BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

N° 42 DECEMBRE 1995

SCIENCES DE L'EDUCATION ET DE L'INFORMATION

Publication gratuite trimestrielle de l'Université d'Aix - Marseille II
58, boulevard Charles Livon. F - 13284 Marseille Cedex 07
Téléphone : (33) 91 39 65 00 Télécopie : (33) 91 31 31 36

Edition 1997

ISSN 0291 - 5413

BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

N° 42 DECEMBRE 1995

SCIENCES DE L'EDUCATION ET DE L'INFORMATION

Publication gratuite trimestrielle de l'Université d'Aix - Marseille II
58, boulevard Charles Livon. F - 13284 Marseille Cedex 07
Téléphone : (33) 91 39 65 00 Télécopie : (33) 91 31 31 36

Edition 1997

ISSN 0291 - 5413

EDITORIAL,

Informatique et devenir.

Il est une époque, d'apparition périodique, où tout devient étrange. Le réel cesse d'être le réel. On plonge dans une sorte de conte de fées, avec ses personnages baroques, ses situations du merveilleux et de l'horrible qui se mélangent et suintent dans une teinte écœurante et fétide. Mais, dont le dénouement, contrairement aux contes anciens, et un peu comme dans les contes modernes, est loin de sacrifier à la Sainte Morale.

Il s'agit des périodes électorales.

Comment alors empêcher "Vouzzavédibisar" et "Editorial" de se confondre un peu ? Tant le tragique touche au comique, comme une certaine gauche confine à la droite tandis que le centre ...

La société subit un bouleversement fantastique, à une échelle encore jamais atteinte, et les campagnes électorales ressemblent à la partie de cartes de César et Escartefigue: « ... à moi il me fend le cœur, et à toi il te fait rien ? ... » Tout le monde aura compris que c'est du Borgne qu'il s'agit, la terreur des urnes.

Pourtant les exemples ne manquent pas dans l'Histoire, d'inventions dont l'intérêt, indiscutable du point de vue de leur utilité, était compensé par le ravage du milieu social. Ce qui a été le cas de toute mécanisation. La diffusion du métier Jacquard a saccagé toute une corporation d'artisans tisserands, les Canuts; mais les événements qui en furent les conséquences sont oubliés depuis longtemps, pensez c'était au début du XIX ème siècle.

Mais jusqu'à présent, la mécanisation, génératrice de chômage s'était étalée sur plus de deux siècles et ne touchait en général qu'une corporation à la fois. Ce qui permettait de juguler plus aisément certaines réactions violentes.

Par ailleurs si la machine supprimait des emplois là où elle s'implantait, elle en créait d'autres car de nouveaux chantiers devenaient possibles. La Terre était considérée comme illimitée, on pouvait donc dévaster et ravager à loisir, la Ville prenait une nouvelle extension. En outre l'illusion pouvait se poursuivre grâce à la dernière guerre, je veux dire celle de 39-45 - que ne préférait pas Brassens - car tout était à reconstruire. Les lendemains qui chantent et "cœtera".

Et puis l'informatique est arrivée. En catimini. Dans les hautes instances, personne ne s'est aperçu de son intrusion. Et encore moins qu'elle envahissait tous les secteurs, en silence, mais de manière inéluctable et implacable; tous les domaines; sans exceptions. La machine à vapeur du XIX ème siècle s'était implantée partout où il y avait quelque chose à faire tourner, l'informatique s'incruste partout où quelque chose bouge, au réel comme au figuré.

L'ordinateur, depuis qu'il sait se miniaturiser s'infiltre comme un virus. On en trouve même sous la peau de certains malades cardiaques. Un Président, si important qu'il puisse encore se croire n'est plus qu'un banal sous-produit de l'informatique, c'est dire. L'automatisation, par sa souplesse, a multiplié le pouvoir de la mécanisation par un facteur immense. Toutes les corporations sont touchées, même dans le secteur du livre où, par exemple, l'ordinateur a remplacé des masses d'ouvriers particulièrement qualifiés, sans offrir en échange l'équivalent en nouveaux emplois.

On aurait pu se réjouir qu'enfin les tâches pénibles, sans grand intérêt culturel, voire les tâches dangereuses puissent enfin être confiées à des automates. Encore eut-il fallu que les "dirigeants" dirigeassent. Gouverner c'est prévoir, certes. Mais comment prévoir quand on est bardé de super diplômes "des", voire de la plus grande école, et qu'on est exclusivement formé à absorber des produits pré mâchés, attitude garante de la réussite. Et je ne parle pas de la masse des subalternes dont la seule activité consiste à lécher la plus grande surface sociale possible pour asseoir une carrière ainsi bien méritée.

Tous les "Saint Jean Bouche d'Or"(*) qui remplissent leur programme électoral de mille et une promesses écrites avec l'encre du "bonheur", un peu comme la Reine d'un Jour, auraient enfin sous la main le moyen de promulguer la mise en œuvre d'une société de rêve. Une société dans laquelle la notion de travail pourrait être régénérée. Les tâches ingrates, inhumaines, désormais inclues dans la machine, réserveraient aux hommes le temps libre et les activités créatrices.

Mais c'est compter sans la notion de profit.

C'est compter sans l'ignorance crasse de l'essentiel de la classe dirigeante. Les penseurs de Rodin de la Politique demeurent imperturbablement aveugles, noyés qu'ils sont, au milieu de leurs petits problèmes issus des siècles périmés, préoccupés de leur petite survie. Surtout respectueux de ce qui assure l'illusion de leur pérennité.

En fait, ils sont l'essence d'un milieu qui exerce une certaine forme de pouvoir, et ils ne savent et ne peuvent, sous peine d'autodestruction, que reproduire les schémas du milieu qui les a conditionnés. Quel que soit leur horizon politique, d'ailleurs. Il suffit de voir leur air embarrassé quand ils sont obligés de constater la vanité de l'application de leurs petites recettes, qui se ressemblent toutes d'ailleurs, ne variant que selon des dosages qui semblent les distinguer, alors qu'elles sont parfaitement identiques dans leur inanité voire leur nuisance.

Mais il faut reconnaître que le moyen de pression est particulièrement puissant. Les nazis l'avaient bien compris qui inscrivaient aux frontons de leurs camps de la mort:

« Arbeit macht frei ! »

Le point d'exclamation est de moi, car cette devise s'applique parfaitement à notre actuelle époque de crise. Demandez aux chômeurs de longue durée et aux S.D.F.

Tu as du travail, tu es un homme (jusqu'à nouvel ordre), tu n'en as pas, tu n'existes plus.

Quelle que soit la couleur du pouvoir, et quels que soient les gens qui l'exercent, on retrouve toujours dans son expression la même propriété fondamentale. Quelle que soit la quantité de richesses en jeu et quel que soit leur taux d'accroissement, le pouvoir politique n'a pas les moyens ou ne se les donne pas, de répartir les richesses; par contre il a celui de répartir les charges; les impôts, les taxes. Avec, en général, un sens très particulier de l'équité. La société ne cessant de s'enrichir, et l'enrichissement s'accroissant de manière phénoménale, en même temps que la population s'appauvrit, il est clair que le pouvoir n'est rien d'autre qu'un gestionnaire du gaspillage. Plus le pouvoir est grand, plus le gaspillage est fort. Des sommets ont été atteints avec les pouvoirs totalitaires qui se sont accomplis en d'abominables ravages.

Et l'informatique non maîtrisée ne peut que pousser encore dans ce sens car elle accroît le pouvoir de ceux qui ont d'autant plus de moyens qu'ils sont dénués de scrupules.

L'économie est une hydre, et les politiciens sont ses esclaves; parviendront-ils un jour à l'enchaîner ? Le veulent-ils ?

Edmond Bianco

(*) : (304-407) Jean Bouche d'Or ou Jean Chrysostome, Père de l'Eglise d'Orient et patriarche de Constantinople. Il fut célèbre pour son éloquence, aussi.

Modélisation, simulation et représentation en IA.

Pierre Livet
(C Pitrouillau) -

A sa naissance, l'IA s'est définie comme la possibilité de réaliser en machine des opérations cognitives ou intellectuelles. pas forcément humaines, puisqu'on pourrait imaginer des mondes possibles où des intelligences non humaines fonctionnent. Il ne s'agissait donc pas d'imitation. L'utilisation comme modélisation de ce que faisait l'IA a donné lieu à la psychologie cognitive. Donc l'IA faisait de la simulation, mais en un sens large: il ne s'agit pas de se borner à reproduire les outputs cognitifs humains à l'aide de machines. On peut produire aussi d'autres outputs. En tous les cas on ne garantit en rien la similitude entre les opérations employées et les opérations psychologiques humaines. Le test de Turing suppose que ces opérations sont en boite noire. Le General Problem solver supposait que il y a des pptés générales de l'intelligence, sans se soucier des domaines particuliers, et que on peut les traduire dans une machine qui traite des symboles. (traitement de l'information).

Qu'en est-il maintenant?

Il y a d'abord eu des changements: l'idée d'une intelligence en général s'est révélée trop ambitieuse, il a fallu revenir à des programmes domaines spécifiques. Puis on s'est aperçu que des procédures de programme n'étaient pas satisfaisantes à elles-seules. Il fallait des représentations du monde dans lequel on agissait, on "procédait", et à partir de là on pouvait tirer des inférences. On s'est aperçu ensuite que l'organisation de ces représentations pouvaient introduire des biais, et que d'un autre coté, il fallait définir quelles procédures avaient accès à quelles données (langage orienté objet). On a pu ajouter à savoir déclaratif et savoir procédural un savoir épisodique (quoi faire quand)

Tout cela ne semble pas remettre en cause le programme "fonctionnaliste" de la simulation en IA: trouver les fonctions qui permettent de produire des outputs similaires aux performances cognitives humaines ou autres. (l'humain sert ici simplement de critère de reconnaissance, pas de critère d'essentialité ou de nature).

Mais il existe au moins deux tendances en IA: 1) celle qui pense que cette étude des fonctions qui produisent des résultats que l'humain reconnait comme cognitif est une étude qui n'a pas à se soucier des propriétés particulières de la cognition humaine. Elle doit utiliser les critères humains uniquement comme point de départ, et ensuite, elle propose des définitions théoriques des ces fonctions, elle tente des implémentations des fonctions ainsi définies, et elle n'a plus alors pour critère que les facilités ou les obstacles qui se manifestent dans la mise au point des programmes. C'est une IA autonomisée, qui se donne à elle-meme ses propres critères (rapidité d'exécution, peu de modifications à faire au programme quand on change de domaine,

assurance que le programme fait bien ce qu'il est censé faire, capacité des concepts de base de trouver des extensions sans qu'on ait à modifier les règles de construction et d'utilisation de ces concepts, etc.).on pense alors qu'il y a une intelligence générale, à la fois pour humaine et pour computers. (Soar).

2) celle qui pense a) que l'IA a intéret à repérer les propriétés de la cognition humaine dont manquent les programmes en cours, et à tenter de retrouver ces propriétés (peut-etre par d'autres moyens que ceux humains). Ou encore, b) plus humanoide, celle qui pense que si on se rapproche du neuronal et du biologique, on pourra plus facilement retrouver ces bonnes propriétés dont manquent les programmes IA actuels. Dans cet esprit, on ne prend pas forcément pour repère l'homme, mais l'animal en général, pris dans l'évolution. (Brooks). Un problème qu'on évoquera en a et en b, c'est de savoir si la cognition peut se faire sans ancrage dans le monde, par des senseurs et de la motricité. D'où la réémergence du robot.

3) ceux qui pensent l'IA dans l'interaction homme-machine. Il n'y a pas alors à simuler, mais à respecter des contraintes imposées aux programmes par les formes de la cognition humaine (et ces contraintes peuvent etre bricolées, mais pas indéfiniment, donc il faut avoir une idée des contraintes cognitives humaines, et ensuite construire des machines qui s'y adaptent, et qui peuvent exponentier la capacité cognitive humaine au niveau du collectif.

On peut partir de l'article de D. Kirsh dans AI 47. Il fait la liste des hypothèses de base de l'IA classique. Puis il repère les mises en question de ces hypothèses.
Hypothèses de base: 1) (conceptualisation) le noyau de l'IA, c'est la conceptualisation de monde utilisée par des systèmes intelligents. Donc l'association entre un savoir déclaratif et un calcul qui est aussi un raisonnement. Cette conceptualisation se amnifeste principalement par des représentations et des règles d'inférence.
2) (désincarnation) On peut les étudier en faisant abstraction de l'incarnation, i e des implémentations des liens réels au monde réel, donc de perception (senseurs) et de de controle moteur.
3) (opérations formelles) les opérateurs qui transforment, déplacent le savoir peuvent s'exprimer dans un langage logico-mathématique ou encore un langage du type langage naturel appauvri et formalisé.
4) (statisme) on peut séparer l'étude du savoir, des inférences, des opérations de retrieval, etc, et l'apprentissage ou le changement évolutif du système cognitif.
5) (universalisme) toute cognition doit pouvoir se formuler dans une architecture des opérations, et des fonctions, conservant les memes principes et concepts de base.
On peut ajouter 6) (individualisme) un système unique peut parvenir à tout traitement cognitif.

3) (non localité) le traitement cognitif est le meme en un lieu et situation et en général

4) (déterminabilité) Il faut éviter l'indétermination, l'incommensurabilité des représentations.

5) (multi niveau) les capacités cognitives sont fondées sur une architecture en multiples niveaux, qui sont plus ou moins reliés à des temps de traitement.

II Les positions diverses

Tout le monde est d'accord sur 10, mais évidemment les versions de ces niveaux sont assez différentes. Par exemple Soar propose : niveau neural, niveau cognitif proprement dit, qui comprend niveau de l'accès aux symboles (mémoire), niveau des opérations de délibération élémentaires (décision), niveau de la composition simple d'opérateurs et de l'ajustement à des buts (gaol) ; puis le niveau rationnel, avec adaptation et buts complexes. Les connexionnistes vont penser qu'il y a niveau sub - symbolique, auquel peut déjà se passer catégorisation, donc inférence immédiate, puis combinaison avec inférence "rationnelle", ie qui demande des changements des paramètres (Hinton) . Les motobotistes vont supposer un niveau d'automatismes d'évitement, sur lequel s'empile un niveau de gradient de direction, etc.

Le logicisme convient pour les 5 (ne dit rien cependant sur l'architecture). Mais Soarn le plus classique, ne se veut pas logiciste: le logicisme serait valide seulement pour niveau rationnel. L'IA aurait pour tache de définir les contraintes liées aux opérations du niveau cognitif, pas du niveau rationnel (ni du niveau neuronal).

Cependant, peu de programmes actuels en IA les adoptent tous les 5. par exemple; Soar adopte 1, 2, 5. Mais il implique une forte importance de l'apprentissage. Il ajoute, comme la plupart, au langage logico-mathématique des "outils" IA (des biais de décision, des marqueurs de controle, des préférences, qui permettent des décisions et évitent les ties. C'est là une nécessité de l'action, et donc aussi l'idée que quand on est dans une impasse, il faut engendrer un nouveau sous-but qui sera de résoudre l'impasse. Ce qui n'est pas logiciste. De meme l'idée que l'apprentissage doit se faire en termes de chunks, qui évitent les impasses en mémorisant les préférences qui ont permis de s'en sortir. Là aussi, cette idée de ne pas utiliser tout l'espace de recherche possible n'est pas logiciste, et c'est elle qui fait le centre de l'IA).

Les connexionnistes peuvent tout refuser (sur les 5 premiers), ou accepter 2. Ils voudront cependant relever le défi de la conceptualisation et du raisonnement (1)

Les motobotistes (Brooks etc.) refusent tous les 5.

les IA distribuée accepte les 5, mais avec modération quant aux représentations (ce peuvent etre des routines sans représentations) et refuse 6 à 9.

Un trait de l'IA, qui montre que l'humain ne constitue pas seulement un critère de reconnaissance des la validité des outputs, mais aussi une heuristique de

recherche, meme pour les universalistes, etc. c'est que les unités conceptuelles sont empruntée aussi bien à la vie de tous les jours (scénarios de Schank, Lenat CYC), et qu'on va donc meler des concepts et opérateurs techniques et des catégories qui sont supposées avoir une origine ordinaire, meme si on en limite de champ d'application pour des raisons liées aux contraintes opératoires.

TCT De l'Ontologie à la pragmatique

On peut ajouter en ce sens que si l'IA était autonomisée, elle pourrait etre compilation d'un bout à l'autre. Or il semble que le déclaratif soit toujours nécessaire, pour des raisons qui tiennent plus à l'articulation avec les humains qu'à autre chose. Ainsi le déclaratif est plus accessible et à l'utilisateur, et au concepteur, qui peut ainsi accéder à sa modélisation au lieu que ce soit une simulation opaque. Bien entendu, le déclaratif est aussi privilégié dans un esprit logiciste, qui consiste à vouloir utiliser la machine pour des buts que le designer n'a pas eu (mais là encore, traductibilité pour différents points de vue humains - utilisateur-concepteur, autre concepteur- sont sous jacentes dans cet apparent logicisme. Il n'est donc pas évident que la distinction entre déclaratif et procédural repose sur des exigences logicistes.Car le logicisme pourrait aussi prétendre que tout est inférence (règles d'inférences, y compris une recognition) La volonté d'articuler savoir et inférence repose donc davantage sur une ontologie implicite de l'IA, sur la volonté d'inscrire une ontologie dans les machines, de manière à pouvoir partager avec elle un environnement cognitif. C'est là une sorte d'option philosophique sur l'homme (sur un idéal formel d'homme) que l'on reporte que la machine IA. Il n'y a donc pas simulation, mais normalisation, idéalisation humaine à rendre effective.

Une autre hypothèse qui découle de la conjonction de 1 et de 3, c'est que l'on ne va pas en IA classique chercher à accrocher la conceptualisation à la réalité. Donc on va pouvoir faire des catégorisations, les lier à des opérations, tout cela sans se demander ce que saisitla machine, i e en quoi sa sémantique lui permet une sorte d'ancrage, de référence dans le monde. Or les concepts ordinaires utilisés par le concepteur viennent de ces ancrages. Donc avec le déclaratif, on a fait en quelque sorte de l'ontologie sans référence. Ce déclaratif peut se réduire à l'encodage propre au système des catégories définies. Ou bien on peut désigner par là les catégories du concepteur en tant qu'elles imposent des contraintes aux opérations par exemple on ne pourra utiliser ensemble des cadres qui sont de niveaux différents (ou encore, plus vers les opérations, cf. le coarse coding qui ne permet pas de répéter un symbole). Est-ce qu'on arrive réellement à ce que l'ontologie créée par ces contraintes reste interne au système? Le problème est que l'IA est toujours interactive. Et il semble que les flexibilités humaines servant toujours de benchmark,

une ontologie qui serait seulement interne ne nous permettrait pas de porter un jugement "universel" sur les opérations du système, sur son mode de fonctionnement. Il y a là une tension dans la position logiciste, universaliste et uniformisante: elle ne permet pas une autonomisation complète, alors meme qu'elle semblait réaliser dans le système l'etre cognitif idéal qu'elle avait d'abord formalisé.

Un point essentiel, ce sont les différences d'approches dans le rapport entre le concepteur et la machine. Dans l'approche universaliste, on suppose que le concepteur doit déjà tout savoir sur le domaine. Ce que peut apprendre la machine, c'est seulement une méthode plus rapide et plus praticable pour mobiliser ce savoir, pour le relier à d'autres savoirs, etc. Ceci meme dans l'idée "systèmes experts" (là, c'est la mobilisation du savoir de l'expert, donc hors machine, que le concepteur est censé dégager dans l'interaction avec la construction du programme).

Au contraire dans des approches et connexionnistes et motobotistes, on suppose que la construction de la machine va nous donner des idées sur les bonnes catégories dans un domaine dans lequel on ne dispose pas de théorie correcte, ni de savoir d'expert. Mais alors la machine ne peut pas etre considérée comme <u>simulation</u>, mais comme <u>exploration cognitive</u>. Elle nous révèle des catégories, éventuellement différentes des notres. Le seul controle que nous ayons, c'est de juger de la tache accomplie, mais pas des représentations internes nécessaires (ni meme, par conséquent, de l'ontologie ou de la carégorisation de la machine). Nous jugeons non pas la correspondance avec notre sémantique, mais par la coordination pragmatique avec nos propres actions. Aurions nous une sorte de savoir implicite mais généralisable sur nos capacités de coordination, leur évantail de possibles? Ce type d'IA serait censé nous le révéler. Mais cela ne dispense pas de modéliser ce que fait la machine (soit pour tenter de déterminer ses meilleures performances, ses limites pour certaines taches (connexionnisme) soit pour tenter de la mieux construire, avec un comportement plus riche et plus adapaté. Mais on a là non plus une modélisation de l'intelligence ou du cognitif en général, mais du système cognitif d'un type de machine. On explore une nouvelle zoologie, l'artefactologie.

Le problème de savoir quelle est l'ontologie d'une machine est alors pris dans un autre sens (premier sens: on a la syntaxe, on connait les opérations internes, il reste à ancrer via du sensori-moteur, et donc à vérifier que c'est bien une sémantique causalement articulée sur le monde, via éventuellement nos propres critères pragmatiques).

cependant elle a trois niveaux de tests: les robots, leur efficacité; la reconnaissance de compatibilité de construction (désincarnée) avec l'utilisateur concepteur. la reconnaissance de similarité avec ce que nous observons, et ce que l'IA nous fait observer, de cognition humaine ou animale. Mais L'IA ajoute à l'ontologie

qui donnerait simplement les catégories et pptés du domaine (environnement), la spécification par les capacités cognitives de l'agent, par ses capacités d'actions, et par ses buts. Donc elle y ajoute un pragmatique, qui est elle-meme conceptualisée. On ne peut se satisfaire d'une "modèle theorie" parce qu'elle ne nous dit pas pourquoi choisir tel modèle (sémantique). On doit articuler sur le modèle sémantique l'analyse des critères d'évaluation d'une pragmatique.

Donc on part de l'articulation pragmatique, et on construit la sémantique comme compatibilité entre les contraintes de construction et les possibilités pragmatiques observées ou supposées. Dans la première perspective (celle du modèle theorie), les concepts, par exemple, sont des composants modulaires d'un réseau sémantique et opératoire. Ils peuvent etre ou non dotés d'un ancrage référentiel. Dans la seconde, l'ancrage est pragmatique (conditions de réussite) et on remonte vers les classifications et opérations de transformation de ces classifications (connexionnisme) pour retrouver les concepts, les catégorisations qui permettent ces réussites et ces échecs. par exemple, on va exiger une indexicalité, une égocentricité du système. Dans Brooks, on ne remonte que vers les contraintes de dérivation d'un système sur un autre, donc non pas des concepts, mais des architectures entre diverses opérations, et ce en fonction de leur échecs observés. A la limite, l'IA n'est qu'un affaire d'ingénieurs, et pas de catégorisateurs. Pas de philosophie de l'esprit là dedans?

Mais la philosophie de l'esprit impose certains idéaux (systématicité), et les moyens d'y arriver (compositionnalité) comme des contraintes qui rélèguent les classifications connexionnistes au range de pseudo-concepts. C'est le retour de l'idéalisation comme imposition normative au design. Il y a là un problème de statut, de reconnaissance pour la simulation IA. Par exemple, pas de concept sans réseau de concepts et sans généralité. Ce n'est pàs là une benchmark pratique, mais une qualification théorique à atteindre. Toute modélisation se doit de définir sa distance par rapport à ces ideaux normatifs, qui peuvent aussi se transformer en hypothèses restrictives permettant de se donner de bonnes conditions (normales, idéales). Or l'IA est en porte à-faux ici, parce qu'elle doit aussi se donner du praticable: elle ne peut donc utiliser le normatif comme conditions idéales restrictives (comme le fait l'économie théorique). Ici la simulation en temps réel, etc. reprend donc le pas sur la modélisation idéalisée.

TV

La modélisation qui passe par la simulation

Au lieu que l'on parte de la modélisation pour simuler, ou qu'on fasse de la simulation une réalisation de la modélisation (en ajoutant des contraintes de praticabilité), on part de la simulation pour modéliser. Nous sommes, nous humains,

des etre engagés dans des simulations en temps réel qui nous apprennent des impossibilités, qui se heurtent à des contraintes. Alors nous pouvons modéliser ces contraintes, en les resimulant sur des machines en interaction entre elles, avec un environnement et avec nous. Nous ne faisons là que les modéliser, sans pouvoir les théoriser de manière définitive (puisque cela dépend de deux simulations, celle des machines et la notre, et par exemple dans connexionnisme, il faut attendre que la machine ait appris pour tenter ensuite de savoir ce qu'elle représente grace à cet apprentissage; c'est donc bien de la simulation, en temps réel. De meme le savoir casuel est sensible à l'ordre de son apprentissage; etc.).

L'utilisation du pragmatique: la "compliance" dans une main qui va attraper la chose dès qu'elle sera à portée. Ou encore les committments de Gasser (qui permettent de garder des repères dans des conflits entre représentations différentes). La communication apparaissant comme la maintenance de ces engagements à travers des contextes différents selon chaque agent. Dans les deux cas, on évite de devoir planifier entièrement la coordination et de la réduire à un meme format. L'engagement peut d'ailleurs n'etre pas déontique et se réduire à l'utilisationde ressources qui ne sont pas disponibles ni pour tout à tout instant ni n'importe comment. La limitation de nos pratiques sert de guide pour nos coordinations. (il y a négociation pour régler les différents entre engagements, donc les différences de perspectives au second degré). Là encore on retrouve l'idée que l'indétermination sera levée simplement parce qu'il faut agir et donc trouver une détermination. On voit que l'IA ne modélise plus les capacités cognitives d'un système individuel, mais qu'elle modélise la façon dont les systèmes individuels simulent leurs interactions et se guident sur les contraintes rencontrées dans ces simulations pour se coordonner.

Nous avons donc mis la machine entre nous et nous et l'environnement. Donc le problème de la représentation devient le problème de savoir ce qui est transposable entre la machine, nous , et l'environnement. (l'environnement est bien sur uniquement ce que nous en représentons, ou ce que la machine en représente). nous pouvons encoder dans la machine des représentations qu'elle ne relie pas à un environnement. Nous pouvons définir des "concepts" ancrés, mais il se peut que la machine en ait une "représentation" différent de nous (cas du connexionnisme). Dans l'idéal nous devrions avoir les memes concepts. Cela en fait semble impossible, à cause du point de départ pragmatique, qui n''utilisele modèle que comme échangeur, n'impliquant rien sur l'ontologie propre de chaque entité. On passe de la sémantique dénotationnelle (référentielle) à une sémantique fonctionnelle, dit Kirsch.

Que devient la théorie?

Les bouts de théorie que nous avons (et qui nous permettent de reprendre le cheminement classique, théoriser, modéliser, simuler) nous permettent

essentiellement de définir des contraintes et des impossibilités pour nos simulations, donc des délimitations de champs pour nos modélisation. Reste que nous pouvons aussi utiliser ces théories comme des modèles normatifs, qui nous permettent de juger qu'une simulation ne permettra jamais tel type de modélisation et qu'elle est donc insuffisante (Fodor). Mais cela n'empeche plus l'IA de continuer à simuler (pour modéliser ensuite), puisque la simulation est censée nous faire découvrir d'autres contraintes. Pour l'instant, il ne semble pas qu'elle nous permette de découvrir d'autres modèles normatifs. Pour cela il reste toujours nécessaire de passer par une théorie. De plus l'exigence d'une théorie est nécessaire quand on se trouve en face d'une multiplicité de petits systèmes ad hoc qui n'ont rien de commun entre eux. mais ic encore , la théorie ne nous donne pas une ontologie, mais les contraintes pour une coordination dans les échangeurs entre systèmes.

(Brouillon) de l'exposé aux séminaires de recherche 1992. Séminaires organisés par le groupe de formation doctorale d'informatique fondamentale et sciences de la computation.

Les annotations sont de Pierre Livet

Encodage des dictionnaires électroniques: problèmes et propositions de la TEI

Jean Véronis et Nancy Ide*

Laboratoire Parole et Langage
U.R.A. 261 CNRS et Université de Provence
29, Avenue Robert Schuman
13621 Aix-en-Provence (France)

Résumé

Cet article décrit les principaux problèmes auxquels la *Text Encoding Initiative* (TEI) a dû faire face pour définir un standard d'encodage des dictionnaires électroniques. Ceux-ci sont, à cause de leur haut degré de structuration et de complexité, parmi les types de textes les plus difficiles traités par la TEI. Les problèmes les plus délicats étaient (1) le conflit entre la généralité de la description visant à représenter le plus grand nombre possible de dictionnaires, et son pouvoir descriptif, c'est-à-dire la capacité à décrire de façon précise la structure particulière d'un dictionnaire donné; (2) la nécessité de rendre compte de points de vue différents sur les dictionnaires encodés, par exemple, comme objet imprimé ou comme base de données.

---

* Nancy Ide est la fondatrice de la TEI, et préside son comité de pilotage.

1. Introduction

1.1. La *Text Encoding Initiative*

La *Text Encoding Initiative* (TEI) est un projet international qui a été créé en 1988 sous l'égide de l'*Association for Computers and the Humanities,* de l'*Association for Computational Linguistics*, et de l'*Association for Literary and Linguistic Computing,* et qui vise à la mise au point d'un ensemble de normes pour la préparation et l'échange de textes électroniques (voir l'historique et la description du projet dans IDE/SPERBERG-McQUEEN, 1995). Le projet a été financé par le *U.S. National Endowment for the Humanities*, la Commission Européenne (DG XIII), la fondation Andrew W. Mellon, et le *Social Science and Humanities Research Council* du Canada.

En mai 1994, la TEI a publié ses *Guidelines for the Encoding and Interchange of Machine-Readable Texts,* connues sous le nom de TEI P3 (SPERBERG-McQUEEN/BURNARD, 1994; voir aussi IDE/VERONIS, 1995), qui proposent un ensemble de conventions d'encodage pour de nombreux types de textes et une grande variété d'applications: publication électronique, analyse littéraire et historique, lexicographie, traitement automatique des langues, recherche documentaire, hypertexte, etc. Les *Guidelines* s'appliquent aux textes écrits ou parlés, sans restriction de langue, de période, de genre ou de contenu et répondent aux besoins fondamentaux de nombreux d'utilisateurs, lexicographes, linguistes, philologues, bibliothécaires, et, de manière générale, de tous ceux qui sont concernés par l'archivage et l'accès à des documents électroniques.

Les règles et recommandations proposées dans les *Guidelines* sont basées sur le langage SGML (*Standard Generalized Markup Language*), qui est un standard international (ISO 8879:1989) d'un usage de plus en plus répandu, et dont nous supposerons ici les principes connus du lecteur (voir par exemple l'introduction de BURNARD, 1995). Rappelons simplement que SGML est un *méta-langage* qui précise des règles permettant la définition de systèmes de *balises* pour chaque type de texte. En règle générale, les éléments du texte sont encadrés par des balises ouvrantes et fermantes, du type **<balise>** ... **</balise>**. Ces balises peuvent contenir des *attributs* fournissant une description de l'élément textuel concerné, et qui se placent sur la balise ouvrante: **<balise attribut=valeur>** ... **</balise>**. SGML permet d'associer à chaque type de texte une Définition de Type de Document (DTD), qui précise les balises autorisées et les agencements légaux de ces balises.

1.2. Le cas des dictionnaires

Les dictionnaires figurent parmi les types de textes les plus complexes traités par la TEI. Chaque entrée d'un dictionnaire est un objet fortement structuré, dans lequel de nombreux mécanismes d'abréviation et d'organisation typographique permettent une présentation condensée des informations. De plus, la structure des entrées varie considérablement d'un dictionnaire à l'autre et dans un même dictionnaire: il semble presque que l'on puisse trouver n'importe quel type d'information à n'importe quelle position d'une entrée dans un dictionnaire ou un autre. Toutefois, malgré ces variations, les lecteurs humains sont capables d'interpréter relativement aisément les entrées de dictionnaire, et ce, le plus souvent, sans consulter les explications introductives. Il est donc clair qu'il existe un certain nombre de principes et de régularités sous-jacentes, qu'une norme d'encodage se doit de saisir. La première difficulté à laquelle a été confronté le groupe de travail de la TEI sur les dictionnaires[1] a donc été la définition d'un schéma d'encodage suffisamment général pour couvrir la plupart des dictionnaires, tout en permettant de décrire les particularités de chacun. Ce conflit entre *généralité* et *pouvoir descriptif* existe pour de nombreux types de textes, mais il semble atteindre son point culminant dans le cas des dictionnaires.

Un deuxième type de problème d'encodage provient du fait que les dictionnaires, contrairement à la plupart des autres types de textes, sont à la fois des *textes* et des *bases de données*[2]. Les dictionnaires ont bien évidemment l'apparence de textes et possèdent de nombreuses caractéristiques communes à tous les types de textes. Néanmoins, les utilisateurs ne lisent pas, en principe, les dictionnaires de manière linéaire de A à Z comme ils le font pour la plupart des textes, mais accèdent à des *entrées* à partir d'une *clé* (la vedette) dans le but de récupérer divers champs d'information associés à cette clé (prononciation, information grammaticale, étymologie, définitions, etc.). Cet accès non linéaire est typique de l'accès aux bases de données. Il est encore plus clair avec les dictionnaires électroniques, qui offrent d'autres modes d'accès: l'utilisateur peut accéder à tous les mots dont la définition contient un mot donné, à tous les mots remplissant un certain nombre de critères (par exemple, tous les verbes relevant du domaine nautique, apparaissant avant 1900), etc. En outre, si l'affichage sur l'écran ressemble toujours plus ou moins à du texte,[3] la représentation interne est rarement celle d'un texte linéaire.

Les dictionnaires présentent donc une forte dualité entre leur *structure de surface* (le texte) et leur *structure profonde* (le contenu informationnel). Une grande partie des informations de la structure profonde n'est pas explicite

dans la structure de surface et nécessite la connaissance des conventions d'abréviation et de présentation des dictionnaires. Par exemple, dans l'entrée qui figure ci-dessous, la structure de surface -- c'est-à-dire la position linéaire des divers éléments -- ne dit pas explicitement que "nom" (*n.*) ne s'applique qu'aux sens 1 et 2, alors que la prononciation s'applique aux six sens.[4]

> **roughcast** (ˈrʌfˌcɑːst) *n.* **1.** a coarse plaster used to cover the surface of an external wall. **2.** any rough or preliminary form, model, etc. *~adj.* **3.** covered with or denoting roughcast. *~vb.* **-casts, -casting, -cast. 4.** to apply roughcast to (a wall, etc.). **5.** to prepare in rough. **6.** (*tr.*) another word for **rough-hew. --ˈroughˌcaster** *n.* [CED]

La dualité structurelle des dictionnaires est source de difficultés d'encodage par le conflit qu'elle entraîne entre deux *vues* différentes du dictionnaire. Un utilisateur donné peut préférer l'encodage d'un point de vue textuel qui conserve la structure de surface (afin, par exemple, de rester fidèle à une version imprimée pré-existante). Cependant, le type d'inférence nécessaire à la récupération de la structure informationnelle profonde à partir de la structure de surface peut être difficile, voire impossible, pour un ordinateur.[5] Si un utilisateur s'intéresse à la vue "base de données" (par exemple afin de visualiser et manipuler le dictionnaire à l'aide d'outils informatiques), il aura besoin d'un encodage explicite des informations qui ne sont qu'implicites dans la structure de surface. Dans certains cas, les utilisateurs souhaiteraient même avoir accès aux deux vues simultanément. Etant donné que les deux vues du dictionnaire sont souvent en conflit, leurs codages peuvent être très différents. Un deuxième défi important pour le groupe de travail de la TEI sur les dictionnaires était de permettre l'encodage des deux vues, soit indépendamment, soit simultanément.

Le présent article est centré sur les deux principaux problèmes que nous venons d'évoquer: d'une part le conflit entre généralité et pouvoir descriptif des schémas d'encodage, et, d'autre part, le conflit entre les vues "texte" et "base de données". Un certain nombre d'autres problèmes relatifs à l'encodage de dictionnaires ne seront pas traités ici, et le lecteur est invité à consulter le chapitre 12 des *Guidelines* (pp. 321-70) pour une description détaillée des conventions d'encodage des dictionnaires proposées par la TEI.

2. Principes généraux

La tâche du groupe de travail sur les dictionnaires était de fournir un ensemble de conventions d'enodage des entrées de dictionnaires, la structuration de niveau supérieur (page de titre, matériau introductif, divisions en noms communs et en noms propres, en langues dans les dictionnaires bilingues, etc.) étant de même nature que dans bien d'autres types de textes.[6] Le groupe de travail a, par ailleurs, limité son champ aux dictionnaires occidentaux modernes, et a testé ses recommendations principalement sur des dictionnaires de taille moyenne, tels que le *PL*, le *PR* ou le *CED*. Les dictionnaires anciens et les dictionnaires "monumentaux" tels que l'*OED* ou le *TLF* ont été volontairement laissés de côté pour la première édition des *Guidelines*.

2.1. Composants de base

De nombreux types d'informations clairement identifiables figurent dans les entrées de dictionnaires: informations sur la forme du mot (orthographe, prononciation, césure, etc.), informations grammaticales (catégorie grammaticale, sous-catégorie, morphologie, etc.), définitions ou traductions, étymologie, renvois, sous-entrées, notes d'usage, exemples, etc.

La première étape dans la réalisation d'une Définition du Type de Document (DTD) SGML pour les dictionnaires est la spécification d'une typologie des éléments *atomiques* qui figurent dans les entrées, accompagnée d'une nomenclature adéquate pour ces éléments. Les éléments atomiques sont ceux qui constituent les champs de base spécifiques aux entrées de dictionnaire. Ces éléments ne contiennent aucun autre champ d'information: leur contenu est une séquence de caractères, éventuellement accompagnée d'éléments communs à tous les types de textes (dates, etc.). L'identification des champs fondamentaux d'information dans les dictionnaires avait reçu l'attention de nombreux chercheurs dans le passé et malgré des désaccords sur les détails, ces champs d'information étaient relativement bien établis avant le travail de la TEI (voir par exemple DANLEX, 1987, AMSLER/TOMPA, 1988).

Certains éléments de dictionnaires sont complexes, c'est-à-dire constitués de groupes d'éléments atomiques. Considérons, par exemple, la définition suivante:

CRAWLER [krole] v.i. Nager le crawl. [PL]

Cette entrée comporte trois parties distinctes: les informations relatives aux formes écrite et parlée de la vedette, les informations grammaticales, et la définition. Dans de nombreux cas, il convient de rendre explicites ces associations ou regroupements; à cette fin, nous avons défini un ensemble de *balises groupantes* permettant le marquage de relations logiques entre éléments. Ainsi, l'encodage de l'entrée ci-dessus serait[7]

```
<entry>
  <form>
    <orth>crawler</orth>
    <pron>krole</pron>
  </form>
  <gramGrp>
    <pos>v</pos>
    <subc>i</subc>
  </gramGrp>
  <def>Nager le crawl</def>
</entry>
```

La première information comporte deux sous-parties, marquées par les balises **<orth>** et **<pron>**; la balise **<form>** assure leur association logique. De la même manière, le composant **<gramGrp>** comporte deux sous-composants, la catégorie grammaticale (**<pos>** pour "part-of-speech") et les informations de sous-catégorisation (**<subc>**). La définition est un composant atomique, constitué du seul texte de définition, sans structure interne.

Outre l'association d'éléments, les balises groupantes servent à restreindre (par le biais de leurs définitions dans la DTD) les balises qu'elles peuvent contenir, permettant ainsi une définition plus étroite de la structure d'entrée autorisée. Par exemple, l'élément **<form>** est défini de manière à contenir **<orth>**, **<pron>**, **<hyph>**, **<syll>**, **<usg>**, **<lbl>** ou un autre **<form>**. Il peut également contenir, dans n'importe quelle position, des séquences de caractères ou d'autres éléments de base des paragraphes (c'est-à-dire, les éléments définis par le modèle de contenu *paraContent* dans TEI P3, chapitre 3, p. 68), afin de permettre l'inclusion éventuelle de texte libre entre les éléments. Le fragment de DTD qui définit **<form>** est donc:

```
<!ELEMENT form - - (orth|pron|hyph|syll|usg|lbl|form
                   |%paraContent)+                    >
```

2.2. Structure hiérarchique et portée

D'une façon quasi-systématique, les entrées de dictionnaires sont structurées de façon hiérarchique: une entrée comporte souvent deux ou plusieurs sous-parties, chacune correspondant à des homographes grammaticaux, qui peuvent se subdiviser à nouveau en sens et sous-sens (figure 1). L'entrée *roughcast* donnée précédemment en est une bonne illustration: elle comporte trois homographes grammaticaux (nom, adjectif, verbe), eux-mêmes subdivisés en plusieurs sens.

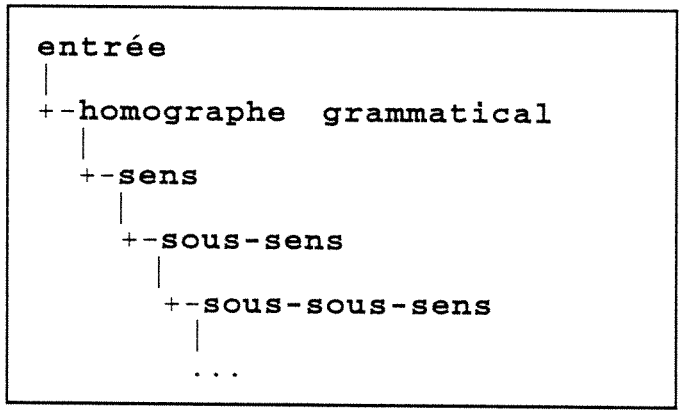

Figure 1. Divisions et sous-divisions des entrées de dictionnaires.

Les hiérarchies peuvent être très profondes dans certains dictionnaires, comme le montre l'entrée *valeur* ci-dessous. Dans certains cas, un ou plusieurs niveaux peuvent être absents (par exemple, le niveau des homographes grammaticaux).

> **valeur** [valœʀ] n. f. **A. I. 1.** Ce par quoi une personne est digne d'estime, ensemble des qualités qui la recommandent. (V. mérite). *Avoir conscience de sa valeur. C'est un homme de grande valeur.* **2.** Vx. Vaillance, bravoure (spécial., au combat). *"La valeur n'attend pas le nombre des années"* (Corneille). ◊ *Valeur militaire (croix de la):* décoration française...
> ...
> **II. 1.** Ce en quoi une chose est digne d'intérêt. *Les souvenirs attachés à cet objet font pour moi sa valeur.* **2.** Caractère de ce qui est reconnu digne d'intérêt...
> ...
> **B. I. 1.** Caractère mesurable d'un objet, en tant qu'il est susceptible d'être échangé, désiré, vendu, etc. (V. prix). *Faire estimer la valeur d'un objet d'art...*
> [DNT]

L'organisation hiérarchique des dictionnaires permet la *factorisation* des

informations sur certains niveaux de la hiérarchie. Les informations ont donc une *portée,* comme les variables d'un langage informatique structuré en blocs tel que Pascal: les informations précisées à un niveau donné de la hiérarchie s'appliquent à tous les niveaux emboîtés. Dans les dictionnaires, les informations relatives à la prononciation, à la forme orthographique, à la catégorie grammaticale, etc. sont généralement mises en facteur à la tête de l'entrée car elles s'appliquent aux différents sens. Par exemple, dans l'entrée *roughcast* citée plus haut, l'orthographe et la prononciation s'appliquent à l'entrée entière, "nom" s'applique aux trois premiers sens, etc. (figure 2).

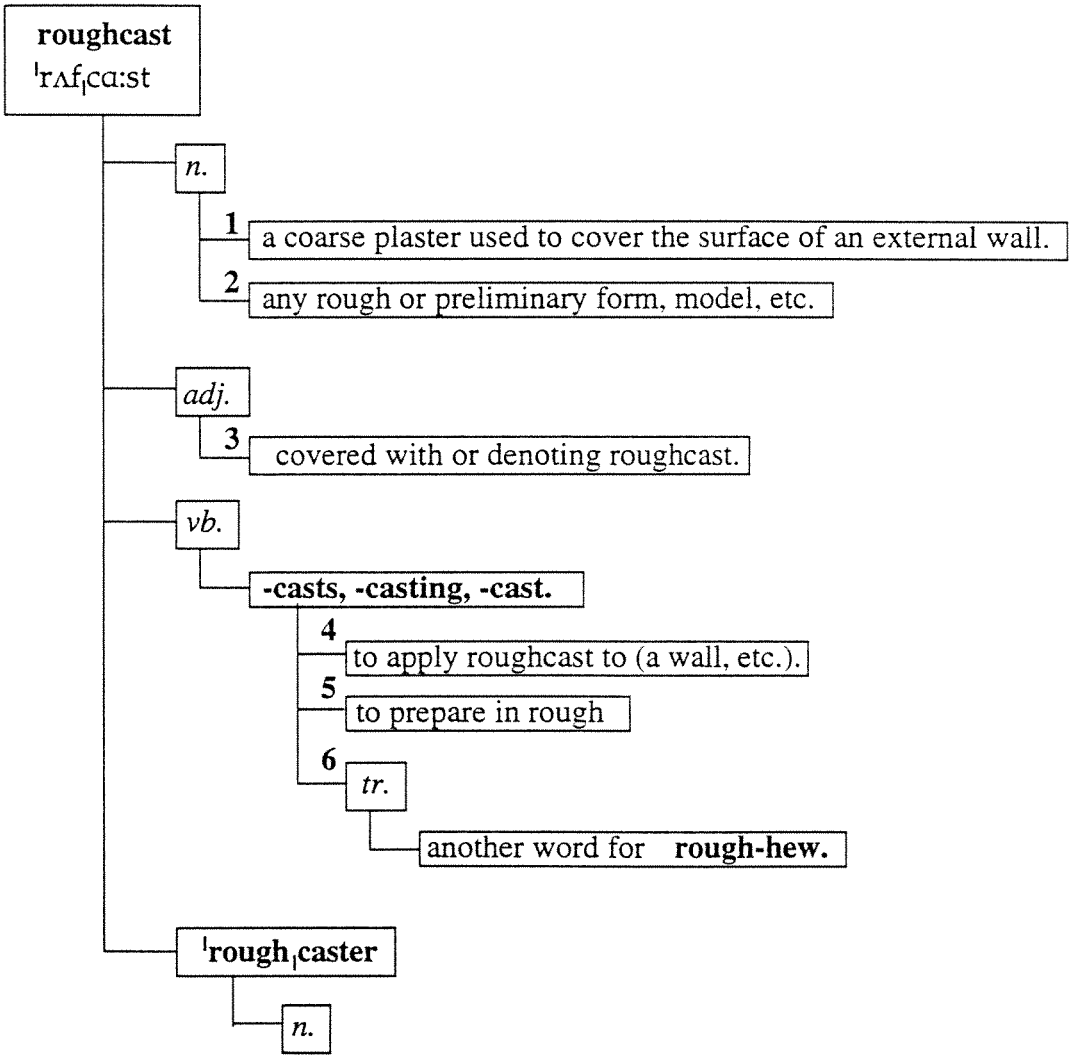

Figure 2. Factorisation et portée.

Les entrées courantes auront généralement des structures telles que les suivantes:

```
<!-- entrée comportant deux sens -->

<entry>
  <form>...</form>
  <gramGrp>...</gramGrp>
  <sense n='1'>...</sense>
  <sense n='2'>...</sense>
</entry>

<!-- entrée comportant deux homographes grammaticaux -->
<!-- comportant chacun deux sens                      -->

<entry>
  <form>...</form>
  <hom n='I'>
    <gramGrp>...</gramGrp>
    <sense n='1'>...</sense>
    <sense n='2'>...</sense>
  </hom>
  <hom n='II'>
    <gramGrp>...</gramGrp>
    <sense n='1'>...</sense>
    <sense n='2'>...</sense>
  </hom>
</entry>
```

3. Traitement de la variation

Il serait assez simple d'écrire une DTD qui décrive la structure d'un dictionnaire sur la base des principes structuraux esquissés ci-dessus. Une telle DTD permettrait d'emboîter les homographes à l'intérieur des entrées, les sens à l'intérieur des homographes, les sous-sens à l'intérieur des sens, etc. En outre, les composants factorisés seraient autorisés aux niveaux appropriés de la hiérarchie. Ainsi, par exemple, **<entry>** serait défini de manière à contenir **<form>** et un ou plusieurs homographes (**<hom>**), **<hom>** serait défini pour contenir **<gramGrp>** et un ou plusieurs **<sens>**, etc.

Malheureusement, la situation n'est pas aussi simple. La structure d'un dictionnaire est de loin plus complexe et plus variable que ne le suggère ce simple schéma, et les sections ci-après donnent un bref aperçu de certains des problèmes rencontrés par le groupe de travail de la TEI dans le développement d'une DTD suffisamment générale pour s'appliquer à une majorité de dictionnaires, tout en offrant une description suffisamment précise de leur structure.

3.1. Variation entre dictionnaires

Bien que les principes d'organisation hiérarchique et de factorisation des informations soient une constante sous-tendant la structure de la quasi-totalité des dictionnaires occidentaux modernes, la marge de variation entre dictionnaires est très importante et rend très difficile la recherche d'une description structurale universelle. Par exemple, les informations étymologiques apparaissent à des endroits différents selon les dictionnaires, comme on peut le voir dans les entrées suivantes:

> **nougat** (ˈnu:gɑ:, ˈnʌgət) *n.* a hard chewy pink or white sweet containing chopped nuts, cherries, etc. [C19: via French from Provençal *nogat,* from *noga* nut, from Latin *nux* nut] [CED]

> **NOUGAT** n.m. (mot prov.). Confiserie de sucre, de miel et de blancs d'oeufs frais ou desséchés, additionnée d'amandes, de noisettes ou encore de pistaches. [PL]

Dans le *CED,* l'étymologie se trouve toujours à la fin de l'entrée, tandis que dans le *PL,* elle se situe toujours au début, après les informations grammaticales. Voici des exemples de fragments de deux DTD qui pourraient rendre compte de ces structures:

```
<!-- fragment de DTD pour le CED (pas TEI)        -->

<!ELEMENT entry - - (form, gramGrp, ..., etym?)     >


<!-- fragment de DTD pour le PL (pas TEI)         -->

<!ELEMENT entry - - (form, gramGrp, etym?, ...)     >
```

Cependant, puisque le but du groupe de travail était de définir une DTD unique applicable à tout dictionnaire, il fallait en théorie permettre *toutes* les variantes possible dans la DTD. Ainsi, pour le cas relativement simple des deux variantes précédentes, il faudrait une définition du type:

```
<!-- fragment de DTD pour le CED et le PL         -->
<!-- (pas TEI)                                    -->

<!ELEMENT entry - - (form, gramGrp,
                    ((etym?, ...)|(..., etym?))    >
```

Cette DTD fusionnée est plus générale mais elle est aussi *surgénératrice*

pour chacun des deux dictionnaires pris isolément. Par exemple, si cette DTD est utilisée pour valider la structure du *PL*, elle autorisera l'apparition d'une étymologie à la fin aussi bien qu'au début de l'entrée, permettant ainsi des accidents et des erreurs.

Cet exemple n'est qu'une illustration simple des types de variation existant entre les structures des dictionnaires. Les étymologies peuvent se trouver dans d'autres endroits encore dans d'autres dictionnaires (voir par exemple l'entrée *nougat* du *PR* dans la section suivante), et le même type de problème existe pour presque tous les composants des entrées. Une DTD qui soit assez souple pour permettre toutes les variantes éventuelles doit donc permettre l'apparition de tout composant en n'importe quelle position. Ainsi, la définition d'une entrée dans la DTD de la TEI est:

```
<!-- Fragment de la DTD de la TEI                    -->

<!ELEMENT entry - - (hom|sense|
                     form|gramGrp|usg|def|etym|eg...)+>
```

Cette définition permet l'emboîtement des balises hiérarchiques de type **<hom>** et **<sense>** à l'intérieur de ceux de type **<entry>**, aussi bien que l'apparition de tout composant de l'entrée, dans n'importe quel ordre et en nombre quelconque. La définition permet donc la description de nombreux dictionnaires, mais permet aussi, en contrepartie, de nombreuses structures qui n'apparaissent probablement pas si l'on considère un dictionnaire donné.

3.2. Variation à l'intérieur d'un dictionnaire

Le problème est aggravé par la grande variabilité de forme des entrées même à l'intérieur d'un dictionnaire donné. En particulier, la plupart des composants de base peuvent apparaître à tout niveau de la hiérarchie. Ainsi, dans l'entrée ci-dessous, la prononciation, qui figure généralement au plus haut niveau et est factorisée sur toute l'entrée, apparait plus bas dans la hiérarchie, au niveau des homographes grammaticaux:

> **overdress** *vb.* (ˌəʊvəˈdrɛs) **1.** to dress (oneself or another) too elaborately or finely. *~n.* (ˈəʊvəˌdrɛs) **2.** a dress that may be worn over a jumper, blouse, etc. [CED]

Il existe en outre un processus complexe de *surcharge* des informations dans la hiérarchie: les dictionnaires donnent fréquemment des informations pour un sens particulier qui prennent le pas et remplacent les informations mises en

facteur à un niveau supérieur. Par exemple:

- La prononciation apparaît au niveau des sens dans le troisième sens du mot *conjure* dans le *CP*, parce qu'il a une prononciation exceptionnelle, différente de celle des autres sens dans l'entrée:

  conjure (ˈkʌndʒə) *vb* **1.** to practice conjuring. **2.** to summon (a spirit or demon) by magic. **3.** (kənˈdʒʊə) to appeal earnestly to... [CP]

- On voit dans l'entrée *heave* du *CED* que la flexion peut être différente pour un sens particulier:

- Parfois, le *PR* donne des informations étymologiques différentes pour un sens particulier:

  NOUGAT [nuga] n.m. - 1750; *nogas* plur. 1595; provenç. *nougo* "noix", d'un lat. pop. *nuca*, class. *nux* "noix" **1.** Confiserie fabriquée avec des amandes (ou des noix, des noisettes) et du sucre caramélisé, du miel. ... **2.** (1928) FIG ET FAM *C'est du nougat !* c'est très facile. ... **3.** (1926; *jambes en nougat* "fatiguées, molles" 1917) POP *Les nougats* : les pieds. ... [PR]

Les variations de structure proviennent non seulement de la complexité du contenu de l'entrée, mais également d'éventuelles modifications dans la politique éditoriale. Ceci est particulièrement vrai pour les grands dictionnaires tels que le *OED* ou le *TLF* qui ont été réalisés sur plusieurs décennies par des équipes de lexicographes de composition changeante.[8]

La variabilité intra-dictionnaire de la structure de l'entrée nécessite une généralité encore plus grande dans la DTD du dictionnaire puisque tous les niveaux hiérarchiques (entrée, homographe, sens, sous-sens, etc.) peuvent en théorie contenir les mêmes éléments. En termes de DTD, ceci veut dire que les balises marquant les niveaux dans la hiérarchie (**<entry>**, **<hom>**, **<sense>**) doivent avoir à peu près le même contenu:

```
<!-- Fragment de la DTD de la TEI                    -->

<!ELEMENT entry - - (hom|sense|
                     form|gramGrp|usg|def|etym|eg...)+ >
<!ELEMENT hom   - - (sense|
                     form|gramGrp|usg|def|etym|eg...)+ >
<!ELEMENT sense - - (sense|
                     form|gramGrp|usg|def|etym|eg...)+ >
```

Il est à noter que l'élément **<sense>** est défini de façon récursive, afin de permettre l'emboîtement des sous-sens à n'importe quelle profondeur.

3.3. Exceptions

Bien que le fragment de DTD du dictionnaire décrit dans la section précédente soit très général et permette diverses structures pour une entrée, il impose tout de même certaines contraintes de régularité qui risquent d'être transgressées dans certains dictionnaires. Par exemple, dans l'entrée suivante, il est nécessaire d'inclure un élément **<pron>** à l'intérieur d'un **<def>**, ce qui n'est pas permis par la DTD:

> **demi•god** /ˈdemɪgɒd/ *n* one who is partly divine and partly human; (in Gr myth, etc) the son of a god and a mortal woman, eg *Hercules* /ˈhɜːkjʊliːz/.
>
> [OALD]

Les exceptions de ce type sont assez imprédictibles et entraînent de grosses difficultés pour l'écriture d'une DTD. On pourrait imaginer la relaxation de toutes les contraintes structurelles dans une DTD qui permettrait tous les agencements possibles d'éléments dans les entrées, mais une telle DTD serait informativement vide et, par conséquent, de peu d'utilité pour la validation de l'encodage, la recherche d'information, ou la génération de la typographie complexe des entrées.

La TEI a adopté un compromis dans lequel l'élément **<entry>** permet de représenter la grande majorité des entrées de dictionnaires, et un élément alternatif **<entryFree>** permet de représenter les entrées atypiques. Cet élément **<entryFree>**, qui ne devrait être utilisé qu'en dernier recours, utilise les mêmes composants qu'**<entry>** mais permet de les combiner de façon totalement libre.

Cette solution n'est pas complètement satisfaisante car, dans de nombreux cas, les entrées divergentes ne présentent une structure atypique que pour un seul sous-sens ou une seule information placés à un endroit inhabituel (comme c'est le cas dans l'exemple ci-dessus). La solution adoptée impose dans tous les cas la relaxation des contraintes structurelles sur l'ensemble de l'entrée. On se heurte toutefois ici à une limitation de SGML qui ne fournit pas de mécanisme permettant de représenter des irrégularités locales dans un document.

4. Encodage de vues multiples

4.1. Vues des dictionnaires

Comme nous l'avons mentionné dans l'introduction, il existe au moins deux *vues* différentes des dictionnaires:

- une *vue textuelle*, correspondant à la "structure de surface" du dictionnaire, c'est-à-dire la suite linéaire de caractères qui constitue le texte d'origine, ainsi que son rendu physique et typographique,
- une *vue "base de données"*, correspondant à la "structure profonde", c'est-à-dire le contenu informationnel des entrées, indépendamment de sa présentation exacte.

Pour prendre un exemple concret, la forme particulière sous laquelle figure un nom de domaine dans un dictionnaire donné (par exemple, *nautical, naut., Naut.,* etc.) serait préservée dans une vue textuelle, alors qu'elle serait normalisée dans une vue "base de données" (par exemple en *nautical*) quelle que soit la forme sous laquelle elle apparaît.

Deux processus symétriques relient ces vues (figure 3):

- le processus de *publication,* qui consiste à générer une représentation textuelle des entrées à partir d'une base de données sous-jacente, en vue d'une impression papier ou d'une visualisation à l'écran;
- le processus de *retro-conversion,* qui consiste à générer une base de données à partir du texte de dictionnaires non informatisés, obtenu sous forme de bande de photocomposition ou par saisie optique ou clavier.

Le processus de publication fait intervenir des choix éditoriaux pour un dictionnaire particulier, tels que l'utilisation de l'abréviation *naut.* pour *nautical,* ou certains styles d'impression particuliers). Cette "traduction" peut être automatisée.

Le processus de rétro-conversion implique la traduction des caractéristiques typographiques du texte (italique, gras, etc.) en identificateurs de champs logiques. Malheureusement, les ambiguïtés multiples, ainsi que les inconsistances des dictionnaires, rendent ce processus délicat et difficilement automatisable.

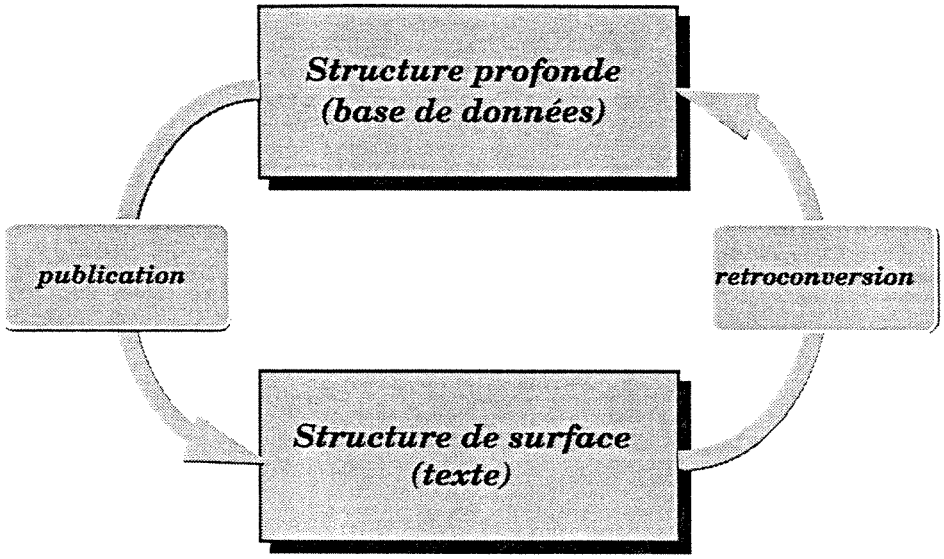

Figure 3. Relations entre vues

Les utilisateurs de dictionnaires informatisés s'intéressent principalement à la récupération des informations contenues dans le dictionnaire. Même si les résultats de requêtes sont présentés sous forme textuelle, les utilisateurs sont donc concernés principalement par la vue "base de données". Par exemple, considérons l'entrée suivante:

thyr(é)ostimuline [tiʀ(e)ostimylin] ... [DNT]

L'utilisateur qui recherche les informations associées à cette entrée utilisera comme clé *thyrostimuline* ou sa variante, *thyréostimuline*, mais probablement pas *thyr(é)ostimuline*.

De même, les éditeurs de dictionnaires, dans la mesure où ils peuvent générer automatiquement le texte, sur papier ou sur écran, à partir de la base de données sous-jacente, n'ont, en principe, besoin d'encoder et de maintenir que cette dernière.

Le groupe de travail de la TEI a toutefois été amené à proposer des mécanismes d'encodage aussi bien pour la vue textuelle que pour la vue "base de données", et a été motivé en cela par deux raisons. Tout d'abord, certains chercheurs, philologues ou historiens, peuvent désirer accéder aux deux vues simultanément, par exemple pour essayer de retracer l'histoire éditoriale d'un dictionnaire. Ainsi, la présence de variantes telles que *"Naut."*, *"nautical"*, etc. pour le domaine *nautique* peut être le signe de "couches" éditoriales distinctes. De même, certains aspects socio-historiques peuvent être reflétés dans la mise en page et la typographie du texte: le *CED*, dans sa deuxième édition (1986),

n'hésite pas à utiliser le mot "*fuck*" comme rappel de la vedette courante en haut de la page 611. En d'autres temps, ou dans d'autres cultures, cet "accident" de mise en page aurait été soigneusement évité.

Par ailleurs, le processus de rétro-conversion mentionné plus haut est éminemment difficile et sujet à erreurs, et il est plus facile, dans un premier temps, de viser l'encodage d'une vue textuelle, très proche de la forme imprimée, que l'on peut analyser dans un deuxième temps, pour obtenir une vue "base de données". Il est d'ailleurs souvent souhaitable de préserver la vue textuelle comme référence, pour le cas où certaines informations seraient perdues ou mal interprétées pendant le déroulement de la deuxième phase de la rétro-conversion.

4.2. Encodage de la vue textuelle

A première vue, l'encodage de la vue textuelle d'un dictionnaire paraît relativement simple: il suffit d'encoder strictement la suite des caractères qui composent les entrées, en identifiant de façon séquentielle les champs d'informations (prononciation, etc.), dans l'ordre où ils se présentent. L'attribut *rend,* qui peut apparaître sur toutes les balises de la TEI permet, le cas échéant, de coder l'information concernant le rendu typographique.

Un principe important dans l'encodage de la vue textuelle est le *principe de restitution*: il devrait être possible, à partir de cette vue, de restituer automatiquement le texte original. Cette possibilité de restituer le texte source à partir de la version encodée est particulièrement importante dans le cas de la rétro-conversion, car elle permet de s'assurer de la fidélité du processus. Il existe un certain nombre de façons différentes de définir ce qui est à restituer (par exemple, un fac-similé d'une version imprimée donnée, sa présentation, sa typographie, etc.). Cependant, pour de nombreux objectifs (comparaison et validation entre texte d'origine et texte encodé, opérations telles que comptage des mots, recherche, réalisation d'une concordance, analyse linguistique, etc.), il suffit de restituer la suite de caractères constituant le texte, indépendamment de sa représentation typographique.

Des conseils simples pour l'encodage de la vue textuelle pourraient donc être les suivants:

1. Aucun des caractères de la séquence initiale ne doit être supprimé ou modifié.
2. Les données d'origine ne doivent pas être précisées sous forme

d'attributs SGML, mais doivent toujours apparaître comme contenu de balises.

3. A part les données d'origine, aucune autre donnée ne doit apparaître en tant que contenu de balise.
4. L'ordre original des données ne doit pas être modifié.

Par exemple, l'entrée

pinna (ˈpɪnə) *n., pl.* **-nae** (-ni:) *or* **-nas**... [CED]

pourrait être encodée comme suit:

```
<!-- vue textuelle stricte                  -->
<!-- avec encodage du rendu typographique -->

<entry>
  <form>
    <orth rend=bold.14>pinna</orth>
    <pron rend=ipa>('pIn@)</pron>
  </form>
  <gramGrp rend=ital><pos>n.</pos>, </gramGrp>
  <form type=inflected rend=ital>
    <num>pl.</num>
    <form>
      <orth rend=bold>-nae</orth>
      <pron rend=ipa>(-ni:)</pron>
    </form>
    or
    <orth>-nas</orth>
  </form>
...
```

Toutefois, le codage ci-dessus est relativement lourd, et dans la mesure où la plupart des dictionnaires ont des conventions typographiques systématiques (vedette en gras, etc.), il n'est pas nécessaire de coder ces informations de façon redondante dans chaque entrée. Elle peuvent être rappelées une fois pour toutes dans l'en-tête du document ("TEI header", cf. TEI P3, chapitre 5, pp. 89-137). Seuls les accidents et exceptions aux conventions par défaut doivent alors être codés de façon explicite. On aura alors un codage du type suivant:

```
<!-- vue textuelle stricte                -->
<!-- rendu typographique implicite        -->

<entry>
  <form>
    <orth>pinna</orth>
    <pron>('pIn@)</pron>
  </form>
  <gramGrp><pos>n.</pos>, </gramGrp>
  <form type=inflected>
    <num>pl.</num>
    <form>
      <orth>-nae</orth>
      <pron>(-ni:)</pron>
    </form>
    or
    <orth>-nas</orth>
  </form>
...
```

A part un certain nombre de conventions typographiques, les dictionnaires utilisent aussi un méta-texte, c'est-à-dire un ensemble de caractères ou d'éléments phrastiques qui n'ont d'autre rôle que d'identifier ou de séparer les champs d'information proprement dits. Ainsi, dans l'entrée *pinna* ci-dessus, les parenthèses autour de la prononciation ne font pas partie de la prononciation elle-même. De même, la virgule qui sépare l'information grammaticale de la vedette ("*n.*") des formes fléchies, ou bien le "*or*" qui sépare les deux formes pluriels possibles, sont des éléments de méta-texte. Ces éléments, que nous appellerons *caractères de rendu* ou *texte de rendu,* sont généralement arbitraires, bien que systématiques pour un dictionnaire donné. On pourrait imaginer une édition différente du *CED,* dont la présentation serait (par exemple):

PINNA /ˈpɪnə/ n. [pl. -**nae** (-ni:), -**nas**] ... [CED]

Dans la mesure ou le texte de rendu est restituable de façon systématique, il n'est pas indispensable de le coder de façon redondante pour chaque entrée, et, à nouveau, les indications pour le retrouver peuvent être consignées dans l'en-tête. Un encodage moins strict de la vue textuelle pourrait ignorer le texte de rendu, qui est automatiquement restituable (par exemple, les parenthèses qui entourent toujours la prononciation dans un dictionnaire donné). Dans ce cas, la suppression des balises devrait reproduire exactement la suite de caractères originale, moins le texte de rendu. Il faudrait alors documenter les conventions de rendu dans l'en-tête du document contenant le dictionnaire encodé, par exemple dans le cas ci-dessus:

- parenthèses autour de la prononciation,
- virgule avant les formes fléchies,
- conjonction *or* entre les formes fléchies,
- point après les informations relatives à la catégorie grammaticale et aux formes fléchies.

Puisque ces éléments sont restituables par un algorithme simple, on peut encoder l'entrée comme suit:

```
<!-- vue textuelle                          -->
<!-- texte de rendu implicite               -->

<entry>
  <form>
    <orth>pinna</orth>
    <pron>'pIn@</pron>
  </form>
  <gram>
    <pos>n</pos>
  </gram>
  <form type=inflected>
    <num>pl</num>
    <form>
      <orth>-nae</orth>
      <pron>-ni:</pron>
    </form>
    <orth>-nas</orth>
  </form>
...
```

4.3. Encodage de la vue "base de données"

L'encodage en vue "base de données" peut impliquer la modification des données d'origine de diverses façons, comme par exemple,

- la normalisation de *nautical, naut., Naut.,* etc., en *nautical*;
- l'extension de *delay, -ed, -ing* en *delay, delayed, delaying*;
- l'extension de *thyr(é)ostimuline* [*tiR(e)ostimylin*] en *thyrostimuline* [*tiRostimylin*] et *thyréostimuline* [*tiReostimylin*];
- l'ajout de la personne, le temps et le nombre pour chacune des formes *sings, singing, sang, sung*;
- la réorganisation de l'ordre des éléments dans une entrée afin de mettre

en évidence leurs liens, comme dans:

> **clem** (klɜm) or **clam** *vb.* **clems, clemming, clemmed** or **clams, clammimg, clammed** ...
>
> [CED]

(où l'on voudra regrouper *clem* et *clam* avec leur formes fléchies respectives);

- la division d'une entrée en deux entrées séparées, comme dans:

> **celi•bacy** /ˈselɪbəsɪ/ *n* [U] state of living unmarried, esp as a religious obligation. **celi•bate** /ˈselɪbət/ *n* [C] unmarried person (esp a priest who has taken a vow not to marry). [OALD]

L'exemple *pinna* donné ci-dessus pourrait être encodé de la manière suivante dans une vue "base de données":

```
<!-- vue base de données:          -->

<entry>
  <form>
    <orth>pinna</orth>
    <pron>'pIn@</pron>
    <form type=inflected>
      <num>pl</num>
      <form>
        <orth type=lat>pinnae</orth>
        <pron>'pIni:</pron>
      </form>
      <orth type=std>pinnas</orth>
    </form>
  </form>
  <gramGrp>
    <pos>n</pos>
  </gramGrp>
...
```

On voit les différences entre cet encodage de l'entrée et l'encodage de la vue textuelle donné dans la section précédente. En particulier, les différentes formes de la vedette sont regroupées et les formes complètes des formes fléchies sont précisées. Ces modifications rendent les données plus conformes à ce qui pourrait apparaître dans une base de données structurée, où toutes les formes apparaîtraient dans un ensemble de sous-champs, et les variantes seraient représentées dans leurs formes complètes, etc. Tout ceci simplifie par exemple les opérations d'interrogation, en facilitant la recherche de toutes les formes variantes d'une forme donnée.

Les modifications telles que celles qui sont souvent demandées pour la

vue "base de données" peuvent rendre impossible la restitution de la suite exacte de caractères de l'original imprimé, si ce dernier existe.

4.4. Encodage simultané des deux vues

Comme nous l'avons mentionné plus haut, il est parfois nécessaire d'avoir accès aux deux vues des données. La solution préférée par la TEI consiste à encoder séparément les deux vues, dans des documents SGML distincts, et à les mettre en correspondance, le cas échéant par des mécanismes d'alignement (voir TEI P3, chapitre 14, "Linking, Segmentation and Alignment," p. 393).

Toutefois, dans certains cas, les vues "base de données" et "texte" d'un dictionnaire ne diffèrent que par un petit nombre d'entrées ou de parties d'entrées, et il n'est guère économique de les encoder dans deux documents différents. Nous avons donc mis au point un certain nombre de mécanismes permettant d'encoder simultanément deux vues des données des dictionnaires dans un même document, à l'aide d'attributs SGML.

Deux principes généraux régissent l'encodage simultané des vues "base de données" et "texte":

> **Principe 1** : Choisir une vue dominante, soit "texte" soit "base de données" et encoder la vue dominante dans le contenu des balises et la vue non dominante dans des attributs SGML.

Par exemple, si l'on souhaite développer "*delay, -ed, -ing*" en "*delay, delayed, delaying*", l'encodage en vue textuelle dominante serait:

```
<form>
  <orth>delay</orth>
  <form type=inflected>
    <orth norm='delayed'>-ed</orth>
  </form>
  <form type=inflected>
    <orth norm='delaying'>-ing</orth>
  </form>
</form>
```

Les formes développées sont précisées dans l'attribut *norm* sur les balises **<orth>** appropriés.

Un encodage des mêmes informations en vue dominante "base de données" serait:

```
<form>
  <orth>delay</orth>
  <form type=inflected>
    <orth orig='-ed'>delayed</orth>
  </form>
  <form type=inflected>
    <orth orig='-ing'>delaying</orth>
  </form>
</form>
```

Ici, l'attribut *orig* est utilisé pour préciser la forme imprimée d'origine des informations qui apparaissent sous forme développée comme contenu de balise.

Des attributs supplémentaires (*split, mergedin, opt*) permettent de saisir d'autres types de divergences entre les vues "texte" et "base de données" (voir TEI P3, p. 365).

Le second principe concerne les réarrangements d'éléments entre les deux vues:

> **Principe 2**: En cas de conflit dans l'ordre des éléments entre les deux vues, utiliser les mécanismes d'alignement de la TEI pour mettre en évidence la correspondance entre les deux encodages (voir TEI P3, section 14.4).

Par exemple, on peut utiliser la balise **<anchor>** et l'attribut de localisation (TEI P3, section 14.3) pour associer la position d'origine et l'élément déplacé, comme dans l'exemple suivant:

```
<entry>
  <form>
    <orth>pinna</orth>
    <pron>'pIn@</pron>
    <anchor id=p1>
    <form type=inflected>
      <num>pl</num>
      <form>
        <orth type=lat>pinnae</orth>
        <pron>'pIni:</pron>
      </form>
      <orth type=std>pinnas</orth>
    </form>
  </form>
  <gramGrp>
    <!-- déplacé          -->
    <pos location=p1>n</pos>
  </gramGrp>
...
```

Ces différents mécanismes permettent de représenter la plus grande partie des divergences entre vues. Il est toutefois conseillé de les utiliser avec parcimonie, car ils peuvent conduire à une grande complexité du document, et d'utiliser le codage en documents distincts dès que les divergences entre vues deviennent importantes.

5. Conclusion

Les propositions de la TEI ont été testées par le groupe de travail sur de nombreuses entrées de dictionnaires dans différentes langues. Plusieurs équipes dans le monde sont, à l'heure actuelle, en train de les appliquer à la création ou à la rétro-conversion des dictionnaires les plus variés, et il est probable que cette utilisation en grandeur réelle aboutira à des propositions de révision et peut-être de simplification ou d'harmonisation. De même, l'extension aux dictionnaires anciens, ou aux gros dictionnaires comme l'*OED* ou le *TLF,* ne manquera pas de faire apparaître de nouveaux problèmes et difficultés. Les principes de base de la norme TEI semblent suffisamment robustes pour supporter une telle extension[9], mais il est concevable que de nouvelles balises ou de nouveaux attributs doivent être développés.

Le développement d'une norme d'encodage des dictionnaires s'est avéré extrêmement difficile. Cependant, d'une manière générale, les difficultés rencontrées par le groupe de travail de la TEI n'ont pas été dues, comme on aurait peut-être pu s'y attendre, à un manque de consensus entre lexicographes sur la typologie des champs d'information et l'organisation des entrées. Les difficultés ont été, pour la plupart, d'ordre technique. Par exemple, le présent article a permis d'exposer deux problèmes importants, d'une part la tension entre la prise en compte d'une grande diversité de structures et la description de dictionnaires spécifiques et, d'autre part, le conflit entre les deux vues possibles des dictionnaires, comme textes et bases de données.

Dans de nombreux cas, il semble que les limites du langage SGML aient été atteintes: si puissant et utile qu'il soit, il a été conçu pour la représentation de documents simples, tels que manuels techniques ou correspondance commerciale, et la complexité de textes tels que les dictionnaires (ou les textes littéraires en général: manuscrits anciens, éditions critiques, etc.) semble indiquer la nécessité d'un langage de représentation de données de nouvelle génération, doté d'une plus grande flexibilité et d'une plus grande capacité expressive. Ne serait-il pas paradoxal que des préoccupations lexicographiques et littéraires contribuent à l'émergence de nouveaux langages informatiques?

Notes

1 Le groupe de travail sur les dictionnaires était composé de Robert Amsler, Susan Armstrong-Warwick, Nicoletta Calzolari, Carol Van Ess-Dykema, John Fought, Nancy Ide, W. Frank Tompa, et Jean Véronis.

2 Il est à noter que, malgré le fait qu'une base de données puisse être générée a partir des informations de n'importe quel texte (tels que les textes historiques décrits dans GREENSTEIN/BURNARD, 1995), un dictionnaire *est* une base de données par destination.

3 Cependant, rien n'empêche un affichage moins linéaire: on peut s'attendre à ce que, dans l'avenir, les dictionnaires électroniques soient de nature beaucoup plus "hypertextuelle" et permettent aux utilisateurs de naviguer dans et entre les entrées, associent au texte des entrées des sons, des images, des exemples extraits de corpus, etc.

4 Dans cet article, on utilisera les abréviations suivantes pour les noms des dictionnaires:

CED	*Collins English Dictionary*
CP	*Collins Pocket Dictionnaire*
DNT	*Dictionnaire de Notre Temps (Hachette)*
OALD	*Oxford Advanced Learner's Dictionary*
OED	*Oxford English Dictionary*
PL	*Petit Larousse*
PR	*Petit Robert*
TLF	*Trésor de la Langue Française*

5 Par exemple, on peut considérer les entrées suivantes du *CED:*

dead man's handle *or* **pedal**...
confidence man *or* **trickster**...

Dans le premier cas, le mot suivant la conjonction *or* remplace le dernier mot du syntagme qui précède; dans le second cas, le mot suivant la conjonction *or* est un remplacement de tout le syntagme précédent. Les formes développées seraient les suivantes:

(dead man's handle) *or* **(dead man's pedal)**
(confidence man) *or* **(trickster)**

Aucun algorithme simple ne peut faire ce type de distinction qui requiert des connaissances sémantiques complexes et difficiles à modéliser dans un ordinateur.

6 Voir le chapitre "Default Text Structure for TEI Documents" de TEI P3.

7 Il est à noter que, dans cet exemple et dans certains exemples qui suivent, on n'encode ni le mot *or* ni les parenthèses autour des prononciations parce qu'ils sont automatiquement restituables: voir la discussion sur le texte de rendu dans la section 4.2 ci-dessous.

8 Par exemple, le *TLF* a été conçu à l'origine pour comporter une quarantaine de volumes, mais ce nombre a été réduit considérablement après la parution des six premiers volumes, ce qui a engendré des modifications importantes dans le format et dans la structure des entrées des volumes suivants. Voir MARTIN 1994.

9 Nous avons pu constater que ces principes sont adéquats dans un travail préliminaire que nous avons mené sur le Tome 14 du *TLF*, dont la version électronique nous a été aimablement confiée par l'INaLF (que J. Dendien et D. Piotrowski en soient remerciés).

Reférences bibliographiques

AMSLER (R.A.), TOMPA (F.W.) (1988), An SGML-Based Standard for English Monolingual Dictionaries. In *Information in Text: Fourth Annual Conference of the UW Center for the New Oxford English Dictionary,* University of Waterloo Center for the New Oxford English Dictionary, Waterloo, Ontario, 61-79.

BURNARD (L.), What is SGML and how does it help, *in* IDE (N.), VÉRONIS (J.) (Ed.), *Text Encoding Initiative: Background and Context.* Dordrecht, Kluwer Academic Publishers, 1995, p. 41-50.

GREENSTEIN (D.), BURNARD (L.), Speaking with one voice: Encoding Standards and the Prospects for an Integrated Approach to Computing in History, *in* IDE (N.), VÉRONIS (J.), *Text Encoding Initiative: Background and Context.* Dordrecht, Kluwer Academic Publishers, 1995, p. 137-148.

IDE (N.), SPERBERG-MCQUEEN (C.M.), The Text Encoding Initiative: its history, goals and future development, *in* IDE (N.), VÉRONIS (J.), *Text Encoding Initiative: Background and Context.* Dordrecht, Kluwer Academic Publishers, 1995, p. 5-15.

IDE (N.), VÉRONIS (J.) (Ed.), *Text Encoding Initiative: Background and Context.* Dordrecht, Kluwer Academic Publishers, 1995, 342p.

ISO 8879:1986. Information Processing--Text and Office Systems--Standard Generalized Markup Language (SGML). *International Organisation for Standardization,* Geneva, 1986 [aussi publié en français par l'*Association Française de Normalisation* (AFNOR) sous la référence AFNOR Z 71-010 -- Traitement de l'information--Systèmes bureautiques--langage normalisé de balisage généralisé (SGML), Paris, 1990].

MARTIN (R.), Présentation (Numéro Spécial: Autour du T.L.F.). *Le français moderne , LXII*, 2, 1994, p. 129-134.

SPERBERG-MCQUEEN (C.M.), BURNARD (L.), *Guidelines for Electronic Text Encoding and Interchange*, Text Encoding Initiative, Chicago and Oxford, 1994.

THE DANLEX GROUP, *Descriptive tools for electronic processing of dictionary data,* Niemeyer, Tubingen , Lexicographica, Series Maior, 1987.

UNIVERSITE d'AIX-MARSEILLE II
FACULTE Pierre PUGET - L.I.T.A.M.
14, rue Puvis de Chavannes
13001 Marseille
FRANCE

Téléphones : 91 13 96 00 - 91 13 96 29 <= nouveau
Télécopie : 91 90 58 29
E-mail : Knippel at FRAIX11.UNIV-AIX.FR <= nouveau
Knippel at FRMRS11.U-3MRS.FR <= nouveau

SEMINAIRES DE RECHERCHE 1994

GROUPE DE FORMATION DOCTORALE D'INFORMATIQUE FONDAMENTALE ET SCIENCES DE LA COMPUTATION

Salle de conférences. Les mercredi de 18h à 19h30

JEUDI 24/02/1994 (exception)	L.POPOVA Professeur Université de Poitiers	Méthodologies de développement des systèmes experts
23/03/1994	M.LAI Consultant - Expert Ingénia	Méthodes de conception orientée objet : BOOCH, HOOD MOOD, OOD
06/04/1994	E.BIANCO Professeur Université d'Aix-Marseille II	Processeur de la procédure formelle
25/05/1994	M.EGEA Maître de Conférences Université d'Aix-Marseille III	Systèmes multi-agents coopératifs
29/06/1994	P.SANCHEZ Ingénieur C.N.R.S. L.A.S.- Marseille	Langages de description des circuits logiques
12/10/1994	I.G.TABAKOW Professeur Université de Sofia - Bulgarie	(à confirmer) Test generation for sequential logic circuits using Petri nets
16/11/1994	M.COTTON, A. PEREZ Maîtres de Conférences Institut Charles Fabry Université de Provence	Un calculateur parallèle personnel pour la simulation de transitions de phase
14/12/1994	I.MATVIICHINE Professeur Académie des Sciences d'Ukraine	Relations scientifiques entre la France et l'Ukraine du XV° siècle à nos jours

Les résumés des interventions seront disponibles au L.I.T.A.M

Université de Provence
Atelier de Reprographie
Centre Saint Charles
3, place Victor Hugo
F - 13331 Marseille Cedex 3