

Publié dans

S.Bolasco. L.Lebart , A.Salem Analisi Statistica dei Dati in Testuali, vol II ,CISU. Rome ,289-292,in vol 2 p 105. 112
vol 2, conférence JADT , 1997
version finale et définitive dans la publication.

**ALADIN : Un Atelier Génie Logiciel Orienté Objets
Pour La Lecture & l'Analyse Intelligente de Textes****Ahmed Seffah & Jean Guy Meunier**

Laboratoire d'Analyse Cognitive de l'Information

Université du Québec à Montréal

CP 8888, Montréal, Québec, Canada

Téléphone : (514) 987 - 0339

Fax : (514) 987 - 8798

Mail : seffah(meunier)@pluton.atoci.uqam.ca

Résumé : De plus en plus, la technologie informatique cherche à dépasser la simple gestion des documents textuels pour en analyser le contenu [4]. Mais dans une perspective intelligente, l'analyse et lecture de textes assistées par ordinateur (LATAO) doit être orientée par les besoins et les objectifs de l'utilisateur. Dans ce papier, nous présentons ALADIN (pour Atelier pour la Lecture et l'Analyse d'Informations) qui est bâtie autour d'une représentation orientée objets unique des informations manipulées tout au long du cycle d'analyse. ALADIN supporte une modélisation du processus LATAO sur les deux niveaux : **(1)** définition cognitive du processus d'analyse (LATAO), **(2)** description de ce processus dans un formalisme orientée objets.

Mots clés : logiciels d'analyses textuelles, informatique linguistique.

1. Introduction

Il existe actuellement plusieurs logiciels de traitement informatique de l'information textuelle, comme par exemple Interleaf, Spirit, Darwin, Sato, Book Manager, Open Text, Texto etc. (station de la BNF) ??? Presque tous ces logiciels se présentent comme des boîtes à outils souvent sophistiqués d'ailleurs, dont l'interaction ne repose pas toujours sur une modélisation intégratrice, globale et cohérente du processus cognitif de lecture et d'analyse de textes. Malgré les nombreuses fonctionnalités offertes par ces logiciels, la solution proposée ne couvre qu'une partie des du processus cognitif de l'utilisateur.

La lecture et l'analyse de texte ne sont pas des activités cognitives complexes. Plusieurs méthodes ont, à travers les siècles été proposées modéliser ce processus cognitif de lecture et d'analyse de textes. Parmi celles-ci on en retrouve une qui a été pratiquée par des milliers de lecteurs, exégètes et interprètes. Cette méthode, appelée philologie soutient que, l'interprétation des textes relève d'une démarche rigoureuse et systématique de lecture et d'analyse. À l'origine, cette méthode prenait surtout la forme d'un examen et d'une étude comparative et analytique de manuscrits et de textes. Mais elle s'est lentement transformée en une démarche rigoureuse de descriptions, de commentaires, de compréhension et d'évaluation des textes. Malgré son âge, cette méthode possède dans son élan général un poids d'expérience qui, dépoussiérée, peut devenir des plus précieuse dans la constitution d'une approche rigoureuse de lecture et d'analyse de textes assistées par ordinateur (LATAO). En effet, cette méthode n'offre pas avant tout des grilles mais des balises qui encadrent le processus technique de la lecture et de l'analyse. Pour utiliser cette méthode cependant, il nous faut l'adapter au cadre cognitif qui traverse la problématique de LATAO et surtout la positionner en regard de son implantation informatique. En un sens, il faut créer une *philologie électronique*. ou ce que Rastier appelle une herméneutique matérielle. Lorsqu'on veut cependant modéliser plus clairement cette méthode aux fins de LATAO, on constate qu'elle n'est pas vue de la même manière par ceux qui doivent en décrire l'architecture. Un philologue, un cogniticien, un informaticien ne la voient pas du même point de vue. Le philologue la considère sous l'angle des tâches qu'il aura à effectuer. Le cogniticien l'approchera en tant que suite d'opérations cognitives. L'informaticien la décrit sous l'angle des fonctionnalités à implanter. Malgré cette divergence, il demeure quand même possible de les réunir dans une modélisation intégratrice de ces trois points de vues. L'objectif de la présente réflexion est de définir, concevoir et développer un *système logiciel capable*, capable de supporter le processus cognitif du LATAO. Nous déploierons cette analyse en x modèle, Un premier cognitif, et deuxième fonctionnel, informatique.

2. Aperçu sur la modélisation cognitive du LATAO

Ce premier niveau de design vise, par des catégories et des concepts appropriés et qui ne sont pas nécessairement formels, à identifier la représentation que le lecteur et l'analyste expert - en l'occurrence de type philologue - se donne de son propre travail. Un examen systématique de cette pratique révélera révlpar analyse de protocole est ici utile. Elle aidera à faire monter que la démarche du philologue se déploie de fait en une suite complexe de tâches et de techniques que l'expérience et la tradition disciplinaire lui ont enseignée et auxquelles il est fidèle. Lorsqu'on tente de la généraliser en termes d'opérations

cognitives cette méthode philologique apparaît de fait comme une instanciation particulière d'un traitement cognitif d'information mais adaptée sur à des intrants de nature textuelle . Aussi , elle possédera les quatre grandes classes spécifiques d'opérations que toute chaîne de traitement d'information présente. Ainsi , elle exigera que le lecteur ou l'analyste : (1) manipule *la matérialité de l'information* textuelle, (2) projecte sur les données textuelles des *catégories descriptives et classificatoire*, (3) *explore* ces informations structurées selon des perspectives propres, (4) *configure* les résultats obtenus en regard de ses propres structures cognitives.

2.1 Les quatres classes d'opérations de LATAO

L'inscription : toute lecture et analyse de texte impliquent un point de départ où l'information textuelle est inscrit sur un support matériel quelconque. Cette inscription exige alors que l'utilisateur puisse manipuler les documents physiques (manuscrits, monographies, fiches, etc.) qui constitue le corpus d'informations textuelles sur lequel portera sa lecture et son analyse. Dans un cadre informatique, cette inscription correspond pour une bonne part aux étapes de manipulation physique de la documentation c'est-à-dire à ce qui est appelé la Gestion électronique des Documents, GED, (Dupoirier, 1994). Cette GED offre en effet, tout un ensemble de fonctions qui touchent la saisie (traitement de texte), la gestion et l'archivage, le rappel (base de données, interfaces), etc. D'un point de vue strictement de l'analyse fonctionnelle, cette manipulation du document ne touche pas encore le véritable contenu textuel. Elle n'atteint la signification du texte que de manière embryonnaire. Elle travaille surtout au niveau du support de l'information. Pour aller plus loin dans le contenu du texte, un travail de description interprétatif doit être entamé.

La description : qui produit un marquage ou un étiquetage du texte. Le texte original sera alors transformé en un texte catégorisé c'est-à-dire un "surtexte". Les types de catégorisations sont nombreux. Elles peuvent être de nature:

- lexicale : mots composés, expressions idiomatiques, expressions numériques, etc.
- syntaxique: :les parties grammaticales,
- sémantique :la signification des expressions, ... ,
- pragmatique :les utilisateurs, les destinataires , ... ,
- argumentatif : les présupposés, les arguments, etc. ,
- logique : les prémisses, les mineures, la conclusion, ... ,
- éditique : les paragraphes, les titres, les sections, ... ,
- narratif: les actants , les actions , ... ,
- hypertextuelle : relations aux autres parties du texte,
- documentaire : mots clefs, mots du thésaurus, etc. ,
- interprétatives sociologie, psychologie, anthropologie,
- stylistique: le genre,

L'exploration : une fois le texte catégorisé, un utilisateur cherchera soit à naviguer dans ce surtexte soit à en extraire des informations pertinentes. Cette exploration sera donc appliquée au texte catégorisé c'est-à-dire au "surtexte" pour en tirer des informations ciblées c'est-à-dire des extraits qui ne sont eux-mêmes que d'autres textes. Cette exploration

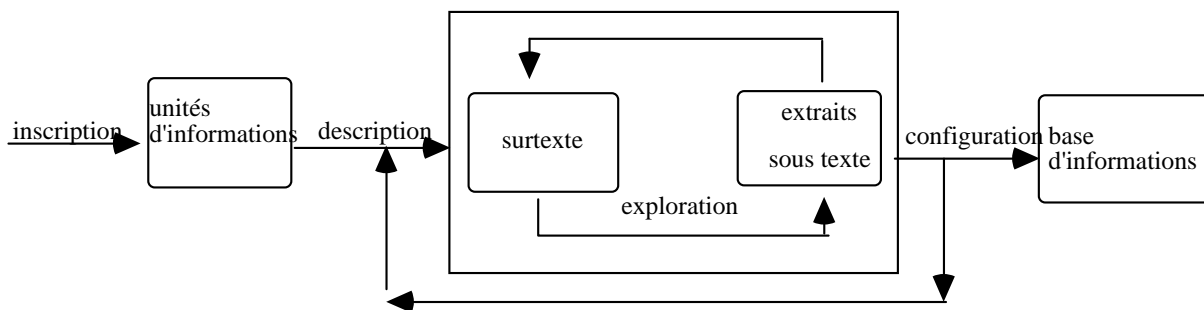
est habituellement paramétrisée par des patrons de fouille qui tantôt sont de types numériques (statistiques) tantôt structuraux (une quelconque grille). Par exemple, cette opération peut être un ensemble de sous-textes spécifiques reliés par une catégorisation commune ou encore des listes, des réseaux, des associations, des segments équivalents ou tout simplement des réponses à des requêtes paramétrisées. Il existe évidemment une grande variété de stratégies d'exploration dont les plus classiques: la concordance, la cooccurrence, le dépistage, l'analyste thématique, la classification, la navigation hypertextuelle. etc.

La configuration : il arrive souvent, surtout lorsque le corpus est immense, que l'exploration donne des résultats dont l'ampleur ou la structure demeure indigeste pour un utilisateur. Or un lecteur voudra souvent construire à partir des nombreuses analyses issues du texte, une base de données structurées qui lui permettra de naviguer dans ses résultats selon ses intérêts. L'utilisateur construira alors une configuration qui traduit les résultats d'analyse pour qu'ils soient plus ajustés à sa modalité propre d'organiser son savoir. . Il voudra configurer son information en une représentation cognitivement pertinente pour lui. En ce sens, il se construira une Base d'Informations (BI) personnelle .

. 2 La chaîne d'analyse

Ces quatre grandes classes d'opérations modélisent de manière générale les grandes fonctions que toute lecture et analyse de texte par ordinateur devront confronter. Chaque parcours ou itinéraire exige des combinaisons spécifiques de ces opérations. A chaque fois cependant, se constituera pour un utilisateur une chaîne particulière propre à son projet d'analyse et de lecture Il y aura autant de parcours possibles que de lecteurs-analystes et de projets de lectures.

On peut représenter plus formellement ces parcours comme des chaînes de traitement. Chacune suit alors une suite de transformations que le lecteur et l'analyste fait subir à un texte pour en atteindre le contenu. Une chaîne de traitement sera, de fait, un choix particulier d'opérations possibles de ces quatre grandes classes d'opérations. Certaines de ces opérations pourront être récursives.



Par exemple, les résultats de l'exploration et de la configuration pourront être réinjectés dans le texte et ainsi ils pourront augmenter de manière continue la richesse du texte et même des analyseurs. Le nouveau texte pourra alors faire l'objet d'une nouvelle exploration et

ainsi de suite. C'est ce modèle cognitif qu'il faut maintenant traduire premièrement dans un *modèle formel orienté objet* et deuxièmement dans un *atelier informatique* .

3. La modélisation fonctionnelle du LATAO

La modélisation formelle du LATAO vise à traduire les objets et opérations de la modélisation précédente dans un formalisme. Elle mettra, entre autre, en évidence que toutes les opérations ne sont pas traduisibles en des algorithmes. Aussi dans certains cas les opérations devront être réalisées manuellement par des experts humains. Certaines autres cependant pourront être réalisables par ordinateur. Et ce qui décide qu'une opération particulière est réalisable par ordinateur est sa nature computationnelle. Ce qui n'est pas toujours évident. Aussi, l'informatisation des opérations devra être simple, mais elle devra être adéquate, fidèle et bien interprétable.

Pour ce faire, une modélisation orientée objets est très appropriée. En effet, la modélisation par objets se construit logiquement indépendamment des langages de programmation. Ce type de modélisation est fondamentalement une nouvelle manière de penser l'implantation informatique elle n'est donc pas avant tout une technique de programmation. Ses avantages les plus significatives viennent de l'aide qu'il apporte aux analystes, développeurs, clients voire même à des chercheurs pour exprimer les concepts abstraits et se les communiquer. Il peut servir de moyen pour la spécification, l'analyse, la documentation et l'interface avec les clients autant que pour la programmation.

Plusieurs modélisations orientées objets ont émergé ces dernières années. Nous avons choisi de travailler avec la méthodologie OMT pour sa puissance expressive et la clarté de sa notation. OMT emploie trois modèles différents pour décrire un système : le modèle objets, le modèle dynamique et le modèle fonctionnel.

Ce modèle construit une structure où les entités de base sont appelées des objets qui possèdent une identité, des inter-relations, des attributs, et sur lesquels sont appliqués des opérations. Le modèle objet est représenté, sous forme graphique, à l'aide de diagrammes d'objets. Un diagramme d'objets est un graphe dont les nœuds sont des classes d'objets et dont les arcs expriment les relations entre ces classes. Une relation peut être de diverses natures : agrégation (simple, alternative), généralisation (héritage), association, composition.

En ce qui nous concerne, ce modèle se déploiera en deux ou trois sous-modèles : introduire ici le dynamique, le fonctionnel etc.

Il décrira (note qu'il décrira plus que ce que je pense) entre autre les différents états d'un certain type d'objets que nous appelons ici un *texte* et les opérations de changement d'états (les opérations sur le texte) sous forme d'objets. (et tout le périphérique d'aide etc.) La description globale du modèle objet pour le LATAO s'avère cependant complexe car d'un fait elle met en évidence de nombreuses dimensions souvent non explicites dans la démarche intuitive. La figure 1.a donne un aperçu de ce premier modèle.

Le modèle dynamique sert à spécifier et à implémenter les aspects du système en relation avec le temps et le séquençement des opérations : les événements qui marquent les changements, les états qui définissent le contexte des événements, et l'organisation des états et des événements. Il modélise donc le contrôle d'un système qui décrit les séquences d'opérations se produisant, sans attacher d'intérêt à ce qu'effectuent les opérations ou à ce quoi elles opèrent. Il se présente sous forme de diagrammes d'états dont les noeuds sont des états et dont les arcs sont des transitions entre les classes d'objets provoqués des événements (les opérations attachées aux objets du modèle objets).

Le modèle fonctionnel décrit la transformation des valeurs dans le système : fonctions, correspondances, contraintes et dépendances fonctionnelles. Le modèle fonctionnel se présente sous forme de diagrammes de flots de données qui montrent les dépendances entre les valeurs et le calcul des valeurs de sortie à partir des valeurs d'entrée et des fonctions, sans ce tenir compte du moment ni même de l'exécution des fonctions.

Figure 1 : Aperçu des trois modèles pour le LATAO (notation OMT)

La modélisation OMT du LATAO nous permet de simplifier la vision de complexité du processus LATAO. Cette modélisation offre un cadre pour la formalisation et l'étude du LATAO selon trois angles d'étude très complémentaires et cohérents.

4. Modélisation informatique du LATAO : l'atelier ALADIN

Les deux modélisation précédentes a savoir : cognitive et formelle sont d'une grande importance. Cependant, pour tirer véritablement profit de leur puissance, il est important de les exploiter d'une façon coordonnée et cumulative. Cette approche passe par la définition d'une plate forme informatique capable de soutenir informatiquement un certain nombre des opérations de –LATAO. Cette définition de la plate forme passera par un concept important utilisé dans les théories de génie logiciel a savoir de "paradigme d'atelier (Girad 89)].

Un atelier de génie logiciel (AGL) se présente comme un ensemble intégré d'outils articulés autour d'un (ou de plusieurs) référentiel(s) qui vise à automatiser toutes les opérations à faire à toutes étapes d'une méthodologie[Afc 92]. En ce qui nous concerne la méthodologie articulée autour des quatre étapes du processus LATAO.

L'atelier doit répondre aux six exigences suivantes : **(1)** répondre aux besoins spécifiques de chaque analyste (terminologue, traducteur, documentaliste, philologue, ...) c'est-à-dire l'atelier doit pouvoir être configurable d'où la nécessité d'un niveau méta de l'atelier [2], **(2)** conseiller l'utilisateur dans sa démarche d'analyse en lui proposant des enchaînements (que nous appelons méthodes d'analyse) plus adaptées à ses objectifs, **(3)** supporter tout le cycle d'analyse c.à.d proposer un panel d'outils capable d'automatiser l'ensemble des opérations à faire et ceci pour toutes les étapes du cycle d'analyse, **(4)** prendre en considération les contraintes et les progrès technologiques liés à la gestion et au transport de l'information en particulier les différents formats (textes non structurés, SGML,...) et les outils pour les

manipuler, (5) permettre d'utiliser divers autres systèmes d'analyse qui sont de nature mathématiques comme les coorccurrence (voir church 19xx Demande a Djamel la référence)les classifieurs (, l'analyse en composante principale (Benzzecri, Lebart etc voir djamel)réseaux de neurones, (Meunier 19xx) champs de Markov [7]) ou bien de nature sémiotiques (syntaxique, sémantique [6] [3]) (6) offrir un interface adaptée à la dynamique itérative et interactive de l'analyse de textes.

La structure générale de l'atelier appelé ALADIN (pour Atelier pour la Lecture l'Analyse d'INformations) que nous proposons repose sur un concept intégrateur unique appelé ici référentiel à savoir celui de représentation de l'analyse en termes d'objets..Autrement dit un atelier est un ensemble intégré d'objets sur lesquels des opérations ou des outils seront appliqués. Chaque outil dans cet l'atelier utilise des attributs de certains objets et contribue au processus LATAO.

La figure suivante montre la structure actuel d'ALADIN qui se caractérise par quatre composantes :

- une base de données qui englobe les objets d'analyse,
- un ensemble intégré d'outils capables d'automatiser le LATAO.
- un interface utilisateur.

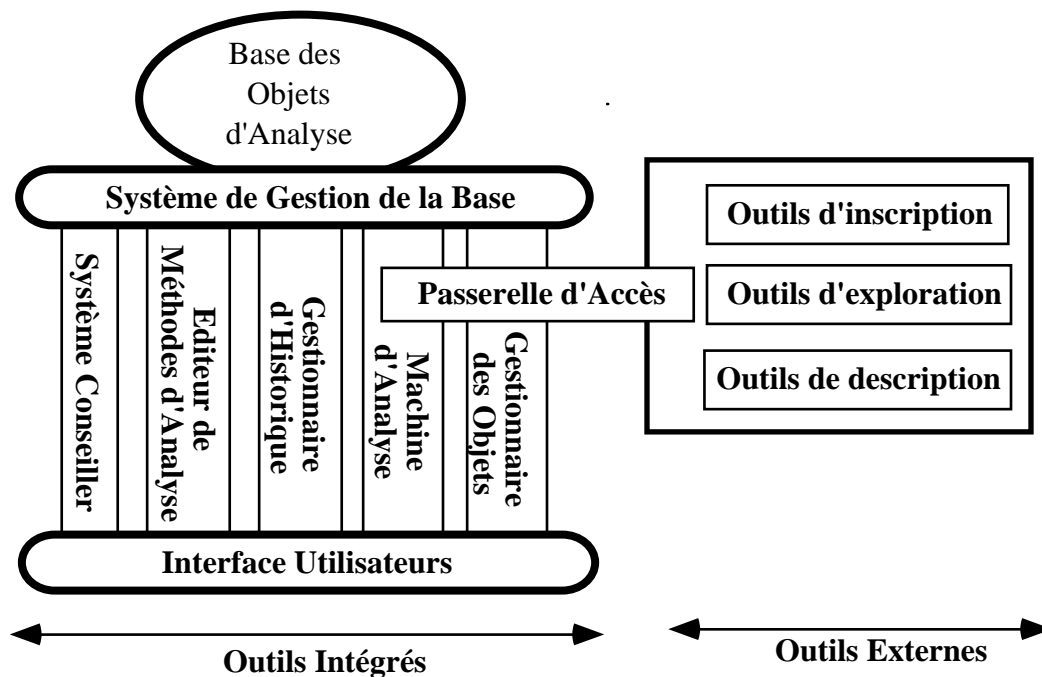


Figure 1 : Architecture générale de l'atelier ALADIN

Dans la classification des outils, nous distinguons deux grandes catégories :

1- Les outils propres c.a.d des outils intégrés dans l'atelier. A l'heure actuelle, on retrouve dans cette catégories les outils suivants :

- le conseiller : celui qui propose des méthodes d'analyse, à la demande de l'analyste—ou lorsque conseiller identifie soit les objectifs poursuivis par l'analyste, soit des méthodes d'analyse appropriées. Ce système constitue une instance particulière du modèle générique proposée dans [Seffah 95], Le lecteur intéressé trouvera donc dans cette publication une description détaillée de l'architecture de ce système,
- éditeur de méthodes d'analyse :
- superviseur :
- gestionnaire d'objets :
- machine d'analyse : qui constitue la composante centrale de l'atelier. g

Dans la deuxième catégorie, on retrouve les outils d'analyses externes c.a.d les outils existants, disponible dans le commerce, pour d'inscrire, décrire ou explorer un texte. Parmi ces outils citons :

- les traitements de textes, lecteurs optiques, les digitaliseurs,
- les descripteurs morphologiques, syntaxiques, sémantiques, grammaticaux, documentaires voire hypertextuelles,
- les outils d'analyse statistiques :
- les classifieurs : analyse en composantes principales, champs de markov, réseaux de neurones,
- les systèmes de gestion de documents textuels : recherche, édition, correction

C'est cette flexibilité de l'atelier dans la construction de chaînes diversifiées de traitement qui matérialise le respect des besoins de l'utilisateur (faire un lien avec cette idée du début)

Notons que la flexibilité de l'atelier c.a.d son aptitude à permettre la construction de chaînes d'analyse appropriées aux besoins de l'utilisateur et regard de ses projets. se mesure par la diversité et la richesse en fonctionnalités des outils de cette deuxième catégorie. Pour supporter cette richesse et diversité, une passerelle d'accès permettra de transformer les données (les résultats) des formats utilisées par les outils à la représentation interne adoptée dans la base d'analyse.

Figure 2 : Une vue panoramique des outils propres dans l'AGL

5. Implantation de l'atelier : la problématique d'intégration dans l'AGL

Un atelier ne se résume pas à une liste une liste éparse d'outils. Dans une implantation informatique ces outils doivent présenter une bonne uniformité (cohérence, complétude) dans leur conception. Or cette cette uniformité pose le problème très épineux de l'intégration. Par quelle moyens assurer en sorte que ces outils soit intégrés? Certains auteurs [Taylor 87] [Ison 89] [Verrall 89] [Blum 90] ont proposée que l'intégration dans un atelier de génie logiciel soit assurée aux quatre niveaux suivants :

- présentation,
- données,
- contrôle,
- iprocessus.

Nous allons donner un bref aperçu sur ces quatres niveaux tout en indiquant les solutions possibles mais surtout les choix que nous avons adoptés.

L'intégration de la présentation concerne l'interface des outils de l'atelier. Une forte intégration de la présentation signifie que tous les outils de l'atelier proposent une interface usager uniforme, c'est à dire qui utilisent les mêmes formes de présentation mais également les mêmes styles de dialogues pour invoquer des fonctions similaires. Fournir des outils ayant le même "look & feel" rend plus simple l'utilisation voire la maîtrise des outils de l'atelier de génie logiciel. Pour ce faire, l'interface utilisateur doit être alors :

- uniforme : identique à tous les outils,
- adaptable aux besoins des différents utilisateurs,
- consistante : gestion pertinente des informations fournies.

L'uniformité (feel and look) des outils s'impose par le choix de la plate forme de réalisation. Cette solution suppose (condition) une séparation explicite de la partie interface utilisateur de la partie modèle. La figure 2 montre l'interface utilisateur définie pour la saisie des paramètres nécessaire pour l'invoquation des outils de descriptions. Dans la réalité cette hypothèse est loin d'être toujours respectée .Ceci présente un sérieux compromis à l'intégration de la présentation.

Par ailleurs, l'adaptabilité de l'interface est assurée essentiellement par le pilote d'outils intégrés. Ce dernier est chargé de délimiter, entre autre, la vision de l'utilisateur sur les outils de l'AGL et sur leurs fonctionnalités respectives. Pour ce faire, il exploite les informations contenues dans le compte utilisateur propre à chaque utilisateur sur ces privilèges et ces restrictions.

L'intégration des données se rapporte à la stratégie de partage et décharge des données exploitée par les outils. L'intégration des données est forte si tous les outils de l'atelier utilisent une seule (unique) structure de données. Ceci nécessite que les données soient modélisées indépendamment des outils susceptibles de les manipuler (création, modification...). De plus, l'organisation de ces données devrait refléter le cycle d'analyse supporté par l'atelier.

Par exemple, il faut que la définition des objets qui constituent ce qu'est un texte soient de même nature aux objets qui constituent le résultat d'une analyse linguistique quelconque ou aux objets qui sortent d'une exploration statistique.. Ce n'est qu'à cette condition que l'intégration des données est assurée. Mais ceci exige évidemment un long travail de réflexion et d'analyse théorique que nous ne pouvons présenter ici.

En ce qui nous concerne, nous avons adopté une intégration forte basée sur une structure de données unique basée sur une modélisation orientée objets. Un module "interface" ou une passerelle de conversion se charge de la transformation des résultats fournis par les outils externes en objets et vice-versa.

Une base de données qui contient toutes ces informations et qui autorise différentes vues sur les objets est une condition nécessaire, mais non suffisante pour assurer une bonne intégration. La réalisation de cette base nécessite de résoudre les problèmes engendrés par :

- la nature des objets : texte, graphique, valeurs qualitatives ou quantitatives
- leur évolution dans le temps
- leur forme de présentation.

L'intégration du contrôle concerne la façon de gérer les communications entre plusieurs outils utilisés dans une même chaîne d'analyse. Une forte intégration du contrôle signifie qu'un outil a les moyens d'invoquer un autre outil qui effectuera une partie du travail d'analyse.

Notre solution, pour assurer l'intégration à ce niveau, se base sur une stratégie maître-esclaves. La machine d'analyse joue le rôle de maître qui contrôle tous les outils externes (esclaves). Cette approche présente l'avantage d'éviter une désorientation de l'analyste.

L'intégration des processus est réalisée s'il est possible de guider les utilisateurs impliqués dans le cycle de vie du didacticiel. Une forte intégration du processus signifie que le modèle du processus d'élaboration contrôle l'invocation des outils. Dans ce cas, un outil ne peut être invoqué que si le modèle de processus choisi autorise cette invocation.

En ce qui nous concerne, l'intégration des processus est prise en compte de la façon suivante. À la demande explicite d'un analyste, l'administrateur de l'atelier définit le compte de l'utilisateur qui regroupe les informations suivantes : la liste des outils et des fonctionnalités de chaque outil, la visibilité sur les objets de la base, les privilèges accordés à l'utilisateur. Lorsqu'il est sollicité, l'atelier examine les informations du compte utilisateur pour configurer l'atelier.

Mais l'idéal serait que l'intégration des processus s'effectue d'une façon dynamique. Cette possibilité passe par la prise en compte du profil de l'utilisateur et du contexte d'utilisation de chacun des outils intégrés. Pour permettre ces deux services, les outils intégrés doivent se comporter comme des agents actifs que l'on peut configurer pour faciliter des activités organisées ou isolées. Il est donc nécessaire de décrire des modèles d'activation d'outils et

de processus-fils qui correspondent le mieux aux tâches à accomplir à une étape du cycle de vie.

6. Impacts LATAO-ALADIN sur la qualité de l'analyse

dans sa conception ALADIN se veut une instanciation informatique d'un processus cognitif complexe qui est la lecture et l'analyse de texte assistée par ordinateur. LATAO. Ce processus, n'est pas similaire au fonctionnement d'un robot-lecteur. Il implique un processus d'aller et retour où l'interprétation se construit au fur et mesure de la pénétration du texte par l'utilisateur en regard de ses projets propres de lecture. Aladin veut de fait donner aux utilisateurs impliquées dans ce processus la possibilité d'assister leur lecture philologique des textes. Les exigences de ce type de lecture (Virbel 19xx) et la dynamique de ce type de lecture exige des plate formes

Cependant le présent travail tant sur la modélisation cognitive du LATAO ou de sa formalisation orientée objets que sur le plan de l'implantation de l'atelier.

7. Bibliographie

[1] Taylor R.N, Baker D.A, Belz F.C, Boehem B.W, Clarke L.A, Fisher D.A. Next generation software environments : principles, problems and research directions. Technical report. University of California, Irvine, 1987.

[2] Skelton J. MetaCase and the Software Process Maturity Model. First International Congress on MetaCASE, Sunderland UK, 1995.

[3] Pustejovsky J, Bergler S, Anick P. Lexical Semantic Techniques for Corpus analysis. In S. Armstrong Edition, Using Large Corpora. Cambridge, MIT Press, 1994.

[4] Meunier J. G, Gastaldy S, Paquin LC. De l'analyse de textes par opposition à la gestion des documents. Revue Informatique Cognitive des Organisations (Québec), Vol 4, 1993.

[5] Meunier, J. G. (1995). La Lecture et l'analyse de texte assistée par ordinateur : La chaîne d'analyse. Cahiers de Recherche Vol 6. Laboratoire d'Analyse Cognitive de l'information. Université du Québec à Montréal, 1995.

[6] Delisle S. Text Processing without a priori domain knowledge : semi automatic linguistic analysis for incremental knowledge acquisition. Ph.D Thesis Ottawa University, 1994.

[7] Bouchaffra D, Meunier J.G. A Markovian Random Field Approach to Information Retrieval. Third International Conference on Document Analysis and Recognition. Montréal August, 1995.

Rastier, F. (1987). *Sémantique interprétative*. Paris: PUF

a ajouter peut être

Landow, G. P. , & Delany, P. (1993). *The Digital Word: Text-Based Computing in the Humanities*. Cambridge: MIT Press.

Rumbaugh, J. , Blaha, M. , Premerlani, W. , E. , F. , & Lorenzen, W. (1991). *Object oriented Modelling and design*. Prentice Hall

Bruner, J. (1990). *Acts of Meaning*. Cambridge, Mass: Harvard University Press.

Virbel, J. (1993). "Reading and Managing Texts on the Bibliotheque de France Station". In P. Delany, & P. Landow (Ed.), *The Digital Word: Text Based Computing in the Humanities*. Cambridge, Mass: MIT Press.