liens. Grâce aux requêtes d'exploitation, dont nous avons vu quelques exemples, l'administrateur du méta–réseau hypertexte connaîtra à tout moment les

répercussions qu'une opération physique sur un document (insertion, modification, suppression) aura sur l'ensemble du fonds hypertextuel et sur les parcours de lecture.

A un niveau plus général, l'utilisation d'un SGBD pour gérer la cohérence des éléments des réseaux hypertextes souligne l'évidente complémentarité entre système de gestion de bases de données et système hypertexte. L'actualisation locale des hyperdocuments a montré que l'utilisation combinée d'outils à finalités différentes était envisageable à condition de respecter leur fonctionnalité originelle : la gestion des données pour les SGBD et la navigation associative pour les systèmes hypertextes.

Par ailleurs et bien que le couplage de l'hypertextualisation automatique avec un outil d'indexation automatique capable de traiter des corpus de texte portant sur des domaines quelconques, reste encore à réaliser, on peut affirmer, qu'aucun écueil technique sérieux ne devrait empêcher la réalisation du couplage tel qu'il a été décrit.

Grâce à la représentation relationnelle des termes, il est possible de construire rapidement une vue logique d'un ensemble de nœuds rattachés à un même terme et de faciliter les opérations manuelles d'insertion ou de suppression de termes, postérieures à l'indexation automatique.

Ce couplage offre aux lecteurs de s'affranchir d'une navigation exclusivement structurelle, reflet de la logique auteur, au profit d'une navigation plus libre s'appuyant sur les termes automatiquement extraits, reflets du message sémantique des nœuds. Mais avant que les lecteurs ne disposent d'une véritable efficacité dans cette forme de navigation, les effets de "bruit" et de "silence" produits par les indexeurs automatiques devront être éliminés. A ce stade, seule

une intervention manuelle des auteurs ou des lecteurs pourra conduire à une réduction du bruit et à une correction du silence. On peut imaginer qu'une intervention visant à valider la pertinence des termes automatiquement extraits, puisse être l'objet de parcours de lecture.

Chapitre 6

Expérimentation

6.1 Techniques de l'hypertextualisation

A partir des hypothèses exprimées dans le chapitre 3 concernant l'adéquation entre structure logique et structure physique dans les documents, nous décrivons la mise en œuvre fonctionnelle de l'hypertextualisation automatique et de l'actualisation des réseaux hypertextes automatiquement générés, présentées respectivement dans les chapitres 4 et 5. Cette description fonctionnelle met en évidence l'adaptabilité de la solution d'hypertextualisation dans un processus industriel de production de documents techniques.

Nous présentons les modules développés pour réaliser la granularisation des documents, la création des réseaux hypertextes, et l'actualisation des réseaux au moyen d'un SGBD relationnelles. La figure suivante permet de distinguer ces trois parties et de constater que le module de granularisation y joue un rôle déterminant.

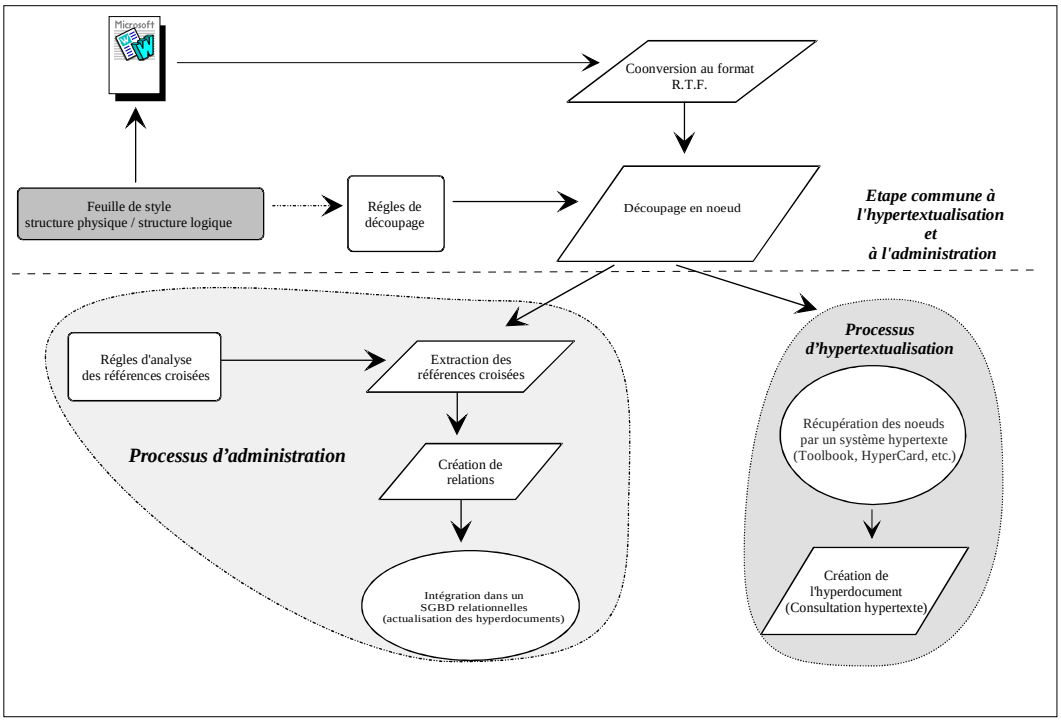

Figure 9 : La chaîne d'hypertextualisation

6.2 L'automate de découpage

La granularisation des documents est effectuée au moyen d'un "automate de découpage" qui prend en entrée les documents, préalablement enregistrés sous le format RTF, et qui, en fonction des paramétres de granularisation produira une liste de "fichiers-nœuds", destinés à la création dynamique des réseaux hypertextes. La figure suivante résume le fonctionnement de ce module.

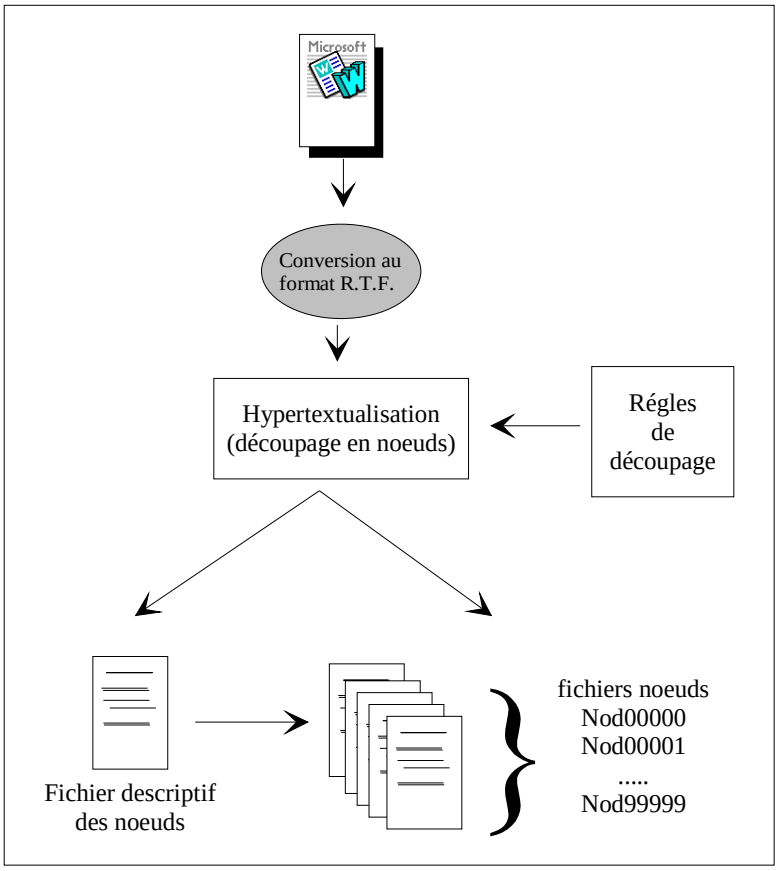

Figure 10 : Le module de découpage

Les évolutions du format RTF et les importantes différences pouvant exister dans la structure physique des documents traités, ont conduit à une forte paramétrisation de cet automate **[cf. Figure 11]** :

- pour qu'il suive, voire même prévienne, les évolutions du format RTF (nous verrons que ces évolutions rendent le

format extrêmement "instable" car elles interviennent très fréquemment au rythme des versions du logiciel de traitement de texte),

- pour qu'il s'adapte à des documents structurés différemment d'un point de vue physique (les documents techniques que nous traitons proviennent de plusieurs projets, et malgré l'apparente homogénéité des présentations, de nombreuses spécificités demeurent),

- pour qu'il permette de faire varier le degré de granularisation.

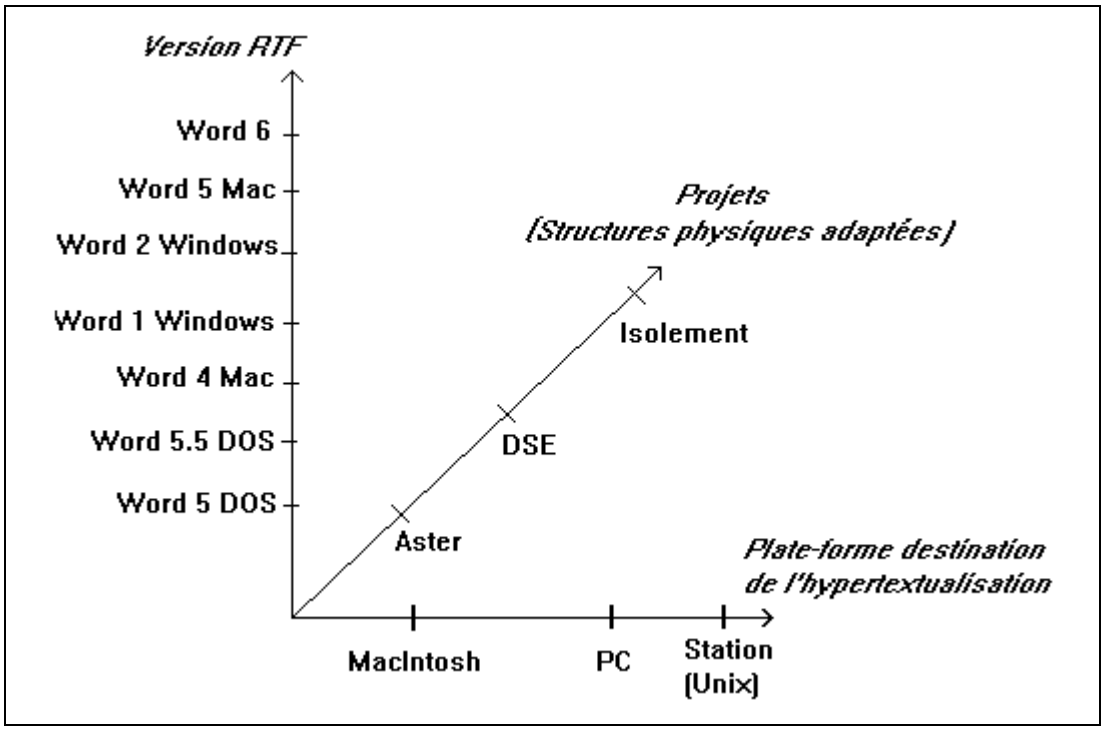

Figure 11 : L'adaptabilité à troix axes de l'automate de découpage

Etant donné que l'hypertextualisation automatique s'insére dans une dynamique industrielle de production de documents **[cf. 3.4]**, il a fallu, dès le début, s'accommoder de l'outil de rédaction retenu par les directions d'Electricité De France. Ce choix s'est porté sur le traitement de texte Word de Microsoft.

L'impossibilité d'obtenir le format natif des fichiers Word et la nécessité de s'adapter aux différentes versions du logiciel (les documents proviennent aussi bien de versions PC ; Word5, Word5.5, Word 1 et Word 2 Windows, que Macintosh, Word 4, Word 5) nous ont conduit à traiter les documents enregistrés au format RTF. Ce format[1], mis au point par la société Microsoft elle-même, a été élaborée pour faciliter les transferts de documents entre les différentes versions du même logiciel fonctionnant sous des systèmes similaires ou non. Ce format est rapidement devenu un "standard" en matière d'échange puisque de nombreux éditeurs de logiciels proposent une option d'importation/exportation au format RTF dans leurs applications de traitement de texte (FrameMaker, Ami Pro, WordPerfect, etc...).

6.2.1 RTF et les problèmes d'homogénéité

Le standard RTF n'est pas un langage de description comme peuvent l'être PostScript, SGML ou ODA. Le RTF est davantage une méthode d'encodage sans règle syntaxique rigoureuse en ce qui concerne la combinaison des mots, symboles de contrôles et groupes dont le format est constitué. Avec cette méthode, il est possible de coder :

1) le texte lui-même (ce sont principalement les caractères accentués qui subissent un codage particulier et un certain nombre de caractères de ponctuation qui ont une signification particulière en RTF),

[1] Les spécifications du format Rich Text Format ont été placées par Microsoft dans le domaine public. Elles sont décrites dans le manuel de références techniques Word de Microsoft (ed. Microsoft Press). Elles peuvent être aussi obtenues directement auprès de Microsoft Corporation Department RTF, 16011 NE 36th Way Box 97017 Rdmond, WA 98073-9717) ou bien encore par un accès ftp anonymous sur le serveur primate.wisc.edu (/répertoire /pub/RTF).

2) les propriétés typographiques et de mise en page du texte formaté (graissage, interlignage, police, corps, indentation, etc...),

3) les informations périphériques : date de création, résumé, taille du document, mots-clefs, nom et emplacement de la feuille de style, polices disponibles, etc...

4) les fonctionnalités du traitement du texte : annotations, tableaux, notes, signets, champs figés (date, page) ou calculés (publipostage, expressions arithmétiques), numérotation automatique, liens dynamiques vers d'autres applications (grapheurs, tableurs), etc...

L'exemple suivant témoigne de ce manque de rigueur dans la syntaxe du RTF. Nous avons formaté de façon semblable une séquence identique de caractères dans trois versions différentes du logiciel Word. A l'issue de l'enregistrement en un format RTF de ces séquences de caractères, nous n'avons pas obtenu trois conversions RTF identiques.

La séquence de caractères "cible" était composée des deux lignes suivantes :

2.2.1 Titre du document
Il est situé en première page ou en page de garde, appuyé à gauche sur l'aplomb 1,4 cm. Style "Titre 0". Il ne comporte pas de numérotation.

La présentation de la première ligne ("*2.2.1 Titre du document*") est :

texte aligné à gauche, retrait gauche de 1,4 cm, indentation de première ligne de -1,4 cm, caractères en gras

Pour la deuxième ligne ("*Il est situé ... pas de numérotation*") :

texte justifié, retrait à gauche 0,7 cm, indentation de première ligne -0,7cm.

Nous obtenons l'encodage RTF suivant pour chacune de ces marques de formatage :

Code RTF	**Formatage**
\fi-794	retrait de première ligne -1,4 cm
\li794	retrait à gauche de 1,4 cm
\qj	texte justifié
\b	gras
\fi397	retrait de première ligne 0,7 cm
\li397	retrait à gauche de 0,7 cm
\s*xx*	indique le numéro de style associé au texte (*xx* est un nombre identifiant de manière unique le style)
\par	mot de contrôle traduisant la fin du paragraphe (caractère de fin de ligne)
\plain	mot de contrôle réinitialisant les propriétés de formatage des caractères (valeur par défaut définie par l'application)
\pard	mot de contrôle réinitialisant les propriétés de formatage du paragraphe
\tab	indique un taquet de tabulation
\caps	indique des lettres majuscules

Nous avons obtenu, en fonction des versions de Word (Word 4 pour Macintosh, Word 5 pour DOS et Word 1 pour Windows) les conversions RTF suivantes :

Version	Séquence RTF
Word 4 pour Macintosh	...\pard\plain \s253\fi-794\li794 \b\f3\fs20 2.1.1\tab {\caps T}itre du document\par \par\plain\qj\fi397 \f3\fs20 \par \pard \s255\qj\fi397\li397 Il est situ\'8e en premi\'8fre page ou en page de garde, appuy\'8e \'88 gauche sur l'aplomb 1,4 cm. Style "Titre 0". Il ne comporte pas de num\'8erotation. \pard \qj\fi397 \par...
Word 5 pour DOS	...\pard \li794\fi-794\s10 \plain \b\f8\fs20 2.1.1\tab \plain \b\caps \f8\fs20 T\plain \b \f8\fs20 itre du document \par\pard \qj\fi397\fi397\s12 \plain \f8\fs20 Il est situ\'82 en premi\'8are page ou en page de garde, appuy\'82 \'85 gauche sur l'aplomb 1,4 cm. Style "Titre 0". Il ne comporte pas de num\'82rotation.\par\pard \qj\fi397\s12 \plain \f8\fs20 \par...
Word 1 pour Windows	...\pard\plain \s9\fi-794\li794 \b\f5\fs20 2.1.1\tab Titre du document \par \pard\plain \qj\fi397 \f5\fs20 \par \pard \s10\qj\fi397\li397 Il est situ\'e9 en premi\'e8ere page ou en page de garde, appuy\'e9 \'e0 gauche sur l'aplomb 1,4 cm. Style "Titre 0". Il ne comporte pas de num\'e9rotation. \par \pard \s10\qj\fi397 \par ...

Plusieurs différences apparaissent :

1) la codification des caractères étendus. En fonction des plates-formes d'exploitation utilisées, les codes des caractères étendus varient. On retrouve généralement 3 types de codes étendus : Macintosh, ASCII et ANSI :

Lettre	Word 4 Macinstosh	Word 5 pour DOS (ASCII)	Word Windows (ANSI)
è	\'88	\'8a	\'e8
é	\'8e	\'82	\'e9
à	\'88	\'85	\'e0

2) La numérotation des styles. Les styles, bien qu'ils dispensent des caractéristiques identiques de formatage, possédent une numérotation propre selon la version du

traitement de texte utilisée. Le tableau suivant précise ces numérotations :

Numéro du style appliqué à la	Word 4 Mac	Word 5 pour DOS	Word Windows
1ère ligne	\s253	\s10	\s9
2ème ligne	\s255	\s12	\s10

3) Le séquencement des mots de contrôle. Dans les trois cas on constate que les mots de contrôle indiquant la réinitialisation des propriétés des caractères et des paragraphes (*pard* et *plain*) n'occupent pas les mêmes places (tantôt regroupés, tantôt isolés). On constate aussi que des mots de contrôle peuvent être utilisés en fonction des versions ; c'est le cas de *caps* qui apparait dans les versions Word 4 Macintosh et Word 5-DOS. A ce propos, on remarque l'utilisation arbitraire des accolades.

Ces marques d'inhomogénéïté minimes qui affectent les propriétés de formatage des caractères et des paragraphes deviennent extrêmement importantes lorqu'elles concernent les fonctionnalités des versions de traitement de texte. La numérotation automatique, la création simplifiée de tableaux, les annotations, l'importation d'images de différents formats, les liens dynamiques avec d'autres applications, les champs (calculés ou non), les caractères spéciaux, etc... sont autant de caractéristiques qui singularisent en terme de fonctionnalités, une version du logiciel de traitement de texte d'une autre. Un document au format RTF devient alors un document "fourre-tout" où le texte saisi par l'utilisateur côtoie les attributs de formatage mais aussi une image des caractéristiques fonctionnelles du programme de traitement de texte. On ne peut nier que les spécificités inhérentes aux différentes versions de Word relèvent essentiellement des fonctionnalités. Comme celles-ci apparaissent directement dans les documents RTF, elles réduisent d'autant leur "transférabilité", jusqu'à remettre en cause la raison d'être d'une telle méthode d'encodage.

On pourrait supposer qu'il existe une "transférabilité ascendante" entre les versions de convertisseur RTF. Ce type de transférabilité sous-entend que le convertisseur RTF accompagnant la version la plus récente du logiciel de traitement de texte est le mieux adapté pour récupérer tous les documents RTF (et ce quelque soit le convertisseur à partir duquel ils ont été générés). Cette transférabilité ascendante est, dans les faits, loin d'être acquise, car même si l'on admet que la nouvelle version d'un produit améliore (fonctionnellement) ledit produit, il est moins vrai que les fonctionnalités proposées dans les anciennes versions du produit se retrouvent systématiquement dans la version la plus récente.

Illustrons notre propos sur les limites de la transférabilité par rapport :

1) aux caractéristiques des feuilles de style.
   Considérons par exemple, la version Word 5 pour DOS et la version Word 1 (ou 2) pour Windows. Plusieurs années séparent ces deux versions du même logiciel, et bien qu'elles offrent l'une et l'autre la possibilité d'associer aux documents, une feuille de style simplifiant le formatage des paragraphes, on constate que la version Word 5 pour DOS propose des possibilités de style sur les caractères, qui n'existent plus dans les versions Word 1 et 2 pour Windows. En effet, dans les versions Word pour Windows, seuls les paragraphes peuvent être formatés au moyen d'un style. Dans la version pour DOS un style peut être appliqué aux paragraphes, à l'intégralité du document (marges, numérotation des pages, vis-à-vis) mais aussi à des séquences de caractères (police, corps, graissage, italique, etc...). Dans les versions pour Windows, alors que le formatage du document est passé du module "Style" au module "Modèle de document", le formatage des caractères au moyen de styles a totalement disparu. A ce propos, on remarque que la plus récente des versions de Word (Word 6) réhabilite à nouveau ce style sur les caractères. En

cette matière, une transférabilité ascendante et descendante existe entre cette dernière version et la version Word 5 pour DOS. En revanche, une perte d'information de formatage entre des documents Word 5 pour DOS vers les versions Word 1 et Word 2 apparaît aussi inéluctable qu'entre la toute dernière version et les versions Word 1 et Word 2.

2) aux images.

Dans les spécifications techniques du RTF distribuées par Microsoft, on peut lire "*The Rich Text Format (RTF) standard is a method of encoding formatted text and graphics for easy transfer between applications...With the RTF standard, documents composed under different operating systems and with different software applications can be transferred between those operating systems and applications*". Or, on peut facilement remettre en cause la transferabilité des images et ce malgré les récentes versions de Word. Prenons la version Word 4 Macintosh qui permet d'intégrer des graphiques au sein d'un document et les versions Word pour Windows, qui autorisent elles aussi des importations de graphiques de différents formats grâce à une pléthore de filtres spécialisés (BMP, PICT, TIFF, WMF, PCX, TGA, TIFF, etc...). Bien que les versions pour Windows soient plus récentes, et que le standard RTF autorisent le codage des images, la récupération par les versions Windows, d'images contenues dans un document RTF généré par Word 4 Macintosh est impossible. Ce sont les primitives graphiques QuickDraw exploitant spécifiquement les capacités des machines de type Apple Macintosh qui rendent illusoire l'exportation de graphiques vers d'autres environnements matériels. De ce fait, on peut s'interroger sur l'intérêt d'encoder en RTF, dans l'optique d'un transfert, une image QuickDraw, dont on sait qu'elle sera, même sous son format RTF, totalement irrécupérable pour une version plus récente du même logiciel de traitement de texte mais fonctionnant dans un environnement différent (comme Word 2 pour Windows).

6.2.2 RTF et hypertextualisation

Les exemples présentés, attestent de "l'instabilité" du format RTF et rendent l'usage de ce format en matière de transfert de documents totalement rédhibitoire. Ce constat est totalement justifié si l'on considére exclusivement le RTF sous son aspect "traitement de texte". Pour l'hypertextualisation automatique, nous avons plutôt considéré le RTF via les styles sous l'aspect du formatage appliqué aux paragraphes composant le document. Il faut souligner à ce propos que, dès qu'un style est appliqué à un paragraphe, tout convertisseur RTF (quelle que soit la version du traitement de texte exploitée) traduit ce marquage de manière homogène : le mot de contrôle \s***xx*** (où ***xx*** est un chiffre identifiant de manière unique le style) précéde la séquence de caractères sur laquelle s'applique le style (et ce jusqu'au prochain *par*).

La représentation des paragraphes en RTF demeure très pauvre (seul le mot de contrôle *par* distingue un paragraphe d'un autre) de sorte qu'il est impossible de leur affecter la moindre valeur logique (en effet, on ne peut pas dire d'un paragraphe qu'il est de type titre, sous-titre, chapitre, etc.). En revanche, en leur associant des propriétés typographiques précises, on peut donner une valeur logique à un paragraphe. C'est le rôle que nous faisons jouer aux styles, qui assurent l'homogénéité de présentation des paragraphes adoptant les mêmes caractéristiques typographiques.

Par rapport à l'hypertextualisation, nous ignorons volontairement dans les documents RTF toutes les codifications relevant des fonctionnalités des logiciels de traitement de texte pour nous concentrer exclusivement sur le texte et ses caractéristiques de formatage.

Ce sont ces objectifs que vise l'automate de découpage, qui à travers une forte paramétrisation peut s'affranchir des évolutions fonctionnelles du format

RTF et proposer un découpage pertinent à partir de différents documents physiquement structurés. Par découpage pertinent, on entend un découpage conforme à ce que l'auteur lui-même aurait pu proposer en regard de l'organisation logique adoptée et mise en évidence au moyen des possibilités de formatage des logiciels de traitement de texte **[cf. 3.3]**.

6.3 Fonctionnalités de l'automate d'hypertextualisation

Le programme dont la description est donnée ci-aprés a donc pour but de préparer des documents électroniques, respectant une certaine mise en page, à devenir des hypertextes. Ce programme ne délivre en aucun cas un hypertexte directement utilisable. Il est le maillon initial indispensable d'une chaîne de traitement mettant en œuvre plusieurs opérations.

6.3.1 Programme et fichier de configuration

L'application, regroupant les fonctionnalités générales de l'hypertextualisation automatique de documents techniques se compose d'un fichier programme (APTE.EXE) et d'un fichier de configuration (APTE.CFG).

6.3.2 Conversions souhaitées

Les documents RTF préservent sous une forme encodée, l'ensemble des caractéristiques des documents générés en format natif par le traitement de texte. Pour décrire les fichiers, la méthode RTF n'utilise que les codes appartenant à la table ASCII-7 bit. Tous les caractères étendus présents dans le document au format natif du traitement de texte se retrouvent codés sur deux caractères (en héxadécimal), et cela dans le souci de respecter la limites des 128 premiers caractères de la table ASCII **[cf. 6.2.1]**.

Cette transcription de caractères étendus s'adapte à l'environnement logiciel et matériel à partir duquel le document est produit. Le format RTF peut ainsi supporter diverses plate-formes qui utilisent des codifications différentes pour des caractères étendus identiques.

Ces plate-formes, au nombre de trois, sont de type :

- ASCII (pour PC sous DOS, machines sous UNIX),
- ANSI (pour PC sous Windows),
- MAC (pour les micro-ordinateurs APPLE Macintosh).

L'automate d'hypertextualisation est à même de traiter des documents RTF provenant de ces trois plates-formes. Il pourra automatiquement à l'issue du processus d'hypertextualisation générer des nœuds d'informations directement utilisables par un de ces trois environnements sans qu'aucune autre étape de conversion ne soit nécessaire. Le type de conversion souhaité (ASCII, ANSI ou MAC) devra être simplement précisé à l'automate (celui-ci détectant automatiquement l'origine du document RTF à partir de la signature d'identification contenue dans le fichier RTF).

6.3.3 Le fichier de configuration APTE.CFG

Ce fichier de configuration permet de procéder à la transcription correcte des caractères étendus et des caractères spéciaux, de relever les balises qui détermineront le découpage physique du document initial, et de corriger l'absence de numérotation dans certains documents...

Concrétement, le fichier est composé de caractères ASCII-7 bit et comporte plusieurs sections dont les noms sont entourés par des crochets ([ ]) :

- **[ExcludedRtfGroups],**
- **[IncludedRtfGroups],**
- **[CharacterSet],**
- **[CharacterTranscription],**

- **[RtfKeyWords],**
- **[ChangeKeyWords],**
- **[SpecialCharacters],**
- **[Styles].**

On trouvera à la fin de cette annexe, un exemple de fichier de configuration utilisé avec les documents du projet "Coordination de l'Isolement" **[cf. A.2] [cf. A.7]**.

6.3.3.1 Section [ExcludedRtfGroups]

Cette section indique à l'automate d'ignorer dans le document RTF d'origine les groupes précisés. La liste ci-dessous donne de manière indicative quelques groupes RTF dont l'automate ne tiendra pas compte :

\info	/* informations administatives du document : auteur, résumé, mots-clef, etc...*/
\stylesheet	/* feuille des styles rattachée au document */
\fonttbl	/* table des polices */
\colortbl	/* table des couleurs. Sans intérêt si le périphérique d'impression est noir et blanc */
\bkmkstart	/* début de signet */
\fldinst	/* champs */
\pict	/* image */
\objdata	/* liens dynamiques vers d'autres applications */
\header	/* en-tête de page courante */
\footer	/* pied de page courante */
...	

6.3.3.2 Section [IncludedRtfGroups]

Ces groupes que l'on demande à l'automate de conserver peuvent devenir des fichiers, si l'utilisateur précise pour chacun des groupes, le nom de fichier associé. L'extraction des informations administratives relatives aux documents EDF que nous traitons (auteur, titre, date, etc...) utilise cette possibilité. Ces informations administratrives généralement placées dans l'en-tête des documents, sont associées, en RTF, aux groupes \headerf et \header. En conservant l'un ou l'autre de ces groupes dans un fichier nommé HDF ou HD, on pourra appliquer judicieusement la grammaire d'identification des documents.

Par exemple, pour associer le nom HD au groupe *header*, le nom HDF au groupe *headerf*, le nom FT au groupe *footer*, il faut procéder de la manière suivante :

```
\header=HD
\headerf=HDF
\footer=FT
\footerf=FTF
```

ll est aussi possible d'associer un nom de fichier à un groupe qui sera rencontré plusieurs fois dans le même document. Le groupe *footnote* qui précise les notes de bas de page en est un exemple flagrant. Pour distinguer et identifier de manière unique ce type de groupe, il faut rajouter à la fin du nom de fichier choisi, le caractère "*" (astérique) :

```
\footnote=NOTE*
```

Si on suppose que le document contient 5 notes de bas de page, les fichiers créés à partir du nom NOTE seront : NOTE.000, NOTE.001, NOTE.002, NOTE.003, NOTE.004.

6.3.3.3 Section [CharacterSet]

Cette section permet à l'automate de reconnaître les modes de représentation des caractères. Le format RTF reconnaît 4 types de représentation (\mac, \pc, \pca et \mac) auxquels on associe des types de représentation que l'automate peut identifier (MAC, ASCII et ANSI). L'association des types RTF aux types génériques reconnus par l'automate sont précisés de la manière suivante :

\mac=MAC
\pc=ASCII
\pca=ASCII
\ansi=ANSI

6.3.3.4 Section [CharacterTranscription]

Cette section spécifie pour chaque mode de représentation de caractères reconnu par l'automate (MAC, ASCII et ANSI définis dans la section [CharacterSet]) la liste des codes numériques (en décimal) des caractères à transcrire. La première ligne de cette section reprend l'ensemble des modes de représentation reconnus par l'automate alors que les lignes suivantes donnent les correspondances numériques des différents modes de représentation et ce, pour un même caractère. On a par exemple :

MAC,ASCII,ANSI
181,230,181
136,133,224
137,131,226
138,132,228
141,135,231
...

6.3.3.5 Section [RtfKeywords]

Cette section indique à l'automate la liste des mots de contrôle RTF à ne pas supprimer dans le document initial afin qu'ils puissent être remplacés par les codes caractères définis dans la section [ChangeKeywords].

```
\par        /* fin de paragraphe */
\tab        /* taquet de tabulation */
\row        /* ligne d'un tableau */
\cell       /* cellule d'un tableau */
...
```

6.3.3.6 Section [ChangeKeywords]

Cette section donne la liste des balises RTF qui seront remplacés par un code caractère unique (donné en décimal). Elle est directement liée aux balises précisées dans la section [RtfKeywords]. Le remplacement s'opère de la manière suivante :

```
\par=10  /* fin de paragraphe remplacé par le caractère Line Feed */
\tab=9    /* taquet de tabulation remplacé par le caractère tabulation */
\row=10
\cell=9
...
```

6.3.3.7 Section [SpecialCharacters]

Cette section décrit les codes de remplacement pour les caractères spéciaux utilisés dans les traitement de textes (espace insécable, tiret insécable, entrée d'index...).

```
\~=32       /* espace insécable remplacé par un caractère espace */
\-=45       /* césure optionnelle remplacée par un tiret */
\_=45       /* césure insécable remplacée par un tiret */
...
```

6.3.3.8 Section [Styles]

Cette section donne la liste des styles RTF qui vont permettre le découpage effectif du document en grains d'informations "autonomes". Les styles précisés dans cette section sont la traduction du convertisseur RTF aux styles définis par l'utilisateur dans le document initial au format natif du traitement de texte. C'est dans le groupe RTF *stylesheet* que la transcription des styles utilisateurs et des styles codés en RTF s'opère. Actuellement, c'est à l'utilisateur de récupérer (en éditant le document sous son format RTF) la liste des identificateurs RTF rattachés aux différents styles du document initial. L'exemple suivant donne un extrait de document RTF contenant la notification des styles définis par l'utilisateur et leur équivalent RTF :

```
    ...{\stylesheet
{\s230\fi-960\li3020\ri15\tx3020 \tqr\tldot\tx8505\tqr\tx9120
\f3\fs20 \sbasedon0\snext0 toc 4;}
{\s231 \fi-960 \li3020\ri15 \tx3020\tqr\tldot
\tx8500\tqr\tx9120 \f3\fs20 \sbasedon0\snext0 toc 3;}
{\s232\fi-880 \li2060\ri4\tx2041\tqr\tldot\tx8500\tqr\tx9120
\f3\fs20 \sbasedon0\snext0 toc 2;}
{\s233\fi-
380\li1160\ri4\sl280\tx1191\tqr\tldot\tx8500\tqr\tx9120
\f3\fs20 \sbasedon0\snext0 toc 1;}
{\s243\qj\fi397\tqc\tx3969\tqr\tx8504
\f3\fs20\sbasedon0\snext243 footer;}
{\s244\qj\brdrb\brdrdb \tx2260\tx6820\tx8020\tx8720 \i\f3\fs20
\sbasedon0\snext0 header;}
{\s252\fi-1191\li1191 \b\f3\fs20 \sbasedon222\snext0 heading
4;}
{\s253\fi-794\li794 \b\f3\fs20 \sbasedon222\snext0 heading 3;}
{\s254\fi-794\li794 \b\f3 \sbasedon222\snext0 heading 2;}
{\s255\fi-794\li794\brdrb\brdrs \b\f3\fs28 \sbasedon222\snext0
heading 1;}
{\qj\fi397 \f3\fs20 \sbasedon222\snext0 Normal;}
{\s1\qj\fi-397\li1191\tx1588 \f3\fs20 \sbasedon222\snext1
Enumer;}
{\s2\qj\f3\fs20 \sbasedon222\snext2 Americain;}
{\s4\brdrb\brdrdb \b\f3\fs36 \sbasedon2\snext4 Titre 0;}
{\s5\tx1200\tx8051\tqr\tx8780 \i\f3\fs20 \sbasedon0\snext5
Entete-texte;}
{\s6\qj\fi-1266\li2060\tx2060 \f3\fs20 \sbasedon222\snext6
Biblio;}
{\s7\qj\fi1020\f3\fs20 \sbasedon222\snext7
Postcript;}}{\colortbl...
```

On peut voir que le styles "toc 4" est traduit par \s230, "toc 3" par \s231, "footer" par \s243, "Americain" par \s2...

La section [Styles] contiendra uniquement la liste des identifiants RTF déterminant un grain d'information. L'exemple ci-dessous montre que ce sont les identifiants RTF correspondant aux styles "heading 1", "heading 2", "heading 3" et "heading 4" qui vont déterminer le granularisation du document :

\s255
\s254
\s253
\s252

6.3.3.9 La section [Styles] pour la numérotation automatique

Certains documents RTF (par exemple ceux du projet Coordination de l'isolement des lignes électriques) privilégient la numérotation automatique des sections au détriment d'une numérotation "en dur" saisie par l'auteur. Il faut s'avoir que dans le document RTF, cette numérotation n'apparaît alors que sous le forme d'une variable "système" qui permet au logiciel de traitement de texte, lors du chargement du fichier, de rebâtir dynamiquement la numérotation du document. L'utilisateur qui introduit un tel document dans l'automate en omettant l'activation de certaines options dans le fichier de configuration, obtiendra un découpage, qui bien qu'étant correct, ne fera pas apparaître clairement les niveaux hiérarchiques. Afin de réintroduire cette numérotation au sein des grains d'information générés, il faudra préciser après chacun des styles RTF dans la section [Styles] du fichier de configuration, le niveau hiérarchique du style dans le document. L'exemple suivant montre que le style \s254 correspondra au niveau 1, le style \s253 au niveau 2, etc...

\s254=1

\s253=2
\s252=3
\s251=4
\s250=5
...

6.4 Création des réseaux hypertextes consultables

C'est à partir des "fichiers-nœuds" produits par l'automate **[cf. figure 9]** que sont construits automatiquement les réseaux hypertextes dans lesquels une navigation est possible. Ces réseaux, tels qu'ils sont conçus, n'offrent pour l'instant que des formes élémentaires de navigation (table des matières motrice, consultation du nœud suivant et du nœud précédent, remontée de parcours, recherche des occurences d'une chaîne de caractères), qu'il est bien sûr possible d'améliorer. Les nombreux travaux sur la consultation des hypertextes peuvent fournir un inépuisable creuset d'idées **[cf. 1.5]**. Dans notre cas, c'est volontairement que nous nous sommes limités à ce résultat en considérant que son caractère élémentaire n'en vérifiait pas moins nos hypothèses concernant la création automatique de réseaux hypertextes à partir de documents techniques. Les quelques copies d'écran que nous proposons ont été réalisées à partir du logiciel hypertexte Toolbook d'Asymetrix, que nous avons utilisé pour construire les réseaux hypertextes dans lesquelles cette navigation élémentaire a été envisagée.

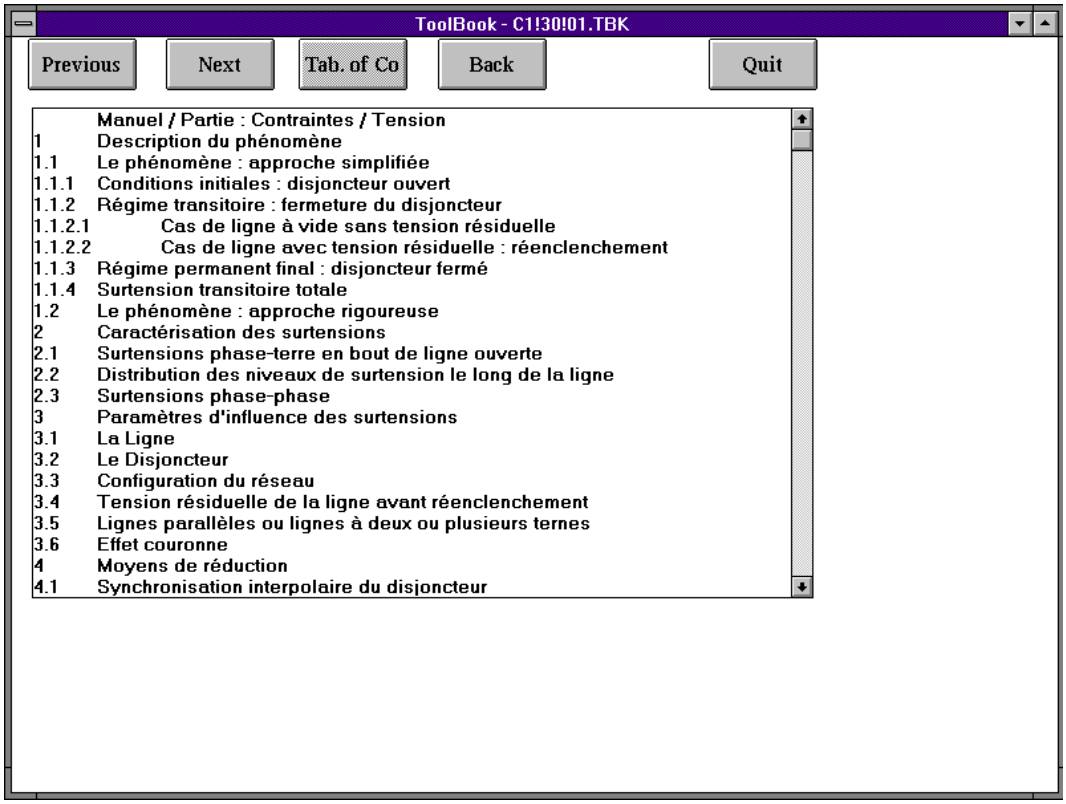

Figure 12 : Le nœud "table des matières"

Ce nœud contient les titres de tous les nœuds constituant le réseau hypertexte. A partir de cette "table des matières", il est possible d'accèder directement au contenu des nœuds en sélectionnant simplement le titre du nœud vers lequel on souhaite se rendre. Les cinq boutons situés dans la zone supérieure de la fenêtre constituent les mécanismes de navigation globale dans le réseau et sont partagés par tous les nœuds. Avec le bouton "Table of Co." il est possible de revenir en un seul saut vers le nœud "Table des matières". Le bouton "Previous" permet de se rendre dans le nœud situé avant le nœud en cours de consultation alors que le bouton "Next" permettra de visualiser le nœud situé juste après le nœud courant. Le bouton "Back" est utilisé pour remonter le parcours de lecture (backtracking). La navigation peut se réduire à la recherche d'occurences de mots. Dans la figure suivante, c'est le mot placé sur fond noir qui servira de facteur commun aux différents nœuds visités :

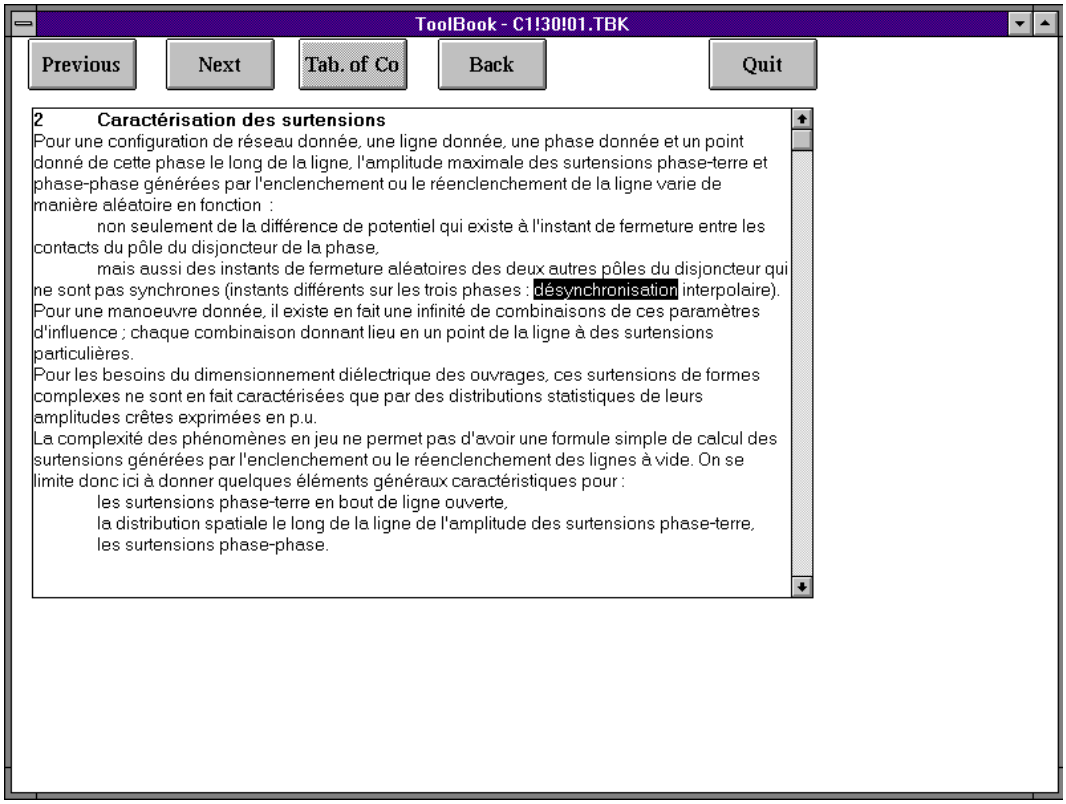

Figure 13 : la consultation des nœuds par la recherche

d'occurences de mots (ici désynchronisation).

6.5 Actualisation

Cette section décrit l'administration de réseaux hypertextes au moyen d'un système de gestion de bases de données relationnelles. Notre travail a consisté à assurer une "communication" suffisante et pertinente entre un système hypertexte et un SGBD, en leur transmettant respectivement, les informations nécessaires pour un fonctionnement optimal. Nous précisons que l'utilisation du terme "communication" qui est faite ici, concerne la faculté d'administrer au moyen d'un système de gestion de bases de données relationnelles, la structure des hyperdocuments dont la manipulation est confiée à un système hypertexte. Cette réalisation n'a pas pour prétention de proposer une solution "clef en main" de gestion électronique industrielle de documents. En effet, l'organisation de la chaîne de processus partant de la production par les auteurs de documents linéaires, à leur manipulation sous forme électronique, et intégrant la périodicité de mise à jour des documents, soulève des nécessités de gestion de projets qui ne sont pas abordées dans cette section. Notre seule ambition est de montrer que l'actualisation correspond à une réalité et qu'elle est un prolongement naturel à l'hypertextualisation.

Pour prototyper cette communication entre SGBD et système hypertexte dans le but d'administrer les réseaux hypertextes, nous avons associé au système hypertexte Toolbook, utilisé pour la consultation des réseaux hypertextes, le système de gestion de bases de données relationnelles Access de Microsoft (version 1.1).

Ce sont les références croisées, conformes à une grammaire **[cf. 4.3][cf. A.4]** et insérées par les auteurs dans les documents, qui vont permettre la navigation d'un hyperdocument à un autre. Afin de prévenir les incohérences dans

le référencement des documents, résultant des modifications apportées par les auteurs dans leurs documents, est apparue la nécessité de disposer d'une vision fédératrice de l'ensemble des documents hypertextualisés et des liens s'y rapportant.

Les sections suivantes s'attachent davantage à décrire la réalisation de cette solution.

6.5.1 Structure des hyperdocuments et représentation relationnelle

A l'issue de l'application des différentes grammaires sur les nœuds résultant de l'hypertextualisation automatique d'un document, nous possédons l'ensemble des informations indispensables au basculement vers la base de données relationnelles et au remplissage des différentes tables relationnelles définies **[cf. A.8]**. A partir de ces informations seront créées, pour chaque document traité, quatre tables relationnelles temporaires, concaténées aux relations globales définies ci-après.

6.5.2 Schéma des relations globales

Les relations DOCUMENTS, SECTIONS, RéféRENCESCROISéES et SECTIONSREFéRENCéES vont contenir respectivement :

- l'ensemble de TOUS les documents hypertextualisés à un instant donné,
- l'ensemble des sections de TOUS les documents hypertextualisés présents dans la relation DOCUMENTS,
- l'ensemble de TOUTES les références croisées présentes dans TOUTES les sections de TOUS les documents,

- les correspondances entre sections ou documents référencés par TOUTES les sections de TOUS les documents où des références croisées ont été déclarées.

Pour des extraits de ces relations **[cf. 6.7] [cf. 6.8] [cf. 6.9] et [cf. 6.10].**

Le schéma relationnel de chacune de ces relations est :

DOCUMENTS (Manuel, Partie, Document, Titre, Auteur, Date)

SECTIONS (Manuel, Partie, Document, Section, Titre)

RéféRENCESCROISéES (Manuel, Partie, Document, Section, ManuelExt, PartieExt, DocumentExt, SectionExt)

SECTIONSREFéRENCéES (Manuel, Partie, Document, Section, Reference)

6.5.3 Schéma des relations temporaires

Les relations **TxtDocument**, **TxtSection**, **TxtRéférencesCroisées** et **TxtSectionRéférencée** possèdent un schéma relationnel équivalent aux relations globales DOCUMENTS, SECTIONS, RéféRENCESCROISéES et SECTIONSREFéRENCéES. Ces relations temporaires vont servir à "intégrer" la structure de chaque document hypertextualisé (en terme de sections, de références croisées et de correspondance section de départ de la référence / point d'arrivée de la référence) aux relations globales traduisant l'ensemble de toutes les structures de tous les documents hypertextualisés. C'est un module spécialisé qui va constituer ces tables temporaires **[cf. A.9]** en recherchant à partir du résultat des analyses lexicales les différentes valeurs des champs composant les tables temporaires.

Cette étape est indispensable pour permettre la mise à jour d'un document déjà présent dans la base de données des relations globales. Cette opération débute par la suppression dans les 4 relations globales de TOUTE présence de l'ancienne version du document dont on souhaite la mise à jour. L'opération se termine par la concaténation des relations temporaires aux relations globales en conformité avec les schémas de chacune des relations :

1) TxtDocument → DOCUMENTS,
2) TxtSection → SECTIONS,
3) TxtRéférencesCroisées → RéféRENCESCROISéES,
4) TxtSectionRéférencée → SECTIONSREFéRENCéES.

6.5.4 Administration des hyperdocuments

Les requêtes présentées ici, sont toutes écrites en SQL. Elles correspondent à des opérations qui permettent par exemple d'intégrer les relations temporaires aux relations globales ou bien d'interroger les relations globales sur l'état des références croisées.

Certaines opérations se composent de plusieurs requêtes SQL qu'il convient d'exécuter séquentiellement. Pour le code SQL de toutes les requêtes donnée **[cf. 6.11]**.

6.5.5 Intégration des relations temporaires aux relations globales

Les 4 requêtes suivantes suppriment dans les relations globales toute trace du document dont une nouvelle version a été produite :

EffaceTxtDocumentDansDOCUMENTS
EffaceTxtDocumentDansRéféRENCESCROISéES
EffaceTxtDocumentDansSECTIONRéFéRENCéES
EffaceTxtDocumentDansSECTIONS

Après la suppression, l'insertion :

InsèreTxtDocumentDansDOCUMENTS
InsèreTxtSectionDansSECTIONS
InsèreTxtRéférencesCroiséesDansRéFéRENCESCROISéES
InsèreTxtSectionRéférencéeDansSECTIONSREFéRENCéES

6.5.6 Requêtes de consultation

Les requêtes proposées sont, bien sûr, loin d'être exhaustives. Elles témoignent de la possibilité de superviser la cohérence de l'ensemble des documents hypertextualisés au sein du fonds hypertexte tout au long de leurs évolutions.

6.5.6.1 Quels sont les documents où le document "xx.xx.xx" est référencé ?

La requête est :

Q Documents où le document "xx-xx-xx" est référencé ?

6.5.6.2 Quels sont les documents cités par le document "xx.xx.xx" ?

La requête est :

Q Documents référencés par le document "xx-xx-xx" ?

6.5.6.3 Le document "xx.xx.xx" référence-t-il d'autres documents ?

La requête est :

Q Références croisées dans le document "xx-xx-xx" ?

6.5.6.4 Quels sont les documents contenant des références croisées ?

La requête est :

Q Références croisées dans tous les documents

6.5.6.5 Quelles sont les sections (ou les nœuds) composant le (hyper)document "xx.xx.xx" ?

La requête est :

Q Sections du document "xx-xx-xx" ?

6.5.6.6 Quels sont les documents qui traitent de "xxxxx" dans le titre de leur document ou dans les titres de leurs sections ?

Cette requête propose une recherche d'occurences de chaine de caractère dans les attributs Titre de la relations DOCUMENTS et TitreSection de la relation SECTIONS. Lorqu'une occurence a été trouvée (indifféremment dans l'une ou l'autre relation), la clé et le titre du document sont retournés.

Q Titre de DOCUMENTS et SECTIONS traitant de"xxxxxx"

6.5.6.7 Quels sont les documents jamais cités dans les référencés croisées ?

Cette question ainsi que les deux suivantes ont été résolues au moyen de plusieurs requêtes SQL appliquées séquentiellement sous la forme d'un "pipe-line" ; les résultats d'une requête sont utilisées en entrée par la requête suivante et ainsi de suite jusqu'à obtenir le résultat final. Cette obligation de passer par

plusieurs résultats intermédiaires provient des instructions SQL indisponibles dans le requêteur SQL livré avec le SGBD relationnelles Access. En effet, le SQL d'Access n'est pas conforme à la norme ANSI SQL puisque certaines clauses (EXIST / NOT EXIST) ou opérateurs (UNION, DIFFERENCE, THETA-JOINTURE) de la norme n'apparaissent pas dans Access. Nous avons compenser l'inexistence de ces opérateurs par l'usage de l'instruction COUNT et de l'opérateur de jointure naturelle (INNER JOIN sous Access) qui nous a contraint à exprimer certaines questions en plusieurs requêtes. La question suivante met en évidence les documents de la relation DOCUMENTS (attributs Manuel, Partie et Document) qui n'apparaissent jamais dans la relation RéFéRENCESCROISéES (attributs ManuelExt, PartieExt et DocumentExt) :

QC Documents Jamais Référencés (0/3)
QC Documents Jamais Référencés (1/3)
QC Documents Jamais Référencés (2/3)
QC Documents Jamais Référencés (3/3)

6.5.6.8 Quels sont les documents qui ne citent aucun autre document ?

QC Documents Ne Contenant Pas De Références Croisées (0/3)
QC Documents Ne Contenant Pas De Références Croisées (1/3)
QC Documents Ne Contenant Pas De Références Croisées (2/3)
QC Documents Ne Contenant Pas De Références Croisées (3/3)

6.5.6.9 Quels sont les documents inexistants mais néanmoins référencés ?

Cette question résolue en 5 requêtes permet de corriger les risques d'incohérence de l'ensemble du fonds hypertexte (cette même question est résolue en une seule requête avec l'ANSI SQL : cf. p. 170).

QC Documents Référencés Mais Inexistants (0/4)
QC Documents Référencés Mais Inexistants (1/4)
QC Documents Référencés Mais Inexistants (2/4)

QC Documents Référencés Mais Inexistants (3/4)

QC Documents Référencés Mais Inexistants (4/4)

6.6 Actualisation et navigation

Dans l'actualisation, nous avons cherché à coupler la consultation associative de l'information dans les hypertextes à la gestion de la structure de l'information proposée par les bases de données. Cette gestion de la structure peut être mise à la disposition des lecteurs non dans leur but d'administration, mais pour qu'ils prennent connaissance des liens structurels entre les informations. Sans pour autant être une alternative aux représentations cartographiques souvent proposées dans les systèmes hypertextes, la connaissance de ces liens structurels aiderait le lecteur à affiner son espace d'exploration et à réduire les risques de désorientation.

On voit alors que le couplage décrit, qui tend, certes à assurer de prime abord la cohérence physique des nœuds et des liens, peut devenir, à travers l'administration des parcours de lecture, un mécanisme supplémentaire d'aide à la navigation. Dès cet instant, l'utilisation de la base de données ne serait plus de l'usage exclusif de l'administrateur pour une gestion exclusive de la structure physique des hyperdocuments, mais pourrait être étendue aux lecteurs qui pourraient s'informer des parcours de lecture ou préparer (structurellement) une future exploration (en ce sens certaines requêtes sont connotées "utilisateur" **[cf. 6.5.6.4] [cf. 6.5.6.6]**).

Il faut souligner que cette solution, quelles que soient les utilisations envisagées, reste assujettie à la bonne volonté des auteurs libres d'intégrer la dimension structurelle aux documents qu'ils rédigent et qui seront hypertextualisés. Les documents des projets Code de Mécanique Aster et Coordination de L'isolement, grâce à leur caractère technique, au mode d'organisation retenu (concernant le plan de classement) et aux consignes

dactylographiques et de mise en page adoptées ont permis de mettre en évidence cette dimension structurelle.

Plus techniquement, l'actualisation des réseaux hypertextes présentée ici gère uniquement la dernière version des documents hypertextualisés et exclut tout contrôle des versions. Les processus de découpage et d'actualisation ont été expérimentés sur PC-MS/DOS. l'automate de découpage et l'analyseur lexical tournent sous DOS alors que le SGBD relationnelles (Access de Microsoft) et le système hypertexte (Toolbook d'Asymetrix) fonctionnent l'un et l'autre sous Windows.

L'adaptation des processus de découpage et d'actualisation à d'autres environnements (Macintosh ou aux stations de travail sous Unix) est, bien sûr, envisageable. Le travail principal consistera à convertir les procédures batch d'enchaînement de tâches écrites pour MS-DOS en script shell pour UNIX (CShell, BourneShell, KornShell, ...). L'écriture de l'automate de découpage et de l'analyseur lexical en langage C a déjà permis de les porter sur station de travail. Mais au-delà du portage de ces applications, c'est jusqu'au couplage SGBD / système hypertexte qui peut être directement transposé aux deux environnements cités. Des systèmes hypertextes existent aussi bien sur Macintosh (Hypercard) que sur station (Multicard, Metacard, etc.), quant aux SGBD, tout système relationnel offrant un mode d'interrogation de type SQL peut être utilisé.

6.7 Extrait de la relation DOCUMENTS (Isolement)

Cet relation indique pour chaque document traité : le code (par exemple, C1.01.01), le titre (par exmple, Type des contraintes de tension), l'auteur et la date de création du document. Le code associé au document est composé des trois premiers attributs et identifie de manière unique un document.

C1	01	00	Types des contraintes de tension	A. SABOT	30.07.92
C1	01	01	Catégories des Contraintes de Tension	A. SABOT	11.02.93
C1	05	03	Surtensions Transitoires à Front Raide	A. SABOT	30.07.92
C1	30	01	l'Enclenchement et au Réenclenchement de ligne à vide	A. SABOT	11.02.93
C1	30	03	l'Elimination des défauts	A. SABOT	30.07.92
C1	60	01	Coup de foudre sur un conducteur de phase de ligne aérienne	A. SABOT	11.02.93
C1	60	05	Coup de foudre sur un pylône ou un câble de garde - Amorçage en retour	Alain XEMARD	10.04.93
C1	60	06	Calcul du taux d'amorçage en retour	A. SABOT	23.12.93
C4	01	00	Généralités sur l'Arrêté Technique	B. GARZILLI - R. SOLER	09.02.94
C4	01	01	Distances prescrites par l'Arrêté Technique	B. GARZILLY - R. SOLER	09.02.94
C4	01	02	Règles de voisinage imposées par l'Arrêté Technique	B. GARZILLI - R. SOLER	09.02.94
C4	02	00	Généralités sur les directives de construction	B. GARZILLI - R. SOLER	09.02.94
C4	02	01	Distances prescrites par les directives de construction	B. GARZILLI - R. SOLER	09.02.94
C1	02	00	Contraintes de tension d'origine interne	A. SABOT	30.07.92
C1	02	01	Contraintes de tension d'origine externe	A. SABOT	30.07.92
C1	05	01	Surtensions Temporaires ou Dynamiques	A. SABOT	05.10.93
C1	05	02	Surtensions Transitoires à Front Lent	A. SABOT	30.07.92
C1	05	04	Surtensions Transitoires à Front Très raide	A. SABOT	30.07.92
C1	20	01	l'Effet Ferranti	A. SABOT	30.07.92
C1	20	02	une Résonance Série	A. SABOT	30.07.92
E1	20	00	Caractérisation de l'activité orageuse	Isabelle BOIRON	22.06.93
E1	30	00	Foudroiement d'une ligne	A. SABOT	22.06.93
E1	40	00	Principes généraux	A. XEMARD	10.02.94
E1	40	01	Modèles électrogéométriques & fonction(s) de répartition des courants associée(s)	A. SABOT	10.02.94
I0	00	00	Définition des isolants électriques	A. PETIT	04.02.94
I0	01	00	Notion de champ électrique	A. PETIT	04.02.94
I0	02	00	Notion de claquage : une approche probabiliste	A. PETIT	04.02.94
I0	04	00	Paramètres d'influence sur la tenue diélectrique des isolants	A. PETIT	04.02.94
I1	06	00	Formules empiriques en choc bi-exponentiel et en 50 Hz	A. PETIT	23.11.93
I1	06	01	Les modèles de la tenue diélectrique dans l'air	A. PETIT	23.11.93

6.8 Extrait de la relation SECTIONS (Isolement)

Cette relation contient toutes les sections de tous les documents traités. Une section est représentée par un numéro de section et un titre de section. On voit par exemple que le document C1.05.03 (Surtensions transitoires à Front Raide) se compose de 11 sections.

C1	05	03	1.	Origines et causes des surtensions à front raide
C1	05	03	1.1.	Causes des surtensions à front raide d'origine externe
C1	05	03	1.1.1.	Causes des surtensions à front raide d'origine interne
C1	05	03	2.	Amplitudes maximales susceptibles d'apparaître
C1	05	03	2.1.	Amplitude maximale entre phase et terre
C1	05	03	2.1.1.	Pour les causes d'origine externe : la foudre
C1	05	03	2.1.2.	Pour les causes d'origine interne
C1	05	03	2.2.	Amplitude maximale entre phases
C1	05	03	2.2.1.	Pour la foudre
C1	05	03	2.2.2.	Pour les causes d'origine interne
C1	05	03	3.	Forme de tension représentative de cette catégorie de surtension pour les simuler lors des essais diélectriques des isolations
C1	30	01	1	Description du phénomène
C1	30	01	1.1	Le phénomène : approche simplifiée
C1	30	01	1.1.1	Conditions initiales : disjoncteur ouvert
C1	30	01	1.1.2	Régime transitoire : fermeture du disjoncteur
C1	30	01	1.1.2.1	Cas de ligne à vide sans tension résiduelle
C1	30	01	1.1.2.2	Cas de ligne avec tension résiduelle : réenclenchement
C1	30	01	1.1.3	Régime permanent final : disjoncteur fermé
C1	30	01	1.1.4	Surtension transitoire totale
C1	30	01	1.2	Le phénomène : approche rigoureuse
C1	30	01	2	Caractérisation des surtensions
C1	30	01	2.1	Surtensions phase-terre en bout de ligne ouverte
C1	30	01	2.2	Distribution des niveaux de surtension le long de la ligne
C1	30	01	2.3	Surtensions phase-phase
C1	30	01	3	Paramètres d'influence des surtensions
C1	30	01	3.1	La Ligne
C1	30	01	3.2	Le Disjoncteur
C1	30	01	3.3	Configuration du réseau
C1	30	01	3.4	Tension résiduelle de la ligne avant réenclenchement
C1	30	01	3.5	Lignes parallèles ou lignes à deux ou plusieurs ternes
C1	30	01	3.6	Effet couronne
C1	30	01	4	Moyens de réduction
C1	30	01	4.1	Synchronisation interpolaire du disjoncteur
C1	30	01	4.2	Synchronisation à la fermeture de chaque pôle du disjoncteur
C1	30	01	4.3	Réducteur de tension inductif
C1	30	01	4.4	Résistance à l'ouverture
C1	30	01	4.5	Résistance d'insertion à la fermeture du disjoncteur
C1	30	01	4.6	Parafoudres
C1	30	01	4.7	Compensation par réactance shunt

C1	30	01	4.8	Moyens de réduction les plus utilisés : état de la technique

6.9 Extrait de la relation SECTIONSRéFéRENCéES (Isolement)

Cette relation indique les références croisées présentes dans les différentes sections des différents documents. Le premier tuple de la relation nous informe que la section 1.1.1 (traitant des Causes des surtensions à front raide d'origine interne) du document C1.05.03 (Surtensions Transitoires à Front Raide) contient une référence vers l'intégralité du document C1.30.02 (la présence des trois astériques dans le dernier attribut précise que c'est le document entier qui est référencé). Le quatrième tuple nous informe que la section 3.4 (Tension résiduelle de la ligne avant réenclechement) du document C1.30.01 (l'Enclenchement et au Réenclenchement de ligne à vide) contient une référence vers la section 4.2 (Synchronisation à la fermeture de chaque pôle du disjoncteur) du même document.

C1	05	03	1.1.1.	C1	30	02	***
C1	05	03	2.2.2.	C1	30	02	***
C1	05	03	2.2.2.	C1	35	01	***
C1	30	01	3.4	C1	30	01	4.2
C1	30	01	4	C1	30	01	4.2
C1	30	01	4.1	C1	30	01	3.2
C1	30	01	4.5	C1	30	01	4.6
C1	30	01	4.5	C1	30	01	4.2
C1	30	01	4.8	C1	30	01	4.5
C1	30	01	4.8	C1	30	01	4.6
C1	30	01	4.8	C1	30	01	4.5
C1	30	01	4.8	C1	30	01	4.2
C1	30	01	4.8	C1	30	01	4.7
C1	30	01	1	C1	01	01	***
C1	30	01	1.1	C1	35	01	***
C1	30	01	1.1.1	C1	35	01	***
C1	30	01	1.1.2.2	C1	35	01	***
C1	30	01	1.1.3	C1	20	02	***
C1	30	01	1.1.3	C1	20	01	***
C1	30	01	1.1.4	C1	01	01	***
C1	30	01	2.3	I1	50	01	***
C1	30	01	3.1	C1	20	01	***
C1	30	01	3.1	C1	20	02	***
C1	30	01	3.3	C1	20	01	***
C1	30	01	3.3	C1	20	02	***

C1	30	01	3.3	C1	20	01	***

6.10 Extrait de la relation RéFéRENCESCROISéES (Isolement)

Cette relation conserve la trace exacte du libellé des références croisées extraites des documents. Le premier tuple indique que le libellé de la référence [3.4.4] porte sur la section 3.4.4 du document C1.01.01. On remarque, a travers le libellé, que le code du document n'a pas été saisi par l'auteur. Ce sont les différentes grammaires d'extraction de références croisées qui permettent de retrouver le document d'appartenance de ces références.

C1	01	01	3.4.4	[3.4.4]
C1	01	00	***	[C1.01.00]
C1	35	01	***	[C1.35.01]
C1	30	01	***	[C1.30.01]
C1	35	01	***	[C1.35.01]
C1	40	02	***	[C1.40.02]
C1	20	04	***	[C1.20.04]
C1	20	04	***	[C1.20.04]
C1	20	01	***	[C1.20.01]
C1	20	02	***	[C1.20.02]
C1	20	03	***	[C1.20.03]
C1	20	06	***	[C1.20.06]
C1	20	07	***	[C1.20.07]
C1	25	01	***	[C1.25.01]
C1	30	01	***	[C1.30.01]
C1	30	02	***	[C1.30.02]
C1	20	05	***	[C1.20.05]
C1	35	02	***	[C1.35.02]
C1	25	01	***	[C1.25.01]
C1	35	01	***	[C1.35.01]
C1	35	02	***	[C1.35.02]
C1	35	01	***	[C1.35.01]
C1	35	02	***	[C1.35.02]
C1	40	01	***	[C1.40.01]
C1	40	02	***	[C1.40.02]
C1	02	01	***	[C1.02.01]
C1	40	03	***	[C1.40.03]
C1	40	04	***	[C1.40.04]
C1	60	01	***	[C1.60.01]
C1	60	02	***	[C1.60.02]
C1	60	05	***	[C1.60.05]
C1	60	03	***	[C1.60.03]

6.11 Code SQL des requêtes

Les requêtes SQL présentées ici ont été produites par le convertisseur automatique d'Access à partir de l'interface d'interrogation (Query By Example). On remarquera le côté un peu "verbeux" des transcriptions qui auraient pu être simplifiées. Sur ce point on peut constater que des caractéristiques ANSI SQL non reconnues par SQL Access telles que la clause EXIST / NOT EXIST, les opérateurs UNION, DIFFERENCE et de théta-jointure obligent à une utilisation massive des jointures naturelles (INNER JOIN) et des fonctions statistiques de comptage (COUNT).

EffaceTxtDocumentDansDOCUMENTS

```
DELETE DISTINCTROW DOCUMENTS.* FROM TxtDocument, DOCUMENTS,
TxtDocument INNER JOIN DOCUMENTS ON TxtDocument.Manuel =
DOCUMENTS.Manuel,TxtDocument INNER JOIN DOCUMENTS ON
TxtDocument.Partie = DOCUMENTS.Partie,TxtDocument INNER JOIN DOCUMENTS
ON TxtDocument.Document = DOCUMENTS.Document;
```

EffaceTxtDocumentDansRéféRENCESCROISéES

```
DELETE DISTINCTROW RéféRENCESCROISéES.* FROM RéféRENCESCROISéES,
TxtDocument,TxtDocument INNER JOIN RéféRENCESCROISéES ON
TxtDocument.Manuel = RéféRENCESCROISéES.Manuel,TxtDocument INNER JOIN
RéféRENCESCROISéES ON TxtDocument.Partie = RéféRENCESCROISéES.Partie,
TxtDocument INNER JOIN RéféRENCESCROISéES ON TxtDocument.Document =
RéféRENCESCROISéES.Document;
```

EffaceTxtDocumentDansSECTIONRéFéRENCéES

```
DELETE DISTINCTROW SECTIONSREFéRENCéES.* FROM TxtDocument,
RéféRENCESCROISéES, SECTIONSREFéRENCéES, TxtDocument INNER JOIN
RéféRENCESCROISéES ON TxtDocument.Manuel = RéféRENCESCROISéES.Manuel,
TxtDocument INNER JOIN RéféRENCESCROISéES ON TxtDocument.Partie =
RéféRENCESCROISéES.Partie, TxtDocument INNER JOIN RéféRENCESCROISéES
ON TxtDocument.Document = RéféRENCESCROISéES.Document,
RéféRENCESCROISéES INNER JOIN SECTIONSREFéRENCéES ON
RéféRENCESCROISéES.ManuelExt = SECTIONSREFéRENCéES.Manuel,
RéféRENCESCROISéES INNER JOIN SECTIONSREFéRENCéES ON
RéféRENCESCROISéES.PartieExt = SECTIONSREFéRENCéES.Partie,
RéféRENCESCROISéES INNER JOIN SECTIONSREFéRENCéES ON
RéféRENCESCROISéES.DocumentExt = SECTIONSREFéRENCéES.Document,
RéféRENCESCROISéES INNER JOIN SECTIONSREFéRENCéES ON
RéféRENCESCROISéES.SectionExt = SECTIONSREFéRENCéES.Section;
```

EffaceTxtDocumentDansSECTIONS

DELETE DISTINCTROW SECTIONS.* FROM TxtDocument, SECTIONS, TxtDocument INNER JOIN SECTIONS ON TxtDocument.Manuel = SECTIONS.Manuel, TxtDocument INNER JOIN SECTIONS ON TxtDocument.Partie = SECTIONS.Partie, TxtDocument INNER JOIN SECTIONS ON TxtDocument.Document = SECTIONS.Document;

InsèreTxtDocumentDansDOCUMENTS

INSERT INTO DOCUMENTS SELECT DISTINCTROW TxtDocument.* FROM TxtDocument;

InsèreTxtRéférencesCroiséesDansRéFéRENCESCROISéES

INSERT INTO RéféRENCESCROISéES SELECT DISTINCTROW TxtRéférencesCroisées.* FROM TxtRéférencesCroisées;

InsèreTxtSectionDansSECTIONS

INSERT INTO SECTIONS SELECT DISTINCTROW TxtSection.* FROM TxtSection;

InsèreTxtSectionRéférencéeDansSECTIONSREFéRENCéES

INSERT INTO SECTIONSREFéRENCéES SELECT DISTINCTROW TxtSectionRéférencée.* FROM TxtSectionRéférencée;

InitialiseTxtDocument

DELETE DISTINCTROW TxtDocument.* FROM TxtDocument;

InitialiseTxtRéférencesCoisées

DELETE DISTINCTROW TxtRéférencesCroisées.* FROM TxtRéférencesCroisées;

InitialiseTxtSection

DELETE DISTINCTROW TxtSection.* FROM TxtSection;

InitialiseTxtSectionRéférencée

DELETE DISTINCTROW TxtSectionRéférencée.* FROM TxtSectionRéférencée;

Q Documents où le document "xx-xx-xx" est référencé ?

SELECT DISTINCT DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document, DOCUMENTS.Titre FROM RéféRENCESCROISéES, DOCUMENTS, RéféRENCESCROISéES INNER JOIN DOCUMENTS ON RéféRENCESCROISéES.Manuel = DOCUMENTS.Manuel, RéféRENCESCROISéES INNER JOIN DOCUMENTS ON RéféRENCESCROISéES.Partie = DOCUMENTS.Partie, RéféRENCESCROISéES INNER JOIN DOCUMENTS ON RéféRENCESCROISéES.Document = DOCUMENTS.Document WHERE ((RéféRENCESCROISéES.ManuelExt="m1") AND (RéféRENCESCROISéES.PartieExt="20") AND (RéféRENCESCROISéES.DocumentExt="06")) ORDER BY DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document, DOCUMENTS.Titre;

Q Documents référencés par le document "xx-xx-xx" ?

```
SELECT  DISTINCT RéféRENCESCROISéES.ManuelExt,
RéféRENCESCROISéES.PartieExt, RéféRENCESCROISéES.DocumentExt,
DOCUMENTS.Titre FROM RéféRENCESCROISéES, DOCUMENTS, DOCUMENTS
AS DOCUMENTS_1, RéféRENCESCROISéES INNER JOIN DOCUMENTS ON
RéféRENCESCROISéES.ManuelExt = DOCUMENTS.Manuel, RéféRENCESCROISéES
INNER JOIN DOCUMENTS ON RéféRENCESCROISéES.PartieExt =
DOCUMENTS.Partie, RéféRENCESCROISéES INNER JOIN DOCUMENTS ON
RéféRENCESCROISéES.DocumentExt = DOCUMENTS.Document, DOCUMENTS_1
INNER JOIN RéféRENCESCROISéES ON DOCUMENTS_1.Manuel =
RéféRENCESCROISéES.Manuel, DOCUMENTS_1 INNER JOIN
RéféRENCESCROISéES ON DOCUMENTS_1.Partie = RéféRENCESCROISéES.Partie,
DOCUMENTS_1 INNER JOIN RéféRENCESCROISéES ON
DOCUMENTS_1.Document = RéféRENCESCROISéES.Document WHERE
((DOCUMENTS_1.Manuel="c1") AND (DOCUMENTS_1.Partie="60") AND
(DOCUMENTS_1.Document="01")) ORDER BY RéféRENCESCROISéES.ManuelExt,
RéféRENCESCROISéES.PartieExt, RéféRENCESCROISéES.DocumentExt,
DOCUMENTS.Titre;
```

Q Références croisées dans le document "xx-xx-xx" ?

```
SELECT DISTINCT RéféRENCESCROISéES.ManuelExt,
RéféRENCESCROISéES.PartieExt, RéféRENCESCROISéES.DocumentExt FROM
RéféRENCESCROISéES, DOCUMENTS, DOCUMENTS INNER JOIN
RéféRENCESCROISéES ON DOCUMENTS.Manuel = RéféRENCESCROISéES.Manuel,
DOCUMENTS INNER JOIN RéféRENCESCROISéES ON DOCUMENTS.Partie =
RéféRENCESCROISéES.Partie, DOCUMENTS INNER JOIN RéféRENCESCROISéES
ON DOCUMENTS.Document = RéféRENCESCROISéES.Document WHERE
((DOCUMENTS.Manuel="M1") AND (DOCUMENTS.Partie="10") AND
(DOCUMENTS.Document="04")) ORDER BY RéféRENCESCROISéES.ManuelExt,
RéféRENCESCROISéES.PartieExt, RéféRENCESCROISéES.DocumentExt;
```

Q Références croisées dans tous les documents

```
SELECT DISTINCT DOCUMENTS.Manuel, DOCUMENTS.Partie,
DOCUMENTS.Document, DOCUMENTS.Titre FROM DOCUMENTS,
RéféRENCESCROISéES, DOCUMENTS INNER JOIN RéféRENCESCROISéES ON
DOCUMENTS.Manuel = RéféRENCESCROISéES.Manuel, DOCUMENTS INNER JOIN
RéféRENCESCROISéES ON DOCUMENTS.Partie = RéféRENCESCROISéES.Partie,
DOCUMENTS INNER JOIN RéféRENCESCROISéES ON DOCUMENTS.Document =
RéféRENCESCROISéES.Document ORDER BY DOCUMENTS.Manuel,
DOCUMENTS.Partie, DOCUMENTS.Document, DOCUMENTS.Titre;
```

Q Sections du document "xx-xx-xx" ?

```
SELECT DISTINCTROW DOCUMENTS.Manuel, DOCUMENTS.Partie,
DOCUMENTS.Document, SECTIONS.Section, SECTIONS.Titre FROM DOCUMENTS,
SECTIONS, DOCUMENTS INNER JOIN SECTIONS ON DOCUMENTS.Manuel =
SECTIONS.Manuel, DOCUMENTS INNER JOIN SECTIONS ON DOCUMENTS.Partie
= SECTIONS.Partie, DOCUMENTS INNER JOIN SECTIONS ON
DOCUMENTS.Document = SECTIONS.Document WHERE
((DOCUMENTS.Manuel="m1") AND (DOCUMENTS.Partie="20") AND
(DOCUMENTS.Document="06")) ORDER BY DOCUMENTS.Manuel,
DOCUMENTS.Partie, DOCUMENTS.Document, SECTIONS.Section, SECTIONS.Titre;
```

Q Titre de DOCUMENTS et SECTIONS traitant de"xxxxxx"

SELECT DISTINCTROW DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document, DOCUMENTS.Titre FROM DOCUMENTS, SECTIONS,DOCUMENTS INNER JOIN SECTIONS ON DOCUMENTS.Manuel = SECTIONS.Manuel, DOCUMENTS INNER JOIN SECTIONS ON DOCUMENTS.Partie = SECTIONS.Partie, DOCUMENTS INNER JOIN SECTIONS ON DOCUMENTS.Document = SECTIONS.Document WHERE ((DOCUMENTS.Titre Like "*tension*")) OR ((SECTIONS.Titre Like "*tension*")) ORDER BY DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document, DOCUMENTS.Titre;

Q Tri de DOCUMENTS par Manuel, Partie, Document

SELECT DISTINCTROW DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document, DOCUMENTS.Titre FROM DOCUMENTS ORDER BY DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document, DOCUMENTS.Titre;

Q Tri de SECTIONS par Manuel, Partie, Document, Section, Titre

SELECT DISTINCTROW DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document, SECTIONS.Section, SECTIONS.Titre FROM DOCUMENTS, SECTIONS, DOCUMENTS INNER JOIN SECTIONS ON DOCUMENTS.Manuel = SECTIONS.Manuel, DOCUMENTS INNER JOIN SECTIONS ON DOCUMENTS.Partie = SECTIONS.Partie, DOCUMENTS INNER JOIN SECTIONS ON DOCUMENTS.Document = SECTIONS.Document ORDER BY DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document, SECTIONS.Section, SECTIONS.Titre;

QC Documents Jamais Référencés (0/3)

DELETE DISTINCTROW TxtDocument.* FROM TxtDocument;

QC Documents Jamais Référencés (1/3)

INSERT INTO TxtDocument (Manuel, Partie, Document) SELECT DISTINCT RéféRENCESCROISéES.ManuelExt, RéféRENCESCROISéES.PartieExt, RéféRENCESCROISéES.DocumentExt FROM RéféRENCESCROISéES;

QC Documents Jamais Référencés (2/3)

INSERT INTO TxtDocument (Manuel, Partie, Document)SELECT DISTINCT DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document FROM DOCUMENTS ORDER BY DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document;

QC Documents Jamais Référencés (3/3)

SELECT DISTINCTROW TxtDocument.Manuel, TxtDocument.Partie, TxtDocument.Document, DOCUMENTS.Titre FROM DOCUMENTS, TxtDocument, TxtDocument INNER JOIN DOCUMENTS ON TxtDocument.Manuel = DOCUMENTS.Manuel, TxtDocument INNER JOIN DOCUMENTS ON TxtDocument.Partie = DOCUMENTS.Partie, TxtDocument INNER JOIN DOCUMENTS ON TxtDocument.Document = DOCUMENTS.Document GROUP BY TxtDocument.Manuel, TxtDocument.Partie, TxtDocument.Document, DOCUMENTS.Titre HAVING (((Count(TxtDocument.Document))=1)) ORDER BY TxtDocument.Manuel, TxtDocument.Partie, TxtDocument.Document, DOCUMENTS.Titre;

QC Documents Ne Contenant Pas De Références Croisées (0/3)

DELETE DISTINCTROW TxtDocument.* FROM TxtDocument;

QC Documents Ne Contenant Pas De Références Croisées (1/3)

INSERT INTO TxtDocument SELECT DISTINCT DOCUMENTS.* FROM RéféRENCESCROISéES, DOCUMENTS, DOCUMENTS INNER JOIN RéféRENCESCROISéES ON DOCUMENTS.Manuel = RéféRENCESCROISéES.Manuel, DOCUMENTS INNER JOIN RéféRENCESCROISéES ON DOCUMENTS.Partie = RéféRENCESCROISéES.Partie, DOCUMENTS INNER JOIN RéféRENCESCROISéES ON DOCUMENTS.Document = RéféRENCESCROISéES.Document;

QC Documents Ne Contenant Pas De Références Croisées (2/3)

INSERT INTO TxtDocument SELECT DISTINCT DOCUMENTS.* FROM DOCUMENTS;

QC Documents Ne Contenant Pas De Références Croisées (3/3)

SELECT DISTINCTROW TxtDocument.Manuel, TxtDocument.Partie, TxtDocument.Document, TxtDocument.Titre FROM TxtDocument GROUP BY TxtDocument.Manuel, TxtDocument.Partie, TxtDocument.Document, TxtDocument.Titre HAVING (((Count(TxtDocument.Titre))=1)) ORDER BY TxtDocument.Manuel, TxtDocument.Partie, TxtDocument.Document, TxtDocument.Titre;

QC Documents Référencés Mais Inexistants (0/4)

DELETE DISTINCTROW TxtDocument.* FROM TxtDocument;

QC Documents Référencés Mais Inexistants (1/4)

INSERT INTO TxtDocument (Manuel, Partie, Document) SELECT DISTINCT RéféRENCESCROISéES.ManuelExt, RéféRENCESCROISéES.PartieExt, RéféRENCESCROISéES.DocumentExt FROM RéféRENCESCROISéES ORDER BY RéféRENCESCROISéES.ManuelExt, RéféRENCESCROISéES.PartieExt, RéféRENCESCROISéES.DocumentExt;

QC Documents Référencés Mais Inexistants (2/4)

INSERT INTO TxtDocument (Manuel, Partie, Document) SELECT DISTINCTROW DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document FROM DOCUMENTS ORDER BY DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document;

QC Documents Référencés Mais Inexistants (3/4)

INSERT INTO TxtDocument (Manuel, Partie, Document) SELECT DISTINCT DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document FROM DOCUMENTS ORDER BY DOCUMENTS.Manuel, DOCUMENTS.Partie, DOCUMENTS.Document;

QC Documents Référencés Mais Inexistants (4/4)

SELECT DISTINCTROW TxtDocument.Manuel, TxtDocument.Partie, TxtDocument.Document FROM TxtDocument GROUP BY TxtDocument.Manuel, TxtDocument.Partie, TxtDocument.Document HAVING (((Count(TxtDocument.Document))=1)) ORDER BY TxtDocument.Manuel, TxtDocument.Partie, TxtDocument.Document;

Conclusion

La problématique sous-jacente à l'hypertextualisation est d'évaluer la possibilité de faire passer l'hypertexte à une dimension résolument industrielle. Les nombreux systèmes développées attestant de la réalité d'un tel mode de représentation et de consultation de l'information, ne lui ont pas permis de quitter ce stade "pré-industriel" où les difficultés inhérentes à la carence méthodologique en matière de conception l'ont maintenu.

L'hypertextualisation sans pour autant être une réponse à cette carence méthodologique, offre néanmoins à l'hypertexte, pour des documents de nature essentiellement technique, la possibilité de passer à une dimension plus opérationnelle. L'information technique, présentée traditionnellement sous la forme de documents rédigés linéairement, posséde cette particularité d'être composée de zones distinctes relativement autonomes; dont la consultation est envisageable en dehors de toute séquentialité de lecture, dispensatrice de sens. Nous avons vu que la structure physique des documents, permettaient potentiellement, de repérer ces différentes zones, de les extraire et de créer dynamiquement les noeuds des futurs réseaux hypertextes.

Il est évident que les auteurs ont un rôle déterminant à jouer dans l'efficacité du processus d'hypertextualisation. En effet, eux seuls, pourront garantir la pertinence de la granularisation et les potentialités hypertextuelles de leurs documents. Alors qu'il est difficile de se passer de leur collaboration pour

disposer d'unités d'informations autonomes, nous avons vu que l'indexation automatique peut pallier à la pauvreté ou à l'inexistence de références croisées, indispensables à une navigation minimale dans les réseaux hypertextes automatiquement générés.

Par rapport à ce dernier point, alors même que nous avions montré que l'hypertexte était un mode de représentation et de consultation de l'information distinct des domaines traditionnels de l'informatique que sont l'algorithmique, les bases de données et l'Intelligence Artificielle, l'hypertextualisation automatique en revanche, montre qu'il est possible de faire contribuer ces différents domaines à l'hypertexte afin d'en faciliter la création, la gestion et la consultation :

- la création. L'hypertextualisation est d'abord un processus algorithmique qui rassemble un ensemble d'actions élementaires systématiques conduisant à l'extraction des grains d'information et à la reconnaissance des références croisées.

- la gestion. La représentation relationnelle des noeuds et des liens permet d'envisager la mise à jour locale de l'ensemble du fonds hypertextuel.

- la consultation. L'Intelligence Artificielle au travers du Traitement du Langage Naturel permet d'imaginer l'indexation automatique de chaque noeud afin d'offrir aux lecteurs une possibilité de navigation sémantique.

La production automatique de réseaux hypertextes est, comme on peut le constater, un processus complexe, qui n'est pas réductible à un simple découpage comme il a été décrit dans les quelques expériences que nous avons présentées. On comprend que, par rapport à cette vision réductrice, l'hypertextualisation automatique puisse être perçue comme un phénomène passager, voué à disparaître

au profit d’un savoir–faire qui devrait finir par se généraliser dans l'attente d'une hypothétique méthodologie de conception.

Notre processus d'hypertextualisation soulève le problème majeur de la qualité de la granularisation. Toute l'efficacité du processus repose, en effet, sur le découpage cohérent des documents à hypertextualiser. Et on voit mal, dans l'état actuel des choses, comment l'évaluation de cette cohérence pourrait être effectuée autrement que par une intervention manuelle.

Le passage à une phase d'industrialisation permettra certainement de le vérifier, mais on peut supposer, dès à présent, que la dynamicité du processus d'hypertextualisation conduira les auteurs à intervenir rapidement sur les documents dont le découpage n'aura pas été satisfaisant. Progressivement, on devrait s'acheminer vers une rédaction de documents linéaires totalement adaptée à une hypertextualisation automatique.

Références Bibliographiques

[Afnor 91] **Afnor**, SGML-ODA : Présentation des concepts et comparaison fonctionnelle, *Danish Standards Association*, collection AFNOR Technique, 1991

[Agosti 89] **Agosti M.**, Is hypertext a new model of information retrieval? Dipartimento di Elettronica e Informatica, Via Gradenigo, 6/A, 35131 Padova, Italy, *89H383033*, 6 pages

[Akscyn 88] **Akscyn R.M., McCracken D.L., Yoder E.A.**, KMS: A distributed hypermedia system for managing knowledge in organizations, *Comm. of the ACM*, Vol. 31 (7), July 1988, pp. 820–835

[Ambriola 90] **Ambriola V., Bendix L., Ciancarini P.**, The evolution of configuration management and version control, *Software Engineering Journal*, November 90, pp. 303–310

[Anderson 90] **Anderson C.J., Veljkov M.D.**, Creating Interactive Multimedia - A practical Guide, *ed. Scott, Foresman and Company*, Macintosh Computer Books, 1990

[Appelt 88] **Appelt W., Carr R., Richter G.**, The formal specification of the document structures of the ODA language, *Document Manipulation and Typography* (Proc. EP'88 Conference), Cambridge University Press, 1988, pp. 95–108

[Appelt 93a] **Appelt W.,** The ODA standard 1993 and future extensions, *Computer Standards and Interfaces*, N°4, Vol. 15, 1993, pp. 343–351

[Appelt 93b] **Appelt W., Tetteh Lartey N.,** The Formal Specification of the ISO Open Document Architecture (ODA) Standard, *Computer Journal*, n°3, Vol. 36, 1993, pp. 269–279

[Astolfi 93] **Astolfi J-P.**, Comment les élèves apprennent-ils ?, *Sciences Humaines*, n°32, Octobre 93, pp. 26–29

[Balpe 90] **Balpe J-P.**, Hyperdocuments, hypertextes, hypermédia, *Ed. Eyrolles*, 1990

[Balpe 93] **Balpe J.-P.**, Traitements linguistiques des données textuelles, *Colloque sur l'analyse des données textuelles*, Montpellier, 21-22 Octobre 93

[Barron 89] **Barron D.W.**, Why use SGML? *Electronic Publishing - Origination, Dissemination and Design*, 2,1, April 89, pp. 3–24

[BasisPlus 90] **Basis Plus**, Text Information Management System - Capabilities Overview, Information Dimensions, May 1990

[Bassano 86] **Bassano J.-C**., DIALECT, un système expert pour la recherche documentaire, *Agence de l'informatique*, vol. 2, 1986, pp. 1327-1352

[Beeri 90] **Beeri C., Kornatzky Y.**, A logical Query Language for Hypertext Systems, *Proc. of the European Conference on Hypertext*, INRIA, France, November 1990, pp. 67–80

[Bernes–Lee 93] **Bernes-Lee T., Conolly D.**, Hypertext Markup Language, *Internet Draft*, July 1993

[Biennier 89] **Biennier F., Pinon P.M., Guivarch M.**, système connexioniste d'aide à la navigation dans les hyperdocuments, *NeuroNîmes*, 1989

[Biennier 90] **Biennier F., Pinon P.M., Guivarch M.**, A connectionist method to retrieve information in hyperdocuments, *International Neural Network Conference*, Paris, Juillet 90

[Boin 92] **Boin M., Levesque J-R.,** Mise en page et consignes dactylographiques de la documentation, *Manuel de Descriptif Informatique, Fascicule Présentation de la Documentation*, (Document D8.01.01), EDF/DER/IMA/MMN, Janvier 92

[Bonnet 89] **Bonnet A.**, Le traitement de la langue naturelle : outils et applications, *L'echo des recherches*, n°138, 4ème trimestre 92, pp. 33–40

[Bonnet 92] **Bonnet A.**, Le traitement du langage en recherche documentaire, *L'echo des recherches,* n°148, 2ème trimestre 92, pp. 61–66

[Bourigault 92] **Bourigault D.**, LEXTER : un Logiciel d'EXtraction de TERminologie, *2ème colloque international de TermNet*, Avignon, Juin 92

[Bourigault 94] **Bourigault D.**, LEXTER : un Logiciel d'EXtraction de TERminologie. Application à l'acquisition des connaissance, *Thèse de doctorat*, EHESS, Juin 94.

[Bradley 92] **Bradley N.**, SGML Concepts, *Aslib Proceedings*, vol. 44 (7/8), July/August 1992, pp. 271–274

[Brown 88a] **Brown P.J**., Hypertext : the way forward, ed. J.C. Van Vliet, *Document manipulation and typography, Proceedings of the International Conference on Electronic Publishing*, Nice (France), Avril 1988, pp.183–191

[Brown 88b] **Brown P.J.**, Linking and searching within hypertext, *Electronic Publishing*, Vol. 1 (1), April 88, pp. 45–53

[Brown 89] **Brown P.J.**, Do we need maps to navigate round hypertext documents? *Electronic Publishing*, Vol. 2 (2), July 89, pp. 91–100

[Camacho 91] **Camacho M., Gauroy P., Kessaci A.**, Hypertextes et Hypermédias, *L'écho des recherches*, n° 146, 4ème trimestre 1991, pp. 61–70

[Campbell 88] **Campbell B.**, **Goodman J.G.**, HAM : a general purpose hypertext abstract machine, *Comm. of the ACM*, vol 31 (7), July 1988, pp. 856–861

[Cavazza 92] **Cavazza M., Zweigenbaum P.**, Compréhension automatique du langage naturel par construction de modèles, *T.S.I*, Vol. 11, n°4, 1992, pp. 119–138

[Chartron 89] **Chartron G.**, **Dalbin S.**, **Monteil M.G.**, **Verillon M.**, Indexation Manuelle et Indexation Automatique - Dépasser les oppositions, *Le Documentaliste*, n° 4-5, Vol. 26, 1989/07, pp. 181–187

[Chaumier 90] **Chaumier J., Dejean M.**, L'indexation documentaire: de l'analyse conceptuelle humaine à l'analyse automatique morphosyntaxique, *Le Documentaliste*, n°6, Vol. 27, 1990/11, pp. 275–279

[Cheiney 89] **Cheiney J.–P.**, Bases de données : le rôle majeur des chercheurs, *Le monde informatique*, 6 Novembre 1989, pp. 65–67

[Chen 76] **Chen P. P-S.**, The Entity-Relationship Model - towards a unified view of data, *ACM TODS*, Vol. 1 (1), 1976, pp. 9–36

[CISI 82] **CISI,** Système SPIRIT - Système Syntaxique et Probabiliste d'Indexation et de Recherche d'Information Textuelles, 1982, 40 pages.

[Clifton 88] **Clifton C.**, **Garcia-Molina H.**, **Hagmann R.**, The design of a document Database, *Proc. ACM Conference on Document Processing systems*, Dec 5-9 1988, Santa Fe, New Mexico, pp. 125–134

[Codd 70] **Codd E. F.**, A Relational Model of Data for Large Shared Data Banks, *Comm. of the ACM*, Vol. 13 (6), 1970, pp. 377–387

[Codd 79] **Codd E. F.**, Extending the Database Relational Model to capture more meaning, *ACM TODS*, Vol. 4, 1979, pp 397–434

[Conklin 87] **Conklin J.**, Hypertext : an introduction and survey, *Computer*, September 1987, pp. 17–41

[Covannon 93] **Covannon E.**, SGML and Hypermedia, <TAG>, The SGML Newsletter, Vol.6, N.5, May 1993, pp. 1–4

[Cybulski 92] **Cybulski J.L.**, **Reed K.**, A hypertext based software engineering environment, *IEEE Software*, March 1992, pp. 62–68

[Dachelet 90] **Dachelet R.**, Hypertexte et hypermédia: Documents - Informations - Connaissances, *Le document Electonique*, Cours INRIA dirigé par Christian BORNES, 11 - 15 Juin 1990, pp. 135–161

[Daniel 90] **Daniel-Vatonne M-C.**, Hypertextes : des principes communs et des variations, *TSI, V*ol. 9 (6), 1990, pp. 475–492

[David 90] **David S., Plante P.**, Termino, *Rapport de recherche*, groupe Recherche et Développement en Linguistique Computationnelle, Université du Québec à Montréal, Novembre 1990

[Dechamp 88] **Dechamp F., Narat V.**, Régles de correspondances entre une spécification BNF et sa représentation par un modèle Entité-Association, *EDF/DER/IMA/TIEM*, HI-27/6248, Octobre 1988

[Derrien 89] **Derrien D., Bouchitté V., Habib M.**, Approche objet pour l'analyse de la structure logique des documents, *BIGRE 63–64* - Mai 1989, pp. 226–235

[Ehrlich 93] **Ehrlich M.-F., Tardieu H., Cavazza M.**, Les modèles mentaux - approche cognitive des représentations, *éd. Masson*, Janvier 93

[**Ermine 92]** **Ermine J.L.**, Un système à base de connaissance pour le diagnostic technique et la recherche documentaire, *Travail et Méthodes*, n 494, 1992, pp. 50-55.

[Faloutos 85] **Faloutos C.**, Access Methods for Text, *Computing Surveys*, Vol. 17 (1), March 1985

[Faribault 84] **Faribault M.**, **Leon J.**, **Meissonnier V.**, **Zarri G.P.**, **Ornato M.**, Analyse du langage naturel en vue de la construction de bases de données factuelles, *Brises, n°4*, 1984/04, pp. 11–16

[Fluhr 77] **Fluhr C.,** Algorithmes à apprentisage et traitement automatique des langues, *Thèse de doctorat*, Université Paris XI, Juin 1977

[Fluhr 84] **Fluhr C.,** Le traitement et l'interrogation des bases de données textuelles, *Informatique et Droit en Europe*, Université Libre de Bruxelles, eds. Bruylant, 1984, pp. 97-114

[Fluhr 89] **Fluhr C., Debili F., Radasoa P.,** About reformulation in full-text IRS**,** *Information Processing and Management*, Vol. 25, N°6, 1989, pp. 647-657

[Fluhr 90] **Fluhr C.,** Dynamic hypertext links, *International Symposium on Trends of Intelligent Hypermedia*, JIPDEC-CID, Tokyo, 25-26/10/1990, pp. 81-87.

[Foss 89] **Foss C.L.**, Detecting lost Users: Empirical studies on browsing hypertext, *89H385299*

[Frisse 88a] **Frisse M.**, Searching for information in a hypertext medical handbook, *Comm. of the ACM*, Vol. 31 (7), July 1988, pp. 880–886

[Frisse 88b] **Frisse M.**, From text to hypertext, *BYTE*, October 1988, pp. 247–253

[Furuta 89a] **Furuta R.**, An object-based taxonomy for abstract structure in document models, *The Computer Journal*, Vol. 32, (6), 1989, pp. 494–504

[Furuta 89b] **Furuta R.**, Concepts and models for structured documents, (eds.) Jacques André, Richard Furuta, and Vincent Quint, *Structured Documents,* Cambridge University Press 1989, pp. 7–38

[Furuta 89c] **Furuta R., Plaisant C., Shneiderman B.**, A spectrum of automatic hypertext constructions, *Hypermedia,* 1,2 (1989), pp. 179–195

[Furuta 89d] **Furuta R., Stots P.D**, Object Structures in Paper Documents and Hypertexts, *Woodman'89*, pp. 147–151

[Garg 88a] **Garg P.K.**, Abstraction mechanisms in hypertext, *Comm. of the ACM,* Vol. 31 (7), July 1988, pp. 862–870

[Garg 88b] **Garg P.K.**, **Scacchi W**., Composition of hypertext nodes, *Learned Information,* Vol. 1, 1988, pp. 63–73

[Gay 89] **Gay A., Le Ponner E.**, Introduction de concepts Hypertext dans un environnement structurée, *Bigre 63 - 64*, Mai 1989, pp. 363–366

[Gilloux 89] **Gilloux M., Lassalle E.**, **Prigent G,** Traitement du langage naturel dans les applications d'annuaire, *L'écho des recherches*, n°138, 4ème trimestre 89, pp. 41–50

[Gilloux 91] **Gilloux M., Lassalle E.**, **Ombrouck J.-M.**, Interrogation en langage naturel du Minitel Guide des Service (MGS), *L'écho des recherches*, n°146, 4ème trimestre 91, pp. 51–60

[Giordano 92] **Giordano G., Piron C., Sanson J-L.**, Application du concept de livre électronique à la consultation des ARD-AID, *EDF/DER/IPN/SID/ISI, N°92H4603R,* 13 Octobre 1992, 42 pages

[Glushko 88] **Glushko R.J.**, **Weaver M. D.**, **Coonan T.A.**, **Lincoln J.E.**, Hypertext Engineering: Practical Methods for Creating A compact Disc Encyclopedia, *Proc. ACM Conference on Document Processing systems,* Dec 5-9 1988, Santa Fe, New Mexico, pp. 11–19

[Goldfarb 91] **Goldfarb C.**, The SGML Handbook, O*xford University Press,* 1991

[Halasz 88] **Halasz F.G.**, Relections on Notecards: seven issues for the next generation of hypermedia systems, *Comm. of the ACM,* Vol. 31 (7), July 1988, pp. 836–852

[Hansen 88] **Hansen W. J.**, **Haas C.**, Reading and writing with computers: A framework for explaining differences in performance, *Comm. of the ACM*, Vol. 31 (9), July 1988, pp. 1080–1089

[Hara 91] **Hara Y**., **Keller A.**, **Rathmann P**., **Wiederhold G.**, Implementing hypertext database relationship throught aggregations and exceptions, *Report N° STAN-CS-91-1381,* Department of Computer Science, Stanford University, Stanford, California 94305, Sept. 1991

[Harmon 89] **Harmon J.E.**, The structure of Scientific and Engineering Papers : A historical perspective, *IEEE Transactions on professional communication*, Vol. 32, N°3, Sept. 89, pp. 132–138

[Heather 89] **Heather M.A.**, **Rossiter B.N**., Theoretical Structures for Object-based Text, *WoodMan'89, Bigre 63-64,* mai 1989, pp. 178–192

[Heaulme 90] **De Heaulme M.**, **Membrado M.**, Recherche documentaire - Nécessité de disposer d'un système conceptuel pour généraliser une recherche documentaire intelligente, *CIMAB - Encyclopédie*, Juillet/Août 1990

[Houbart 87] **Houbart G.**, Un mariage de raison : Bases de données et CAO, *Micro–Systèmes*, Octobre 1987, pp. 107–115

[Howgate 87] **Howgate P.V.**, SGML - Standard for document encoding and interchange, *SGML User's Group*, 1987

[Ingold 87] **Ingold R.**, Reconnaissance de structures de textes en lecture optique, *Structure de documents*, Bigre n°53, Mai 1987, pp. 68–76

[Ingold 90] **Ingold R.**, Reconnaissance de la structure logique d'un document par une méthode d'analyse descendante, *Reconnaissance automatique de l'écrit*, Bigre n°68, Mai 1990, pp. 26–34

[ISO 58] Références bibliographiques. Eléments essentiels, 1ère édition, *ISO R77,* 1958

[ISO 78] Documentation - Numérotation des divisions et subdivisions dans les documents écrits, *ISO 2145*,1978

[ISO 82] Documentation - Présentation des documents scientifiques et techniques, ISO *5966,* 1982

[ISO 85] Documentation - Méthodes pour l'analyse des documents, la détermination de leur contenu et la sélection des termes d'indexation, *ISO 5963,* 1985

[ISO 86a] Documentation - Présentation des thèses et documents assimilés, *ISO 7144,* 1986

[ISO 86b] Text and Office Systems - Standard Generalized Markup Language, *ISO 8879,* 1986

[ISO 89] Text and Office Systems - Open Document Architecture, *ISO 8613,* 1986

[ISO 92] Information Technology - Hypermedia / Time Base Structring Language (Hytime), *ISO 10744*, 1992

[Jones 86] **Jones P.J., Bell C.L.M.**, MORPHS - an intelligent retrieval system, *Aslib Proceedings*, 38, (3), March 1986, pp. 71-79.

[Kesselman 88] **Kesselman M.**, **Trapasso L.**, Hypertext and the End-User, *Learned Information*, Vol. 1, 1988, pp. 219–225

[Lucarella 90] **Lucarella D.**, A Model for Hypertex-Based Information Retrieval, *Proc. of the European Conference on Hypertext*, INRIA, France, November 1990, pp. 81–93

[Marchionini 88] **Marchionini G.**, **Schneiderman B.**, Finding Facts vs. Browsing knowledge in hypertext systems, *Computer*, January 1988, pp. 70–79

[Mc Knight 88] **Mc Knight C.**, **Richardson I.**, Hypertext: some human factors lessons to be learned from early commercial packages, *HUSAT Research Center*, Department of Human Sciences, Loughborough University of Technology, Loughborough, Great-Britain, 1988

[Mc Knight 89] **Mc Knight C.**, **Dillon A.**, **Richardson J.**, Problems in hyperland? A human factors perspective, *Hypermedia*, 1,2, (1989), pp. 167–178

[Mc Leod 88] **Mc Leod I.A.**, Hypertext as a document retrieval application, *Departement of Computing and Information Science*, Queen's University, Kingston, Ontario K7L3N6

[Menon 88] **Menon B.**, Indexation automatique et intelligence artificielle: quelques questions de stratégies, *INRIA*, Congrès de Benodet, 6-10/06/88, pp. 143–175

[Microsoft 90] **Microsoft Corporation**, Microsoft : Rich Text Format Specification, *Manuel Descriptif spécifications RTF*, 1990

[Miranda 91] **Miranda. S.**, L'art des bases de données III - Evaluer SQL, *ed. Eyrolles*, 1991

[Moch 91] **Moch R., Moreau R., Moussé J., Ganascia J.-G., Peccoud D., Recoque A., Liénard J.-S., Laroche–Bouvy D., Briot M., Haton J.-P., Gatty J., Chauvet G., Rabischong P., Truscelli D., Courbin P., Funck–Brentano J-L.**, Intelligence artificielle et bon sens, *Collection Fredrik R. Bull*, éd. Masson, Janvier 91

[Monteil 90] **Monteil M.G., Penot N.**, Indexation automatique. Fonctionnement. Principes Généraux, *Rapport interne EDF/DER/IPN/SID*, n°G91H3002922, Déc. 90

[Moreau 88] **Moreau D.,** Typologie de la communication électronique d'entreprise, *GFI Conseil*, Septembre 1989, N°2

[Nanard 91] **Nanard M.,** Les Hypertextes et les Hypermédias, *Ecole d'été EDF CEA INRIA sur les documents électroniques*, Juillet 91

[Newcomb 91] **Newcomb S. , Kipp N., Newcomb V.T.**, Hytime, the Hypermedia/Time-based Document Structuring Language. *Communications of the ACM*, Vol. 34 (11), Novembre 1991, pp. 67–82

[Nielsen 89] **Nielsen J.**, Hypertext and Hypermedia, *Academic Press*, 1989

[Nielsen 90] **Nielsen J**., the art of navigating through hypertext, *Comm. of the ACM*, Vol. 33 (3), March 1990, pp. 298–310

[Noyelle 88] **Noyelle Y.**, Traitement des langages évolués, *Manuels informatiques Masson*, 1988

[Osterbye 92] **Osterbye K.**, Structural and Cognitive Problems in providing Version Control for Hypertext, *Proc. ACM ECHT Conference*, Milano, November 30-December 4, 1992, pp. 33–42

[Pasquier 89] **Pasquier-Boltuck J., Collaud G., Monnard J.**, Conception et programmation par objets d'un système interactif de création et de consultation de livres électroniques, *Woodman'89*, 29-31 Mai, 1989, Rennes, France, BIGRE n°63-64, pp. 7–17

[Peterlongo 89] **Peterlongo M**., Object Oriented environment O.D.A. editing, *Bigre 63-64*, Mai 1989, pp. 123–131

[Puthuff 78] **Puthuff S.H**, Technical innovations information storage and retrieval, *IEEE Trans. on magnetics*, n° 4, Vol. 14, July 1978, pp 143–148

[Quint 87] **Quint V.**, Les systèmes pour la manipulation de documents structurés, *Bigre + Globule n°53*, Mai 87, pp. 39–87

[Quint 91] **Quint V.**, Les documents structurés, *Ecole d'été CEA INRIA EDF sur les documents électroniques*, 01/07-12/07 1991

[Quint 92] **Quint V., Vatton I.,** Combining Hypertext and structured Documents in Grif, *ACM ECHT Conference*, Milano, November 30 - December 4, 1992, pp. 23–32

[Rada 91] **Rada R.**, Hypertext an paper: a special synergy, *Int. Journal. Inf. Manag.*, 11, 1991, pp. 14–22

[Raymond 88] **Raymond D.R., Tompa F. WM.**, Hypertext and the oxford english dictionary, *Comm. of the ACM*, Vol. 31 (7), July 1988, pp. 871–879

[Richard 90] **Richard G.**, **Rizk A.**, Quelques idées pour une modélisation des systèmes hypertextes, *TSI*, Vol. 9 (6), 1990, pp. 505–514

[Ritchie 89] **Ritchie I.**, Hypertext - Moving Towards large volume, *The Computer Journal*, Vol. 32 (6), 1989, pp. 516–523

[Role 91] **Role F.**, La norme SGML pour décrire la structure logique des documents, *Documentaliste-Sciences de l'information*, Vol. 28 (4-5), 1991, pp. 187–192

[Sabot 92] **Sabot A.,** Principes de gestion, Manuels et dossier de la documentation, *PPRD Coordination de l'Isolement, Dossier d'Administration, Fascicule Plan de la documentation*, (Document A0.00.00), EDF/DER/ERMEL/LGE, Juillet 92

[Salton 91] **Salton G.**, Developments in Automatic Text Retrieval, *Science*, Vol. 253, Aug. 91, pp. 974–980

[Sandvad 89] **Sandvad E.**, Hypertext in an Object-Oriented Programming Environment, *Woodman'89*, Bigre 63 - 64, pp. 30–41

[Savoy 88] **Savoy J.**, Le livre électronique EBOOK3, *T.S.I.*, Vol. 7 (5), 1988, pp. 437–450

[Schütt 90] **Schütt H.A., Streitz N.A**, HyperBase: A Hypermedia Engine Based on a Relational Database Management System, *Proc. of the European Conference on Hypertext*, INRIA, France, November 1990, pp. 95–108

[Shen 89] **Shen S.N.**, **Zhang J.**, Knowledge oriented hypermedia system, *Woodman'89*, Bigre 63 - 64, pp. 307–316

[Silberztein 93] **Silberztein M.**, Dictionnaires électroniques et analyse automatique de textes, *Collection Informatique Linguistique*, éditions Masson, Mai 1993

[Smith 86] **Smith J.M.**,The implication of SGML for the preparation of scientific publications, *The Computer Journal*, Vol. 29 (3), 1986, pp. 13–20

[Smith 88] **Smith J.B., Weiss S.F.**, Hypertext, *Comm. of the ACM*, Vol. 31 (7), July 1988, pp. 816–819

[Southall 88] **Southall R.**, Visual structure and the transmission of meaning, *Document Manipulation and Typography*, Proc. EP'88 Conference, Cambridge University Press, 1988, pp. 35–45

[Stonebraker 86] **Stonebraker M.**, Document Processing in a Relational Database System, *The INGRES Papers*, Addison-Wesley, 1986, pp. 357–375

[Stotts 88] **Stotts P.D.**, **Furuta R.**, Adding Browsing Semantics to the Hypertext Model, *Comm. of the ACM on Document Processing Systems, Santa Fe, New Mexico*, December 5-9, 1988, pp. 43–60

[Stotts 89] **Stotts P.D.**, **Furuta R.**, Petri-Net-Based Hypertext: Document Structure with Browsing Semantics, *ACM Transactions on Information Systems*, Vol. 7 (1), January 1989, pp. 3–29

[Tazi 89] **Tazi S.**, Aide à la structuration des documents pour les systèmes hypertextes, *Bigre 63-64*, Mai 1989, pp. 317–330

[Timbal 84] **Timbal-Duclaux L.**, La qualité des écrits scientificques et techniques, *Pub. EDF-DER*, 2ème édition, 1984

[Tremblay 85] **Tremblay J-P., Sorenson P.,** The theory and practice of compiler writing, *Computer Sciences Series*, Mc GrawHill; 1985

[Van Dam 88] **Van Dam A.**, Hypertext'87 Keynote address, *Comm. of the ACM*, Vol. 31 (7), July 1988, pp. 887–895

[Van Rijsbergen 71] **Van Rijsbergen C.J.,** An algorithm for information structuring and retrieval,. *Comput. J. 14, 4*, 1971, pp. 407–412

[Ventura 88] **Ventura C.A.**, Why switch from paper to electronic manuals? *Proc. ACM Conference on Document Processing systems*, Dec 5-9 1988, Santa Fe, New Mexico, pp. 111–116

[Vidal 93] **Vidal M.-P.**, Les bases de données, *Relations, n°8*, Mai 1993, Pub. interne EDF/GDF, pp. 1–7

[Virbel 87] **Virbel J.**, L'apport de connaissances linguistiques à l'interprétation des structures textuelles, *Structure des documents*, Bigre + Globule n°53, Mai 1987, pp. 77–97

[Vuldy 93] **Vuldy J.-L., Jolly V.**, Réalisation d'une base documentaire appliquée au comportement diélectrique des isolations dans l'air, *Coll. de notes internes de la DER*, 93NO00032, Fév. 93, 26 pages

[Wang 91] **Wang B.**, **Hitchock P.**, INTERSECT: A general purpose hypertext system based on an object oriented database, *Database and Expert System applications*, 1991, ed. Springer-Verlag (USA) pp. 459–464

[Weyer 82] **Weyer S. A.,** The design of a dynamic book for information search, *International Journal Man-Machine Studies*, 17, 1982, pp. 87–107

[Wilkinson 87] **Wilkinson R.T.**, **Robinshaw H.M.**, Proof-reading: VDU and paper text compared speed, accuracy and fatigue, *Behaviour and Information Technology, 6*,2 (April-June 87) pp. 125–133

[Wright 83] **Wright P.**, **Lickorish A.,** Proof-reading texts on screen and paper, *Behaviour and Information Technology, 2*, 3, July-September 1983, pp. 227–235

Annexe

A.1 Extrait d'un document "Code de Mécanique Aster©"

Manuel d'Utilisation
Fascicule U9.01
Document U9.01.01

Un exemple complet de calcul en mécanique linéaire stationnaire avec *Aster*

Résumé :

On propose un exemple complet du calcul des contraintes mécaniques dans les soudures d'un piquage de 20" sur une tuyauterie de 32" sous un cas de charge mécanique et thermo-mécanique.

Après l'exposé du problème à modéliser, des choix de modélisation résultant des cas de charge, et des grandeurs physiques à calculer, on expose les commandes à mettre en œuvre dans *Aster* Version 2.3 pour réaliser cette étude.

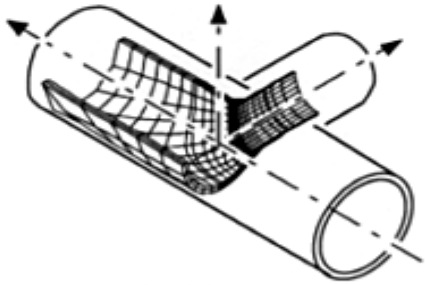

Figure a : Modélisation 3D piquage 20"/32"

Ce document est en quelque sorte une illustration par l'exemple, des documents

- "Déroulement d'une étude avec *Aster*" [U2.01.01]
- "Procédures d'accès à partir de terminaux IBM" [U2.00.02]

dont il est conseillé d'avoir parcouru, au préalable, le contenu.

A.2 Extrait d'un document "Coordination de L'isolement"

Le principe consiste à placer en général aux deux extrémités de la ligne (à moins qu'il n'y ait qu'une seule extrémité possible pour l'alimentation) des réactances phase-terre (dites shunts) dont le rôle est de diminuer la capacité apparente de la ligne. Les valeurs de ces réactances sont choisies pour qu'à vide la tension en bout de ligne ne dépasse pas la tension la plus élevée du réseau. On parle alors de compensation à 30, 50, ... % quand l'énergie réactive consommée par les réactances est égale à 30, 50, ...% de l'énergie réactive produite par la capacité de la ligne. Par exemple pour une compensation à 75% avec une pulsation à fréquence industrielle Ω, on a la relation entre L_c (valeur de l'inductance de compensation) et C (capacité directe de la ligne) :

(form 3)

$$2L_c\Omega = \frac{4}{3}\frac{1}{C\Omega}$$

Ces réactances peuvent être raccordées :

- directement à la ligne,
- ou au tertiaire des transformateurs des postes d'extrémité.

4.2. Compensation par capacité série

La compensation par capacité série [fig 5] n'est pas utilisée dans le seul but de réduire l'effet FERRANTI car la mise en oeuvre de capacité série C_s est beaucoup plus lourde que celle de réactances shunts. Cependant si pour les besoins de transit de courant une telle installation est montée, elle peut jouer un rôle pour la réduction des surtensions temporaires dues à l'effet FERRANTI ou au phénomène de résonance série [C1.20.02]. En effet le mécanisme est double :

- d'une part il y a un diviseur de tension,
- et d'autre part la capacité globale vue de l'entrée de la ligne est diminuée.

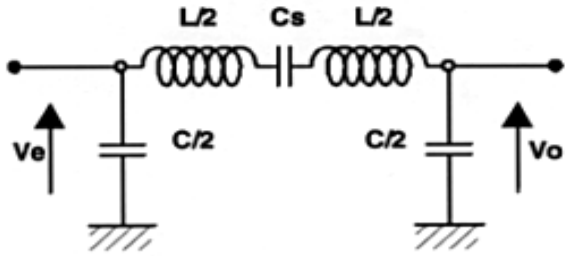

Figure 5 : Compensation par capacité série C_s (avec L et C inductance et capacité équivalente de la ligne)

A.3 Liste de styles et leurs caractéristiques

Catégories	styles	caractères
Texte	Américain	Geneva 10 point, justifié.
	Biblio	Geneva 10 point ; retrait gauche 3,63 cm ; première ligne - 2,23 cm ; justifié ; taquets de tabulation 3,63 cm.
	Standard	Geneva 10 point ; retrait première ligne 0,7 cm ; justifié.
	Enumér	Geneva 10 point ; retrait gauche 2,1 cm ; première ligne - 0,7 cm ; justifié taquets de tabulation 2,8 cm.
Titre des documents	Titre 0	Américain + Geneva 18 point, gras, aligné à gauche encadrement bas (double)
Titre des chapîtres	Titre 1	Geneva 14 point, gras, retrait gauche 1,4 cm, première ligne - 1,4 cm aligné à gauche, encadrement bas (simple).
	Titre 2	Geneva 12 point, gras, retrait gauche 1,4 cm, première ligne - 1,4 cm aligné à gauche.
	Titre 3	Geneva 10 point, gras, retrait gauche 1,4 cm, première ligne - 1,4 cm aligné à gauche.
	Titre 4	Geneva 10 point, gras, retrait gauche 2,1 cm, première ligne - 2,1 cm aligné à gauche.
Table des matières	Tdm 1	Standard + retrait, gauche 2,05 cm ; première ligne - 0,67 cm ; droit 0,01 cm , aligné à gauche ; interligne 14 point ; taquets de tabulation 2,1 cm, 15 cm aligné à droite …,16,08 cm aligné à droite.
	Tdm 2	Standard + retrait, gauche 3,63 cm ; première ligne - 1,55 cm ; droit 0,01 cm , aligné à gauche ; taquets de tabulation 3,6 cm, 15 cm aligné à droite …,16,08 cm aligné à droite.
	Tdm 3	Standard + retrait, gauche 5,33 cm ; première ligne - 1,69 cm ; droit 0,03 cm , aligné à gauche ; taquets de tabulation 5,33 cm, 15 cm aligné à droite …,16,08 cm aligné à droite.
	Tdm 4	Standard + retrait, gauche 5,33 cm ; première ligne - 1,69 cm ; droit 0,03 cm , aligné à gauche ; taquets de tabulation 5,33 cm, 15 cm aligné à droite …,16,08 cm aligné à droite.
Bas de page	Pied de page	Standard + taquets de tabulation 7,5 cm centré, 15 cm aligné à droite …
haut de page	En-tête	Standard + italique : retrait première ligne 0 cm ; encadrement bas (double) ; taquets de tabulation 3,99 cm, 12,03 cm, 14,14 cm, 15,38 cm.
	Entête-texte	Standard + italique : retrait première ligne 0 cm aligné à gauche, taquets de tabulation 2,12 cm, 14,02 cm, 15,49 cm, aligné à droite.

A.4 Description des en–têtes

Les en–têtes des documents techniques relatifs aux projets menés à la Direction des Etudes et Recherches d'Electricité De France, contiennent des informations administratives qui permettent par exemple de situer le document dans un projet, d'en connaître le titre ou d'identifier l'auteur **[cf. A.1] [cf.A.2]**.

Les en–têtes sont constitués de plusieurs "champs", parmi lesquels on trouve : le nom et la version du projet, le titre du document, la date de rédaction du document, le ou les auteur(s), le numéro de page, le nombre total de page, le département d'origine, etc..

Formellement, ils peuvent s'exprimer de la manière suivante :

```
<En-Tete> ::=            <Nom Projet> 'Version' <Version Projet>
                         'Titre ' <Titre Document> 'Date :' <Date>
                         'Auteur(s) :' <Auteur> { <Auteur> }
                         'Clé :' <Clef Document>
                         'Page :' <Numéro page> '/'<Nombre pages>

<Version Du Projet> ::= <Caractères>
<Date Rédaction> ::=<Jour> '/' <Mois> '/' <Année>
                     <Jour> ::=   Nombre
                     <Mois> ::=   Nombre
                     <Année> ::=  Nombre
<Auteur> ::=         Chaîne de caractères
<Numéro de la page> ::= Nombre
<Nombre de pages> ::=   Nombre
<Département> ::=    Chaîne de caractères
<Indice document> ::=   Nombre
<Diffusion> ::=      Chaîne de caractères
```

A.5 La description BNF d'un en–tête

En-tete= NomProjet Blancs GroupeVersion Blancs GroupeTitre Blancs GroupeDate Blancs GroupeAuteur Blancs GroupeClef Blancs GroupePage

NomProjet= StrCaractères

StrCaractères= LetMin StrCaractères | LetMaj StrCractères | Espace StrCaractères

GroupeVersion= LibVersion StrVersion

LibVersion= "Version " | "version" | Version :" | "version :"

StrVersion= Chiffre Point Chiffre

GroupTitre= LibTitre StrTitre

LibTitre= "Titre " | "titre " | Titre :" | "titre :"

StrTitre= StrCaractères

GroupeAuteur= LibAuteur...

GroupeDate= LibDate...

GroupeClef= LibClef...

GroupePage= LibPage...

Blancs= Espace Blancs | Tabulation Blancs

Espace= " "

Point= "."

Nombre= Chiffre Nombre

Chiffre= "0" |...| "9"

LetMin= "a" | "b" | "c" | "d" | "e" | "f" |...| "z"

LetMaj= "A" | "B" | "C" |...| "Y" | "Z"

A.6 La description BNF de références croisées

La grammaire donnée ci–dessous permet d'extraire diverses références (documents, sections, formules, graphiques) des documents EDF.

```
Entree=        RefBibExt | RefTitExt_1 | RefTitExt_2 | RefTitExt_3 | RefTitExt_4 |
               RefTitExt_5 | RefTitExt_6 | RefTitExt_7 | RefTitExt_8 | RefTitExt_9
               | RefTitExt_10 | RefTitExt_11 | RefTitExt_12 | RefForExt_1 |
               RefForExt_2 | RefForExt_3 | RefForExt_4 | RefForExt_5 | RefForExt_6
               | RefForExt_7 | RefForExt_8  | RefForExt_9 | RefFigExt_1 |
               RefFigExt_2 | RefFigExt_3 | RefFigExt_4 | RefFigExt_5 | RefFigExt_6
               | RefFigExt_7 | RefFigExt_8 | RefFigExt_9 | RefBibInt | RefTitInt_1
               | RefTitInt_2 | RefTitInt_3 | RefTitInt_4 | RefTitInt_5 |
               RefTitInt_6 | RefTitInt_7 | RefTitInt_8 | RefFigInt_1 | RefFigInt_2
               | RefFigInt_3 | RefFigInt_4 | RefFigInt_5 | RefFigInt_6 |
               RefFigInt_7 | RefFigInt_8 | RefFigInt_9 | RefForInt_1 | RefForInt_2
               | RefForInt_3 | RefForInt_4 | RefForInt_5 | RefForInt_6 |
               RefForInt_7 | RefForInt_8 | RefForInt_9
RefBibExt*=    CrochetO IdMan IdPar Point IdFas Point IdDoc CrochetO Nombre
               CrochetF CrochetF
RefTitExt_1*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqParag Num1 Point
               Num2 Point Num3 Point Num4 Point CrochetF
RefTitExt_2*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqParag Num1 Point
               Num2 Point Num3 Point Num4 CrochetF
RefTitExt_3*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqParag Num1 Point
               Num2 Point Num3 Point CrochetF
RefTitExt_4*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqParag Num1 Point
               Num2 Point Num3 CrochetF
RefTitExt_5*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqParag Num1 Point
               Num2 Point CrochetF
RefTitExt_6*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqParag Num1 Point
               Num2 CrochetF
RefTitExt_7*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqParag Num1 Point
               CrochetF
RefTitExt_8*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqParag Num1 CrochetF
RefTitExt_9*=  CrochetO IdMan IdPar Point IdFas Point IdDoc CrochetF
RefTitExt_10*= CrochetO IdMan IdPar Point IdFas CrochetF
RefTitExt_11*= CrochetO IdMan IdPar CrochetF
RefTitExt_12*= CrochetO IdMan CrochetF
RefForExt_1*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFor Num1 Point
               Num2 Point Num3 Point Num4 Point SeqTiret Nombre CrochetF
RefForExt_2*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFor Num1 Point
               Num2 Point Num3 Point Num4 SeqTiret Nombre CrochetF
RefForExt_3*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFor Num1 Point
               Num2 Point Num3 Point SeqTiret Nombre CrochetF
RefForExt_4*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFor Num1 Point
               Num2 Point Num3 SeqTiret Nombre CrochetF
RefForExt_5*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFor Num1 Point
               Num2 Point SeqTiret Nombre CrochetF
RefForExt_6*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFor Num1 Point
               Num2 SeqTiret Nombre CrochetF
```

```
RefForExt_7*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFor Num1 Point
               SeqTiret Nombre CrochetF
RefForExt_8*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFor Num1
               SeqTiret Nombre CrochetF
RefForExt_9*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFor Nombre
               CrochetF
RefFigExt_1*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFig Num1 Point
               Num2 Point Num3 Point Num4 Point SeqTiret LetMin CrochetF
RefFigExt_2*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFig Num1 Point
               Num2 Point Num3 Point Num4 SeqTiret LetMin CrochetF
RefFigExt_3*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFig Num1 Point
               Num2 Point Num3 Point SeqTiret LetMin CrochetF
RefFigExt_4*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFig Num1 Point
               Num2 Point Num3 SeqTiret LetMin CrochetF
RefFigExt_5*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFig Num1 Point
               Num2 Point SeqTiret LetMin CrochetF
RefFigExt_6*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFig Num1 Point
               Num2 SeqTiret LetMin CrochetF
RefFigExt_7*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFig Num1 Point
               SeqTiret LetMin CrochetF
RefFigExt_8*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFig Num1
               SeqTiret LetMin CrochetF
RefFigExt_9*=  CrochetO IdMan IdPar Point IdFas Point IdDoc SeqLibFig LetMin
               CrochetF
RefBibInt*=    CrochetO Nombre CrochetF
RefTitInt_1*=  CrochetO SeqParag Num1 Point Num2 Point Num3 Point Num4 Point
               CrochetF
RefTitInt_2*=  CrochetO SeqParag Num1 Point Num2 Point Num3 Point Num4 CrochetF
RefTitInt_3*=  CrochetO SeqParag Num1 Point Num2 Point Num3 Point CrochetF
RefTitInt_4*=  CrochetO SeqParag Num1 Point Num2 Point Num3 CrochetF
RefTitInt_5*=  CrochetO SeqParag Num1 Point Num2 Point CrochetF
RefTitInt_6*=  CrochetO SeqParag Num1 Point Num2 CrochetF
RefTitInt_7*=  CrochetO SeqParag Num1 Point CrochetF
RefTitInt_8*=  CrochetO SeqParag Num1 CrochetF
RefForInt_1*=  CrochetO SeqLibFor Num1 Point Num2 Point Num3 Point Num4 Point
               SeqTiret Nombre CrochetF
RefForInt_2*=  CrochetO SeqLibFor Num1 Point Num2 Point Num3 Point Num4 SeqTiret
               Nombre CrochetF
RefForInt_3*=  CrochetO SeqLibFor Num1 Point Num2 Point Num3 Point SeqTiret Nombre
               CrochetF
RefForInt_4*=  CrochetO SeqLibFor Num1 Point Num2 Point Num3 SeqTiret Nombre
               CrochetF
RefForInt_5*=  CrochetO SeqLibFor Num1 Point Num2 Point SeqTiret Nombre CrochetF
RefForInt_6*=  CrochetO SeqLibFor Num1 Point Num2 SeqTiret Nombre CrochetF
RefForInt_7*=  CrochetO SeqLibFor Num1 Point SeqTiret Nombre CrochetF
RefForInt_8*=  CrochetO SeqLibFor Num1 SeqTiret Nombre CrochetF
RefForInt_9*=  CrochetO SeqLibFor Nombre CrochetF
RefFigInt_1*=  CrochetO SeqLibFig Num1 Point Num2 Point Num3 Point Num4 Point
               SeqTiret LetMin CrochetF
RefFigInt_2*=  CrochetO SeqLibFig Num1 Point Num2 Point Num3 Point Num4 SeqTiret
               LetMin CrochetF
RefFigInt_3*=  CrochetO SeqLibFig Num1 Point Num2 Point Num3 Point SeqTiret LetMin
               CrochetF
RefFigInt_4*=  CrochetO SeqLibFig Num1 Point Num2 Point Num3 SeqTiret LetMin
               CrochetF
```

```
RefFigInt_5*=  CrochetO SeqLibFig Num1 Point Num2 Point SeqTiret LetMin CrochetF
RefFigInt_6*=  CrochetO SeqLibFig Num1 Point Num2 SeqTiret LetMin CrochetF
RefFigInt_7*=  CrochetO SeqLibFig Num1 Point SeqTiret LetMin CrochetF
RefFigInt_8*=  CrochetO SeqLibFig Num1 SeqTiret LetMin CrochetF
RefFigInt_9*=  CrochetO SeqLibFig LetMin CrochetF
Num1*=         Nombre
Num2*=         Nombre
Num3*=         Nombre
Num4*=         Nombre
IdMan*=        LetMaj
IdPar*=        Chiffre
IdFas*=        Chiffre Chiffre
IdDoc*=        Chiffre Chiffre
SeqLibFig=     Blancs LibFig Blancs | Blancs LibFig | LibFig Blancs | LibFig
LibFig=        "Figure" | "figure" | "Fig." | "fig." | "Fig" | "fig"
SeqLibFor=     Blancs LibFor Blancs |  LibFor Blancs | Blancs LibFor | LibFor
LibFor=        "Formule" | "formule" | "Form." | "form." | "Form" | "form" |
               "For." | "for." | "For" | "for"
Blancs=        Espace Blancs | Tabulation Blancs
SeqTiret=      Tiret | Blancs Tiret | Tiret Blancs | Blancs Tiret Blancs
SeqParag=      Parag | Blancs Parag | Parag Blancs | Blancs Parag Blancs
SeqCrochetO=   Blancs CrochetO Blancs | CrochetO Blancs | Blancs CrochetO |
               CrochetO
SeqCrochetF=   Blancs CrochetF Blancs | CrochetF Blancs | Blancs CrochetF |
               CrochetF
Nombre=        Chiffre Nombre
Chiffre=       "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
LetMin=        "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" |
               "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" |
               "w" | "x" | "y" | "z"
LetMaj=        "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" |
               "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" |
               "W" | "X" | "Y" | "Z"
CrochetF=      "]"
CrochetO=      "["
Espace=        " "
Tabulation=    " "
Point=         "."
Tiret=         "-"
Parag=         "§"
```

A.7 Fichier de configuration pour les documents Isolement

```
[ExcludedRtfGroups]
\info
\stylesheet
\fonttbl
\colortbl
\xe
\tc
\bkmkstart
\bkmkend
\template
\nextfile
\fldinst
\pict
\headerf
\footerf
\header
\footerf
\footer

[IncludedRtfGroups]

[CharacterSet]
\mac=MAC
\pc=ASCII
\pca=ASCII
\ansi=ANSI

[CharacterTranscription]
MAC,ASCII,ANSI
181,230,181     # µ
136,133,224     # à
137,131,226     # â
138,132,228     # ä
141,135,231     # ç
142,130,233     # é
143,138,232     # è
144,136,234     # ê
145,137,235     # ë
148,140,238     # î
149,139,239     # ï
153,147,244     # ô
154,148,246     # ö
157,151,249     # ù
158,150,251     # û
159,129,252     # ü
161,248,176     # °
165,45,45       # (puce)
201,32,32       # ...
213,39,146      # '
216,152,255     # ÿ
```

```
[RtfKeywords]
\par
\tab
\row
\cell
\up6

[ChangeKeywords]
\par=10
\tab=9
\row=10
\cell=9
\up6=94

[SpecialCharacters]
\|=32
\:=32
\~=32
\-=45
\_=45
\{=123
\}=125

[Styles]
\s4
\s254
\s253
\s252
\s251
\s250
```

A.8 Extrait des redirections des analyses lexicales (Isolement)

```
<Version>1.0</Version>
<LigProjet1>Coordination de l'Isolement      Version
    1.0</LigProjet1>
<Version>1.0</Version>
<LigProjet1>Coordination de l'Isolement      Version
    1.0</LigProjet1>
<Titre>l'Enclenchement et au Réenclenchement de ligne à
vide</Titre>
<StrDate>11.02.93</StrDate>
<Auteurs>A. SABOT</Auteurs>
<Code_Manuel>C1</Code_Manuel>
<Part_Num>30</Part_Num>
<Doc_Num>01</Doc_Num>
<Document>C1.30.01</Document>
<NbPage>32</NbPage>
<Departement>Service ERMEL/LGE</Departement>
<Indice>C</Indice>
<Diffusion>Participants PPRD</Diffusion>
=============================================_NOD00000
=============================================_NOD00001
<Section>1</Section>
<TitreSection>Description du phénomène</TitreSection>
<Ext1_Manuel>C1</Ext1_Manuel>
<Ext1_Part>01</Ext1_Part>
<Ext1_Doc>01</Ext1_Doc>
<Ext1_Manuel>C1</Ext1_Manuel>
<Ext1_Part>01</Ext1_Part>
<Ext1_Doc>01</Ext1_Doc>
<Ext2_Manuel>C1</Ext2_Manuel>
<Ext2_Part>01</Ext2_Part>
<Ext2_Doc>01</Ext2_Doc>
<Ext2_Manuel>C1</Ext2_Manuel>
<Ext2_Part>01</Ext2_Part>
<Ext2_Doc>01</Ext2_Doc>
<LigReference>[C1.01.01]</LigReference>
=========================================_NOD00002
<Section>1.1</Section>
<TitreSection>Le phénomène : approche simplifiée</TitreSection>
<Ext1_Manuel>C1</Ext1_Manuel>
<Ext1_Part>35</Ext1_Part>
<Ext1_Doc>01</Ext1_Doc>
<Ext1_Manuel>C1</Ext1_Manuel>
<Ext1_Part>35</Ext1_Part>
<Ext1_Doc>01</Ext1_Doc>
<Ext2_Manuel>C1</Ext2_Manuel>
<Ext2_Part>35</Ext2_Part>
<Ext2_Doc>01</Ext2_Doc>
<Ext2_Manuel>C1</Ext2_Manuel>
<Ext2_Part>35</Ext2_Part>
<Ext2_Doc>01</Ext2_Doc>
<LigReference>[C1.35.01]</LigReference>
```

A.9 Fichier de configuration ISOL.MKR

```
[GLOBAL]
Auteurs
Code_Manuel
Part_Num
Doc_Num
Titre
StrDate

[DOCUMENT.TXT]
Code_Manuel
Part_Num
Doc_Num
Titre
Auteurs
StrDate

[SECTION.TXT]
Code_Manuel
Part_Num
Doc_Num
Section
TitreSection

[SECTREF.TXT]
Code_Manuel
Part_Num
Doc_Num
IntSection
LigReference

[SECTREF.TXT]
Ext1_Manuel
Ext1_Part
Ext1_Doc
Ext1_Section
LigReference

[SECTREF.TXT]
Ext2_Manuel
Ext2_Part
Ext2_Doc
Ext2_Section*
LigReference

[REFCROIS.TXT]
Code_Manuel
Part_Num
Doc_Num
Section
Code_Manuel
Part_Num
```

Doc_Num
IntSection

[REFCROIS.TXT]
Code_Manuel
Part_Num
Doc_Num
Section
Ext2_Manuel
Ext2_Part
Ext2_Doc

[REFCROIS.TXT]
Code_Manuel
Part_Num
Doc_Num
Section
Ext1_Manuel
Ext1_Part
Ext1_Doc
Ext1_Section

Table des matières

Chapitre 2 Hypertexte, Bases de Données et Intelligence Artificielle

Chapitre 3 L’hypertextualisation par la structure des documents

Chapitre 4 Le processus d'hypertextualisation automatique

Chapitre 5 Hypertextualisation et Actualisation Apport du modèle relationnel

Chapitre 6 Expérimentation

Illustrations