

Construction des documents multistructurés
dans le contexte des Humanités numériques

Pierre-Édouard Portier

2010

Table des matières

1	Introduction	4
1.1	Trois essais introductifs	4
1.1.1	La notion de document	4
1.1.2	La notion de signe	7
1.1.3	La notion d'annotation	10
1.2	Introduction à la structure de l'exposé	14
2	État de l'art des solutions pour la représentation et l'interrogation des documents multistructurés	15
2.1	Introduction	15
2.2	Illustration de la problématique	19
2.3	Les dimensions de l'analyse	23
2.4	Un état de l'art des principales approches	24
2.4.1	Solutions historiques	24
2.4.2	Les modèles propriétaires	28
2.4.3	Les modèles "compatibles XML"	37
2.4.4	Synthèse	53
2.5	Conclusion	54
3	Méthodologie pour la construction des documents multistructurés	56
3.1	Introduction	56
3.1.1	Omniprésence des hiérarchies	56
3.1.2	Explications au sujet du privilège accordé aux hiérarchies	57
3.1.3	Conclusion	61
3.2	Définitions	61
3.2.1	Arborescence	61
3.2.2	Éléments de notre modèle pour la représentation des documents multistructurés	62
3.3	Stratégie	63
3.4	Implémentation	67
3.5	Conclusion	68

4	Ballade avec les signes	70
4.1	Structuration de l'étude du concept de signe	70
4.1.1	Projet de construction d'une machine sémiotique	70
4.1.2	Structure sérielle de l'exposition de la machine sémiotique	71
4.1.3	Présentation des principaux personnages	72
4.2	Introduction aux trois catégories du phénomène	73
4.2.1	Le signe ... un phénomène!	73
4.2.2	Les trois catégories de Peirce	75
4.3	L'événement, ou l'articulation de la priméité et de la secondéité .	78
4.3.1	La priméité	78
4.3.2	La priméité de l'événement	79
4.4	La tiercéité et la relation	81
4.4.1	Le concept de tiercéité	81
4.4.2	Le concept de relation	84
4.5	La tiercéité et le signe	89
4.5.1	Quelques définitions du signe	89
4.5.2	Le signe triadique : representamen, objet et interprétant .	92
4.5.3	Conclusion	95
4.6	La différence et le signe	95
4.6.1	Perspectives de l'étude	95
4.6.2	La différence comme relation	97
4.6.3	Différence dans les mondes physique et mental	98
4.6.4	La carte et le territoire	99
4.6.5	Différence et énergie	100
4.6.6	Conclusion	101
4.7	Un dialogue pour conclusion	101
5	Évolution et appropriation des classes de termes et des voca-	
	bulaires de relations au moyen d'un modèle réflexif des actions	
	d'annotation	104
5.1	Introduction	104
5.2	Approches existantes	105
5.2.1	Réflexivité	105
5.2.2	Contexte	107
5.2.3	Modèle de la trace des actions d'interaction	109
5.3	Présentation d'un autre point de vue sur les notions de contexte,	
	de trace et d'apprentissage	112
5.3.1	Introduction	112
5.3.2	Traces, Archive et Documents	112
5.3.3	Document, Communication et Contexte	112
5.3.4	Contexte et écriture	113
5.3.5	Apprentissage	114
5.3.6	Apprentissage et changement	114
5.3.7	Apprentissage et contexte	114
5.3.8	Apprentissage de type 3	115
5.3.9	Conclusion	115

5.4	Modélisation et représentation des actions d'annotation	115
5.4.1	Enregistrement d'une représentation des actions d'annotation	115
5.4.2	Visualisation des représentations des actions d'annotation	117
5.5	Conclusion	119
6	DINAH, une plateforme philologique pour la construction de documents multistructurés	120
6.1	Introduction	120
6.1.1	Les plateformes historiques	120
6.1.2	Les plateformes basées sur les technologies du Web	121
6.2	Architecture logicielle de DINAH	121
6.3	Processus de développement de DINAH	124
6.3.1	Introduction	124
6.3.2	Point de vue historique	124
6.3.3	Un processus itératif	126
6.4	Scénario d'utilisation	126
6.5	Conclusion	134
7	Conclusion et perspectives	136
7.1	Bilan	136
7.2	perspectives	137
7.2.1	Utiliser la contrainte d'acyclisme pour assister la construction des vocabulaires	137
7.2.2	Gestion fine de la localisation des données	137
7.2.3	Création générique des mécanismes de traçage des actions d'annotation	137
	Appendices	138
A	Deux fonctions Javascript	139
A.1	makeIntervals	139
A.2	makeHtml	141

Chapitre 1

Introduction

1.1 Trois essais introductifs

Structure de l'introduction Pour introduire notre travail, nous avons choisi de rejoindre successivement trois points de vue qui dévoilent notre sujet d'étude. Ils sont choisis élevés pour laisser les détails invisibles. Le projet est de rendre audibles les différentes tonalités du texte à suivre. Un petit essai exprime chaque point de vue. Les trois essais ont respectivement pour centres les notions de document, de signe et d'annotation.

1.1.1 La notion de document

Le sujet de l'étude Nous étudions le processus de création de documents. Nous posons que ces derniers sont les produits d'interactions qui se jouent au sein d'un système qui comprend plateformes informatiques et opérateurs humains. Au fil de cette étude, nous ferons attention à conserver et à expliciter ce point de vue global ou systémique. Afin d'introduire ce travail, nous mentionnons maintenant quelques notions qui se verront par la suite définies avec précision.

Le contexte documentaire de l'étude D'abord, et sans encore se poser la question ontologique de savoir ce qu'est un document, il est nécessaire de décrire, même grossièrement, les types de documents auxquels nous aurons affaire. C'est aussi l'occasion d'introduire le contexte applicatif qui a permis d'éprouver nos hypothèses. Nous avons travaillé avec des chercheurs de l'ENS Lyon au sein de l'Institut Desanti¹. Cet institut s'est donné pour mission de valoriser les travaux du philosophe Jean-Toussaint Desanti. Une des particularités de ce dernier est d'avoir entretenu un rapport savant avec les mathématiques. Cette partie de sa recherche donnera lieu à la publication de son œuvre principale : *Les Idéalités mathématiques, recherches épistémologiques sur le développement de la théorie des fonctions de variables réelles* (1968). Par ailleurs, et nous ne pouvons ici

1. <http://institutdesanti.ens-lyon.fr/>

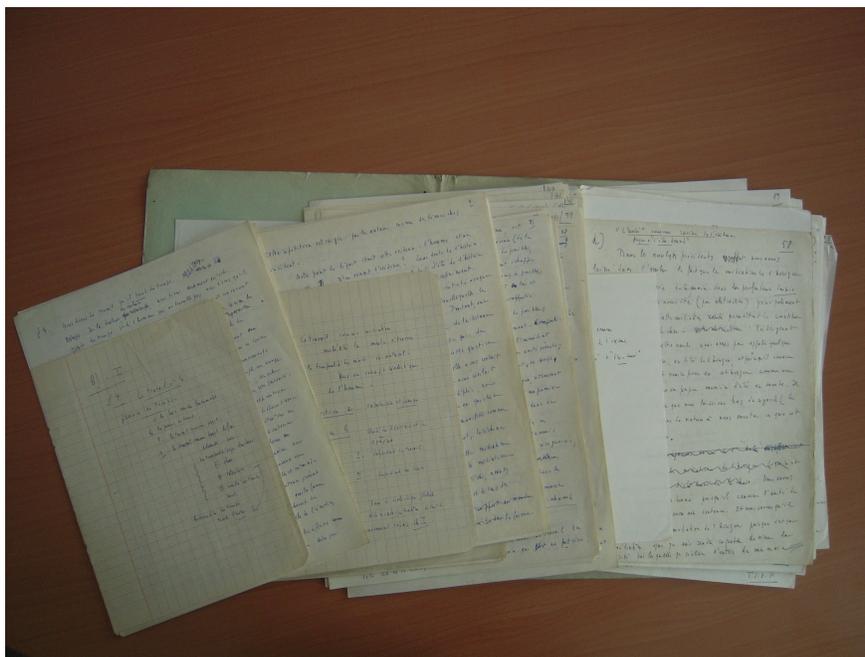


FIGURE 1.1 – Photographie d’une pochette extraite de l’archive Jean-Toussaint Desanti

qu’être anecdotiques, il s’engage en 1940 dans la résistance ; il est proche de Jean-Paul Sartre et d’André Malraux ; il rejoint le parti communiste qu’il quittera en 1956 ; il dirige la thèse de doctorat d’État de Jacques Derrida etc. L’Institut s’est vu confier l’ensemble des manuscrits du philosophe qui furent numérisés par l’ENS puis remis à un organisme de conservation, l’IMEC. Cette archive comprend des notes diverses rédigées le plus souvent sur des cahiers d’écolier, des feuilles éparses regroupées dans ce que nous appelons des pochettes qui peuvent aussi comprendre divers documents comme des articles de presse, des papiers personnels, etc. (voir la figure 1.1 pour un exemple de pochette extraite de l’archive). Quelquefois nous trouvons des documents qui préparent une œuvre connue et publiée mais dans la majorité des cas il s’agit d’inédits.

Les objectifs de ce travail exploratoire Les membres de l’institut Desanti, que ce soit en tant que chercheurs, ingénieurs ou éditeurs participent à faire redécouvrir la pensée de ce grand philosophe. Nous participons à cette entreprise en imaginant de nouvelles conditions à l’étude critique des textes. En effet, l’apparition du domaine informatique au sein de ce système critique n’apporte pas uniquement de nouveaux outils mais, change la pratique même de cette étude... C’est notre thèse. Cependant, tout apport théorique serait comme nul, s’il n’était accompagné des programmes qui permettent de le valider, c’est-à-dire de vérifier

expérimentalement que cette idée s'intègre harmonieusement dans un système existant. Dans notre cas, nous pensons que cette intégration est une transformation substantielle des systèmes qui supportent l'étude critique des textes, à tel point que nous pouvons certainement parler de l'apparition d'une nouvelle pratique de la philologie. Cette transformation, nous avons pu la vérifier. En effet, nous avons développé une série de programmes informatiques qui forment un tout cohérent, une plate-forme philologique, nommée DINAH. Elle est utilisée par les chercheurs de l'Institut, et à observer les interactions qu'elle suscite, nous avons pris conscience de ce changement de pratique. Toujours afin d'introduire notre propos, esquissons une description des principales interactions qui, lorsqu'elles sont composées, mènent d'images de pages manuscrites numérisées à des documents complexes.

Quelques opérations essentielles L'archive manuscrite Jean-Toussaint Desanti peut-être modélisée comme un ensemble de collections. Une collection est définie comme un ensemble ordonné de pages, à chaque page peuvent être associées une ou plusieurs collections (ces dernières sont appelées sous-collections de leur collection parente). Une première opération consiste à naviguer dans cette hiérarchie souvent profonde. Une seconde opération doit permettre la création de nouvelles collections qui sont des assemblages d'éléments (pages ou collections) extraits des collections primaires, c'est-à-dire celles qui correspondent à l'état de l'archive telle que léguée à l'ENS par Jean-Toussaint Desanti. Une troisième opération, qui peut être qualifiée de complexe, c'est-à-dire nécessitant d'être décomposée en sous opérations, est la transcription. Elle comprend la création d'une chaîne formée de la transcription des caractères manuscrits, mais aussi l'annotation d'intervalles de cette chaîne, le découpage de zones dans l'image de la page manuscrite, l'association de ces zones à des intervalles du texte. L'annotation d'un intervalle de la chaîne de caractères transcrite consiste à associer un terme à cet intervalle. Par exemple, il pourra s'agir d'associer à la chaîne de caractères «Kant », le terme « Emmanuel Kant » qui est le label d'un identifiant unique qui caractérise le fameux auteur. Une quatrième opération consiste à tisser des liens entre éléments quelconques du système. Par exemple, il pourra être énoncé qu'une collection a servi à préparer la rédaction d'une page appartenant à une autre collection ; ou bien que tels mots transcrits ou telles zones d'une image sont les indices d'une certaine proximité (historique, conceptuel,...) entre deux éléments de l'archive (page, collection,...). Chacune de ces opérations est rendue possible grâce à la plate-forme Dinah. Nous pouvons seulement noter pour l'instant que le changement de pratique mentionné plus haut, relève principalement de deux opérations implicites : la création de classes (par définition, une classe est un ensemble de termes qui peuvent être utilisés pour caractériser des éléments quelconques) et de vocabulaires (par définition, un vocabulaire est un ensemble de relations qui peuvent être utilisées pour relier des éléments quelconques). Après avoir introduit ces quelques opérations, il est temps d'approcher une première fois la notion abstraite de document.

Une approche de la notion de document Dans notre contexte applicatif, ce que nous appelons document possède la structure d'un graphe, c'est un ensemble d'éléments interconnectés : des pages manuscrites numérisées, du texte transcrit et annoté, des collections de pages ordonnées, des zones d'images de manuscrits isolées, des relations nommées entre éléments quelconques,... On peut imaginer d'autres objets qui s'ajouteraient à cet ensemble. Il semble ainsi difficile de définir les frontières du document. Mais cette difficulté n'est pas propre au contexte numérique. Seulement, ce dernier réifie des relations qui étaient tenues auparavant implicites. Si nous cherchons un invariant pour le document, nous trouvons le couple forme/contenu et ses variantes. Comme c'est souvent le cas, distinguer un couple, ici forme/contenu ou signifiant/signifié ou..., revient à susciter un troisième terme, ici le processus interprétatif. Ce troisième terme aura tendance, au moins dans un premier temps, à mettre l'accent sur un des deux premiers termes (ici la forme, le signifiant,...). Pour l'instant, nous ne faisons que citer un certain nombre de difficultés sur lesquelles nous reviendrons dans le cours de notre exposé. Il est suffisant de retenir que, premièrement, il est délicat de définir statiquement la notion de document, et deuxièmement, on gagne en efficacité à considérer le document comme un graphe. Une opération courante consiste à découper ce graphe en sous-graphes qui sont d'ailleurs souvent des arbres. C'est au sens de cette opération de découpage que nous pouvons dire d'un document qu'il est multistructuré. Maintenant que nous avons introduit la notion de document dans le contexte de la plate-forme Dinah nous sommes prêts à énoncer notre problématique principale.

Un énoncé de la problématique Nous avons vu que les documents sont des objets complexes construits à l'occasion d'interactions diverses au sein d'un système qui comprend programmes informatiques et utilisateurs humains. De plus, nous avons affirmé qu'au cours de sa construction un document pouvait gagner plusieurs niveaux de structuration. Dans ce contexte, nous nous donnons pour problématique l'étude des processus de création de ces différentes structures. Cette étude n'est pas neutre et passive, mais engagée et active car elle s'accompagne de la création d'outils qui transforment ces processus.

1.1.2 La notion de signe

Le sujet de l'étude Nous projetons de créer ce que nous choisirons de nommer une machine sémiotique. Nous voudrions en fournir en quelque sorte les plans, et en entamer la construction. Sémiotique, qualifie ce qui a trait aux signes. Pourquoi avons-nous ressenti le besoin d'user de cette notion ? Et quel privilège lui donner par rapport aux notions de structure, de trace, de différence... qui, nous le verrons, proviennent d'une même évidence. En fait, et nous aurons à le montrer, il n'y a rien qui fasse du signe une notion plus fondamentale que, par exemple, celles tout juste négligemment citées. Il faut donc que la raison de ce choix soit pragmatique, personnelle, historique,... Et nous la trouvons en fait dans la proximité que nous entretenons avec un certain nombre d'auteurs, à commencer par Charles Sanders Peirce. Ce dernier, un auteur américain de la

fin du XIXe siècle, est considéré comme un des deux fondateurs de la sémiotique ou étude des signes. L'autre étant le linguiste Ferdinand de Saussure.

Une exposition de la démarche C'est des écrits sur le signe de Peirce que nous tirerons une explication de la notion de signe et, d'ailleurs, de toute une série de notions connexes. Nous fonctionnerons en effet par séries de concepts. Il y aura une série principale qui du concept général de phénomène nous conduira à celui de signe. Deux séries secondaires viendront se greffer sur cette série principale. La première articulera les concepts de langage, de communication et de différence pour finalement exprimer cette notion de machine sémiotique. La seconde développera le concept d'événement pour proposer une interprétation du couple lecture/écriture (autrement dit, une interprétation de l'interprétation). Sur la première de ces deux séries secondaires se greffera une série tertiaire qui développera le concept d'apprentissage. Mais, pour le moment, il s'agit d'introduire ce projet de construction d'une machine sémiotique, en évitant de rentrer dans les détails de son développement.

Une introduction au concept de signe Nous devons donc faire appel à une compréhension commune du mot « signe ». En ce sens, il semble qu'on entende par signe, quelque chose de perçu qui tient lieu d'une autre chose (pouvant être toujours, et étant le plus souvent, inaperçue). Admettons que cette médiation soit le domaine du signe. La relation de tenant lieu pourra prendre de multiples formes auxquelles seront associés des types de signes différents. Par exemple, l'icône est le signe qui renvoie à son objet par des qualités qui lui sont propres, il en va ainsi du portrait photo qui ressemble à celui dont il est le portrait, ou bien des icônes de nos logiciels (la disquette pour l'action de sauvegarder, le pinceau pour l'action de dessiner, etc.). L'indice quant à lui renvoie à son objet parce qu'il en est affecté, autrement dit il n'existerait pas sans l'existence de son objet. Nous pouvons prendre l'exemple classique de la fumée qui est indice du feu. Enfin, le symbole renvoie à son objet en vertu d'une loi ou d'une habitude. Prenons l'exemple du code de la route dont la plupart des signes appartiennent à cette catégorie. Nous pourrions citer d'autres types de signes, et d'ailleurs nous le ferons en temps voulu, mais cette première énumération nous semble suffire à convaincre de ce que le concept de signe, en tant qu'objet présent tenant lieu d'un objet potentiellement absent, trouve une application privilégiée dans le domaine des interactions médiées par des outils informatiques.

Un énoncé de la problématique Si donc, nous participons, comme il était dit plus haut, à la transformation d'une activité, l'étude critique des textes, par l'introduction d'outils informatiques, et si, de plus, nous tenons au succès et à la qualité de cette entreprise, il semble que nous ayons tout intérêt à porter notre attention sur le jeu de la notion de signe au contact de nos productions logicielles. De plus, cette activité, dont nous pronostiquons, et même attendons, la transformation, est loin d'être neutre dans son rapport au concept de signe. En effet, elle a trait à la mise en valeur, à la compréhension et à l'interprétation de

textes. Or, ces dernières opérations passent par la création, toujours renouvelée, de signes qui viennent se greffer au texte pour en orienter l'interprétation. Notre questionnement a pour objet premier cette dernière création. Quelles en sont les conditions ? Dans quels processus plus larges s'inscrit-elle ? Certains signes, une fois créés, sont réutilisés dans de nouveaux contextes : comment assurer une certaine cohérence dans l'usage d'un signe ? Peut-il y avoir un apprentissage du bon usage d'un signe ? Peut-il y avoir un apprentissage de la création même de nouveaux signes ? Peut-on orienter la compréhension d'un texte par l'ajout de signes ? Quel sens donner à la notion de document dans ce contexte sémiotique toujours mouvant, en éternelle croissance ? Autant de questions qu'il nous faudra adresser au cours de notre ballade autour du signe.

La différence et le signe Sur ce chemin, nous croiserons heureusement des aides précieuses auxquels nous emprunterons, pour souvent ensuite les déformer, quelques notions. Pour le moment, citons seulement le nom de Gregory Bateson qui nous sera cher. Il s'agit d'un américain inclassable qui, entre autres nombreuses choses, a pensé de manière extrêmement originale le concept d'information. Nous nous attardons sur ce point dans cet essai introductif afin d'éclaircir le sens du nom donné à notre objectif : une machine sémiotique. Nous venons d'expliquer l'emploi de l'adjectif « sémiotique », il nous reste à expliquer pourquoi il vient qualifier le mot « machine ». Or, dans l'explication qu'il propose du concept d'information, Grégory Bateson, fait jouer le concept de différence, et de ce jeu nous montrerons que naît une interprétation unifiante de ce qu'il fallait avant appeler l'interaction entre un homme et une machine. Ainsi, ce qu'il faut entendre par « machine sémiotique » ne correspond ni à ce que l'on met communément sous le terme de machine, ni à une autre manière de désigner la dualité homme/machine, mais à un système pas plus humain que machinique et dont l'existence est justifié par sa puissance d'explication. Mais, c'est à notre tour de nous expliquer !

La carte et le territoire Bateson nous parle de cette classe de représentations (est-elle identique à la classe de toutes les représentations ?) qui se modélisent comme un ensemble de différences. Son illustration archétype pourrait être la relation entre la carte et le territoire. En effet, la carte retient du territoire un ensemble de différences (altitude, vitesse de circulation atteignable par un véhicule roulant motorisé, etc.). Parmi ces différences, sont isolés des ensembles, il s'agit autrement dit de différences de différences dont fait d'ailleurs signe ce qui est couramment nommé la "légende". Mais, comme aimait à le souligner Kozzyski, la carte n'est pas le territoire. C'est-à-dire que seul un nombre réduit des différences du territoire sont conservées sur la carte : les différences qui "font une différence". La "différence qui fait une différence", c'est ainsi que Bateson définit parfois l'information. Par ailleurs, "faire une différence", c'est conserver une certaine puissance d'explication. Et "conserver une puissance d'explication", c'est ne pas exclure du modèle (la carte) des différences dont l'absence rendrait le phénomène (le territoire) inexplicable. Ainsi, ce découpage phénoménologique

ne cesse de différer en fonction du sujet d'étude que se donne l'observateur (cette dernière notion devrait elle-même être reconsidérée selon ce point de vue informationnel, mais nous déborderions alors du cadre introductif). Par exemple, pour expliquer le phénomène d'un homme aveugle rejoignant un parc, sa canne, en tant qu'elle est le véhicule de la transformation de différences du chemin, est indissociable du système étudié. Une fois un banc atteint, et l'attention portée sur le chant des oiseaux, la canne, qui ne transporte plus de "différences qui font une différence", disparaît du système étudié. Nous pouvons remarquer une similitude entre le signe et l'unité d'information (la différence qui fait une différence).

Une approche de la machine sémiotique C'est ainsi que nous pouvons introduire la notion de machine sémiotique comme un agencement de signes dont la valeur (ou réussite) dépend de sa puissance d'explication relativement à un sujet d'étude qui n'est pas toujours bien défini. Il s'agit donc pour nous de parvenir à la construction d'une telle machine pour le cas de l'étude critique de textes. Pour mener à bien cette entreprise, nous aurons besoin d'entrer dans les détails d'un ensemble de concepts centrés sur celui de signe et qui n'ont été dans cet essai introductif qu'esquissés. Cette démarche a priori abstraite semblera étrange dans le contexte d'un travail qui, entre autres résultats, propose concrètement une plate-forme logicielle qui peut être vue comme assistant des chercheurs philologues dans leurs études. Elle est cependant nécessaire car, justement, il ne s'agit pas (seulement) de produire un programme autour duquel se constituerait un groupe d'utilisateurs qui partageraient l'impression d'un gain (en facilité, rapidité, etc.) à adopter ce médium. Car il s'agit (aussi), répétons le, de participer à une entreprise plus risquée de constitution d'une activité dont la nouveauté ne se réduit pas à une variante de la dualité Homme/Machine mais à sa dissolution dans un agencement (de différences) différent pour lequel les frontières classiques (la peau, ...) ne sont plus pertinentes. Ainsi, ce risque serait démesuré si nous ne pouvions compter sur un fond abstrait (au sens d'extrait) qui n'attire pas en direction des anciennes habitudes dichotomiques (et nous pensons en particulier à ce qui s'est rangé longtemps sous la rubrique "intelligence artificielle"). Ce fond, il a fallu le construire.

1.1.3 La notion d'annotation

Contexte de l'étude Nous cherchons à développer un système générique pour l'annotation sémantique de documents dans un environnement multi-utilisateurs. Comme nous l'avons vu précédemment, ce système sera in fine spécialement adapté pour les documents de type images manuscrites associées à leur transcription. Mais, cette spécialisation apparente n'est que le résultat d'un effort de présentation d'une plate-forme qui en elle-même reste adaptée à l'annotation de tous types de documents textuels ou picturaux. Par ailleurs, la plupart des résultats que nous proposerons, et qui tendent à rendre l'opération d'annotation aussi efficiente que possible en présence de plusieurs utilisateurs

qui travaillent sur des documents similaires, s'appliqueraient aussi à l'annotation de documents vidéo bien que nous n'ayons pas développé les outils qui permettraient d'éprouver cette hypothèse. Posons qu'une annotation soit de la greffe d'information sur un fragment de document. Remarquons qu'à la limite, au minimum un fragment peut toujours être extrait du document, la greffe porte alors sur le document "en entier". Par ailleurs, la connaissance des lois qui régissent fragmentation et greffe dépend toujours du contexte technique où sont plongés les documents.

Le standard XML et la fragmentation Cependant, des standards existent qui constituent toute une économie où se fixent ces contextes et où sont promus leurs usages. Parmi ces derniers, le langage XML se distingue par l'ampleur de son adoption et la diversité des domaines qu'il touche. Son succès est dû, entre autres, à son modèle générique qui répond dans un même mouvement aux deux opérations de fragmentation et de greffe. Cette dernière remarque n'étant pleinement justifiée que dans le contexte de documents textuels, nous restreignons notre explication, dans un premier temps introductif, à ces derniers. La syntaxe du langage XML permet de rendre explicite une distinction de types logiques entre le texte original et un méta-texte qui vient en décrire des fragments. Par exemple, un haïku du poète Issa Kobayashi peut être encodé en XML comme suit :

```
<lg type="haiku">
<l>chrysanthèmes en fleur</l>
<l>flotte aussi dans l'air</l>
<l>une odeur d'urine</l>
</lg>
```

Les fragments de texte sont isolés par un couple de "balises". Une balise ouvrante est de la forme : `<nomBalise att1="val11 val12" att2="val21">`, avec sa balise fermante de la forme : `</nomBalise>`. Le texte initial est considéré comme une séquence (ou "chaîne") d'éléments insécables (appelons les "caractères"). Un couple de balises greffe sur une sous-chaîne de caractères l'information portée par le nom de la balise et un ensemble non ordonné de couples attribut/valeurs. Les balises peuvent être imbriquées mais ne peuvent pas se chevaucher. Ainsi, une fois imposée l'existence d'un couple de balises qui comprend la chaîne entière du document original, nous en déduisons que tout document XML est la représentation linéaire d'une structure arborescente. En conclusion, en ce qui concerne la fragmentation, étant donné un document textuel, le langage XML ne permet pas de construire tous les ensembles de fragments possibles, mais seulement ceux qui ne contiennent pas d'éléments qui se chevauchent.

Le standard XML et la greffe d'information Étudions à présent comment ce langage gère la greffe d'information. Trois éléments la dirige : le choix du nom de la balise, le choix des attributs et l'imbrication des balises. Dans l'exemple précédent, les balises "lg" et "l" sont présentes, la balise "lg" possède

un attribut qui a pour nom "type" et pour valeur "haïku", et enfin, trois balises "l" sont imbriquées dans une balise "lg". Quel sens donner à ces trois types d'informations? Que signifient les signes "lg" et "l"? Comment interpréter la valeur d'un attribut nommé "type" dans le contexte d'une balise nommée "lg"? Mais avant tout, à qui adresser ces questions? Car, il faut bien que ces signes aient été définis. Quand même implicitement, quand même à une seule reprise, c'est-à-dire par l'auteur de cet essai lorsqu'il rédige l'exemple du haïku, il faut que ces signes aient été choisis pour résonner autrement qu'en leur seule articulation alphabétique : "elle - gé". A moins que ce ne soit le chaos, le bruit, le signe en tant que signe.

Sémantique et correction grammaticale En fait, nous faisons ici les naïfs pour les besoins de l'introduction, mais nous savons, avec nos lecteurs, que le langage XML propose un mécanisme de schéma qui permet de tester la validité d'un document en fonction d'une grammaire. Ainsi, le créateur du dialecte XML de description d'un haïku pourrait déclarer formellement que tout élément "l" doit être fils d'un élément "lg", que tout élément "lg" peut posséder un attribut "type" dont la valeur est une séquence de caractères alphabétiques, etc. En présence d'un tel schéma, nous pourrions décider de la correction grammaticale de l'exemple précédent. Mais, selon l'exemple célèbre de Noam Chomsky, la phrase "Colorless green ideas sleep furiously.", bien que correcte du point de vue de la grammaire anglaise, ne porte pas de sens car elle ne serait jamais apparu dans aucun discours. Mais, si cet exemple sert notre argumentation en confirmant qu'il ne suffit pas d'une grammaire bien définie pour interpréter le sens de documents rédigés dans un dialecte XML, il nous faut pourtant le discuter. En effet, cela ne fait aucun doute que la phrase ci-dessus citée revêt bien des sens ... Dont l'un d'entre eux, et non le moindre, serait d'être un exemple de phrase grammaticalement correcte mais qui n'a pas de sens! Mais nous reviendrons sur ces difficultés dans le cours de l'exposé.

Documentation des vocabulaires Maintenant que nous comprenons l'insuffisance d'une grammaire comme seul guide pour interpréter un document encodé au moyen du formalisme XML, nous en déduisons la nécessité d'une documentation qui oriente l'interprétation des différents éléments qui composent un dialecte XML. Sur notre exemple précédent, nous avons annoté le haïku au moyen de balises choisies parmi celles proposées par le dialecte XML TEI (Text Encoding Initiative). La TEI est un consortium international qui développe et maintient un jeu de balises XML bien adapté à l'encodage d'une grande variété de textes (historiques, philosophiques, poétiques, littéraires, sociologiques, etc.). Une part importante du travail de la TEI consiste à rédiger une documentation précise de chacun des éléments XML proposés. Cette dernière, en langue naturelle, doit permettre l'apprentissage d'un usage correcte du vocabulaire de la TEI. Elle est illustrée par de nombreux exemples.

Une économie de la construction des vocabulaires Nous percevons que la création de vocabulaires de termes d'annotation relève d'une économie complexe. Pour mieux illustrer ce processus, prenons l'exemple des microformats². Ce mouvement consiste en une nouvelle explicitation du contexte sémantique d'un dialecte XML : XHTML. Ce dernier nous est bien connu puisque la plupart des pages visibles sur le Web sont le résultat de l'interprétation par un programme (le navigateur) de documents rédigés dans ce langage. C'est dire qu'au moment de sa conception, ce vocabulaire était pensé pour permettre de greffer de l'information au sujet de la présentation du document. L'objectif des microformats est de rendre explicites, au sein d'un vocabulaire existant (XHTML), de nouvelles dimensions sémantiques qui déclenchent des interprétations hors du seul domaine de la présentation. Cette explicitation doit avoir lieu au sein des limites syntaxiques imposés par le schéma qui spécifie la validité d'un document XHTML. Prenons pour seul exemple l'attribut "rel". Dans le contexte traditionnel d'un document XHTML, sa valeur presque exclusive est "stylesheet" et sert à indiquer la présence d'une feuille de style utilisable pour formater le document courant. Les microformats proposent de l'utiliser pour spécifier la relation entre la source et la destination d'un hyperlien. Par exemple, assigner la valeur "directory" à l'attribut "rel" d'un hyperlien, signifie que la destination du lien est une liste de répertoires dont l'un des éléments pointe vers la page en cours. De même, ajouter la valeur "tag" à l'attribut "rel" d'un hyperlien signifie que la destination est une description du sujet de la page (ou d'une portion de la page) en cours. A l'aide de l'exemple des microformats nous espérons avoir éclairci ce que nous entendons par une "économie" de la spécification du contexte sémantique d'un vocabulaire de termes d'annotation. Finalement, nous avons montré, au moyen de l'exemple du langage XML, que la greffe d'information, en tant qu'inséparable de toute opération d'annotation, implique un contexte sémantique qui doit être rendu, même si toujours partiellement, explicite.

Application de la distinction fragmentation/greffe à notre système

Rappelons que nous nous proposons de développer un système générique pour l'annotation de documents. Le langage XML peut sembler une base solide pour un tel système. Et en effet, même si finalement nous ne le prenons pas pour fond, une analyse plus précise de la manière dont il comprend les deux faces de l'annotation (fragmentation et greffe d'information) nous procure les éléments nécessaires à la construction de notre système. D'abord, une étude des implications de la restriction hiérarchique de la fragmentation nous ouvre à une lecture neuve de la problématique dite des documents multistructurés. Ces derniers partagent, entre autres caractéristiques, de ne pas convenir à une mise en forme arborescente. Ils questionnent ainsi un des fondements du formalisme XML. Nous choisissons de les approcher, non pas en tant qu'objets finis, mais dans le mouvement de leur construction. C'est-à-dire que nous nous intéressons aux moments de la construction d'un document qui introduisent localement une structure de graphe et non plus seulement un arbre. Nous essayons de détermi-

2. <http://microformats.org/>

ner si ces "brusques sautes de syntaxe" peuvent être porteuses de sens. Ensuite, nous cherchons comment expliciter le contexte sémantique de la greffe d'information afin de permettre la construction dynamique de vocabulaires de termes d'annotation. Nous trouvons le germe d'une solution dans la présentation en contexte des actions de création, transformation et utilisation des termes d'annotation. Au moyen de fonctions de manipulation de cette trace d'événements d'interaction, nous parvenons à construire dynamiquement la documentation d'un vocabulaire de termes d'annotations.

1.2 Introduction à la structure de l'exposé

Nous commencerons la synthèse de notre travail exploratoire par une étude des solutions existantes pour la représentation et l'interrogation des documents multistructurés. Aux termes de cette première étude, nous découvrons qu'aucune solution ne prend pleinement en compte la construction des documents multistructurés. C'est pourquoi nous consacrons un chapitre à l'élaboration d'une méthodologie pour la construction des documents multistructurés. Ensuite, nous ponctuons cette première contribution par une étude du concept de signe. Il se trouve en effet qu'à concevoir notre méthodologie nous avons assez naturellement rejoint une perspective, pressentie comme originale, sur les interactions Homme-Machine. Ainsi, cette étude du concept de signe est le moyen que nous avons trouvé pour approfondir ce pressentiment. Munis de ce bagage théorique, nous abordons ensuite la présentation de la deuxième partie de notre contribution : l'utilisation de la trace des actions d'annotation afin de documenter et de contrôler l'évolution des classes de termes et des vocabulaires de relations. Finalement, nous décrivons l'architecture et présentons un scénario d'utilisation de DINAH, notre plateforme philologique pour la construction de documents multistructurés.

Chapitre 2

État de l'art des solutions pour la représentation et l'interrogation des documents multistructurés

2.1 Introduction

XML XML (eXtensible Markup Language) [15] est le standard qui domine aujourd'hui les formalismes propres à la description et à la manipulation des documents électroniques. En effet, toute une galaxie d'outils et de méthodes l'accompagne et tendent à le rendre indispensable. Des éditeurs spécialisés aux langages de requête [27, 11] en passant par les langages de transformation[26], il s'agit d'un environnement très complet qui permet le développement d'applications documentaires spécialisées. Or les systèmes d'information consomment et produisent de nombreux documents électroniques. Ainsi, la documentation technique, l'édition électronique, la mise à disposition de données métier sur le Web, les bibliothèques électroniques, etc. sont autant d'exemples d'activités qui reposent sur une utilisation intensive de documents.

L'origine de XML Avant que des standards n'apparaissent, les formats de documents étaient spécifiques aux outils qui permettaient de les manipuler. Il existait donc une multitude de formalismes le plus souvent incompatibles et qui impliquaient de nombreux traitements intermédiaires lorsqu'il s'agissait de combiner les traitements offerts par différents outils. C'est pourquoi ont été créés des langages qui servent à décrire génériquement la structure d'un document. Ainsi, à la fin des années quatre-vingts, a été défini SGML (Standard Generalized Markup Language) [45], un langage déclaratif pour la création et l'échange de documents structurés. SGML permet d'encoder la structure logique (souvent

complexe) d'un document à l'aide de balises qui sont insérées au fil du texte et qui peuvent être définies au moyen d'une grammaire dans un document séparé. La grande force de ce standard réside en la séparation de la structure logique (encodée avec le texte) et des instructions de présentation. Ces dernières peuvent être définies dans une feuille de styles indépendante du document. Ainsi, Airbus pouvait reprendre la documentation de la SNECMA (une société qui construit des moteurs) et l'afficher selon ses propres normes internes de présentation. Depuis 1998, XML semble avoir pris la place de SGML et être devenu le langage standard pour la description de documents structurés. En fait, XML est principalement la sélection d'un sous-ensemble de SGML choisi pour faciliter l'implémentation de parsers.

Les documents semi-structurés Les documents XML sont souvent qualifiés de semi-structurés. Qu'est-ce à dire ? Mais d'abord, que faut-il entendre par des données non structurées ? Il s'agit de données qui peuvent être de n'importe quel type, qui ne semblent pas suivre quelques format ou règles, au fond, qui ne sont pas prédictibles. Au contraire des "données structurées". Ces dernières sont organisées en fragments élémentaires qui peuvent être appelés "entités". Les entités similaires sont regroupées et forment des classes (ou des relations). Les entités d'un même groupe possèdent les mêmes éléments de description appelés "attributs". Ces attributs sont typés et leur ordre importe. Le modèle relationnel est un exemple de modèle de description de données structurées. Les données semi-structurées correspondent à une forme d'adaptation du monde des bases de données à celui des documents. Elles définissent aussi des fragments élémentaires (entités) et permettent de regrouper les entités similaires. Cependant, les entités d'un même groupe n'ont pas nécessairement les mêmes attributs. De plus, l'ordre des attributs peut ne pas compter et ils peuvent parfois être omis. Enfin, le type d'un même attribut peut changer en fonction de l'entité (par exemple, une adresse e-mail ou un nom de personne peut ne pas toujours avoir la même forme).

Documents XML et structure hiérarchique La procédure même de balisage d'un texte implique une structuration hiérarchique. Nous pouvons émettre l'hypothèse selon laquelle la hiérarchie capturée dans une structure arborescente [42] correspond à un des niveaux d'analyse possible pour interpréter les données d'un document. Par exemple, pour un discours argumentatif, il peut s'agir dans un premier cas d'une analyse de la structuration logique du document, et dans un second d'une analyse des formes rhétoriques employées. Gonet et Tompa [46] ont proposé en 1987 de modéliser un texte annoté par une grammaire qui définit l'organisation des termes d'annotation. Ainsi, le document annoté est représentable par l'arbre de dérivation de cette grammaire (les nœuds non terminaux correspondent aux balises et les fragments de texte aux feuilles). A l'époque, leur cas d'étude consiste en la création d'une version électronique du Oxford English Dictionary. Ils remarquent que l'approche des bases de données relationnelles si elle convient bien aux données d'entreprise,

n'est pas adaptée aux données principalement textuelles. Ils proposent l'exemple d'une référence bibliographique. Une modélisation relationnelle pourra parfaitement encoder les relations entre l'auteur, le titre, la source, etc. Mais elle ne conservera pas la forme textuelle de l'entrée bibliographique elle-même. C'est pourquoi ils proposent – approche innovante à l'époque – d'associer une grammaire au texte libre et développent des opérateurs pour la mise à jour du texte et de la grammaire ainsi que des opérateurs de requête. Cette première approche a été suivie de vives discussions au sein des disciplines que recouvre le terme d'"Humanités". En effet, la structuration hiérarchique n'est en rien inhérente à la forme textuelle. Si les premières approches de modélisation des données textuelles imposent une telle structure arborescente c'est que la création d'une grammaire pour un langage de représentation de ces structures est assez facile. De nombreux contre-exemples qui appellent une modélisation basée sur les graphes (afin d'autoriser les chevauchements d'éléments structuraux) ont rapidement été trouvés [83].

Plusieurs hiérarchies indépendantes Nous nous intéressons ici aux situations de modélisation documentaire pour lesquelles une unique structuration arborescente ne convient pas. Il s'agit en particulier des cas où plusieurs niveaux d'analyse sont requis et répondent à différents usages d'un même texte. Les documents dont il est ici question seront donc modélisés au moyen de plusieurs structures hiérarchiques. Cette contrainte est loin d'être neuve. En effet, déjà avec SGML, un document était le plus souvent considéré posséder deux structures différentes : l'une logique, l'autre physique. Cependant, dans ce dernier cas, il s'agissait le plus souvent de générer automatiquement la structure physique à partir de la structure logique (au moyen d'un programme de transformation, une feuille de styles DSSSL ou XSLT). Dans le cas général qui nous intéresse, d'autres types de structures peuvent être associées à un document : structure sémantique [68, 82], structure temporelle ou spatiale [22, 19], etc. Ces structures sont le plus souvent mutuellement indépendantes et sont manipulées séparément.

Plusieurs hiérarchies dépendantes Par ailleurs, il existe des applications qui font un usage intensif des documents semi-structurés et dont les utilisateurs demandent à pouvoir associer *simultanément* plusieurs structures d'un même texte. C'est-à-dire qu'il faut parfois gérer des relations entre différentes structurations hiérarchiques d'un même texte. Ainsi, il est nécessaire de pouvoir formellement représenter plusieurs hiérarchies dont des éléments peuvent éventuellement se chevaucher [41]. D'ailleurs, à l'origine même du formalisme SGML, ce besoin avait été reconnu par le biais de l'option CONCUR (nous y reviendrons). Le présent état de l'art se place précisément dans ce contexte et cherche à analyser les solutions existantes pour la représentation de documents dont certains termes d'annotation peuvent se chevaucher. Une telle étude nous semble nécessaire car, récemment, de nombreux travaux [102] ont proposé des solutions différentes à ce problème. Aussi, nous prenons pour point de départ un

premier état de l'art déjà publié [18]. Par ailleurs, dans le cadre de notre inscription au sein du projet d'édition électronique des manuscrits de Jean-Toussaint Desanti, nous avons rencontré à plusieurs reprises des situations qui introduisent ce même problème – qu'il faut bien qualifier de syntaxique – de chevauchement de termes d'annotations.

Les deux axes de l'analyse Nous nous intéresserons aux différentes solutions existantes selon deux axes principaux d'analyse :

- La représentation formelle de plusieurs hiérarchies concurrentes qui décrivent le même document. En effet, nous ne faisons cas que des contextes où chaque structure représente un point de vue sur le texte indépendamment des autres structures. En l'absence de relations de dépendances fortes entre les structures, il n'y a donc aucune raison de chercher à les regrouper sous une même hiérarchie (c'est d'ailleurs le plus souvent impossible en raison des chevauchements entre termes d'annotation).
- L'interrogation d'un document multistructuré qui doit permettre de prendre en compte à la fois son contenu et éventuellement plusieurs de ses structures. Il s'agit donc ici de rendre compte, quand il existe, de la puissance d'expression d'un langage de requête.

Domaine de l'étude Nous nous engageons dans la présentation d'un état de l'art des principaux modèles pour la représentation et l'interrogation de documents multistructurés. Nous considérons seulement les structures qui viennent décrire le *texte* d'un document. D'autres modalités et médias pourraient être pris en compte (flux sonore, vidéo, partitions d'images, etc.), ils introduiraient cependant des problèmes de synchronisation entre structures associées à des médias différents. Or nous n'avons pas souhaité analyser ces difficultés supplémentaires dans le cadre de notre état de l'art car, premièrement elles n'ont encore que très peu été étudiées et deuxièmement nous ne les avons pas en pratique rencontrées dans le cadre de nos expérimentations.

Conditions de l'étude Nous systématisons notre étude à l'aide d'une liste de dix dimensions selon lesquelles seront analysées toutes les solutions. Par ailleurs, nous répartirons ces mêmes solutions selon trois catégories. De plus, nous avons choisi d'illustrer chaque solution à l'aide d'un exemple provenant des archives Jean-Toussaint Desanti. Il s'agit d'un petit fragment d'un cahier manuscrit. En effet, c'est dans le cadre d'un travail mené avec des chercheurs de l'Institut Desanti¹ que nous avons été sensibilisés *en pratique* aux problématiques auxquelles les travaux que nous nous apprêtons à décrire tentent d'apporter une solution.

Organisation de l'étude Nous organisons notre étude en quatre parties. La section 2.2 décrit la problématique au moyen du cas d'étude extrait des archives J.-T. Desanti. La section 2.3 présente et définit les dimensions de notre analyse.

1. <http://institutdesanti.ens-lyon.fr/>

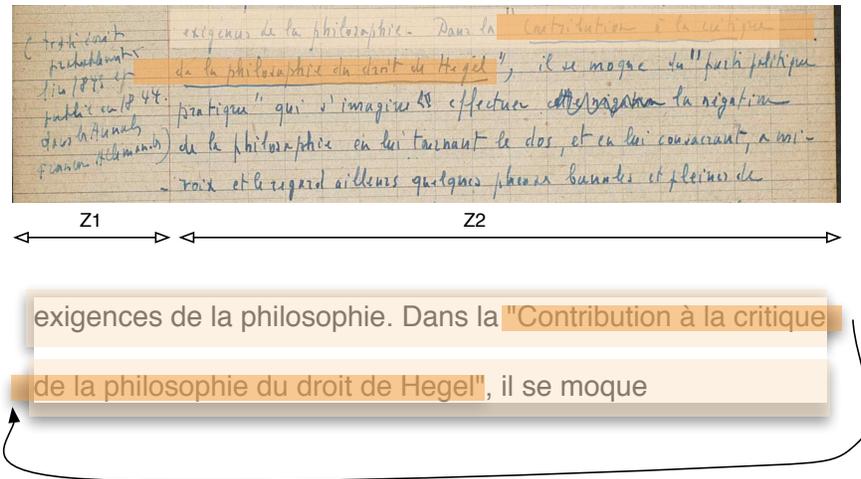


FIGURE 2.1 – Extrait d'un manuscrit de Jean-Toussaint Desanti

La section 2.4 divise les travaux existant en trois catégories et décrit pour chacune d'entre elles une sélection de contributions parmi les plus représentatives. Enfin, la section 2.5 conclut et ouvre sur notre propre manière d'approcher les documents multistructurés; nous nous focaliserons en effet sur un aspect très peu exploré : leur processus de construction.

2.2 Illustration de la problématique

Importance de la TEI Au sein des disciplines couramment désignées au moyen du terme "Humanités" (philosophie, histoire, philologie, littérature, etc.), le besoin est très tôt apparu de structurer de manières multiples un même texte (qu'il soit historique, littéraire, poétique, etc.). La TEI² a été (et est) le premier porte-parole de ce besoin. Elle propose un ensemble de recommandations (les "Guidelines for electronic Text Encoding and Interchange" qui en sont à la version P5) afin de faciliter l'échange et la circulation de documents électroniques, avant tout au sein de la communauté scientifique. Ces recommandations s'intéressent à toutes les formes de texte écrit ou parlé, quels que soient leur langue, leur période, leur genre ou leur contenu. Elles se donnent pour objectif la proposition de solutions adaptées pour la représentation, le stockage et l'accès aux documents électroniques.

Choix d'un exemple Ainsi, nous choisissons d'illustrer cet état de l'art au moyen d'un fragment de document manuscrit du philosophe Jean-Toussaint Desanti. Il s'agit d'un extrait où il est fait référence à un texte de Marx qui est

2. Text Encoding Initiative. <http://www.tei-c.org/>

ensuite analysé. Les chercheurs qui travaillent sur l'archive Desanti sont particulièrement intéressés par l'étude des concepts développés dans un texte, sa datation, ses relations avec d'autres documents qui appartiennent ou non à l'archive, etc. Par exemple, il peut s'agir de découvrir qu'un document jusqu'alors inédit est le support d'une étude préparatoire pour une œuvre connue et publiée. Ou encore, les chercheurs de l'Institut peuvent essayer d'établir, preuves textuelles à l'appui, l'évolution d'un concept créé (ou transformé, ou simplement employé) par Desanti.

Description de l'exemple La figure 2.1 présente l'image du fragment de manuscrit qui nous servira d'exemple. Le texte apparaît sur le morceau d'une page d'un cahier petit format. Deux zones sont physiquement distinguables : Z1 correspond au texte en marge du cahier, et Z2 au texte du corps de la page du cahier. De plus, et pour poursuivre cette description physique, le texte de Z2 est rédigé sur cinq lignes.

Pour plus de clarté, voici la transcription du texte de Z1 :

```
(texte écrit  
probablement  
fin 1843 et  
publié en 1844  
dans les Annales  
Franco-Allemandes)
```

Ensuite, voici la transcription du texte de Z2 :

```
exigences de la philosophie. Dans la "Contribution à la critique  
de la philosophie du droit de Hegel", il se moque du "parti politique  
pratique" qui s'imagine "effectuer cette négation la négation  
de la philosophie en lui tournant le dos, et en lui consacrant, à mi-  
voix et le regard ailleurs quelques phrases banales et pleines de
```

Structure physique Ainsi, cette structure physique forme une première arborescence qui peut être encodée sous toute forme isomorphe à celle ci-dessous :

```
<page>  
<marge>  
  <ligne>(texte écrit</ligne>  
  <ligne>probablement</ligne>  
  <ligne>fin 1843 et</ligne>  
  <ligne>publié en 1844</ligne>  
  <ligne>dans les Annales</ligne>  
  <ligne>Franco-Allemandes)</ligne>  
</marge>  
<corps>  
  <ligne>exigences de ... la "Contribution à la critique</ligne>
```

```

    <ligne>de la philosophie du droit de Hegel", ...</ligne>
    <ligne>pratique" qui s' imagine "effectuer ...</ligne>
    <ligne>de la philosophie en lui tournant le dos, ...</ligne>
    <ligne>voix et le regard ailleurs quelques phrases ...</ligne>
  </corps>
</page>

```

Structure des références En plus de cette première structure qui peut être qualifiée de physique, apparaît clairement une structure des références. Nous trouvons le titre d'un texte de Marx, des citations extraites de ce texte, ainsi que le nom du journal où il fût publié. Cette structure des références forme une seconde arborescence qui peut être encodée sous toute forme isomorphe à celle ci-dessous :

```

<references>
(texte écrit
probablement
fin 1843 et
publié en 1844
dans les <titre_journal>Annales
Franco-Allemandes</titre_journal>)
exigences de ... <titre_oeuvre>"Contribution à la critique
de la philosophie ..."</titre_oeuvre>, ... du <citation>"parti ...
pratique"</citation> qui s' imagine <citation>"effectuer ...
de la philosophie en lui tournant le dos, et en lui consacrant, à mi-
voix et le regard ailleurs quelques phrases banales et pleines de
</references>

```

Structure des altérations Une troisième structure correspond aux altérations du texte (raturé, barré, souligné, etc.). Cette structure des altérations forme une troisième arborescence qui peut être encodée sous toute forme isomorphe à celle ci-dessous :

```

<alterations>
(texte écrit
probablement
fin 1843 et
publié en 1844
dans les Annales
Franco-Allemandes)
exigences de la philosophie. Dans la <souligne>"Contribution ...
de la philosophie du droit de Hegel"</souligne>, ...
pratique" ... <rature>cette négation</rature> la négation
de la philosophie en lui tournant le dos, et en lui consacrant, à mi-
voix et le regard ailleurs quelques phrases banales et pleines de
</alterations>

```

Structure des annotations Nous pouvons facilement distinguer deux niveaux d'énonciation dans le texte : le texte en marge commente un fragment du texte de la page. Nous parlerons génériquement d'une annotation : un fragment de texte annote un autre fragment. Il s'agit d'une structure sémantique qui est souvent rencontrée. Elle forme une quatrième arborescence qui peut être encodée sous toute forme isomorphe à celle ci-dessous :

```
<annotations>
<annotation>
<annotation_content>
(texte écrit
probablement
fin 1843 et
publié en 1844
dans les annales
franco-allemandes)
</annotation_content>
```

```
exigences de la ... <annotation_target>"contribution à la critique
de la ..."</annotation_target></annotation>, il se moque ...
pratique" qui s'imagine "effectuer cette négation la négation
de la philosophie en lui tournant le dos, et en lui consacrant, à mi-
voix et le regard ailleurs quelques phrases banales et pleines de
</annotations>
```

Structure syntaxique Enfin, nous considérons une structure syntaxique simplifiée qui distingue les phrases et les mots. Elle forme une cinquième arborescence qui peut être encodée sous toute forme isomorphe à celle ci-dessous :

```
</phrases>
<p>( <w>texte</w> <w>écrit</w>
<w>probablement</w>
<w>fin</w> <w>1843</w> <w>et</w>
<w>publié</w> <w>en</w> <w>1844</w>
<w>dans</w> <w>les</w> <w>annales</w>
<w>franco-allemandes</w>)</p>
```

```
Etc. Etc.
</phrases>
```

Résumé des structures Les cinq structures sont indépendantes. Chacune annote le texte différemment des autres. Soit l'exemple du fragment de texte "contribution à". Il appartient à la structure syntaxique en tant qu'étant constitué de deux mots qui appartiennent à une phrase. Il appartient à la structure des annotations en tant qu'étant la cible d'une annotation. Il appartient à la structure des altérations en tant qu'étant souligné. Il appartient à la structure

des références en tant que partie du titre d'une œuvre. Enfin, il appartient à la structure physique en tant qu'apparaissant sur une ligne. De plus, notons que le titre de l'œuvre citée chevauche deux lignes.

Utilisation des structures Nous sommes à la recherche de solutions pour la prise en compte simultanée de ces cinq structures. C'est-à-dire que nous avons besoin de modèles qui permettent de les représenter et de les interroger. En particulier, il faudrait pouvoir répondre aux requêtes suivantes :

- Quelles sont les phrases dont une partie a été annotée par Desanti ? (combinaison de la structure des annotations et de la structure syntaxique)
- Quelle est la citation qui utilise le plus de lignes ? (combinaison de la structure des références et de la structure physique)
- Quels sont les mots coupés en fin de ligne ? (combinaison des structures physique et syntaxique)
- Quelles sont les pages qui contiennent des œuvres dont le titre a été annoté ? (combinaison de la structure des références, de la structure des annotations et de la structure physique)

Deux types d'approches Pour répondre à cette problématique, nous distinguons deux types d'approches :

1. Les premières génèrent un unique document multistructuré (noté DMS) composé des différentes structures. Cependant, pour pouvoir utiliser correctement les outils XML classiques (interrogation, transformation, etc.), il faut alors que le DMS soit représentable sous la forme d'une unique hiérarchie. Or, comme nous l'avons vu plus haut, il arrive souvent que des termes d'annotation se chevauchent : c'est la principale difficulté à laquelle doivent se confronter les approches qui appartiennent à cette première catégorie.
2. Les secondes gèrent plusieurs documents, un pour chaque structure. Le principal problème réside alors dans l'interrogation de ces différentes structures concurrentes : comment naviguer d'une hiérarchie à l'autre ?

2.3 Les dimensions de l'analyse

Nous avons sélectionné dix dimensions selon lesquelles seront analysés les principaux travaux qui gèrent des documents multistructurés :

- *Objectif et contexte*. Nous introduirons pour chaque solution ses objectifs et le contexte général dans lequel elle s'inscrit.
- *Types de média*. Nous préciserons si les documents sont uniquement textuels ou s'ils peuvent contenir des images, des sons, des vidéos, etc.
- *Modèles de données*. Nous indiquerons les modèles logiques (forme d'arbre, forme de graphe, etc.) et physiques de données sur lesquels se fondent les diverses solutions.

- *Privilège accordée à une structure* Pour les cas où une structure est privilégiée, que ce soit au niveau logique (c'est-à-dire par l'utilisateur) ou au niveau physique (par le système), nous expliquerons comment les autres structures en dépendent.
- *Validation des DMS* Certaines solutions peuvent proposer une manière de schéma pour valider un document multistructuré en tenant compte de l'ensemble de ses hiérarchies [33].
- *Opérateurs de requête* Nous déterminerons si l'utilisateur peut formuler des requêtes qui mettent en jeu le contenu ainsi que plusieurs structures. De plus nous étudierons les cas où des opérateurs spécifiques ont été définis pour rassembler des structures. Enfin, nous analyserons, s'ils existent, les opérateurs développés spécifiquement pour l'interrogation des documents multistructurés.
- *Gestion des modifications des DMS* Nous déterminerons s'il est possible de mettre à jour les données et/ou les structures d'un DMS.
- *Compatibilité avec XML* Nous indiquerons si une syntaxe XML est associée au modèle physique de données. De plus, nous déterminerons si les outils standards de la galaxie XML (XPath, XQuery, XSLT, etc.) restent utilisables et s'il a été nécessaire de les étendre afin qu'ils puissent prendre en compte les DMS.
- *Implémentation* Les éventuelles implémentations seront mentionnées et nous essaierons d'analyser leur qualité.

2.4 Un état de l'art des principales approches

Introduction Cette section propose un état de l'art des différentes approches qui permettent la description et l'interrogation des documents multistructurés. Nous les analyserons selon les dimensions qui viennent d'être introduites. De plus, nous les illustrerons à l'aide de l'exemple introduit plus haut et qui a été extrait des archives du philosophe Jean-Toussaint Desanti.

Trois catégories Nous choisissons de classer les propositions selon trois catégories. La première comprend les premiers travaux à s'être intéressés à la représentation de plusieurs structures hiérarchiques concurrentes au sein d'un même texte (cf. section 2.4.1) — il s'agit le plus souvent de solutions uniquement syntaxiques. La seconde correspond aux modèles propriétaires (cf. section 2.4.2). La dernière comprend les contributions qui restent compatibles avec le formalisme XML et ses divers outils (cf. section 2.4.3).

2.4.1 Solutions historiques

Introduction Comme nous l'avons déjà rapidement esquissé plus haut, le besoin d'intégrer des balises définies dans différentes grammaires – par exemple exprimées dans le formalisme des DTD – au sein d'un unique document, a très vite été ressenti. L'option CONCUR [45] du standard SGML a tenté de répondre

à ce besoin. Cependant, et malgré la qualité des réflexions qui les ont accompagnées, les propositions introduites par SGML n'ont que très rarement été implémentées. Leurs spécifications étaient en effet bien trop complexes. Ainsi, l'idée que portait l'option CONCUR n'a pas été retenue par le standard XML. C'est, entre autres raisons, pourquoi la TEI [92] a, elle aussi, proposé un ensemble de bonnes pratiques pour l'encodage de documents électroniques multistrukturés. Cependant, ces dernières solutions sont le plus souvent seulement syntaxiques. Nous avons choisi d'introduire la solution CONCUR (pour son importance historique) ainsi que la technique des "milestones" proposée par la TEI (pour un exemple de solution purement syntaxique).

L'option CONCUR du standard SGML

Les objectifs de CONCUR CONCUR sert à la représentation de documents textuels annotés au moyen de termes qui appartiennent à plusieurs structures hiérarchiques. Il s'agit uniquement de la définition d'une syntaxe qui permet de représenter de tels documents dans un simple fichier texte. Ainsi, aucun modèle logique n'est défini. Ci-dessous, nous proposons un extrait de notre exemple manuscrit, où les deux structures physique et référentielle ont été encodées ensemble. Nous remarquons que chaque balise se trouve simplement préfixée par un label qui identifie sa structure d'appartenance.

```
<!DOCTYPE S1 SYSTEM "dtds/physical.dtd">
<!DOCTYPE S2 SYSTEM "dtds/references.dtd">
<(S2)references>
<(S1)page>
  <(S1)marge>
    <(S1)ligne>(texte écrit</(S1)ligne>
    <(S1)ligne>probablement</(S1)ligne>
    <(S1)ligne>fin 1843 et</(S1)ligne>
    <(S1)ligne>publié en 1844</(S1)ligne>
    <(S1)ligne>dans les <(S2)titre_journal>Annales</(S1)ligne>
    <(S1)ligne>Franco-Allemandes</(S2)titre_journal></(S1)ligne>
  </(S1)marge>
  <(S1)corps>
    <(S1)ligne>exigences ... <(S2)titre_oeuvre>"Contribution ...
    ...
  </(S1)corps>
</(S1)page>
</(S2)references>
```

Relations entre structures Remarquons qu'il n'existe pas de structure privilégiée par rapport aux autres. Chaque structure est définie dans une grammaire séparée (DTD). Ainsi, il n'y a pas de DTD pour le document multistrukturé pris dans son ensemble. C'est pourquoi les relations entre structures ne peuvent pas être modélisées. D'où, par ailleurs, l'impossibilité de valider un tel document.

Requêtes Aucun opérateur spécifique pour la prise en compte simultanée des structures concurrentes et de leur contenu n'a été défini.

Mise à jour des structures et du contenu La mise à jour d'un fragment de texte ou d'une structure est possible. Cependant, c'est à l'utilisateur de maintenir la cohérence de son document. Il est important de noter qu'à l'époque où cette solution est apparue, les documents qui étaient utilisés dans le champ disciplinaire des Humanités n'étaient que très rarement modifiés. C'est d'ailleurs une constante de la plupart des approches que nous analyserons dans cet état de l'art que de ne pas prendre sérieusement en compte les possibilités de modifications des documents. Ainsi, notons en passant que notre apport personnel consistera en une réflexion sur la *construction* des documents multistructurés.

Compatibilité XML Finalement, CONCUR n'est pas compatible avec XML puisque le document final contient des chevauchements (voir sur l'exemple précédent le titre de l'œuvre qui chevauche deux lignes). De toute façon, XML n'existait pas à l'époque de la spécification de cette option ! De plus, l'option CONCUR n'a jamais été implémentée (sans doute à cause de sa trop grande complexité) ...

Conclusion CONCUR est historiquement la première solution proposée au problème de la représentation de plusieurs structures hiérarchiques éventuellement concurrentes dans un même document. Mais c'est une solution propriétaire et seulement syntaxique.

Milestones (TEI)

Introduction Dans le contexte de la TEI, plusieurs solutions syntaxiques pour la représentation de plusieurs hiérarchies de termes d'annotation au sein d'un même texte ont été proposées (comme l'usage des milestones ou de la fragmentation des éléments, etc.). Elles consistent en un ensemble de bonnes pratiques recommandées par la TEI. Elles ne forment pas un standard mais sont cependant suffisamment générales pour pouvoir être appliquées dans différents contextes. Ces solutions sont relativement similaires et nous choisissons de seulement illustrer l'usage des "milestones".

Quelques éléments caractéristiques La syntaxe des milestones est exprimée dans le formalisme XML. Elle préconise l'usage d'éléments vides pour modéliser les chevauchements de termes d'annotation. Ainsi, nous pouvons considérer que le modèle logique des milestones est le modèle de données XML [42]. Quant à son modèle physique, il s'agit du texte annoté dans la syntaxe XML. Ce dernier peut donc être stocké dans n'importe lequel des entrepôts de données XML existants. Par ailleurs, bien qu'aucune structure ne soit privilégiée, c'est à l'utilisateur de choisir les éléments qui seront représentés au moyen de milestones. Comme visible ci-dessous, nous choisissons de représenter la structure

physique des lignes au moyen de milestones. Ainsi, nous réglons syntaxiquement le problème du chevauchement avec les éléments annotés qui appartiennent à la hiérarchie des références. Remarquons que le choix aurait pu être fait d'utiliser des milestones pour les éléments de la structure des références plutôt que pour ceux de la structure physique. A ce sujet, rien ne semble pouvoir raisonnablement et génériquement justifier le choix d'un utilisateur. Par ailleurs, il est évident qu'avec une telle solution, les structures n'existent pas en elles-mêmes (en particulier, seule une moitié arbitraire des éléments qui "posent un problème de chevauchement" a besoin d'être traitée).

```

<references>
<page>
  <marge>
    <Dlign xml:id="L1"/>(texte écrit<Fligne corresp="#L1"/>
    <Dlign xml:id="L2"/>probablement<Fligne corresp="#L2"/>
    <Dlign xml:id="L3"/>fin 1843 et<Fligne corresp="#L3"/>
    <Dlign xml:id="L4"/>publié en 1844<Fligne corresp="#L4"/>
    <Dlign xml:id="L5"/>dans <titre_journal>Annales<Fligne corresp="#L5"/>
    <Dlign xml:id="L6"/>Franco...</titre_journal><Fligne corresp="#L6"/>
  </marge>
</corps>
  ...
</corps>
</page>
</references>

```

Validation A chaque structure pourrait être associée une DTD mais aucune DTD ne peut être associée au document dans son entier. Ainsi, lorsqu'il est encodé par des milestones, il est impossible de vérifier la validité d'un document multistructuré au moyen des outils XML classiques.

Interrogation Aucun opérateur spécifique pour la manipulation de documents multistructurés modélisés par des milestones n'a été proposé.

Modifications Cette technique produit un unique document multistructuré. Ainsi, la mise à jour du contenu textuel ne pose pas de problème. Cependant, de nombreuses difficultés peuvent apparaître lors de la mise à jour des structures. En effet, la structure globale du document peut rester cohérente tandis que les parties qui font appel aux milestones peuvent perdre leur structure hiérarchique (dans l'exemple précédent, il suffit de supprimer une balise Fligne sans supprimer la balise Dlign correspondante ...). C'est pourquoi cette approche n'est utilisée que dans les cas où les données structurées sont très peu mises à jour.

Compatibilité XML Bien que la syntaxe utilisée soit celle du formalisme XML, les outils traditionnels qui attendent une structuration arborescente (XPath,

XQuery, etc.) ne pourront pas être convenablement utilisés à cause des parties du document qui sont encodées par des milestones. Par contre, remarquons qu'un utilisateur n'aura pas de peine à comprendre et à éditer les sources d'un tel document multistructuré.

Conclusion En résumé, la TEI propose différentes solutions syntaxiques au problème de la représentation de plusieurs structures hiérarchiques concurrentes de termes d'annotation dans un même document. Elles sont "compatibles XML" en tant qu'elles en empruntent la syntaxe, mais l'exploitation des différentes structures hiérarchiques d'un document XML encodé au moyen de ces techniques n'est pas (ou peu) possible par le biais des outils standardisés (XPath, XQuery, XSLT, etc.).

2.4.2 Les modèles propriétaires

Introduction Cette seconde catégorie comprend quatre propositions : les graphes d'annotation [9] qui sont apparus dans le domaine de la linguistique, le modèle MSDM [1] à l'origine défini par le groupe de travail "Multistructures" de l'Institut des Sciences du Document Numérique (ISDN) de la région Rhône-Alpes, LMNL [98], un nouveau langage balisé, et finalement MVDM [37] [38], un modèle pour la représentation des différentes structures d'un document conçu pour autoriser la recherche d'information au moyen d'une approche de classification adaptée au domaine des documents multistructurés. Toutes ces contributions sont basées sur un modèle de graphe.

Les graphes d'annotation

Contexte Les graphes d'annotations ont été créés dans le contexte de recherches en linguistique. Ils se présentent comme un formalisme capable de prendre en compte simultanément l'ensemble des phénomènes linguistiques (phonétique, prosodie, morphologie, syntaxe, pragmatique, etc.) afin de proposer des outils et des ressources spécifiques pour l'analyse de la langue. Ainsi, les graphes d'annotation permettent de représenter les multiples niveaux d'annotations nécessaires à l'étude d'un corpus audio. Le même contenu primitif peut être annoté par différents sous-ensembles d'arcs (au sens des arcs d'un graphe) qui correspondent aux différentes structures d'annotation. Ainsi, un sous-ensemble du graphe correspond à un niveau d'analyse linguistique. Les phénomènes linguistiques fondamentaux sont supposés pouvoir être représentés sous la forme d'un ensemble discret et ordonné d'instantanés qui constituent les nœuds du graphe. Finalement, les données sont discrétisées en fragments élémentaires semblables et ordonnées en fonction du temps.

Application aux données textuelles Les graphes d'annotations peuvent facilement être adaptés aux données textuelles, comme le montre la figure 2.2. L'unité de fragmentation élémentaire des données devient le caractère. Ainsi, le modèle logique de données est un graphe dont les arcs sont décorés par un

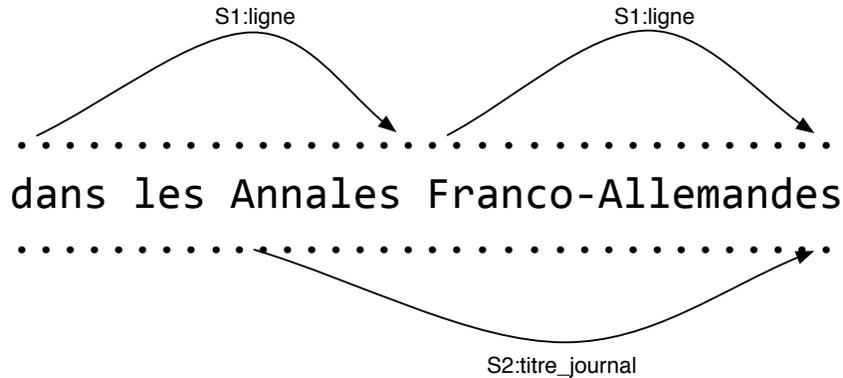


FIGURE 2.2 – Illustration des graphes d’annotation

terme d’annotation préfixé de l’intitulé de la structure à laquelle il appartient (par exemple, `S2:titre_journal`) et les noms sont une unité élémentaire de texte (dans le cas le plus générique il s’agit du caractère mais, parfois le mot peut suffire). Puisque les nœuds d’un graphe peuvent avoir plusieurs parents (un arbre peut être défini comme un graphe dont les nœuds fils n’ont jamais qu’un seul nœud père), le problème du chevauchement local de termes d’annotation ne se pose pas. Par ailleurs, un graphe peut être traduit en un document XML en faisant usage des mécanismes de référence (grâce aux attributs `ID` et `IDREF`). Dans ce dernier cas, le modèle physique est un document balisé qui peut être utilisé pour l’échange des graphes d’annotations.

Organisation et validation des structures Les graphes d’annotations, puisqu’ils présentent un modèle générique de graphe, n’accordent de privilège à aucune des structures hiérarchiques. Par ailleurs, aucun mécanisme de schéma n’est proposé pour la validation des graphes d’annotations.

Interrogation Il n’existe pas de langage de requête qui permette d’interroger un graphe d’annotations. Cependant, quelques extensions à XPath ont été proposées pour permettre de travailler sur la version sérialisée en XML d’un graphe d’annotations [7, 8]. Ces dernières nous ont semblé très spécifiques au domaine de la linguistique.

Mise à jour des structures et des données Puisque les structures partagent les mêmes données et appartiennent à un même graphe, leur mise à jour ne pose aucun problème. Par ailleurs, le seul contexte applicatif connu pour ce modèle étant l’annotation linguistique de sources audio, la mise à jour des données n’est jamais utilisée car ces dernières sont toujours connues a priori.

Compatibilité XML Bien qu'un graphe d'annotations puisse se présenter comme un document XML, l'usage intensif des références que cette serialisation implique empêche le plus souvent l'utilisation des outils XML classiques. En fait, pour les besoins de la serialisation, une structure est privilégiée et peut faire l'objet des traitements XML traditionnels, mais non les autres.

Implémentation Atlas [61] est une implémentation des graphes d'annotations. Par ailleurs, un essai de stockage des graphes d'annotations dans un SGBD relationnel a été initié [6].

Conclusion En résumé, les graphes d'annotations sont un modèle propriétaire générique pour la représentation de plusieurs structures qui décrivent un même contenu. Ce dernier doit être coupable en une séquence d'éléments unités (caractères, secondes d'enregistrement audio, etc.). Enfin, et bien qu'une serialisation XML soit possible, les outils standards de la galaxie XML ne sont en général pas utilisables sur les graphes d'annotations.

Le modèle MSDM

Introduction MSDM est un modèle pour la représentation de documents structurés et permet l'intégration d'un ensemble de structures distinctes au sein d'un même document. De plus, dans le cadre de ce modèle, des opérations ont été précisément définies qui permettent de travailler simultanément sur plusieurs structures concurrentes (c'est-à-dire, en particulier, que des éléments de ces structures peuvent se chevaucher). Il est intéressant de remarquer que cette approche se base sur une étude détaillée des pratiques documentaires et de leurs transformations. Il s'agissait principalement d'une analyse de l'impact des technologies hypermedia (c'est-à-dire surtout la possibilité de créer toujours plus de liens typés dans un environnement ouvert, ce qu'aujourd'hui recouvre le terme de "web des données") sur l'analyse et la gestion des documents. Notons de plus que les données sur lesquelles s'appliquent les structures peuvent être multimedia, l'unique contrainte étant, comme nous le verrons, qu'elles soient découposables séquentiellement (c'est-à-dire selon une unique dimension, que celle-ci soit temporelle, spatiale, etc.).

Modèles logique et physique Comme nous le voyons sur la figure 2.3, le modèle logique de données est un graphe. Ce qui est appelée "structure de base" est un partitionnement (au sens ensembliste, c'est-à-dire une familles de fragments mutuellement exclusifs) optimal du contenu encore non annoté du document afin de permettre le partage de ces fragments par les différentes structures documentaires qui viennent se greffer sur la structure de base. Des arcs relient les éléments des structures documentaires à la structure de base. Par exemple, un arc relie un élément **ligne** de la structure physique à un ensemble de fragments de la structure de base dont la réunion représente cette ligne. De plus, des arcs peuvent être créés entre éléments de structures documentaires différentes. Le

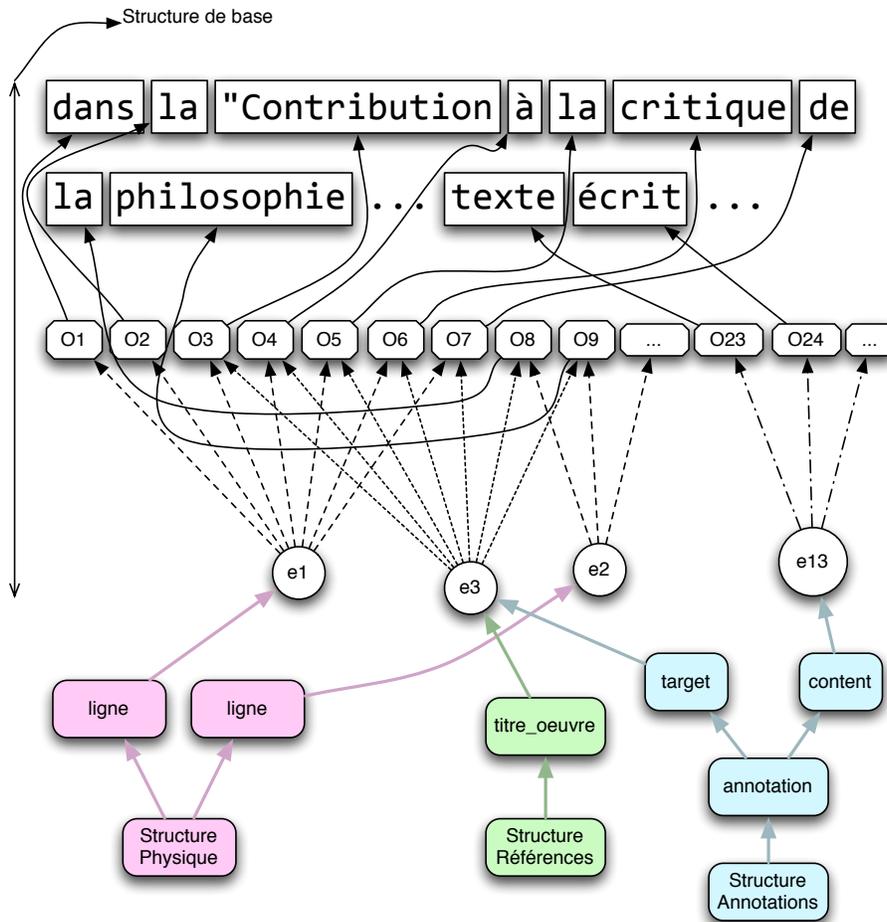


FIGURE 2.3 – Illustration du modèle MSDM

modèle MSDM est seulement logique et n'impose aucun modèle physique. Cependant, nous verrons qu'un modèle physique qui utilise le formalisme XML a été implémenté sous le nom de MultiX [25].

Organisation des structures La structure de base peut être considérée comme une structure principale privilégiée sur laquelle viennent se greffer les autres structures. Cependant, c'est une structure qui ne possède aucune représentation qui serait visible par un utilisateur. En ce sens, parmi les structures documentaires que manipule l'utilisateur, aucune n'est privilégiée.

Validation des documents multistrués Il n'existe aucun mécanisme de schéma qui permettrait la validation d'un document multistrué instance du modèle MSDM.

Mise à jour d'un document multistrué Le modèle se voit accompagné d'un ensemble d'opérations pour éditer des documents multistrués ainsi que pour fusionner et diviser des structures. Les opérations d'édition incluent l'ajout et la suppression de nœuds dans les structures documentaires, d'arcs entre une structure documentaire et la structure de base, et enfin d'arcs entre structures documentaires. L'opérateur de fusion permet de construire une unique structure documentaire à partir de plusieurs structures. L'opérateur de division est l'exact inverse de celui de fusion et permet de construire plusieurs structures documentaires à partir d'une structure initiale. Cependant, toutes les opérations de mise à jour des données ou des structures impliquent une reconstruction totale de la structure de base, ce qui représente un calcul relativement coûteux.

Compatibilité XML MultiX [25] est une instance dans le formalisme XML du modèle MSDM. Elle ne propose pas de nouveaux chemins XPath mais un ensemble de fonctions récursives XQuery qui permettent l'expression de requêtes prenant simultanément en compte les différentes structures. Par exemple, la requête : "Quels sont les titres d'œuvres qui chevauchent deux lignes?" qui fait appel aux structures physique et des références, pourrait être exprimée ainsi :

```
let $doc := doc("manuscript.xml")
for $l in $doc//msd:DS[@name = "physique"]//line,
    $t in $doc//msd:DS[@name = "references"]//titre_oeuvre
where multix:share-fragments($l, $t)
    and not(multix:include-content-of($l, $t))
return multix:rebuild($t)
```

Afin de comprendre cette requête, il faut savoir, premièrement, que la fonction "share-fragments" vérifie si deux éléments qui appartiennent à des structures documentaires ont au moins un fragment de la structure de base en commun ; deuxièmement, que la fonction "include-content-of(e1,e2)" vérifie si l'en-

semble des fragments de la structure de base qui composent l'élément e1 font aussi partie de la composition de l'élément e2.

Encodage des structures documentaires Finalement, notons qu'un système de gestion de documents multistrués a été développé [24]. Il se fonde sur un parser nommé MXP (Multi-XML Parser) qui permet de construire un document MultiX (c'est-à-dire une instance XML du modèle MSDM) à partir d'un ensemble de documents XML définis séparément pour chaque structure.

Conclusion En résumé, MSDM propose une solution pour la représentation et l'interrogation de documents multistrués textuels. Il s'agit d'un modèle propriétaire. Des requêtes complexes qui adressent simultanément différentes structures sont rendues possibles grâce à un jeu de fonctions XQuery qui gèrent en particulier les chevauchements entre termes d'annotations.

LMNL

Introduction LMNL (Layered Markup Annotation Language) est un langage balisé différent de XML. Sa conception a été dirigée par la nécessité de proposer une réponse convenable au problème du chevauchement des termes d'annotation. Par ailleurs, en tant que langage balisé, il permet, comme XML, d'ajouter des couches d'annotations structurées à un document XML.

La notion de limen ou couche Contrairement à XML, la notion de hiérarchie ne fait pas explicitement partie du modèle LMNL. Par contre, il y a une notion de couche ou limen³. Ainsi, un document LMNL se fonde sur une première couche textuelle. Il s'agit d'une séquence de valeurs atomiques (par exemple, et le plus souvent, des caractères Unicode). De plus, un document peut être adressé au moyen de **régions**. Une région est nommée, elle possède un identifiant, un contenu textuel et une ou plusieurs **annotations**. Chaque annotation est relative à une région, possède un rang qui la positionne parmi l'ensemble des autres annotations relatives à sa région, un nom et une couche textuelle (qui peut elle-même posséder des régions, etc.). Nous illustrons ci-dessous et sur notre exemple habituel la syntaxe proposée par LMNL :

```
[Manuscrit [auteur}Jean-Toussaint Desanti{]}
[Page}
[marge}
  [ligne}(texte écrit{ligne]
  [ligne}probablement{ligne]
  [ligne}fin 1843 et{ligne]
  [ligne}publié en 1844{ligne]
  [ligne}dans les [titre_journal}Annales{ligne]
```

3. Les auteurs de LMNL employaient à l'origine le mot "couche" mais lui préfèrent aujourd'hui celui de limen : <http://www.piez.org/wendell/LMNL/Amsterdam2008/presentation-slides.html#page21>

```

    [ligne]Franco-Allemandes){titre_journal}{ligne]
    {marge]
    [corps}
    ...
{Manuscrit]

```

Remarquons que le chevauchement des termes d'annotation "ligne" et "titre_journal" a pu être exprimé sans difficulté.

Organisation des structures Aucune structure n'est privilégiée.

Validation des documents multistructurés Aucun mécanisme de validation n'a été proposé pour les documents multistructurés qui sont des instances du modèle LMNL. Cependant, les auteurs font référence au projet Creole de J. Tension [97] qui pose les premières briques d'un langage de validation pour les documents multistructurés. Théoriquement, ce langage devrait s'adapter à différents modèles.

Mise à jour des documents multistructurés Comme pour les solutions précédentes, l'existence physique d'un unique document multistructuré permet la mise à jour aussi bien du contenu textuel que des structures. Cependant, c'est à l'utilisateur d'assurer la cohérence de son document.

Compatibilité XML LMNL est doté d'une fonctionnalité d'import/export depuis et vers le formalisme XML. Cependant, lorsque des chevauchements apparaissent, ils sont traduits par des milestones. Ainsi, les problèmes déjà mentionnés dans la section dédiée à la TEI persistent.

Implémentation Une implémentation du modèle LMNL est disponible (voir le site : <http://www.piez.org/wendell/LMNL/lmnl-page.html>).

Conclusion En résumé, LMNL est un modèle et une syntaxe qui se présente comme une alternative à XML. LMNL est principalement concerné par la résolution du problème de chevauchement des termes d'annotation. De plus, le modèle possède un mécanisme récursif d'annotation (une annotation est isomorphe à un document). Cependant, face à XML, l'adoption d'un nouveau standard reste difficile.

MVDM

Contexte MVDM (Multi View Document Model) [37] [38] est un modèle pour la représentation des différentes structures d'un document multimédia.

Un modèle de graphe MVDM décrit un modèle de graphe. Ainsi, les sources documentaires sont fragmentées en nœuds. De plus, les relations (c'est-à-dire les arcs du graphe) sont typées et non nécessairement hiérarchiques.

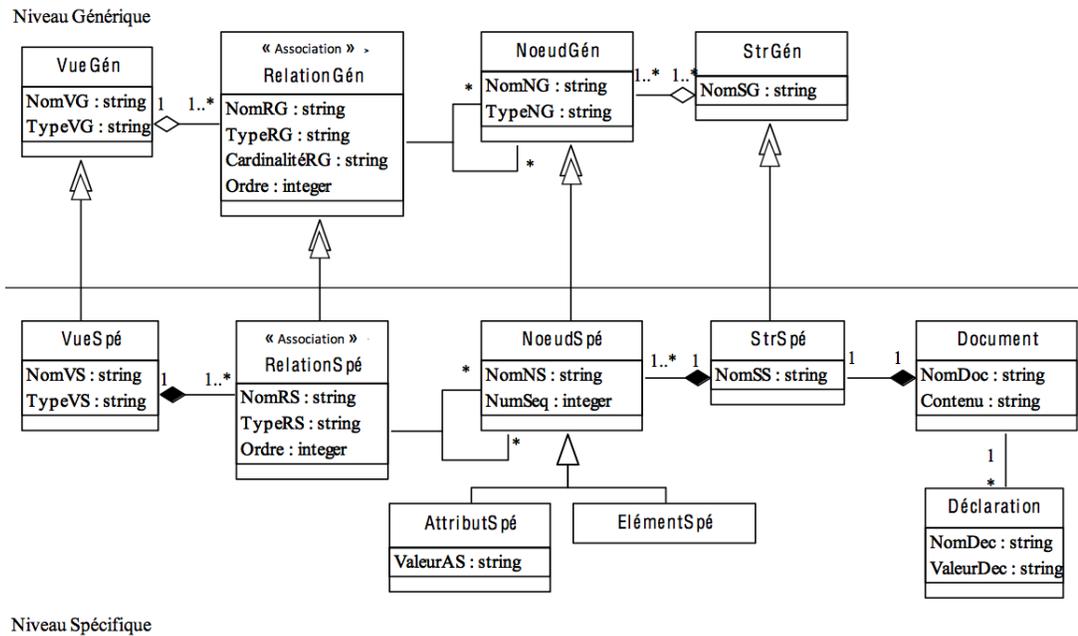


FIGURE 2.4 – Illustration du modèle MVDM

Quelques spécificités MVDM introduit le concept de vue comme sous-ensemble d’instances de relations. Par ailleurs tout nœud peut-être considéré en tant que document. C’est-à-dire qu’un nœud peut être fragmenté en nœuds liés par des relations. Par exemple, au niveau d’un nœud de type “image”, des zones seront parfois distinguées. Finalement, les nœuds font référence au contenu au moyen d’index. Dans le cas de données textuelles, un index correspondra à un couple d’entiers qui représentent un intervalle de caractères Unicode. Dans le cas d’une image, l’index sera par exemple composé des coordonnées d’une zone polygonale. Etc.

Deux niveaux de modélisation Le modèle est organisé selon deux niveaux (voir la figure 2.4). Au niveau spécifique, un nœud (qui peut être de type attribut ou élément) est connectable à d’autres nœuds au moyen d’une relation. Comme nous l’avons dit plus haut, une vue est un ensemble d’instances de relations. Une structure est un ensemble de nœuds (remarquons qu’un nœud peut être partagé par plusieurs structures). Un document est ainsi composé d’un ensemble de structures. Finalement, un objet de type “Déclaration” permet de préciser des méta-données au sujet d’un document (date de création, dernière mise à jour, etc.). De plus, un niveau générique sert à modéliser les caractéristiques qui permettent de regrouper en “classes” des instances du niveau spécifique. Ainsi, cette proposition s’intéresse au problème essentiel de la classification : comment associer une vue spécifique à une vue générique ?

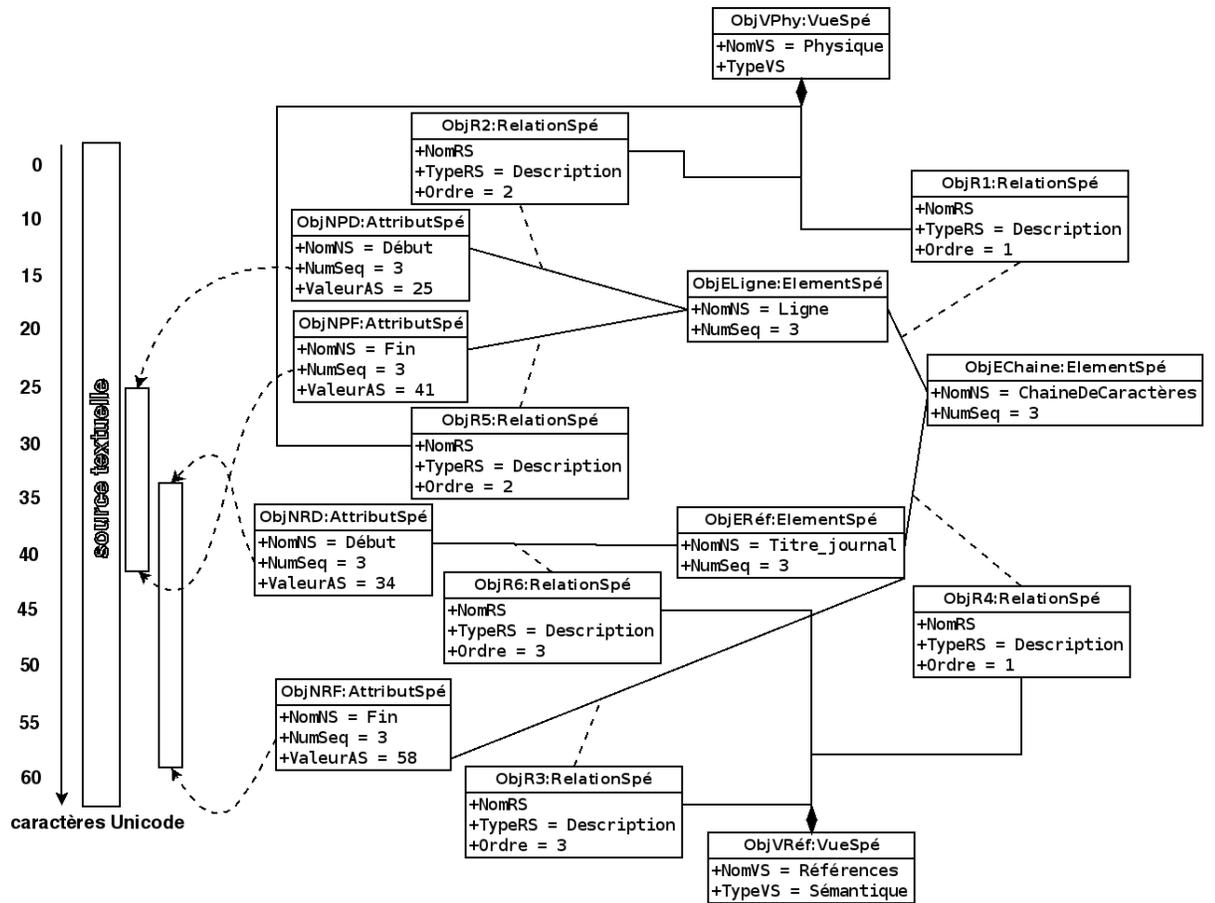


FIGURE 2.5 – Illustration d’une instance du modèle MVDM

Instance du modèle La figure 2.5 présente une instance du modèle MVDM sur une petite partie de notre exemple de document multistructuré. Étant donné qu’il s’agit d’un modèle de graphe, les chevauchements ne posent pas en eux-mêmes de problèmes.

Validation des structures Aucun mécanisme de validation des structures n’est proposé.

Interrogation L’interrogation est prise en compte à deux niveaux : la recherche documentaire et l’analyse multidimensionnelle. Nous ne décrivons que le premier aspect. La recherche de documents est divisée en plusieurs étapes. D’abord, la sélection d’un ensemble de nœuds d’une structure générique. Ensuite, l’attribution de valeurs aux nœuds sélectionnés : il peut s’agir de relations numériques (<, >, etc.) ou ensemblistes sur des chaînes de caractères (inclusion,

union, intersection, etc.). Puis, une requête SQL est générée. Il faut ensuite gérer les chevauchements. En effet, la gestion des conjonctions entre valeurs attribuées aux nœuds de la requête peut impliquer de retourner un fragment de contenu qui est à l'intersection de plusieurs nœuds spécifiques. Ainsi, il ne semble pas possible d'exprimer des requêtes explicites *sur* des événements de chevauchement (par exemple, trouver les citations qui chevauchent deux lignes). Enfin, une visualisation des résultats est présentée.

Compatibilité XML et mise à jour des structures et des données La mise à jour des structures et des données n'est pas envisagée. En pratique, la redondance initiale du contenu documentaire est supposée. Afin de pouvoir construire une instance du modèle MVDM, il faut qu'un document multistructuré soit a priori représenté par un ensemble de documents (mono)structurés. Ces derniers sont des documents XML. Nous ne pouvons donc pas considérer ce modèle comme étant "compatible" avec le formalisme XML.

Implémentation Un prototype nommé MDOCREP (Multistructured Document REpository) a permis d'évaluer le modèle MVDM en utilisant un SGBD relationnel et un client écrit dans le langage de programmation JAVA.

Conclusion En résumé, MVDM est un modèle de représentation des documents multistructurés dont l'objectif premier est de permettre une classification d'un corpus documentaire en exploitant la richesse des différentes structures.

2.4.3 Les modèles "compatibles XML"

Introduction Cette troisième et dernière catégorie comprend neuf modèles. Le premier, les arbres colorés [53], emprunte un modèle mathématique bien connu et l'utilise pour étendre le modèle XML afin de pouvoir gérer simultanément plusieurs hiérarchies de termes d'annotation. Le second est basé sur la notion de "delay nodes" [64] et consiste en l'ajout d'un nouveau type de nœud au modèle XDM (modèle sur lequel reposent XPath et XQuery). Le troisième correspond à un usage particulier des structures de traits ("feature structures") de la TEI [95]. Le quatrième consiste en l'extension d'une base de données XML (MonetDB) [2] afin de permettre le stockage et l'interrogation de documents multistructurés. Le cinquième, MuLaX [48], est une implémentation de l'option CONCUR de SGML mais adaptée au monde XML. Le sixième est une utilisation naïve du formalisme RDF [99] afin de gérer les chevauchements de termes d'annotation. L'antépénultième, EARMARK [73], est une approche basée sur la définition ontologique des concepts qui permettent de définir un langage balisé qui sache gérer les documents multistructurés. Cette définition permet aussi de générer des contraintes qui assurent la validité d'un document. L'avant dernier, les GODDAGs [93, 52], correspond sans doute au premier travail vraiment abouti, aussi bien au niveau du modèle que des primitives qui lui sont associées et permettent de manipuler ses instances. Enfin, MSXD [21], est une solution qui

définit un modèle et un mécanisme de schéma pour les documents multistru-
turés, ce dernier mécanisme peut être considéré comme une première tentative
de construction d'un processus de validation prenant en compte simultanément
plusieurs hiérarchies de termes d'annotation [33].

Les arbres colorés

Introduction Cette proposition se base sur une manière optimale de facto-
riser les données pour pouvoir les décrire au moyen de plusieurs hiérarchies de
termes d'annotations. D'ailleurs, le titre de l'article est : "Colorful XML : One
Hierarchy Isn't Enough"! C'est une solution que nous qualifions de "compatible
XML" car elle consiste principalement en une extension XQuery qui permet de
"colorer" les nœuds d'un arbre XML.

Modèle logique Le modèle logique des arbres colorés est une extension du
modèle de données XML [42]. Il comprend un ensemble fini de couleurs. A
chaque nœud peuvent être associées plusieurs couleurs. L'ensemble des nœuds
d'une même couleur forme une hiérarchie. Par ailleurs, les arbres colorés peuvent
être stockés dans une base de données XML si elle est étendue afin de pouvoir
gérer la coloration des nœuds.

Organisation des structures Il n'y a pas de structure (c'est-à-dire de cou-
leur!) qui serait a priori privilégiée. Par ailleurs, il n'existe aucun mécanisme de
schéma qui permettrait de typer un arbre coloré et ainsi de valider son contenu.

Exemple d'arbre multicolore La figure 2.6 illustre ce modèle sur un extrait
de notre exemple manuscrit de Jean-Toussaint Desanti. L'arbre est bicolore. Une
couleur est associée à la structure physique, l'autre à la structure des références.
Nous remarquons que trois des nœuds unités (ceux qui contiennent les feuilles
textuelles) partagent les deux couleurs.

Opérateurs pour la navigation et les requêtes Les auteurs proposent
une extension aux chemins XPath [27] afin de pouvoir naviguer entre les hié-
rarchies. Il s'agit de l'ajout d'un premier pas au processus de navigation XPath.
Ce pas consiste à choisir la couleur d'un nœud (c'est-à-dire, choisir une struc-
ture). Suivent ensuite les classiques choix d'un axe, évaluation d'une éventuelle
condition, etc. De plus, une extension pour XQuery [11] a aussi été développée
et permet d'ajouter de nouvelles couleurs aux nœuds au cours d'une requête.

**Mise à jour des documents multistru-
turés** Pour ce qui est de la mise
à jour d'un arbre coloré, les auteurs expliquent qu'il serait possible d'étendre le
langage XUpdate⁴ grâce à l'extension qu'ils ont déjà proposée pour XPath.

4. <http://xmldb-org.sourceforge.net/xupdate/>

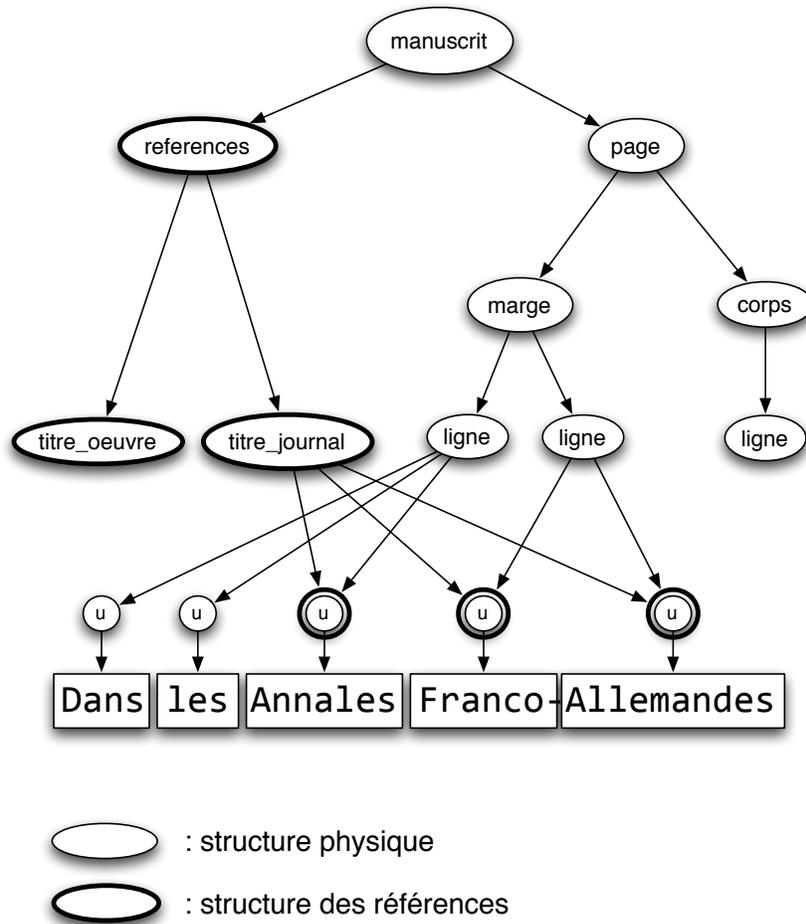
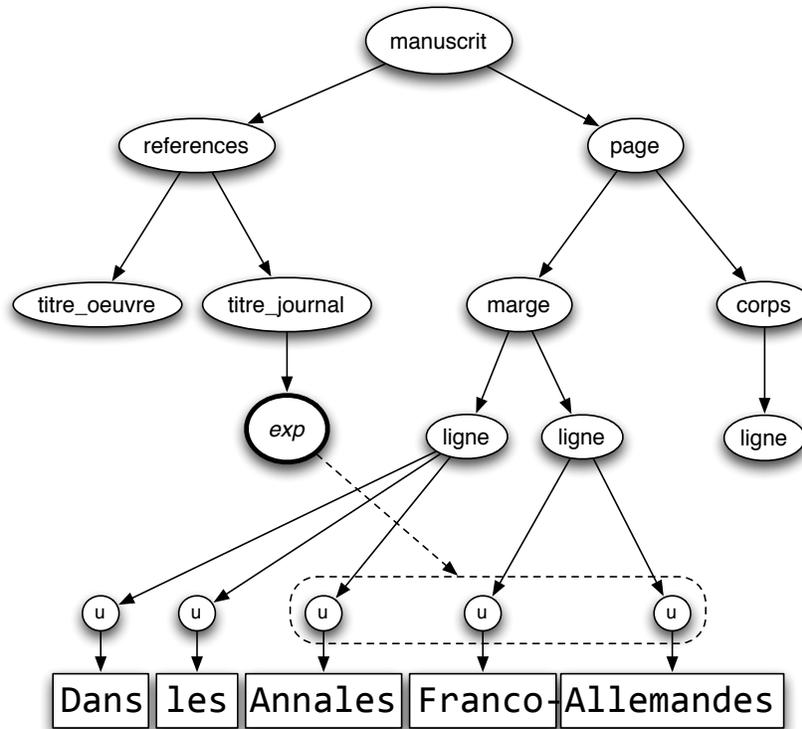


FIGURE 2.6 – Illustration du modèle des arbres colorés



$exp = (../..///u)[position() = 3 \text{ to } 5]$

FIGURE 2.7 – Illustration d’une instance du modèle des "delay nodes"

Implémentation Finalement, une implémentation des arbres colorés a été développée sous la forme d’une extension de la base de données XML Timber⁵.

Conclusion En résumé, cette proposition est applicable aux documents textuels en choisissant, par exemple, pour unité de base le caractère Unicode. Par ailleurs, aussi bien le modèle que le langage de requête sont compatibles XML.

Delay Nodes

Introduction Jacques LeMaître [64] propose d’ajouter un nouveau type de nœud au modèle XDM sur lequel sont basés XPath et XQuery. Ces nœuds représentent potentiellement, sous la forme d’une requête XQuery, certains nœuds enfants de leur nœud parent (voir la figure 2.7).

5. <http://www.eecs.umich.edu/db/timber/>

Le modèle des "delay nodes" Ce modèle de documents multistructurés est très similaire à celui des arbres multicolores (voir la figure 2.8) car les documents sont aussi modélisés comme un ensemble d'arbres XML. Les différentes couleurs ou les nœuds virtuels sont des mécanismes qui autorisent localement un nœud à posséder plusieurs parents. Cependant, dans le cas des "delay nodes", aucune extension aux langages XPath et XQuery ne sont nécessaires pour naviguer parmi les multiples structures. En contre-partie, il est impossible d'atteindre le nœud parent d'un delay node mais seulement de naviguer parmi ses descendants.

Interrogation des documents multistructurés Nous proposons ci-dessous un exemple de requête qui permet de trouver les titres d'œuvres qui chevauchent deux lignes :

```
for $t in $manuscrit//titre_oeuvre
where
    not ($t/u[1]/parent::ligne is
        $t/u[last()]/parent::ligne)
return $t
```

Conclusion En résumé, le principal avantage de cette proposition réside dans la possibilité d'utiliser assez naturellement XPath (bien que, comme nous l'avons mentionné, il ne soit plus possible de naviguer selon tous les axes) sans avoir pour autant besoin de passer par une extension. Cependant, puisque la représentation XML d'un document multistructuré est, entre autres éléments, constituée de requêtes XQuery, la mise à jour d'un document semble très difficile.

Les structures de traits de la TEI

Introduction La TEI a proposé et soutenu un format pour la représentation des structures de traits ("feature structures"). Ce format a été adopté comme standard en 2006. Il peut être utilisé comme format de représentation pour des documents XML annotés au moyen d'ensembles de termes hétérogènes (qui mélangent par exemple, morphologie, phonématique, syntaxe, etc.) [95].

Le modèle des structures de traits Les structures de traits sont un format générique pour la représentation des connaissances. Elles peuvent être vues comme des fonctions partielles dont le domaine serait un ensemble de "traits" et le domaine image un ensemble de valeurs. Elles peuvent être partielles car elles ne produisent que des informations correctes mais non nécessairement toutes les informations qui définissent les objets qu'elles décrivent. Les valeurs sont soit atomiques (symboliques ou binaires) soit complexes (des structures de traits ou des collections). Il existe deux notations alternatives : les matrices dites "attributs/valeurs" (AVM) ou les graphes. Dans la notation AVM, les valeurs sont placées à la droite de leurs traits, et des crochets permettent de délimiter la portée des sous-structures de traits. De plus, les listes sont indiquées par des chevrons tandis que les accolades indiquent les ensembles.

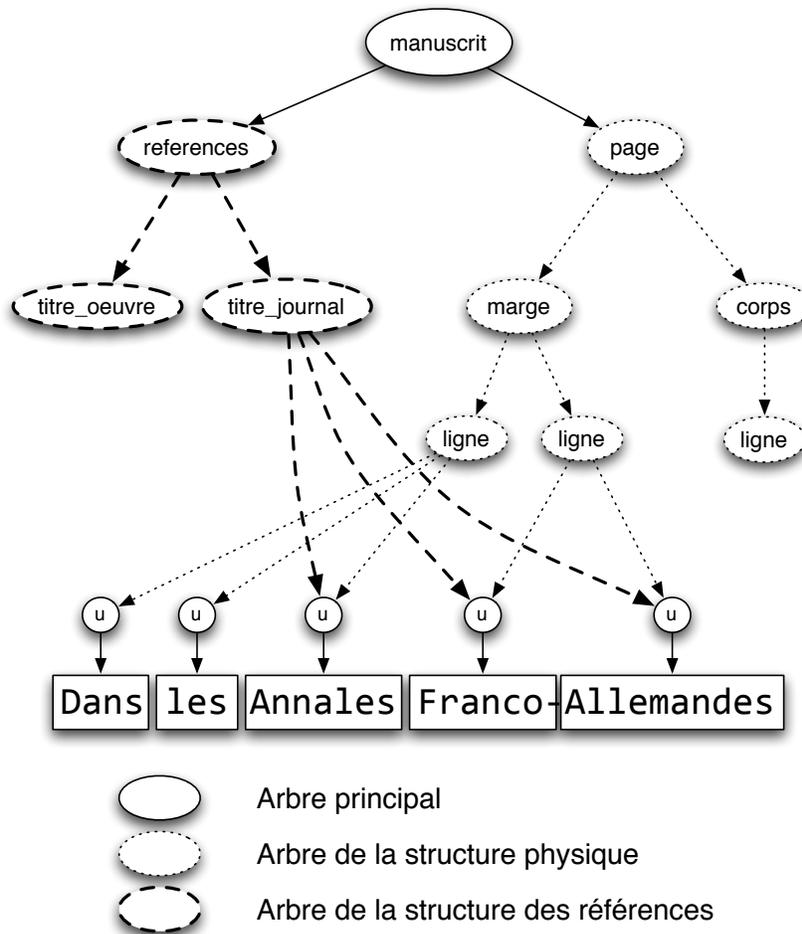
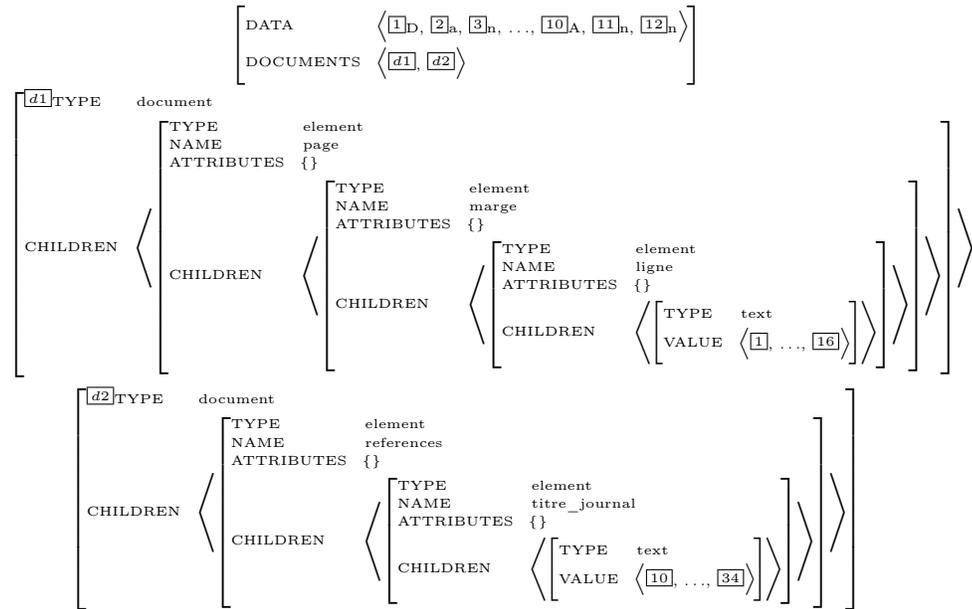


FIGURE 2.8 – Illustration du modèle des "delay nodes"

Illustration du modèle sur notre exemple Nous choisissons la notation AVM afin de représenter notre exemple :



De plus, le sous-vocabulaire de la TEI dédié aux structures de traits permet de les représenter sous la forme de documents XML :

```

<fs>
  <f name="DATA">
    <vColl org="list">
      <vLabel name="1"><string>D</string></vLabel>
      <vLabel name="2"><string>a</string></vLabel>
      ...
    </vColl></f>
  <f name="DOCUMENTS">
    <vColl org="list"><fs>...</fs> <fs>...</fs></vColl>
  </f>
</fs>

```

Opérateurs de manipulation des instances du modèle Les structures de traits reposent sur des notions mathématiques solides. En particulier, les opérations d'unification et de généralisation sont bien définies. Le résultat de l'unification des structures de traits F et G, s'il existe, est la structure de traits H telle que F subsume H et G subsume H (il est dit que F subsume G si F porte moins d'information que G). L'unification peut échouer, F et G sont alors dites incompatibles. La généralisation est l'opération duale de l'unification : le résultat de la généralisation de F et G est la plus spécifique des structures K telle que K subsume F et K subsume G. Ainsi, la généralisation peut être utilisée à profit pour déterminer ce qui est commun à plusieurs représentations d'un document. Cependant, l'unification échouera le plus souvent. A ce sujet, les auteurs expliquent que nous restons libres d'unifier seulement des parties

de la représentation en s'aidant de règles préétablies pour la composition et la décomposition des structures.

Conclusion En résumé, les deux points forts de cette proposition sont la représentation XML standard due à la TEI et les fondations mathématiques solides du modèle. De plus, les opérations d'unification et de généralisation offrent des perspectives très intéressantes pour combiner les documents multistructurés. Cependant, il n'existe aucun mécanisme pour l'interrogation des structures de traits.

MonetDB/XQuery

Introduction MonetDB/XQuery est une base de données XML. Elle possède une extension [2] qui permet la gestion de documents multistructurés grâce à une séparation physique des annotations et du texte annoté. C'est la technique dite de "stand-off markup". Elle dépend de la possibilité de découper le document en régions. Sur notre exemple, les régions peuvent être identifiées par des couples d'entiers qui représentent des intervalles de caractères Unicode :

```
<?xml version="1.0" encoding="utf-8"?>
<manuscrit>
  <references>
    <titre_journal start="10" end="34"/>
    ...
  </references>
  <physique>
    <page>
      <marge>
        <ligne start="1" end="16"/>
      ...
    </marge>
    ...
  </page>
  ...
</physique>
...
</manuscrit>
```

Opérateurs de requête Des opérateurs de requête très optimisés ont été développés sous la forme de quatre nouveaux axes pour le langage XPath. Ces nouveaux axes reposent sur des algorithmes très efficaces qui prennent pleinement en compte les propriétés de la base de données MonetDB. Nous présentons ci-dessous un exemple de requête qui trouve les références à des titres de journaux chevauchant deux lignes :

```
let $manuscrit := doc("manuscrit.xml")
```

```

for $t in $manuscrit//titre_journal
where
    $t/select-wide::ligne
return $t

```

"select-wide" est un pas XPath qui renvoie seulement les nœuds chevauchant le nœud contexte.

Conclusion En résumé, cette proposition est très efficace, une grande attention a été accordée au développement d'algorithmes performants. Cependant, comme pour de nombreuses solutions basées sur la technique du "stand-off markup", la mise à jour des données ou des structures est relativement coûteuse. Finalement, aussi bien le modèle que le langage de requête sont entièrement compatibles avec XML.

MuLaX

Introduction MuLaX [48] est une adaptation de l'option SGML CONCUR au formalisme XML. Il s'agit d'un format documentaire qui permet d'unifier en un unique document des documents XML qui partagent le même contenu. Afin de distinguer les différentes couches d'annotations, le nom des balises est préfixé par un identifiant de structure.

Validation Un document MuLaX peut être projeté sur autant de documents XML qu'il possède de structures. Chaque projection produit un document XML bien formé. Un schéma XML peut être associé à chaque projection. Pour les auteurs, un document MuLaX est valide si et seulement si chacune de ses projections est valide par rapport à son schéma XML. Mais il reste impossible de formaliser des relations entre structures. Autrement dit, nous ne pouvons pas considérer qu'il s'agisse d'une réelle validation d'un document multistructuré.

Interrogation Aucun opérateur de requête n'a été défini. Cependant les auteurs affirment qu'il devrait être possible d'exprimer des chemins similaires à ceux qui existent pour XPath.

Modification Comme dans le cas de l'option CONCUR, les changements de contenu et de structures sont rendus possibles par l'existence d'un unique document. De plus, un éditeur a été développé sous la forme d'une extension à l'EDI Eclipse spécialisée dans la création de documents MuLaX.

Exemple Nous proposons un exemple de la syntaxe MuLaX appliquée à notre exemple :

```

<?mlx version="1.0" encoding="iso-8859-1"?>
<(2)references>
<(1)page>

```

```

<(1)marge>
  <(1)ligne>(texte écrit</(1)ligne>
  <(1)ligne>probablement</(1)ligne>
  <(1)ligne>fin 1843 et</(1)ligne>
  <(1)ligne>publié en 1844</(1)ligne>
  <(1)ligne>dans les <(2)titre_journal>Annales</(1)ligne>
  <(1)ligne>Franco-Allemandes</(2)titre_journal></(1)ligne>
</(1)marge>
<(1)corps>
  <(1)ligne>exigences...<(2)titre_oeuvre>"Contribution ...</(1)ligne>
  ...
</(1)corps>
</(1)page>
</(2)references>

```

Conclusion En résumé, cette proposition est une adaptation de l'option SGML CONCUR au formalisme XML. Elle est intéressante car jamais auparavant il n'y avait eu d'implémentation de cette idée. Mais, le manque d'opérateurs de requêtes ne nous permet pas de la recommander.

RDFTef

Introduction Les auteurs visent à proposer une solution au problème du chevauchement des termes d'annotation qui appartiennent à différentes structures grâce à l'utilisation de RDF [57] et RDFS [16]. Les données sont supposées être textuelles. Le modèle logique est la structure de graphe RDF, tandis que le modèle physique correspond à la sérialisation de ce graphe dans une syntaxe XML standardisée.

Modèle logique Les différentes structures hiérarchiques dont les termes peuvent se chevaucher sont représentées dans le même graphe. Une première "couche" représente par exemple la structure physique tandis qu'une seconde représente la structure des références. Une structure principale doit être choisie et correspond au découpage du contenu textuelles en fragments élémentaires (par exemple les mots et la ponctuation). L'ordre de lecture de ces unités peut être conservé grâce à une propriété "next" qui les relie (dans le modèle RDF, les propriétés sont des arcs orientés). Dans l'exemple de la figure 2.9, pour la structure physique, composée de lignes, les propriétés "first_symbol" et "last_symbol" font respectivement référence au premier et au dernier mot de la ligne (même principe pour la structure des références).

Validation et interrogation Aucun mécanisme n'est proposé pour définir le schéma d'un document multistructuré. Dans la première version de ce modèle, les documents multistructurés pouvaient être interrogés grâce à une adaptation du langage de requête RDF SeRQL [13]. Dans la dernière version, il s'agit de

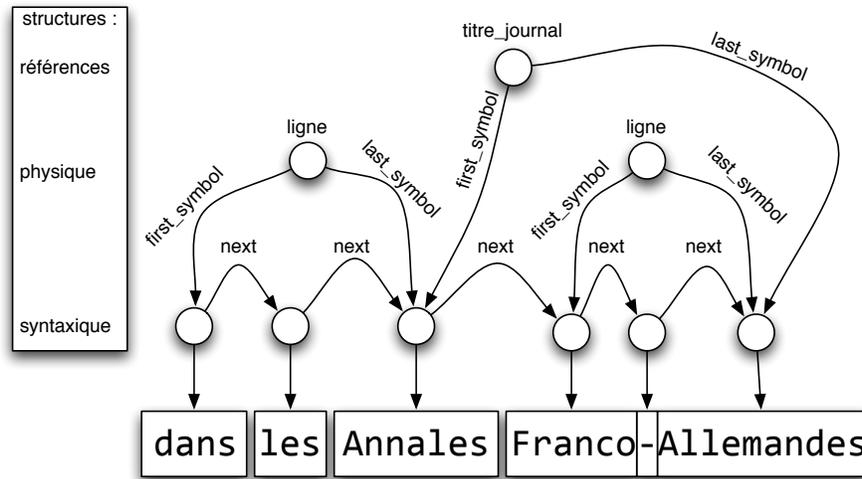


FIGURE 2.9 – Illustration du modèle RDFTef

fonctions ajoutées au langage de requête SPARQL. Par exemple, grâce à la fonction "Precedes", nous pouvons écrire la requête qui retourne les références à un titre de journal chevauchant des lignes :

```
PREFIX function: <java:org.rdftef.query.functions.>
PREFIX desanti: <http://desanti.org/vocab.owl#>
SELECT ?t
WHERE {
  ?l a desanti:Ligne .
  ?t a desanti:Titre_journal .
  ?l desanti:first_symbol ?lf . ?l desanti:last_symbol ?ll .
  ?t desanti:first_symbol ?tf . ?t desanti:last_symbol ?tl .
  FILTER (function:Precedes(?lf, ?tf)) .
  FILTER (function:Precedes(?tf, ?ll)) .
  FILTER (function:Precedes(?ll, ?tl)) .
}
```

Mise à jour et implémentation Ni la mise à jour du texte, ni celle des structures ne sont prévues par cette solution. Une implémentation⁶ est disponible qui utilise la librairie RDF Java Jena.

Conclusion En résumé, cette proposition est basée sur le standard RDF qui est lui-même syntaxiquement compatible avec XML. Nous pouvons considérer cette solution comme une implémentation des graphes d'annotation au moyen de

6. <http://rdftef.sourceforge.net/#download>

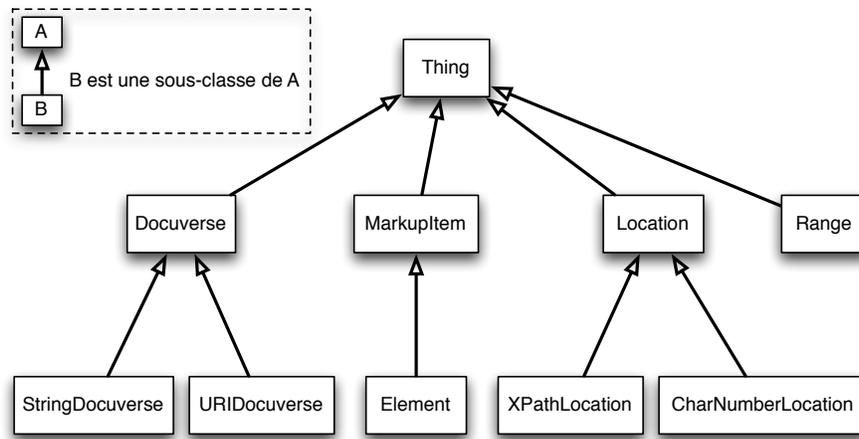


FIGURE 2.10 – Extrait de la hiérarchie des classes EARMARK

technologies récentes et standards. A la différence notable de l'ajout de quelques fonctions au langage SPARQL pour pouvoir interroger un document multistructuré.

EARMARK

Introduction EARMARK [73] (Extreme Annotational RDF MARKup) est un langage pour la représentation de documents multistructurés à l'aide d'une ontologie OWL (voir la figure 2.10 qui représente les relations d'héritage entre quelques-unes de ses classes).

Modèle Les classes "Location" et "Range" permettent de définir explicitement le mécanisme du "stand-off markup" utilisé pour représenter les annotations qui décrivent les documents sources, eux-mêmes des individus du type "Docuverse" (soit "StringDocuverse pour les documents stockés localement, soit URIDocuverse pour les documents distants). Les annotations sont elles-mêmes des individus du type "MarkupItem" (dont les sous-classes sont Element, Attribute ou Comment). Comme dans le cas de la solution précédente (RDFTef), les documents EARMARK peuvent s'exprimer sous la forme d'un ensemble de triplets RDF. De plus, un outil a été développé qui permet de linéariser un document EARMARK sous la forme de documents XML en utilisant, par exemple, les "astuces" proposées par la TEI (balises vides, partitionnement, etc.).

Conclusion En résumé, cette proposition est basée sur une définition ontologique de la plupart des concepts utilisés par les solutions "compatibles XML", mais aussi de ceux qui relèvent des syntaxes spécialisées comme LMNL ou TexMecs. De plus, des feuilles de style XSLT ont été développées et permettent de

convertir des sous-ensembles d'un document EARMARK vers des documents XML. Cette approche ontologique est certainement celle qui s'approche au plus près d'une solution viable pour la création de schémas qui permettraient de valider des documents multistrués. En fait, il est toujours possible de définir des contraintes OWL ou bien des règles SWRL. Cependant, bien que le langage SPARQL puisse être utilisé pour interroger des documents EARMARK, les requêtes complexes nous semblent souvent difficiles à écrire.

GODDAG

Introduction Cette proposition cherche à permettre la représentation et l'interrogation des documents multistrués. Ce travail correspond sans doute à la première tentative d'un réel système de gestion documentaire qui prend en compte les éventuels chevauchements de termes d'annotation. A l'origine, ce système a été développé pour permettre l'édition de manuscrits médiévaux anglais. Cependant, une extension a été proposée [29] pour permettre au système de gérer non seulement du texte mais aussi des fragments d'images.

Modèle logique Ce travail se base sur le modèle logique des GODDAGs (General Ordered Descendant Directed Acyclic Graph) [93]. Comme nous le voyons sur la figure 2.11, plusieurs structures arborescentes sont définies sur le même texte. C'est-à-dire que ces arbres partagent les mêmes feuilles (les fragments de texte). Ainsi, la plupart des propriétés propres aux arbres sont conservées, mais certains nœuds peuvent évidemment avoir plusieurs parents.

Exemple Sur notre exemple, le texte "Annales" possède un nœud parent du type "Ligne" dans la hiérarchie de la structure physique, et un nœud parent du type "Titre_journal" dans la hiérarchie de la structure des références. Par ailleurs, tous les nœuds ont au moins un ancêtre commun, la racine du document.

Modèle physique Deux modèles physiques ont été proposés. Le premier consiste à créer une méta-structure, médiatrice, reliant les différentes structures hiérarchiques qui restent des documents XML autonomes. Le second – qui est celui finalement choisi pour l'implémentation – génère un unique document multistrué grâce à une généralisation des arbres DOM [49] qui permet de représenter des documents XML avec plusieurs hiérarchies. Ainsi, cette approche ne privilégie aucune des structures.

Validation des documents multistrués Cette approche n'est accompagnée d'aucun mécanisme de schéma pour les documents multistrués. Ainsi, les relations entre structures ne peuvent pas être décrites.

Interrogation Les auteurs ont proposé d'étendre XPath [51] par l'ajout de nouveaux axes qui permettent d'identifier des nœuds en fonctions de leurs relations et indépendamment des structures auxquelles ils appartiennent. Il est ainsi

possible d'accéder à tous les ancêtres (c'est l'axe `xancestor`) d'un nœud pour toutes les hiérarchies auxquelles il appartient. En suivant la même idée, les axes `xdescendant`, `xfollowing`, et `xpreceding` ont été définis. De plus, et afin de gérer les chevauchements, il y a les axes `overlapping`, `following-overlapping` et `preceding-overlapping`.

Exemple de requête Sur notre exemple, le texte "Annales" a pour `xancestors` les nœuds `Ligne`, `Marge`, `Page`, `Titre_journal` et `References`. De plus, la requête : "Quels sont les titres de journaux qui chevauchent deux lignes?" s'exprimerait ainsi :
`/xdescendant::Ligne/overlapping::Titre_journal`.

Enfin, remarquons, qu'il reste impossible de restreindre une requête à un sous-ensemble des structures (c'est-à-dire que tous les ancêtres de chaque hiérarchie sont nécessairement visités).

Mises à jour Les auteurs ne font aucune référence à la mise à jour des données (leur application cible ne le demande pas). Par contre, des opérations d'extraction et d'intégration de structures permettent une forme de mise à jour des structures.

Implémentation Comme nous le précisons plus haut, une implémentation est disponible.

Conclusion En résumé, cette proposition est la première à avoir associé un modèle compatible XML à un mécanisme de requête lui-même compatible XML. Mise à part l'impossibilité de définir un schéma pour les documents multistructurés, la solution nous semble très complète. Cependant, l'extension de XPath nous paraît bien plus contraignante que ne l'aurait, par exemple, été la création de nouvelles fonctions XQuery.

MSXD

Introduction L'originalité de cette proposition [21] (la dernière que nous analyserons) vient de ce qu'elle définit à la fois un modèle, un langage de requête *et* un mécanisme pour la validation des documents multistructurés. Elle se donne quatre objectifs. Premièrement, il s'agit de pouvoir gérer plusieurs structures hiérarchiques indépendantes qui décrivent le même contenu textuel. Ces différentes structures forment autant de documents XML bien formés. Deuxièmement, il faut permettre aux utilisateurs d'ajouter des annotations à chacun des documents structurés. Troisièmement, il faut pouvoir contraindre certaines relations entre les structures concurrentes (ainsi qu'entre annotations). Quatrièmement, il s'agit d'étendre XQuery afin de pouvoir construire des requêtes qui prennent en compte les différentes structures et le contenu. L'application principale qui a motivé le développement de cette solution est l'étude de textes littéraires.

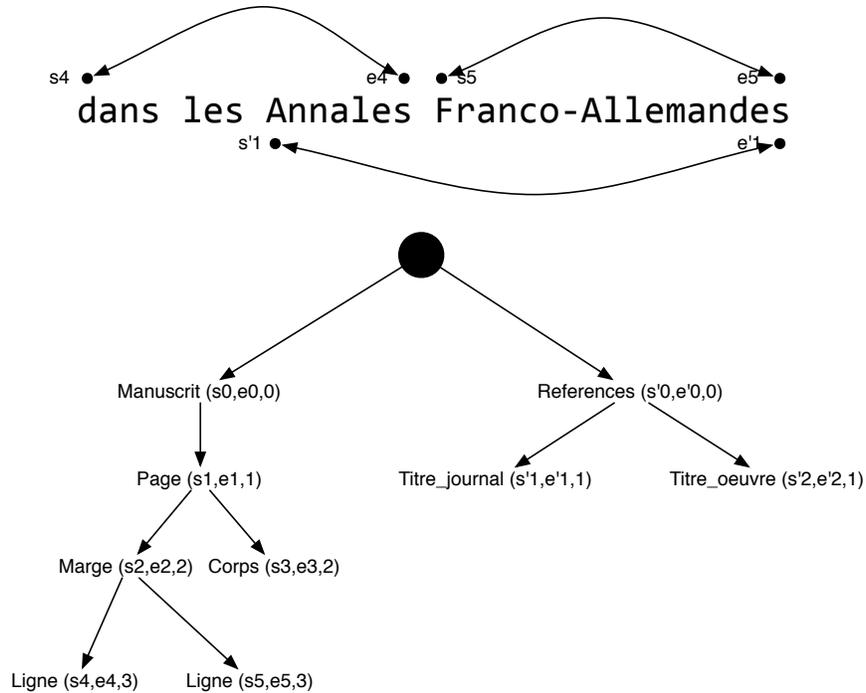


FIGURE 2.12 – Illustration du modèle MSXD

Modèle Cette approche est compatible avec XML car elle se base sur une utilisation de hedge (c'est-à-dire le type des automates sur lesquels se fonde RelaxNG [28]). Un document multistructuré (voir la figure 2.12) est défini comme un ensemble de fragments (qui peuvent être annotés) d'un même texte, regroupés en structures hiérarchiques dont des éléments peuvent éventuellement se chevaucher. Ainsi, pour chaque structure, le texte peut être fragmenté différemment. Un fragment est identifié par une paire (s_i, e_i) qui représente un intervalle du texte à structurer. Chaque structure partage un même nœud racine et est composée de son nom et d'un ensemble de triplets (s_i, e_i, l_i) qui indiquent (grâce à leurs deux premiers éléments) la position d'un fragment du texte et (grâce à leur dernier élément) le niveau de cet élément dans la hiérarchie de la structure considérée. Ainsi, chaque structure accompagnée de ses annotations peut être représentée dans un document XML bien formé. Par ailleurs, le document multistructuré en tant que tel n'est jamais instancié. En effet, l'idée est de conserver la nature hiérarchique de chaque structure afin de pouvoir utiliser les outils XML classiques et de permettre aux utilisateurs d'ajouter des annotations. Finalement, chaque structure existe indépendamment des autres et aucune n'est privilégiée.

Validation A ce modèle est associée une algèbre définie sur les structures et les annotations d'un document afin de permettre la spécification de contraintes entre structures. Ainsi, il est possible de définir un schéma pour documents multistrukturés sous la forme d'un ensemble de règles (les relations de Allen sont utilisées pour contraindre la position relative des fragments et des annotations au sein des structures).

Interrogation Cette proposition s'accompagne d'une extension au langage XQuery [20] afin de permettre la construction de requêtes qui font jouer plusieurs structures hiérarchiques concurrentes sur un même texte. Pour ce faire, elle étend la sémantique des filtres du langage XQuery. Il s'agit en effet d'éviter de changer la structure même du langage. Des fonctions spécifiques et des opérateurs ont été ajoutés à XQuery afin de calculer les valeurs des relations de Allen entre différents éléments. Nous proposons ci-dessous l'expression de la requête qui trouve les titres de journaux chevauchant deux lignes :

```
for $l in $doc//Ligne return
  $doc//Titre_journal[. is-overlapping $l]
```

Mise à jour La mise à jour des structures ou des données n'a pas été considérée par cette approche.

Implémentation Un prototype a été développé (voir le site <http://sis.univ-tln.fr/msxd/>).

Conclusion En résumé, cette proposition est compatible XML et fournit un modèle formel et un langage de requête défini en tant qu'extension XQuery et n'en modifiant pas la structure interne. De plus, les utilisateurs peuvent annoter les structures. Enfin, MSXD propose une solution viable pour la définition de schémas pour les documents multistrukturés. Cependant, en pratique, il n'est pas facile de définir de nombreuses contraintes entre structures.

2.4.4 Synthèse

Le tableau suivant résume les principales caractéristiques des propositions que nous venons d'étudier.

	modèles de données	compatible XML	structure principale	schéma	requêtes	mise à jour	implémentation
CONCUR	aucun	syntaxe SGML	non	non	non	oui	-
TEI (milles-tones)	aucun	syntaxe XML	oui	non	oui	-	-
Graphes d'annotations	graphe propriétaire	syntaxe XML pour la sérialisation	non	non	extension linguistique pour XPath	oui	oui
LMNL	graphe propriétaire	langage balisé spécifique et import/export XML	non	non	non	oui	oui
MVDM	graphe propriétaire	import	non	non	RI orientée classification	non	oui
MSDM	graphe propriétaire	import/export XML	oui	non	fonctions XQuery spécifiques pour l'interrogation et la manipulation des structures	par reconstruction totale	oui
Arbres colorés	Extension du modèle de données XML	export	non	non	extension des chemins XPath	oui	oui
"Delay nodes"	Extension du modèle XDM	nouvelle syntaxe introduite pour la sérialisation des delay nodes	oui	non	XQuery sans extension	non	oui
Structures de traits	fonctions partielles	syntaxe XML	non	non	généralisation/unification		oui
MonetDB	Stand-Off markup	syntaxe XML	non	non	extension des chemins XPath	oui	oui
MuLaX	Extension du modèle de données XML	projection d'un document XML pour chacune des structures	non	non	non	oui	oui
RDFTef	modèle de données RDF	syntaxe XML	oui	non	nouveaux filtres SPARQL	-	non
EARMARK	ontologie OWL	oui : grâce à des feuilles de styles XSLT	non	contraintes OWL	SPARQL	non	oui
basé sur les GODDAGs	GODDAG	extension de l'arbre DOM et import/export XML	non	non	extension des chemins XPath	oui	oui
MSXD	Extension du modèle de données XML	Un document XML par structures	non	expression de contraintes	extension XQuery	non	oui

2.5 Conclusion

Nos objectifs Notre intention était de présenter un état de l'art des solutions existantes pour la gestion des documents multistructurés. Un document multistructuré textuel est la description d'un texte au moyen de plusieurs structures hiérarchiques concurrentes (c'est-à-dire en particulier que certains de leurs éléments peuvent se chevaucher). En général, nous avons pu remarquer que chaque structure correspond à un niveau d'analyse du texte en vue d'un usage particulier. Notre objectif était de faire la liste des solutions qui permettent de représenter et d'interroger de tels documents.

Organisation de l'étude Avant d'étudier ces différents travaux, nous avons choisi de les répartir en trois catégories : les solutions historiques (principalement syntaxiques), les solutions basées sur des modèles propriétaires et les solutions "compatibles XML".

Synthèse Finalement, comment choisir entre ces différentes solutions ? La nature du projet qui demande de gérer des documents multistructurés peut aider à faire ce choix :

- S'il s'agit uniquement de pouvoir décrire plusieurs structures au sein d'un même document, l'usage des solutions proposées par la TEI ou bien de LMNL semble indiqué.
- Si des documents XML bien formés existent déjà et qu'il faut pouvoir les manipuler avec les outils traditionnels de la galaxie XML, des approches comme les GODDAGs, MSDM ou MSXD semblent indiquées.
- Si les données sont facilement découpables en atomes isomorphes (par exemple, le caractère Unicode), alors les arbres multicolores et les "delay nodes" sont envisageables.
- Si les documents ne semblent pas posséder une organisation hiérarchique, EARMARK est indiqué. MVDM peut aussi être envisagé si l'application cible nécessite de classer de nouveaux documents.
- S'il s'agit principalement de pouvoir exécuter efficacement des requêtes sur ces documents multistructurés, alors MonetDB semble la solution la mieux adaptée.

Conclusion Finalement, nous remarquons que chacune des ces solutions suppose que peu ou pas de modifications seront apportées aux documents, que ce soit dans leur contenu ou leurs structures. De plus, la question de savoir ce qui fait l'unité d'une structure, autrement dit, comment sont choisis les termes qui appartiennent à une même structure, n'est pas posée. Il est supposé qu'il existe une forme de consensus sur ce qui permet de distinguer les différentes structures. Finalement, le problème qu'adressent ces solutions est principalement celui de la représentation (et de l'interrogation) des documents multistructurés mais non celui de leur construction. C'est pourquoi nous allons maintenant proposer une méthodologie pour la construction des documents multistructurés.

Chapitre 3

Méthodologie pour la construction des documents multistructurés

3.1 Introduction

3.1.1 Omniprésence des hiérarchies

Introduction De la précédente revue des solutions existantes pour représenter des documents multistructurés, nous pouvons, entre autres points, retenir qu’il s’agit presque toujours (l’exception revenant à EARMARK) de gérer ensemble plusieurs structures hiérarchiques qui décrivent les mêmes données. Á n’accen-tuer que cette seule dimension, des solutions qui, et nous avons pris le temps de le montrer dans la section précédente, se distinguent pourtant nettement lorsqu’elles sont considérées sous les angles de l’interrogation, de la validation, de l’efficacité, etc., se ressemblent tout à coup étrangement.

Un présupposé structurel C’est comme si elles dépendaient d’un présup-posé implicite et commun : une condition acceptable (et peut-être nécessaire, mais en tout cas non suffisante) pour définir l’unité d’une structure est la forme arborescente. Autrement dit, étant donné un ensemble de termes d’annotations qui décrivent une donnée textuelle, si cet ensemble forme une hiérarchie, il se peut qu’il représente aussi une structuration cohérente des données ; si au contraire, il forme un graphe, il se peut qu’il soit découpable en plusieurs sous-ensembles qui forment chacun un arbre et représentent chacun une structuration cohérente des données.

Une première explication Où trouver la raison de ce présupposé ? Serait-ce uniquement une conjoncture technologique ? Le langage XML domine l’ensemble

des formalismes de représentation des données semi-structurées. De plus, il impose une structuration arborescente des couches d'annotations. Ainsi, il est la principale raison de l'émergence du dit "problème de représentation des documents multisttructurés". Finalement, il en commande aussi la solution : la description d'un document au moyen d'une unique hiérarchie de termes d'annotations n'est pas suffisante, *plusieurs hiérarchies* sont nécessaires. Alors que d'autres solutions, a priori plus satisfaisantes, comme l'utilisation d'un formalisme de représentation de graphes, son langage de requête et ses mécanismes de schéma (RDF, SPARQL, RDFS et OWL) paraîtraient plus adaptées.

A la recherche d'une seconde explication Avant de nous résoudre à ce triste constat, il nous fallait enquêter un peu plus. Or, nous venons de le montrer, l'état de l'art des solutions techniques existantes ne semble pas pouvoir nous ouvrir cet espace qui laisserait naître une autre explication, mettons, plus... positive. Nous avons donc observé dans toutes sortes de directions à la recherche d'approches qui proposeraient une explication rationnelle, et non seulement conjecturale, de la domination des structures hiérarchiques. Finalement, nous avons trouvé ce problème explicitement traité dans le cadre naissant de la théorie des systèmes et en particulier à travers les travaux de Herbert A. Simon et Valentin F. Turchin. Nous profitons de cette introduction pour indiquer plus précisément la nature de ces travaux et comment ils nous ont servi à approcher le problème, non plus de la seule représentation, mais de la *construction* des documents multisttructurés.

3.1.2 Explications au sujet du privilège accordé aux hiérarchies

Herbert A. Simon

Herbert A. Simon définit la hiérarchie Dans "The architecture of complexity" [90], Herbert Simon commence par définir une *hiérarchie* de la façon suivante :

By a *hierarchic system*, or hierarchy, I mean a system that is composed of interrelated subsystems, each of the latter being, in turn, hierarchic in structure until we reach some lowest level of elementary subsystem.

Il accentue ainsi l'aspect récursif de la hiérarchie. C'est-à-dire qu'il y a répétition d'une forme de similarité structurelle. Par exemple, un livre est composé de chapitres qui sont composés de paragraphes qui sont composés de mots. Ou bien, un livre est composé de pages qui sont composées de lignes qui sont composées de caractères. Insister sur l'aspect récursif revient à distinguer la hiérarchie de la partition (au sens ensembliste). D'ailleurs, dans une note de bas de page Simon précise :

The mathematical term "partitioning" will not do for what I call here a hierarchy ; for the set of subsystems, and the successive

subsets in each of these defines the partitioning, independently of any systems of relations among the subsets. By hierarchy I mean the partitioning in conjunction with the relations that hold among its part.

C'est donc en fonction des relations entre leurs différents éléments (simples ou complexes) que des structures se distinguent. C'est pourquoi, si les deux structures physique et logique, proposées en exemple plus haut, partitionnent chacune l'ensemble des fragments de base (les caractères Unicode), elles représentent pourtant des hiérarchies distinctes.

Hiérarchies et systèmes symboliques Herbert Simon commence par donner des exemples de hiérarchies dans les domaines des systèmes sociaux, biologiques et physiques. Puis il ajoute :

One very important class of systems has been omitted from my examples thus far : systems of human symbolic production. A book is a hierarchy in the sense in which I am using the term. It is generally divided into chapters, the chapters into sections, the sections into paragraphs, the paragraphs into sentences, the sentences into clauses and phrases, the clauses and phrases into words. We may take the words as our elementary units, or further subdivide them, as the linguist often does, into smaller units. If the book is narrative in character, it may divide into "episodes" instead of sections, but divisions there will be.

Nous sommes évidemment heureux de voir notre intuition confirmée par l'auteur lui-même : ce que nous allons apprendre des hiérarchies s'appliquera à notre cas d'étude!

Trois hypothèses pour expliquer l'omniprésence des hiérarchies Simon explore trois hypothèses complémentaires afin d'expliquer cette observable faveur naturelle pour les hiérarchies. D'abord, il considère les processus d'évolution des systèmes complexes. Ensuite, il s'intéresse à la catégorie des systèmes presque décomposables. Finalement, il cherche à déterminer ce qui permet et ce qui caractérise la *description* d'un système complexe.

Evolution et hiérarchie Nous ne rentrons pas dans les détails de l'établissement du rapport entre évolution et hiérarchies tel que le propose Herbert Simon. Nous sommes en effets surtout intéressés par sa troisième hypothèse. Mais mentionnons simplement qu'il parvient à montrer, à partir de nombreux exemples empruntés à des disciplines variées, que la présence de formes intermédiaires stables accélère grandement l'évolution des systèmes complexes à partir de systèmes simples. De plus, les formes complexes résultantes seront "par construction" hiérarchiques. Ainsi, parmi toutes les formes complexes possibles, celles qui sont hiérarchiques possèdent relativement plus de temps pour

évoluer. Permettons-nous un simple exemple. Philippe II construit l'empire macédonien et le lègue à son fils qui l'agrège au sous-ensemble Perse ainsi qu'à d'autres systèmes politiques pour former le grand empire d'Alexandre. À la mort d'Alexandre (32 ans), son empire ne s'est pas soudainement évanoui mais s'est trouvé fragmenté en importants sous-systèmes qui le composaient. Ainsi, pour être Alexandre, il est nécessaire de naître dans un monde où de grands systèmes politiques stables existent déjà.

Systèmes en parties décomposables et hiérarchies Simon montre que les hiérarchies possèdent la propriété d'être en parties décomposables. C'est-à-dire que les relations à l'intérieur d'un composant sont en général plus fortes que celles entre les composants d'un même niveau de la hiérarchie. Cette propriété a pour effet de séparer au sein d'une hiérarchie les comportements dynamiques de fréquences élevées – qui mettent en jeu la structure interne des composants – de ceux de fréquences plus basses – qui concernent les interactions entre composants. Ainsi, l'horizon des agendas prévisionnels des cadres d'une entreprise est d'autant plus lointain que la position dans l'organisation hiérarchique est élevée. De même, les durées moyennes des interactions entre employés d'une entreprise et celles des intervalles entre deux interactions augmentent avec le niveau des employés dans l'organisation hiérarchique.

Description des systèmes complexes Il découle de ce qui vient d'être dit que la hiérarchie est un bon moyen pour décrire un système complexe. Autrement dit, s'il existe des systèmes complexes qui ne soient pas hiérarchiques, il est fort probable qu'ils restent hors de portée de notre observation ou de notre compréhension. De plus, et en raison du rapport étroit entre hiérarchies et répartition de la dynamique (voir paragraphe précédent), Simon rappelle qu'il est souvent intéressant (et caractéristique de la démarche scientifique) de passer de la description de l'état d'un système complexe à celui de son processus de création.

Conclusion des thèses de Herbert Simon Finalement, pour Simon, nous pouvons nous attendre à ce que, sous l'hypothèse d'un monde où le complexe provient du simple, les systèmes complexes soient des hiérarchies. De plus, la propriété de quasi-décomposabilité permet (ou facilite) la description d'un système complexe et de sa genèse. Par ailleurs, les systèmes symboliques qui nous intéressent sont aussi des systèmes complexes. Donc, les thèses de Simon s'appliquent aux documents multistrukturés et permettent de justifier l'apparition de structures hiérarchiques lors de la description de ces documents.

Valentin F. Turchin

Concepts, reconnaissance, classifieurs et hiérarchies Dans le second chapitre de "The phenomenon of science" [100], Valentin Turchin propose à son tour une explication de la présence très importante des structures hiérarchiques

dans les phénomènes (naturels ou artificiels) que nous observons. Il commence par introduire la notion de "concept abstrait" comme l'unité, par-delà une expérience particulière, d'un ensemble de situations expérimentées. Par exemple, les concepts abstraits de "point" ou de "segment" ne correspondent pas à un point ou à un segment particulier mais au point et au segment en tant que tels. Mais, comment savoir si quelqu'un possède un concept abstrait ? Par exemple, celui du point ? Il n'y a pas d'autre solution que de lui montrer une série d'images et de lui demander pour chacune s'il s'agit ou non d'un point. Ainsi, à la notion de concept abstrait est opérationnellement liée celle de reconnaissance. Ensuite, Turchin propose d'appeler un réseau de neurones simple qui assure une tâche de reconnaissance, un discriminateur. Un discriminateur est un classifieur élémentaire, il range l'ensemble de toutes les situations qu'il perçoit en deux classes mutuellement exclusives. La combinaison de plusieurs discriminateurs constitue un classifieur complexe qui arrange les situations perçues en plusieurs classes. Turchin montre ensuite qu'une solution efficace pour reconnaître des situations complexes (par exemple, le dessin d'une maison) consiste à créer un classifieur organisé hiérarchiquement. Par exemple, un premier niveau reconnaîtra des segments, un second des polygones et un troisième des maisons.

Émergence des hiérarchies Finalement, Turchin émet l'hypothèse selon laquelle tout système complexe dont le processus d'évolution a consisté en une succession d'essais/erreurs possède une organisation hiérarchique. L'idée est de remarquer que lorsque la combinaison d'un petit nombre d'éléments s'avère utile, elle est conservée *en tant que nouvel élément* pouvant participer à d'autres combinaisons avec des éléments similaires. Citons Turchin :

[...] any complex system, whether it has arisen naturally or been created by human beings, can be considered organized only if it is based on some kind of hierarchy or interweaving of several hierarchies. At least we do not yet know any organized systems that are arranged differently.

Nous sommes heureux de rencontrer une référence explicite à la nécessité d'utiliser plus d'une hiérarchie pour décrire certains systèmes complexes. Plus loin, l'auteur précise :

To be more universal a system must resemble not one pyramid but many pyramids whose apexes are arranged at approximately the same level and form a set of concepts (more precisely, a set of systems of concepts) [...] But the pyramids may overlap in their middle parts (and, they are known to overlap in the lower part, the receptors). Theoretically the total number of pyramid apexes may be as large as one likes, and specifically it may be much greater than the total number of receptors. This is the case in which the very same information delivered by the receptors is represented by a set of pyramids in a set of different forms figured for all cases in life.

"Stand-off markup" avant l'heure Comme il est étonnant et passionnant de trouver, dans un contexte pourtant apparemment si éloigné, une description presque parfaite de la technique du "stand-off markup"! Ce travail de Turchin nous semble très précieux et nous pourrions avec plaisir continuer à le décrire mais nous risquerions alors de trop nous éloigner de notre sujet d'étude ...

Refal Permettons-nous seulement une dernière remarque anecdotique, Valentin Turchin a conçu un langage de programmation, Refal¹. C'est un langage fonctionnel (à la Lisp), mais avec un point de vue déclaratif qui se caractérise par un système de gestion de patterns (à la prolog, mais en chaînage avant). De plus, les arbres et non les listes en sont la structure de base. Un jour, Andrei V. Klimov lui a fait connaître XML car il pensait que Refal serait bien adapté à son traitement. Ainsi est né l'article : "Refal : the language for processing XML documents" [101]!

3.1.3 Conclusion

Nous sommes maintenant convaincus que le privilège accordé, par les solutions de gestion de documents multistrukturés, aux structures hiérarchiques est bien fondé. Ainsi, nous avons eu l'idée d'exploiter explicitement la contrainte d'organisation hiérarchique afin d'assister la création des documents multistrukturés. Mais, avant de décrire la méthodologie pour la construction des documents multistrukturés qui en découle, nous avons besoin de définir plus précisément certaines notions et en particulier celle d'arborescence.

3.2 Définitions

3.2.1 Arborescence

Puisque nous aurons affaire à des structures d'arbres, nous proposons la liste des notions nécessaires à la définition de la notion d'arborescence.

suite liste ordonnée d'objets

graphe couple $G = (N, A)$ où

- N est un ensemble de nœuds
 - A est un ensemble d'arrêtes
- et une arrête est un sous-ensemble de N à deux éléments.

boucle arrête qui connecte un nœud à lui-même

graphe simple graphe sans boucles et avec au plus une arrête entre deux nœuds

chemin suite de nœuds telle que pour chaque nœud de la suite il existe une arrête qui relie ce nœud au nœud suivant de la suite.

connexion deux nœuds u et v d'un graphe G sont connectés si G possède un chemin de u à v

1. <http://www.refal.org/>

- cycle** chemin dont les premier et dernier nœuds sont identiques
- arbre** graphe simple connecté et sans cycles
- graphe orienté** graphe dont les arrêtes sont des paires ordonnées
- arbre orienté** graphe orienté qui serait un arbre si les directions des arrêtes n'étaient pas considérées.
- arbre enraciné** arbre orienté dont un nœud est désigné comme racine et dont toutes les arrêtes possèdent une direction naturelle.
- arborescence** arbre enraciné dont toutes les arrêtes pointent en direction opposée de la racine.

3.2.2 Éléments de notre modèle pour la représentation des documents multistructurés

Bien qu'ici notre intention soit de proposer une méthode qui permette la construction de documents multistructurés, nous avons besoin auparavant d'une modélisation précise de ce que nous entendons par "document multistructuré". De plus, notre plateforme philologique (DINAH), qui se verra démontrée au moyen de scénarios d'utilisation dans un prochain chapitre, repose entièrement sur ce modèle. Ainsi, nous définissons maintenant l'ensemble des éléments du modèle :

- ressource** tout objet identifié de manière unique par une URI. En particulier, les intervalles, les zones, les termes, les classes, les relations, les vocabulaires et les documents sont des ressources.
- intervalle** un couple $(D, (inf, sup))$ où D est l'identifiant d'un document textuel et (inf, sup) est un couple d'entiers qui désigne un intervalle du texte de D (l'unité de base est le caractère Unicode).
- zone** un couple $(I, ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)))$ où I est l'identifiant d'une image et $((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$ sont les coordonnées en pixels des points d'une zone polygonale de l'image (le segment $((x_n, y_n), (x_1, y_1))$ ferme le polygone).
- terme** une chaîne de caractères Unicode (rappelons qu'il s'agit de la description de sa représentation, car un terme est une ressource)
- classe** un ensemble de termes
- nom de relation** une chaîne de caractères Unicode (rappelons qu'il s'agit de la description de sa représentation, car un nom de relation est une ressource)
- vocabulaire** un ensemble de noms de relations
- relation** un triplet $(rel, res1, res2)$ où rel est un nom de relation et $res1, res2$ sont des ressources. Autrement dit, il s'agit d'un arc orienté et étiqueté entre deux ressources.
- rhizome** le graphe orienté formé de l'ensemble des relations. Remarquons qu'il est souvent utile de considérer les sous-graphes du rhizome formés des relations dont les noms appartiennent à un même vocabulaire V .

has_term il s'agit du nom d'une relation du vocabulaire *transcription*. Il est réservé. Pour chaque relation de la forme (has_term, i, t) , i doit être un intervalle et t un terme.

structure documentaire étant donnée une source textuelle D et une classe C , il s'agit de l'arborescence implicite formée de l'ensemble des intervalles de D qui sont associés à un terme de C par la relation *has_term*. Les relations de hiérarchie sont déduites des relations d'inclusion des intervalles. Notons en particulier que, puisqu'il s'agit d'une arborescence, deux intervalles d'une structure documentaire ne peuvent pas se chevaucher.

document multistructuré en général il peut s'agir de tout graphe extrait du rhizome afin de répondre à une question particulière. Dans un sens restreint, il s'agit de l'ensemble des structures documentaires associées à une source textuelle D .

Conclusion Maintenant que nous avons donné une définition précise des documents multistructurés, nous pouvons proposer une stratégie pour faciliter leur construction.

3.3 Stratégie

Introduction Nous étudions la construction des documents multistructurés. Il s'agit d'un processus d'annotation d'une source textuelle. Par ailleurs, nous avons montré qu'il était utile de faire des structures documentaires des arborescences. De plus, nous savons – et notre état de l'art le confirme – qu'un document librement annoté formera occasionnellement une arborescence mais le plus souvent un graphe. Enfin, nous nous plaçons sous l'hypothèse d'une absence de connaissance a priori des classes de termes. Ainsi, notre problème est de déterminer comment à la fois assister l'utilisateur dans l'annotation de son document et assurer la condition d'arborescence des structures documentaires.

Exemple Afin d'illustrer cette section, nous reprenons l'exemple qui a servi de fil conducteur à l'état de l'art. Nous essayons maintenant d'imaginer sa dynamique. Supposons un utilisateur qui, d'abord transcrit le texte de la page puis annote quelques éléments. C'est-à-dire qu'il va, par exemple, associer à l'intervalle de texte qui correspond au titre de journal le terme : "titre_journal". Puis, il va annoter quelques auteurs : ..Spinoza, Bateson ..grâce aux termes de la classe "Auteur". Puis, afin de pouvoir aligner l'image du manuscrit et sa transcription, il commence à annoter les lignes. A un moment donné, le titre de journal déjà annoté chevauche une ligne nouvellement annotée.

Le moment de la résolution En effet, il ne peut en être autrement, à considérer le document dans la dynamique de sa construction, il faut que les chevauchements de termes d'annotation soient des événements discrets. *À un moment*

donné, ...ils apparaissent. C'est forcément à l'occasion de ces événements que la stratégie de résolution des chevauchements doit jouer.

La stratégie en elle-même Nous proposons de détecter automatiquement ces événements de chevauchements de termes d'annotation et si les termes appartiennent à la même classe nous avertissons l'utilisateur du conflit et l'invitons à réorganiser les classes de termes afin de distinguer deux structures documentaires là où une seule structure était présente. Nous sommes en pratique assurés de la pertinence de cette indication grâce à la forte cohérence sémantique des structures arborescentes (se référer à la discussion en début de chapitre).

Illustration de la stratégie La figure 3.1 illustre notre stratégie de résolution des chevauchements de termes d'annotation. Nous pouvons remarquer qu'un intervalle associé au terme "Spinoza" de la classe "Auteur" chevauche deux autres intervalles associés aux terme "Ligne" de la classe "S". Cependant, cette situation ne fait pas l'objet d'une détection automatique et du déclenchement d'un avertissement utilisateur car les termes appartiennent à des classes différentes. Ainsi, les structures documentaires restent arborescentes. Par contre, lorsqu'un intervalle qui chevauche deux lignes est annoté comme étant un "Titre_journal" (nuage *i1* de la figure 3.1), il s'agit cette fois d'une instance de chevauchement entre deux termes d'annotation qui appartiennent à la même classe "S". Il faut régler cette situation car sinon, une structure documentaire ne serait plus arborescente. L'utilisateur peut par exemple choisir de distinguer une structure physique en regroupant sous une nouvelle classe nommée "Physique" les termes "Ligne" et "Page", et en isolant dans une autre structure le terme "Titre_journal" (nuage *i2* de la figure 3.1).

Illustration de l'instance de document multistructuré produit La figure 3.2 propose une représentation possible de la portion du rhizome que les opérations mises en jeu par l'exemple pourraient avoir produit.

Illustration de quelques situations d'interaction Bien qu'un prochain chapitre sera tout entier consacré à une analyse de notre plateforme philologique, DINAH, il est utile ici de montrer quelques copies d'écrans relatives aux séquences d'interactions nécessaires à la création du document donné en exemple. La figure 3.3 montre l'interface de transcription : des termes ont été associés à des intervalles, une zone polygonale de l'image manuscrite a été isolée, un avertissement a été déclenché à cause du chevauchement de deux termes d'annotation appartenant à la même classe. La figure 3.4 montre une interface pour l'édition des classes où des termes peuvent être déplacés d'une classe dans une autre, des classes peuvent être créées, etc. Cette interface est souvent utilisée pour résoudre les conflits de chevauchement en réorganisant les classes. Finalement, la figure 3.5 montre comment des relations peuvent être visualisées et créées.

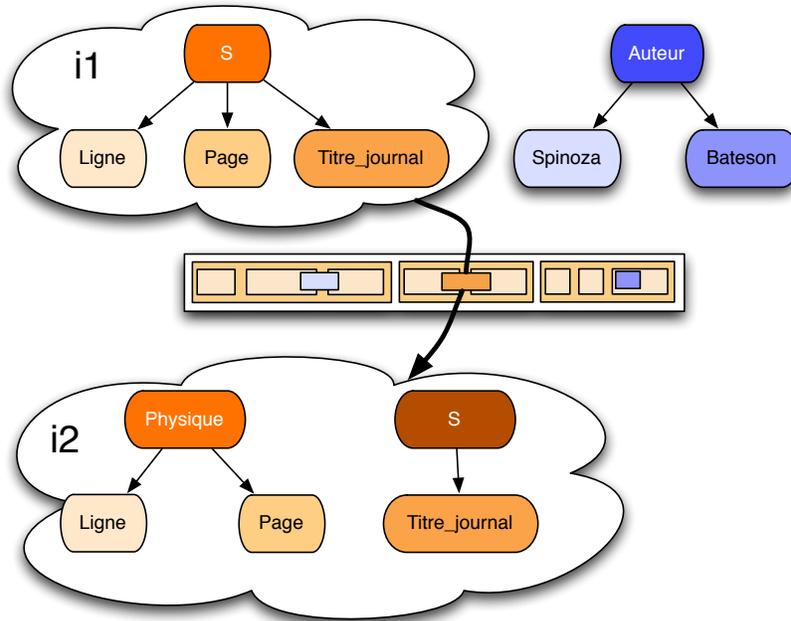


FIGURE 3.1 – Illustration de la stratégie de résolution des chevauchements de termes d’annotation

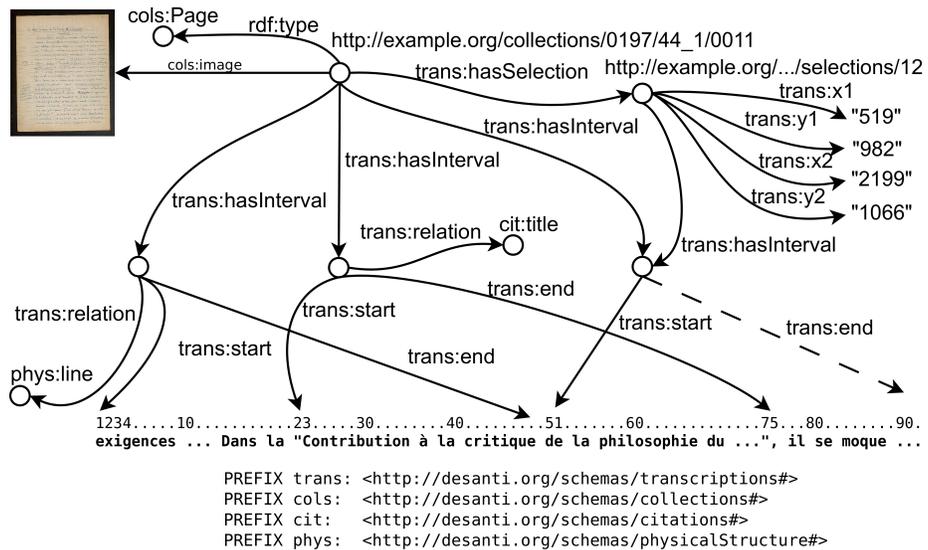


FIGURE 3.2 – Représentation d’un fragment du rhizome suite à quelques opérations d’annotation

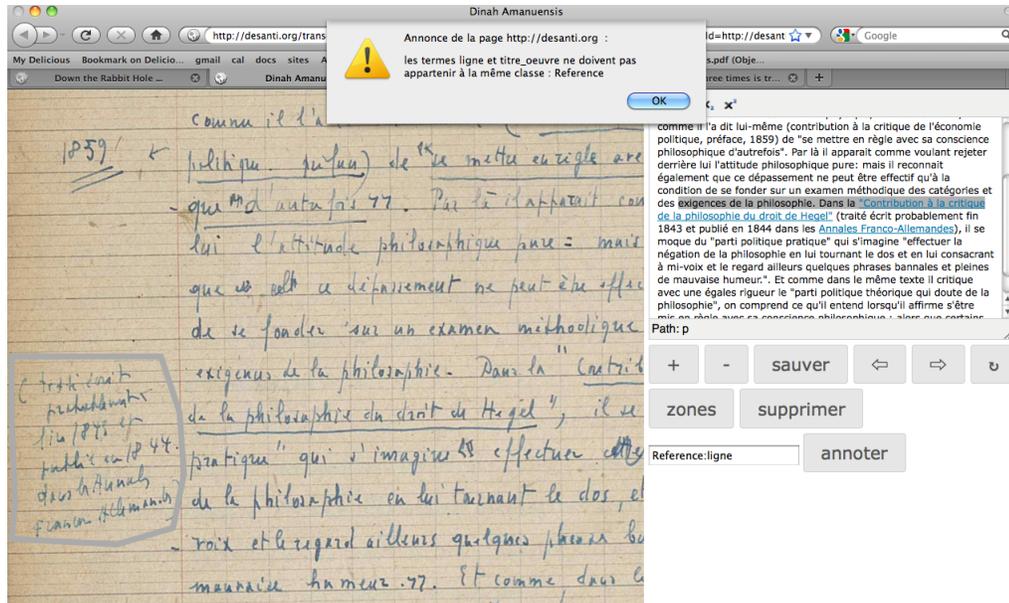


FIGURE 3.3 – Interface de transcription lors d’une situation de conflit entre deux termes d’annotation (titre_journal et ligne)

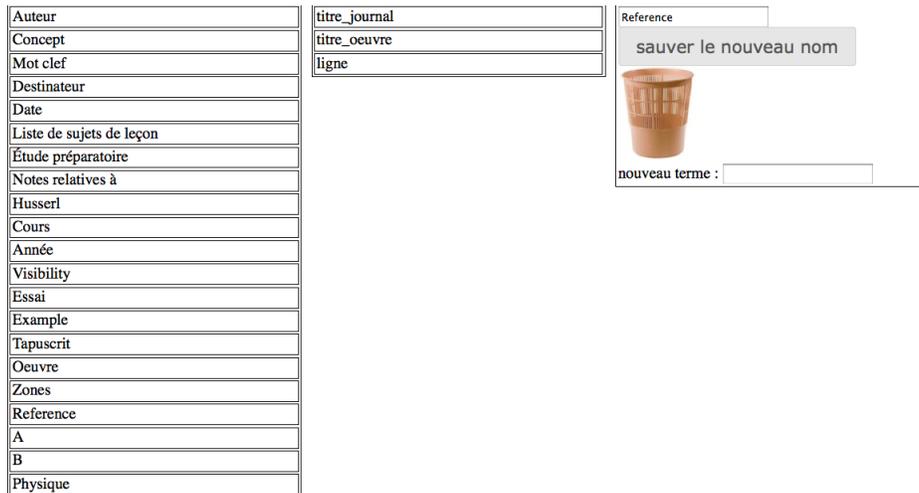


FIGURE 3.4 – Édition des classes afin de résoudre le conflit de chevauchement de deux termes d’annotation

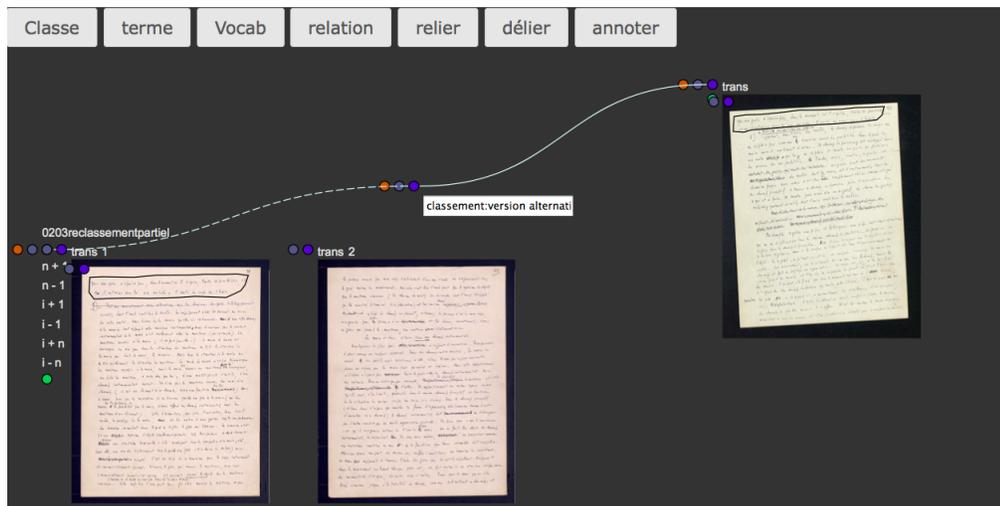


FIGURE 3.5 – Interface pour la visualisation et la création de relations

3.4 Implémentation

RDF Nous avons implémenté notre modèle au moyen du formalisme RDF. Un prochain chapitre détaillera les différents choix architecturaux. Nous pouvons simplement remarquer que la correspondance entre notre modèle et RDF est évidente. Il suffit de poser que "Intervalle", "Zone", "Terme", "Classe", "Vocabulaire", ... sont des classes RDF et que nos relations sont des triplets RDF.

Édition DOM et chevauchements Par ailleurs, la description d'un problème technique intéressant semble avoir sa place ici. Comme la figure 3.3 le montre, notre interface réside dans un navigateur internet. Il s'agit d'une contrainte de notre projet, afin d'être largement diffusable, de ne demander aucune installation, de permettre un travail collaboratif, il fallait que notre plateforme soit entièrement sur le Web. Or, lorsqu'il s'agit de développer un éditeur WYSIWYG, même basique, dans un navigateur Web, il faut se plier à la contrainte hiérarchique de XHTML (qui est un dialecte XML). Ainsi, la question se pose de savoir comment réagit le navigateur lorsqu'un utilisateur tente de faire se chevaucher deux nœuds de l'arbre DOM. Nous n'avons trouvé aucune documentation explicite de ce comportement pour les navigateurs les plus connus (IE, Firefox, Safari, Chrome). Après un peu de rétro-ingénierie, nous avons découvert un comportement partagé que nous illustrons sur un exemple minimal.

Exemple Soit le nœud suivant :

```
<p>aaabbb</p>
```

Supposons qu'un terme d'identifiant "i1id" et appartenant à la classe A soit associé au texte "aaa". Nous choisissons de représenter cette situation ainsi :

```
<p><span id="i1id" class="A">aaa</span>bbb</p>
```

Maintenant, supposons qu'un utilisateur sélectionne le texte "abb" et essaie de lui associer un terme d'identifiant "i2id" et qui appartient à la classe "B". Du point de vue de notre modèle, cela ne pose aucun problème car les deux intervalles qui se chevauchent sont associés à des termes qui appartiennent à des classes différentes. Cependant, du point de vue du modèle de l'arbre DOM, ce chevauchement n'est pas acceptable ! Voici comment réagissent les navigateurs internet susmentionnés :

```
<p><span id="i1id" class="A">aa<span id="i2id" class="B">a  
</span></span><span id="i2id" class="B">bbb</span></p>
```

La figure 3.6 résume ce résultat. Ainsi, nous avons dû développer deux fonctions (en Javascript afin qu'elles puissent s'exécuter sur les navigateurs Web clients) : la première, "makeIntervals", construit les intervalles de notre modèle à partir d'une instance d'arbre DOM ; la seconde, "makeHtml", construit un arbre DOM à partir des intervalles d'une instance de notre modèle. Ces fonctions, relativement simples, nous ont semblé suffisamment intéressantes pour les mettre en annexe.

3.5 Conclusion

Nous avons constaté qu'aucune des approches qui s'attachent à la résolution du problème de la représentation des documents multistructurés ne prend en compte leur aspect dynamique. Nous avons choisi de développer une méthodologie spécifique pour la construction des documents multistructurés. Après avoir vérifié le bien fondé théorique des structures arborescentes, nous avons compris que l'apparition locale d'une structure de graphe au sein de ce qui était un arbre ne pose plus de problème lorsqu'un point de vue dynamique (autrement dit opérationnel) est adopté. Au contraire, nous interprétons ces situations comme des incitations à distinguer plus finement les classes des termes employés pour décrire les documents. Ainsi, bien que techniquement notre modèle logique soit isomorphe à celui de RDF, nous contraignons volontairement les structures documentaires à rester des arborescences. Ce choix a pour conséquence la conservation de structures documentaires solides sur lesquelles peuvent être ajoutées toutes sortes de relations qui vont former un rhizome non contraint. C'est pourquoi nous allons par la suite introduire de nouveaux mécanismes qui vont permettre de documenter et d'offrir une certaine cohérence au rhizome des relations. Mais avant, nous introduisons les concepts qui forment les fondations de notre démarche : il s'agit d'une petite étude sur la notion de signe, elle aimerait éclairer autrement notre méthodologie de construction des documents multistructurés et introduire à l'utilisation ultérieure de la trace des actions d'interaction.

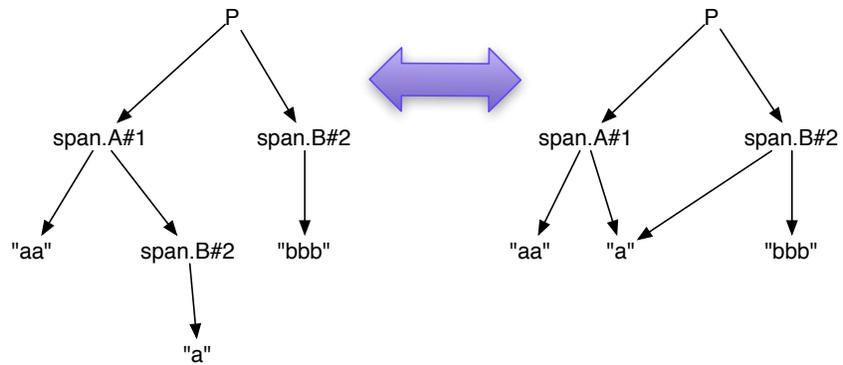
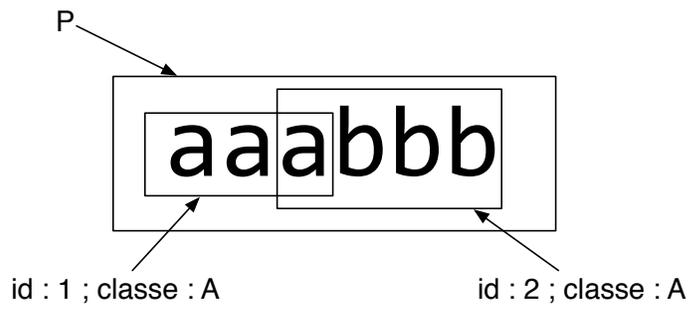


FIGURE 3.6 – Illustration de la gestion, par les navigateurs Web, des chevauchements dans les arbres DOM

Chapitre 4

Ballade avec les signes

4.1 Structuration de l'étude du concept de signe

4.1.1 Projet de construction d'une machine sémiotique

La machine sémiotique Nous avons pour projet de construire une machine sémiotique. Ce dernier couple de termes, nos oreilles l'entendent résonner d'une telle manière que les couples conceptuels classiques s'y dissolvent : homme/machine, actif/passif, utilisateur/outil, etc. Déjà le précédent chapitre avait le goût de cette émulsion. En effet, notre "résolution" de la problématique des documents multistructurés n'avait d'effet que dans un contexte dynamique où les documents sont en cours de construction et reposait sur une orientation de l'attention (de l'utilisateur) en direction d'un phénomène syntaxique (l'apparition local d'une forme de graphe au sein de ce qui était un arbre) dont suivait une explicitation de l'intention (de l'utilisateur) qui, par l'introduction d'un changement dans la structuration des termes d'annotation, influençait l'évolution de l'ensemble de la base documentaire.

Le couplage Homme/Machine et la circulation des signes Si, par commodité, nous faisons encore (mais entre parenthèses) référence à un "utilisateur", ce n'est pas trop forcer l'interprétation que de remarquer le couplage fort entre le sujet et l'objet, l'Homme et la Machine, etc. qu'implique la méthodologie proposée pour la construction des documents multistructurés. Si, comme nous le pensons, il ne s'agit pas d'une situation exceptionnelle, mais de la possibilité d'une orientation actuelle des "systèmes d'information" en tant qu'organes de circulation et de transformation de signes, il devient nécessaire de s'équiper d'un bagage théorique qui permette d'habiter ce nouvel espace mental. C'est de ce constat que naît la réflexion qui suit. Que le concept de signe soit à l'origine de cette dernière peut sembler dans un premier temps arbitraire. Mais il répond en fait à de nombreux critères qui peuvent justifier que ce privilège lui soit accordé.

L'étrangeté du concept de signe D'abord, il est suffisamment extérieur aux domaines de l'informatique, de l'ingénierie documentaire, et même des sciences de l'information et de la communication. Cette excentricité peut nous éviter de répéter (même involontairement) des thèses classiques dont nous sommes comme invités, par le pressentiment exposé plus haut, à nous écarter afin d'éventuellement leur offrir de nouvelles perspectives, une nouvelle lecture possible.

Proximité du concept de signe et de notre champ d'expérimentation Ensuite, il entretient au contraire une certaine proximité avec notre sujet d'expérimentation privilégié, à savoir l'étude de sources textuelles, puisqu'il résonne naturellement avec les concepts de langage, de communication, de lecture et d'écriture, etc. Ces rapprochements nous aident à maintenir la cohérence de nos propos.

Le signe, un concept unifiant Finalement, à lui offrir toute notre attention, nous découvrons un réseau original de concepts et d'auteurs, que par ailleurs nous connaissons et admirons, et que ce nouveau centre éclaire d'une lumière nouvelle (la révolution copernicienne n'est pas loin !). Or, découvrir de l'inconnu à partir d'éléments (au moins pour partie) connus, semble toujours plus assuré que la plongée aveugle en des fonds inexplorés.

4.1.2 Structure sérielle de l'exposition de la machine sémiotique

Série principale Nous organiserons notre exposé par séries de concepts. Une série principale, dont l'auteur Charles Sanders Peirce assurera l'unité, fera émerger le concept de signe d'un développement sur les catégories de l'apparaître. Deux autres séries se grefferont sur elle.

Première série secondaire D'abord, au niveau du passage entre deux catégories de l'apparaître que sont la priméité et la secondéité, s'installera un essai sur le concept d'événement et le couple lecture/écriture.

Seconde série secondaire Ensuite, au niveau de l'articulation entre les concepts de relation et de signe, sera développée une série qui, à travers les concepts de langage, de communication et de différence, rendra avec bien plus de précision (que telle qu'à présent esquissée) la notion de machine sémiotique.

Première série tertiaire Au niveau de cette série secondaire se greffera une série tertiaire qui, au moyen d'une articulation des concepts de contexte et d'itérabilité, et en faisant principalement appel aux auteurs Gregory Bateson et Jacques Derrida, introduira une analyse du concept d'apprentissage. Menée à son terme, cette dernière tendra paradoxalement à dissoudre la construction, pourtant si attentivement échafaudée, de la machine sémiotique !

Introduction du concept de trace Une fois ces parcours sériels achevés, nous pourrions introduire le concept de trace dont l'articulation dépendra entièrement des développements précédents.

4.1.3 Présentation des principaux personnages

Un panthéon introductif Avant d'entamer cette ballade, nous aimerions présenter brièvement les personnages qui la ponctuent. Ainsi, même s'ils doivent par la suite nous faire des tours, apparaitre brusquement à la croisée d'un sentier de traverse comme diables sautant hors leurs boîtes, nous serons à peine étonnés et accueillerons joyeusement leurs facéties, en vieux amis.

Charles Sanders Peirce Commençons par présenter Charles Sanders Peirce dont les concepts nous tiendront souvent lieu de points de vue depuis lesquels initier de nouvelles interprétations de concepts empruntés à d'autres auteurs. C'est un américain du XIX siècle dont les contributions importantes relèvent de nombreux domaines : logique, sémiologie, philosophie, géologie, ... De ses biographies nous apprenons sa grande indépendance et son mode de vie libertin qui l'auraient toujours tenu à l'écart des postes universitaires auxquels ses talents pouvaient le faire prétendre. De plus, il a peu publié de son vivant. Aujourd'hui encore, ses travaux n'ont pas fini d'être publiés sous le titre des "Collected Papers of Charles Sanders Peirce" [71] qui comprennent déjà six volumes sur une trentaine prévus. Avant d'étudier des fragments de ses écrits directement en anglais, nous nous sommes initiés à sa pensée au moyen du remarquable ouvrage "Écrits sur le signe" [72] qui est constitué de traductions et commentaires par G. Deledalle. Et, en effet, nous faisons principalement référence à Peirce pour ses travaux en sémiologie.

Ferdinand de Saussure A la même époque que Peirce, mais indépendamment, le linguiste suisse Ferdinand de Saussure initia lui aussi des travaux sur le signe. Nous n'y ferons que peu référence, et seulement pour marquer grossièrement ce qui le distingue de Peirce.

William James, David Hume, Gilles Deleuze et Platon Quant au philosophe anglais William James (le frère de Henry, romancier), il considérait Peirce comme l'un des meilleurs représentants du pragmatisme, mouvement dont on peut dire qu'il était lui-même un des pères. Nous pouvons considérer le pragmatisme comme héritant de l'empirisme dont David Hume, philosophe écossais du XVIII siècle, est un des fondateurs. Si nous sommes intéressés par ces deux auteurs et les mouvements philosophiques qu'ils représentent, c'est en tant que nous sommes influencés par la lecture qu'en fait le philosophe du XX siècle Gilles Deleuze. En effet, il leur attribue la construction partagée d'une conception très originale de la relation. Comme nous le montrerons, le problème du concept de relation remonte à l'origine grecque de la philosophie, mettons au moins à Platon. Mais il subit une transformation exceptionnelle à l'occasion de l'empirisme,

du pragmatisme et de la philosophie de Deleuze. Ce dernier nourrit aussi notre lecture de Peirce à travers les références faites à sa sémiologie dans les deux ouvrages (sur le cinéma) "images-mouvement" [30] et "images-temps" [31].

Gregory Bateson, Alfred Korzybski et Jacques Derrida Gregory Bateson est un autre auteur américain du XX siècle auquel nous ferons référence et qui accepte l'héritage du concept de relation. Nous utiliserons son concept de différence (ce qui nous amènera à citer Alfred Korzybski resté fameux pour son énoncé : "la carte n'est pas le territoire" [58]), ainsi que sa théorie de l'apprentissage. Nous éclairerons cette dernière au moyen de réflexions empruntés au philosophe français Jacques Derrida sur les notions de contexte et d'itérabilité.

Charles Péguy Finalement, nous clôturons notre "petit panthéon portatif" sur Charles Péguy auquel nous reprendrons, à fin d'illustration, certains de ses développements sur la notion d'événement et sur l'opération de lecture/écriture.

Synthèse La figure 4.1 propose une représentation synthétique de cette section.

4.2 Introduction aux trois catégories du phénomène

4.2.1 Le signe ... un phénomène !

Situation de l'essai introductif Comme convenu, nous reprenons, dans un premier temps, une articulation conceptuelle propre à C.S. Peirce qui, d'une analyse catégorielle des phénomènes, aboutit à une définition du signe. Peirce défend la nécessité d'une analyse ternaire de l'ensemble des phénomènes. En essayant de convaincre de la validité de cette hypothèse, nous introduisons une première fois ces trois catégories. Ensuite, nous ferons une analyse plus fine de chacune d'entre elles. Nous serons alors prêts à aborder le concept du signe, ce qui mettra un terme au développement principal. Entre temps, nous aurons profité des excursions sur les séries secondaires déjà introduites précédemment.

Fin de l'entreprise Alors, et si notre entreprise est couronnée de succès, nous aurons conquis un point de vue qui embrasse avec cohérence l'ensemble de notre projet, éclaire autrement le chapitre précédent sur la construction des documents multistructurés et introduit au chapitre suivant sur l'utilisation de la trace pour documenter les vocabulaires d'annotation.

Qu'attendre d'une analyse des phénomènes ? Une part importante des travaux de Peirce est consacrée à l'analyse des phénomènes. Par phénomène, il faut entendre tout ce qui est observé, c'est-à-dire tout ce qui est présent à l'esprit. Pour l'instant, prenons dans un sens très vague cet aspect de présence

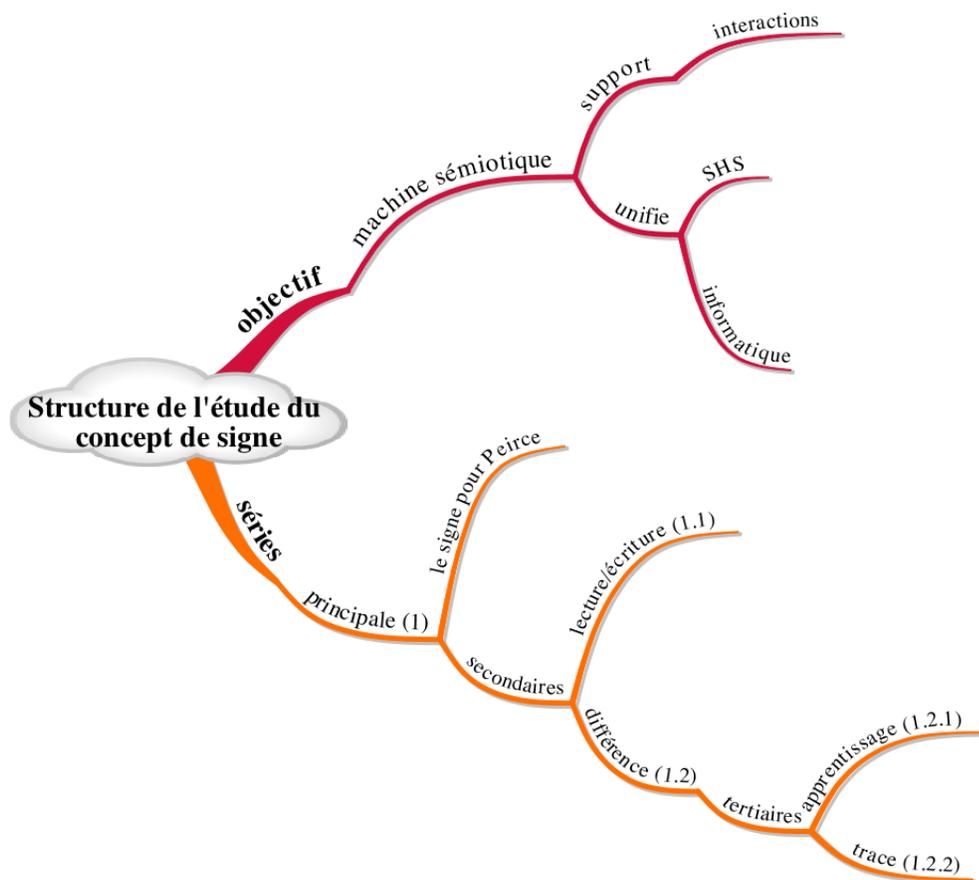


FIGURE 4.1 – Carte de synthèse sur la structure de l'étude du concept de signe

comme une indication de la non pertinence des questions qui (naturellement) pourraient venir (à l'esprit, justement) : l'esprit de qui ?, à quel moment ?, en quel lieu ?, etc. Cependant, il va de soi que la mise à l'écart immédiate de telles questions doit se trouver justifiée.

Une vérification expérimentale Elle le sera si l'analyse proposée des phénomènes peut, en tout cas, en toute situation, être répétée. Cette répétition peut prendre la forme de la lecture même de cet essai. En effet, si chaque lecture, toujours neuve et inconnue, actualise l'analyse proposée, c'est-à-dire en vérifie expérimentalement la pertinence, c'est bien qu'il suffit de dire du phénomène qu'il est ce qui est toujours présent à l'esprit, sans avoir à (et d'ailleurs sans pouvoir) se préoccuper de savoir de qui est-ce l'esprit etc. ... Puisque, en effet, en ce qui concerne ce type de questions, le chercheur (le questionneur, etc.) est toujours identique au cherché (au questionné, etc.). Ainsi, les résultats de l'analyse qui suit doivent valoir "en toute occasion", ils doivent toujours pouvoir se vérifier expérimentalement ... En fait, ils ne peuvent être vraiment compris qu'à l'occasion d'une telle vérification.

4.2.2 Les trois catégories de Peirce

La secondéité

La matière et les faits Nous présentons d'abord la seconde catégorie de Peirce, car c'est la plus accessible. Il l'appelle "secondéité". C'est la catégorie des faits, de l'actuel, de l'individuel, de l'événement (même si nous verrons plus loin que la notion d'événement relève aussi d'une autre dimension qui ne s'épuise pas dans l'actuel). Les faits résistent et s'imposent (à la volonté par exemple), ils sont matériels. La matière (les forces : action/réaction) est perçue immédiatement – c'est une autre manière d'affirmer qu'elle s'impose.

La secondéité de la secondéité Ainsi, la secondéité n'est pas seulement la deuxième des trois catégories de Peirce, elle est aussi seconde en elle-même. En effet, le fait s'impose au sujet (premier) en tant qu'objet (second). Le sujet ne semble jamais capable d'écarter, de ne pas résister, de ne pas s'opposer à l'objet qui lui apparaît.

L'événement actuel Autrement dit, la secondéité est ce qui dans l'événement (quel qu'il soit) est actuel, c'est-à-dire l'ensemble des relations qui inscrivent l'événement dans l'espace-temps. C'est pourquoi, l'événement, en tant qu'actuel, n'existe que pour un sujet premier et seulement par rapport à un environnement second. Ainsi introduite, cette catégorie du fait, de l'actualité de l'événement, de l'individuel etc. peut sembler saturer la description d'un phénomène quelconque. Pourtant, deux autres catégories sont nécessaires à Peirce.

La tiercéité

La loi Continuons par l'analyse de la troisième, la tiercéité. Selon une perspective d'abord très élevée, cette catégorie décrit la conformité grossière des faits à certaines lois. Elle rend compte de la prédictibilité des faits ou du caractère général que revêt la secondéité future et qui la rend pensable pour un sujet présent. Tout est dans ce terme de "pensable", il y a en effet une identité de la pensée et de la loi. Peirce propose l'exemple de l'évolution du concept de force. Considérée comme cause du mouvement, la force est le concept central d'une mécanique qui ne parvient pas à expliquer de nombreux phénomènes (secondéité). A partir de Galilée, la force est pensée comme accélération, c'est-à-dire non plus la relation entre deux positions (vitesse) mais entre trois. Maurice Clavelin propose une très belle analyse de cette transformation dans son ouvrage "Essai sur les origines et la formation de la mécanique classique" [86].

Gravité, inertie, force et poids Remarquons en passant que les découvertes de Galilée ont lieu au sein du cadre conceptuel Aristotélécien mais le rendent inopérant et provoque sa transformation. Par exemple, à l'ancienne conception de l'inertie comme inertie d'une gravité sans force motrice, est substituée l'égalité du poids et de la force motrice au sein du concept de gravité! Il doit y avoir une grande joie (un grand jeu) dans l'emploi des mêmes mots mais arrangés autrement pour désigner des formations conceptuelles si différentes.

La force est accélération D'une façon similaire, c'est à l'occasion d'une analyse de la relation du moment d'un corps à l'inclinaison d'un plan que l'observation d'une valeur constante en tout point du plan n'a plus permis de concevoir la force comme proportionnelle à la vitesse. Mais c'est seulement à la toute fin de son œuvre que Galilée établira explicitement la force comme accélération. Cette découverte est l'illustration analogique de la conquête de la tiercéité.

Synthèse analogique Nous précisons "analogique" car la force en tant que transport de vitesses appartient déjà à la tiercéité mais ne révèle pas autant ce que la secondéité a de général que lorsqu'elle est considérée comme la variation de rapports de vitesses. En résumé, la tiercéité est à la loi et à l'esprit ce que la secondéité est au fait et à la matière.

La priméité

De quelques difficultés ... Il nous reste à introduire la première catégorie de Peirce, la priméité. Elle a quelque chose à voir avec la notion de qualité. En tout cas, il faut qu'elle soit en même temps différente de la secondéité – du fait dual, de la rencontre des corps, de la matière – et de la tiercéité – la pensée. Il faut donc que nous ne puissions ni la penser ni la percevoir. Dans quelles difficultés nous nous trouvons! D'ailleurs, Peirce nous avertit : "Souvenez-vous seulement que toute description que nous en faisons ne peut qu'être fausse." [72]. Voilà qui prête à sourire ... Pour le moment, contentons-nous d'une expression



FIGURE 4.2 – Carte de synthèse sur les trois catégories peircéennes du phénomène

approximative de la priméité : elle est la qualité en tant que pure possibilité. Par exemple, le rouge mais non en tant qu'un certain objet est rouge – la rougeité du rouge. Cette dernière n'existe pas mais s'actualise dans la multiplicité des rouges perçus.

1/2 En fait, la conception que nous avons de la priméité (e.g. rougeité) est sans doute déduite de l'expérience sensible – elle reflète une généralité (ou éternité) de l'événement actuel mais non sous le régime de la loi comme le fait la tiercéité.

Avant goût Mais ces idées délicates se verront peut-être éclaircies par l'essai qui suit où nous analysons l'articulation entre priméité et secondéité afin de développer une conception de l'événement. Nous nous intéressons plus particulièrement aux événements qui relèvent du couple lecture/écriture.

Synthèse La figure 4.2 propose une représentation synthétique de cette section.

4.3 L'événement, ou l'articulation de la priméité et de la secondéité

4.3.1 La priméité

De la priméité à l'événement Nous nous rendons au lieu de l'articulation entre priméité et secondéité. Ce sera l'occasion d'aborder une première fois les rivages de l'événement. D'ailleurs, notre étude ne nous amène pas à être concernés par tous les événements mais seulement par celui que symbolise le couple lecture/écriture. Nous choisissons de commencer par approfondir notre compréhension du concept de priméité.

Première définition de la priméité Dans une lettre adressée à Lady Welby, Peirce définit la priméité [71] ainsi ¹ :

Firstness is the mode of being of that which is such as it is, positively and without reference to anything else.

La priméité est le mode d'être de ce qui est (entendre ici "ce qui apparaît"), tel qu'il est positivement et sans référence à quoi que ce soit d'autre.

Seconde définition de la priméité Dans un cours donné en 1903, il propose une définition peut-être plus opératoire que la précédente :

A Firstness is exemplified in every quality of a total feeling. It is perfectly simple and without parts; and everything has its quality. Thus the tragedy of King Lear has its Firstness, its flavor *sui generis*. That wherein all such qualities agree is universal Firstness, the very being of Firstness. The word *possibility* fits it, except that possibility implies a relation to what exists, while universal Firstness is the mode of being of itself. That is why a new word was required for it. Otherwise, "possibility" would have answered the purpose.

Ainsi, la priméité serait une qualité ou un sentiment mais seulement en tant que pure possible, et non en tant qu'expérimenté, senti. Par exemple, le rouge (une qualité) en tant que rouge – la "rougéité" du rouge – relève de la priméité.

La priméité est-elle impensable ? Cependant, toutes les descriptions de la priméité appartiennent, en tant que construction langagières, au domaine de la tiercéité. C'est pourquoi nous pouvons affirmer qu'à la limite les propositions "Ce n'est pas rouge." et "C'est rouge." sont identiques tant que nous sommes intéressés par une présentation immédiate de la qualité rouge – de la pure possibilité du rouge. D'ailleurs, Peirce nous donne un exemple similaire [72] :

...le mot *rouge* signifie quelque chose, quand je dis que la précession des équinoxes n'est pas plus rouge que bleue, et (que) cela signifie exactement ce que cela signifie quand je dis que le rouge de l'aniline est rouge.

1. <http://www.helsinki.fi/science/commens/terms/firstness.html>

La priméité comme qualité singulière Afin de clarifier un peu le vocabulaire, nous empruntons un adjectif cher à Gilles Deleuze pour affirmer que la priméité est la qualité *singulière*. Le singulier se distingue du général et de l'individuel.

Singulier VS Général En effet, nous avons vu que la généralité est la qualité de la loi, c'est-à-dire de la tiercéité. Nous pouvons encore nous appuyer sur un texte de Peirce (The Logics of Mathematics) [72] :

Je puis imaginer une conscience dont toute la vie, toujours pareille, qu'elle fût bien éveillée, assoupie ou rêvant, ne consisterait en absolument rien d'autre qu'une couleur violette ou une odeur de chou pourri. ... Le fait que je puisse l'imaginer montre que ce sentiment n'est pas général dans le sens où la loi de la gravitation est générale. Car personne ne pourrait imaginer que cette dernière loi aurait de l'être de quelque genre que ce fût, s'il était impossible qu'il existât deux masses quelconques de matière ou s'il n'y avait pas des choses comme le mouvement.

Singulier VS Individuel De même, ce qui est éprouvé comme réel – les faits, les états de choses – sont des qualités mais en tant qu'effectuées ou actualisées dans une chose individuée, dans un milieu déterminé – un espace-temps déterminé. Ainsi, l'individuation repose dans la secondéité. Nous empruntons le terme d'individuation à Gilbert Simondon [91]. Selon notre lecture de ce dernier, nous comprenons que les opérations d'individuation supposent un champ pré-individuel. Celui-ci correspond à la priméité. C'est une distribution de potentiels dans un champ physique. Et ce n'est pas pour rien que ces potentiels sont appelés les singularités du champ. Ainsi, nous voyons que la priméité se distingue de la secondéité comme le singulier se distingue de l'individuel.

4.3.2 La priméité de l'événement

Les deux faces de l'événement De plus, nous avons le sentiment qu'il y a des qualités qu'aucune expérience (secondéité) – c'est-à-dire aucun événement ou fait, en tant que passage ou transformation d'un état de choses à un autre – n'actualisera jamais complètement. Cette intuition, de nombreux auteurs la partagent. Par exemple, pour Gilles Deleuze, Maurice Blanchot ou Charles Péguy, tout événement possède deux faces. D'abord, l'expérience qui est la part de l'événement qui s'actualise, le passage d'un état de choses à un autre, la secondéité de Peirce. Ensuite, un reste, "la part de l'événement qui déborde sa propre actualisation" pour reprendre une formule de Blanchot [10], une qualité qui ne s'actualise dans aucun état de chose, la priméité de Peirce.

La lecture et l'écriture comme événements Péguy s'intéresse particulièrement aux événements de lecture et d'écriture. Pour lui, toute lecture est un accomplissement – il emploie le terme d'accomplissement auquel nous avons

préféré celui d'actualisation – qui n'a jamais fini de s'accomplir. Nous nous permettons ici une (longue) citation extraite du début de l'ouvrage *Clio* [70] dans lequel la muse de l'histoire converse directement avec Charles Péguy (l'auteur).

Une citation de Charles Péguy Nous devons en effet laisser un peu d'espace à cet auteur car son style étonnant ne permet (pratiquement) pas la citation (nous recommandons amoureusement la lecture de ce petit livre peu connu).

La simple lecture est l'acte commun, l'opération commune du lisant et du lu, de l'auteur et du lecteur, de l'œuvre et du lecteur, du texte et du lecteur. Elle est une mise en œuvre, un achèvement de l'opération, une mise à point de l'œuvre, une sanction singulière, une sanction de réalité, de réalisation, une plénitude faite, un accomplissement, un emplissement ; c'est une œuvre qui (enfin) emplit sa destinée. ...c'est un des mystères les plus inquiétants peut-être de la destination temporelle, un des plus pleins, des plus bourrés d'inquiétude, que nulle œuvre, si achevée soit-elle, et qu'elle nous paraisse, et peut-être qu'elle ait paru à l'auteur son père, nulle œuvre pourtant n'est temporellement si achevée, n'a temporellement si complètement reçu son chef qu'elle ne doive encore en un autre sens (et peut-être au fond en le même sens, car tous les hommes sont hommes, et cet auteur est homme, et nous aussi, petits, nous sommes hommes, et quoi que nous en ayons nous continuons l'auteur même en un sens) être perpétuellement achevée comme inachevée, au titre d'inachevée, qu'elle n'ait à recevoir et qu'elle ne reçoive et qu'elle ne doive recevoir perpétuellement un chef, un couronnement lui-même perpétuellement inachevé. ... Un zéro de lecture d'une œuvre en est en un sens le découronnement suprême. ... Si dur que soit ce marbre du Pentélique et quelle qu'en soit la patine, non seulement il reçoit perpétuellement, éternellement temporellement, les atteintes physiques du temps, atteintes à la considération desquelles nous sommes habitués par les considérations et souvent les contemplations de tous les philosophes, mais en même temps, dans tout ce même temps il reçoit perpétuellement, éternellement temporellement, d'autres singulières atteintes, les atteintes, les couronnements et les découronnements incessants, les achèvements perpétuellement inachevés, les inachèvement réellement achevés, réellement acquis, réellement obtenus, les couronnements perpétuellement incouronnés et les découronnement perpétuellement et réellement incouronnés aussi de notre collaboration perpétuelle à tous tant que nous sommes, tout petits que nous sommes. C'est ici le plus grand mystère peut-être de l'événement, ...

Une synthèse Quel joie de trouver ainsi exprimé ce que nous pressentions de l'événement – qu'il a comme deux faces, l'une, tournée vers son actualisation temporelle et qui relève de la secondéité, de ce que nous percevons comme réel, des faits ; l'autre qui est l'*immédiat* (priméité) et qui ne se laisse découvrir que

partiellement à travers les médiations de la pensée (tiercéité). Autrement dit, il y a toujours une qualité (de tel texte – l'Illiade, un texte de J.T. Desanti, etc., de telle sculpture, ...) qui ne se confond avec aucun état de choses mais que chaque actualisation de l'événement (de lecture, de vision, etc.) révèle en partie.

Impact sur notre recherche Finalement, en tant que nous construisons des outils dont la destinée est de transformer les manières d'accéder à une œuvre – sa lecture, son interprétation – nous trouvons essentielle la découverte de cette conception de l'événement. En effet, elle oriente notre recherche nous forçant à considérer chaque lecture, chaque interprétation individuelle de l'œuvre comme neuve, c'est-à-dire comme pouvant actualiser une part du potentiel de l'œuvre – et en même temps, comme participant d'un mouvement de découverte qui doit comprendre toutes les lectures précédentes.

Exemple de la multistructuralité Nous remarquons que si nous avons pu considérer l'événement syntaxique local de chevauchement de termes d'annotation comme le véhicule possible d'un effet plus global qui implique une restructuration des vocabulaires de termes d'annotations, c'est peut-être grâce à cette conception de l'événement. En effet, il s'agit bien d'encoder l'effet d'un événement de lecture sur la "compréhension" de l'œuvre.

Une ouverture ... D'un point de vue plus général, nous croyons qu'une forme de sensibilité aux découvertes qui ont pu être faites sur des concepts aussi divers que le signe, l'événement, la relation, etc. n'a pas fini d'apporter des transformations (au moins) intéressantes au sein de domaines (pour nous, mettons, l'informatique) qui leur sont étrangers. Nous continuons à présent notre parcours en montrant comment le rapport de la tiercéité à la relation amène au concept de signe.

Synthèse La figure 4.3 propose une représentation synthétique de cette section.

4.4 La tiercéité et la relation

4.4.1 Le concept de tiercéité

Avant-propos Nous devons à présent analyser le concept de tiercéité. Nous pourrions alors montrer comment le signe est un phénomène de tiercéité particulier. Ce passage de la tiercéité au signe nécessitera une première analyse du concept de relation.

Peirce définit la tiercéité Dans cette même lettre que nous citons plus tôt au sujet de la secondéité et que Peirce adresse à Lady Welby, est définie succinctement la tiercéité :

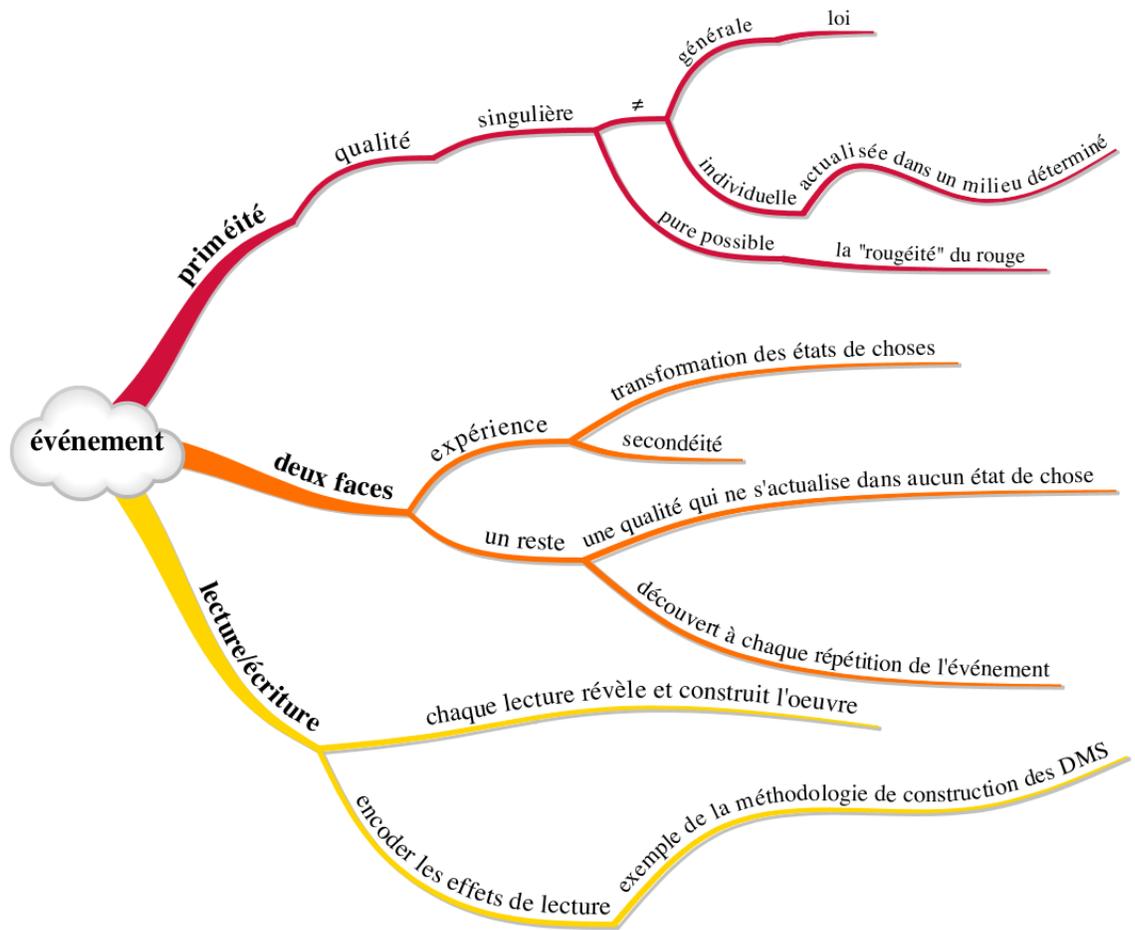


FIGURE 4.3 – Carte de synthèse sur l'événement de lecture/écriture

Thirdness is the mode of being of that which is such as it is, in bringing a second and third into relation to each other.

Ce que Deledalle traduit par :

La tiercéité est le mode d'être de ce qui est tel qu'il est en mettant en relation réciproque un second et un troisième.

Relation et médiation Or, d'après nos précédents essais, la secondéité est le domaine de la relation. C'est-à-dire, rappelons le, celui des faits, de la matière, des chocs, des duels – au fond, de tout ce qui est éprouvé comme réel – des événements inscrits dans l'espace-temps, c'est-à-dire de la multiplicité des relations qu'un objet entretient avec son environnement, ce qui est aussi nommé "état de choses", et de l'évolution temporelle de ces relations, appelée expérience – le passage d'un état de choses à un autre. Oui, mais la secondéité, c'est la relation sauvage, *laissée tranquille* : "le mode d'être de ce qui est tel qu'il est par rapport à un second mais sans considération d'un troisième quel qu'il soit". La tiercéité c'est la relation pensée, réifiée, domestiquée : "le mode d'être de ce qui est tel qu'il est en mettant en relation réciproque un second et un troisième."

La tiercéité dégénérée Peirce nous donne un exemple éclairant [71]² sur ce qui distingue la pure relation (propre à la secondéité) de la médiation de la tiercéité – nous choisissons d'employer ce nouveau terme pour appuyer la distinction – :

Analyze for instance the relation involved in 'A gives B to C'. Now what is giving? It does not consist in A's putting B away from him and C's subsequently taking B up. It is not necessary that any material transfer should take place. It consists in A's making C the possessor according to *Law*. There must be some kind of law before there can be any kind of giving, – be it but the law of the strongest. But now suppose that giving *did* consist merely in A's laying down the B that C subsequently picks up. That would be a degenerate form of Thirdness in which the thirdness is externally appended. In A's putting away B, there is no thirdness. In C's taking B, there is no thirdness. But if you say that these two acts constitute a single operation by virtue of the identity of B, you transcend the mere brute fact, you introduce a mental element.

Nous trouvons dans l'*Essai sur le don* de Marcel Mauss [67], la confirmation que tout don s'accompagne toujours d'un symbole qui est une forme de présence explicite de la loi.

La tiercéité et la loi Mais il y a encore au moins une autre difficulté, il faut bien comprendre que jamais une action/réaction (c'est-à-dire de la secondéité) n'a pour *cause* la loi (c'est-à-dire de la tiercéité) qui la régit. Par exemple, la loi qui affirme qu'aux conditions normales de température et de pression, l'eau bout

2. <http://www.unav.es/gep/Welby12.10.04.html>

à 100° C, n'a jamais fait bouillir de l'eau ! Les lois sont hypothétiques : SI [100° C ET CNTP] ALORS [l'eau bout]. La tiercéité c'est la loi, mais considérée en elle-même, c'est-à-dire en dehors de toute expérience qui la vérifie. En résumé, la pensée, la loi, la relation en tant que relation caractérisent la tiercéité.

4.4.2 Le concept de relation

Une conception antique

Platon et la relation D'ailleurs, le concept de relation intéresse depuis longtemps une partie de l'humanité. Les Grecs – mais nous nous intéressons seulement à Platon – distinguaient deux formes de jugement : l'attribution et la relation. Le jugement d'attribution se rencontre sous la forme "A est B". Pour Platon, c'est la source de grandes difficultés. En effet, à l'exclusion des tautologies – "Le rouge est rouge.", "La rose est rose." ?, ... –, cette forme de jugement met à mal le principe d'identité – "La rose est rouge.". Offrir une solution à ce problème demandera d'introduire une nouvelle notion, la prédication ou attribution d'une propriété à un sujet. Ceci a pour effet de rendre asymétrique le jugement d'attribution, de distinguer un sujet et un objet.

La relation est une Idée Mais les problèmes deviennent bien plus délicats pour l'autre forme de jugement : le jugement de relation. Un exemple classique est : "A est plus petit que B". Ici, A et B sont deux "sujets", il est impossible de considérer B comme une propriété qui serait attribuée à A. Par ailleurs, si "plus petit que" est considéré comme étant une propriété alors le problème vient de ce qu'un C sera toujours déterminable tel que A en vienne à posséder les deux propriétés "plus petit que" et "plus grand que". Dans ce dernier cas, une entorse est faite non plus au principe d'identité mais au principe de non contradiction. La solution envisagée par Platon est de faire de la relation une Idée à part entière. Alors, nous pourrions dire que A participe à l'Idée de "plus petit" en tant qu'il se rapporte à B, Etc. C'est sans aucun doute une très belle création de l'esprit – la relation pensée indépendamment de ses termes.

Extrait du Phédon C'est dans le dialogue *Le Phédon* [17] que nous avons trouvé l'expression la plus simple de cette idée :

PHEDON – Ceci du moins je crois : après s'être mis d'accord avec Socrate, et après avoir reconnu que chacune de ces Formes (Idées) existe et est une réalité déterminée, et que les autres choses reçoivent leur dénomination de leur participation à ces Formes, après tout cela donc, Socrate posa cette question : "Alors, Cébès, si ce langage est bien le tien, quand tu affirmes que Simmias est plus grand que Socrate mais moins grand que Phédon, est-ce que cela ne se traduit pas ainsi : qu'il existe alors, en Simmias, ces deux choses, grandeur et petitesse ?

— C'est bien ce que je veux dire

— En réalité, pourtant, tu es bien d'accord, l'énoncé "Simmias dépasse Socrate" ne formule pas ce qu'il en est en vérité? Car, je pense, si "dépasser" constitue une propriété naturelle de Simmias, cela ne tient pas au fait que Simmias est Simmias mais bien à la grandeur qui se trouve être la sienne? Et que ce soit Socrate qu'il dépasse, cela ne tient pas davantage au fait que Socrate est Socrate, mais que Socrate a de la petitesse relativement à la grandeur de l'autre?

— C'est vrai.

— Pas plus d'ailleurs que Simmias n'est dépassé, lui, par Phédon, du fait que Phédon est Phédon, mais bien parce que Phédon a de la grandeur relativement à la petitesse de Simmias?

— Oui, c'est bien cela.

— Très bien. En conséquence, Simmias possède comme dénomination aussi bien "petit" que "grand" puisque, se situant entre les deux, il soumet d'une part sa petitesse à la grandeur du premier pour en être dépassé; et présente d'autre part sa grandeur à la petitesse du second afin de la dépasser."

Alors, avec un sourire : "J'ai tout l'air, dit-il, d'être en train de rédiger un contrat! Mettons, en tout cas, que ce soit à peu près comme je dis." Cébès approuva. "Voici où je veux en venir : à te voir partager la même opinion que moi. Car ce qui m'apparaît ne m'apparaît pas seulement à propos de la grandeur en soi – elle ne consent jamais à être en même temps grande et petite –, mais aussi à propos de la grandeur qui est en nous : jamais elle n'accueille en elle le petit, ni ne consent à être dépassée. De deux choses l'une : ou bien elle s'enfuit et se retire quand s'avance sur elle son contraire, le petit, ou bien, du fait de l'arrivée de ce contraire, elle périt complètement. Mais elle ne consent pas à être autre que ce que précisément elle était, en restant là et en recevant la petitesse, à la façon dont moi, par exemple, une fois que, l'attendant de pied ferme, j'ai reçu la petitesse, c'est tout en continuant d'être ce que précisément je suis et tout en restant le même, que je suis petit ; mais la grandeur qui est en moi, elle, n'a pas l'audace, étant grande, d'être petite! Exactement de la même façon, la petitesse qui est en nous ne consent à aucun moment à devenir grande ou à l'être ; et aucun autre contraire, quand il est encore ce qu'il était, ne peut, en même temps, ni devenir ni être son propre contraire, mais ou bien il s'en va, ou il périt du fait de se trouver soumis à une telle action.

— Cela me paraît tout à fait évident", dit Cébès.

Entendant cela, quelqu'un prit la parole – lequel exactement de ceux qui étaient présents, je ne m'en souviens pas. "Par tous les dieux! Au cours de vos raisonnements précédents; est-ce que vous n'êtes pas tombés d'accord juste sur le contraire de ce qui se dit maintenant? Sur le fait que le plus grand naît du moins grand et le moins grand du plus grand, et, pour le dire carrément, qu'il y a

génération des contraires à partir de leurs contraires ? Or on vient de dire, il me semble, que cela ne peut jamais se produire ?" Socrate tourna la tête vers lui et l'écouta. "C'est brave à toi, dit-il, de nous remettre cela en mémoire ! Pourtant tu ne réfléchis pas qu'il existe une différence entre ce qu'on dit maintenant et ce qui a été dit tout à l'heure. Car à ce moment-là on disait : d'une chose contraire naît une chose contraire ; mais on dit à présent : le contraire, en lui-même ne peut jamais devenir son propre contraire, qu'il s'agisse du contraire qui est en nous ou du contraire dans sa nature propre. Car vois-tu, ami, à ce moment-là nous parlions de choses qui possèdent des propriétés contraires et nous les appelions par les noms qui leur venaient de ces contraires ; tandis qu'à présent nous parlons des contraires en eux-mêmes, de ceux qui, présents dans les choses, confèrent leurs noms aux choses que nous dénommons ainsi.

Les conceptions modernes

Une solution extravagante ... Afin de ne pas occulter l'immense diversité de l'esprit, remarquons qu'au XVII^{ème} siècle Leibniz propose une solution diamétralement opposée ! Il considère que tout jugement est jugement d'attribution. Contrairement à ce que nous venons de présenter, il faut pour lui que "A est plus petit que B" soit un jugement d'attribution. Ainsi, "plus petit que B" est une propriété de A. Autrement dit, B doit être "compris" dans A. Mais, symétriquement, à partir du jugement "B est plus grand que A", il faut en déduire que A est compris dans B. Ainsi, pour Leibniz, tout concept d'être réel exprime la totalité du monde [63]. Il est alors obligé d'introduire un nouveau concept, l'harmonie préétablie, etc., etc. Mais notre travail ne nous amène pas à être concernés par l'architecture conceptuelle leibnizienne ...

Pluralisme, réalisme, empirisme, une lecture deleuzienne D'autres mouvements philosophiques ont repris cet héritage du concept de relation. Nous devons dès maintenant avertir qu'en ce domaine, nos lectures sont influencées par les analyses de Gilles Deleuze qui leur ont servi de prisme. Cependant, nous pouvons renvoyer à une très belle analyse de Stéphane Malderieux, *Pluralisme anglais et pluralisme américain : Bertrand Russell et William James* [65], où est rendue explicite l'équation "pluralisme = réalisme = empirisme" propre à Deleuze (et qui se trouve principalement dans son ouvrage *Empirisme et subjectivité* [32]).

William James, les termes s'effacent au profit de la relation James reprend la création conceptuelle de Platon – les relations sont des Idées – sous une forme extrême puisqu'à la limite il n'y a, pour lui, plus de termes mais seulement des relations. Autrement dit, les termes sont des faisceaux de relations. C'est ainsi qu'il écrit [54] :

We ourselves are constantly adding to the connexions of things, organizing labour unions, establishing postal, consular, mercantile,

railroad, telegraph, colonial, and other systems that bind us and things together in ever wider reticulations ... From the point of view of these partial systems, the world hangs together from next to next in a variety of ways.

Un tissu de relations Il s'agit donc de modéliser les phénomènes comme un tissu de relations. Un point de vue sur un fragment de ce tissu pourra consister à interpréter les zones fortement connectées comme des termes et celles de densité plus faible comme des relations. Mais, ce qui est considéré comme terme pourra toujours apparaître, dans le contexte d'une nouvelle analyse, comme un faisceau de relations.

La pensée et les relations James (comme Peirce) identifie le domaine de la pensée à la création de relations [55] :

The idea however doesn't immediately leap the gulf, it only works from next to next so as to bridge it, fully or approximately.

Ainsi, pour James, connaître c'est interpréter l'expérience vécue sous la forme de liens non évidents (non immédiatement perceptibles) :

My thesis is that the knowing here is made by the ambulation through the intervening experiences ... Intervening experiences are thus as indispensable foundations for concrete relation of cognition as intervening space is for distance.

David Hume et les deux formes de relations Afin de compléter cette conception, indiquons que David Hume [50] distingue les relations naturelles (concrètes) des relations philosophiques (abstraites).

La relation concrète La relation concrète est le résultat d'une opération automatique qui fonctionne par proximité et qui est faiblement transitive. Prenons l'exemple de la ressemblance. A la vision de la photographie de mon voisin, une relation concrète est créée avec l'homme photographié : je pense à mon voisin. Mais ce type de relation s'épuise vite (c'est ce qu'il fallait entendre par "transitivité faible") : mon voisin me fait penser à ma voisine qui ressemble tellement à une amie d'enfance qui elle-même me fait penser à mon village natal ... Et rapidement la chaîne de relations concrètes s'épuise. Ainsi, la création de relations concrètes est automatique et produit des séries courtes.

La relation abstraite Les relations abstraites sont définies comme étant les circonstances ou les raisons qui amènent à mettre en relation (concrètement) deux idées. Ainsi, les relations abstraites modélisent l'interprétation, l'écoute, l'attention, etc. L'interprétation ne fonctionne jamais par relations concrètes. Par exemple, assister à un cours ne consiste pas à – premièrement percevoir des mots (signes) – deuxièmement associer par relations naturelles les idées qui correspondent aux mots – troisièmement comprendre le sens de l'exposé! En

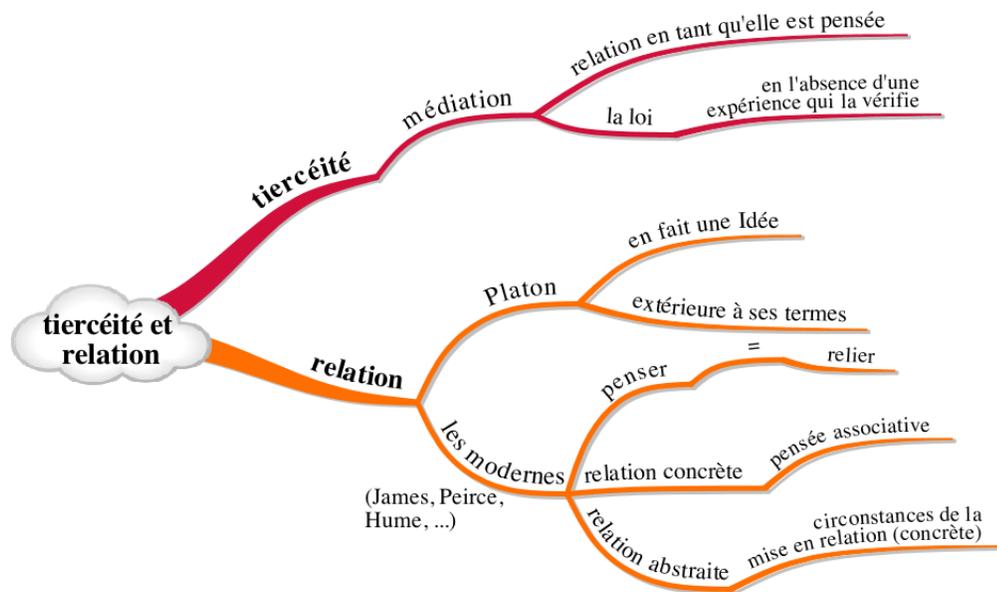


FIGURE 4.4 – Carte de synthèse sur le rapport de la tiercéité et de la relation

fait, ce qui vient d'être décrit est peut-être un modèle simpliste de l'auditeur complètement inattentif : il se laisse porter par le flux des relations concrètes, au final nous dirons de lui qu'il n'a rien compris, qu'il n'a pas écouté. Comprendre, c'est au contraire confronter une perception toujours incomplète (ce qui est réellement entendu de ce qui est dit) à des faisceaux de relations abstraites qui peuvent être assimilés à des "régions du sens". C'est pourquoi deux personnes différentes ne comprennent jamais la même chose, un cours peut fonctionner pour l'un – il aura su se placer aux bons niveaux de sens, il aura su trouver le faisceau de relations abstraites adéquat – mais non pour l'autre.

Tiercéité, relation et ... signe ! Maintenant que nous avons montré comment la tiercéité était du domaine de la pensée, des relations abstraites, de la médiation, nous allons analyser comment pour Peirce le signe est une forme de tiercéité.

Synthèse La figure 4.4 propose une représentation synthétique de cette section.

4.5 La tiercéité et le signe

4.5.1 Quelques définitions du signe

Définitions nominales

Première définition nominale Peirce [71] introduit le concept de signe dans le contexte de la tiercéité. Par exemple³ :

The third is thought in its role as governing Secondness. It brings the information into the mind, or determines the idea and gives it body. It is informing thought, or *cognition*. But take away the psychological or accidental human element, and in this genuine Thirdness we see the operation of a sign.

Seconde définition nominale Ou encore :

The most characteristic form of thirdness is that of a sign ; and it is shown that every cognition is of the nature of a sign.

Signe et connaissance Ainsi, le signe est une tiercéité du type constitutive de connaissance. Mais nous sommes ici en présence de définitions nominales – de la tiercéité constitutive de connaissance – qui ne permettent pas encore de distinguer efficacement ce qu’est un signe.

Troisième définition nominale Nous savons par ailleurs que la tiercéité est le domaine de la relation (médiation), et Peirce introduit aussi le signe en tenant compte de cette dimension :

In its genuine form, thirdness is the triadic relation existing between a sign, its object, and the interpreting thought, itself a sign, considered as constituting the mode of being of a sign. A sign mediate between the interpretant sign and its object. [...] A sign therefore is an object which is in relation to its object on the one hand and to an interpretant on the other, in such a way as to bring the interpretant into a relation to the object, corresponding to its own relation to the object. I might say similar to its own for a correspondence consists in a similarity ; but perhaps correspondence is narrower.

Signe et relation Autrement dit, nous ne pensons pas trahir la pensée de Peirce en affirmant que le signe est une tiercéité qui révèle (ou crée) des relations qui ne sont pas seulement les relations concrètes que nous avons vu précédemment. Le signe amène celui qui l’interprète à la connaissance de ce dont il est le signe en provoquant une référence à des relations abstraites que Peirce nomme interprétant. Le signe ne rend pas seulement compte de relations naturelles – concrètes – mais il permet de faire sens, de comprendre une situation, de rendre

3. <http://www.helsinki.fi/science/commens/terms/thirdness.html>

les relations efficientes. En effet, pour ne pas risquer de confondre avec les relations naturelles qui *agissent* – c'est le domaine de la secondéité, le choc des boules de billards – nous dirons que comprendre une situation c'est rendre les relations qui la constituent *efficientes*. Et rendre les relations efficientes, c'est interpréter les signes. Nous n'échappons pas encore aux seules définitions nominales.

Une définition originale de la connaissance Pour résumer, nous venons de décrire deux aspects qui distinguent le signe d'autres tiercéités : il est constitutif de connaissances et il rend les relations efficientes. Ainsi, nous pouvons à la limite proposer que constituer des connaissances, c'est rendre les relations efficientes !

Ferdinand De Saussure, une approche dichotomique du signe

Un linguiste Suisse En tout cas, et juste avant d'analyser une définition réelle du signe, nous remarquons que Peirce donne à cette notion un parfum qui la différencie nettement de ses cousines homonymes. En particulier, à la même époque, le linguiste suisse Ferdinand de Saussure [87] développe une autre conception du signe.

Langue et parole En tant que linguiste, il s'intéresse au rapport langue/parole. La langue est une construction psychique et social alors que la parole est un acte individuel, un phénomène physique et physiologique. Cependant, l'étude de la langue – qui n'est que pure construction mentale – n'est possible qu'à travers celle des faits de parole. C'est pourquoi Saussure s'intéresse au signe linguistique.

Signifiant et signifié Pour lui – comme pour Peirce et sans doute pour tous les penseurs du signe –, le signe est toujours signe de quelque idée ou chose. De plus, il l'analyse selon deux dimensions inséparables : le signifiant et le signifié. Le signifiant est une représentation d'un phénomène sonore (et à la limite, le phénomène sonore lui-même) tandis que le signifié est une représentation de l'idée ou de la chose dont il est fait signe. Saussure insiste sur le rapport arbitraire ou social du signifiant et du signifié.

La parole privilégiée Par ailleurs, puisqu'il considère le signifiant du signe comme la représentation d'un phénomène *sonore*, il accorde un privilège particulier à la parole, ce qui l'amène à étudier l'inscription du signe dans un contexte avant tout séquentiel et linéaire. Ainsi, il distingue deux types de rapports du signe avec son contexte d'énonciation.

Rapport syntagmatique Le rapport syntagmatique est celui temporel d'un signe avec ceux qui le précèdent et le suivent. Par exemple, pour interpréter le – autrement dit, attribuer une valeur au – signe "émouvante" dans l'énoncé : "Son interprétation reste *émouvante* pour un amateur.", il faut considérer les signes qui l'entourent.

Rapport paradigmatique Le rapport paradigmatique est celui différentiel d'un signe avec tous ceux qui pourraient (ou auraient pu) le remplacer : "Son interprétation reste ___ pour un amateur.". Cet aspect différentiel reflète bien le caractère codé de la langue (nous pensons au codage de Huffman, bien qu'ici il ne s'agisse sans doute pas uniquement d'optimiser le temps de transmission d'un message ...). Mais l'histoire de ce code ne peut se faire qu'au moyen d'une étude de signes linguistiques particuliers qui s'inscrivent (s'énoncent) toujours dans une séquence, sur le modèle de l'acte de parole.

Peirce VS Saussure Finalement, nous remarquons (trop?) facilement que cette conception est fondamentalement binaire et s'oppose à celle, ternaire, de Peirce. Par ailleurs, elle décrit un système fermé de signes linguistiques qui entretiennent des rapports différentiels qui seuls permettent de leur attribuer une valeur. Au contraire, ce rapport prédominant du signe à la parole n'existe pas dans les travaux de Peirce.

Résumé des trois catégories de Peirce

Tableau synthétique Après avoir considéré le signe comme une forme particulière de tiercéité, Peirce propose une définition efficace, réelle du signe. Cependant, avant de la discuter, et puisque nous atteignons un point de l'exposé où nous n'aurons plus à présenter les concepts délicats de priméité, secondéité et tiercéité, nous proposons le tableau synthétique suivant.

	modalité	quantité	caractères
priméité	possible	singularité	qualité, potentialité
secondéité	réel	individualité	fait, état de choses
tiercéité	nécessaire	généralité	loi, sens, relation

Définitions réelles du signe

Première définition réelle Robert Marty a isolé 76 occurrences de définitions du signe dans les écrits de Peirce⁴! Nous tenons ici à montrer ce qui concrètement distingue le signe d'autres tiercéités. Commençons notre analyse sur la base de deux définitions :

A sign, or representamen, is something which stands to somebody for something in some respect or capacity. It addresses somebody, that is, creates in the mind of that person an equivalent sign, or perhaps a more developed sign.

Seconde définition réelle

A sign is a Cognizable that, on the one hand, is so determined (i.e., specialized, bestimmt) by something *other than itself*, called its Object [...], while, on the other hand, it so determines some actual or potential Mind, the determination whereof I term the Interpretant

4. <http://www.cspeirce.com/menu/library/rsources/76defs/76defs.htm>

created by the Sign, that that Interpreting Mind is therein determined mediately by the Object.

Reformulation Ainsi, le signe est quelque chose – que nous appellerons *image*, afin d'insister sur l'aspect de phénomène perceptible, et pour marquer la correspondance avec les analyses de Gilles Deleuze sur les types d'images au cinéma – qui vaut pour une autre image (son objet), par l'intermédiaire d'autres images qui rapportent la première à la seconde (son interprétant).

4.5.2 Le signe triadique : representamen, objet et interprétant

Interprétant

Interprétant VS interprète Une première difficulté tient dans la confusion possible entre interprétant et interprète. Il faut qu'un interprète soit touché par un signe pour qu'en son esprit apparaisse, déterminé par l'objet du signe, un interprétant, c'est-à-dire une série de signes liés au premier et qui en développe, complète ou détermine la signification.

Choix d'un exemple de signe Tout signe peut servir d'exemple. Non seulement un signe écrit. Cependant – et nous suivons (malheureusement) en cela les commentaires de Gérard Deledalle, ainsi que les textes de Peirce – nous choisissons d'illustrer ces définitions au moyen d'un exemple de signe linguistique.

Le signe "Bénin" Prenons le signe "Bénin", en dehors de tout contexte, et en considérant seulement son expression sonore – ce qui implique en particulier de ne pas tenir compte de la majuscule initiale. Son representamen est le mot "Bénin". Son objet peut être le pays d'Afrique ou l'ensemble polysémique qui tourne autour de l'adjectif (doux, bienveillant, propice, non alarmant, ...) – c'est-à-dire les deux homonymes homographes et homophones du mot "Bénin". Pour déterminer la signification du mot "Bénin", il faut au moins un autre signe, appelé interprétant, auquel il renvoie. Par exemple, si "remède" vient à l'esprit de l'interprète, la signification de ce signe suffit à déterminer la signification du signe initial.

Interprétant et signification Les significations des interprétants "Afrique", "Niger", "Burkina Faso", "Porto Novo", "Dahomey", "indépendance", ... détermineraient une autre signification du même signe "Bénin". Qu'est-ce qui permet de choisir parmi tous les interprétants possibles d'un signe? En premier lieu, le contexte d'énonciation du signe, par exemple la proposition dans laquelle il apparaît, peut imposer ce choix.

Influence de l'interprète sur le choix de l'interprétant Mais, l'expérience individuel de l'interprète joue aussi. Nous citons à ce propos Spinoza qui

s'est beaucoup intéressé à la notion de signe, en particulier dans le cadre de son Traité Théologico-Politique où il s'agit, entre autres choses, de la défense d'une lecture (interprétation) de l'ancien testament. Cependant, nous trouvons notre illustration dans une scolie du livre deux de l'Éthique [94] :

[...] par là nous comprenons clairement pour quelle raison l'Esprit, de la pensée d'une chose, tombe aussitôt dans la pensée d'une autre chose qui n'a aucune ressemblance avec la première ; comme, par ex., de la pensée du mot *ponum*, un Romain tombera aussitôt dans la pensée d'un fruit qui n'a aucune ressemblance avec ce son articulé, ni rien de commun avec lui sinon que le Corps de cet homme a souvent été affecté par les deux, c'est-à-dire que cet homme a souvent entendu le mot *ponum* alors qu'il voyait ce fruit, et c'est ainsi que chacun, d'une pensée, tombera dans une autre, suivant l'ordre que l'habitude a, pour chacun, mis dans son corps entre les images des choses. Car un soldat par ex., voyant dans le sable des traces de cheval, tombera aussitôt de la pensée du cheval dans la pensée du cavalier, et de là dans la pensée de la guerre, etc. Tandis qu'un Paysan tombera, de la pensée du cheval, dans la pensée de la charrue, du champ, etc., et ainsi chacun, de la manière qu'il a accoutumé de joindre et d'enchaîner les images des choses, tombera d'une pensée dans telle ou telle autre.

L'habitude C'est en effet pour une grande part l'accoutumance ou l'habitude qui fait que la signification d'un nombre en pratique assez petit de signes (interprétant) détermine la signification d'un signe.

Changements de signification Par ailleurs, de ce qui vient d'être dit, découle en particulier que si la signification d'un mot peut changer avec le temps c'est qu'il y a apparition de nouveaux interprétants. Ainsi, il faut qu'il y ait des situations où l'interprétant naturel – c'est-à-dire lié par l'habitude – d'un signe soit défié par de nouveaux signes qui proviennent de l'expérience et prétendent au statut d'interprétant.

Reprise de l'exemple du signe "force" Nous avons déjà analysé une telle situation précédemment avec l'histoire du concept de force dont la signification passe de "transport de vitesses" à "variation de rapports de vitesses" à l'occasion (pour faire vite) d'interprétations nouvelles d'expériences sur le mouvement d'un corps sur un plan incliné.

Interprétant et relations abstraites C'est pourquoi, les distinctions que nous faisons à l'occasion de notre essai sur la tiercéité entre relations naturelles et relations abstraites s'appliquent parfaitement à l'étude du changement dans la signification d'un signe. Ce rapprochement apparait d'autant plus évident si nous considérons pour un temps le signe, non en lui-même (representamen) ni

par rapport à son objet, mais uniquement par rapport à son interprétant – la série de signes liée au premier qui en détermine la signification.

Interprétant immédiat Alors, le signe considéré par rapport à son interprétant (dorénavant abrégé SI) peut être analysé au moyen des trois catégories de Peirce. Le signe qui correspond à la priméité du SI, Peirce le nomme "rhème" et c'est celui qui, pour son interprétant, est signe de possibilité. Peirce le nomme aussi "interprétant immédiat". Par exemple (emprunté à Deledalle), l'interprétant immédiat de "grenade" sera tout à la fois "ville", "arme" et "fruit". Car en effet, la priméité est le domaine de la pure possibilité. En logique formelle, le rhème correspond directement au terme.

Interprétant dynamique Le signe qui correspond à la secondéité du SI, Peirce le nomme "dicensigne" et c'est celui qui, pour son interprétant, est signe d'existence réelle. Peirce le nomme aussi "interprétant dynamique", et il correspond assez bien à la proposition en logique formelle. Par exemple, "Grenade est une ville en Espagne." est un dicensigne, un interprétant dynamique du signe "Grenade". Au dicensigne correspond ainsi la relation naturelle qui, rappelons le, est automatique et s'épuise assez rapidement : "Grenade" -> "Ville" -> "Espagne" -> ...

Interprétant final Le signe qui correspond à la tiercéité du SI, Peirce le nomme "argument" ou "interprétant final". Son correspondant en logique formelle est le raisonnement. C'est lui qui assure la clôture de la série de l'interprétant dynamique. C'est donc un mécanisme qui contrôle le développement des interprétants dynamiques. Il peut prendre deux formes. D'abord, l'habitude, qui est une forme d'apprentissage dont le résultat est la reconnaissance de contextes, l'association automatique d'une série finie de signes (l'interprétant final) à un signe lorsqu'il est reconnu dans un certain contexte. Ensuite, une seconde forme qui nous intéresse au plus haut point, il s'agit pour l'interprète de clore l'interprétant dynamique d'une telle manière que la situation devienne compréhensible. Autrement dit, c'est se placer au bon "niveau de sens", choisir le bon "faisceau de relations abstraites". Par exemple, lorsqu'apparaît plus haut le signe "Grenade est une ville en Espagne", si l'interprète de ce signe clôt l'interprétant dynamique au moyen par exemple des signes "Reconquista"->"Rois catholiques" alors il n'a pas compris ce dont il s'agit ; par contre si un interprétant final du type "dicensigne"->"interprétant dynamique"->"proposition" apparaît à son esprit alors il a compris.

Objet

Une analyse similaire à celle qui vient d'être proposée pour le SI, peut être menée au sujet du signe considéré par rapport à son objet (dorénavant abrégé SO).

Icône Le signe qui correspond à la priméité du SO, Peirce le nomme icône : c'est un signe qui renvoie à l'objet par des qualités qui lui sont propres (par exemple la ressemblance de la photographie d'un homme avec cet homme).

Indice Le signe qui correspond à la secondéité du SO, Peirce le nomme indice. C'est un signe qui n'existerait pas sans l'existence de son objet, autrement dit, il renvoie à son objet parce qu'il est affecté par cet objet (par exemple, la fumée qui est indice du feu).

Symbole Enfin, le signe qui correspond à la tiercéité du SO, Peirce le nomme symbole. C'est un signe qui renvoie à l'objet en fonction d'une loi ou d'une habitude.

Representamen

Nous pourrions mener une troisième et dernière analyse ternaire au sujet du signe considéré en lui-même (ou representamen), mais cela nous mènerait à affronter de nouvelles et importantes difficultés qui ne serviraient pas notre propos.

4.5.3 Conclusion

Nous allons conclure ce chapitre par un court essai sur le concept de *différence* tel que développé par Gregory Bateson. Nous montrerons comment la différence est très similaire au signe triadique de Peirce. Nous espérons par ailleurs que, malgré l'apparente froideur conceptuelle qui aura pu transparaître de ce chapitre bientôt clos, seront passées les idées que nous avons crues nécessaires à l'ouverture d'un point de vue neuf sur l'inscription des médiations symboliques (programmes) de l'informatique dans un champ sémiotique qui finalement n'est rien sinon notre quotidien.

Synthèse La figure 4.5 propose une représentation synthétique de cette section.

4.6 La différence et le signe

4.6.1 Perspectives de l'étude

Gregory Bateson Gregory Bateson a développé une conception extrêmement originale de la différence. Gilles Deleuze était très intéressé par les travaux de Bateson (cela ne fait aucun doute, on trouve des références explicites à l'œuvre de ce dernier dans les enregistrements audio des cours du premier ⁵).

5. <http://utime.unblog.fr/2007/11/01/rencontre-deleuze-bateson-un-langage-des-relations/>

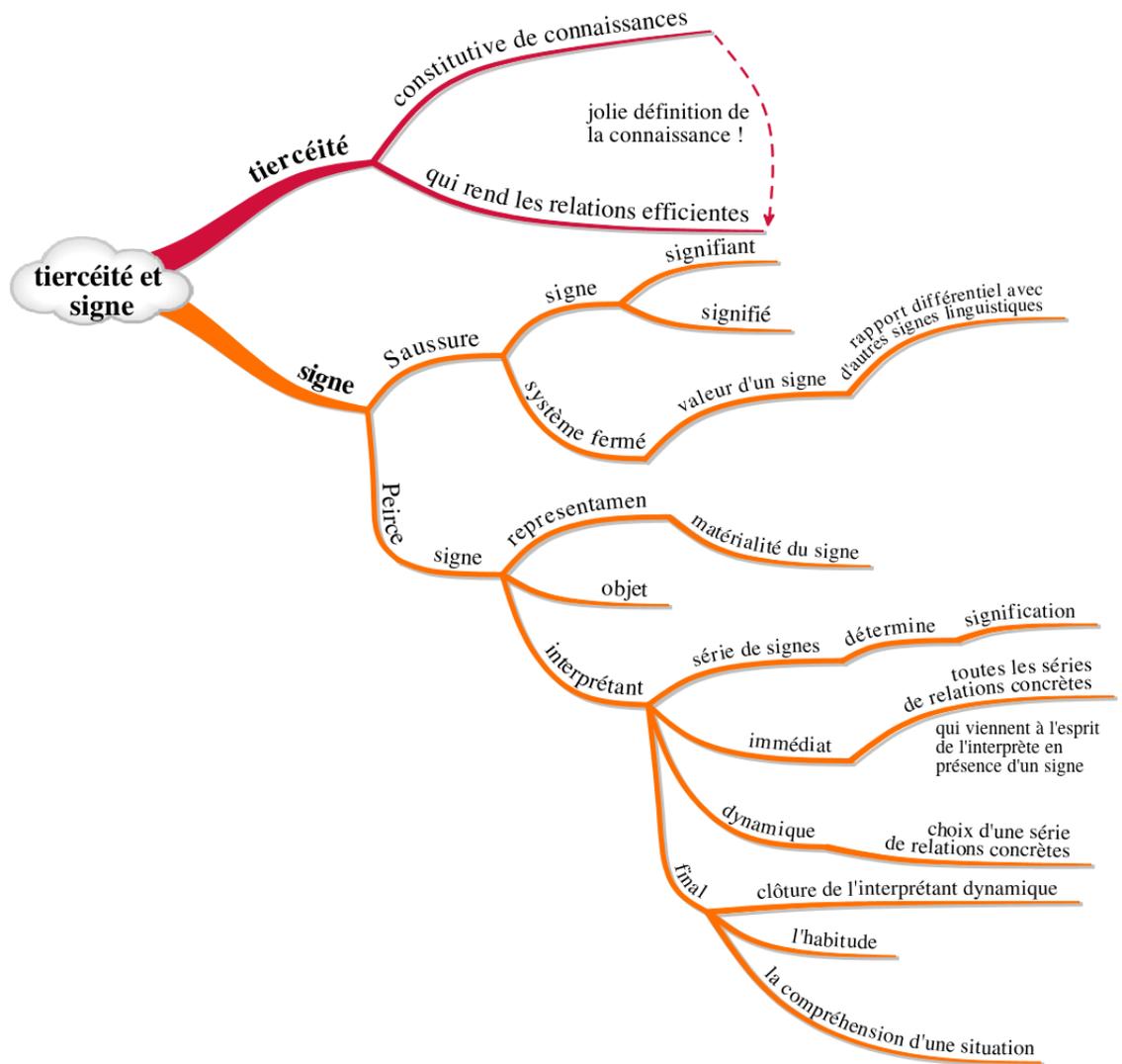


FIGURE 4.5 – Carte de synthèse sur le rapport de la tiercité et du signe

Relations avec notre interprétation de la pensée de Gilles Deleuze

Or, lorsque nous avons expliqué dans les sections précédentes en quoi le signe est une forme particulière de tiercéité, nous avons eu besoin d'une conception précise de la relation que nous avons trouvé au croisement de lectures très diverses qui — de Platon à Peirce, en passant par William James ou David Hume — étaient orientées par les interprétations de Deleuze. Ainsi, rien d'étonnant à ce que nous ayons cherché à mieux comprendre le travail de Bateson.

Bateson et le concept de différence Nous avons eu alors le bonheur de découvrir deux éléments essentiels qui offrirent un nouvel éclairage à une partie importante de notre recherche ! Premièrement, cette notion de *différence* qui, comme nous allons l'expliquer dans un instant, répète (et donc précise, assure, etc.) les découvertes faites sur la notion de signe telle que conçue par Peirce.

Bateson et la notion d'apprentissage Deuxièmement, une conception extraordinaire de la notion de *contexte*, développée dans le cadre d'une étude sur les différents niveaux d'apprentissage, et qui sera au cœur de notre prochain chapitre qui traite des bienfaits d'une utilisation de représentations de la trace des actions d'interaction.

4.6.2 La différence comme relation

Où chercher la différence ? Selon une première approche, naïve mais nécessaire, la différence est une sorte de relation. En tant que telle, elle se doit d'être, comme nous l'avons montré plus haut, extérieure à ses termes. Ce point est établi très clairement par Bateson [3] qui, en ce sens, se rattache aux *penseurs de la relation* :

Une explication de Bateson

But what is a difference? A difference is a very peculiar and obscure concept. It is certainly not a thing or an event. This piece of paper is different from the wood of this lectern. There are many differences between them—of color, texture, shape, etc. But if we start to ask about the localization of those differences, we get into trouble. Obviously the difference between the paper and the wood is not in the paper; it is obviously not in the wood; it is obviously not in the space between them, and it is obviously not in the time between them. (Difference which occurs across time is what we call "change.") A difference, then, is an abstract matter.

Avant-goût S'entendre dire de la différence qu'elle est matière abstraite ... Rien ne pouvait mieux nous satisfaire ! Car — mais il faudra le démontrer —, résonne déjà (à nos oreilles, s'entend) cette autre expression par ailleurs introduite ... les *relations abstraites* ... et, maintenant nous le savons, non loin repose le signe !

4.6.3 Différence dans les mondes physique et mental

Introduction à l'étude Sans suivre très exactement les développements de Bateson, nous pensons lui rester fidèle, en continuant d'introduire — mais plus précisément — son concept de différence au moyen d'une analyse de la différence — car il nous faut bien employer ce même mot de "différence" et éviter d'échouer trop tôt au rivage des synonymes — entre ce qui est entendu par "effet" dans le contexte d'abord physique puis mental.

L'effet physique L'effet est très certainement ce dont on sait l'existence dépendre de certaines conditions ou événements concrets (comme un impact, une force, ...).

"L'effet" pour le monde mental Mais l'effet, dans le "monde mental", est ce dont l'existence dépend de la nouvelle d'une différence. Ainsi, l'exemple de la lettre que je m'attendais à recevoir mais que je n'ai pas reçue. La pensée me vient de ne pas avoir reçu la lettre attendue, c'est la nouvelle d'une différence — une cause qui semble ne reposer sur aucun support matériel. Cette pensée pourra en déclencher toute *une série* d'autres (qui à leur tour etc. etc. nous retrouvons l'interprétant) : autrement dit, elle fera (au moins) une différence.

La différence qui fait une différence C'est pourquoi, pour Bateson, la relation cause/effet transposée au monde mental se résume dans le fameux énoncé : "une différence qui fait une différence". L'unité d'information, l'idée élémentaire, c'est la différence qui fait une différence. En ce qui nous concerne, nous pouvons tout aussi bien et sans risques appeler "signe" la "différence qui fait une différence" car il semble bien s'agir, à la lettre, d'une tiercéité (domaine du mental, de la pensée) qui rend les relations efficientes. Ainsi, reprenant avec humour la fameuse expérience de pensée de Kant avec son morceau de cire, Bateson [3] écrit :

Humour américain

I suggest that Kant's statement can be modified to say that there is an infinite number of *differences* around and within the piece of chalk. There are differences between the chalk and the rest of the universe, between the chalk and the sun or the moon. And within the piece of chalk, there is for every molecule an infinite number of differences between its location and the locations in which it *might* have been. Of this infinitude, we select a very limited number, which become information. In fact, what we mean by information—the elementary unit of information—is a *difference which makes a difference*, and it is able to make a difference because the neural pathways along which it travels and is continually transformed are themselves provided with energy. The pathways are ready to be triggered. We may even say that the question is already implicit in them.

Différence et énergie Ici, Bateson, en introduisant la notion d'énergie, nous permet de progresser dans notre compréhension du signe considéré à la "frontière" des "mondes" mental et physique.

4.6.4 La carte et le territoire

Du territoire à la carte Afin de nous faciliter la tâche, considérons à la lumière de ce nouveau point de vue le fameux énoncé de Korzybski : "la carte n'est pas le territoire". Qu'est-ce que la carte retient du territoire ? A l'évidence, un certain nombre de différences. Différences d'altitude, de climat, de vitesse d'exploration, etc.

La sélection des différences Qu'est-ce qui préside à la sélection d'un ensemble limité de différences quand un nombre toujours immense d'entre elles était à portée d'observation ? Sans aucun doute l'utilité déduite d'un usage présumé : sont conservées uniquement ces différences qui seront censées faire une différence pour un lecteur modèle de la carte. Comment s'opère la collecte des différences ? Par la médiation d'instruments d'observation. Et d'un observateur n'est-ce pas ? A strictement parlé, non ! Mais précisons ce qui doit être entendu par "instrument d'observation" ...

La nature de l'observation Il s'agit d'un support pour la circulation et la transformation des différences. Ainsi, un hypsomètre relève la nouvelle d'une différence de températures d'ébullition et la transforme en une différence de pression barométrique puis en une différence d'altitude dont la nouvelle est à son tour relevée sur l'instrument (qui émet de l'énergie, des photons) par une rétine qui la transforme en une différence chimique (grâce à ses photorécepteurs) puis en une différence électrique (grâce à ses neurones) dont la nouvelle est transportée par le nerf optique à l'encéphale etc. etc. Notre monde phénoménal : un circuit merveilleux de transport et de transformation de différences !

La vision est action Toutes les dichotomies traditionnelles semblent évanouies. Ainsi, et seulement pour exemple, nous pouvons affirmer que voir c'est agir. Je vois un serpent, j'ai peur (c'est l'exemple emprunté à la tradition hindoue par le scientifique spécialiste de neurosciences Joseph E. LeDoux [62]). C'est-à-dire qu'une information sensorielle est créée par mon thalamus qui transfère la nouvelle à mon complexe amygdalien qui s'activant va déclencher des tremblements, une hausse de mon rythme cardiaque, un changement dans mon rythme respiratoire, peut-être vais-je me figer ou au contraire faire un pas en arrière ? etc. etc. Du temps passe et ce circuit est à nouveau parcouru mais cette fois, la nouvelle de la différence après avoir passé par le thalamus transite par le cortex avant d'être redirigée vers l'amygdale. La différence change de nature, en fait, ce n'était pas un serpent mais une corde : mes cheveux retombent sur mon crâne, ma respiration retrouve son rythme etc. etc. La vision est action !

Mirage et territoire Autrement dit, il n'y a rien qui soit de la nature du territoire en soi. Il n'y a rien sinon des cartes, et des cartes de cartes, et des cartes de cartes de cartes ... Finalement, les éléments de la carte, les différences qui font une différence, sont les signes que nous sommes parvenus à définir dans les sections précédentes.

La légende de la carte et les relations abstraites Et pour clore l'analogie, remarquons que la légende de la carte qui représente des différences de différences correspond bien à ce qui était appelé plus haut les "niveaux de sens" ou "faisceaux de relations abstraites". Pour comprendre une carte, il est nécessaire de connaître sa légende. En général, pour interpréter les signes, il est nécessaire de se placer aux bons "niveaux de sens" (nous aurons l'occasion de revenir sur cet aspect lors de notre étude de la trace).

4.6.5 Différence et énergie

Réseau et énergie Nous expliquions plus haut que la considération par Bateson du rapport entre énergie et différence allait nous permettre de mieux comprendre le fonctionnement d'un système de signes. En effet, nous avons montré que pour qu'une différence fasse une différence (pour qu'une relation, pour qu'un signe soit efficient) un réseau où circule l'énergie doit toujours être présent. De plus nous avons illustré clairement comment un corps (quel qu'il soit) était partie intégrante de ce réseau.

La limite des corps Aussi, commençons-nous par remarquer que, du point de vue de ce circuit de différences, la frontière d'un corps n'a aucune raison de s'identifier à une membrane (telle que la peau). Cependant, dans les exemples que nous venons de donner, des différences dans la manière d'encoder et de transmettre les différences peuvent permettre de distinguer des corps. En effet, les systèmes nerveux sont des circuits toujours alimentés en énergie et qui sont prêts à recevoir la nouvelle d'une différence sans que cette dernière ait besoin de fournir sa propre énergie (voir l'exemple donné plus haut de la lettre jamais reçue). C'est sans doute ce qui nous pousse à distinguer si nettement ce qui relève du "monde mental" de ce qui relève du "monde physique".

Ecologie de l'esprit Mise à part cette différence qu'il faut garder à l'esprit, nous pouvons ainsi étudier les conditions sous lesquelles ce réseau de transport et de transformation des différences (ou signes) est découppable en sous-systèmes que Bateson nomme "esprits". Mais ce critère de découpage, nous l'avons déjà donné au sujet des cartes et de leurs légendes ... il s'agit de rendre les phénomènes explicables !

Un aveugle provençal Soit un homme aveugle qui marche en ville. Sa canne percute le sol et transporte des différences qui lui servent à repérer son environnement. Maintenant, assis au café, il écoute se dérouler une partie de pétanque

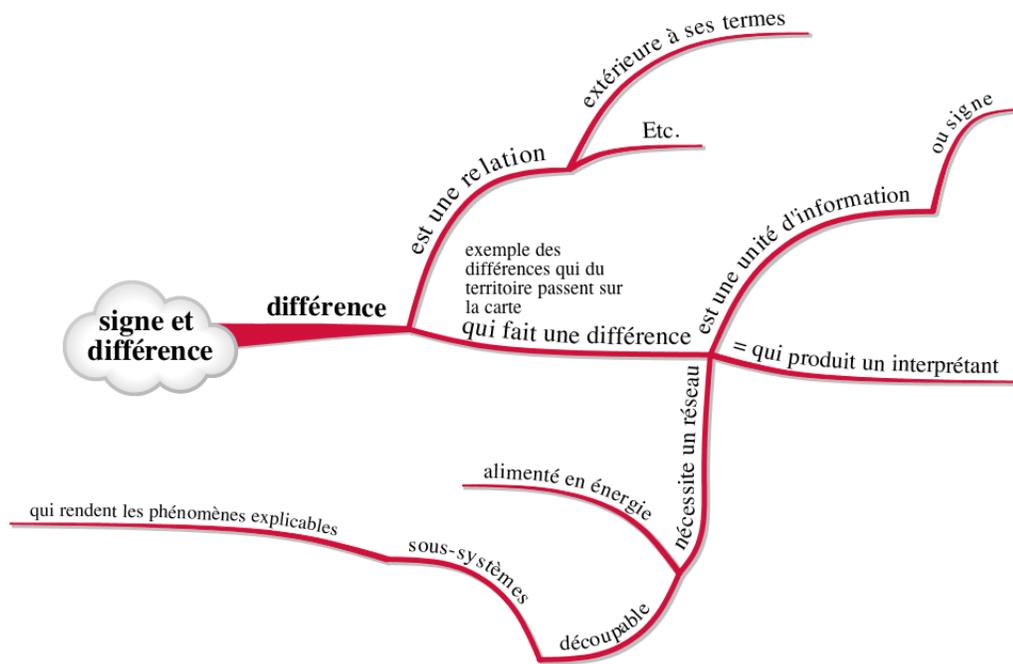


FIGURE 4.6 – Carte de synthèse sur le rapport du signe et de la différence

et sirote un jaune. Sa canne repose, calée contre la table. Pour comprendre la première situation, la canne doit faire partie intégrante du système observé. Dans le cas de la deuxième situation, la canne n'appartient évidemment pas au système.

4.6.6 Conclusion

L'objectif de cette section était de réaffirmer autrement l'essentiel du concept de signe. Comme l'écrit Montaigne :

Il y a plus affaire à interpréter les interprétations qu'à interpréter les choses ; et plus de livres sur les livres, que sur aultre subiect : nous ne faisons que nous entregloser.

Synthèse La figure 4.6 propose une représentation synthétique de cette section.

4.7 Un dialogue pour conclusion

Alea iacta est. C'est l'heure du bilan. L'auteur accorde un entretien.
— Et à quoi vous a servi, monsieur Portier, ce chapitre étrange ?

— En principe, et premièrement, à rendre intelligible le concept de signe dans le complexe de son histoire récente.

— Admettons. Mais comment situer cette "histoire de signes" dans le contexte de votre recherche ?

— D'abord, et selon une perspective relativement naïve, un programme informatique peut être considéré comme un support pour la circulation et la transformation de signes.

— Où voulez-vous en venir ? Votre dernière proposition est difficilement critiquable. Elle me semble si générique qu'elle perd tout intérêt !

— Un instant ... Cette étude a permis de comprendre un peu mieux ce qu'il en était de la signification d'un signe. En particulier, à l'occasion de la rencontre avec ce concept intrigant et trompeur d'interprétant. Car, je vous le rappelle, il ne s'agit surtout pas de l'interprète du signe. D'ailleurs, il est plutôt question de faire disparaître ces notions d'interprète et d'interprété au profit d'une configuration dynamique de signes, d'un réseau de signe, de ...

— Je vous coupe. S'il vous plaît de ne pas céder à ce penchant pour la digression, nous aurons peut-être une chance de nous entendre ! Vous parliez de l'interprétant. En quoi cette notion vous est-elle utile ?

— L'utilité. Oui, oui. Vous ne m'y prendrez plus. Je répondrai au plus juste. Je serai économe.

— Ne le prenez pas mal ...

— Non, non. Donc, l'interprétant ... Vous savez, il s'agit de ces séries de signes qui, liées au signe perçu, permettent d'en développer, compléter ou déterminer la signification. Souvenez-vous des exemples de signes linguistiques tels que "bénin" ou "grenade".

— C'est la question de savoir s'il s'agit de la ville espagnole, de l'arme ou du fruit. N'est-ce pas ?

— Exactement. Ce qui m'a vraiment intéressé, c'est la distinction que fait Peirce de trois qualités d'interprétants. L'interprétant immédiat, l'interprétant dynamique et l'interprétant final.

— Ah, Peirce ... vous l'aimez bien !

— C'est vrai. Un drôle de personnage. Enfin ... J'essayais de rappeler la distinction entre trois formes d'interprétants. L'interprétant dynamique ce sont les séries de signes qui apparaissent pour un interprète au contact d'un signe en l'absence d'un contexte bien défini.

— Peut-être qu'un exemple pourrait ... ?

— Bien ... hmmm ... ah oui ! Elle et lui flânent dans un parc. Sur le bord du chemin, il aperçoit un bout de papier froissé. Il le ramasse. C'est un papier aux bords déchirés sur lequel est écrit : "grenade". Il ne sait pas de quoi il s'agit. Il aimerait comprendre. Il dit à haute voix : "grenade". Son amie enchaîne sur la description du souvenir d'un séjour qu'ils passaient ensemble en Espagne quelques années auparavant. Elle vient de faire le saut d'un interprétant immédiat qui reste ouvert à toutes les interprétations vers un interprétant dynamique qui fige une signification particulière : grenade, Espagne, voyage, vacances, ... Lui se refuse à faire ce saut car il désire comprendre le sens de ce signe. Il en reste à l'interprétant immédiat, il aurait besoin d'autres signes pour faire un

interprétant dynamique. Il est attentif, son point de vue sur son environnement est dirigé par cette volonté de savoir, il veut d'autres signes à interpréter. Un peu plus loin sur le chemin, à moitié recouvert de débris, émerge d'une poubelle un carnet noir. Il le feuillette. Il y a des pages manuscrites, des schémas ... il ne comprend pas grand chose. Par moment on croirait à des mathématiques ou à de la biologie, mais tout à côté on dirait bien de la philosophie (en tout cas on peut lire des noms de philosophes) ... ou serait-ce de l'informatique ? Il y a des schémas d'architecture, quelques algorithmes, de drôles de graphes ... Ah, voilà une page déchirée qui s'accorde parfaitement au petit papier trouvé plus tôt ! Il est heureux car il a de quoi s'amuser : tout ces signes qu'il va s'agir d'arranger afin de donner sens à cette "grenade". S'il parvient à construire quelque chose de cohérent alors il faudra parler d'interprétation final. Voilà.

— Quel comédien ! Et l'informatique dans tout ça ?

— Vous disiez ce chapitre étrange. Peut-être est-ce dû en effet au manque de présence explicite de la discipline informatique ? J'ai pourtant essayé d'expliquer en quoi la méthodologie de construction des documents multistructurés, proposée dans le chapitre précédent, demandait, pour être toute entière comprise, l'éclairage de cette drôle de théorie du signe. En résumé, il s'agissait de remarquer qu'utiliser la détection automatique du chevauchement de termes d'annotation pour inciter à la création d'un nouveau vocabulaire consistait bien à encoder l'effet d'une lecture sur la compréhension d'une œuvre en précisant la signification d'un signe (le terme d'annotation dont l'ajout a provoqué un chevauchement).

— C'est intéressant. Je reste un peu sur ma faim, mais ...

— Attendez la suite ! L'utilisation à venir de la trace des actions d'interaction fera plein usages des idées développées ici.

— D'accord.

Chapitre 5

Évolution et appropriation des classes de termes et des vocabulaires de relations au moyen d'un modèle réflexif des actions d'annotation

5.1 Introduction

Bilan Nous avons expliqué comment notre notion de documents multistructurés consiste en un ensemble de structures documentaires (des arborescences de termes d'annotations) dont des éléments peuvent entrer dans des relations quelconques avec d'autres éléments de structures documentaires ou avec des fragments d'images, etc. pour finalement former un graphe orienté que nous avons appelé rhizome. Nous avons proposé de partitionner l'ensemble des termes d'annotations en classes et l'ensemble des relations en vocabulaires. Nous avons conçu une méthodologie pour la construction des documents multistructurés permettant d'assister rationnellement la construction des classes.

Un modèle réflexif des actions d'annotation Cependant, le modèle de représentation et de manipulation des documents que nous proposons, de part sa grande ouverture, demande à être accompagné d'un mécanisme de documentation et de contrôle des classes de termes et des vocabulaires de relations. Un tel mécanisme, nous pensons l'avoir trouvé dans une représentation manipulable des actions d'annotation. C'est-à-dire que nous proposons aux utilisateurs une représentation de leurs propres activités d'annotation et que nous leur offrons les outils permettant de manipuler cette représentation. Nous allons montrer

comment cette introduction d'une activité réflexive participe à maintenir la cohérence des produits de notre système dynamique de construction de documents multistructurés.

Plan Des travaux, bien que peu nombreux, indiquaient déjà cette direction, et nous commencerons par en proposer une synthèse. Ensuite, nous construirons, sur l'ouverture déjà entamée au chapitre précédent, un nouveau point de vue sur les notions de contexte, de trace et d'apprentissage. À partir de cette nouvelle perspective, nous pourrons présenter notre modèle et notre représentation des actions d'annotation.

5.2 Approches existantes

Introduction Relativement peu de travaux se sont intéressés explicitement à des systèmes réflexifs qui offrent aux utilisateurs une représentation manipulable de leurs propres actions d'interactions. Cependant, si l'idée nous est venue d'expérimenter une telle approche (qui s'est d'ailleurs révélée pertinente), il a bien fallu que des études antérieures nous y conduisent. Nous allons maintenant les présenter. Il s'agit d'abord d'un travail sur l'importance du développement de modèles réflexifs pour la conception des systèmes interactifs qui visent à faciliter la collaboration des utilisateurs [40]. Ensuite, nous avons été intéressés par une analyse de la notion de contexte [23]. Finalement, des travaux proposent une modélisation de systèmes à base de traces qui se révéleront proches de notre solution [60][89].

5.2.1 Réflexivité

Problématique Paul Dourish [40] est sans doute l'un des premiers chercheurs à avoir étudié les systèmes interactifs dans leurs *contextes* d'utilisation et non sur la seule base de cas d'utilisations déterminés a priori. Ainsi, il isole trois dimensions qu'il juge importantes et qu'un modèle *a priori* d'utilisation ne peut pas embrasser :

- l'adaptation et amélioration de son environnement par l'utilisateur lui-même
- la prise en compte des aspects sociaux ne permet plus la prédiction absolue des interactions et des activités qui se déploieront autour du système interactif
- la co-adaptation des systèmes et des pratiques ; nous avons déjà introduit ce point dans notre exposé, les pratiques se voient transformées par l'introduction d'interfaces Homme-Machine au moins autant qu'elles participent à la transformation de ces interfaces

Ainsi, Dourish s'intéresse aux moyens nécessaires pour une conception dynamique des systèmes interactifs afin de prendre en compte les aspects précédents. Mais d'abord, il tente d'isoler certaines caractéristiques saillantes de ces nouveaux systèmes :

- *infrastructures ouvertes* – dans l'idéal, il devrait s'agir de systèmes de base génériques que les utilisateurs peuvent transformer, au niveau des fonctionnalités comme au niveau de la présentation
- *dynamiques* – il faut que ces systèmes puissent adapter leurs réactions à des patterns d'usage
- *évolutifs* – il s'agit de permettre la co-adaptation : les transformations apportées par les utilisateurs doivent être conservées du passage d'une version du système à une autre

Notre lecture de la problématique Finalement, nous pensons pouvoir synthétiser ces nouveaux besoins sous la forme d'une proposition adressée aux concepteurs de systèmes interactifs (c'est-à-dire une proposition que nous nous adressons !) : l'intérêt devrait se porter sur rendre interactives les opérations de construction et de manipulation des modèles de domaine plutôt que sur leur construction a priori.

Vers une solution Pour en revenir aux considérations de Dourish, il pense trouver le germe d'une solution pour cette nouvelle conception des systèmes interactifs dans la notion de *réflexivité*. Il approche cette notion au moyen de la réflexivité d'un programme informatique. Il s'agit d'un programme qui possède un modèle de son propre comportement. De plus, il peut modifier ce modèle et par là changer son comportement.

3-LISP L'exemple le plus extrême d'un langage qui permet une telle réflexivité est sans doute 3-LISP [85]. Il s'agit en effet d'un dialecte de LISP pour lequel l'interpréteur est lui-même représenté par un programme manipulable ! C'est-à-dire qu'il est par exemple possible de changer, à l'exécution, l'implémentation du mécanisme de gestion des variables (par valeur, par référence, ...) ; ou encore d'accéder à la pile des appels de fonctions qui ont abouti à l'exécution de l'instruction courante et de modifier cette pile, etc.

MOP Si le cas de 3-LISP reste assez confidentiel, le protocole à méta-objets (Meta-Object Protocole, MOP) est un exemple de plus haut niveau qui possède de nombreuses applications. Il s'agit d'un modèle manipulable du modèle de la programmation orientée objet. Les exemples les plus connus de MOP sont CLOS [12] et Smalltalk [84][43]. Dans tous les cas, le programmeur peut modifier la sémantique du langage. Par exemple, il peut ajouter la gestion de (ou modifier la manière dont est géré) l'héritage multiple.

Application À partir de ces réflexions sur la réflexivité des langages de programmation, l'auteur développe une bibliothèque logicielle pour la conception de systèmes CSCW (Computer Supported Cooperative Work). Sa solution est donc très orientée en direction des programmeurs et ce malgré le manifeste sur lequel s'ouvre sa recherche et qui promeut une nouvelle et grande ouverture

des systèmes interactifs. Malgré tout, son approche règle un certain nombre de problèmes techniques :

- *la répartition des données* – certaines données peuvent être stockées localement, d'autres sur un serveur . . . la bibliothèque n'impose aucune solution et permet même d'adapter le choix du lieu de stockage dynamiquement.
- *la gestion des conflits* – selon la nature du domaine applicatif, les conflits à l'occasion d'accès concurrents à une même ressource doivent être gérés différemment. Si la sécurité des données est un facteur essentiel (lorsqu'il s'agit par exemple de concevoir les plans d'une machine), des verrous explicites sont nécessaires. Dans d'autres cas (par exemple, un tableau blanc pour une activité de brainstorming), une harmonisation a posteriori des données est plus adaptée. La nature réflexive de la librairie permet au programmeur de s'adapter à ces diverses situations.
- *Etc.*

Conclusion Nous pensons aussi que c'est par l'introduction de propriétés réflexives que nous pouvons assurer une forme de contrôle sur l'évolution de systèmes très ouverts tels notre plateforme de construction de documents multistructurés. Cependant, en ce qui nous concerne, cette réflexivité ne peut pas s'appliquer à l'unique niveau de la conception des systèmes (c'est-à-dire concerner les seuls programmeurs) mais doit faire partie intégrante de l'utilisation du système. Nous montrerons que cette nouvelle forme de réflexivité permet un apprentissage de l'usage des relations et des termes, et qu'elle participe à la mise en œuvre de cette forme légère de contrôle de l'évolution des classes et des vocabulaires.

5.2.2 Contexte

Problématique Prenant en quelque sorte la suite des travaux de Dourish sur la notion de réflexivité, sont apparues de nombreuses recherches centrées sur la notion de *contexte*. Mais peu de travaux s'intéressent à un contexte qui ne soit pas seulement constitué d'éléments facilement formalisables (tels que les enregistrements de capteurs physiques) et destiné à permettre des calculs, c'est-à-dire des traitements automatiques. Parmi ces derniers, [23] rend compte explicitement de ces deux points de vue sur le contexte : des structures de données destinées au calcul d'une part, et des représentations qui deviennent partie intégrante du processus d'interprétation de l'utilisateur d'autre part. Nous pouvons facilement concevoir l'influence des travaux autour de la réflexivité sur ce second point de vue.

Ethnométhodologie Chalmers [23], désirant concilier ces deux points de vue, cherche un fondement théorique qui pourrait l'aider. Il se tourne dans un premier temps vers l'ethnométhodologie. Il s'agit de l'étude des méthodes qui dirigent en partie nos activités ordinaires. Remarquons, en aparté, que pour les défenseurs de cette discipline, l'essence de l'outil serait sa disparition. L'outil a à se rendre

invisible. Or, lorsque nous défendions, à la fin du précédent chapitre, un nouveau point de vue sur la notion de corps (il s'agissait d'expliquer que les frontières d'un corps ne peuvent se définir que localement et afin de permettre la circulation des différences qui rendent la situation étudiée explicable), il était plutôt question d'ouvrir le fonctionnement de l'outil à l'utilisateur. Sans cette ouverture, le couplage Homme-Machine ne nous semble pas possible. Cependant, le résultat de ce couplage revient en effet à une disparition de l'outil . . . mais aussi de l'utilisateur ! Fin de l'aparté. Chalmers retient en tant qu'aspects positifs de l'ethnométhodologie la situation centrale offerte aux *actions* humaines, l'intégration de la réflexivité et l'étude des actions *dans leur contexte*. Mais il déplore que ce contexte soit toujours celui d'un sujet présent et que peu d'intérêt soit réservé à la répétition des contextes. C'est-à-dire qu'il aimerait considérer que des actions entre elles différentes se jouent au sein de contextes similaires. Nous montrerons, en ce qui nous concerne, que l'hypothèse d'une identité de certains contextes est nécessaire au développement de toute notion d'apprentissage.

Cercle herméneutique N'ayant pas trouvé entière satisfaction auprès de l'ethnométhodologie, Chalmers se tourne vers la théorie du *cercle herméneutique* de Gadamer [44]. En résumé brutal, il s'agit, entre autres points, de proposer que les interprétations présentes (aussi bien que les actions, que les instants de compréhensions, etc.) sont influencées par les expériences passées et en même temps influencent (ou mettent en forme) ces mêmes expériences. Autant dire – mais c'est certainement déformer la pensée de Gadamer – que la pensée du passé ne peut qu'être toujours présente ! Ainsi, Chalmers propose de faire de la représentation des expériences d'interactions passées une ressource disponible pour les interactions présentes.

Expérimentations L'auteur [23] teste cette idée au moyen d'une expérience de visite instrumentée et en binômes de la ville de Glasgow. Un terminal mobile est fourni aux utilisateurs. Il dispose d'un récepteur GPS, d'un récepteur Wifi, d'un appareil photographique, de la représentation d'une carte de la ville décorée d'objets (les autres utilisateurs, les monuments, etc.), d'un navigateur Web, . . . De plus, l'historique des utilisations est analysé en fonction du contexte courant pour trouver des situations similaires qui permettent de recommander des actions à l'utilisateur. Enfin, chaque utilisateur se voit présenter une représentation du contexte de son binôme, ce qui a permis de vérifier le bénéfice de l'influence mutuelle introduite par un partage de contextes.

Bonne pratique Finalement, Chalmers en déduit une bonne pratique pour la conception des systèmes interactifs :

A general system design goal we suggest here is to interconnect objective representations of system structure with other more subjective and "historical" representations. In each would be tools for abstraction, selection and so forth. For example, class hierarchies and database table structures might possibly be used in the first case,

while clusters, selections and time windows might be useful abstractions over past use of the same system. One could concentrate on and interact with particular components or elements of system structure, and the system could highlight in a neighbouring display the patterns of past use of those selected components : where they were used, who used them, and so forth. [...] Note that focusing on the objective component could trigger the subjective component to act as a "contextual" display, and vice versa, so that neither is primary : users could use activity in one to feed into the other and then back again.

Conclusion Ce que nous retenons en particulier de cette proposition, est l'interconnexion des représentations objectives de la structure du système avec d'autres représentations (que Chalmers qualifie de "subjectives" en tant qu'elles proviennent du résultats d'interactions d'un utilisateur avec le système) des usages du système. Remarquons que la présence des représentations de la structure interne du système n'a pas à être continue. Nous pensons même qu'il est utile de la laisser discrète. Autrement dit, de temps en temps, l'outil se fait explicitement présent en tant qu'outil, et ces moments peuvent servir à approfondir le couplage Homme-Machine. C'est ce que nous avons proposé au moyen de la contrainte d'arborescence imposée aux instances des termes d'une classe. Le respect de cette contrainte demande au système de se faire parfois présent en proposant à l'utilisateur de restructurer ses termes d'annotation.

5.2.3 Modèle de la trace des actions d'interaction

Problématique Un petit nombre de travaux [60][89] traitent explicitement du problème de la représentation et de la manipulation de la trace des actions d'interactions. Commençons par quelques remarques générales afin de situer la problématique. La trace informatique est nécessairement un objet *construit*, autrement dit, le résultat de l'exécution d'un programme. Cette construction produit une séquence chronologique d'observés typés. C'est au concepteur du système tracé de déterminer quels seront les observés. Avant d'entrer dans la description des solutions existantes de modélisation de la trace informatique, nous pensons utile d'entamer une courte discussion sur la notion plus générale de trace.

Dictionnaire À interroger les dictionnaires, nous apprenons que la trace peut être :

- *une empreinte* – le reste d'une action, et pour reprendre le vocabulaire de Peirce que nous avons introduit à l'occasion du chapitre sur les signes : un indice. Rappelons qu'il s'agit de la secondéité du signe considéré par rapport à son objet : ce qui n'existerait pas sans l'existence de son objet.
- *une petite quantité*
- *le lieu* de l'intersection d'un "objet" avec un plan de projection

Antiquité En partant du premier sens de la trace comme empreinte, nous pouvons faire un lien avec les théories de Platon. Pour lui, les rapports mémoire/imagination et vérité/erreur sont similaires et se ramènent au concept de la fidélité entre le souvenir et l'empreinte. Cette conception est très bien illustrée par la métaphore du bloc de cire que nous trouvons dans le théétète :

Socrate – [...] Est-ce qu'il est possible, quand d'abord on ne sait pas une chose, de l'apprendre ensuite ?

Théétète – C'est possible, bien sûr.

Socrate – Puis aussi, donc, une autre, et une autre ?

Théétète – Et pourquoi non ?

Socrate – Eh bien, accorde-moi de poser, pour les besoins de ce que j'ai à dire, qu'est contenu en nos âmes un bloc malléable de cire : plus grand pour l'un, plus petit pour l'autre ; d'une cire plus pure pour l'un, plus sale pour l'autre, et assez dure, mais plus humide pour quelques-uns, et il y en a pour qui elle se situe dans la moyenne.

Théétète – Je pose.

Socrate – Eh bien, affirmons que c'est là un don de la mère des Muses, Mémoire : exactement comme lorsqu'en guise de signature nous imprimons la marque de nos anneaux, quand nous plaçons ce bloc de cire sous les sensations et sous les pensées, nous imprimons sur lui ce que nous voulons nous rappeler, qu'il s'agisse de choses que nous avons vues, entendues ou que nous avons reçues dans l'esprit. Et ce qui a été imprimé, nous nous le rappelons et nous le savons, aussi longtemps que l'image en est là : tandis que ce qui est effacé ou ce qui s'est trouvé dans l'incapacité d'être imprimé, nous l'avons oublié, c'est-à-dire que nous ne le savons pas.

Théétète – Qu'il en soit ainsi.

[...]

Socrate – [...] il peut arriver que, te connaissant et connaissant Théodore, c'est-à-dire conservant dans ce fameux bloc de cire, exactement comme celles laissées par des anneaux, les marques que vous y avez laissées tous les deux, je vous voie tous les deux de loin et pas suffisamment : je m'efforce, en rapportant à la vision appropriée la marque propre à chacun de vous, la plaçant sur sa propre trace, de l'y faire coïncider, afin qu'il y ait reconnaissance ; et voilà que j'ai manqué les traces : comme ceux qui se chaussent à l'envers, je les ai interverties, et je vais faire se rencontrer la vision de chacun de vous avec la marque qui ne lui appartient pas ; ou encore il se passe quelque chose de semblable aux troubles que subit la vision dans les miroirs, quand elle détourne à gauche le flux qui s'écoule à droite : c'est quand je suis victime du même trouble que je mets à côté de la marque. Ce qui s'ensuit à ce moment-là, c'est bien de penser une chose pour l'autre, c'est-à-dire d'avoir des opinions fausses.

Ainsi, le rapport de la vérité et de l'erreur correspond à un ajustement entre l'empreinte et l'image de l'empreinte. Cette dernière, Platon la nomme *eikon*,

ce qui nous convient parfaitement car, nous avons vu que pour Peirce, l'icône c'est la priméité du signe considéré par rapport à son objet : c'est un signe qui renvoie à son objet par des qualités qui lui sont propres.

Modèle L'originalité des travaux dont nous rendons compte ici ([60], [89]) tient dans la définition d'un *modèle* de la trace informatique : M-trace. Cette modélisation semble avoir pour origine une relation analogique avec les classiques "systèmes à base de connaissances" (SBC). Il s'agit en effet de construire un "système à base de traces" (SBT) : un SBC dont la source de connaissances est un ensemble de traces d'interactions.

Trace primaire Nous avons déjà défini la trace d'interactions comme une séquence d'éléments observés. Il s'agit de la trace de plus bas niveau, la trace première. Remarquons (car ce n'est peut-être pas assez marqué, justement, dans les articles cités) qu'ici le concepteur du système tient un rôle essentiel. C'est lui qui choisit les informations portées par les observés élémentaires. Les auteurs expliquent que cette phase doit se faire avec la participation de l'utilisateur.

Système de transformation Cette trace primaire peut ensuite être transformée en traces de plus haut niveau qui sont des points de vue sur la trace primaire. C'est-à-dire que ces transformations font appel à de nouvelles connaissances. Pour les auteurs [60], ces connaissances sont formalisées en tant que *signatures de tâches*. Ces signatures sont des patterns qui permettent de détecter des sous-séquences de la trace de niveau n et de construire une nouvelle trace de niveau $n + 1$ pour laquelle les sous-séquences détectées deviennent des objets élémentaires qui sont des *épisodes* de la tâche dont la signature a servi à transformer la trace.

Système de requête Le modèle prévoit de pouvoir exécuter des requêtes sur la trace (quelque soit son niveau).

Implémentation Il n'y a pas pour l'instant d'implémentation totale de ce modèle. [89] propose un ensemble de règles de transformations qui permettent de traduire le modèle en Datalog. Ainsi, une implémentation exacte du modèle est en théorie possible.

Conclusion Cette approche est la plus avancée en terme de modélisation. Et il est d'ailleurs possible de considérer le modèle et l'implémentation que nous proposerons comme un sous-ensemble du modèle M-trace.

5.3 Présentation d'un autre point de vue sur les notions de contexte, de trace et d'apprentissage

5.3.1 Introduction

Notre problème est de savoir quels mécanismes introduire pour, tout en conservant notre plateforme de création de documents multistructurés ouverte, donner une certaine cohérence aux termes et aux relations qui seront créés librement par les utilisateurs. L'étude des travaux existant nous porte à placer la notion de contexte (son historique, sa représentation et sa manipulation) au cœur de notre solution. Cependant, nous avons besoin de fondements théoriques qui nous permettent d'articuler cette notion avec celles qui sont déjà essentielles à notre réflexion : les documents et l'apprentissage. Ces fondements, nous les avons trouvés auprès de deux auteurs : Jacques Derrida pour le rapport du document au contexte ; Gregory Bateson pour le rapport de l'apprentissage au contexte. Si nous parvenons à faire de ces réflexions un tout cohérent, nous serons prêts à modéliser la dynamique de notre système de construction de documents multistructurés.

5.3.2 Traces, Archive et Documents

Rappelons que nous avons défini l'Archive comme le lieu des traces documentaires. En fait, cette définition nous l'avons empruntée à un texte de Derrida [35] dans lequel il définit en particulier l'Archiviste comme celui responsable de la sélection des traces qui seront conservées – autrement dit, l'Archiviste est responsable de l'oubli ! Cette approche nous a permis de considérer le document comme une reconfiguration de l'Archive en vue de répondre à une question. Formellement, c'est ainsi que nous avons auparavant défini un document multistructuré comme un sous-graphe du rhizome (le graphe orienté formé de l'ensemble des relations entre fragments documentaires).

5.3.3 Document, Communication et Contexte

Si un document répond à une question, il est sans doute aussi destiné à communiquer cette réponse. Or, c'est à l'occasion d'un détour par cette notion de "communication" que Jacques Derrida [34] rejoint la notion de "contexte". En effet, à considérer le mot "communication" en tant que signe, nous pouvons nous demander s'il *communiqué* un sens univoque ? ou bien plusieurs ? Mais, quelque soit la réponse, la question suppose déjà une compréhension de la communication en tant que transmission d'un sens unique ! Or la communication peut aussi désigner des mouvements, des transmissions qui ne sont pas sémantiques (communication d'un mouvement, communication de deux lieux, etc.). Ainsi, ce dernier sens "physique" de la communication serait-il son "sens propre" dont serait dérivé métaphoriquement le sens sémio-linguistique ? Ça ne semble pas

convaincant car la notion de métaphore s'explique justement par les notions de mouvement, de transport, ...! Ainsi, le signe "communication" est en soi équivoque. Nous retrouvons maintenant les concepts appris de Peirce, car, pour Derrida, l'équivocité du mot "communication" demande afin d'être réduite la présence d'un *contexte*. Mais ce contexte, il s'agit évidemment de l'interprétant : un série d'autres signes dont les significations vont permettre de préciser la signification du premier signe. Là où Derrida nous intéresse, c'est lorsqu'il affirme que la détermination d'un contexte ne peut jamais être saturée.

5.3.4 Contexte et écriture

De cette impossible saturation de tout contexte, Derrida parvient à introduire une nouvelle conception de l'écriture. L'écriture dans le sens courant est un moyen de communication. C'est une forme d'extension des communications orales et gestuelles. C'est une forme de représentation mais dans un sens bien précis : elle suppose une présence physique qui s'atténue de manière homogène jusqu'à être seulement représentée (rendue présente). Dans cette représentation, et sauf accident "physique", le sens est conservé. Ainsi, l'écriture ne serait qu'une forme de réduction mécanique et homogène, un encodage des moyens de communication présentiels (en particulier la parole), et l'écriture alphabétique en serait l'aboutissement. C'est évidemment un point de vue extrême dont Derrida se sert pour introduire son concept de l'écriture. L'écriture ne serait pas fondée sur une extension homogène de la présence mais au contraire sur une absence absolue. En fait, une double absence : celle du destinataire et celle de l'émetteur. Le destinataire au moment où l'écriture se fait est absent. L'émetteur, au moment de la lecture, est absent : il est en particulier dans l'impossibilité de défendre ce qu'il a "voulu dire" (d'ailleurs, écrire ne serait-ce pas essentiellement se séparer d'un vouloir-dire?). Ces absences ne sont pas seulement physiques, mais doivent être considérées dans l'absolu. En particulier, de l'absence absolue du destinataire se déduit la propriété d'itérabilité de tout signe écrit. Cette propriété offre à tout signe écrit de pouvoir s'abstraire de son contexte original. Autrement dit, c'est la possibilité de la citation. Reprenons un exemple de Derrida que nous nous permettons d'étendre légèrement. Soit un signe écrit extrait du texte de Husserl intitulé les "Recherches logiques" : "le vert est ou". Ce signe peut être cité dans un nouveau signe : "'le vert est ou" est un exemple d'agrammaticalité". Ce nouveau signe peut être cité dans un nouveau signe : "'le vert est ou" est un exemple d'agrammaticalité" est un exemple de l'itérabilité de tout signe écrit.", etc. Cette propriété d'itérabilité rend tout signe écrit équivoque et interdit de le réduire à un élément de communication. Cette notion d'itérabilité nous convient parfaitement car elle défend et éclaire l'hypothèse qui a présidé à la création de notre plateforme pour la création de documents multistructurés : l'archive documentaire doit toujours être découparable en fragments qui peuvent entrer dans des relations quelconques quelle que furent leurs origines.

5.3.5 Apprentissage

Maintenant que semble mieux assurées l'importance et la pertinence de notre définition des documents multistrukturés, nous avons d'autant plus besoin d'être convaincus du bien fondé d'une utilisation de la trace des actions d'interactions comme support de l'apprentissage (et donc du contrôle) des termes et des relations qui vont avoir tendance à se multiplier dans un système aussi ouvert que le nôtre. Nous trouvons cette confirmation dans une formidable analyse, par Gregory Bateson, de la notion d'apprentissage [4].

5.3.6 Apprentissage et changement

Bateson part de l'observation que tout apprentissage dénote un changement. Il profite ensuite de l'analogie physique du mouvement qui est un changement de position, de la vitesse qui est un changement de mouvement, de l'accélération qui est un changement de vitesse, ... pour proposer une analyse de l'apprentissage en plusieurs niveaux. L'apprentissage de niveau 0 correspond à une quasi absence de changement dans la réponse à un stimulus répété. Cela correspond parfaitement aux formulations du langage courant du type : "L'odeur des cuisines du restaurant universitaire m'*apprend* qu'il est environ dix heures."

5.3.7 Apprentissage et contexte

L'apprentissage de niveau 1 correspond à un changement dans l'apprentissage de niveau 0. C'est celui qu'illustrent la plupart des expériences psychophysiques, comme celle fameuse du chien de Pavlov. Au départ, à l'audition du coup de sifflet, le chien ne salive pas. Plus tard, au *même* coup de sifflet, il salive. Nous avons souligné le mot "même" car il est en effet essentiel de supposer que les contextes peuvent se répéter. Si on ne considère pas des contextes itérables, il ne peut pas y avoir d'apprentissage de niveau 1 mais seulement des interprétations au niveau 0 : dans un premier contexte, le chien ne salive pas ; dans un second contexte, qui n'a rien à voir avec le premier, le chien salive. L'apprentissage de niveau 2 doit correspondre à un changement dans l'apprentissage de niveau 1 : c'est donc l'apprentissage de la reconnaissance des contextes. Des expériences très simples peuvent facilement prouver l'existence de l'apprentissage de niveau 2. Prenons l'exemple du "reverse learning". Dans un premier temps, un sujet apprend sur une série d'exemples à distinguer entre deux catégories d'objets. Soit R1 (respectivement R2) la réponse adaptée aux stimulus qui représentent des objets de la catégorie X (respectivement Y). Après une première phase d'apprentissage, X se met à signifier R2 et Y, R1. Une nouvelle phase d'apprentissage a lieu, puis les significations sont à nouveau inversées etc. Si, après plusieurs inversions, l'apprentissage devient plus rapide, c'est que le sujet aura appris quelque chose du processus même de l'inversion. Il s'agit bien d'un apprentissage dans la manière de détecter les contextes. La majorité de ce que nous appelons les traits de caractère (hostile, anxieux, narcissique, fataliste, dominant, soumis, dépendant, etc.) décrivent le résultat d'un apprentissage de

niveau 2. C'est-à-dire qu'ils décrivent une manière de découper le contexte. Essayons d'illustrer ce découpage en conservant notre définition naïve du contexte (ou de la trace, qui n'est qu'une représentation historique du contexte) en tant que séquence d'observés. Soit une situation d'interaction entre deux sujets *A* et *B*. Les observés sont les postures, les gestes, les paroles, etc. Soit une première représentation d'une sous-séquence de cet échange : a_i, b_i, a_{i+1} où a_i représente l'observé d'une action de *A* et b_i l'observé d'une action de *B*. Ici, *A* semble avoir émis un stimulus auquel *B* a répondu, puis *A* a renforcé (positivement ou négativement) la réponse de *B*. La même sous-séquence peut être interprétée autrement : b_{i-1}, a_i, b_i . Maintenant, c'est *A* qui semble répondre à un stimulus de *B*. Enfin, la séquence peut être interprétée ainsi : a_{i-1}, b_{i-1}, a_i . *A* semble réagir à une réponse de *B*.

5.3.8 Apprentissage de type 3

Que peut signifier un changement dans l'apprentissage de niveau 2? Mais, commençons par remarquer qu'il est très difficile de modifier les acquis de l'apprentissage de niveau 2. En effet, ce second niveau concerne l'apprentissage des manières d'articuler le contexte. Or, cette articulation du contexte a tendance à s'auto-valider car les signes émis par l'individu appartiennent évidemment au contexte mais dépendent en même temps de son articulation! Ainsi, un apprentissage de niveau 3 passerait peut-être par une mise à nu des mécanismes de l'apprentissage de niveau 2 (ici Bateson n'est pas très clair). Nous pouvons supposer qu'il impliquerait une perte du "caractère", autrement dit de la notion d'individu ...

5.3.9 Conclusion

Grâce à cette étude, nous comprenons que la présentation d'une représentation du contexte des actions d'interactions passées assistera les utilisateurs dans leur apprentissage du fonctionnement du système.

5.4 Modélisation et représentation des actions d'annotation

5.4.1 Enregistrement d'une représentation des actions d'annotation

Notre modèle de documents multistructurés étant formalisé en RDF, nous choisissons de définir un vocabulaire RDF très simple pour la représentation des actions d'annotation :

```
PREFIX users: <http://desanti.org/schemas/users#>
PREFIX traces: <http://desanti.org/schemas/traces#>
PREFIX trans: <http://desanti.org/schemas/transcription#>
INSERT INTO <http://desanti.org/> {
```

```

traces:Action      a          rdfs:Class      .

traces:hasDoer     a          rdf:Property .
traces:hasDoer     rdfs:domain traces:Action .
traces:hasDoer     rdfs:range  users:User  .

traces:hasTimestamp a          rdf:Property .
traces:hasTimestamp rdfs:domain traces:Action .

traces:withArgument a          rdf:Property .
traces:withArgument rdfs:domain traces:Action .
}

```

Pour chaque nouveau type d'objet ajouté au système, le programmeur doit fournir une sous-propriété de `traces:withArgument`. Par exemple, à l'introduction d'objets de type Intervalle, il faut ajouter au modèle des actions d'annotations les triplets suivants :

```

PREFIX traces: <http://desanti.org/schemas/traces#>
PREFIX trans:  <http://desanti.org/schemas/transcription#>
INSERT INTO <http://desanti.org/> {

  traces:withInterval rdfs:subPropertyOf traces:withArgument .
  traces:withInterval rdfs:range        trans:Interval      .
  traces:withInterval rdfs:label        "intervalle"        .
}

```

Ensuite, tracer une action d'annotation particulière devient très simple. Par exemple, pour tracer l'action d'isoler un nouvel intervalle de texte, il suffit du patron de requête SPARQL suivant (entre deux barres verticales sont indiquées les variables) :

```

PREFIX trans: <http://desanti.org/schemas/transcriptions#>
PREFIX user:  <http://desanti.org/users/>
PREFIX traces: <http://desanti.org/schemas/traces#>

INSERT INTO <http://desanti.org/>
{
  <|id|/intervals/|timestamp|> a trans:Interval .

  <|id|> trans:hasInterval <|id|/intervals/|timestamp|> .

  <|id|/intervals/|timestamp|> trans:hasStart "|start|" .
  <|id|/intervals/|timestamp|> trans:hasEnd  "|end|" .
  <|id|/intervals/|timestamp|> trans:hasTerm <|term|> .

  <|id|/actions/|timestamp|> a trans:AddInterval .
}

```

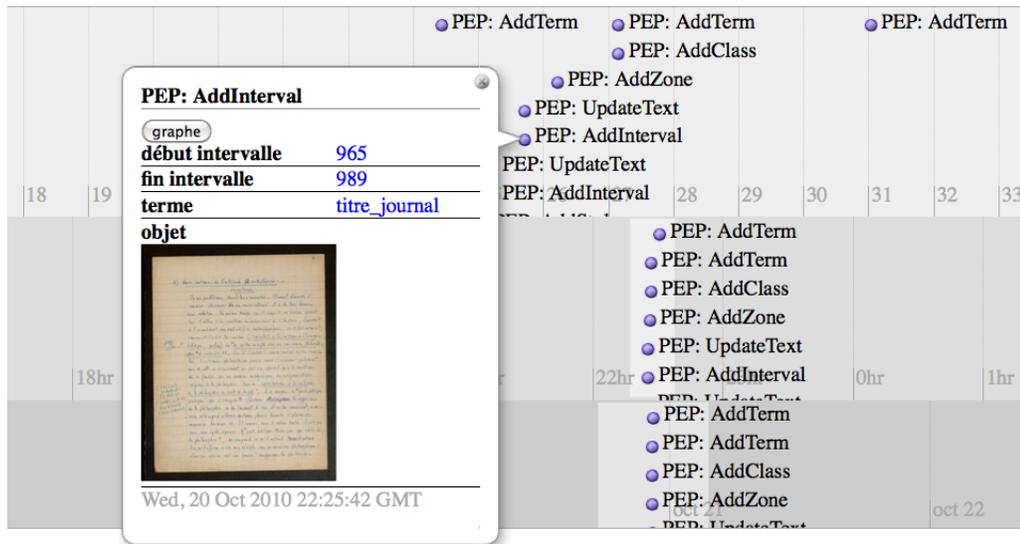


FIGURE 5.1 – Représentation chronologique de la trace des actions d’annotation

```

</id|/actions/|timestamp|> traces:hasDoer          user:|user|          .
</id|/actions/|timestamp|> traces:hasTimestamp    "|timestamp|"        .
</id|/actions/|timestamp|> traces:withIntervalStart  "|start|"            .
</id|/actions/|timestamp|> traces:withIntervalEnd    "|end|"              .
</id|/actions/|timestamp|> traces:withTerm          <|term|>              .
</id|/actions/|timestamp|> traces:withObject        <|id|>                .
}

```

Ainsi, nous avons opté pour une modélisation très légère des actions d’annotation. Cette modélisation répond exactement à nos besoins.

5.4.2 Visualisation des représentations des actions d’annotation

Nous proposons une vue chronologique des actions d’annotation. Sur la figure 5.1, une action de type **AddInterval** a été sélectionnée. Remarquons que cette action a lieu juste après l’ajout d’un nouveau terme et juste avant la délimitation d’une nouvelle zone de l’image d’une page manuscrite. De plus, toute action peut être représentée dans l’interface générique de visualisation et de création des relations (voir la figure 5.2).

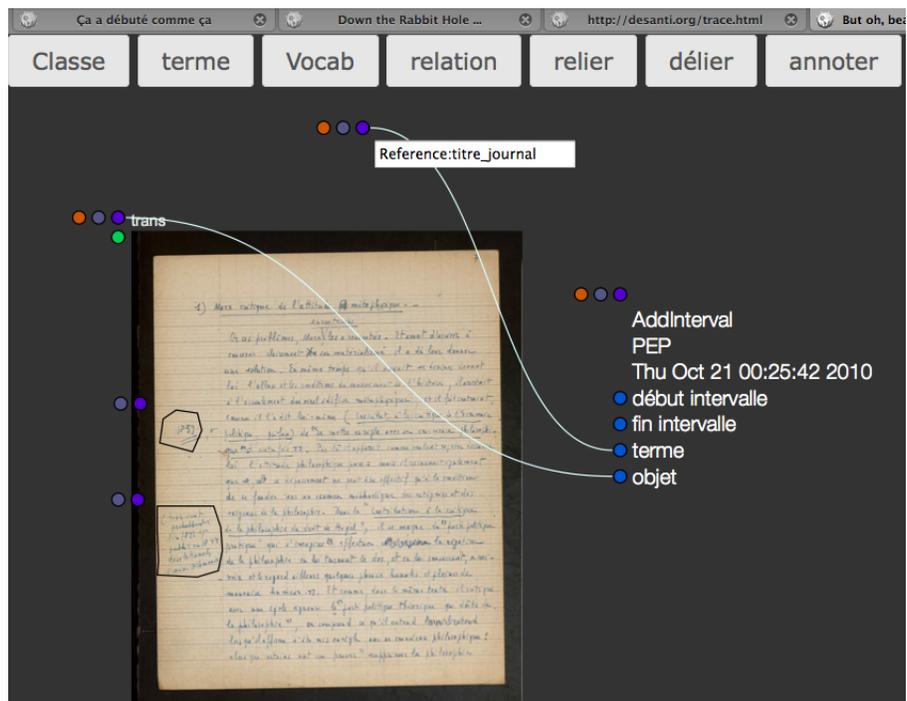


FIGURE 5.2 – Représentation d’une action d’annotation dans l’interface générique pour la manipulation des relations

5.5 Conclusion

Nous venons d'introduire une modélisation et des mécanismes de représentation et de manipulation de la trace des actions d'annotation. Nous avons montré comment une telle approche pouvait offrir la réflexivité nécessaire à une croissance contrôlée des vocabulaires de relations et des classes de termes.

Chapitre 6

DINAH, une plateforme philologique pour la construction de documents multistructurés

6.1 Introduction

Plan Nous allons commencer par décrire l'architecture logicielle de notre plateforme philologique pour la création de documents multi-structurés. Puis, nous décrirons le processus itératif de développement de DINAH. Enfin, nous proposerons un scénario d'utilisation qui servira à illustrer concrètement notre système. Ce scénario est extrait de la situation réelle d'interaction d'un chercheur de l'Institut Jean-Toussaint Desanti¹ avec DINAH.

Autres approches Bien que notre travail traite principalement d'un sujet inédit, la création des documents multistructurés, en tant qu'il trouve son application première dans le domaine des sciences humaines, il peut être comparé avec d'autres plateformes philologiques. Nous profitons de cette introduction pour engager cette comparaison. Nous répartissons ces approches en deux catégories. Premièrement, les systèmes qui possèdent un intérêt "historique". Deuxièmement, ceux, plus récents, qui utilisent les technologies du Web sémantique.

6.1.1 Les plateformes historiques

BAMBI [14] (Better Access to Manuscripts and Browsing of Images) est, selon ses auteurs, un système hypermedia permettant aux historiens de lire et

1. <http://institutdesanti.ens-lyon.fr/>

de transcrire des manuscrits, d'ajouter des annotations et de naviguer entre les mots de la transcription et les zones correspondantes de l'image du facsimile du manuscrit. C'est le premier essai d'une plateforme logicielle pour outiller l'activité philologique. Remarquons, que cette solution ne permet pas les annotations typées.

DEBORAH [69] (Digital Access to Books of the Renaissance) est un projet dont une partie consiste en une bibliothèque numérique accompagnée de mécanismes pour le travail collaboratif. Il a introduit la notion de "livres virtuels". Un livre virtuel est la représentation d'un chemin parmi toutes les pages de l'archive. Cependant, ces livres virtuels ne sont pas des ressources à part entière et, en particulier, ne peuvent pas être annotés. Cependant, nous pouvons considérer cette fonctionnalité comme l'amorce d'un système réflexif qui place l'utilisateur face à sa propre activité.

HyperNietzsche [36] (aujourd'hui renommé NietzscheSource) était un projet pionnier dans le domaine des bibliothèques numériques. Il propose un mécanisme de chemin très similaire aux livres virtuels du projet DEBORAH. Cependant, comme pour les livres virtuels, les chemins n'étant pas des ressources à part entière, ils ne peuvent pas vraiment participer à un processus collaboratif qui permettrait de les échanger et de les annoter.

6.1.2 Les plateformes basées sur les technologies du Web

Web sémantique Collate [96], TALIA [47], PINAKES [88], BRICKS [5] and JeromeDL [59] sont toutes des plateformes philologiques basées sur l'utilisation des technologies du Web sémantique. Elles offrent chacune des mécanismes de haut niveau pour l'annotation collaborative. Mais elles ne sont accompagnées d'aucun mécanisme de convergence pour permettre d'isoler et de documenter les vocabulaires de termes d'annotation.

Armarius [39] est utilisé pour classer et annoter des collections de manuscrits. Il permet seulement des annotations génériques non typées. Mais, il offre aussi une vue de l'ensemble des actions des utilisateurs ayant eu lieu durant la session courante. De plus, les auteurs prévoient d'utiliser des algorithmes de détection de motifs dans les graphes afin, par exemple, d'affecter des probabilités aux prochaines actions possibles. Ainsi, ce travail est comparable à notre utilisation des traces.

6.2 Architecture logicielle de DINAH

Introduction La figure 6.1 présente une vue logique des différents composants de la plateforme DINAH. Nous allons maintenant décrire chacun de ces composants.

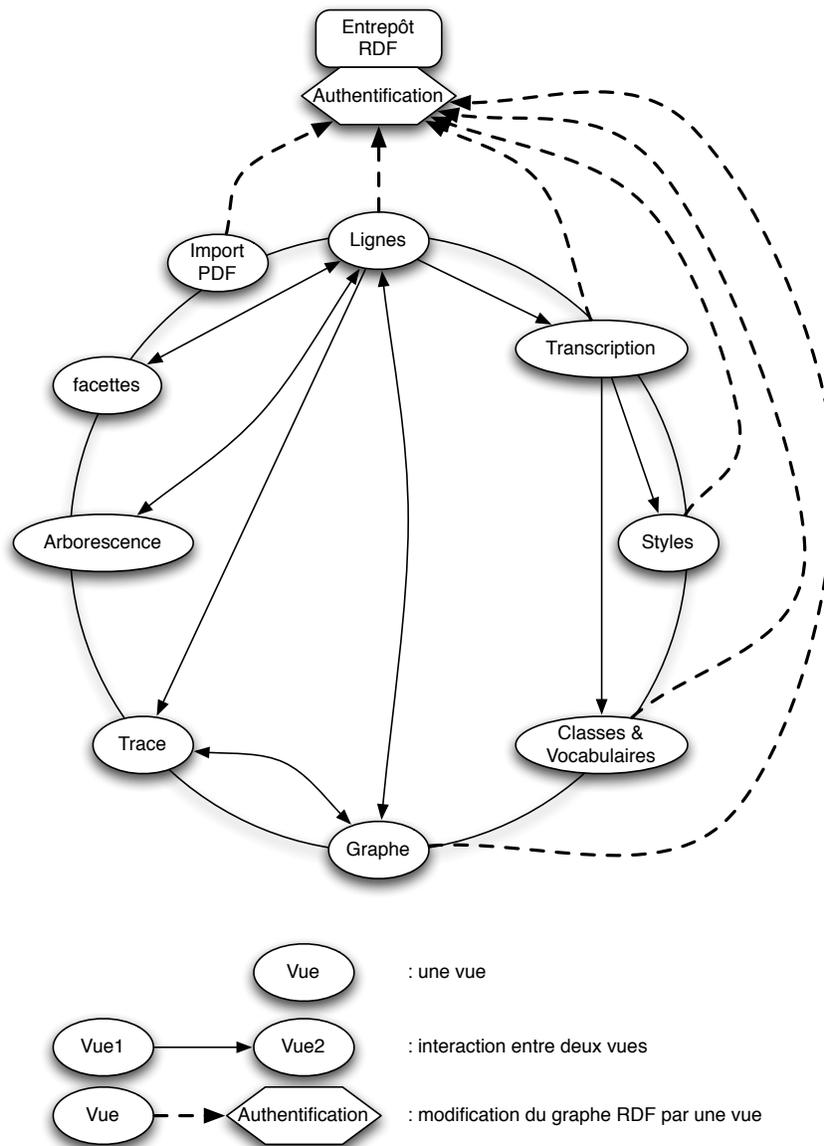


FIGURE 6.1 – Vue logique de la plateforme philologique DINAH

Entrepôt RDF Rappelons que nous avons formalisé notre modèle pour la création de documents multistructurés au moyen d'un graphe RDF. Ainsi, l'entrepôt RDF est un composant central de notre système. Il prend la place qu'occuperait une base de données classique. Il stocke l'ensemble des triplets RDF et propose une interface SPARQL afin de pouvoir émettre des requêtes sur le graphe. Nous avons choisi pour entrepôt RDF, ARC ².

Authentification Puisque DINAH propose un environnement multi-utilisateurs et que les clients sont des navigateurs Web, tous les accès à l'entrepôt RDF sont protégés par un module d'authentification écrit en PHP.

Vue en lignes La vue en ligne permet :

- de naviguer dans les collections de pages manuscrites
- de créer de nouvelles collections
- de modifier l'ordre des pages d'une collection
- d'associer des dossiers à une collection. Un dossier est une vue thématique de la collection. Par exemple, un dossier "datation" peut contenir toutes les pages d'une collection qui contiennent des indices de datation.

Vue arborescente La vue arborescente permet de naviguer dans les sous-collections d'une collection. Cette vue utilise la librairie javascript InfoViz ³.

La vue de la transcription La vue de la transcription permet :

- de transcrire le texte de l'image d'une page manuscrite
- d'isoler des intervalles du texte transcrit
- d'associer aux intervalles du texte transcrit des termes d'annotation
- d'isoler des zones de l'image du facsimile de la page manuscrite
- de relier des zones de l'image à des intervalles de la transcription

Cette vue utilise une version modifiée de l'éditeur wysiwyg Javascript tinyMCE ⁴. Elle utilise aussi la librairie Javascript jQuery ⁵.

La vue des styles La vue des styles permet de changer les styles associés aux classes de termes.

La vue des facettes La vue des facettes permet de rechercher des fragments documentaires par un ajout itératif de filtres. Cette vue utilise la librairie javascript Simile/Exhibit ⁶.

2. <http://arc.semsol.org/>

3. <http://thejit.org/>

4. <http://tinymce.moxiecode.com/>

5. <http://jquery.com/>

6. <http://www.simile-widgets.org/exhibit/>

Vue de la trace La vue de la trace permet de naviguer de manière chronologique dans l'historique des actions d'annotation. Cette vue utilise la librairie javascript Simile/Timeline⁷.

Vue du graphe La vue du graphe permet :

- de visualiser les relations entre les fragments documentaires (page, transcription, zone, intervalle, terme, relation, classe, vocabulaire, ...)
- de créer de nouvelles relations entre fragments documentaires

Cette vue utilise la librairie Raphaël⁸ qui permet de créer des dessins vectoriels au format SVG.

La vue des classes et des vocabulaires La vue des classes permet :

- de créer de nouvelles classes
- de créer de nouveaux termes
- de supprimer une classe
- de supprimer un terme
- de changer un terme de classe

Il en va de même pour la vue des vocabulaires.

L'import PDF Enfin, il est possible de créer une nouvelle collection à partir d'un document PDF.

6.3 Processus de développement de DINAH

6.3.1 Introduction

Le développement de DINAH s'est fait dans le contexte d'une étroite collaboration avec un groupe de cinq utilisateurs (des chercheurs et ingénieurs de l'Institut Jean-Toussaint Desanti). Dans un premier temps, nous proposons un bref retour sur différents prototypes qui ont précédé la version actuelle de DINAH. Puis, nous décrivons le processus itératif de développement qui a permis de proposer une réponse, validée par des expérimentations, au problème de l'analyse critique de sources textuelles et aux besoins de gestion des connaissances qui en découlent.

6.3.2 Point de vue historique

Ruby on Rails Un premier prototype utilisait la plateforme de développement d'applications Web nommée Rails⁹ et le langage de programmation Ruby¹⁰. D'un point de vue "ingénierie logicielle", ce choix d'outils a permis le développement très rapide d'une solution utilisable. Cependant, lorsqu'il s'agit

7. <http://www.simile-widgets.org/timeline/>

8. <http://raphaeljs.com/>

9. <http://rubyonrails.org/>

10. <http://www.ruby-lang.org/>

de ne plus respecter les guides de modélisation que propose (ou impose ...) la plateforme Rails, ce choix d'architecture n'est plus satisfaisant. Or, dans le contexte d'une situation de recherche, cette contrainte est rapidement apparue comme un frein trop important pour que le développement se poursuive selon ce premier choix d'architecture.

Haskell et Tcl/Tk Un second prototype se base sur un schéma d'architecture client/serveur. Le serveur est un programme écrit dans le langage de programmation fonctionnelle Haskell¹¹. Le choix de ce langage était motivé par l'envie d'expérimenter des mécanismes de validation pour les documents multi-structurés. En effet, des mécanismes de typage devaient nous permettre d'assurer la préservation, par une suite d'opérations d'édition, de la validité d'un document relativement à un ensemble de schémas de structures documentaires. Cependant, nous nous sommes rendus compte a posteriori que de tels travaux avaient déjà été menés [97]. Finalement, nous avons choisi le langage Tcl et le toolkit graphique Tk¹² pour le prototypage rapide d'éléments d'IHM. Nous nous sommes rapidement trouvés limités par le faible nombre de widgets proposés par Tk, ainsi que par l'absence d'une gestion satisfaisante du "glisser/déposer".

Haskell et Squeak Ainsi, nous sommes passés côté client de Tk à Squeak¹³ qui est une mise en œuvre moderne de l'environnement Smalltalk¹⁴. En effet, Squeak implante Morphic [66], un kit de construction d'interfaces graphiques qui est une alternative au traditionnel modèle MVC (Modèle Vue Contrôleur)¹⁵. Le grand avantage de ce système est de pouvoir construire et éditer des objets graphiques à la fois par manipulation directe et à partir de programmes. Ainsi, il est étonnamment facile de construire un nouvel objet graphique (ou "Morph") par composition et extension d'autres objets graphiques. Cette souplesse nous a permis d'organiser des sessions de travail en direct avec un groupe de trois (ou parfois deux) utilisateurs où nous avons pu tester différents modes d'interaction dans un temps très court.

Haskell et les technologies du Web Lorsque le nombre des utilisateurs intéressés par tester notre approche est devenu important, nous avons eu besoin de passer à une solution plus simple à déployer. Ainsi, nous avons écrit une nouvelle version de l'IHM qui s'exécutait dans un navigateur Web et utilisait des technologies du Web (XHTML, CSS, Javascript).

Entrepôt RDF et les technologies du Web A force de retravailler le noyau Haskell de notre système, nous l'avons peu à peu réduit à un programme relativement générique de gestion de graphe. Ainsi, pour rendre notre système

11. <http://www.haskell.org/>

12. <http://www.tcl.tk/>

13. <http://www.squeak.org/>

14. <http://www.smalltalk.org/versions>

15. <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>

plus accessible à d'autres programmeurs et profiter d'un standard bien défini, nous avons remplacé le programme côté serveur par l'entrepôt RDF Arc¹⁶. En effet, un entrepôt RDF est un système générique pour la gestion de données sous la forme de graphes. De plus, nous avons ainsi pu bénéficier d'un point d'accès SPARQL¹⁷ qui a rendu, à peu de frais, notre système interopérable car s'intégrant au "Web sémantique" ou "Web des données"¹⁸.

6.3.3 Un processus itératif

Dans le cadre de notre travail avec les chercheurs de l'Institut Jean-Toussaint Desanti, nous avons adopté un cycle de développement fait d'itérations courtes et inspiré des différentes méthodes Agiles¹⁹. Nous nous rencontrons une à deux fois par semaine. Une nouvelle version de DINAH était livrée toutes les deux à trois semaines (sauf dans le cas de certains changements lourds d'architecture). Ainsi, nous sommes assurés d'avoir bien répondu aux besoins des chercheurs de l'Institut Jean-Toussaint Desanti (ces derniers utilisent d'ailleurs DINAH quotidiennement ...). De plus, nous avons présenté DINAH, soit dans le cadre de colloques, soit lors de rencontres plus informelles, à des chercheurs qui travaillent sur d'autres projets d'analyse critique de sources textuelles. Leurs points de vue nous ont souvent permis d'améliorer notre système.

6.4 Scénario d'utilisation

Introduction Afin d'offrir un peu plus de matérialité à la description de notre plateforme DINAH, nous décrivons maintenant un petit scénario d'utilisation. Nous nous plaçons "dans la peau" d'un utilisateur.

Consultation Je commence par ouvrir la consultation des couvertures – il s'agit d'une version spécialisée de la navigation par facettes qui affiche uniquement les couvertures des collections afin d'offrir une vue synoptique de l'archive – (voir la figure 6.2).

Restriction Comme je suis à la recherche d'un cahier, je restreins la vue des couvertures aux seuls cahiers en cliquant sur le type "Cahier" (voir la figure 6.3).

Ouverture d'un cahier Je sélectionne le cahier 109 et il s'ouvre dans la vue des lignes (voir la figure 6.4).

Transcription J'ouvre la page 7 du cahier 109 dans la vue transcription. Je transcris la page, je crée quelques intervalles de texte et leur associe des concepts, des dates, etc. (voir la figure 6.5).

16. <http://arc.semsol.org/>

17. <http://www.w3.org/TR/rdf-sparql-query/>

18. <http://www.w3.org/2001/sw/>

19. <http://www.agilejournal.com/>

1010 Items

Trier par : [libellés](#), [puis par...](#) • Grouper selon le tri

1.	 0109
1.	 0109/6_1
3.	 0109/6_1/2_1
	 0109/6_2

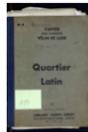
Type

91	Cahier
724	Insert
195	Pochette

FIGURE 6.2 – Vue synoptique de l'archive

91 Cahier filtrés sur un total de 1010 items ([Réinitialiser tous les filtres](#))

Trier par : [libellés](#), [puis par...](#) • Grouper selon le tri

1.	 0109
2.	 0110
3.	 0111
	 0112

Type

91	Cahier	<input checked="" type="checkbox"/>
724	Insert	<input type="checkbox"/>
195	Pochette	<input type="checkbox"/>

FIGURE 6.3 – Vue synoptique des cahiers de l'archive

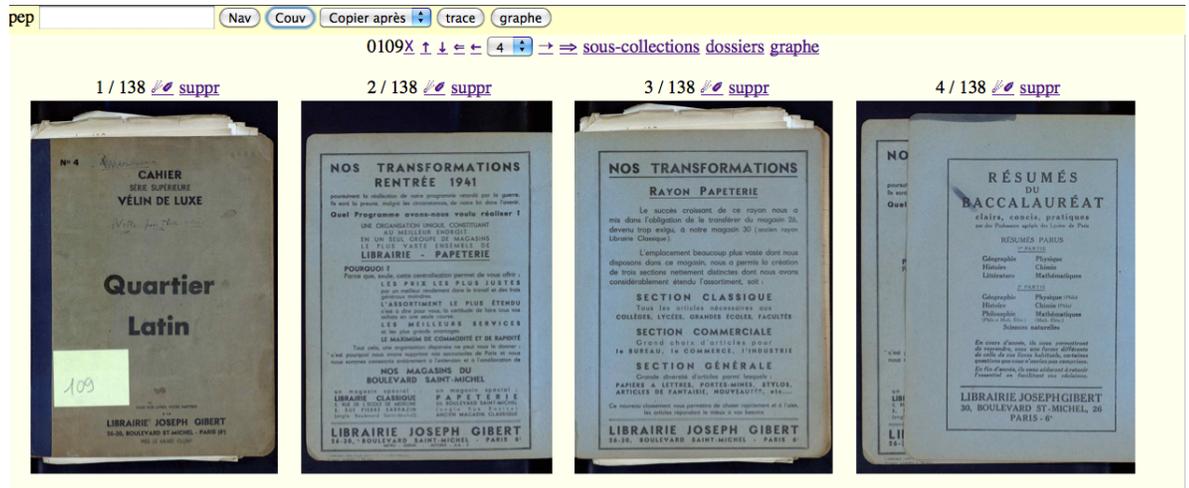


FIGURE 6.4 – Ouverture d'un cahier dans la vue par lignes

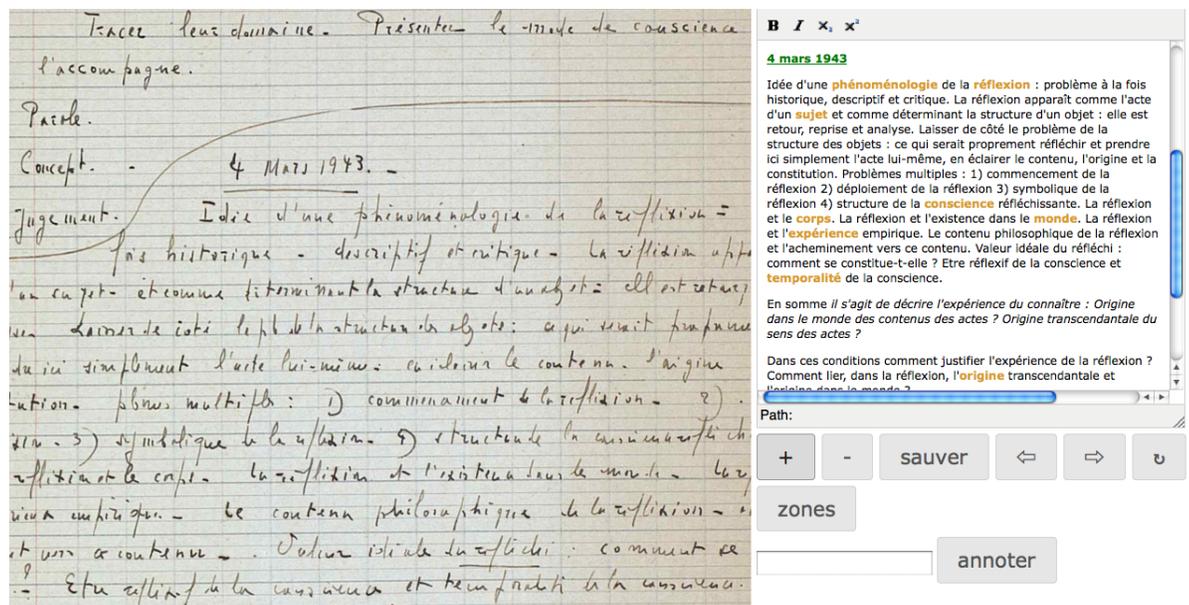


FIGURE 6.5 – Interface pour la transcription

Éditer le style de la classe => Auteur

Couleur ■ + ■ + ■ = ■

Italique

Gras

Décoration souligné barré normal

Exemple

sauver le nouveau style

FIGURE 6.6 – Modification des styles associés aux classes

Styles Je m’amuse à changer le style associé à la classe des noms auteurs (voir la figure 6.6).

Classes Je vais regarder la composition de la classe des mots-clefs, et je peux l’éditer (voir la figure 6.7).

Dossiers Je reviens à l’interface par lignes et j’associe au cahier 109 un dossier "datation" qui regroupe tous les éléments qui permettent de dater le cahier (voir la figure 6.8).

Consultation Maintenant, j’aimerais consulter tout ce qui dans l’archive a été annoté comme concernant Husserl et la notion d’ego. J’utilise la navigation par facettes et saisi dans le champ de texte libre : "Husserl" (voir figure 6.9). Puis, je sélectionne uniquement ce qui concerne aussi le mot clef "ego" (voir figure 6.10).

Arborescence Pour me rendre compte de la structure de la pochette 200, je déploie une partie de l’arborescence de ses inserts (voir figure 6.11).

Zones J’ouvre la page 33 du 4ème insert du 1er insert de la pochette 200 et j’isole une zone de l’image de la page manuscrite (voir figure 6.12).

Relations J’ouvre cette même page dans la vue en graphe (voir figure 6.13). J’apprends que cette page est un version alternative d’une collection qui se trouve tout à fait ailleurs dans l’archive! (Remarquons en particulier que les zones, les relations, les pages, les collections etc. sont tous des objets à part entière et peuvent entrer individuellement dans des relations quelconques).

Trace Finalement, un collègue me demande d’aller jeter un coup d’œil à la page 82 du cahier 159 sur laquelle il a travaillé. Je l’ouvre dans la vue de type graphe (voir figure 6.14). Je remarque qu’il affirme que cette page renvoie à

Auteur	mécanisme
Concept	phénomène
Mot clef	structure
Destinateur	langage
Date	production
Liste de sujets de leçon	histoire
Étude préparatoire	développement
Notes relatives à	impossible
Husserl	valeur
Cours	morale
Année	formalisme
Visibility	finalité
Essai	nature humaine
Exemple	élément
Tapuscrit	concept
Oeuvre	conscience
Zones	système
Reference	exprimable
A	mathématiques
B	sujet
Physique	épistémologie

Mot clef

sauver le nouveau nom



nouveau terme :

FIGURE 6.7 – Éditeur de classes

pep Nav Couv Copier après trace graphe



0109->datation X ↑ ↓ ⇄ (4) ⇒ SUPPR sous-collections dossiers graphe

1 / 13 [suppr](#) 2 / 13 [suppr](#) 3 / 13 [suppr](#) 4 / 13 [suppr](#)



[datation](#) [créer](#)

FIGURE 6.8 – Création d'un dossier "datation" associé au cahier 109

Les collections J-T. Desanti

6 Items filtrés sur un total de 87 items ([Réinitialiser tous les filtres](#))

Trier par : [libellés, puis par...](#) • Grouper selon le tri

1.  0123	<input type="text" value="husserl"/> Type <ul style="list-style-type: none">3 Cahier1 Insert2 Page
2.  0124	Auteur <ul style="list-style-type: none">4 (missing this field)1 heidegger1 husserl1 sartre
3.  0125	Concept <ul style="list-style-type: none">6 (missing this field)

FIGURE 6.9 – Navigation par facettes pour trouver ce qui concerne Husserl

1.  <http://desanti.org/collections/0109/0014>

Auteur

- 1 heidegger
- 1 husserl

Concept

- 1 (missing this field)

Mot clef 1

- 1 ego

FIGURE 6.10 – Navigation par facettes pour trouver ce qui concerne Husserl et le mot-clef "ego"

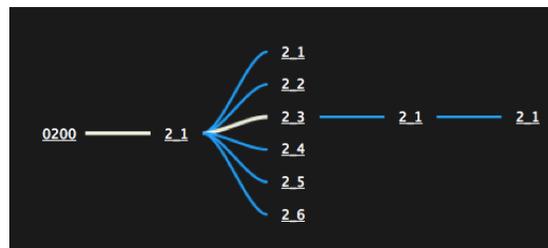


FIGURE 6.11 – Vue synoptique des cahiers de l'archive

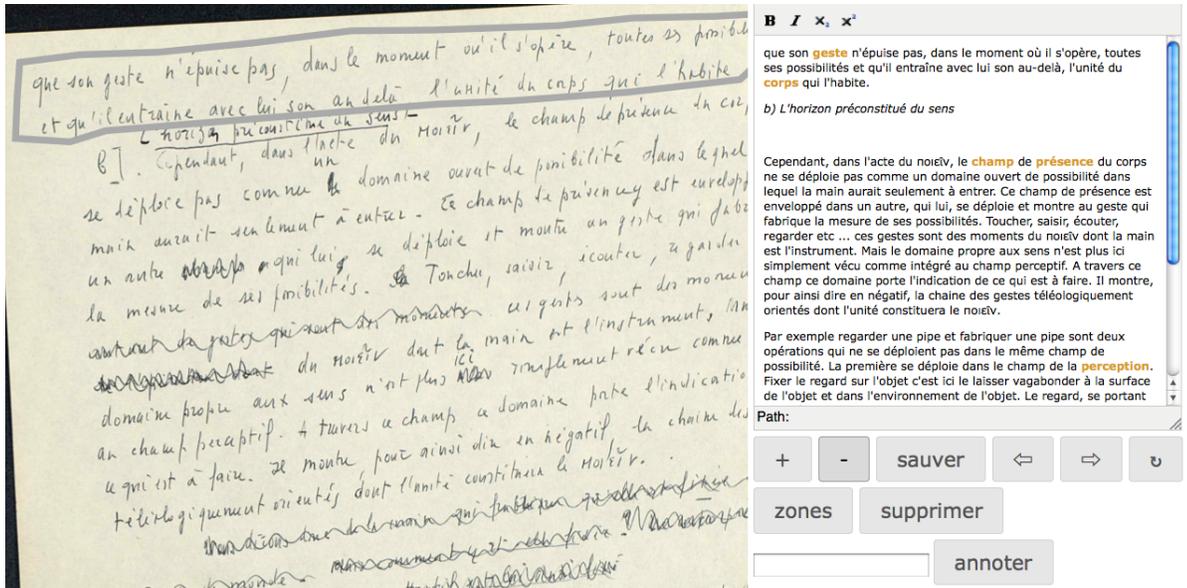


FIGURE 6.12 – Création d'une zone d'image de facsimile de page manuscrite

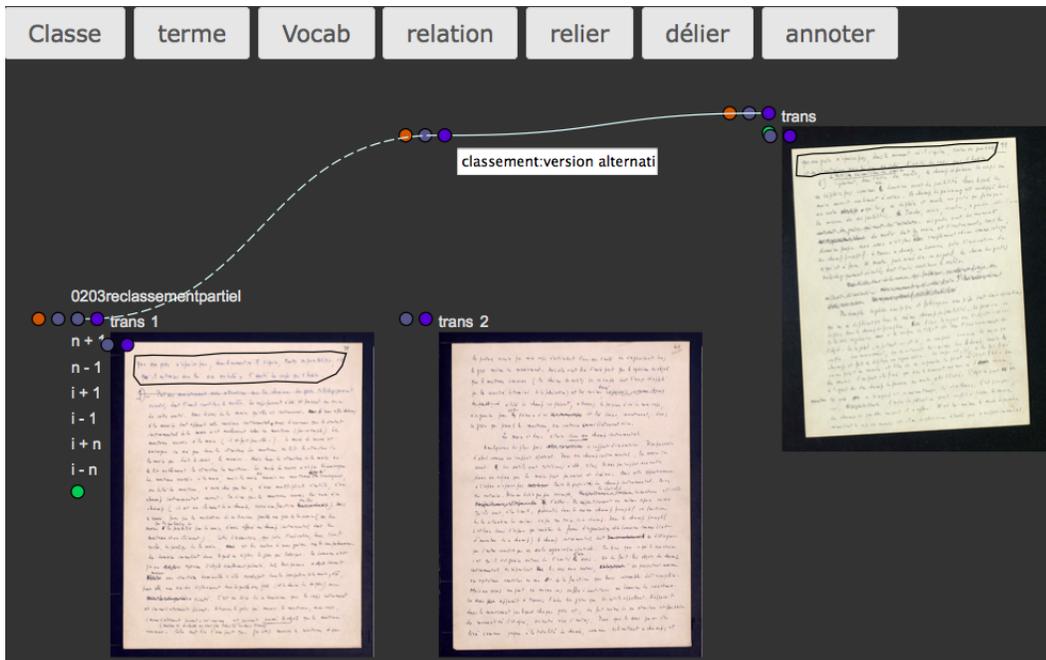


FIGURE 6.13 – Consultation et création de relations

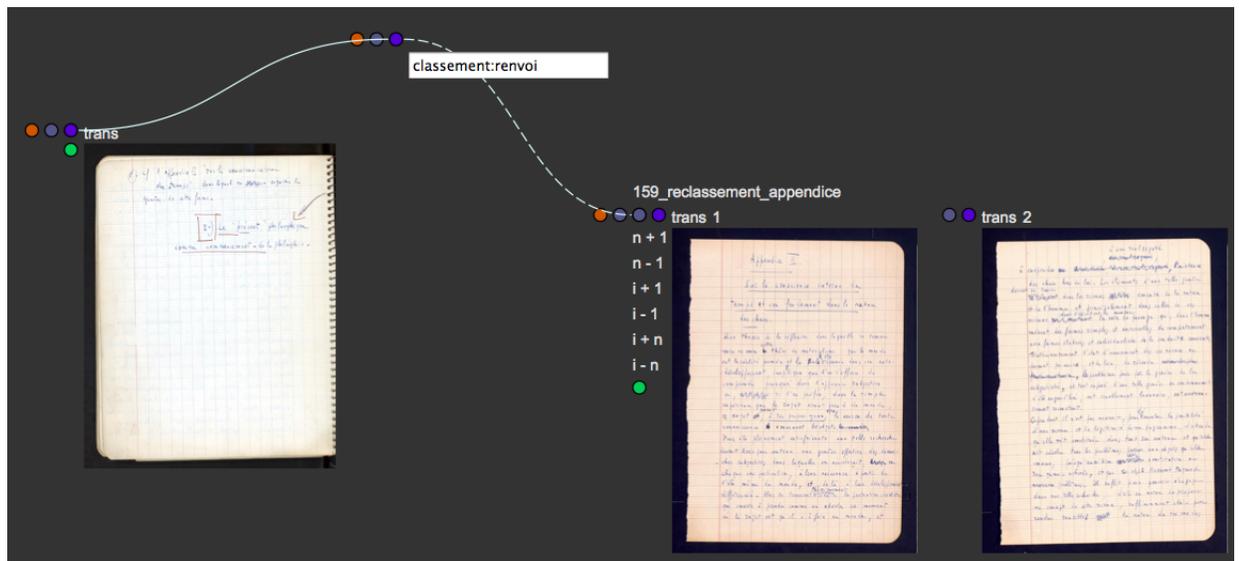


FIGURE 6.14 – Les relations de la page 159

une autre collection, mais je ne suis pas certain de savoir ce qu'il entend par renvoyer. J'accède donc à la trace de toutes les instances d'utilisation de la relation "classement :renvoi" (voir figure 6.15). Je pourrais explorer chacune de ces instances et me faire une idée de la bonne manière d'utiliser cette relation. Mais, dans un premier temps, je vais simplement ouvrir dans le graphe une représentation de l'action d'annotation de mon collègue (quand il affirme que cette page renvoie etc.). Je remarque que cette action (en tant qu'action) a déjà été annotée. Cette annotation sert de documentation et m'est d'une aide précieuse (voir figure 6.16). Par ailleurs, si je ne suis décidément pas d'accord avec mon collègue, je peux annoter son action pour exprimer mon désaccord et entamer une discussion ...

6.5 Conclusion

Nous avons montré comment notre système de construction des documents multistructurés (DINAH) implémente l'ensemble des éléments qui avaient été présentés dans les chapitres précédents : de la méthodologie de construction des documents multistructurés à l'utilisation de la trace des actions d'annotation comme moyen de contrôle et de documentation des vocabulaires de relations et des classes de termes.

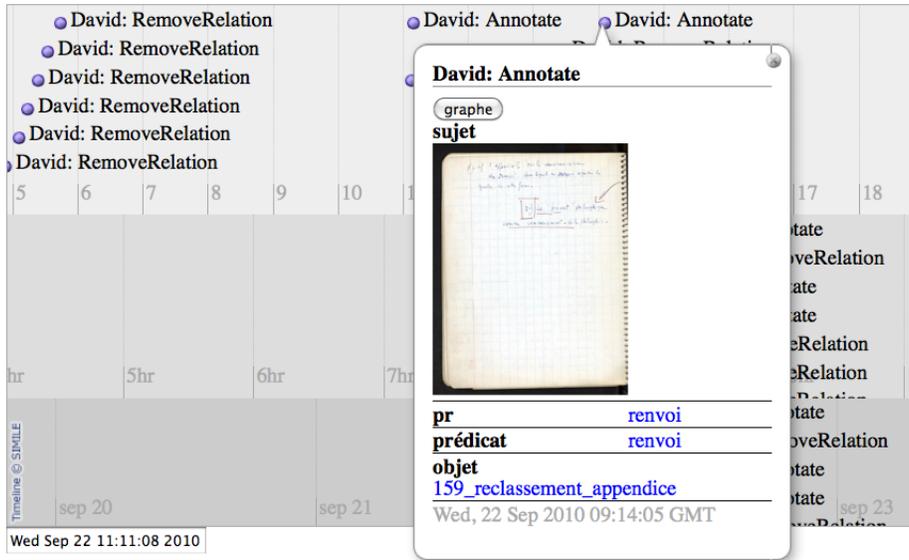


FIGURE 6.15 – Vue de la trace centrée sur les action qui mettent en jeu la relation "classement :renvoi"

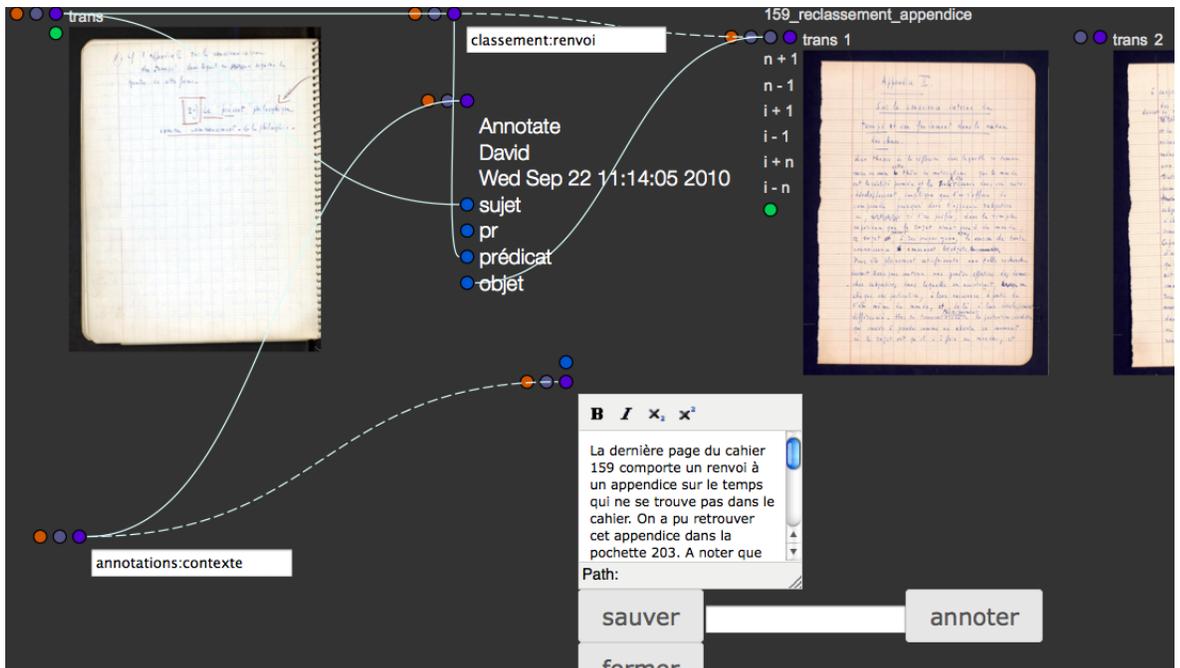


FIGURE 6.16 – Représentation d'une action sur le graphe des relations

Chapitre 7

Conclusion et perspectives

7.1 Bilan

Nous avons mené un travail dans la continuité des recherches existantes sur la notion de document multistructuré. En effet, si les problèmes de représentation et de manipulation de ces documents sont très étudiés, celui de leur construction est resté jusqu'à présent non traité. Or, nous n'avons pas suivi la voie conventionnelle qui tend à localiser le cœur du problème dans la difficulté technique de la représentation des hiérarchies concurrentes. Au contraire, nous avons choisi de considérer les chevauchements de termes d'annotation comme des signes déclencheurs de la création de nouvelles structures. Pour mener à bien ce retournement, nous avons eu à faire ce qui a pu ressembler extérieurement à un détour mais dont nous ne doutons pas de la rigoureuse nécessité : une étude du concept de signe. Ainsi, ce qui s'est d'abord présenté comme un point de vue original sur la dynamique des documents multistructurés a fini par devenir une construction théorique propre à offrir une lecture en partie neuve des phénomènes d'interactions Homme-Machine. C'est sur cette nouvelle base que nous avons poursuivi notre démarche par la proposition d'une incitation aux activités réflexives afin d'assurer, autant que faire ce peut, la cohérence de l'évolution des vocabulaires de relations et des classes de termes. Ainsi, grâce à l'introduction d'une nouvelle médiation manipulable : la trace des actions d'annotation, nous avons pu atteindre une forme souple de contrôle des vocabulaires tout en conservant un système dynamique, générique et très ouvert. Par ailleurs, nous n'aurions sans doute pas su adopter ces points de vue excentrés qui nous furent si utiles si l'application première de notre système de création de documents multistructurés n'avait été un projet passionnant – car mené par des chercheurs et ingénieurs passionnés – d'étude et d'édition des manuscrits d'un célèbre philosophe français : Jean-Toussaint Desanti.

7.2 perspectives

Nous avons pour projet de poursuivre ce travail selon au moins trois axes : l'utilisation d'une nouvelle contrainte (la condition d'acyclisme des graphes) pour assister la construction des vocabulaires, une gestion fine et dynamique de la localisation des données et la création d'une librairie qui permette l'addition générique des mécanismes de traçage des actions d'annotation.

7.2.1 Utiliser la contrainte d'acyclisme pour assister la construction des vocabulaires

Alors que nous utilisons la contrainte d'arborescence pour assister automatiquement la construction des classes de termes, nous aimerions trouver un autre type de contrainte applicable aux vocabulaires de relations. Évidemment, il ne peut pas s'agir de l'arborescence puisque, d'après les définitions de notre modèle, les relations forment un graphe orienté. Nous faisons l'hypothèse – qu'il faudra vérifier – que si les instances des relations d'un même vocabulaire ne forment entre elles que des graphes acycliques alors, le vocabulaire est bien formé, dans le cas contraire, il est nécessaire de le scinder en sous vocabulaires bien formés. Pourquoi cette intuition ? Pour la simple raison que le respect de la contrainte d'acyclisme produit des graphes qui possèdent globalement une structuration hiérarchique en plusieurs niveaux [56].

7.2.2 Gestion fine de la localisation des données

Pour le moment, nous reposons sur une architecture client-serveur, toutes les données étant stockées sur le serveur. Si cette solution est très pratique technologiquement, elle ne nous satisfait pas. Nous aimerions réfléchir aux nouvelles contraintes qu'introduirait une architecture pair-à-pair qui aurait de nombreux bénéfices immédiats : une plus grande flexibilité pour les utilisateurs, la possibilité de travailler hors-ligne et le respect du secret de l'utilisateur qui reste en particulier maître de sa trace.

7.2.3 Création générique des mécanismes de traçage des actions d'annotation

Pour le moment, lorsqu'une nouvelle forme d'interaction est ajoutée au système il faut penser à assurer son traçage. Nous avons montré que cette opération est très simple. Cependant, il faudrait l'automatiser afin que le développeur n'ait pas même à y penser.

Appendices

Annexe A

Deux fonctions Javascript

A.1 makeIntervals

```
makeIntervals = function(editor){
//given an Editor, computes the intervals of annotated text
//
//an interval is of the form:
//{start: 10, end: 18, class: "class1276699414",
// id: "DBIdOfTheInterval OR DBIdOfTheTerm"}
//
//for a new unsaved interval i, i.id is the id of the term
//used to annotate the interval
//
//for a saved interval i, i.id is the id of the interval
//
//for an interval i, i.class is of the form "classxxxxxxxxxx"
//where "http://desanti.org/classes/xxxxxxxxxx" is the
//class of the term used to annotate i
    var body=editor.getBody(),
        nbChars=0,
        intervals=[],
        text="",
        processChilds;

    processChilds=function(elem, supNbChars){
        //supNbChars is an offset, it is the number of chars that have
        //already been treated before treating elem
        var nbChars = 0,
            //nbChars is the number of chars that have already been treated
            //while processing elem
            subNbChars = 0;
        //subNbChars is the number of chars of a sub-element of elem
        //before being added to nbChars
    }
```

```

$.each(elem.childNodes, function(i,child){
    var start, end, wasPartOfOverlap=false;
    //start is the ordinal number of the character
    //that starts the interval
    //end is the ordinal etc.
    //wasPartOfOverlap is a boolean value that is true if
    //child is part of a fragmented interval
    //
    //In fact, the browser makes this transformation
    //(the sign "..." is a visual breakline and doesnt exist):
    //
    //<p><span id="id1" class="A">aa</span id="id2" class="B">...
    //abb</span></span></p>
    //
    //becomes:
    //
    //<p><span id="id1" class="A">aa<span id="id2" class="B">...
    //a</span></span><span id="id2" class="B">bbb</span></p>
    //
    if(isATextNode(child))
    {
        nbChars += child.textContent.length;
        text += child.textContent;
    }
    else if(isAInlineFormatingNode(child))
    {
        text += "<" + child.tagName + ">";
        nbChars += processChilds(child,
supNbChars +
nbChars +
"<>".length +
child.tagName.length);
        nbChars += "<>".length + "</>".length + 2*child.tagName.length;
        text += "</" + child.tagName + ">";
    }
}

else if(isASpanNode(child))
{
    subNbChars = 0;
    if(child.innerHTML != ""){
        //an annotation that has just been deleted can introduce
        //an empty span. Thus this previous test.
        subNbChars = processChilds(child, supNbChars + nbChars);
        start = supNbChars + nbChars + 1;
        end = supNbChars + nbChars + subNbChars;
        wasPartOfOverlap = false;
        $.each(intervals, function(){
            // see the previous explanation on the automatic
            // browser transformation of overlaps
            if(this.id === child.id && (this.end + 1) === start){
                wasPartOfOverlap = true;
            }
        });
    }
}

```

```

        this.end = end;
    }
    });
    if(!wasPartOfOverlap){
        intervals.push({
            start: start,
            end: end,
            class: child.className,
            id: child.id
        });
    }
}
    nbChars += subNbChars;
}
});
return nbChars;
}
$.each(body.childNodes, function(i,child){
    if(isAPNode(child)){
        text += "<P>";
        nbChars += processChilds(child, nbChars + "<P>".length);
        nbChars += "<P></P>".length;
        text += "</P>";
    }
});
return {
    intervals: intervals,
    text: text
};
}

```

A.2 makeHtml

```

makeHtml = function(intervals, text){
//intervals: [interval]
//interval: {id: "DBIdOfTheInterval", start: "23", end: "76",
//classNum: "classxxxxxxxxxx"}
//where "http://desanti.org/classes/xxxxxxxxxx" is the
//class of the term used to annotate the interval
//
//returns an HTML string where the intervals are modelized by:
//<span id="DBIdOfTheInterval" class="classxxxxxxxxxx">text annotated</span>
//
    var offsetStart=0,
        offsetEnd=0,
        tagOpenLength = '<span class="" id="">'.length,
        tagCloseLength = '</span>'.length,
        resolveOverlap,
        previousInterval,

```

```

    openAndCloseOffset,
    openOffset,
    overlap = false,
    intervalsToSkip = [];

resolveOverlap = function(i1, i2){
//i1: already treated interval ; i2: to be treated interval
//if i1 & i2 overlap returns true and adds i2 to intervalsToSkip and
//makes two new intervals i2a & i2b around the overlapping point
//if i1 & i2 do not overlap returns false and do nothing
//
    var i2a = {id: i2.id, classNum: i2.classNum,
                start: i2.start, end: i2.end},
        i2b = {id: i2.id, classNum: i2.classNum,
                start: i2.start, end: i2.end};
    if(i2.start > i1.start && i2.start <= i1.end && i2.end > i1.end){
        i2a.end = i1.end;
        i2b.start = i1.end + 1;
        intervals.push(i2a);
        intervals.push(i2b);
        intervalsToSkip.push(i2);
        return true;
    }else if(i2.end >= i1.start && i2.end < i1.end && i2.start < i1.start){
        i2a.end = i1.start - 1;
        i2b.start = i1.start;
        intervals.push(i2a);
        intervals.push(i2b);
        intervalsToSkip.push(i2);
        return true;
    }
    return false;
}

for(var i = 0; i < intervals.length; i+=1){
    offsetStart=0;
    offsetEnd=0;
    overlap = false;

    for(var j=0; j<i; j+=1){
        if($.inArray(intervals[j], intervalsToSkip) !== -1){ continue; }
        overlap = resolveOverlap(intervals[j], intervals[i]);
        if(overlap){ break; }
        previousInterval = intervals[j];
        openOffset = previousInterval.classNum.length +
            previousInterval.id.length + tagOpenLength;
        openAndCloseOffset = openOffset + tagCloseLength;
        if(intervals[i].start > previousInterval.end){
            offsetStart += openAndCloseOffset;
        }
        else if(intervals[i].start >= previousInterval.start){

```

```

        offsetStart += openOffset;
    }
    if(intervals[i].end > previousInterval.end){
        offsetEnd += openAndCloseOffset;
    }
    else if(intervals[i].end > previousInterval.start){
        offsetEnd += openOffset;
    }
}
if(!overlap){
    text = text.slice(0,offsetStart+intervals[i].start-1)+
        '<span class="'+intervals[i].classNum+'" id="'+intervals[i].id+'">'+
        text.slice(offsetStart+intervals[i].start-1, offsetEnd+intervals[i].end)+
        '</span>' +
        text.slice(offsetEnd + intervals[i].end);
}
};
return text;
}

isATextNode = function(node){
    return node.nodeType === 3;
}
isAnElement = function(tagName){
    return function(node){
        return ((node.nodeType === 1) && node.tagName === tagName);
    };
}
isASpanNode = isAnElement("SPAN");
isAPNode = isAnElement("P");
isAStrongNode = isAnElement("STRONG");
isAEmNode = isAnElement("EM");
isASubNode = isAnElement("SUB");
isASupNode = isAnElement("SUP");
isAInlineFormattingNode = function(node){
    return (isAStrongNode(node) || isAEmNode(node) ||
        isASubNode(node) || isASupNode(node));
}
}

```

Bibliographie

- [1] R. Abascal, M. Beigbeder, A. Benel, S. Calabretto, B. Chabbat, P.A. Champin, N. Chatti, D. Jouve, Y.Prié, B. Rumpler, and E. Thivant. Modéliser la structuration multiple des documents. In *H2PTM Hypertexte et Hypermédia Créer du sens à l'ère du numérique*. Hermès, september 2003.
- [2] W. Alink, R. A. F. Bhoedjang, Arjen P. de Vries, and Peter A. Boncz. Efficient xquery support for stand-off annotation. In *XIME-P*, 2006.
- [3] G. Bateson. *Steps to an Ecology of Mind*. University of Chicago Press, 2000.
- [4] Gregory Bateson. The logical categories of learning and communication. In *Steps to an Ecology of Mind*, pages 279–308. University of Chicago Press, Chicago, 4 edition, 1971.
- [5] Massimo Bertoncini. On the move towards the european digital library : Bricks, tel, michael and delos converging experiences. In *Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007, Budapest, Hungary, September 16-21, 2007, Proceedings*, volume 4675 of *Lecture Notes in Computer Science*, pages 440–441. Springer, 2007.
- [6] S. Bird and P. Buneman. Towards a query language for annotation graphs. In *Second International Language Resources and Evaluation Conference*, 2000.
- [7] S. Bird, Y. Chen, S. B. Davidson, H. Lee, and Y. Zheng. Extending XPath to support Linguistic Queries. In *Programming Language Technologies for XML (PLANX)*, pages 35–46, january 2005.
- [8] S. Bird, Y. Chen, S. B. Davidson, H. Lee, and Y. Zheng. Designing and evaluating an XPath dialect for linguistic queries. In *22nd International Conference on Data Engineering (ICDE)*, april 2006.
- [9] S. Bird and M. Liberman. A formal framework for linguistic annotation. In *Speech Communication 33(1,2)*, pages 23–60, september 2001.
- [10] M. Blanchot. *Au moment voulu*. Gallimard, 1979.
- [11] S. Boag, D. Chamberlin, M. F. Fernández, D. Florescu, J. Robie, and J. Siméon. XQuery 1.0 : An XML Query Language. Draft, W3C, 2003.
- [12] Daniel G. Bobrow, Linda G. DeMichiel, Richard P. Gabriel, Sonya E. Keene, Gregor Kiczales, and David A. Moon. Common lisp object system specification. *SIGPLAN Not.*, 23(SI) :1–142, 1988.

- [13] J. Boekstraand and A. Kampman. SeRQL : An RDF Query and Transformation Language. In *ISWC'2004*, 2004.
- [14] Andrea Bozzi and Sylvie Calabretto. The digital library and computational philology : The bambi project. In *ECDL*, volume 1324 of *Lecture Notes in Computer Science*, pages 269–285. Springer, 1997.
- [15] T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0. Rec., W3C, 1998.
- [16] D. Brickley and R. V. Guha. RDF Vocabulary Description language 1.0 (RDF Schema). Rec., W3C, 2004.
- [17] L. Brisson. *Platon, oeuvres complètes*. Flammarion, 2008.
- [18] E. Bruno, S. Calabretto, and E. Muriasco. Documents textuels multi structurés : un état de l’art. *Revue i3*, 7(1), 2007.
- [19] E. Bruno, J. Le Maitre, and E. Muriasco. Temporalisation d’un document XML. In *Document Numérique*, volume 8(4), pages 125–141, 2004.
- [20] E. Bruno and E. Muriasco. Describing and querying hierarchical structures defined over the same textual data. In *Proceedings of the 2006 ACM Symposium on Document Engineering (DocEng 2006)*, pages 147–154, Amsterdam, The Netherlands, October 2006.
- [21] Emmanuel Bruno and Elisabeth Muriasco. Msxd : a model and a schema for concurrent structures defined over the same textual data. In *Proceedings of the DEXA Conference, Lecture Notes in Computer Sciences 4080*, pages 172–181, Krakow, Poland, September 2006.
- [22] S. Calabretto, G. Lallich-Boidin, and F. Sedes. Les temps du document numérique. In *Rapport final de l’AS 95 du RTP 33. Site web : <http://fantas-stic.fr.st/>*, 2004.
- [23] Matthew Chalmers. A historical view of context. *Comput. Supported Coop. Work*, 13(3-4) :223–247, 2004.
- [24] N. Chatti, S. Calabretto, and J.M. Pinon. Vers un environnement de gestion de documents à structure multiple. In *Actes de la Conférence Bases de Données Avancées BDA'2004*, pages 47–64, octobre 19-22 2004.
- [25] Nouredine Chatti, Souha Kaouk, Sylvie Calabretto, and Jean-Marie Pinon. MultiX : an XML-based formalism to encode multi-structured documents. In *Proceedings of Extreme Markup Languages'2007, Montréal (Canada)*, August 2007.
- [26] J. Clark. XSL Transformations (XSLT) V1.0. Rec., W3C, 1999.
- [27] J. Clark and S. Derosé. XML Path Language (XPath) V1.0. Rec., W3C, 1999.
- [28] J. Clark and M. Murata. RELAX NG Specification. Technical report, OASIS, 2001.
- [29] A. Dekhtyar, I. E. Iacob, J. W. Jaromczyk, K. Kiernan, N. Moore, and D. Carr Porter. Multihierarchical xml markup of image-based electronic editions. *International Journal on Digital Libraries (IJDL)*, 6(1) :55–69, 2006.

- [30] G. Deleuze. *Cinéma 1 : L'image-mouvement*. Éd. de Minuit, 1983.
- [31] G. Deleuze. *Cinéma 2 : L'image-temps*. Éd. de Minuit, 1985.
- [32] G. Deleuze. *Empirisme et subjectivité : essai sur la nature humaine selon Hume*. Presses universitaires de France, 1993.
- [33] S. DeRose. Markup overlap : a review and a horse. In *Extreme markup language 2004 Conference Proceedings*, 2004.
- [34] Jacques Derrida. Signature, événement, contexte. Congrès international des sociétés de philosophie de langue française <http://www.jacquesderrida.com.ar/frances/signature.htm>, August 1971.
- [35] Jacques Derrida. Trace et archive, image et art. Dialogue. Collège iconique. INA. http://www.jacquesderrida.com.ar/frances/trace_archive.htm, June 2002.
- [36] Paolo D'Iorio. Nietzsche on new paths : The hypernietzsche project and open scholarship on the web. In *Maria Cristina Fornari, Sergio Franzese (éds.), Friedrich Nietzsche. Edizioni e interpretazioni, Pisa ETS*, 2007.
- [37] Karim Djemal, Chantal Soulé-Dupuy, and Nathalie Vallès-Parlangeau. Formal modeling of multistructured documents. In Oscar Pastor, André Flory, and Jean-Louis Cavarero, editors, *RCIS*, pages 227–236. IEEE, 2008.
- [38] Karim Djemal, Chantal Soulé-Dupuy, and Nathalie Vallès-Parlangeau. Management of documentary multistructurality : Case of documentary versions. In André Flory and Martine Collard, editors, *RCIS*, pages 325–332. IEEE, 2009.
- [39] Reim Doumat, Elod Egyed-Zsigmond, Jean-Marie Pinon, and Emese Csiszar. Online ancient documents : Armarius. In *ACM DocEng'08*, Proceeding of the Eighth ACM Symposium on Document Engineering, pages 127–130. ACM, September 2008.
- [40] Paul Dourish. Developing a reflective model of collaborative systems. *ACM Trans. Comput.-Hum. Interact.*, 2(1) :40–63, 1995.
- [41] P. Durusau. Implementing concurrent markup in XML. In *Extreme Markup Languages Conference Proceedings*, august 2001.
- [42] M. Fernandez, A. Malhotra, J. Marsh, M. Nagy, and N. Walsh. XQuery 1.0 and XPath 2.0 Data Model. Draft, W3C, 2003.
- [43] Brian Foote, Brian Foote, Ralph E. Johnson, and Ralph E. Johnson. Reflective facilities in smalltalk-80. In *In Proceedings OOPSLA '89, ACM SIGPLAN Notices*, pages 327–335. ACM Press, 1989.
- [44] Hans-Georg Gadamer. *Truth and Method*. Continuum, 2004.
- [45] C. F. Goldfarb and Y. Rubinsky. *The SGML handbook*. Clarendon Press, Oxford, 1990.
- [46] G. Gonnet and F. W. Tompa. Mind your grammar : a new approach to modelling text. In *the 13th Conference on Very Large Data Bases (VLDB'87)*, pages 339–346, 1987.

- [47] Daniel Hahn, Michele Nucci, and Michele Barbera. The talia library platform - rapidly building a digital library on rails. In *4th Workshop on Scripting for the Semantic Web*, 2008.
- [48] Mirco Hilbert, Andreas Witt, Montréal Québec, and Oliver Schonefeld. Making concur work. In *Extreme Markup Languages*, 2005.
- [49] A. Le Hors, P. Le Hégarret, G. Nicol, L. Wood, M. Champion, and S. Byrne. Document Object Model (DOM) Level 3 Core Specification. Rec., W3C, 2004.
- [50] D. Hume and T.L. Beauchamp. *An enquiry concerning human understanding : a critical edition*. Oxford University Press, USA, 2000.
- [51] I. E. Iacob and A. Dekhtyar. A framework for processing complex document-centric xml with overlapping structures. In *SIGMOD '05 : Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 897–899, New York, NY, USA, 2005. ACM Press.
- [52] I. E. Iacob, A. Dekhtyar, and K. Kaneto. Parsing concurrent XML. In *Proceedings of the 6th ACM International Workshop on Web Information and Data Management (WIDM 2004)*, pages 23–30, november 2004.
- [53] H. V. Jagadish, L. V. S. Lakshmanan, M. Scannapieco, D. Srivastava, and N. Wiwatwattana. Colorful XML : One Hierarchy Isn't Enough. In *SIGMOD Conference*, pages 251–262, 2004.
- [54] W. James. *Pragmatism*. Harvard Univ Pr, 1975.
- [55] W. James, F. Bowers, and I.K. Skrupskelis. *The meaning of truth*. Harvard Univ Pr, 1975.
- [56] Cliff Joslyn. Semantic webs : A cyberspatial representational form for cybernetics. In *Proc. 1996 European Conf. on Cybernetics and Systems Research*, pages 905–910, 1996.
- [57] G. Klyne and J. J. Carroll. Resource Description Framework (RDF). Rec., W3C, 2004.
- [58] A. Korzybski. A non-Aristotelian system and its necessity for rigour in mathematics and physics. Paper presented before the American Mathematical Society at New Orleans, LA, meeting of the American Association for the Advancement of Science, December 28, 1931. Reprinted in. *Science and Sanity*, pages 747–761, 1933.
- [59] Sebastian Ryszard Kruk, Tomasz Woroniecki, Adam Gzella, and Maciej Dabrowski. Jeromedl - a semantic digital library. In Jennifer Golbeck and Peter Mika, editors, *Semantic Web Challenge*, volume 295 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
- [60] Julien Laflaquière, Lotfi Sofiane Settouti, Yannick Prié, and Alain Mille. A trace-based System Framework for Experience Management and Engineering. In *Second International Workshop on Experience Management and Engineering (EME 2006) in conjunction with KES2006*, October 2006.

- [61] C. Laprun, J.-G. Fiscus, J. Garofolo, and S. Pajot. A practical introduction to ATLAS. In *proceedings of Lexical Ressources and Evaluation Conference (LREC 2002)*, Spain, May 2002.
- [62] J. LeDoux. The emotional brain. *New York*, 1996.
- [63] G.W. Leibniz and R. Zimmermann. *Monadologie*. Braum "uller und Seidel, 1847.
- [64] J. LeMaitre. Describing multistructured xml documents by means of delay nodes. In *DocEng '06 : Proceedings of the 2006 ACM symposium on Document engineering*, pages 155–164, New York, NY, USA, 2006. ACM.
- [65] S. Madelrieux. Pluralisme anglais et pluralisme américain : Bertrand Russell et William James. *Archives de philosophie*, (3) :375–393, 2006.
- [66] John H. Maloney and Randall B. Smith. Directness and liveness in the morphic user interface construction environment. In *ACM Symposium on User Interface Software and Technology'95*, pages 21–28, 1995.
- [67] M. Mauss. Essai sur le don. *Sociologie et anthropologie*, pages 143–279, 1950.
- [68] M. Nanard and all. La métaphore du généraliste : acquisition et utilisation de la connaissance macroscopique sur une base de documents techniques. In *Acquisition et Ingénierie des Connaissances - Tendances actuelles. N. Aussenac-Gilles, P. Laublet, C. Reynaud. Toulouse : CEPADUES*, pages 285–304, 1996.
- [69] David M. Nichols, Duncan Pemberton, Salah Dalhoumi, Omar Larouk, Claire Belisle, and Michael B. Twidale. Debora : developing an interface to support collaboration in a digital library. In *European Conference on Digital Libraries*, pages 239–248. Springer, 2000.
- [70] C. Péguy. *Clio : dialogue de l'histoire et de l'âme pa "ienne*. Ed. de la Nouvelle revue française, 1917.
- [71] C. S. Peirce. *Collected Papers of Charles Sanders Peirce*. Harvard University Press, 1931.
- [72] Charles S. Peirce. *Écrits sur le signe*. Seuil, Paris, 1978.
- [73] Silvio Peroni and Fabio Vitali. Annotations with earmark for arbitrary, overlapping and out-of order markup. In Uwe M. Borghoff and Boris Chidlovskii, editors, *ACM Symposium on Document Engineering*, pages 171–180. ACM, 2009.
- [74] Pierre-Edouard Portier and Sylvie Calabretto. Classement collaboratif de manuscrits. In LSIS-USTV, editor, *RJCRI/CORIA 09 (Conférence en Recherche d'Information et Applications)*, pages 511–518, May 2009.
- [75] Pierre-Edouard Portier and Sylvie Calabretto. Creation and Maintenance of Multi-structured documents. In *ACM DocEng 2009, DocEng '09 : Proceedings of the 9th ACM symposium on Document engineering*, pages 181–184, September 2009.

- [76] Pierre-Edouard Portier and Sylvie Calabretto. Methodology for the construction of multi-structured documents. In *Balisage : The Markup Conference 2009, Montréal, Canada, August 11 - 14, 2009*, Balisage Series on Markup Technologies, vol. 3 (2009), August 2009.
- [77] Pierre-Edouard Portier and Sylvie Calabretto. Modélisation des connaissances dans le cadre de bibliothèques numériques spécialisées. In Jean-Gabriel Ganascia and Pierre Gançarski, editors, *Extraction et gestion des connaissances*, Revue des Nouvelles Technologies de l'Information, pages 391–396. Cépaduès-Editions, January 2009.
- [78] Pierre-Edouard Portier and Sylvie Calabretto. Méthodologie pour la création de documents multistructurés. In André Péniuniou Max Chevalier, editor, *27ème Congrès INFORSID'2009*, pages 211–226. IRIT Press, May 2009.
- [79] Pierre-Edouard Portier and Sylvie Calabretto. DINAH, a philological platform for the construction of multi-structured documents. In Joemon Jose Mounia Lalmas, editor, *The European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, Research and advanced technology for digital libraries, pages 364–375. springer, September 2010.
- [80] Pierre-Edouard Portier and Sylvie Calabretto. Multi-structured documents and the emergence of annotation vocabularies. In *Balisage : The Markup Conference 2010*, Balisage Series on Markup Technologies, August 2010.
- [81] Pierre-Edouard Portier and Guy Caplat. Autour de la confrontation de modèles. In éditions Tec&Doc Lavoisier, editor, *MOSIM : Conférence Internationale de Modélisation et Simulation*, pages 1977–1984, March 2008.
- [82] L. Poulet, S. Calabretto, and J.M. Pinon. Semantic structuring of documents. In *Proceedings of the Third Basque International Workshop on Information Technology, BIWIT'97*, pages 118–124, july 1997.
- [83] A. Renear, E. Mylonas, and D. Durand. Refining our notion of what text really is : The problem of overlapping hierarchies. In *Research in Humanities Computing*. Oxford University Press, 1996.
- [84] Fred Rivard. Smalltalk : a reflective language. In *In Proceedings of REFLECTION '96*, 1996.
- [85] Jim des Rivières and Brian Cantwell Smith. The implementation of procedurally reflective languages. In *LFP '84 : Proceedings of the 1984 ACM Symposium on LISP and functional programming*, pages 331–347, New York, NY, USA, 1984. ACM.
- [86] J.D. Robert. Maurice Clavelin, La philosophie naturelle de Galilée. Essai sur les origines et la formation de la mécanique classique. *Revue Philosophique de Louvain*, 71(10) :338, 1973.
- [87] F. Saussure. Cours de linguistique générale (1916). *Paris : Payot*, pages 1967–74, 1949.

- [88] Andrea Scotti and Daniele Nuzzo. Pinakes – a modeling environment for scientific heritage database applications. In *Proc. of Reconstructing science – Contributions to the enhancement of the European scientific heritage Workshop*, Ravenna, Italy, 2001.
- [89] Lotfi Sofiane Settouti, Yannick Prié, Pierre-Antoine Champin, Jean-Charles Marty, and Alain Mille. A Trace-Based Systems Framework : Models, Languages and Semantics. Research Report, 2009. Work in progress.
- [90] Herbert A. Simon. The architecture of complexity. In *Proceedings of the american philosophical society*, volume 106, December 1962.
- [91] G. Simondon. *L'individuation psychique et collective*. Aubier, 1992.
- [92] C. M. Sperberg-McQueen and L. Burnard. TEI P4 Guidelines for Electronic Text Encoding and Interchange, 2001.
- [93] C. M. Sperberg-McQueen and C. Huitfeldt. Goddag : A data structure for overlapping hierarchies. In *DDEP/PODDP*, pages 139–160, 2000.
- [94] Spinoza. *ETHICA Ordine Geometrico Demonstrata*. Éditions du Seuil, 1999.
- [95] J. Stegmann and A. Witt. Tei feature structures as a representation format for multiple annotation and generic xml documents. In *Proceedings of Balisage : The Markup Conference 2009. Balisage Series on Markup Technologies, vol. 3 (2009)*. doi :10.4242/BalisageVol3.Stegmann01., august 2009.
- [96] Adelheit Stein, Jürgen Keiper, Laura Bezerra, Holger Brocks, and Ulrich Thiel. Collaborative research and documentation of european film history : The collate collaboratory. In *In International Journal of Digital Information Management (JDIM), special issue on “Web-based collaboratories – from centres without*, pages 30–39, 2004.
- [97] J. Tennison. Creole : Validating overlapping markup. In *Proceedings of XTech 2007 : The Ubiquitous Web Conference*, May 2007.
- [98] J. Tennison and W. Piez. Layered Markup and Annotation Language (LMNL). In *The Late breaking paper presented at Extreme Markup*, 2002.
- [99] G. Tummarello, C. Morbidoni, and E. Pierazzo. Toward an RDF and Logic Programming based framework for textual content markup and querying. In *Digital Resources for the Humanities conference (DRH 2005)*, september 2005.
- [100] Valentin F. Turchin. *The phenomenon of science*. Columbia University Press (New York), 1977.
- [101] Valentin F. Turchin. Refal : the language for processing xml documents. http://www.refal.ru/english/xmlref_1.htm, January 2000.
- [102] A. Witt. Multiple hierarchies : New aspects of an old solution. *Interdisciplinary Studies on Information Structure*, pages 55–85, 2005.