MILLAN Patrick
DESS I.D.C.
(Informatique Double Compétence)

Juin-Septembre 2002

Remerciements

Je tiens particulièrement à remercier l'ensemble des membres de l'U.R.I. pour leur accueil :

- Claire, pour sa disponibilité, son aide, la liberté et le respect qu'elle a accordée à mon travail en tant que responsable, et pour m'avoir expliqué les principes peu évidents de Sdoc et Ndoc.

- Dominique, pour son initiation à l'I.lib, et pour m'avoir fait découvrir toutes les petites choses qui font que l'on est un très bon informaticien.

- Martial, pour sa patience à débuguer certains points de mes programmes, et ses connaissances en Java, et Word, et sa passion pour Windows.

- Xavier, pour son soutien, ses encouragements, et ses conseils concernant la théorie des graphes.

- Patricia, pour sa disponibilité, et sa faculté à résoudre tous nos petits problèmes.

- Ivana, pour son maillot de Ronaldo.

- Fabienne, pour sa bonne humeur.

- Alain, pour son acharnement à me trouver de nouveaux objectifs toutes les 5 minutes.

Merci à tous les membres du DESS IDC, en général (groupes 1 et 2), pour leurs mails, leur solidarité (groupe 2), et plus particulièrement à :

- Aurélie, pour m'avoir aidé dans l'orientation de mes programmes, pour m'avoir envoyé des solutions toutes faites, pour sa présence, même en étant à 200 km de l'I.N.I.S.T., le test critique remarquable de mon site, et pour beaucoup d'autres choses encore.

- Olivier, pour son soutient, le test de mon site.

- Maud, une binôme exceptionnelle.

- JoJo, pour son oreille attentive.

- David, pour ces divertissements.

Merci à toute ma famille, à tous mes amis pour leur intérêt et leur soutien à mon travail.

Introduction générale

Ce rapport fait part de l'ensemble de mes activités à l'I.N.I.S.T.- C.N.R.S. entre le premier Juin 2002 et le 31 Août 2002. Il prend en compte l'ensemble des connaissances acquises, et les travaux réalisés durant cette période.
Ces travaux font partie d'un projet national sur lequel, je suis, et serai amené à travailler jusque la fin décembre 2002 au plus tard.
La durée légale de mon stage s'étalant sur une durée de quatre mois depuis début juin, mon statut au sein de l'entreprise se situe durant cet intervalle à mi-chemin entre celui de stagiaire et celui d'auxiliaire.
En tant que stagiaire, je considère mes activités comme faisant partie d'une formation me permettant d'acquérir des connaissances sur le milieu documentaliste, l'infométrie, la bio-informatique. En conséquence, une partie de ce rapport sera consacrée à un résumé de l'ensemble de ces nouvelles connaissances.
En tant qu'auxiliaire, je suis tenu de fournir un travail à l'entreprise dont je suis l'employé. Ce travail doit permettre à celle-ci de progresser dans son domaine d'activité. Ce rapport présentera une partie des résultats de ma recherche au bénéfice de celle-ci. Ces résultats sont la propriété exclusive de mon employeur, et ne doivent en aucun cas faire l'objet d'une quelconque diffusion. Ils sont partiels, car comme il a été cité précédemment, faisant partie d'un projet dont la date de clôture est prévue pour la fin de l'année en cours.

1. Présentation de l'entreprise

1.1. L'I.N.I.S.T. (Institut de l'Information Scientifique et Technique)

1.1.1. Historique

- **1939** création du Centre National de la Recherche Scientifique (C.N.R.S.)
- **1939** création du service de documentation du C.N.R.S.
- **1947** le service de documentation acquiert le statut de laboratoire propre du C.N.R.S. et devient "Centre de documentation"
- **1970** création du Centre de Documentation Scientifique et Technique (C.D.S.T.) et du Centre de Documentation en Sciences Humaines (C.D.S.H.)
- **1984** décision du transfert du C.D.S.T. en Lorraine par le Ministère de l'Industrie et de la Recherche
- **1985** choix du projet d'architecture de M. Jean Nouvel
- **1986** pose de la première pierre le 28 janvier
- **1987** début de la construction à Vandoeuvre-lès-Nancy
- **1988** création de l'Institut de l'Information Scientifique et Technique du C.N.R.S.
- **1992** décision du C.I.A.T. (Comité Interministériel de l'Aménagement du Territoire) du transfert à Vandoeuvre-lès-Nancy des activités du C.D.S.H.

1.1.2. Missions et moyens

L'I.N.I.S.T. a pour missions la collecte, le traitement et la diffusion des résultats de la recherche scientifique internationale, en France ou à l'étranger, tout en apportant une valeur ajoutée lors de la commercialisation de produits spécialisés. Ces employés perpétuent un savoir-faire qui permet à tous, professionnels comme particuliers, d'accéder aux résultats de recherches dans tous les domaines de la connaissance. C'est dans ce but également que l'I.N.I.S.T. dispose d'une ouverture partielle au grand public dans ses propres locaux (**cf.** www.inist.fr/kiosque).

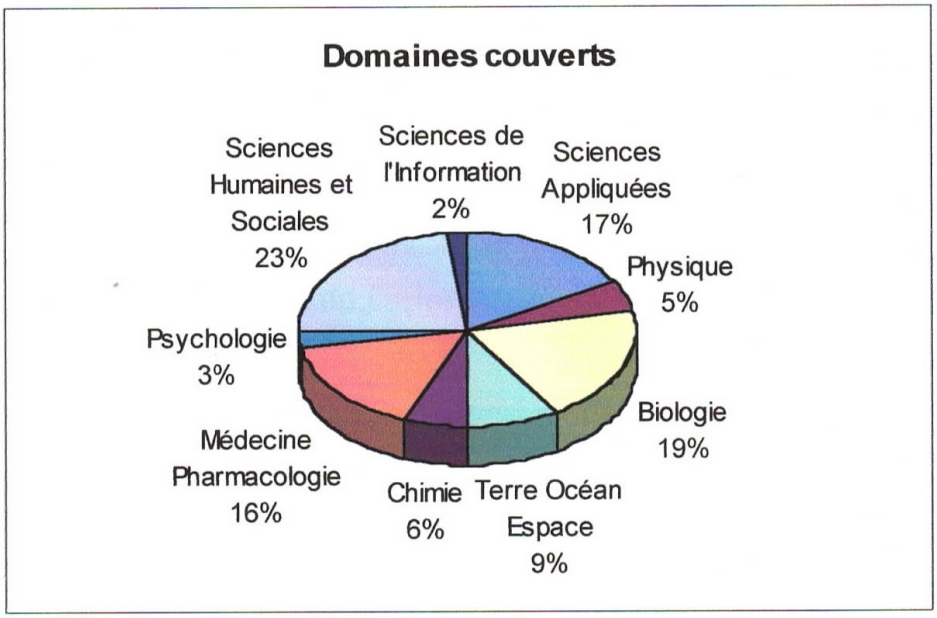


Figure 1 **Domaines couverts par le fond de l'I.N.I.S.T.**

Le fonds documentaire de l'I.N.I.S.T. couvre l'ensemble de la littérature scientifique et technique mondiale (**figure 1**), ce qui représente plus de 34 km de rayonnages, un stock de disques optiques d'une croissance de 2 à 3 millions de pages par an (depuis 1990).
Tous ces supports regroupent actuellement quelques 23 000 périodiques, dont 9 200 collections en cours, 56 000 rapports scientifiques, 60 000 comptes-rendus de congrès internationaux, et de 100 000 thèses de Sciences et Techniques soutenues en France (depuis 1985).
L'I.N.I.S.T. met à disposition gratuite un portail Internet, www.connectscience.inist.fr , représentant un ensemble de ressources et de services d'Information Scientifique et Technique dans un environnement personnalisé. De nouvelles possibilités sont offertes grâce à ARTICLE@INIST, le catalogue en ligne pour la recherche et la commande de documents, téléchargeable également gratuitement à partir du site www.inist.fr. Cet outil permet, entre autre, l'accès à près de 7 millions de références d'articles et de monographies. La mise à jour y est quotidienne avec près de 3 000 nouvelles notices.

Tous ces moyens satisfont sur les propres fonds de l'institut, 93% des demandes de copies de documents. Si ceux-ci sont indisponibles directement, un réseau d'une centaine de bibliothèques ou organismes, français et internationaux, assurent le suivi.

La collecte des documents a permis la transformation et l'enrichissement de l'information, d'où la création de deux bases de données bibliographiques, multilingues et multidisciplinaires : PASCAL et FRANCIS.

PASCAL Créée en 1973, elle contient environ 12 millions de notices bibliographiques avec une croissance de 450 000 par an. Elle est remise à jour hebdomadairement, et est interrogeable en Français, Anglais et Espagnol. Elle couvre des domaines scientifiques tels que la chimie, la physique, les sciences de l'ingénierie, la médecine, et la biologie.

FRANCIS Créée en 1972, spécialisée dans les sciences humaines, sociales et l'économie, elle dispose de 1,8 millions de notices bibliographiques (avec accroissement de

60 000 par an, et une remise à jour mensuelle). Elle est interrogeable en Français et en Anglais.

Consultables sur CD-ROM, minitel, ou en ligne sur divers serveurs, ces deux bases facilitent les recherches, et permettent à diverses unités de L'I.N.I.S.T une personnalisation ponctuelle des services, ou une veille technologique.

La fourniture des copies ou des articles sélectionnés - en version originale ou traduite – constitue l'étape finale du traitement de l'information scientifique et technique.

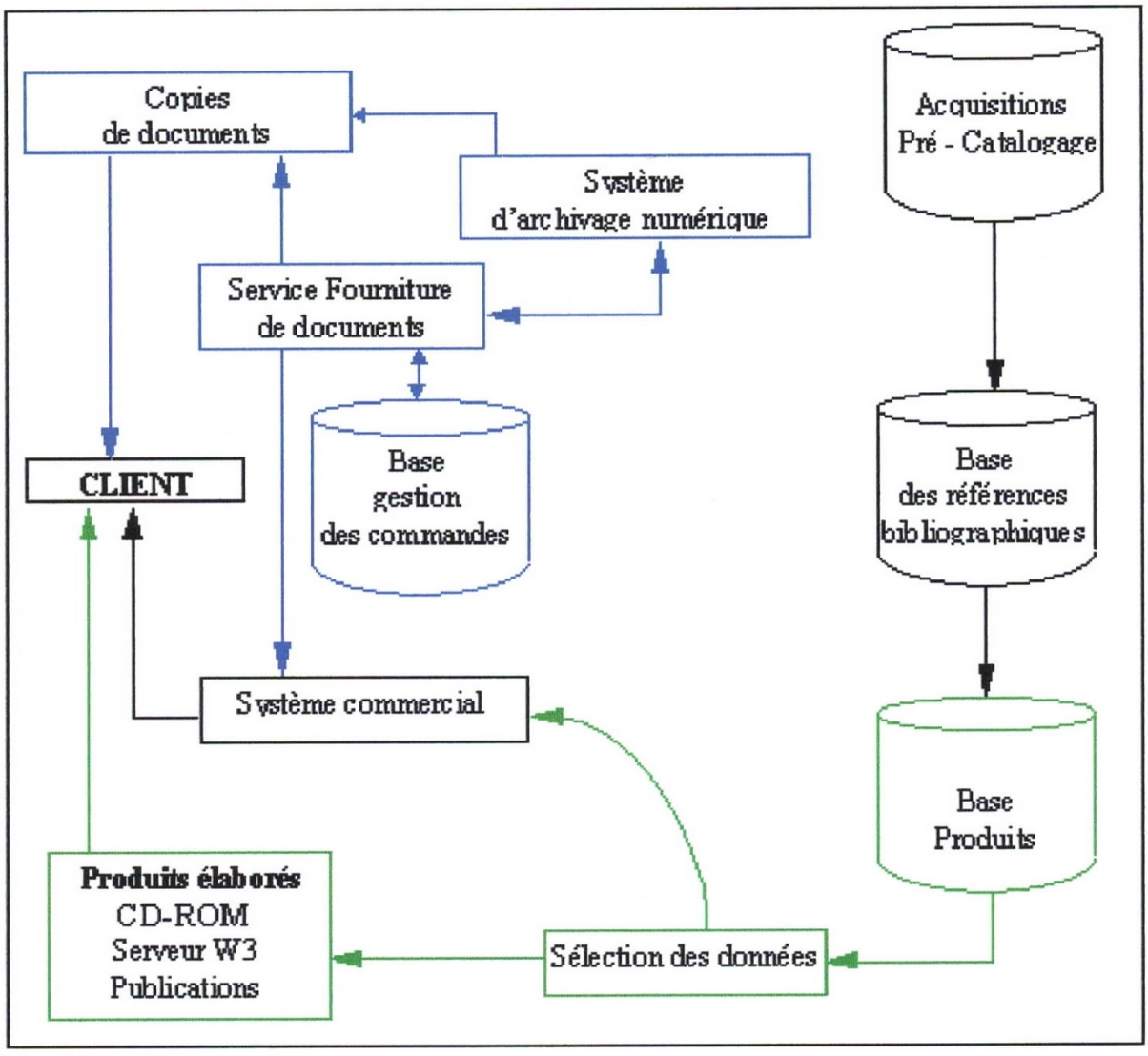


Figure 2 Le système d'information de l'I.N.I.S.T.

Le circuit permettant le traitement de documents par l'I.N.I.S.T. (**figure 2**) commence par l'acquisition et le catalogage. L'ensemble des bases est ensuite alimenté par divers intermédiaires. Le client, l'intervenant final, dispose donc de notices bibliographiques, ou des copies d'articles.

1.1.3. Les locaux

Eléments caractérisant l'I.N.I.S.T., les locaux ont été conçus par Jean NOUVEL, architecte concepteur de l'Institut du Monde arabe de Paris.
Ils se déploient sur 18 000 m² de verre, d'aluminium et de béton.
La disposition de l'ensemble se veut le reflet du flux de l'information drainée dans les quatre bâtiments (APOLLO, HERMES, ARIANE et COSMOS) par la passerelle principale et ses deux transversales.

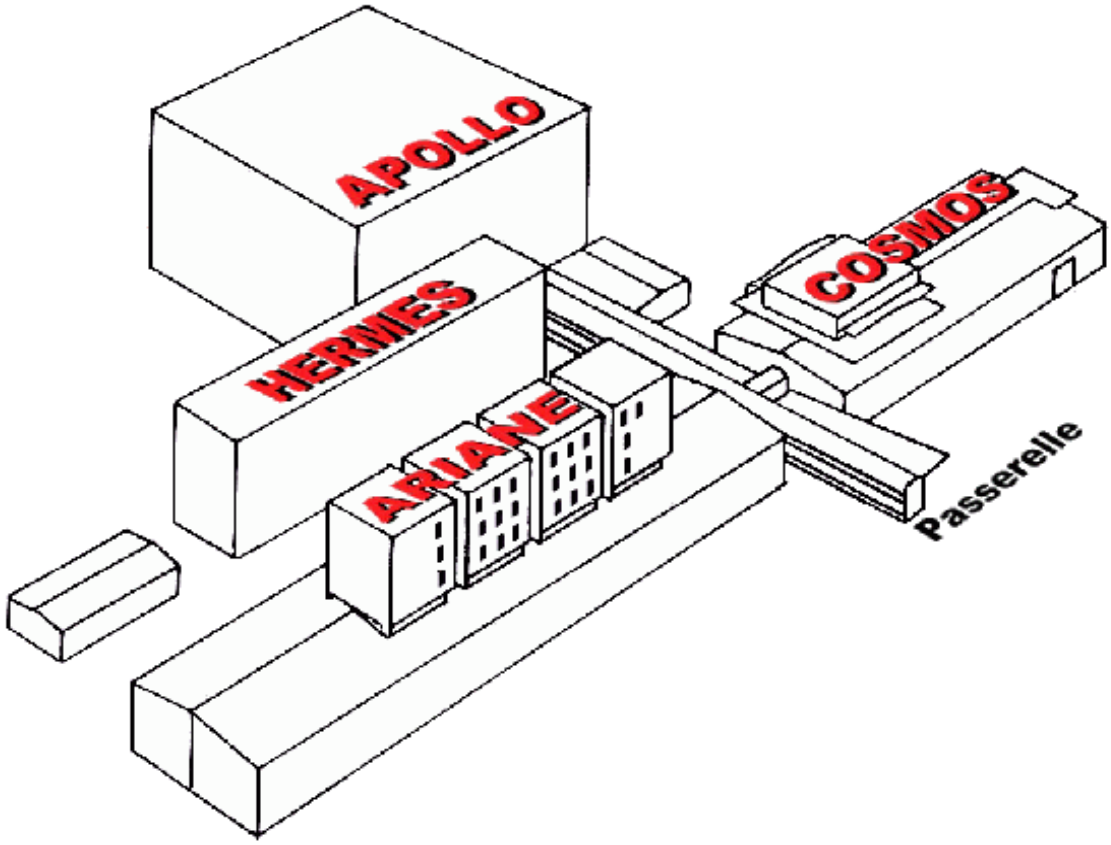


Figure 3 Plan des locaux de l'I.N.I.S.T.

ARIANE	Direction générale - Administration et finances - Ressources humaines - Salle de conférence - Informatique et études - U.R.I.
HERMES	Bases de données Pascal Francis - Direction du Développement
APOLLO	Archivage numérique - Collections et fourniture de documents
COSMOS	Restaurant

1.2. L'U.R.I.(Unité de Recherche et de l'Innovation)

1.2.1. Historique et composition

Créée sur les bases d'un Programme de Recherche en Infométrie (P.R.I.) de 1992, l'Unité de Recherche et de l'Innovation (U.R.I.) est mise en place en 1998 à l'I.N.I.S.T.. Elle est constituée d'une équipe de quatre ingénieurs à double compétence, informatique et scientifique, spécialisés dans les sciences et technologies de l'information (Xavier Polanco, le responsable de l'U.R.I., et des chefs de projets Dominique Besagni, Claire François, Ivana Roche, Alain Zasadzinski, et Martial Hoffmann), et d'une assistante de gestion (Patricia Gautier).

1.2.2. Spécialisation de l'U.R.I. dans l'infométrie

Le terme "infométrie" sert à désigner d'une manière générale les analyses métriques de l'information (statistiques, probabilités et analyses des données).Son application, comme le détaille ce rapport, permet la production de nouvelles informations sur, non pas une, mais l'ensemble des données, de leurs origines, à leurs interactions.

L'U.R.I. a pour mission de faire progresser la recherche appliquée, de concevoir et produire des instruments performants pour l'information scientifique et technique (I.S.T.), soit de nouveaux indicateurs, de nouveaux outils, et de nouvelles méthodes propres à l'infométrie. Tout ceci nécessite la mobilisation de compétences variées en informatique, linguistique, statistique, biologie, physique, mathématique...

L'U.R.I. se doit, en conséquence, d'être impliquée dans de nombreux projets nationaux et internationaux, ayant pour objectifs de valoriser le traitement, l'exploitation de l'information scientifique et technique. Ceci est le cas du projet Bio-informatique I.G.R. servant de contexte à mon stage.

1.2.3. Le projet Bio-informatique I.G.R. – Cancer de la thyroïde

L'évolution rapide actuelle des connaissances sur la séquence complète du génome humain modifie radicalement les approches de la localisation et d'identification des gènes, et en particulier ceux impliqués dans le cancer.
Suite aux événements de 1986 en Ukraine à Tchernobyl, l'intérêt public s'est orienté plus particulièrement vers le cancer de la thyroïde, son origine, ses développements, son traitement.

Le but des projets de recherche, regroupant le projet Bio-informatique I.G.R., est d'associer une expertise au niveau de la définition des familles de gènes par le développement d'outils intégrés d'analyse de séquences protéiques, à une expertise de l'analyse textuelle de résumés

bibliographiques concernant ces familles, en partant en particulier de corpus déjà expertisé (par exemple CancerGene).
Les objectifs informatiques y sont de trois natures :

- La création d'environnement de définition de signatures pertinentes de familles de gènes ;
- L'amélioration des méthodes d'analyse bibliographique automatique ;
- L'intégration dans des bases de données spécialisées.

Le projet bio-informatique expérimentale est réalisé par l'U.R.I. en collaboration avec l'Institut Gustave Roussy (I.G.R.), un Centre de Génétique oncologique de la région parisienne basé à Villejuif.
Il a donc pour objectif de réaliser une expertise de famille de gènes impliquée dans le cancer de la thyroïde en utilisant des approches moléculaires (ne concernant pas directement l'équipe de l'U.R.I.) et textuelles (partie du travail réservée à l'U.R.I.).
Les résultats de ces travaux devront ensuite être intégrés dans la base de données dédiée aux gènes de ce cancer et à leur implication dans les altérations génétiques.

2. L'informatique dans le cadre de la recherche infométrique

2.1. Problématique

L'analyse de l'information nécessite l'appropriation d'une grande masse d'information dont on exclut logiquement tout parcours séquentiel. Elle répond aux besoins de veille scientifique et technologique, et d'analyse stratégique de la recherche. Toutes données bibliographiques et/ou textuelles est extraites et structurées, en utilisant des indicateurs comme :

- Les mots-clés, indicateurs de la connaissance véhiculée par des documents ;
- Les classes (clusters), agrégats de mots-clés et de documents, indiquant les thèmes et centres d'intérêts regroupant une information (articles, auteurs, institutions...) ;
- Les cartes, procurant une vision globale, et stratégique pour apprécier la position relative dans l'espace des connaissances thématiques couvertes par des items analysés.

L'analyste cherche à identifier dans une masse de données, l'information utile, celle comportant un intérêt pour son client.

2.2. Exploitation des données à l'U.R.I.

Durant un processus d'identification, trois phases sont distinctes : le stockage, l'accès à l'information et l'analyse proprement dite.
La considération de ces phases comme fonctions d'un système informatique est envisageable.
L'U.R.I. a orienté son développement vers les techniques analytiques d'informations stockées dans des bases de données, celles-ci étant rendues accessibles par l'ensemble du système infométrique.
L'outil infométrique de l'I.N.I.S.T. est composé de quatre blocs :

- Les statistiques bibliométriques utilisant le serveur MIRIAD, géré par Dominique Besagni, permettant l'accès aux données Pascal et Francis, et effectuant des recherches en fonction de requêtes. Le corpus obtenu est alors l'objet de statistiques bibliométriques.
- L'indexation automatique par des méthodes de linguistique informatique (plate-forme Infométrie Langage et Connaissance I.L.C.).
- Le traitement infométrique de classification et de cartographie, Neurodoc et Sdoc, sous la responsabilité de Claire François.
- Le stockage et l'accès à l'information structurée, HENOCH, organise les résultats des programmes Neurodoc et Sdoc (méthode de classification et cartographie basée sur les mots-asociés) dans un SGBD (Système de Gestion de Base de Données) relationnel et les met à disposition des utilisateurs au travers d'une architecture client-serveur pour Internet.

Les trois premiers blocs de cet outil (les blocs statistiques, indexations et classifications) sont interfacés directement par un outil nommé Stanalyst, ou indirectement par d'autres moyens comme Visa.
Tous ces blocs sont de plus exploitables indépendamment les unes des autres, par l'intermédiaire de programmes BATCH.
Dans le cas du projet I.G.R. par exemple, la classification est lancée indépendamment, et la visualisation des résultats est réalisée à l'aide de Visa.

2.3. Les modules infométriques : Ndoc et Sdoc

2.3.1. Introduction aux méthodes Ndoc et Sdoc

Les profils documentaires et les recherches rétrospectives produisent de petites bases documentaires de quelques centaines de documents peu exploitables par simple lecture.

Ces méthodes permettent de répondre à la demande d'information élaborée et de fournir des outils d'aide à la décision et à l'analyse de l'information stratégique dans une perspective de veille technologique.

Leur approche définit un processus de production de cartes montrant l'organisation conceptuelle d'un domaine scientifique(**figure 4**). Elle est basée sur l'analyse statistique des mots-clés indexant les documents : ces derniers sont préexistants dans les collections de notices bibliographiques extraites de bases de données telles que PASCAL ou FRANCIS, ils peuvent être définis automatiquement par l'outil I.L.C..

L'interprétation de ces cartes, leur utilisation, dépend de celui qui la visualise :

- un ingénieur documentaliste pourra s'en servir comme une aide à la création de thésaurus,
- un 'veilleur technologique' pour visualiser la liste des concepts déterminants d'un domaine,

- un scientomètre pour mesurer l'évolution d'un domaine donné et répondre à des questions du type 'qui fait quoi et où ?'.

La diffusion de ces résultats peut se faire par l'intermédiaire des sites : henoch.inist.fr et visa.inist.fr .

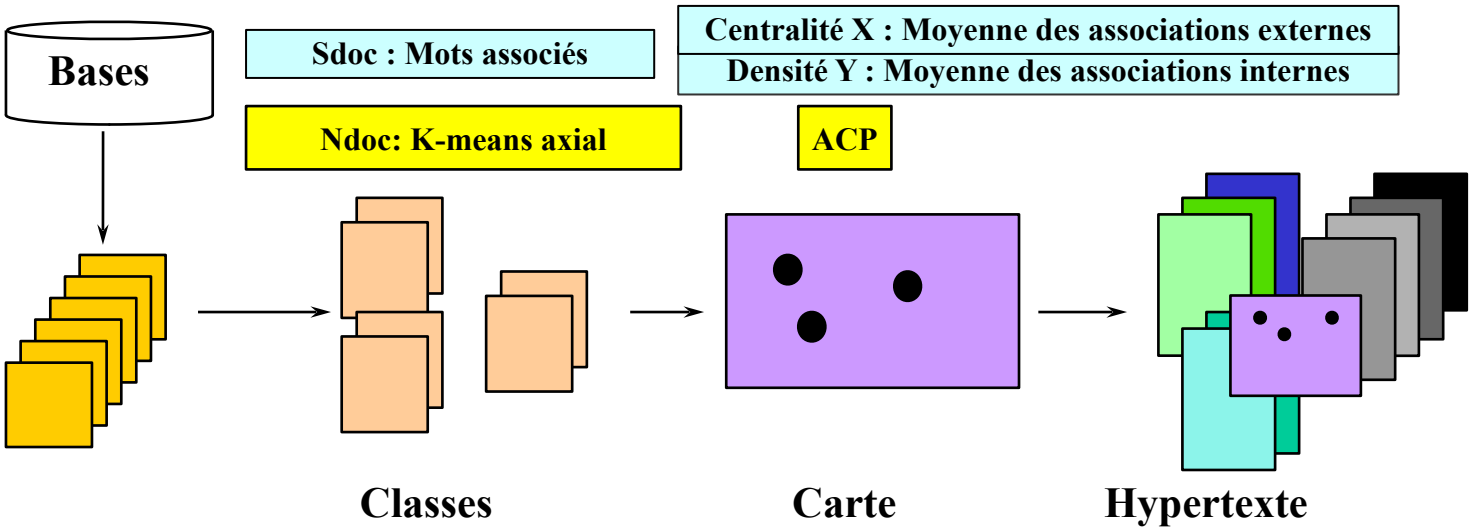


Figure 4 Principe des modules infométriques.

2.3.2. La méthode Ndoc

2.3.2.1. Principe

Le logiciel Ndoc (ou Neurodoc) extrait d'un corpus documentaire un ensemble de thèmes organisés sur une carte globale de thèmes. Cet outil utilise une approche neuronale. En effet les premières études d'application des réseaux de neurones artificiels à l'infométrie ont abouti à définir la méthode de classification des k-means axial (**figure 6**). Définie par LELU en 1990, cette méthode s'inspire du formalisme neuronal des cartes auto-adaptatives de KOHONEN et permet de modéliser les connaissances apportées par un corpus documentaire. Neurodoc applique donc la méthode des k-means axial comme algorithme de classification non hiérarchique, et une Analyse en Composantes Principales (A.C.P.) pour la représentation des classes obtenues sur une carte (**figure 5**).

En clair chaque classe, ou thème, est représentée par un axe sur lequel se regroupent et s'ordonnent à la fois les documents et les mots-clés.

x1

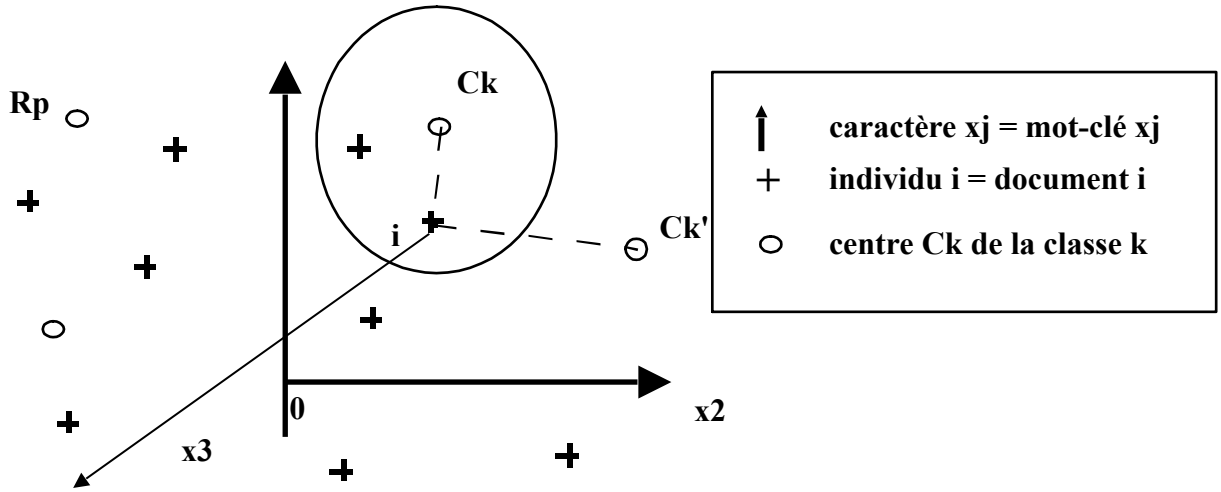


Figure 5 Principe des K.means.

Les thèmes correspondent à un type de classe particulier :

- ces classes sont recouvrantes car un document ou un mot-clé peut appartenir à plusieurs classes à la fois ;
- les éléments, documents et mots-clés de chaque classe, sont ordonnés selon un degré de ressemblance au type idéal de la classe.

Un thème est constitué des listes triées:

- des mots-clés,
- des documents,
- des auteurs,
- des sources associées à ce thème.

Les documents et mots-clés sont ordonnés selon la valeur de leur projection sur l'axe représentant ce thème. Plus la valeur de cette projection est élevée, plus le mot-clé ou le document est proche du type idéal du thème. Les auteurs et les sources sont extraits à partir des documents associés au thème, ils sont triés selon le même procédé que les documents.

Les thèmes sont situés les uns par rapport aux autres sur une carte globale constituée par une projection, sur un plan, des thèmes représentés dans l'espace des mots-clés ; pour cela, une A.C.P. recherche les directions d'allongement maximum d'un nuage de point représentant les classes pour permettre de déterminer un plan sur lequel tous ces points sont ensuite projetés orthogonalement.

2.3.2.2. Lecture d'une carte Ndoc

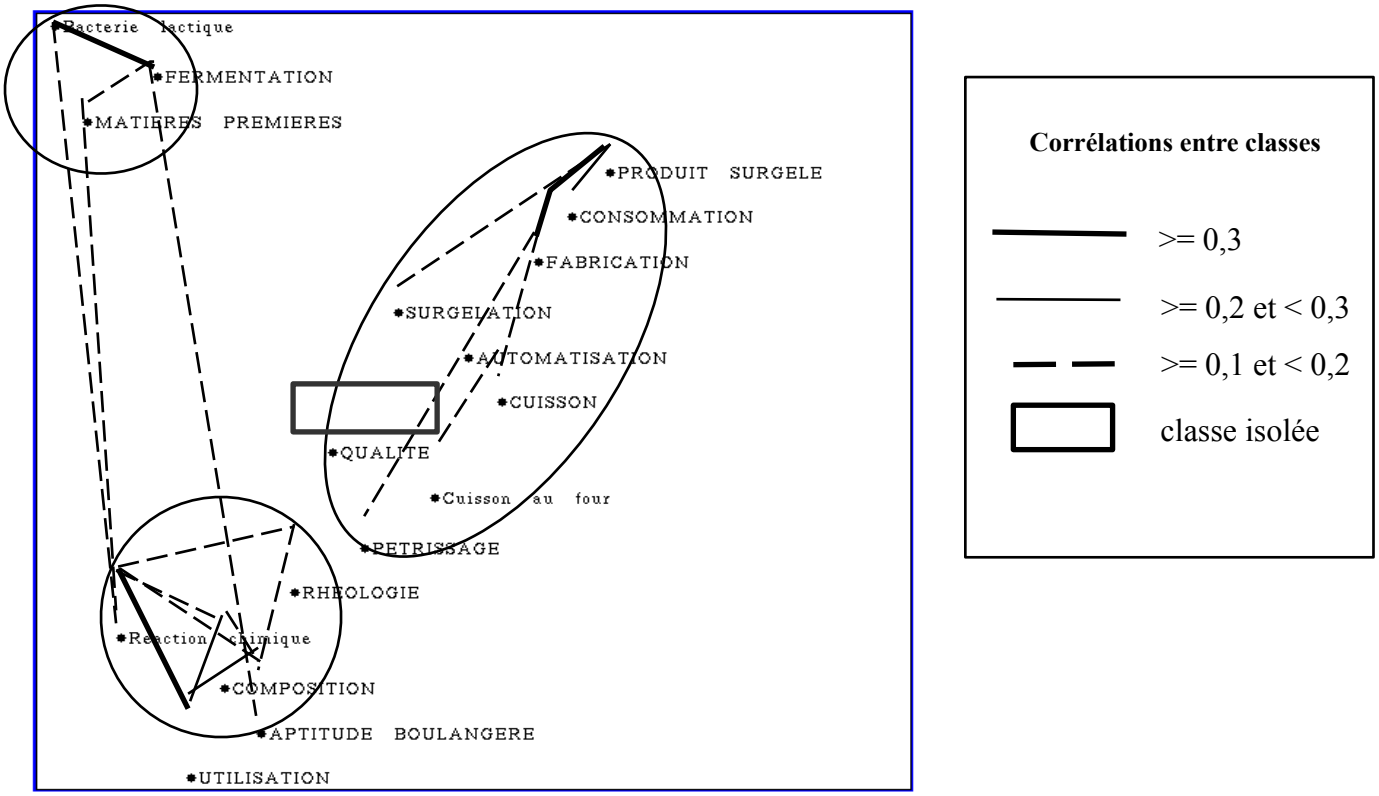


Figure 6 **Exemple d'école de la modélisation Ndoc sur un corpus traitant de la panification.**

2.3.3. La méthode Sdoc

2.3.3.1. Principe

Sdoc est une application s'appuyant sur la bibliothèque I.lib. Elle est issue de LEXIMAPPE mise au point au C.D.S.T. et à l'école des Mines de Paris

Une carte Sdoc est un réseau de groupe de mots, avec des relations entre ces mots. Un groupe de mot est appelé cluster, et représente un thème pour le domaine choisi.

Un indice statistique permet de mesurer la force associative de deux mots-clés (ils sont utilisés dans l'indexation des documents). Cet indice est fonction du nombre d'occurrences de chacun des termes et du nombre d'occurrences des deux termes dans le même document. L'ensemble des associations entre mots-clés forme un réseau valué d'associations. Afin de découper le réseau en clusters (groupement de mots avec des relations entre ces mots), l'algorithme de classification basé sur la méthode du simple lien, présenté dans le paragraphe suivant, est appliqué sur les relations entre mots-clés.

D’un point de vue sémantique, les études réalisées montrent que les clusters s’apparentent aux thèmes de recherche que l’on peut trouver dans un domaine scientifique. De plus, les clusters peuvent admettre des relations avec d’autres clusters, et à chaque cluster est associée une liste (triée par degré de pertinence) de références bibliographiques.

On obtient ainsi un réseau structuré et hiérarchisé de clusters (par rapport au réseau plat des associations entre mots).
Les clusters sont ensuite positionnés sur un plan bidimensionnel (Y, X) selon leur "densité" et "centralité", constituant ainsi une carte :

- la densité (Y) d'un cluster est exprimée par la valeur moyenne des associations entre mots-clés formant le cluster, ou associations internes ;
- la centralité (X) d'un cluster est exprimée par la valeur moyenne des associations entre les mots qui le constituent et les mots d'autres clusters, ou associations externes.

2.3.3.2. Algorithme de la méthode du ‘simple lien’ de Sdoc

La méthode appliquée est une extension de la méthode de classification hiérarchique dite du ‘simple lien’. La description qui suit est celle de la méthode utilisée mais lorsqu’il y a divergence entre ces deux méthodes les différences sont expliquées.

La première étape consiste à calculer une distance de base entre les éléments à classer ; cette distance correspond au coefficient d’association entre deux mots-clés.

Ce coefficient mesure la proximité des mots-clés en fonction de leur nombre d’apparitions communes dans les documents qu’ils indexent.

Ce coefficient calculé, la deuxième étape consiste à trier les couples de mots-clés par coefficient d’association décroissant.

Puis, pour construire les clusters on parcourt séquentiellement la liste de couples :

- au départ, chaque élément (mot-clé) est un cluster ;
- pour chaque association de la liste, on exécute l’algorithme suivant :
  - si les mots-clés du couple appartiennent au même cluster, le lien entre ces deux mots devient un lien interne au cluster (ce lien est une information interne au cluster qui met en évidence une relation privilégiée entre deux éléments de celui-ci)
  - si les mots-clés appartiennent à deux clusters différents, leur coefficient d’association représente la proximité entre les clusters à qui ils appartiennent.

Dans la méthode du ‘simple lien’, on doit fusionner les deux clusters pour n’en obtenir plus qu’un, mais ici on fait jouer des paramètres tels que le nombre maximum d’éléments (de mots-clés), de liens internes et de liens (à la fois internes et externes).

On ne fusionne deux clusters que si les différents critères de taille sont respectés.

Cette restriction par rapport à la méthode du ‘simple lien’ est nécessaire afin de ne pas obtenir de trop grand cluster.

Lorsque la fusion n'est pas possible, on crée un lien externe entre les deux clusters. Le choix de ne pas fusionner pose un problème lorsque le parcours de la liste nécessite à nouveau la fusion de l'un d'entre eux.

Il faut alors faire un choix :

- on sature les deux clusters à l'origine de l'échec de la fusion. Ceci empêche toute fusion future avec d'autres clusters plus petits et de proximité plus faible,
- on sature le plus gros des deux clusters : ce choix signifie que seul le plus gros des deux clusters a atteint un niveau caractéristique de complexité,
- on ne sature aucun des deux clusters, ce choix signifie la maximisation de la taille des clusters dans la limite des contraintes posées par les différents paramètres.

2.3.3.3. Carte Sdoc

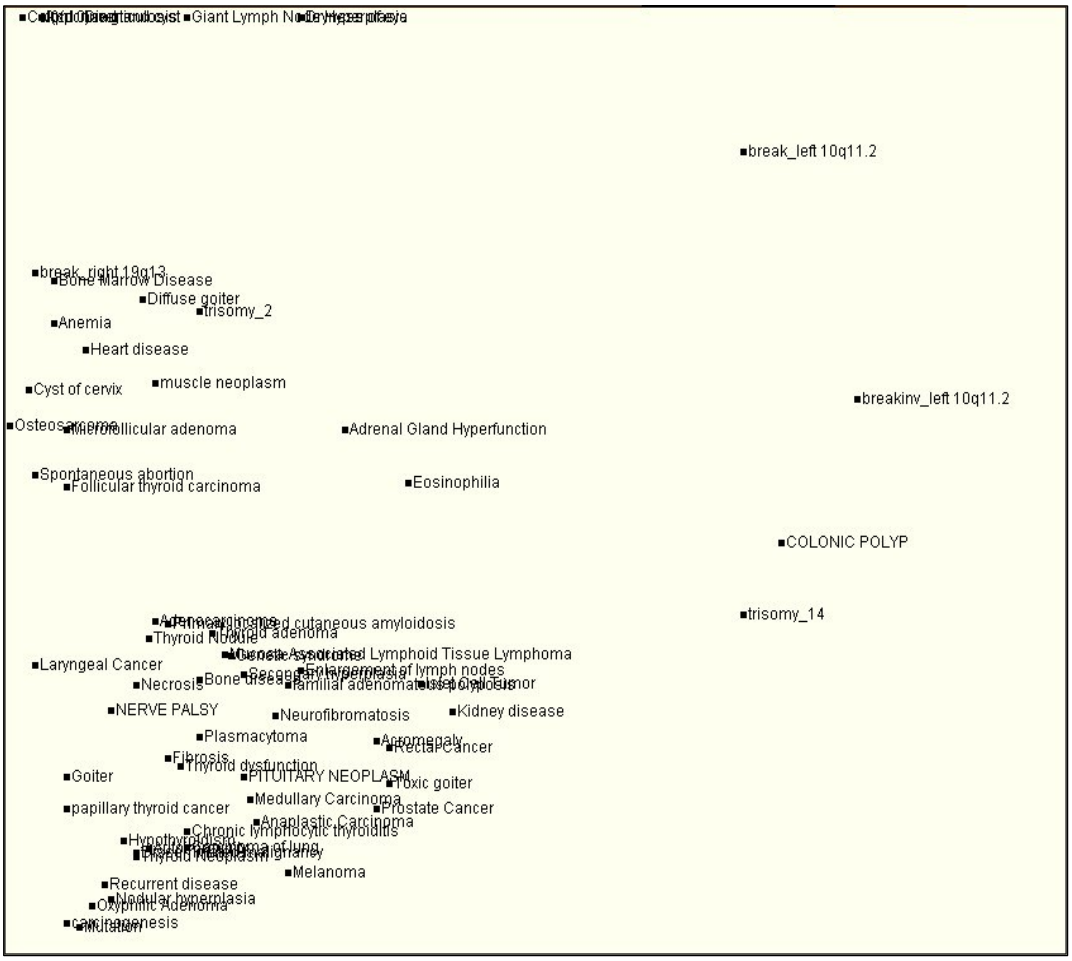


Image 1 Exemple de carte Sdoc.

2.4. Principes informatiques de base utilisés par l'U.R.I.

Sur le plan informatique, l'U.R.I. applique une approche modulaire implémentée sous UNIX grâce à des bibliothèques de fonctions programmées en C ou des API (Application Program Interface) C++ ou Java.
Pour définir les formats d'échange de données entre les modules, l'équipe emploie la norme internationale S.G.M.L. (Standard Generalized Mark-up Language, ISO 8879:1986), permettant une description de structures logiques dans des documents textuels ou multimédias. Ce méta-langage donnent des règles d'écriture de langages à balise dont s'inspire H.T.M.L. (HyperText Mark-up Language), célèbre depuis l'avènement du World Wide Web.
L'intérêt du système de marquage, s'exprime dans les structures arborescentes où les nœuds sont balisés et étiquetés (<X> et </X>).Une grande lisibilité est possible pour l'homme et la machine ainsi qu'une vérification, à l'aide d'un programme appelé parseur, de la conformité de documents vis à vis d'un modèle. La D.T.D (Définition de Type de Document) prend compte du nom, de la séquence, de la fréquence, et des attributs des éléments d'un document.

Le volume stocké nécessite un système de répartition des données dans plusieurs fichiers. Cette répartition est faite à l'aide d'une structure H.F.D. (Hierarchical File organization for Documentation)(**figure 7**).

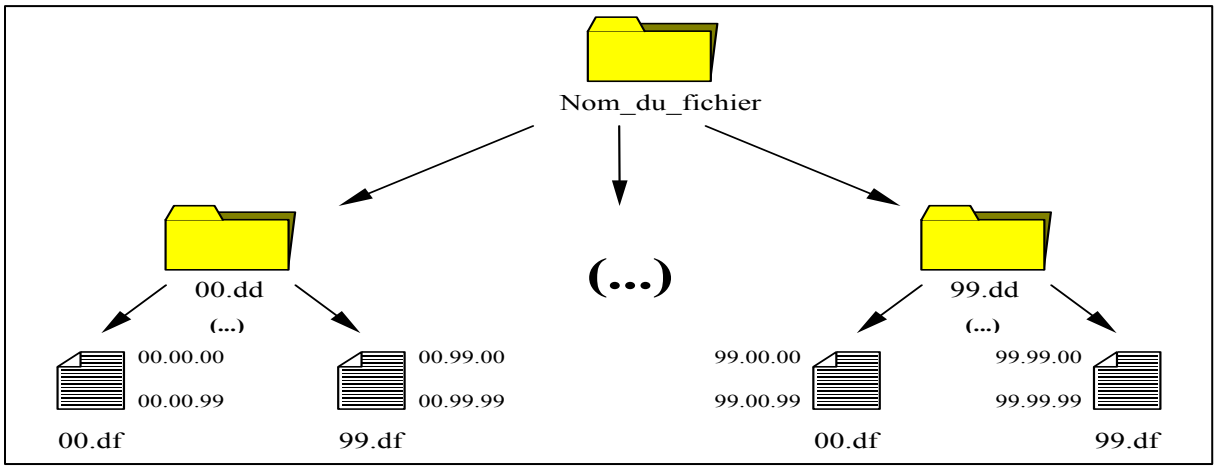


Figure 7 Organisation H.F.D. des fichiers.

Pour manipuler aisément ces structures et données, un ensemble de commandes et fonctions a été développé par l'équipe sous le nom générique de « I.lib » (I.N.I.S.T. Library). Ces outils souples d'utilisation, répondent à un principe de modularité, et s'enchaînent grâce au mécanisme de « pipe » d'Unix.

3. Analyse de l'existant : Visa, application sous-jacente à la plate-forme infométrique Stanalyst.

3.1. Problématique informatique

Une grande partie de mon travail à mon arrivée dans l'I.N.I.S.T., consista à assimiler l'environnement informatique dans lequel j'allais évoluer. Le projet sur lequel j'étais amené à travailler était déjà en cours de réalisation.

L'objectif était de me préparer à intégrer de nouvelles fonctionnalités au sein d'une interface graphique appelée Visa, accessible depuis un navigateur HTML. Une homogénéisation de l'ensemble des pages de cette interface était également envisagée.

Il s'agit donc dans ce paragraphe de décrire les choix informatiques retenus lorsque le projet I.G.R a commencé son développement:

- Décrire l'architecture logicielle de l'application Visa ;
- Présenter les fonctionnalités en place ;
- Développer une maquette permettant de mettre en évidence l'ergonomie de cette application.

L'articulation de ce paragraphe se fait autour des thèmes suivants :

- Architecture : une vue d'ensemble de l'application depuis son origine au sein de Stanalyst, jusqu'à ses aménagements particuliers.
- Accès aux différentes pages de l'application.

3.2. Choix de développement de la plate-forme

L'application Visa fait partie de la station Stanalyst mise au point par l'I.N.I.S.T.-C.N.R.S.. Cette station se compose d'un ensemble de modules relatifs à la recherche d'informations dans des bases de données documentaires.

Ces modules existent sous forme de scripts écrits en langage de shell Unix, et pour certains accompagnés de pages HTML générées dynamiquement par des scripts Perl écrits dans des C.G.I. (Common Gateway Interface).

Toutefois, cette station comporte un incovénient majeur en n'étant limitée qu'à la base Pascal/ Francis, elle n'est pas encore adaptée à une gestion multi-base nécessaire au projet I.G.R.. Les données à intégrer dans ce projet ont une origine supplémentaire : la base Medline.

La base Medline est une base de données bibliographique biomédicale créée par la National Library of Médicine (N.L.M.) dépendant de l'institut National de la santé américaine (National Institutes of Health). Cette base est consultable gratuitement sur Internet sur http://www.nlm.nih.gov/ .

Pour permettre cette plus grande souplesse d'utilisation, il fut décidé d'utiliser une version de Visa désolidarisée de Stanalyst. Mon travail devait se concentrer sur cette version, sans oublier que, malgré tout, mes modifications pouvaient être, à cours terme, réintégrées dans Stanalyst.

3.3. Matériel et logiciels

Le serveur HTML est localisé sur une station Unix, il s'agit d'un serveur Apache.

Les postes clients sont des PC ou Mac équipés des navigateurs HTML classiques (Netscape, Internet Explorer).

Les contraintes à respecter sont les suivantes :

- Utilisation du serveur Apache de l'I.N.I.S.T.-C.N.R.S. (**cf. « fichier de configuration du serveur Apache » en annexe**),
- Développement sous forme de scripts écrits en shell Unix et langage Perl, pour rester cohérent avec l'existant,
- Visualisation sous des navigateurs HTML de type Netscape et Internet Explorer : navigateurs utilisés couramment à l'I.N.I.S.T.-C.N.R.S., et faire-part de mises à jour de ces logiciels pouvant apporter un plus à l'application développée.

3.4. Architecture de l'application Visa

3.4.1. Principe architectural de la station Stanalyst et positionnement de l'interface Visa

Les différents modules auxquels est rattaché Stanalyst, sont organisés de la façon suivante:

- Les modules appelés et les données de travail sont localisés sur le serveur.
- La liste des utilisateurs est mémorisée sur le serveur.
- L'utilisateur déclenche les traitements et visualise les résultats sur le navigateur HTML de son poste client.

L'accès à la station est conditionné à l'identification de l'utilisateur (saisie d'un nom et d'un mot de passe).

Il faut bien distinguer ici les pages accessibles à un utilisateur, et les traitements sous-jacents. Chaque page a une fonctionnalité bien précise et s'appuie sur un ou plusieurs outils existants :

- Le module "Corpus" gère la création de corpus, à partir de l'outil Miriad, ce qui permet la génération de corpus par exécution de requêtes construites par l'utilisateur. Les corpus peuvent ensuite être exportés à destination des modules Indexation et Infométrie.

- Le module "Bibliométrie" gère la création d'analyses statistiques, à partir de Miriad également pour la création des profils d'analyse.
- Le module "Indexation" permet d'indexer automatiquement un corpus, en s'appuyant sur les outils I.L.C. d'indexation, et Visa (pour la visualisation de résultats).
- Le module "Infométrie" gère la classification à partir des outils Sdoc, Neurodoc (traitements) et Visa (visualisation).

Tous ces modules, suivants les blocs d'exploitation de données de l'U.R.I., sont utilisables indépendamment les uns des autres, hors station Stanalyst, par l'intermédiaire de programmes BATCH.

Les modules utilisent un ensemble de répertoires de travail contenant les programmes, scripts, paramétrages nécessaires à son fonctionnement, ainsi que l'ensemble des projets créés par les utilisateurs.

- L'ensemble de répertoires contenant des programmes, lexiques, librairies, sources, données et paramétrage est enrichi, et intègre la gestion d'utilisateurs commune à l'ensemble des modules.
- Le principe utilisé est étendu aux autres modules.

Les schémas des **figures 8 et 9** résument l'architecture de la plate-forme.

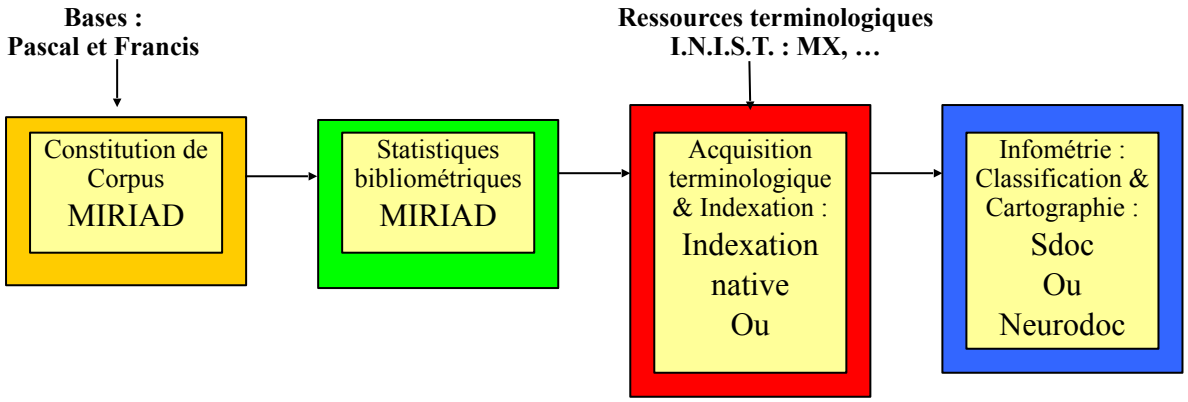


Figure 8 **Architecture du système d'information de l'U.R.I..**

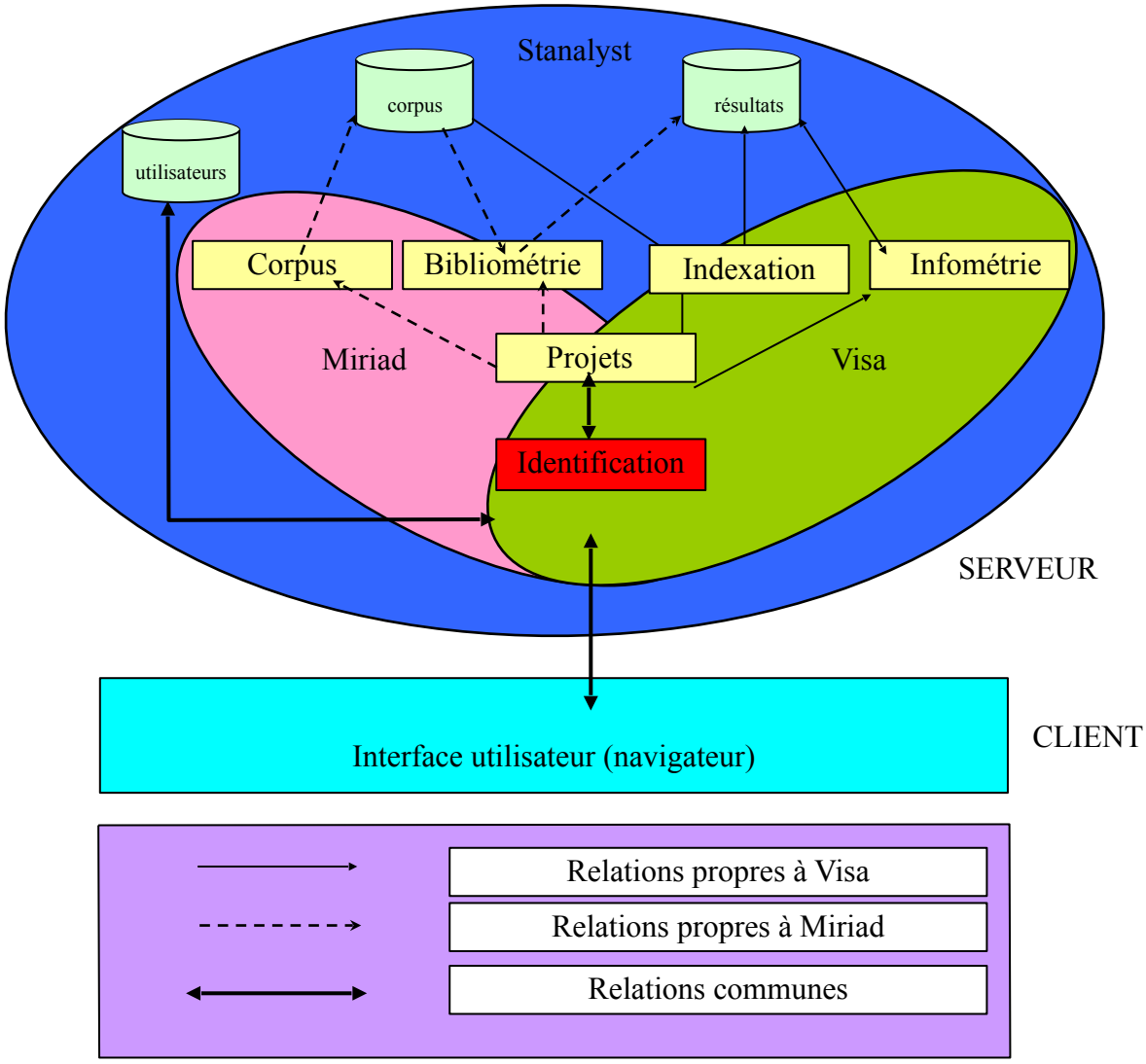


Figure 9 **Les relations client/serveur et interfaces de visualisation des données propres à l'U.R.I..**

Sur Stanalyst, un utilisateur s'identifie par un nom et un mot de passe : la liste des projets utilisables comporte les projets dont il est propriétaire et les projets pour lesquels il bénéficie d'une autorisation d'accès.
Tout utilisateur recensé peut créer et supprimer des projets (On ne peut supprimer que ses propres projets).

3.4.2. L'interface Visa actuelle désolidarisée de Stanalyst

Visa permet, actuellement, la visualisation des résultats du traitement de l'information, et fonctionne relativement indépendamment :

- L'outil possède sa propre interface HTML, utilisable depuis le serveur Intranet de l'I.N.I.S.T.-C.N.R.S. qui implémente également une gestion d'utilisateurs et met ainsi en évidence la notion de répertoire public ou privé.
- Les divers outils infométriques s'activent en amont à partir de commandes shell Unix, couplées à des fichiers de paramétrage : il s'agit de fichiers ASCII lus à l'exécution de chaque commande. L'exécution des traitements génère des fichiers de trace (log, err, res) mais aussi des fichiers de données exploitables directement par Visa : il s'agit alors des données indexées et/ou classifiées, structurées en SGML.

L'outil Visa permet donc la consultation, par un navigateur HTML, des résultats obtenus par les outils I.L.C. et Infométrie (**figure 10**).

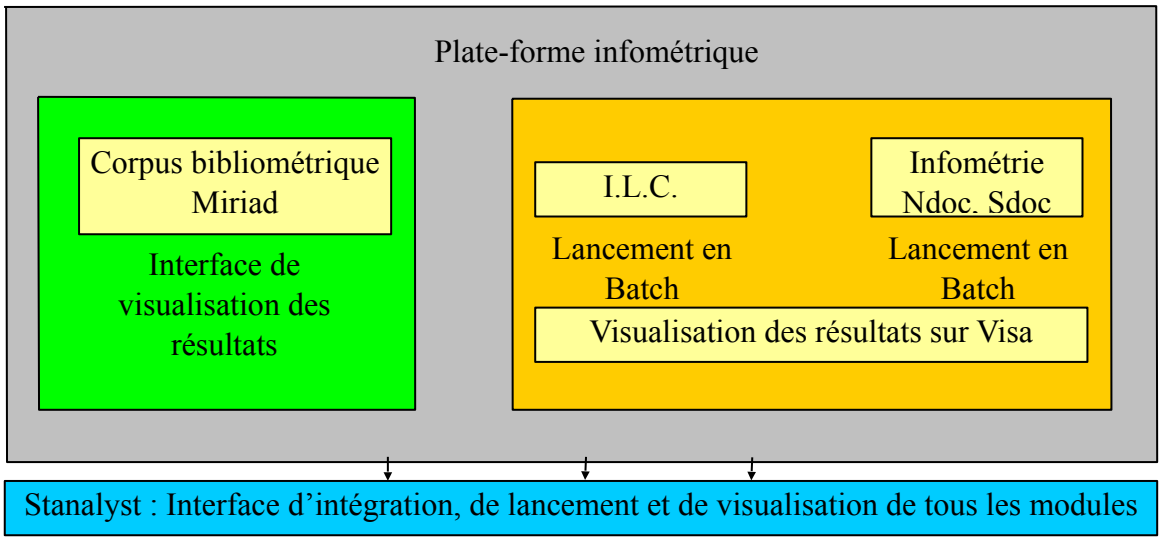


Figure 10 Place de Visa dans la plate-forme infométrique.

On devra mettre en place une interface graphique centralisant mieux l'ensemble des traitements accessibles à l'utilisateur de façon conviviale, et rapide.

Dans le but de gérer le transfert de données d'un module à l'autre, mais aussi d'étendre la notion de répertoire privé à tous les modules, l'organisation des répertoires sur le serveur mis en place fut celle décrite par la **figure 11**.

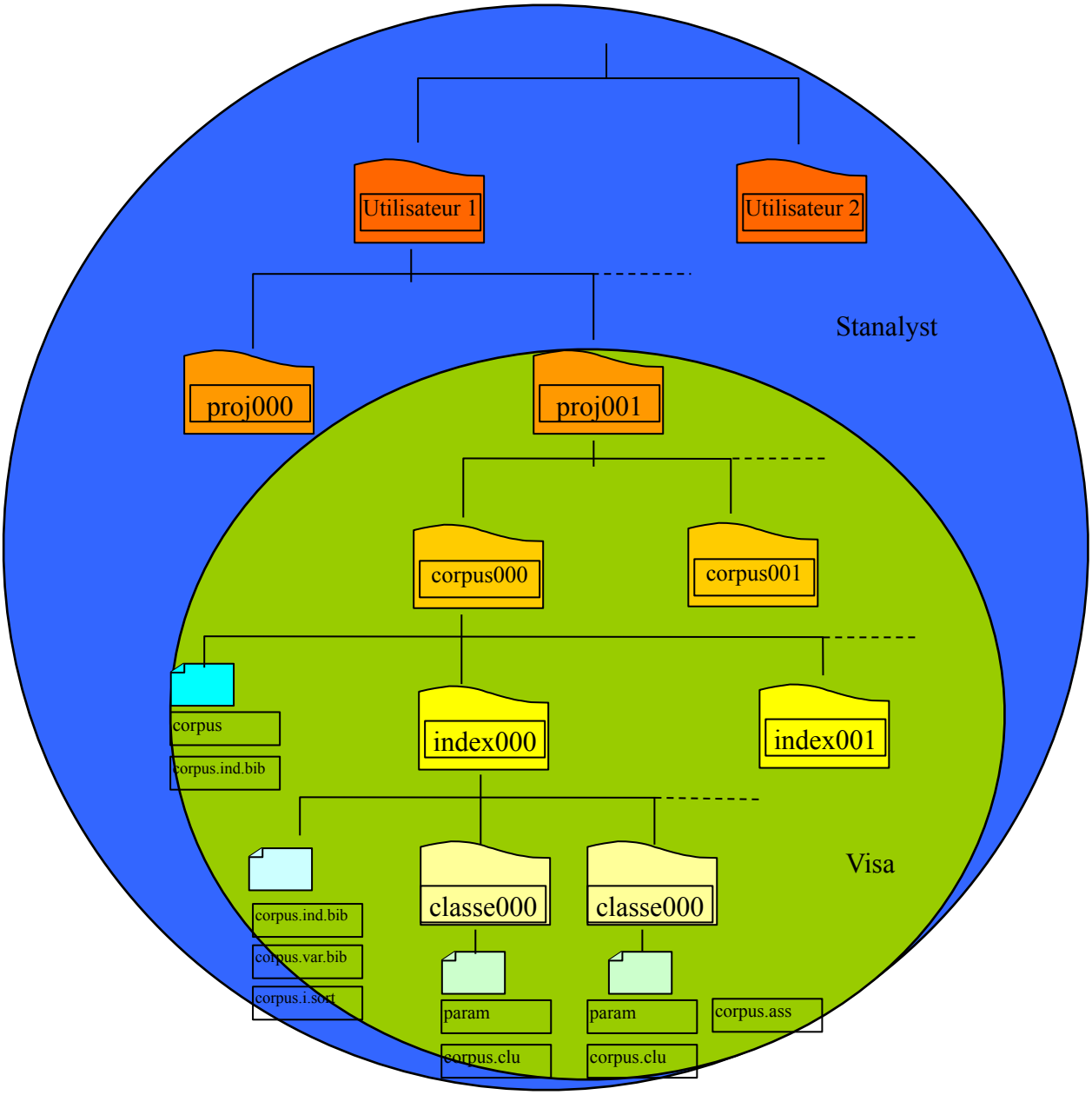


Figure 11 **Organisation des répertoires du serveur de l'U.R.I., organisations propres à Stanalyst et Visa (dans le cadre du projet I.G.R.).**

Au sein d'un même projet, sur Stanalyst, on peut effectuer le transfert de corpus en recopiant les fichiers concernés dans le bon sous-répertoire.

Visa n'utilise qu'un seul projet : celui de l'I.G.R. sur le cancer de la thyroïde.

En conséquence, l'interface Visa étudiée ne prend en compte qu'une seule branche de l'arborescence des projets. La **figure 11** décrit cette restriction en idéalisant la situation. Le projet I.G.R. n'étant pas finalisé, et sujet à de perpétuels changements, les répertoires ne

suivent pas une disposition arborescente équilibrée. Cette optimisation sera réalisée lors de la dernière phase du projet, après nettoyage des répertoires.

Dans l'immédiat, le déplacement sur cet arbre n'est pas facilité, et nécessite la définition de nombreux chemins de recherche dans les fichiers .clu, pour les informations concernant les classes (ou clusters), et .ass, pour celles portant sur les associations.

3.4.3. Les contraintes de l'application et de son développement: la concurrence des traitements, sa gestion sur Stanalyst et comparaison avec le projet I.G.R..

Les modules de Stanalyst sont tous amenés à exécuter des scripts sur le serveur. Actuellement, Miriad utilise une file d'attente, gérée par un démon, pour éviter de bloquer l'interface utilisateur.

Stanalyst reprend également ce principe de la file d'attente en l'étendant aux modules Indexation et Infométrie.

Les différentes demandes sont gérées par une file unique, et la présence des informations nécessaires à chaque opération est contrôlée à chaque étape de la demande (paramétrage, confirmation…).

Par cette façon de procéder, l'application respecte les contraintes suivantes :

- Ne pas surcharger la machine sur laquelle sont exécutés les traitements.
- Respecter la cohérence des arborescences de répertoires en évitant des traitements parallèles massifs. Il est impossible de lancer une analyse statistique en même temps qu'une requête sur le même répertoire, requête qui détruirait alors les résultats d'analyse au fur et à mesure de leur création.
- L'interface interdit le lancement d'un traitement dès lors qu'un autre lui a déjà été demandé pour le même répertoire, ou l'un de ces parents.

Tout ceci nécessite le couplage des lancements du serveur et du démon, ce dernier utilisant le fichier .jobs situé à la racine du serveur, et contenant les informations pour chaque traitement en attente :

- Le numéro d'ordre du traitement (0 en cas d'exécution en cours, l'utilisateur demandeur).
- Le type de traitement : requête, analyse, indexation, classification.
- Le répertoire concerné, identifié par son nom physique.

A chaque demande d'exécution, l'application Stanalyst ajoute une ligne dans le fichier.jobs et réveille le démon. Celui-ci lit alors le fichier .jobs et tente de traiter toutes les demandes qu'il contient. Une fois la tâche accomplie, le démon se rendort.

Le démon se charge de l'exécution des scripts liés aux traitements, l'application Stanalyst se contentant de surveiller la présence de résultats dans le répertoire courant, en rafraîchissant régulièrement la page d'accès de chaque module.

Dans le cas du projet I.G.R, cette configuration n'est plus respectée : l'interface Visa est indépendante, elle ne prend en compte, comme il a été précisé précédemment, qu'une partie fixe de l'arborescence des fichiers, et ne possède qu' « un seul » utilisateur (au sens informatique: un seul jeu de nom et de mot de passe).

Plusieurs personnes travaillant sur la même base informatique, il a fallu trouver une solution pour me permettre de développer de nouvelles applications sans gêner le fonctionnement des travaux annexes.

L'option retenue fut de cloner le serveur de Visa dans un autre emplacement de la machine l'hébergeant (une machine Unix Sun appelée Yoda), puis de relancer le serveur Apache, après modification des fichiers de configuration (**cf. « fichier de configuration du serveur Apache » en annexe**), sur un nouveau port, rendant le tout accessible par intranet.

Les fichiers exploités de données se retrouvent figés, et ne bénéficient pas des mises à jour des résultats des travaux d'indexation, d'où un certain nombre de problèmes lors du passage des applications du prototype au produit fini. Des caractères dans les mots clés des nouveaux jeux de données comme « ‘ » ont généré des erreurs inattendues dans les CGI, et ont dû être récupérées.

Mais globalement, cette configuration indépendante m'a procuré un confort de développement certain, sans entrer en conflit avec d'autres utilisateurs.

3.4.4. Principe de l'interface Visa

Depuis Stanalyst, un lien permet de diriger l'utilisateur vers l'application : l'accès se fait au moyen d'une page d'accueil. Le choix d'un répertoire permet d'accéder à la page d'accès qui constitue donc la page principale de tous modules sélectionnés, pour un répertoire donné : on y retrouve les informations relatives au répertoire, des commandes de gestion (habilitations, déchargement etc.…) et des commandes spécifiques au module (requêtes, analyse statistique…). Dans le cas des modules Indexation et Infométrie, il faut bien sûr choisir d'abord un répertoire de corpus et/ou d'indexation avant de choisir le répertoire de travail.

L'accès à Visa est standard, avec une page HTML statique inspirée du principe de Stanalyst. Mais dans le cas présent des restrictions ont été appliquées à l'interface, afin de remplir les contraintes propres à une analyse prédéfinie (**figure 12 et Image 2**). Les accès aux options de créations, de suppressions, et de choix d'une analyse ont été rendus inactifs.

Dans cette page, on ne trouve pas comme pour Stanalyst de bouton pour s'identifier sous un nouveau nom d'utilisateur.

Les parties visibles sont :

- Une zone de titre "Traitements linguistiques et analyse thématique de données textuelles".
- Trois boutons permettant les actions suivantes :
  - "Choix d'une analyse" pour le choix d'un répertoire.

- "Création d'une analyse" pour la création d'un répertoire (rendu non fonctionnel pour les besoins du projet).
- "Suppression d'une analyse"pour la suppression de répertoires (rendu non fonctionnel pour les besoins du projet).

- Une zone d'identification de l'utilisateur, caractérisée par des champs de saisie pour le nom et le mot de passe, est lancée automatiquement lors de la connexion. (Ce qui peut être contourné par l'entrée de l'URL : http://IGR:bioinfo@'Nom :Code d'accès'@'nom de la machine serveur':'port utilisé'/.

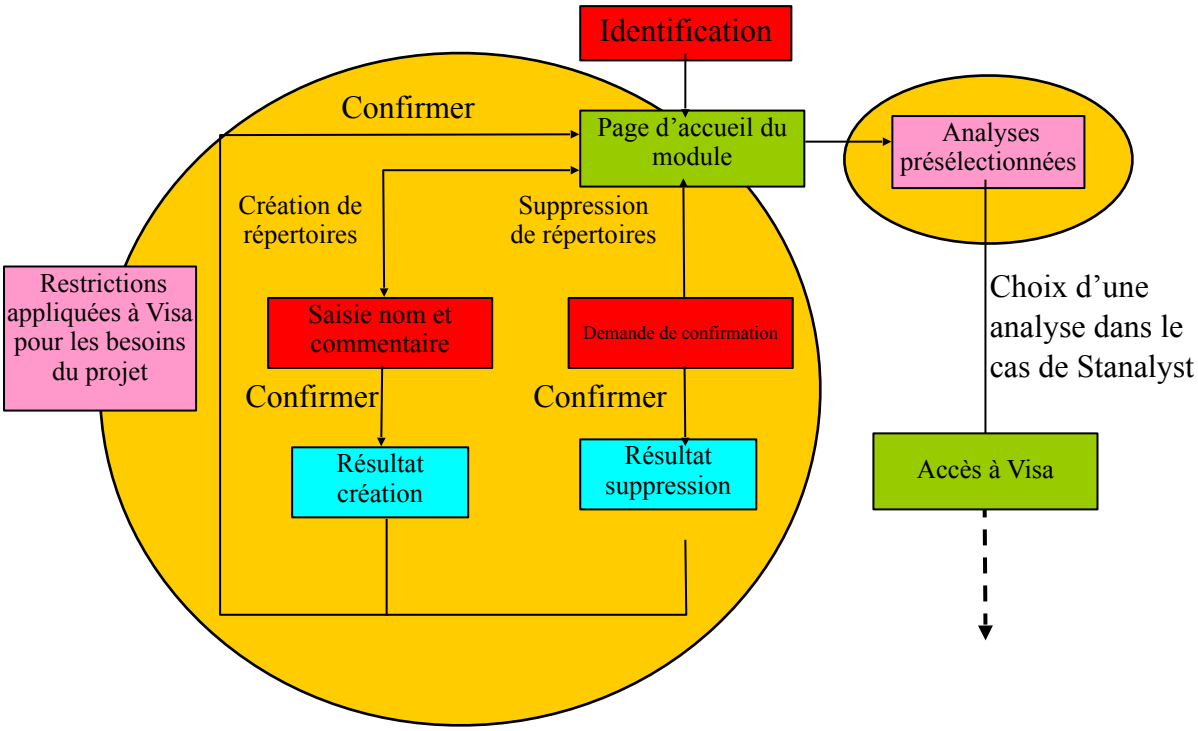


Figure 12 Accès standard aux modules d'information de l'U.R.I., et restrictions appliquées à Visa.

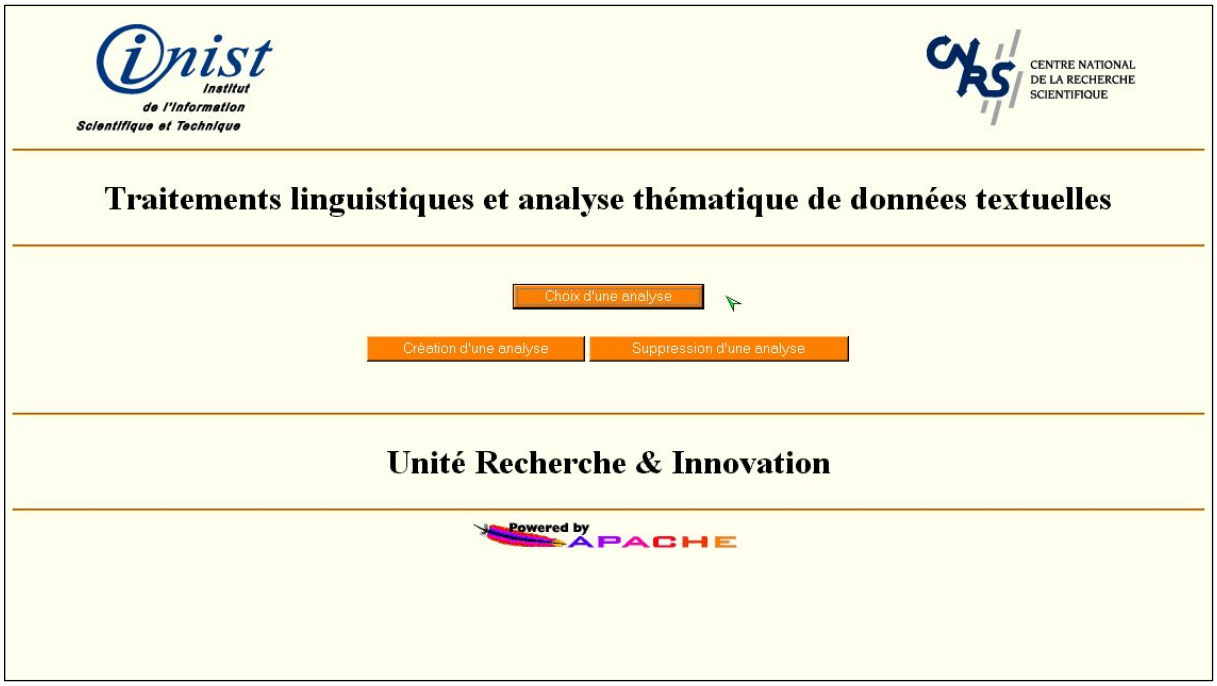


Image 2 Aperçu de la page d'accueil.

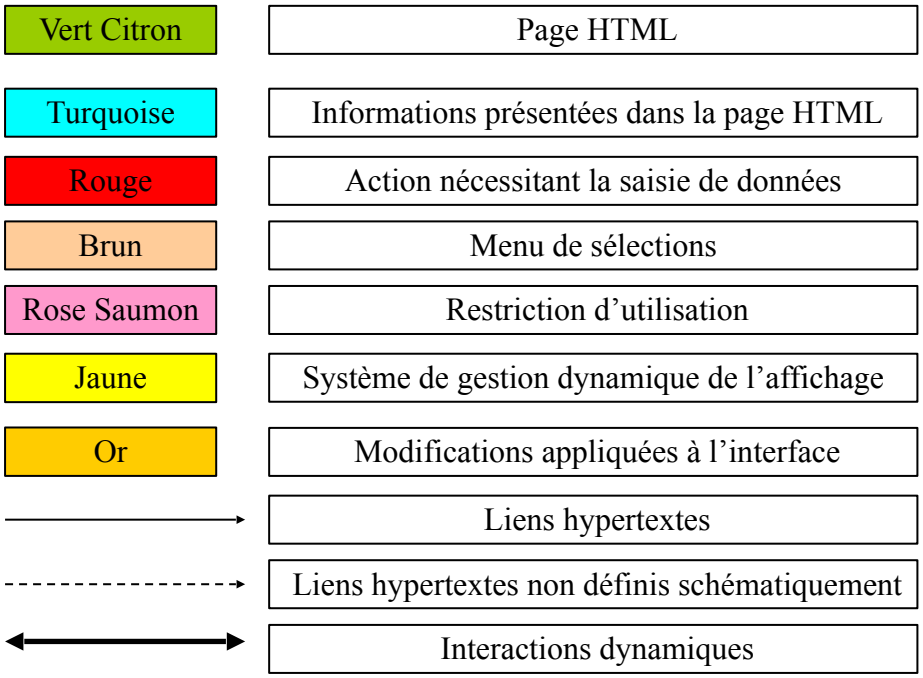
De plus, les pages HTML de l'application contiennent différents menus regroupant des fonctions disponibles à tout moment :

- Il est constitué d'un ensemble d'icônes, de titres et de boutons, le but étant de minimiser leur place occupée à l'écran, pour en réserver suffisamment pour la visualisation des données.
- Le contenu, et les nouvelles fonctionnalités doivent pouvoir évoluer en fonction du contexte : certaines opérations peuvent être effectuées à tout moment, d'autres non, ce qui est dû en partie aux différences d'analyse entre Ndoc et Sdoc (cf paragraphes concernant Ndoc et Sdoc).
- Le choix disponible en permanence est le retour à la page d'accueil.
- Les accès à différents liens doivent être disponibles dès que l'utilisateur a sélectionné une classe ou un mot-clé. La sélection d'un projet est suivie de la page d'accueil au module Corpus.

3.4.5. La navigation dans Visa

La **Figure 13** décrit l'enchaînement des pages dans Visa avant la réalisation des travaux de modifications.

Légende des schémas de navigation



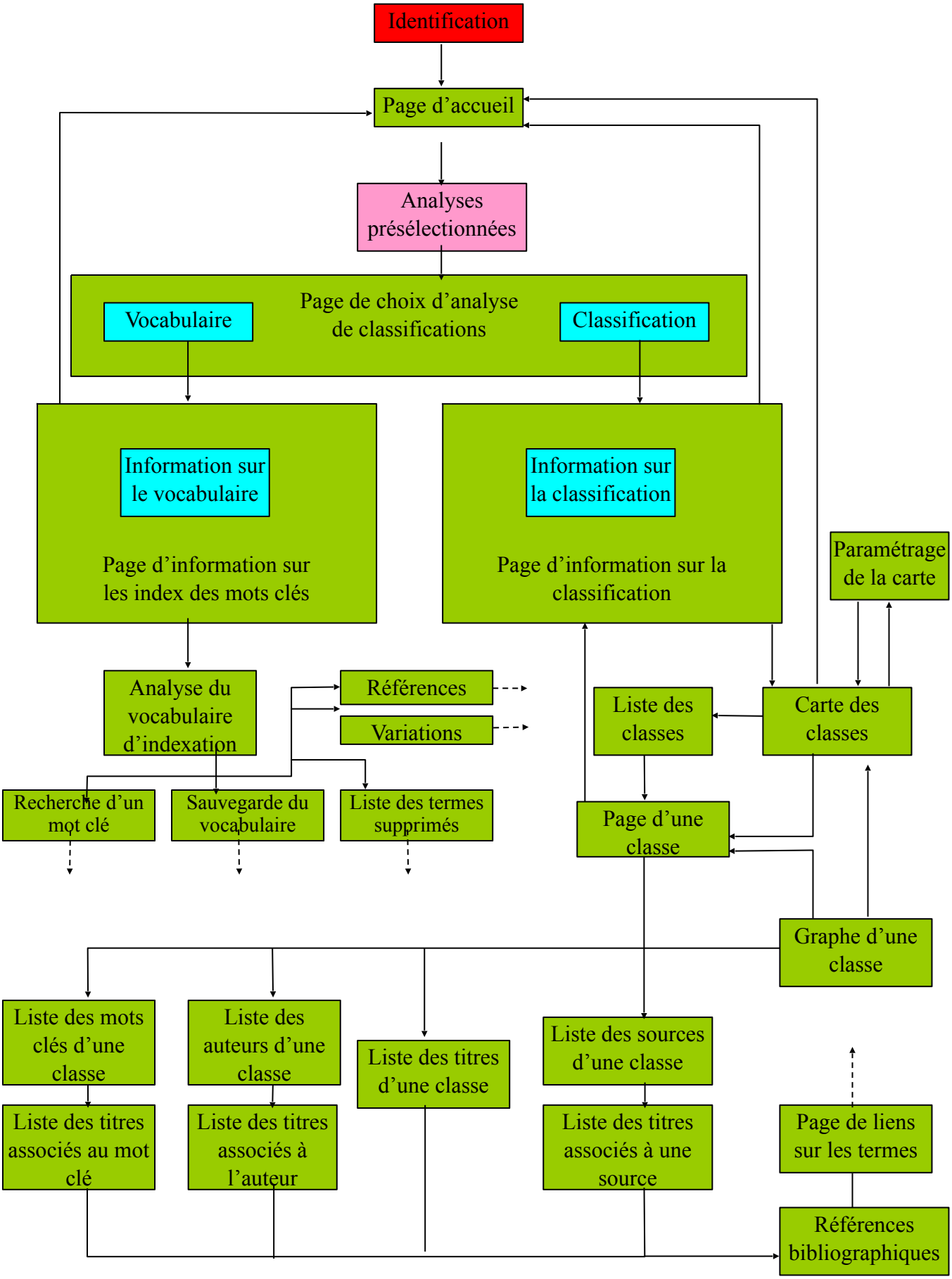
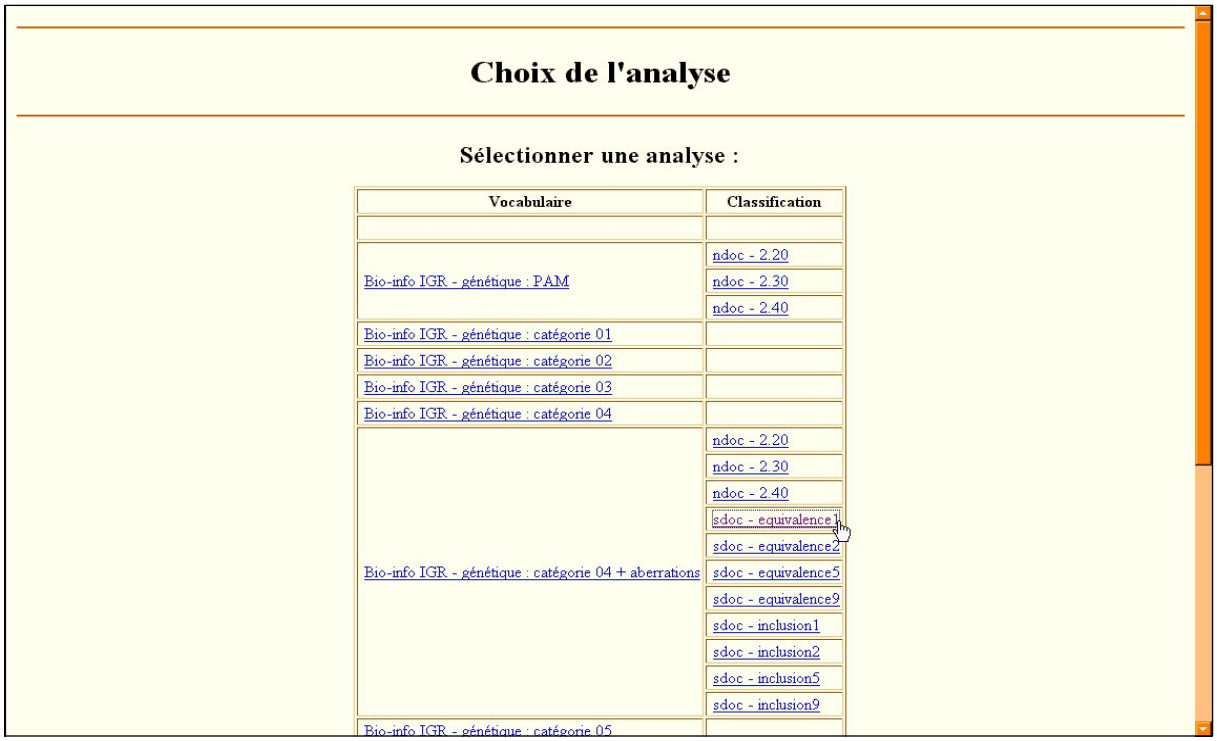


Figure 13 **Schéma de navigation à l'intérieur du module Visa.**

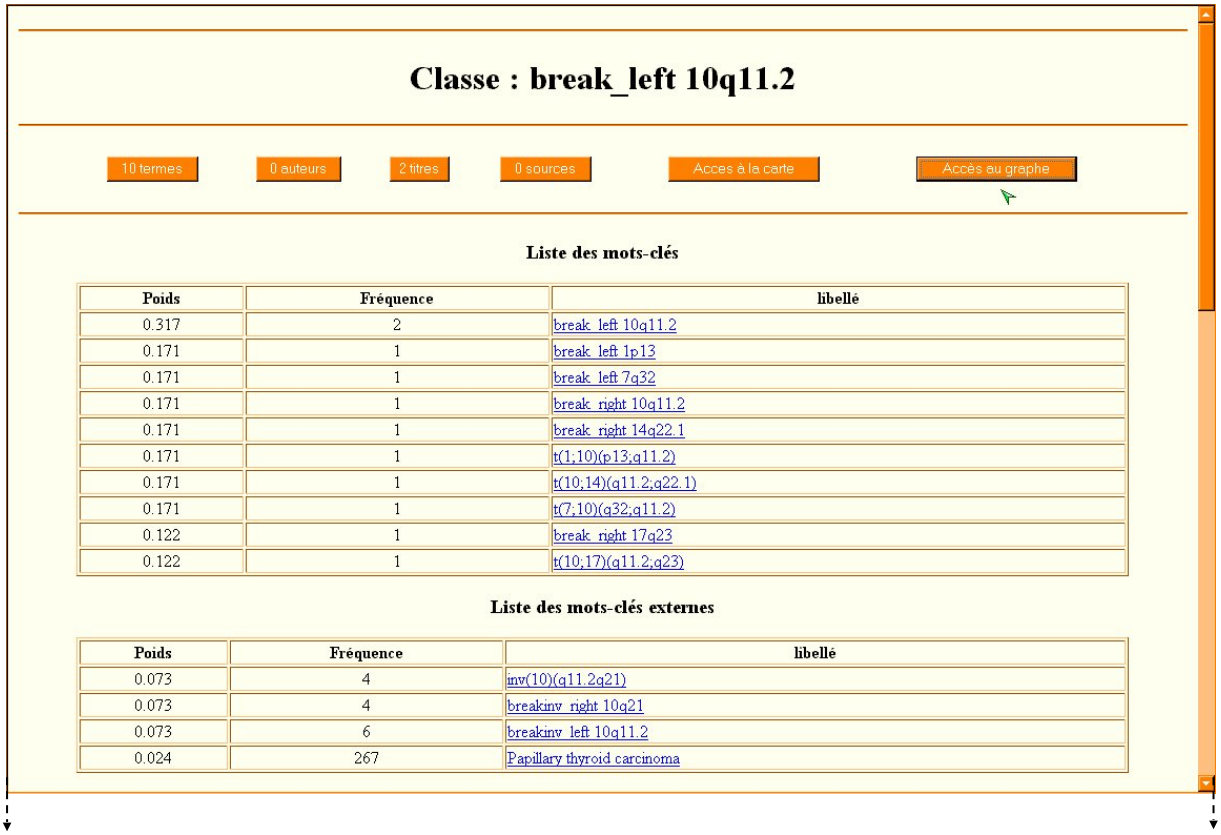


Choix de l'analyse

Sélectionner une analyse :

Vocabulaire	Classification
Bio-info IGR - génétique : PAM	ndoc - 2.20
	ndoc - 2.30
	ndoc - 2.40
Bio-info IGR - génétique : catégorie 01	
Bio-info IGR - génétique : catégorie 02	
Bio-info IGR - génétique : catégorie 03	
Bio-info IGR - génétique : catégorie 04	
Bio-info IGR - génétique : catégorie 04 + aberrations	ndoc - 2.20
	ndoc - 2.30
	ndoc - 2.40
	sdoc - equivalence1
	sdoc - equivalence2
	sdoc - equivalence5
	sdoc - equivalence9
	sdoc - inclusion1
	sdoc - inclusion2
	sdoc - inclusion5
	sdoc - inclusion9
Bio-info IGR - génétique : catégorie 05	

Image 3 Page de choix d'analyse de classifications.



Classe : break_left 10q11.2

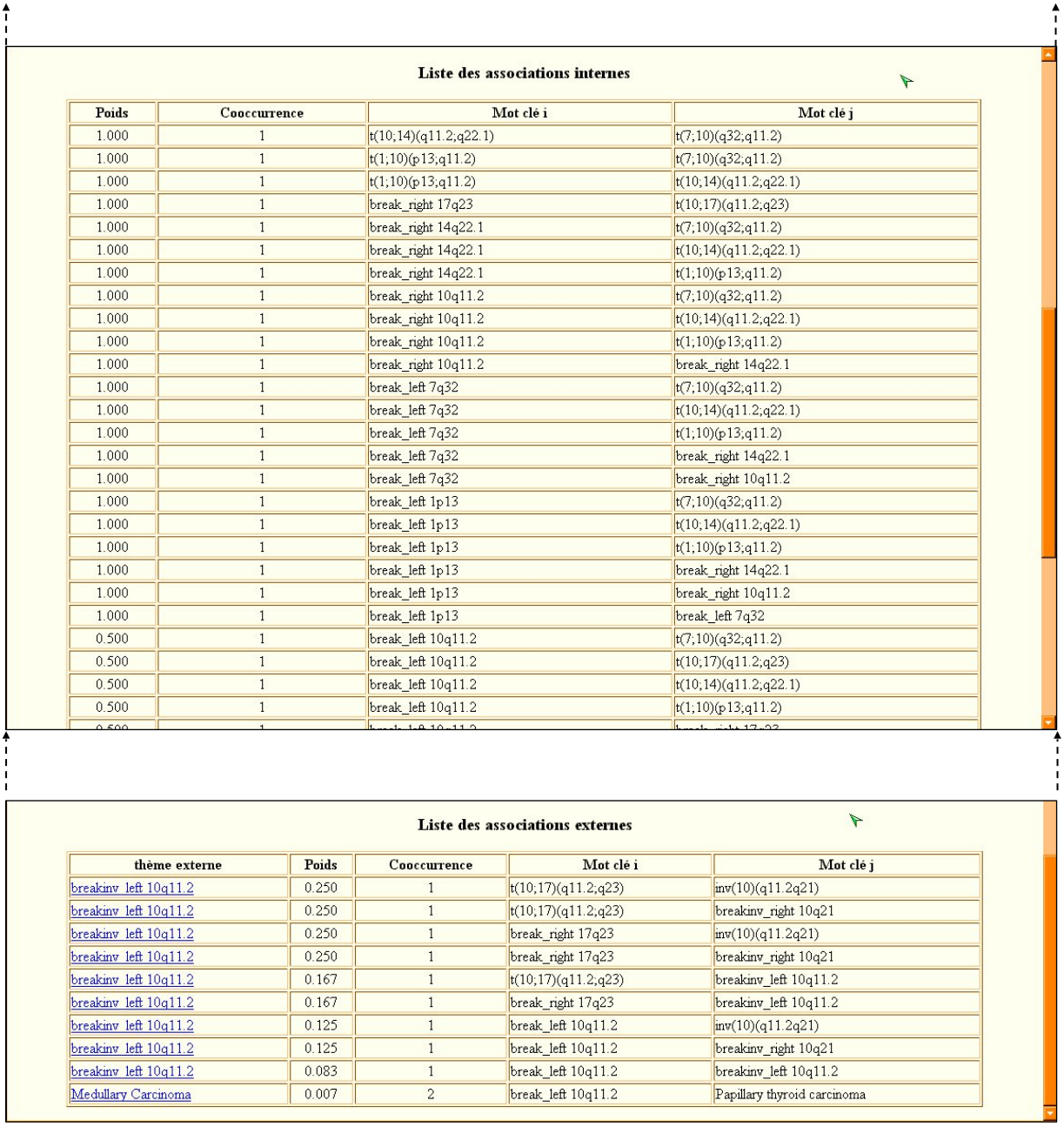
10 termes | 0 auteurs | 2 titres | 0 sources | Acces à la carte | Accès au graphe

Liste des mots-clés

Poids	Fréquence	libellé
0.317	2	break_left 10q11.2
0.171	1	break_left 1p13
0.171	1	break_left 7q32
0.171	1	break_right 10q11.2
0.171	1	break_right 14q22.1
0.171	1	t(1;10)(p13;q11.2)
0.171	1	t(10;14)(q11.2;q22.1)
0.171	1	t(7;10)(q32;q11.2)
0.122	1	break_right 17q23
0.122	1	t(10;17)(q11.2;q23)

Liste des mots-clés externes

Poids	Fréquence	libellé
0.073	4	inv(10)(q11.2q21)
0.073	4	breakinv_right 10q21
0.073	6	breakinv_left 10q11.2
0.024	267	Papillary thyroid carcinoma



Liste des associations internes

Poids	Cooccurrence	Mot clé i	Mot clé j
1.000	1	t(10;14)(q11.2;q22.1)	t(7;10)(q32;q11.2)
1.000	1	t(1;10)(p13;q11.2)	t(7;10)(q32;q11.2)
1.000	1	t(1;10)(p13;q11.2)	t(10;14)(q11.2;q22.1)
1.000	1	break_right 17q23	t(10;17)(q11.2;q23)
1.000	1	break_right 14q22.1	t(7;10)(q32;q11.2)
1.000	1	break_right 14q22.1	t(10;14)(q11.2;q22.1)
1.000	1	break_right 14q22.1	t(1;10)(p13;q11.2)
1.000	1	break_right 10q11.2	t(7;10)(q32;q11.2)
1.000	1	break_right 10q11.2	t(10;14)(q11.2;q22.1)
1.000	1	break_right 10q11.2	t(1;10)(p13;q11.2)
1.000	1	break_right 10q11.2	break_right 14q22.1
1.000	1	break_left 7q32	t(7;10)(q32;q11.2)
1.000	1	break_left 7q32	t(10;14)(q11.2;q22.1)
1.000	1	break_left 7q32	t(1;10)(p13;q11.2)
1.000	1	break_left 7q32	break_right 14q22.1
1.000	1	break_left 7q32	break_right 10q11.2
1.000	1	break_left 1p13	t(7;10)(q32;q11.2)
1.000	1	break_left 1p13	t(10;14)(q11.2;q22.1)
1.000	1	break_left 1p13	t(1;10)(p13;q11.2)
1.000	1	break_left 1p13	break_right 14q22.1
1.000	1	break_left 1p13	break_right 10q11.2
1.000	1	break_left 1p13	break_left 7q32
0.500	1	break_left 10q11.2	t(7;10)(q32;q11.2)
0.500	1	break_left 10q11.2	t(10;17)(q11.2;q23)
0.500	1	break_left 10q11.2	t(10;14)(q11.2;q22.1)
0.500	1	break_left 10q11.2	t(1;10)(p13;q11.2)

Liste des associations externes

thème externe	Poids	Cooccurrence	Mot clé i	Mot clé j
breakinv_left 10q11.2	0.250	1	t(10;17)(q11.2;q23)	inv(10)(q11.2q21)
breakinv_left 10q11.2	0.250	1	t(10;17)(q11.2;q23)	breakinv_right 10q21
breakinv_left 10q11.2	0.250	1	break_right 17q23	inv(10)(q11.2q21)
breakinv_left 10q11.2	0.250	1	break_right 17q23	breakinv_right 10q21
breakinv_left 10q11.2	0.167	1	t(10;17)(q11.2;q23)	breakinv_left 10q11.2
breakinv_left 10q11.2	0.167	1	break_right 17q23	breakinv_left 10q11.2
breakinv_left 10q11.2	0.125	1	break_left 10q11.2	inv(10)(q11.2q21)
breakinv_left 10q11.2	0.125	1	break_left 10q11.2	breakinv_right 10q21
breakinv_left 10q11.2	0.083	1	break_left 10q11.2	breakinv_left 10q11.2
Medullary Carcinoma	0.007	2	break_left 10q11.2	Papillary thyroid carcinoma

Image 4 Page d'une classe.

4. La modélisation des données et l'accès à l'information: les améliorations apportées à Visa.

4.1. Cahier des charges des nouvelles fonctionnalités à intégrer dans Visa

Les modifications de l'interface Visa ont pour principaux buts d'améliorer la convivialité, la visibilité et l'accessibilité des informations véhiculées. Elles devront répondre aux demandes suivantes :

- Les principes de visualisation de l'information énoncés dans Sdoc et Ndoc (cf. chapitre 2), devront être suivis, et retranscrits fidèlement du point de vue graphique.
- Les cartes obtenues seront paramétrables facilement, avec des modifications visibles instantanément sans l'usage de pages HTML intermédiaires, comme cela était le cas dans la version précédente de Visa.
- Dans le cas de Sdoc les liens pourront être affichés à volonté de façon globale ou particulière, et paramétrables selon le même principe cité ci-dessus. En aucun cas, il ne doit y avoir de rupture franche aussi bien du point de vue graphique que de celui du code, entre les deux représentations Sdoc et Ndoc (le module de visualisation doit être commun).
- Les graphes de clusters visibles dans Sdoc devront faire apparaître un plus grand nombre d'informations comme la fréquence des mots-clés, la cooccurrence des liens internes mais aussi externes, et donc, en conséquence, les mots-clés externes avec le nom de leurs clusters respectifs.
- Un dispositif d'accès à un panel de liens concernant les informations sur tous les mots-clés et toutes les classes, doit être mis en place pour permettre une navigation facile, et une recherche rapide de données.

La **Figure 14** décrit l'enchaînement des pages dans Visa après la réalisation des travaux de modifications.

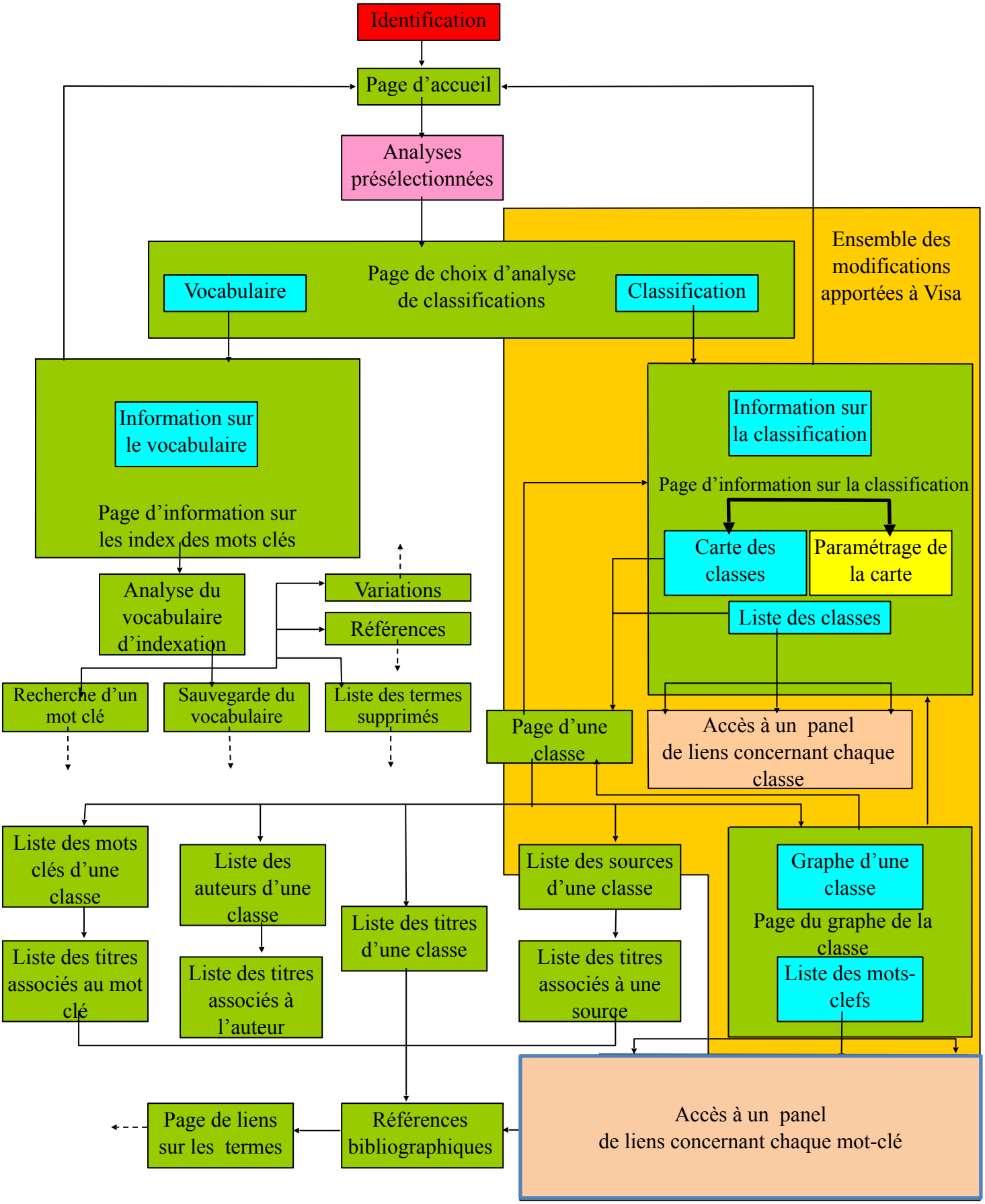


Figure 14 **Nouveau schéma de navigation souhaité à l'intérieur du module Visa.**

Du point de vue technique :

- Il serait préférable que le code soit concis, mais surtout transportable pour s'adapter à un maximum de machines.
- Il doit bénéficier d'un temps de chargement acceptable (La plupart des futurs utilisateurs de Visa ne bénéficient pas de connexion à Internet à haut débit).
- Dans le but d'une future exploitation, le code devra être commenté, et faire l'objet d'une documentation détaillée.

4.2. Les solutions techniques apportées pour l'élaboration des améliorations de Visa

Ce paragraphe se veut être un résumé des modifications apportées à l'interface Visa.

Il présente de façon succincte toutes ces modifications suivant le cahier des charges prédéfini, les problèmes et solutions rencontrées durant leur élaboration.

Afin de ne pas surcharger ce rapport, pour plus de détails, des renvois vers l'annexe permettent d'accéder aux codes commentés, à la documentation des programmes créés ou modifiés durant le stage, ainsi qu'aux diverses notices utilisées pour la réalisation de ceux-ci.

4.2.1. Les cartes Sdoc et Ndoc

4.2.1.1. Approche globale, et choix de l'élaboration d'un applet Java

Dans le cas de ces cartes une solution avait déjà été utilisée précédemment faisant appel à un concept d'image gif greffée sur une page HTML avec liens hypertextes positionnés sous les points représentant les clusters. L'ensemble de ce système était généré par un package de programmes C, C++, et Perl.

Le principe était relativement efficace, mais peu homogène au niveau du code, et difficile à modifier. Le paramétrage était laborieux et nécessitait l'ouverture d'une page HTML. De plus, l'affichage des liens de Sdoc n'était pas pris en compte.

La création d'une applet Java fut la solution adoptée, avec la nécessité de redéfinir toutes les fonctionnalités de l'ancienne version 'gif', et d'implémenter celles comprenant les liens.

4.2.1.2. Les choix propres à Java

Les contraintes techniques concernant la compatibilité du système couplées à celles touchant à la vitesse de téléchargement, et à la minimisation de l'occupation mémoire du programme, rendirent le choix du catalogue de classes AWT de la bibliothèque Java quasiment obligatoire.

Ce dernier, bien que relativement ancien et bien moins complet que celui, plus moderne, des classes Swing, a l'avantage d'être lisible par la plupart des navigateurs et machines clients. Mais surtout la comparaison du temps de téléchargement de deux versions d'applet, une étant en AWT et l'autre en Swing, montra la plus grande rapidité de l'applet AWT (plus du triple de temps pour Swing par rapport à AWT).

4.2.1.3. Les paramètres de l'applet

Pour permettre la création de l'applet approprié, la création d'un C.G.I. fut accomplie sur la base d'anciens travaux.

Les paramètres à prendre en compte furent:

- Le nom de l'analyse, dans ce cas la classification, son code d'accès dans la structure H.F.D.. Ce nom de code est celui du fichier, de type anaXXXXXX (**cf. « exemple de fichier de définition des chemins dans la structure H.F.D. » en annexe**), placé dans le répertoire « analyse » de l'application. Celui-ci contient tous les chemins d'accès aux informations propres à cette analyse dans l'arborescence ordonnée des fichiers.
- Une liste des noms de la classe regroupant les mots clés, les codes d'accès dans la structure H.F.D.. Ces codes sont les noms des fichiers dans un (ou des) répertoire(s) de type « .clu », pour .cluster, (**cf. « exemple simplifié de fichiers normalisés SGML de type .clu » en annexe**) contenant toutes les structures d'analyse de type SGML. On récupère ensuite les coordonnées spatiales définies par respectivement Sdoc ou Ndoc.
- Une liste des liens orientés de la classe et leur force respective dans le cas de Sdoc. Cette fois, c'est un fichier de type « .ass », pour .association, (**cf. « Exemple simplifié de fichiers normalisés SGML de type .ass » en annexe**) qui sera traité sur le même principe que celui des noms de classe.
- Les dimensions propres à la carte, dans le cadre de la représentation graphique de l'applet.
- La base de l'URL dans le contexte de l'application.

Ces paramètres présents dans les structures SGML des fichiers.clu et.ass, issus de l'analyse globale des données, furent extraits grâce à l'emploi de I.lib.

4.2.1.4. Récupération des paramètres de l'applet

Le fonctionnement du C.G.I. lançant l'applet de la carte suit ce principe :

1. Les droits d'accès aux pages traitant l'analyse sont vérifiés. Pour cela, on récupère les données contenues dans une variable d'environnement globale, et on les compare avec les tables d'utilisateurs reconnus. Tout refus d'accès se verra

sanctionné par l'affichage d'une page HTML appropriée, et générée automatiquement en fin de C.G.I..

2 Sur le même principe de variable globale, le nom de l'analyse, son code, son dénominateur et son type, sont décomposés et placés dans des variables locales.

3 Les chemins d'accès aux fichiers à traiter sont lus dans le fichier ayant pour nom le code de l'analyse.

4 Des commandes I.lib sont appliquées aux données pour récupérer les paramètres souhaités dans des tableaux. Les manuels de toutes commandes I.lib utilisées dans les C.G.I. conçus durant ce stage, sont placés **en annexe**. La commande IhfdCat (respectivement IhfdZCat pour les fichiers zippés) concatène et affiche d'abord sur la sortie standard le contenu du fichier H.F.D. dont le nom est placé en paramètre. Ce nom correspond au code de la classification. Puis la commande IsgmlForm, 'pipée' avec la précédente, extrait les informations contenues entre les balises SGML de nom placé en paramètre. On obtient deux tableaux, chacune des lignes du premier tableau obtenu correspond à une classe, chacune des lignes du second correspond à un lien.

5 Le code HTML devant être généré dynamiquement par le C.G.I. est écrit entre doubles cotes précédées de la commande « print » afin d'être écrit sur la sortie, et envoyé au client. Il faut alors protéger (à l'aide d'un « \ »), voir supprimer, certains caractères spéciaux pouvant être mal lus par l'interpréteur Perl.

6 Les paramètres de l'applet sont extraits des lignes des tableaux, et toutes les informations utiles sont concaténées entre des séparateurs reconnus par l'applet.

4.2.1.5. Structuration de données dans l'applet et implémentation du tri rapide

Dans l'ancienne version de représentation graphique, deux modes de positionnement des points sur la carte étaient accessibles:

- Le mode de représentation spatiale, suivant à la lettre les lois énoncées par Sdoc ou Ndoc.
- Le mode de représentation par rang, enrichissant le premier mode, en positionnant les points selon leurs rangs en x et en y dans la représentation spatiale.

Les coordonnées en x et en y de Sdoc ou Ndoc sont récupérées de façon brute, sous forme de réels, par l'applet. Celui-ci doit prendre en charge le tri de ces paramètres et fournir les rangs de chaque point.

Trier les points à l'appel de la représentation par rang semblant être une mauvaise solution, il était préférable de définir les rangs dès l'initialisation de l'applet.

Lors du lancement de l'application Internet Java, les informations concernant un point (le nom, le code H.F.D., les coordonnées x et y) sont d'abord stockées dans un objet de type PointCarte. Puis tous ces objets PointCarte sont rangés dans un outil Vector Java.

Chaque objet de type PointCarte renferme un tableau de deux entiers (issus de la transformation en pixels, des réels entrés en paramètre, comme il sera décrit par la suite), contenant les coordonnées (x et y) fournie par les outils d'analyse, et ce pour permettre de standardiser le tri. La seule méthode de classe réalisant le tri n'est pas spécifique à x ou à y, elle trie simplement à l'aide d'un indice correspondant à un entier.

Le Vector lors de l'initialisation, est trié deux fois, selon x puis y. Après chaque tri, l'indice du PointCarte du Vector sert de rang, en x ou en y, à celui-ci. Ces indices sont stockés dans des variables de l'objet PointCarte dans le but d'être réutilisés par la suite.

L'algorithme de tri implémenté est celui du tri rapide appliqué à un Vector.

Pour les liens, les données récupérées en brut les concernant sont écrites de la façon suivante: Libellé de l'origine du lien(séparateur 1)libellé de l'aboutissement du lien(séparateur 2)poids du lien.

Une carte pouvant comprendre au maximum une centaine de points pour le double de liens en moyenne, la structuration de données des liens ne pouvait se faire sous forme de matrice sous peine de voir celle-ci presque vide (Dans le cas maximum, on aurait : (100*100-200) 9800 cases vides pour 200 remplies).Une liste chaînée peut éventuellement satisfaire à la demande en étant moins gourmande en mémoire.

Mais dans l'immédiat, la solution la plus simple, à savoir un tableau d'objet de type Liens (contenant le libellé du point d'origine, celui du point d'arrivé, et le poids de la relation) a été retenue.

4.2.1.6. Ergonomie de l'applet et flexibilité des données graphiques

Les données brutes fournies par les outils infométriques si elles sont rigoureusement justes, sont, au contraire, inexploitables graphiquement.

Les coordonnées spatiales ne sont pas compatibles avec les coordonnées graphiques exprimées en pixels. Il est donc nécessaire de réaliser des manipulations sur celles-ci avant de les utiliser pour le placement des points.

Il ne faut pas oublier non plus que la taille de la carte doit être facilement ajustable.

Les types de traitement ne fournissent pas les mêmes informations : il n'y a pas de liens sur Ndoc. Les modes de paramétrage propres aux liens sur Sdoc ne doivent pas apparaître dans ce cas, et il ne faut pas que le fonctionnement de l'applet en soit perturbé.

Dès lors, plusieurs choix furent adoptés :

- Ajuster les valeurs des coordonnées et des rangs en fonction de la taille prédéfinie de la carte, juste après le passage des paramètres lors de l'initialisation de l'applet. Les coordonnées sont transformées en pixels. La notion de rang perd son sens au profit de celle de coordonnées graphiques relatives les unes par rapport aux autres.
- Implémenter les fonctions de paramétrages pour qu'elles s'appliquent au niveau de la méthode d'affichage standard de l'applet (méthode paint()), et utiliser au maximum la méthode de réaffichage standard lors de chaque modification (méthode

repaint()). En conséquence, toutes les variables impliquées dans le paramétrage graphique doivent être globales.

- Placer les composants graphiques propres au paramétrage des liens seulement après le passage des paramètres dans l'applet, et les soumettre à leurs conditions. Par la suite, ceci a également rendu possible le calcul automatique du poids maximum des liens présents dans une carte. Ce poids sert alors de valeur étalon pour les listes de sélection de l'affichage des liens.

Un 'petit défaut' de Java a en même temps été corrigé : l'orientation du référentiel graphique de Java. L'origine du graphe a du être basculée de l'angle Nord/Est vers celui Sud/Est.

4.2.1.7. Reconstitution de l'URL par l'applet, visualisation des noms des points et des liens orientés

L'applet de la carte devait, comme pour l'ancienne version gif, permettre l'accès par simple click aux tableaux d'information sur la classe.

Le choix d'implémenter l'applet avec les interfaces Java MouseMotionListener et MouseListener ajouta d'autres possibilités à l'ensemble.

L'applet recrée l'URL à partir des données entrées en paramètre et stockées dans des variables de l'objet PointsCarte.

Chaque click est compté :

- Le premier déclenche l'affichage d'un lien orienté avec le nombre de documents concernés en son milieu. Ce lien peut avoir pour origine, ou comme arrivé, le point ciblé selon l'option choisit dans la fenêtre de sélection « orientation des liens » (« from » pour l'origine, « to » pour l'arrivée).
- Le second renvoie au tableau de la classe ciblée. L'URL est reconstituée dynamiquement par l'applet à chaque événement de ce type.

Chaque mouvement de la souris, entraîne le test et la comparaison des coordonnées graphiques de celle-ci avec celles des points de la carte, et éventuellement des liens. En cas de concordance, le nom des points, ou liens, est affiché dans une lucarne.

La première fonction de cette lucarne a été de facilité la mise au point et le débuggage du programme.

L'introduction d'autres fonctionnalités de ce type, utilisant les événements de la souris, est prévue ultérieurement.

4.2.1.8. Exemples des fonctions de l'applet des cartes Sdoc et Ndoc

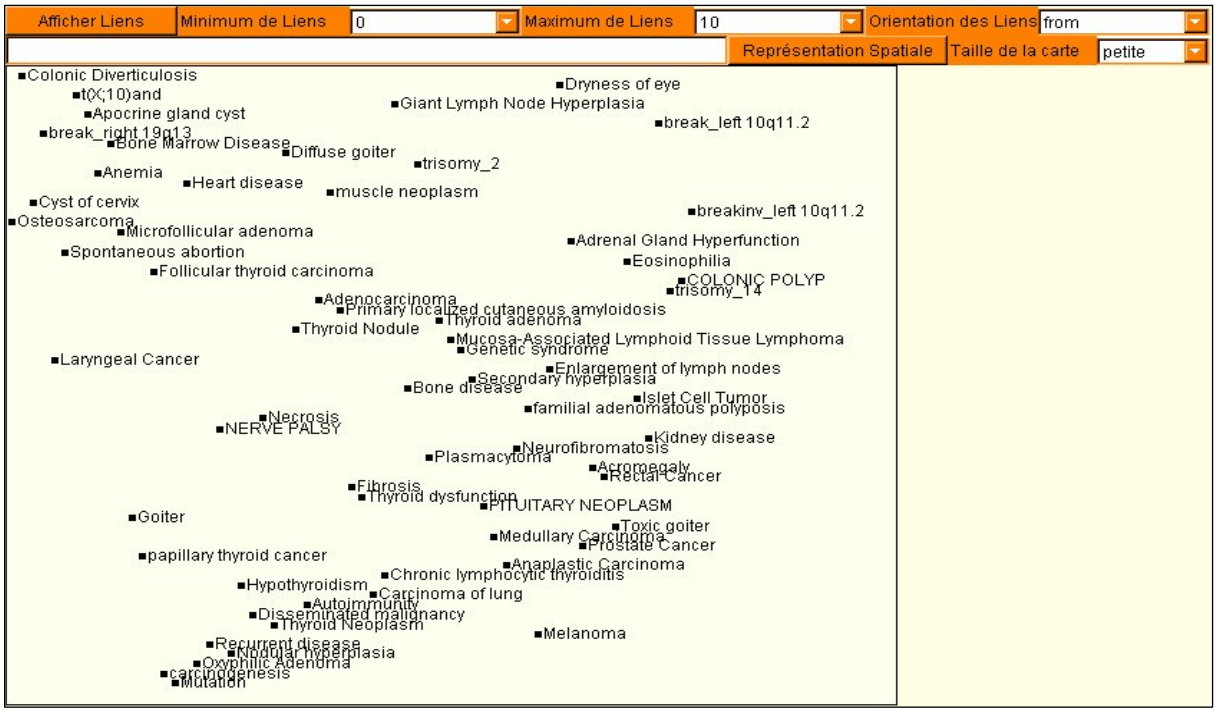


Image 5 Application de l'applet à Sdoc, en petite taille, et représentation par rang.

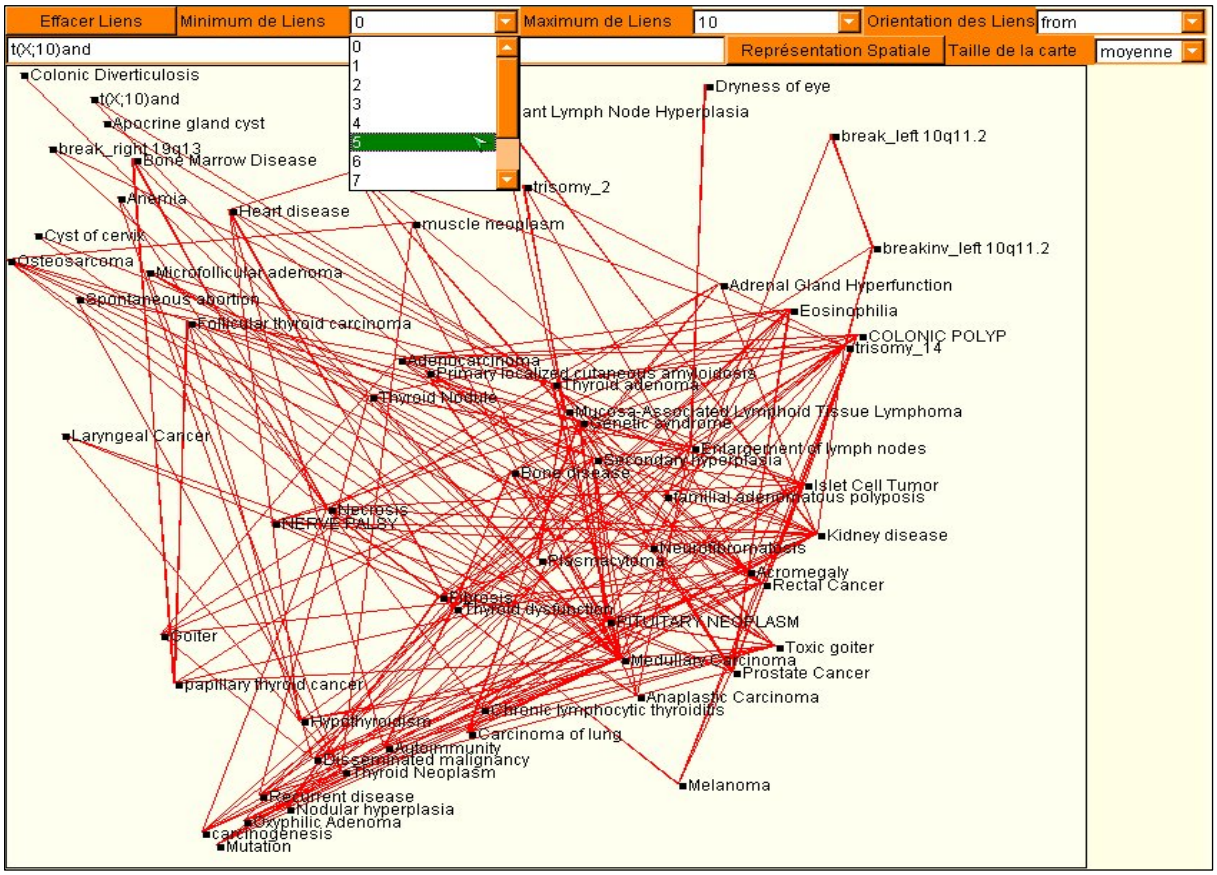


Image 6 Application de l'applet à Sdoc, en taille moyenne, avec représentation par rang, et liens affichés.

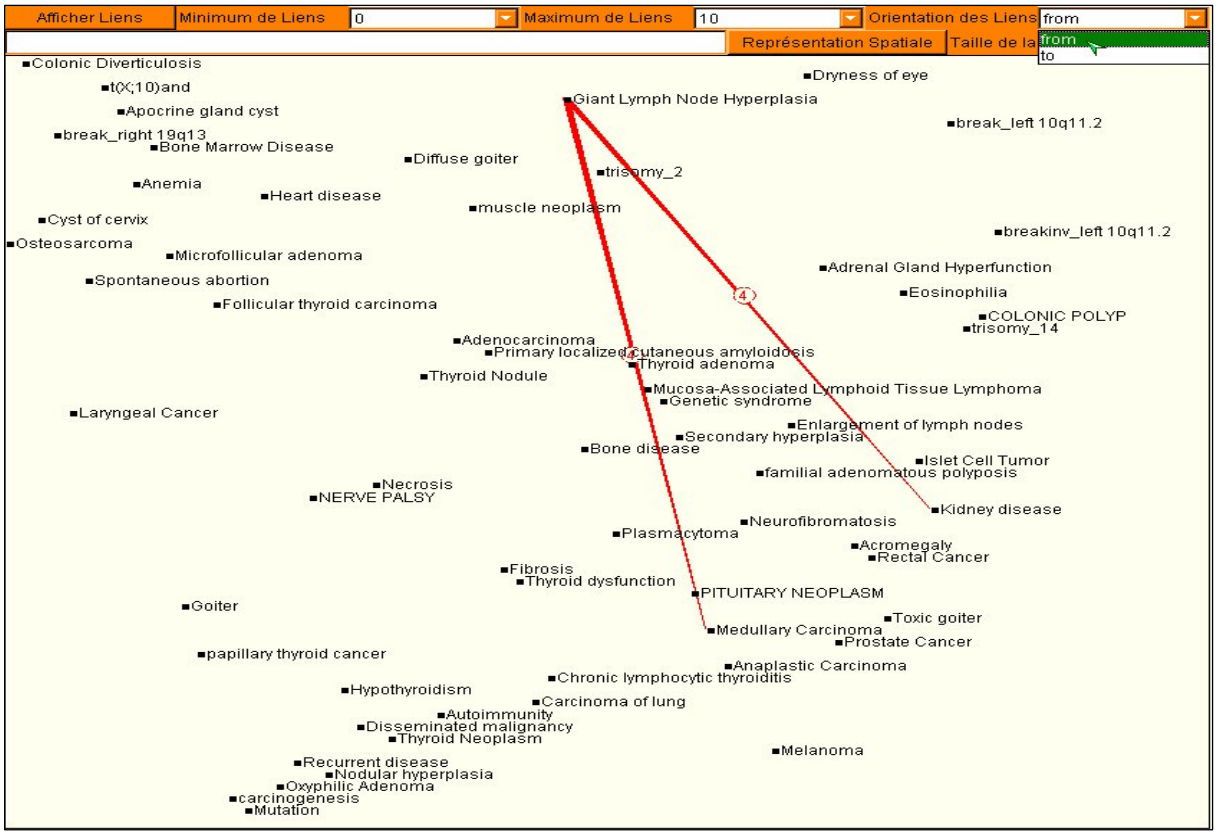


Image 7 Application de l'applet à Sdoc, en grande taille, avec représentation par rang, et liens orientés affichés.

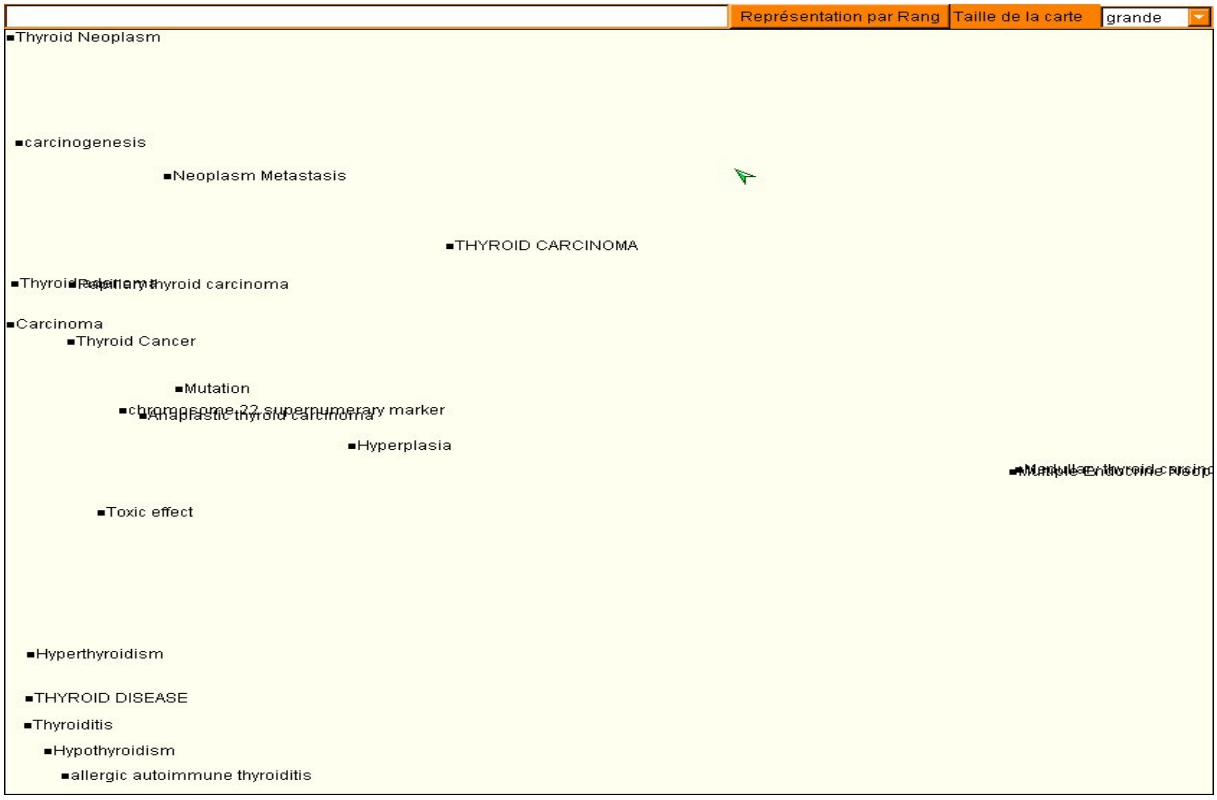


Image 8 Application de l'applet à Ndoc, en grande taille, avec représentation spatiale.

4.2.2. Le graphe relationnel Sdoc

4.2.2.1. Analyse de l'existant

La base de l'applet modélisant le graphe relationnel existait avant l'amélioration de Visa. Il avait été récupéré sur Internet et avait été intégré à Visa par Dominique Besagni.

Cet applet était assez simple et apportait peu d'information, il ne représentait que les liaisons internes à un cluster, et le programme n'était absolument pas commenté.

Il fallait donc y rajouter les mots-clés externes en les rendant distinct du point de vue graphique, indiquer le nom des clusters externes correspondant, les fréquences de chaque mot-clé, et le poids de chaque lien.

Tout applet Java nécessitant le passage de paramètres lors de son initialisation, le C.G.I. le prenant en charge a dû être modifié en conséquence(**cf. « Code du C.G.I. acces_graphe » en annexe**).

4.2.2.2. Modifications de l'applet du graphe

A part la recherche des informations dans la structure SGML de données, et l'assimilation de la I.lib, la principale subtilité de ces modifications résida dans la réutilisation du code déjà en place sans devoir le casser complètement (**cf. « Code de l'applet des graphes Sdoc » en annexe**).

Il a fallu réutiliser les objets de type Nodes pour qu'ils puissent prendre en compte la différence entre liens externes et liens internes. Plutôt que de créer une nouvelle classe spécifique aux nœuds externes, l'ajout d'un booléen dans la classe Nodes a permis cette distinction. Il ne restait alors plus qu'à ajouter de nouvelles méthodes de classes pouvant faire logiquement la séparation entre les nœuds internes et externes, dans leur création, ce qui est relativement complexe (Les données passées en paramètre ne faisant pas la différence entre les types de nœuds, ce fut à l'applet de le faire), et dans leur affichage, ce qui est très simple grâce au booléen.

Les attributs supplémentaires ont été ajoutés dans des variables des classes Edges et Nodes.

Ils ont été disposés ensuite dans des zones graphiques modulables selon leur taille.

Une partie non négligeable du travail consista à ajuster ces tailles et les polices de caractères, en fonction de tous les cas de figure envisageables, ce qui dans le principe ne représente pas de grandes difficultés, mais qui en pratique est assez long et laborieux. Plusieurs essais sont nécessaires pour le positionnement au pixel près de l'attribut, d'où autant de compilations du programme (pour généralement un chiffre modifié dans celui-ci).

4.2.2.3. Visualisation de l'applet du graphe

Le graphe Sdoc est animé. Les nœuds se positionnent automatiquement de façon à respecter les poids des liens. L'**image 9** représente une vue éclatée du graphe. Les nœuds sont positionnés manuellement pour rendre l'ensemble plus visible. Pour cela, le bouton « Stop » est activé, et, à l'aide de la souris, les nœuds sont déplacés. Pour relancer l'animation, on appuie sur le bouton « Restart », et les nœuds reprennent alors progressivement une position respectant les contraintes imposées.

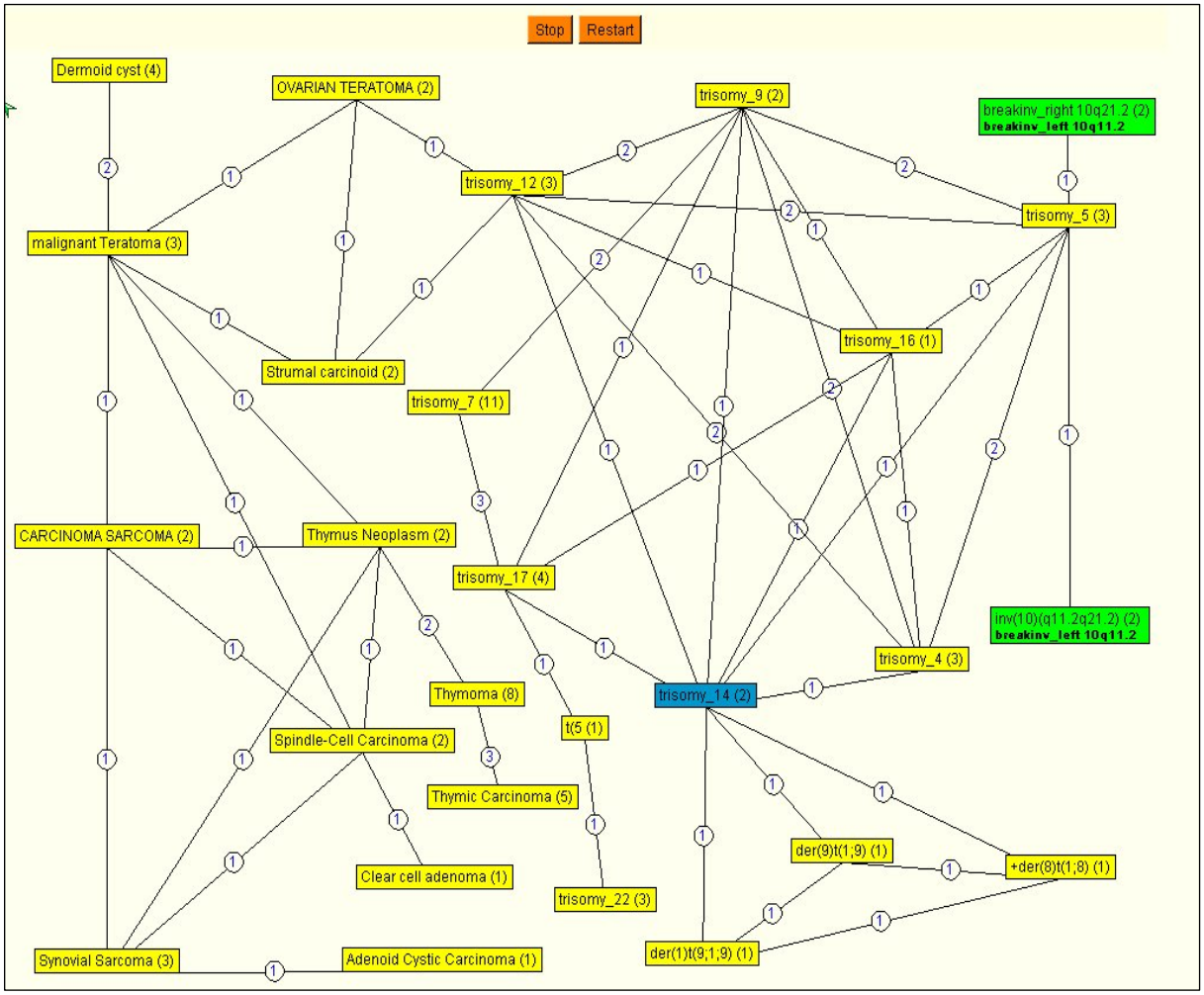


Image 9 Applet du graphe Sdoc.

4.2.3. L'accès à un panel de liens concernant chaque mot-clé : le « coolmenu », une solution JavaScript pratique et efficace

4.2.3.1. Recherche de solutions

Pour l'accès aux ressources, plusieurs solutions ont été envisagées :

- Faire apparaître les liens directement dans l'applet Java à l'aide de menus contextuels, ou popupmenus Java. Cette solution fut testée avec un succès relatif. En effet les menus popup fonctionnaient correctement dès le premier lancement, mais l'utilisation de l'historique du navigateur les rendaient complètement 'amnésiques', ils ne se souvenaient plus de leur fonction première (????). Le problème avait pour origine le vidage plus ou moins aléatoire de la mémoire cache d'Internet Explorer. Ce bug a été identifié comme un problème connu du logiciel de navigation, mais les solutions apportées par le constructeur, à savoir l'utilisation de 'méta' dans l'entête du document HTML, mais aussi à la fin de celui-ci, se sont avérées insuffisantes, et cette solution a été définitivement abandonnée.
- L'écriture de liens URL simples renvoyant à une page HTML contenant d'autres liens URL, correspondant aux accès aux ressources. Cette solution fonctionne sans nuisances apparentes et est déjà employée dans d'autres endroits du site, mais elle a l'inconvénient d'être lourde à mettre en place, et peu rapide et conviviale.

Finalement, une recherche de solutions sur Internet a permis de découvrir l'existence d'une solution JavaScript pratique, gratuite, en 'open source', et avec comme seule obligation de faire apparaître le 'Copyright' dans la page HTML utilisant ce procédé : Le 'coolmenu' développé par Dynamic HTML central.com et créé par Thomas Brattli (**Image 10**).

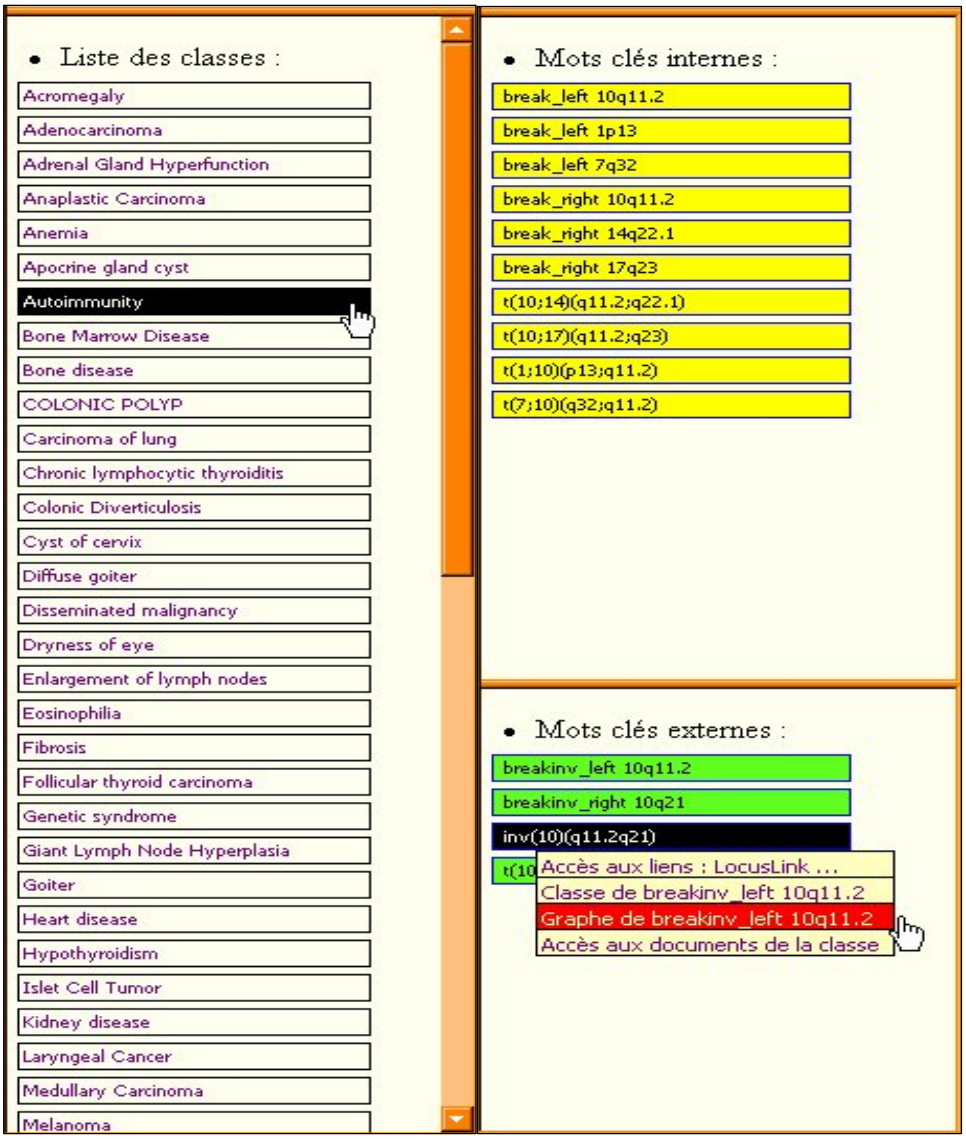


Image 10 Illustrations du « coolmenu »

4.2.3.2. Intégration des menus

Le travail consiste à comprendre le fonctionnement et l'utilisation de ce script, son paramétrage, et à l'intégrer dans les C.G.I. souhaités (**cf. « Code javascript du « coolmenu » » en annexe**).

Au niveau des C.G.I. (**cf. les C.G.I. « choix_ analyse_2_legende, legende_interne et legende_externe » en annexe**), l'intégration se fait suivant les étapes suivantes :

1 Le style des blocs du menu est défini dans l'entête de la page HTML, entre les balises <style> et </style>. Les couleurs du fond et du cadre des blocs, la taille, la couleur et la police des caractères sont décrits pour les blocs de niveaux 0, 1 et 2, en fonction des événements générés par la souris. Le niveau 0 représente les premiers Items du menu, le niveau 1, les sous-menus de ces Items, et le niveau 2, les sous-menus des Items de niveau 1.

2 Entre les balises <javascript> et </javascript>, la disposition spatiale des blocs par rapport à l'ensemble de la page web, puis les uns par rapport aux autres, est paramétrée.

3 Le contenu du menu est introduit, en respectant la hiérarchisation des blocs. Dans les lignes d'appels de fonctions javascript, on place en paramètre de la méthode « oCMenu.makeMenu() », 'le nom de variable du bloc menu', 'le nom de variable du bloc menu auquel il se rattache (rien s'il s'agit d'un bloc de niveau 0)', 'le texte à afficher dans ce bloc', 'le lien URL correspondant à ce bloc', 'le mode d'affichage de la page raccrochée à l'URL'.

4 La construction du menu est initialisée avec la ligne « oCMenu.construct() ».

4.2.3.3. Homogénéisation des pages

Il fallait aussi homogénéiser l'ensemble des pages pour que l'information présentée soit la plus compacte et la plus lisible possible, mais il fallait surtout éviter les chevauchements entre l'applet et le menu.

Des structures de page comportant plusieurs frames ont alors été mises en place suivant un schéma définit, avec un C.G.I. programmé par frame. La **figure 15** décrit le principe général des structures des frames. Il existe des variations entre les pages traitant les cartes et celles traitant les graphes (**cf. figures A et B de l'annexe, et les images 11 et 12**).

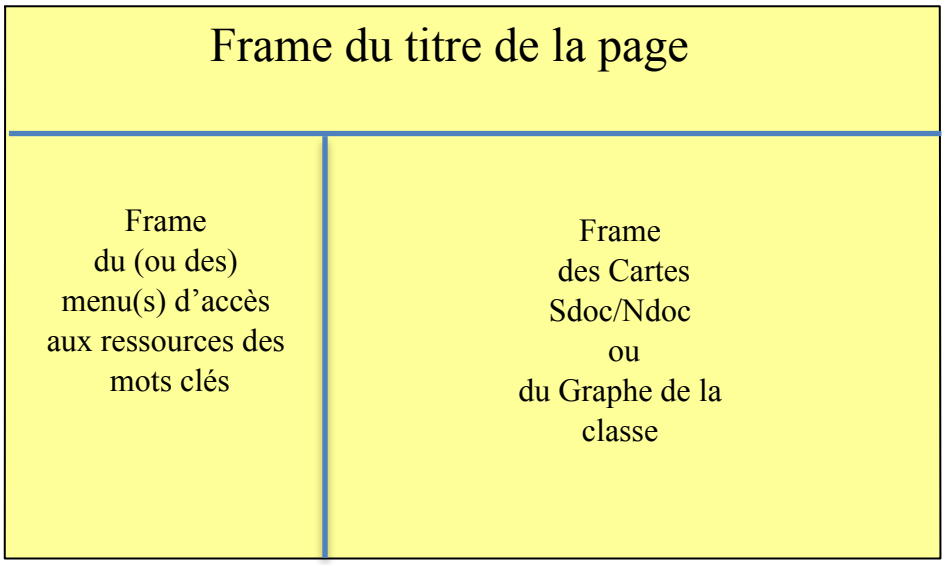


Figure 15 Structure des frames des pages de Visa.

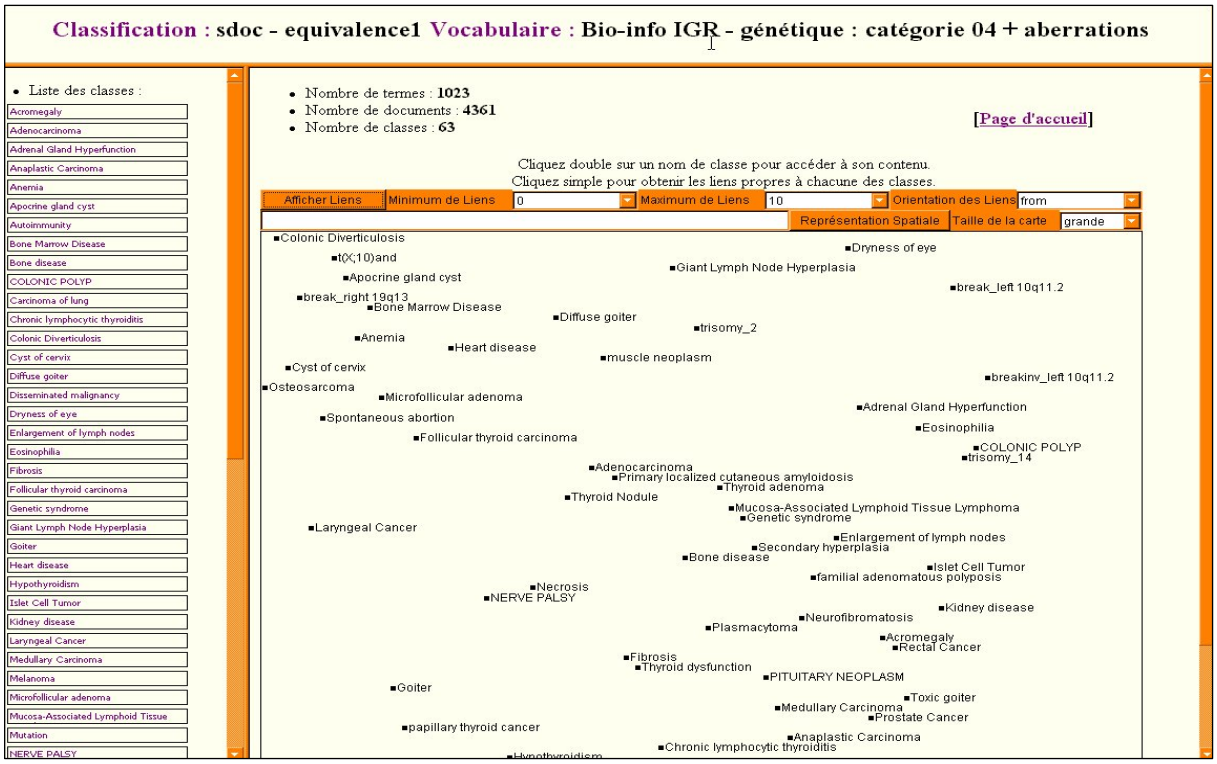


Image 11 Aspect général de la page des cartes Sdoc et Ndoc.

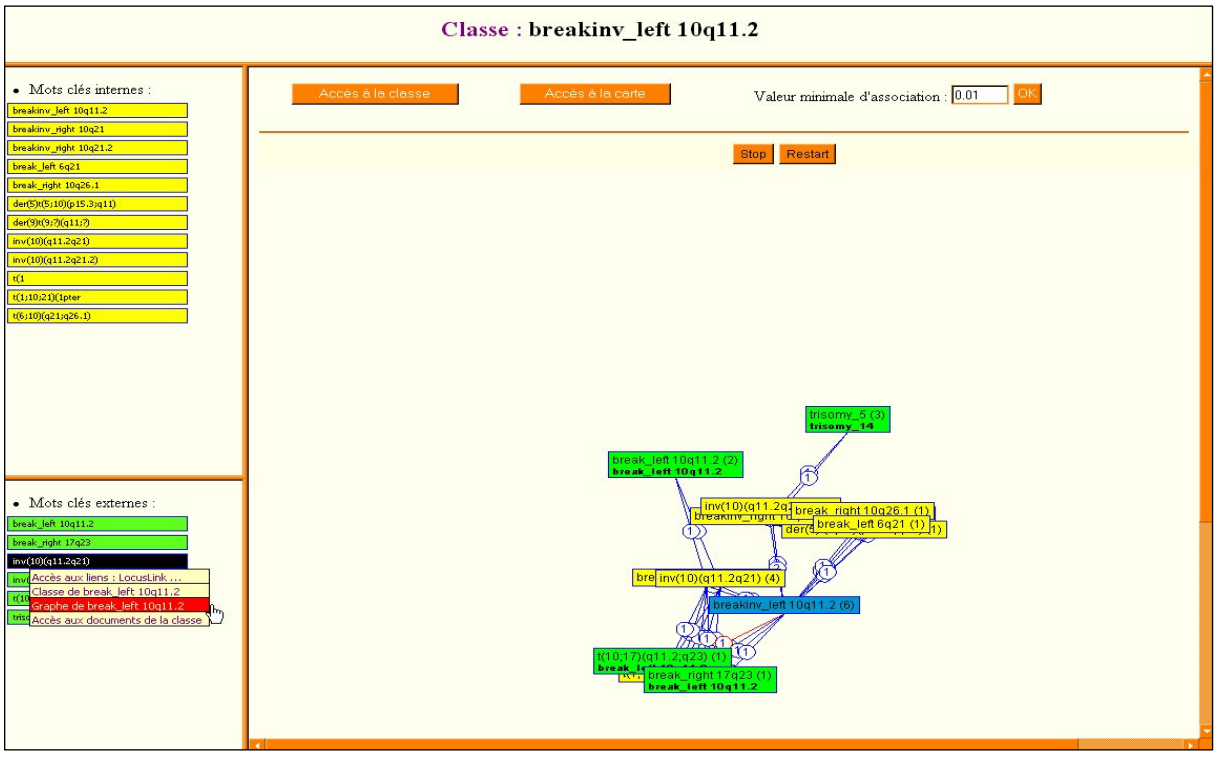


Image 12 Aspect général de la page des graphes Sdoc.

Conclusion générale

Le projet I.G.R. représente une avancé pour l'analyse expérimentale de données spécifiques. Son module de visualisation a pour ambition de permettre l'ouverture de nouvelles perspectives dans le domaine de la recherche scientifique et technique. En facilitant l'accès à l'information, il simplifie le travail bibliographique nécessaire à toutes études sérieuses.

D'un point de vue personnel, ce stage m'a fait de découvrir des notions peu, ou pas, abordées durant ma formation universitaire.

Durant les quelques mois correspondant à la conception et la réalisation de l'interface de visualisation de ce projet, je me suis familiarisé avec le concept des C.G.I., le langage de programmation Perl, les normes SGML, et une méthode d'exploitation de celles-ci : le manuel de fonction de la bibliothèque I.N.I.S.T., la I.lib.
J'ai pu approfondir, compléter, renforcer mes connaissances en Java, et en C.

Mais, plus que l'aspect purement informatique, nécessaire à la validation du DESS double compétence, ces travaux m'ont apporté un regard plus critique sur les notions propres à la bio-informatique, sur ma première compétence en général, la biologie.
J'ai découvert le monde de la documentation, ces différentes méthodes de travail, d'analyse, de représentation de l'information.
Je me suis renseigné sur ces principaux débouchés en pleine expansion dans le monde scientifique.

Cette expérience professionnelle ne me renvoie pas l'image d'une formation pour une compétence complémentaire en informatique, mais plutôt celle d'une spécialisation dans une double compétence, dans la biologie et l'informatique : la bio-informatique.