Université Henri Poincaré Nancy 1
Université Nancy 2
Institut National Polytechnique de Lorraine

D.E.S.S. Information Scientifique et Technique
Intelligence Economique
Année universitaire 2000-2001

Développement des applications de DILIB "IMD" et "Transcriptome"

par

Claude Nemurat

Maîtres de stage :

Jacques DUCLOY
Institut de l'Information
Scientifique et technique
54514 Vandoeuvre-lès-Nancy
03 83 50 71 20

Dr Bertrand RIHN
Institut National de Recherche
et de Sécurité
54514 Vandoeuvre-lès-Nancy
03 83 50 20 62

Stage effectué du 1er Mai au 31 Juillet 2001 à :

Institut National de Recherche et de Sécurité
et
Institut National de l'Information Scientifique et Technique

Université Henri Poincaré Nancy 1
Université Nancy 2
Institut National Polytechnique de Lorraine

D.E.S.S. Information Scientifique et Technique
Intelligence Economique
Année universitaire 2000-2001

Développement des applications de DILIB "IMD" et "Transcriptome"

par

Claude Nemurat

Maîtres de stage :

Jacques DUCLOY
Institut de l'Information
Scientifique et technique
54514 Vandoeuvre-lès-Nancy
03.83.50.20.00

Dr Bertrand RIHN
Institut National de Recherche
et de Sécurité
54514 Vandoeuvre-lès-Nancy
03.83.50.46.00

Stage effectué du 1er Mai au 31 Juillet 2001 à :

Institut National de Recherche et de Sécurité
et
Institut National de l'Information Scientifique et Technique

Avant propos

Ce stage a été réalisé dans le cadre du DESS Information Scientifique et Technique-Intelligence Economique cohabilité par les trois universités de Nancy (Université Henri Poincaré Nancy 1, Université Nancy 2, Institut National Polytechnique de Lorraine). Il est issu d'une collaboration entre l'Institut National de Recherche et de Sécurité (INRS) et l'Institut National de l'Information Scientifique et Technique (INIST). L'objectif du stage consiste à optimiser et à automatiser les applications d'une plate-forme documentaire développée par l'INIST mises en place à l'INRS.

Je tiens à remercier :

- Florian MAZUR pour son suivi, sa disponibilité, et ses précieux conseils,
- Philippe HOUDRY pour son attentive relecture,
- Bertrand RIHN pour ses conseils lors de la rédaction du rapport,
- Jacques DUCLOY pour la confiance qu'il m'a accordée,
- Françoise GRANJEAN pour m'avoir permis de réaliser ce stage,
- Alain ZASADZINSKI pour l'apport de ses connaissances en biologie moléculaire,
- Michel SERVAIS pour son aide lors de l'installation des applications à l'INRS,
- Catherine CZYSZ pour son aide dans le règlement des questions administratives,
- Sébastien VACHENC pour son apport concernant l'exploitation du serveur "Transcriptome ",
- Tous les membres du DPS, du centre de documentation et du service informatique de l'INRS pour leur accueil, leur sympathie et leur soutien.

Note : Tous les mots et sigles suivis d'une "*" ont une définition dans le glossaire. Les numérotations entre "[ ]" renvoient à la bibliographie.

SOMMAIRE

1 Introduction

Dans un centre de recherche comme l'Institut National de Recherche et de Sécurité (INRS*), la production bibliographique est très importante, les domaines d'investigation de l'institut pour la prévention des accidents du travail et des maladies professionnelles étant très étendus. Afin de gérer et d'exploiter de façon optimale une telle quantité d'information, le système informatique se doit d'être performant. Par ailleurs, le fond documentaire doit être accessible facilement pour tous les chercheurs. C'est dans cette optique que l'INRS utilise des produits tels que " AIRS Web " et la plate-forme documentaire *Documentation and Information Library* (DILIB*).

Jacques Ducloy, concepteur de la plate-forme DILIB et responsable du Département Produit et Services de l'Institut National de l'Information Scientifique et Technique (INIST*), a permis le développement d'une nouvelle version de DILIB intéressant l'INRS. DILIB est un outil permettant l'exploitation de gros corpus documentaires tels que ceux possédés par l'INRS. Le centre de documentation de l'INRS utilise DILIB depuis plusieurs années car cet outil offre l'avantage d'évoluer selon les besoins propres de l'institut et permet, grâce à l'intranet, une consultation aisée et dynamique de toute l'information documentaire par l'intermédaire de l 'application INRS Multi Data (IMD*).

DILIB offre également des fonctionnalités intéressantes pour l'analyse de l'information. C'est ce qui a suscité l'intérêt de Bertrand Rihn, chercheur à l'INRS. Il a souhaité utiliser DILIB pour l'exploitation de données bibliographiques liées aux résultats d'une étude des gènes impliqués dans le mésothéliome humain (Cancer de la plèvre). Cela a donné lieu à la création de l'application " Génome " rebaptisée application " Transcriptome " au cours de mon stage.

Mon travail lors de ce stage a consisté à automatiser autant que possible les deux applications de DILIB implantées à l'INRS et faire évoluer les fonctionnalités de l'application " Transcriptome ".

2 Présentation de l'Institut National de Recherche et de Sécurité (INRS).

2.1 Son rôle et son statut.

L'INRS [1] a pour rôle de contribuer sur le plan technique, par tous les moyens appropriés, à la prévention des accidents du travail et des maladies professionnelles pour assurer la protection de l'homme au travail et sa sécurité.

Les différentes activités de l'institut s'exercent autour de quatre grands thèmes :

- mieux identifier et connaître les risques professionnels,
- analyser leurs conséquences pour la santé et la sécurité de l'homme au travail,
- rechercher comment les combattre et les maîtriser,
- faire connaître et enseigner les moyens de leur prévention.

L'INRS est au cœur du dispositif français de prévention des risques professionnels. C'est une composante de l'Institution prévention côté Sécurité sociale.(Figure 1). Son budget provient d'une subvention d'équilibre attribuée par la Commission des Accidents du Travail et des Maladies Professionnelles de la Caisse Nationale de l'Assurance Maladie des Travailleurs Salariés (CNAMTS*).

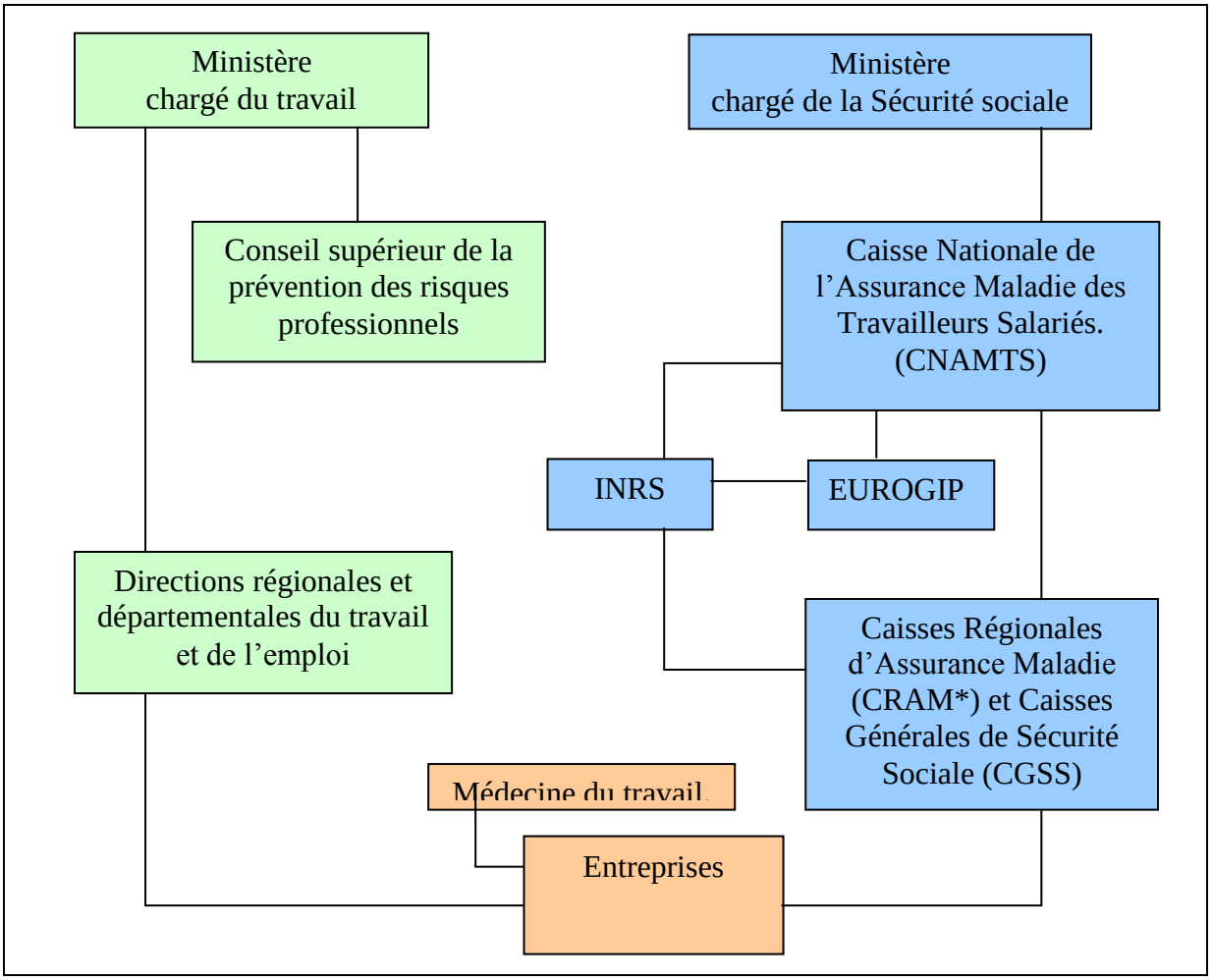

Figure 1 : Situation de l'INRS dans le dispositif français de prévention des risques professionnels.

L'INRS exerce ses activités au profit des entreprises du régime général de toutes les branches d'activité (métallurgie, chimie, transports, services…), en partant des besoins exprimés par la CNAMTS, les ingénieurs et les contrôleurs des services de prévention des CRAM (Caisses Régionales d'Assurance Maladie), les partenaires sociaux, les médecins du travail, les inspecteurs du travail…

2.2 Ses différentes missions.

Les éléments de cette présentation sont tirés d'un document de l'INRS [2]

2.2.1 L'assistance.

Elle est pleiomorphe :

- assistance documentaire (bibliothèque ouverte au public), technique et médicale,
- élaboration de guides de prévention, bases de données,
- participation à l'élaboration de normes et textes de référence,
- réalisation d'essais, de mesures et d'analyses, notamment à la demande des CRAM,
- missions spéciales de contrôle des produits chimiques, des machines dangereuses et des équipements de protection.

2.2.2 Les études et recherches.

Elles consistent en :

- programmation, conduite d'études en santé et travail, coordination par projets, valorisation,
- veille, investigations scientifiques et techniques,
- conception et validation d'outils, méthodes et procédés de prévention,
- publications scientifiques et techniques, colloques.

2.2.3 L'information.

Elle se concrétise en :

- sensibilisation : campagnes nationales,
- publications périodiques et non périodiques,
- banques de données, cédérom, sites Internet,
- conception et réalisation d'affiches, stands et audiovisuels,
- organisation de journées techniques, colloques.

2.2.4 La formation.

Elle s'articule autour de :

- conception et organisation de stages pour spécialistes de la prévention,
- enseignement à distance,
- formation de formateurs, assistance pédagogique,
- formation à la prévention dans l'enseignement,
- conception d'outils pédagogiques.

2.3 Le centre de Vandoeuvre.

Le centre de Vandoeuvre-lès-Nancy compte environ 400 personnes et six départements Etudes et Recherches. Il est chargé d'élaborer les projets du programme d'études et de recherches de l'INRS, et de les soumettre à l'avis des différentes instances qui les examinent avant leur adoption éventuelle par le conseil d'administration.

Les Projets de recherche correspondent à des demandes émanant de l'extérieur (CRAM*, CNAM*, médecine du travail, organisations professionnelles…) et sont déclinés dans le cadre d'un Plan à Moyen Terme (PMT) tous les cinq ans.

2.3.1 La documentation

L'INRS Paris possède un important fonds documentaire, constitué d'ouvrages, de brochures, d'articles et de périodiques. Son corpus s'accroît de 3000 documents par an, dont environ 1000 ouvrages et brochures. L'ensemble du fonds documentaire est consultable par le grand public dans le centre de documentation parisien de l'INRS.

Le centre de Vandoeuvre-lès-Nancy possède son propre centre de documentation créé en 1970. Celui-ci, qui emploie quatre personnes, est strictement à usage interne. Il met à la disposition des chercheurs ses ressources propres (plus de 160000 références) ainsi que toutes les ressources de l'institut. Ce service a pour mission de fournir aux chercheurs du centre l'information dont ils ont besoin pour la réalisation de leurs programmes d'étude et de recherche.

Des ordinateurs mis à leur disposition permettent de consulter les bases de données du centre sur l'intranet. On peut également consulter des bases de données sur cédérom (*medline*, toxline, CC-info, …). L'accès aux ressources documentaires se fait via Intranet depuis 1996. La documentation s'est ensuite intégrée au Site Inter-ligne ouvert en 2000.

2.3.2 Le réseau informatique et les bases de données.

Les bases de données de l'INRS sont gérées avec le logiciel AIRS. Le module client du logiciel permet à la fois de consulter les bases de données sur l'Intranet et de les enrichir. Les chercheurs ont donc la possibilité d'alimenter des bases de données spécifiques à leurs thématiques de recherche. La gestion du serveur AIRS est assuré par Michel Servais. Il est chargé de la création des nouvelles bases, qui pourront ensuite être utilisées par les chercheurs. Comme nous le verrons par la suite, ces bases sont aussi consultables par l'intermédiaire d'un serveur d'investigation généré avec la plate-forme documentaire DILIB.

2.3.4 Le laboratoire de cancérogenèse.

Le laboratoire de cancérogenèse, dirigé par Bertrand Rihn, fait partie du département Polluant et Santé, qui a pour vocation la recherche en toxicologie dans le domaine de l'évaluation des risques dus aux expositions professionnelles aux produits chimiques. Les recherches du groupe dirigé par Bertrand Rihn concernent deux thématiques particulières :

- L'action mutagène des toxiques industriels sur des modèles murins transgéniques*.
- L'étude des cancers professionnels, en particulier le mésothéliome* (cancer de la plèvre provoqué par l'amiante) par des techniques de biologie moléculaire, dont la technique des puces à ADN*.

Ce sont les résultats de l'étude de l'expression des gènes* impliqués dans le mésothéliome*, par la technique des puces à ADN*, que nous avons exploités au cours de mon stage à l'INRS dans le cadre de l'application " Transcriptome *" généré à partir de la plate-forme DILIB.

3 Présentation de l'Institut National de l'Information Scientifique et Technique (INIST).

3.1 Son rôle et son statut.

Unité de service du Centre National de la Recherche Scientifique (CNRS*), l'INIST* [3] est le premier centre intégré européen d'Information Scientifique et Technique (IST*). Fournisseur de copies de documents, producteur de bases de données multilingues et multidisciplinaires recensant l'essentiel de la littérature internationale dans la plupart des domaines de la recherche, l'INIST étend aujourd'hui son offre de services sur internet.

3.2 Les missions de l'INIST.

Les éléments de cette présentation sont tirés du fascicule de présentation de l'INIST [4].

3.2.1 Une mission de service public.

L'INIST a pour principal objectif de servir les différents acteurs de la recherche publique, qu'il s'agisse du CNRS ou d'autres Etablissements Publics à caractère Scientifique et Technique (EPST), ou de l'enseignement supérieur (universités et grandes écoles), afin d'améliorer la collecte, l'analyse et la diffusion de l'information scientifique.

3.2.2 Un accès à l'information pour le milieu socioéconomique.

Les entreprises ont besoin de connaître l'état des recherches dans leur domaine d'activité ainsi que dans les secteurs connexes, afin d'être à même d'adapter au mieux leur propre stratégie de développement. De nombreux laboratoires de recherche privés ont recours quotidiennement aux différents services proposés par l'INIST :

- Services de recherche sur internet (ARTICLE@INIST, ARTICLESCIENCES).
- Bases de données (PASCAL*, FRANCIS*).

3.2.3 Développer l'accès à l'information électronique.

L'INIST offre à ces utilisateurs la possibilité d'identifier et de localiser un document, et d'en faciliter l'accès par l'intermédiaire de ses réseaux (service de fourniture de copies de documents primaires). C'est l'un des principaux enjeux lancé aux acteurs de l'Information Scientifique et Technique (IST*). C'est dans cette perspective que l'INIST met en place en 2001 un portail d'IST qui proposera, dans un environnement personnalisé et évolutif, un ensemble de ressources et de services produits par l'INIST et ses partenaires.

3.2.4 Développer la veille.

L'INIST étudie et développe de nouveaux outils de veille technologique et documentaire pour le traitement bibliométrique et l'analyse infométrique des données issues de diverses sources d'information, et en particulier de ses bases.

Ces applications constituent une aide à l'élaboration de stratégies scientifiques, tant pour les chercheurs que pour les entreprises.

3.3 Le département Produits et Services (DPS).

Le Département Produits et Services assure la constitution des bases bibliographiques de l'INIST, la fabrication des produits et la mise en place des services et leurs exécutions. Il comprend différents services :

- Des services de production (Fourniture de document, Formation, Traduction)
- Des services scientifiques (Sciences de la vie, Sciences Humaines et Sociales, Science Exactes et de l'Ingénieur).
- Des services transversaux (Gestion de Production et Budget, Ingénierie et Partenariat, cellule de veille).

Ses objectifs consistent à assurer les prestations de production et à entreprendre une mutation technologique, par exemple le déploiement de nouvelles compétences liées aux développement des nouvelles technologies. Dans ce cadre, la boîte à outils DILIB* est utilisée tant pour mettre à disposition des résultats de recherche bibliographique (présentation des résultats de recherches effectuées pour ses clients sous forme de serveurs d'investigations) que dans une optique de mutation technologique (formations internes à l'utilisation de DILIB).

4 La plate-forme DILIB.

4.1 Origine du projet DILIB.

La première version de la plate forme DILIB*, nommée ILIB (Information LIBrary), a été réalisée dans le cadre d'un projet de l'ancien Département Recherche et Produits Nouveaux de l'INIST. Cette première application a bénéficié des résultats de nombreux travaux antérieurs, notamment la plate-forme de production de l'Agence Nationale du Logiciel (ANL), des activités documentaires du Centre Inter universitaire de Ressources en Informatique de Lorraine et de l'Institut NAtional de la Langue Française (INALF). L'expérience s'est ensuite poursuivie au Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA), où l'application a évolué sous le nom de DILIB. Grâce à la collaboration de l'INRS, le projet DILIB a été réinitialisé à l'INIST l'an passé, notamment à l'occasion du stage de DESS* de Aude NEDELCOT [5].

4.2 Présentation de DILIB.

DILIB [6] est une plate-forme pour l'Ingénierie du Document (ID) et de l'Information Scientifique et Technique (IST). Celle-ci permet d'exploiter des documents de formats initialement différents et de générer des applications multi-bases.

Cette plate-forme contient différents éléments :

- une boîte à outils SGML/XML,
- des composants pour construire des Systèmes de Recherche d'Information,
- des modules infométriques,
- des interfaces avec des logiciels disponibles sur le marché (dont la publication sur le web).

Elle permet à la fois l'exploration de corpus documentaires par l'intermédiaire d'une interface conviviale de navigation hypertexte, et leur exploitation par une analyse infométrique poussée.

Les domaines d'application de DILIB sont vastes mais il existe trois principaux produits cibles :

- l'investigation documentaire,
- la construction de Systèmes de Recherche d'Information,
- la mise en place d'outils pour les bibliothèques électroniques.

4.3 DILIB et la norme SGML/XML.

Selon les bases de données, les notices bibliographiques ont une structure différente. Par exemple, le format des notices dans les bases de données de l'INRS est spécifique au logiciel de gestion documentaire utilisé (AIRS*). Pour que ces données soient accessibles à tous, il est nécessaire de les reformater dans un format universel. Le *Standard Generalized Markup Language* (SGML*) est une norme qui permet de décrire et de représenter tous les documents structurés. Conçue au départ pour faciliter l'échange de gros volumes de documents scientifiques et techniques, dans le cadre d'un projet du département de la défense des Etats Unis, cette norme s'est imposée comme un langage de balisage généralisé par le monde des éditeurs.

La norme SGML décrit la structure logique des documents, indépendamment de leur format de départ, ce qui rend plus facile leur exploitation ultérieure. Le balisage consiste en l'intégration de marques délimitant les différentes parties du document.(Annexes 1 et 2).

Cette norme possède de nombreux avantages :

- La codification d'un document se fait en utilisant un jeu de caractères minimum (ISO 646*) et une codification spéciale pour les caractères spéciaux (accents, symboles mathématiques …) facilitant ainsi sa manipulation et sa portabilité.
- La norme sépare le contenu et la forme des documents. La cohérence de l'architecture des données traitées est donc indépendante du contenu de celles-ci.
- Le balisage favorise l'utilisation d'analyseurs lexico-syntaxiques, simplifiant l'exploitation des résultats.
- Le langage de balisage *HyperText Markup Langugage* (HTML*), utilisé sur le Web*, n'est autre qu'une *Document Type Definition* (DTD*) de la norme SGML qui mélange structure et présentation. L'utilisation de document au format SGML est donc aisée sur le web.
- La norme SGML est très largement répandue.

Pour pouvoir traiter des corpus au format SGML/XML*, DILIB* intègre des commandes d'adaptation des formats serveurs ou Texto*. Ceci permet de reformater les notices provenant de bases de données diverses avant leur utilisation. DILIB offre par ailleurs des outils supplémentaires facilitant le traitement des données au format SGML/XML.

4.4 Génération d'un serveur d'investigation DILIB.

Pour générer une application de DILIB*, il est nécessaire de disposer d'un certain nombre d'informations (Figure 2).

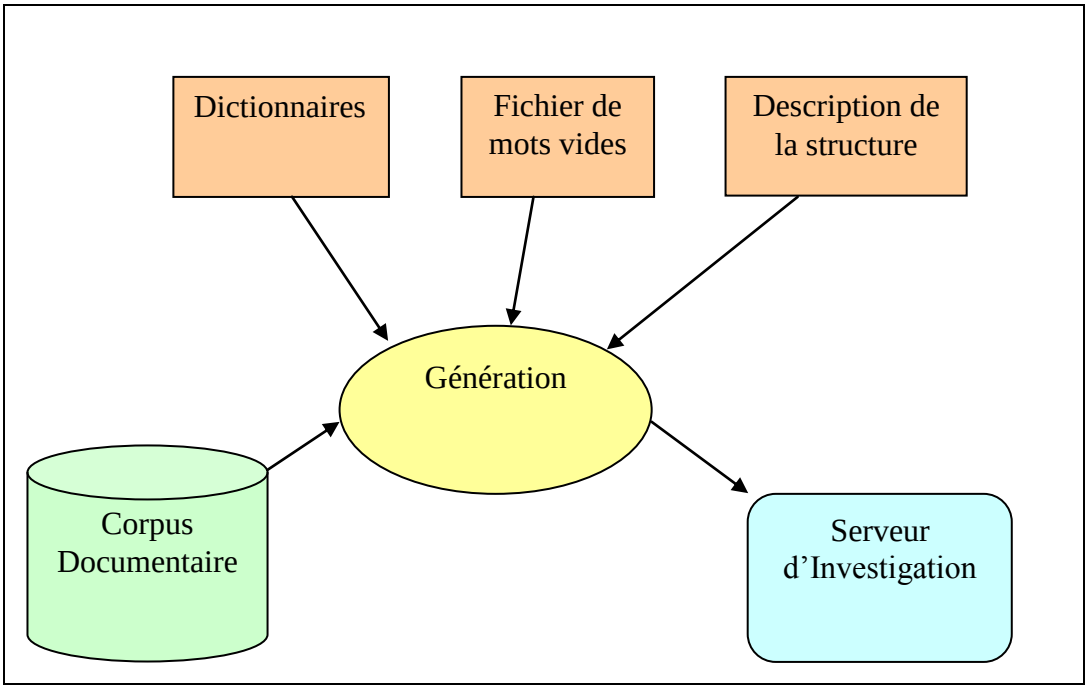

Figure 2 : Schéma de génération.

- Les dictionnaires (anglais et français) contiennent des informations nécessaires à la présentation du serveur (titres, sous-titres, intitulés des index…).
- Les fichiers de mots vides permettent de supprimer les termes inutiles lors des indexations textuelles (indexation intégrale du texte pour les mots des titres et des résumés).

- Le fichier de description du serveur permet de fixer la structure du serveur (par exemple les champs indexés). Les informations de ce fichier sont introduites dans un fichier de *makefiles** permettant de générer le serveur en fonction du paramétrage établi dans le fichier de description.

4.4.1 Le langage de définition d'un serveur DILIB.

Il s'agit de décrire formellement un serveur d'investigation pour permettre une génération automatique des différentes bases ou fichiers contenant les données du serveur et de produire automatiquement des interfaces de navigation. Le fichier de description du serveur est un fichier au format "ed "*. C'est un format XML* simplifié où certaines balises de fin sont omises (Figure 3).

```
<server code=NomServeur>
    <base code=NomBase>
         <index code=aut>
             <path>doc/aut/e#
         </index>
         <index code=mc>
             <path>doc/mc/e#
         </index>
    </base>
</server>
```

Figure 3 : Structure du fichier de description.

Ce fichier sera dans un premier temps converti au format XML*. Le fichier XML permettra de générer les fichiers de *makefiles** permettant l'élaboration du serveur.

4.4.2 Les différentes étapes de la génération.

La génération du serveur d'investigation comprend différentes étapes allant du traitement du corpus bibliographique à l'affichage sur un serveur Web* (Figure 4).

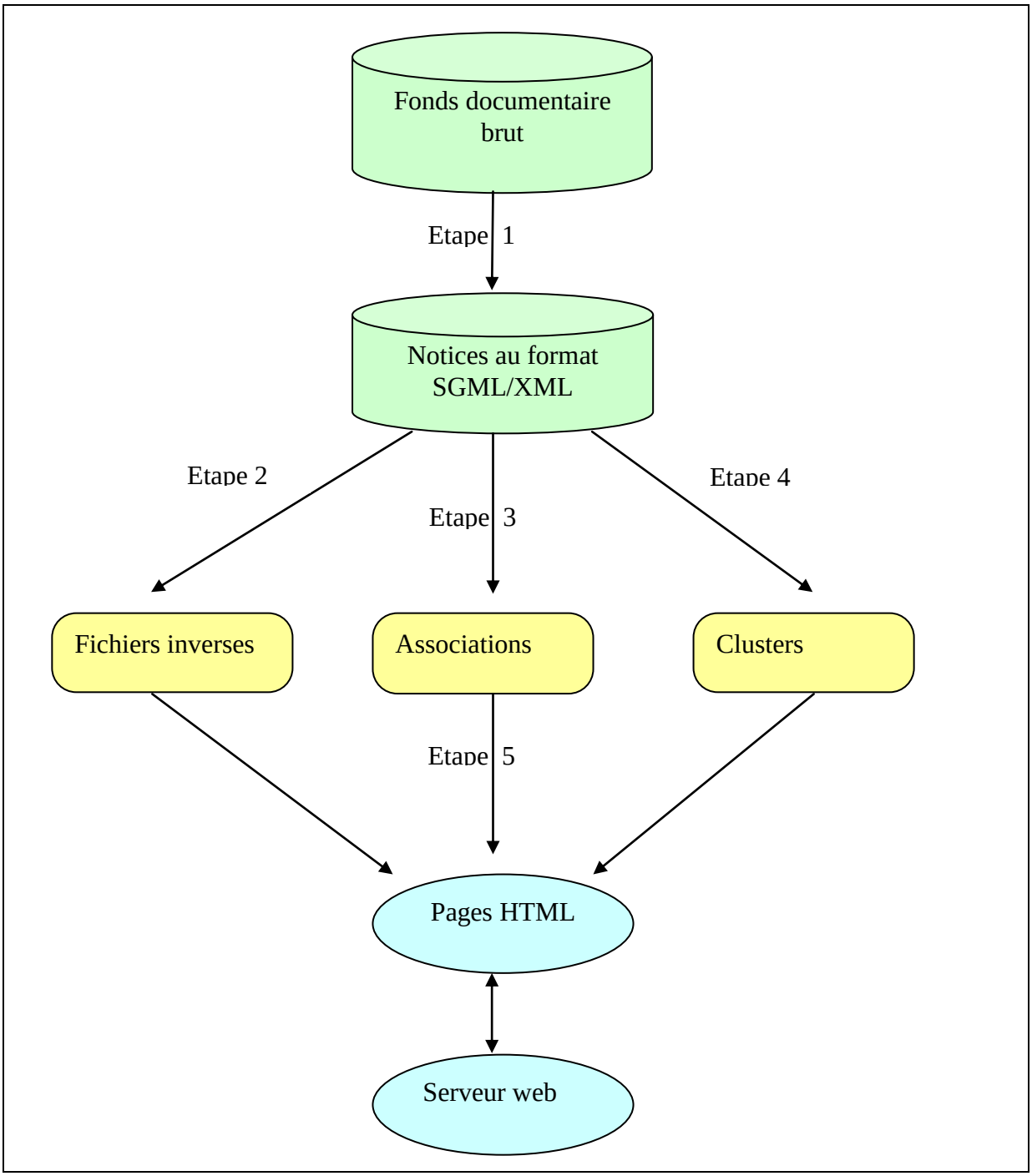

Figure 4 : Les étapes de la création du serveur.

Etape 1 : La normalisation.

Cette étape permet dans un premier temps de convertir le fonds documentaire utilisé au format XML*. Pour cela nous utilisons des fonctions spécifiques. Par exemple, la fonction " AirsToEd " utilisée dans le développement de l'application du centre de documentation de l'INRS permet le passage des notices du format AIRS* au format XML*.

Les données sont ensuite stockées sous la structure *Hierarchic File organization for Documentation* (HFD) dans le cadre de l'usage standard de DILIB*. Cette structure permet de traiter en standard jusqu'à un million de références bibliographiques réparties en cent répertoires de cent fichiers contenant eux mêmes jusqu'à cent notices (Figure 5).

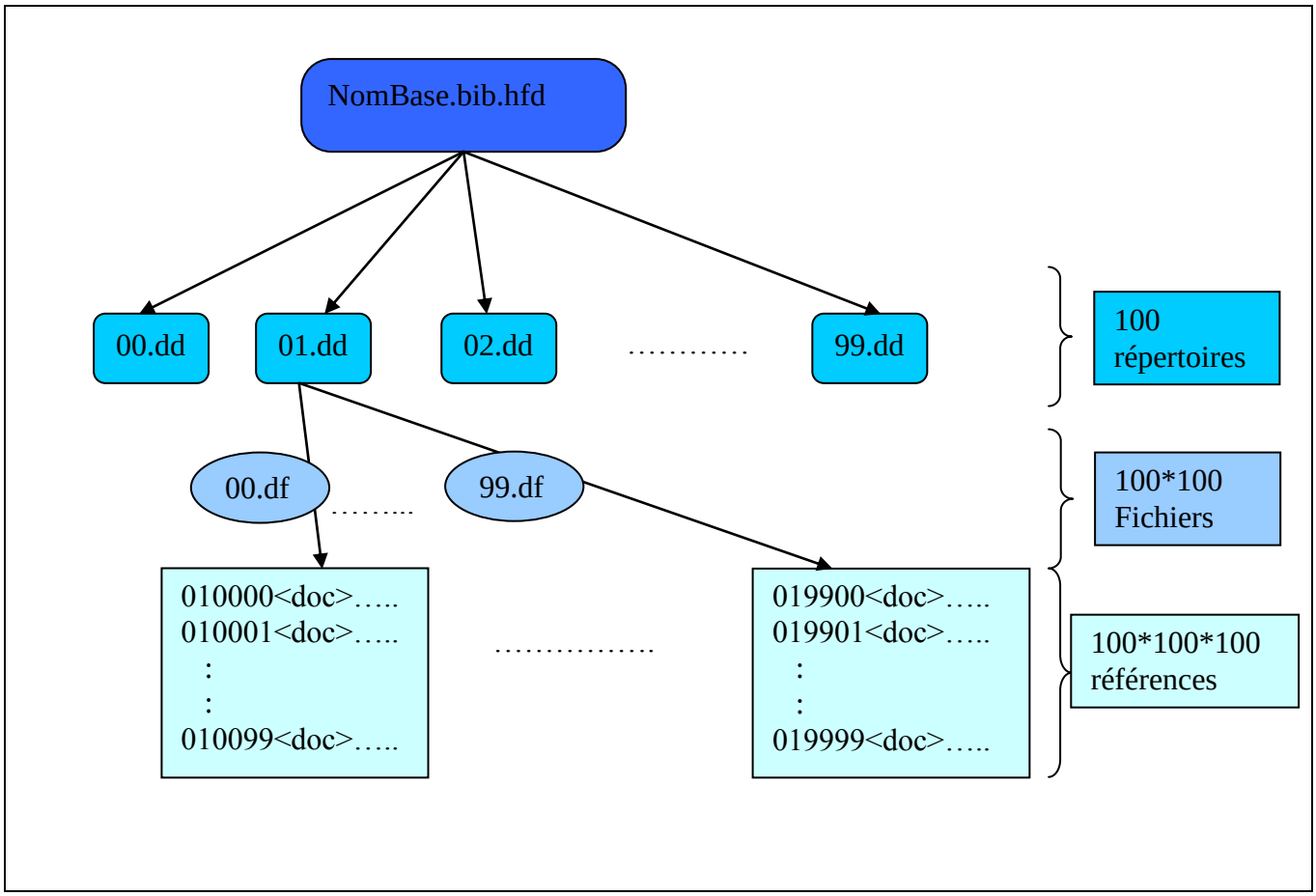

Figure 5 : Structure hiérarchique des fichiers HFD.

Les fichiers inverses d'index et d'association sont stockés suivant cette architecture et les données elles-mêmes rangées en XML dans les fichiers. Ce type d'organisation offre un accès plus facile aux données et donc un temps de traitement plus rapide. De plus, chaque fichier peut être traité indépendamment, ce qui permet d'éviter une saturation de la mémoire.

Etape 2 : Indexation des données.

Cette étape consiste en la création des fichiers inverses d'index (Figure 6). DILIB utilise de nombreux fichiers inverses afin de récupérer les informations nécessaires à la construction de graphes de navigation. Ces fichiers permettent d'accéder de façon directe aux objets reliés entre eux par une propriété commune (par exemple tous les mots-clés). Pour chaque terme indexé (`<kw>`), il est indiqué dans le fichier d'index le nombre de références dans lesquelles le terme est cité (`<f>`), ainsi que la clé de ces références dans les fichiers HFD (`<l>`).

```
<idx>
  <kw>Abnormalities, Multiple</kw>
  <lc>abnormalities, multiple</lc>
  <f>2</f>
  <l>
    <e>000144</e>
    <e>000185</e>
  </l>
</idx>
<idx>
```

Figure 6 : Extrait du fichier d'index en forme indentée.

La forme indentée de la figure 5 représente en fait le contenu d'une ligne du fichier d'index.

Etape 3 : création des associations.

Cette étape consiste à extraire les associations de termes issus du même champ dans les notices. Les fichiers inverses d'association contiennent des informations propres à chaque terme (<ti> et <tj>) ainsi que la fréquence de co-apparition des deux termes (<fij>).(Figure 7).

```
<assoc>
  <ti>
    <kw>Peptidylprolyl Isomerase</kw>
    <f>7</f>
  </ti>
  <tj>
    <kw>Carrier Proteins(c)</kw>
    <f>15</f>
  </tj>
  <fij>7</fij>
</assoc>
```

Figure 7 : Extrait du fichier d'associations en forme indentée.

Etape 4: construction des *clusters**.

Un *cluster* représente un thème. C'est un agrégat composé d'un ensemble de termes. La construction des clusters s'effectue à partir des associations précédemment établies selon la technique du simple lien (Figure 8).

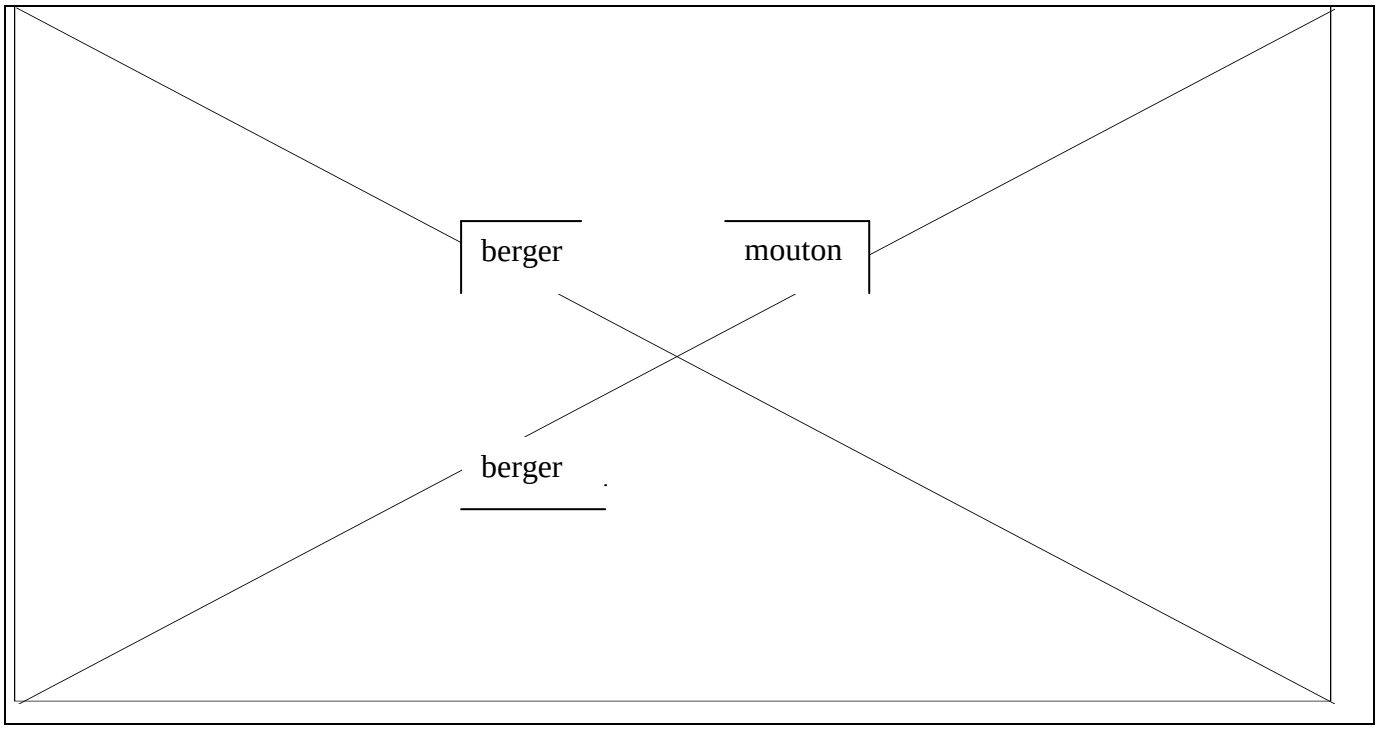

Figure 8 : Schéma de formation des clusters suivant la méthode du simple lien.[6]

L'association chien-milou ne peut être intégrée préférentiellement dans un des deux *clusters*. De plus il ne serait pas pertinent de regrouper les deux *clusters* car cette association à un faible poid par rapport aux autres. C'est une condition d'arrêt de formation du *cluster*. Cette association servira à établir une relation externe* entre les termes des deux *clusters.*

Etapes 5: Accès Internet .

Cette dernière étape permet la visualisation des résultats sur le web*. Tous les fichiers nécessaires à l'exécution des *Common Gateway Interface* (CGI*) sont créés lors de la génération du serveur d'investigation. Les CGI sont des scripts ou des programmes qui sont exécutés par un serveur http* lorsque le client en donne l'ordre (par un lien par exemple). Ceux ci permettent la création dynamique de pages HTML, le stockage ou la prise en compte d'informations (remplissage de formulaires par exemple). DILIB peut alors générer dynamiquement les différentes pages HTML du serveur afin de permettre sa consultation sur le web.

Remarque :

La consultation sur Internet des applications créées avec DILIB nécessite l'utilisation d'un serveur *HyperText Transfert Protocol* (*http**) pour la prise en compte de l'exécution des différentes commandes dynamiques (CGI) de DILIB. Apache est le serveur *http* le plus utilisé dans le monde de l'Internet de par sa robustesse et sa gratuité. Chaque serveur Apache est destiné à une application particulière (comme celles de DILIB par exemple). Il a donc fallu installer un serveur avec des paramètres spécifiques pour toutes les applications qui allaient être créées avec DILIB.

Les détails sur l'installation et le paramétrage de ces serveurs sont consignés dans le rapport de DESS* de Aude NEDELCOT [5].

5 L'application INRS Multi Data (IMD).

5.1 Présentation de l'application

L'application IMD est une application de DILIB installée au centre de documentation de l'INRS. Elle comprend des serveurs d'investigation générés à partir des bases de données de l'INRS dans l'environnement DILIB. L'intérêt d'un tel serveur est de proposer une interface d'interrogation conviviale intégrant différentes fonctions d'analyse de l'information (par exemple associations ou clusters de mots-clés).

L'application IMD est complémentaire de l'interface d'interrogation AIRS de l'INRS. Cette dernière présente des options d'interrogation plus complètes mais n'offre aucune fonctionnalité en terme d'analyse de l'information ou de navigation interactive.

5.2 Objectif des modifications.

Un constat : Nécessité de modifier 7 fichiers différents pour intégrer une nouvelle base dans l'application. Obligation de réimporter les corpus entiers réactualisés afin de procéder à une mise à jour.

Un besoin : Faciliter et systématiser la mise à jour de l'application.

La solution adoptée : Modulariser les composantes de l'application IMD.

L'application est constituée de trois couches logiques :

- La boîte à outils : DILIB
  Version v0.21
- Le châssis : IMD
  Il contient une description générique standard (un modèle) des bases susceptibles d'être accueillies, ainsi qu'une liste des bases avec leur localisation.
- Les éléments : Bases
  Chaque base est accompagnée de son fichier de description, venant compléter ou bien se substituer aux éléments de la description générique. Les opérations préliminaires sur chaque base sont définies à partir de la liste des bases.

Le but des modifications apportées pendant le stage est d'établir une indépendance maximale de ces différentes composantes, afin de pouvoir agir sur un niveau sans affecter les autres termes, la mise à jour se limitant alors à la modification de deux fichiers :

- la liste des bases,
- la description des bases.

Lors de la génération du serveur, deux types de traitement sont alors éffectués pour chaque occurrence de la liste des bases :

- des traitements généraux,
- des traitements spécifiques définit par la description des bases.

5.3 Méthodologie.

5.3.1 Répartition par couche logique et factorisation des traitements.

Tout d'abord, il a été nécessaire de bien identifier les différentes composantes élémentaires de l'application et de les affecter aux différentes couches logiques. Ce travail préliminaire a fait apparaître de nombreux traitements répétitifs d'une base à une autre.

Dans notre optique d'automatisation, il était nécessaire de factoriser les traitements au maximum. Cela permet de réduire le nombre de programmes et de fichiers nécessaires au lancement de l'application, et de faciliter la mise à jour. Ces traitements " généraux " constituent le châssis de l'application IMD.

5.3.2 Réécriture des scripts de génération selon le modèle en couches.

Pour réaliser ces traitements généraux, nous utilisons les informations propres à chaque base contenues dans deux fichiers :

- Le fichier IMD.desc.ed (Annexe 3) .

  Il contient deux types d'informations : d'une part la description générique des bases issues de AIRS, d'autre part les descriptions propres à chaque base permettant de prendre en compte leurs spécificités (par exemple le nom des champs à indexer ou encore le type de séparateur d'occurrences). Ce fichier sert à paramétrer la construction du serveur d'investigation (index, associations, clusters).

- Le fichier IMD.bases.list (Annexe 4) .

  Il contient la liste des bases avec pour chaque base, son nom, son intitulé, ainsi que la localisation de son fichier de description et de ses données. Cette dernière indication permet de prendre les données là où elles se trouvent sans avoir à les dupliquer inutilement dans le répertoire où l'on construit l'application. Lorsque l'on traite comme ici des corpus de 100000 notices, cela permet une économie non négligeable de l'espace disque.

On utilise ces informations propres à chaque base dans les traitements généraux en les passant dans des "variables de script*" principalement et quelques "variables d'environnement". Par exemple, si une opération nécessite l'utilisation du nom de la base, il suffit d'utiliser une variable $NAME qui prend le nom de la base que l'on est en train de traiter.

Remarque :

Le "format ed*" est un format pivot pour DILIB*. Lors du lancement de l'application, un fichier miroir de IMD.desc.ed au format XML est généré. C'est ce fichier miroir IMD.desc.sg que l'on utilisera pour accéder aux différentes informations contenues dans le fichier IMD.desc.ed.

5.4 Modifications apportées.

5.4.1 Référencement des bases

Pour pouvoir réaliser des traitements généraux en réalisant des itérations sur l'ensemble des bases, nous avons créé le fichier "IMD.bases.list". C'est une liste élaborée, sous forme d'un document XML (Figure 9), permettant d'accéder à différentes informations sur les bases, informations contenues précédemment dans les fichiers "liste_bases " et " BASES_DECLARATION ".

```
<imd>
      <base>
      <code>INOR</code>
      <data>/doc/airs/BASESN/INOR/BD/INOR.BIB.1</data>
      <desc>$SERVER_DIR/INOR.desc.ed</desc>
      <intitule>Base Bibliographique de l'unit&eacute;
      Suret&eacute; des Syst&egrave;mes Electroniques</intitule>
      </base>
</imd>
```

Figure 9 : Fichier IMD.bases.list

Exemple d'application 1:

```
SgmlSelect -g imd/base/code#INOR/../..-p @g1| SgmlSelect -g imd/base/data#
-p @g1 < IMD.bases.list
```

Cette commande permet de sélectionner le contenu textuel du nœud `<data>` pour la base INOR

5.4.2 Factorisation des traitements.

Un des principaux apports de cette restructuration consiste en l'élaboration de cinq *shells* permettant de généraliser certains traitements à toutes les bases :

- GenereShell.sh (Annexe 5) : Ce *shell** permet de générer les *shells* de préparation des bases (prétraitement et organisation des données sous forme de HFD*, un *shell* par base). Avant, il fallait écrire le shell de préparation de chaque base que l'on souhaitait inclure dans l'application.

- GenereMakeFile.sh (Annexe 6) : Ce *shell* permet de générer le fichier de *makefil*e* de chaque base. Le fichier de *makefile* permet de générer le serveur en fonction des informations contenues dans le fichier de description du serveur " IMD.desc.ed " (Annexe 3). Auparavant, il était nécessaire de compléter manuellement deux fichiers de *makefile* lors de l'ajout d'une base.

- Genere.def.path.input.sh (Annexe7) : Ce *shell* génère un fichier permettant de déclarer et d'exporter des variables intégrant certaines données du fichier de déclaration des bases (Annexe 4), en particulier les intitulés des bases affichés sur la page d'introduction du serveur.

- Genere.all.base.stat.sh : Ce *shell* sert à générer des statistiques sur le nombre de document indexés pour chaque base, ainsi que sur le nombre d'éléments dans leurs index.

- Genere.desc.cgi.sh : Ce *shell* permet de générer les fichiers nécessaires à l'exécution des *Common Gateway Interface* (CGI*).

Ces shells réalisent des traitements généraux à partir d'informations spécifiques à chaque base contenues dans les fichiers "IMD.bases.list" et "IMD.desc.ed".

Ils prennent tous le fichier " IMD.bases.list " en paramètre (Annexe 8, shell de lancement de l'application).

Exemple d'application 2 :

```
For name in `SgmlSelect -s imd/base/code# -p @s1 < $1`
      do
      traitements généraux
      done
```

$1 représente le paramètre "IMD.bases.list". Cette commande permet de réaliser une boucle de traitement pour l'ensemble des bases (pour plus de détails, se reporter aux listings des différents shells dans les annexes).

5.4.3 Complément sur "GenereShell.sh" et "GenereMakeFile.sh".

Nous avons vu dans la partie sur DILIB que le fichier de description (IMD.desc.ed) était converti au format SGML (IMD.desc.sg), et permettait la création du fichier de *makefiles* (IMD.mk) servant à la génération du serveur.

Ce fichier de description ne permet cependant pas d'introduire des listes de commandes spécifiques trop importantes. Pour l'applications IMD, les commandes spécifiques de prétraitement des bases n'ont donc pas été introduites dans le fichier de description. Ces commandes sont réalisées par des *shells* de prétraitement des bases (générés par le *shell* " GenereShell.sh "). Ceux-ci sont pris en compte par le *shell* " GenereMakeFile.sh " qui génère un fichier de *makefile* intégrant les commandes de prétraitement des bases, lequel fichier venant compléter le fichier de *makefile* standard. (Figure 10).

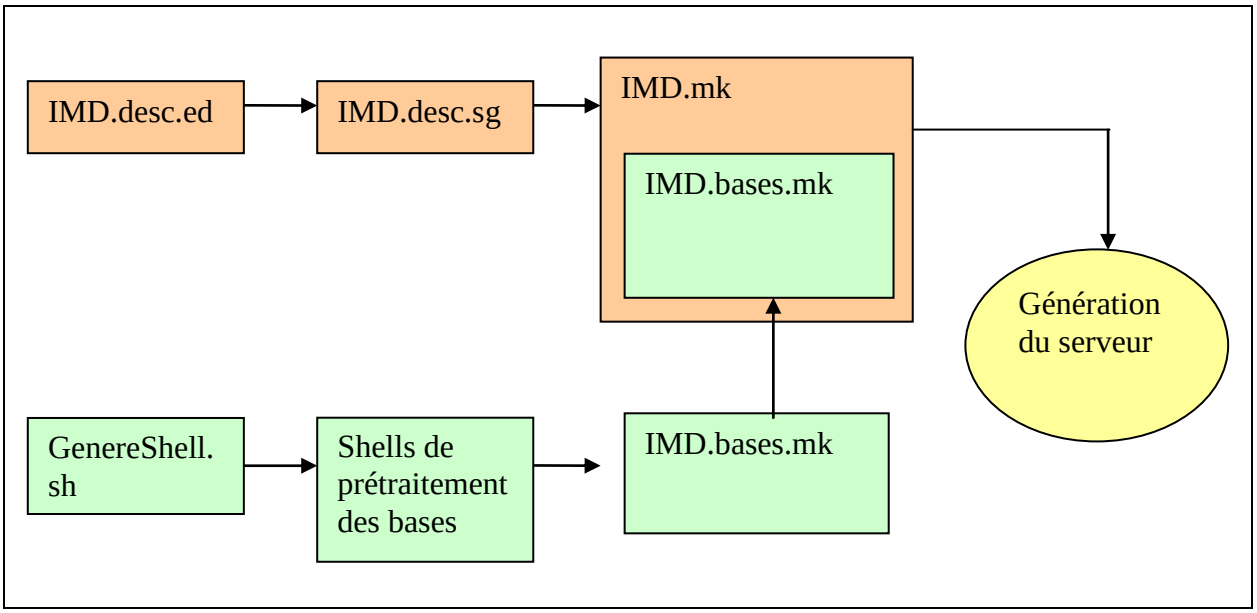

Figure 10 : Spécification du fichier de makefile.

5.4.4 Analogie avec le modèle objet.

La logique de l'architecture mise en place est similaire à la logique de programmation objet. Nous disposons d'une classe d'objet " base ". Chaque base documentaire de l'application représente une instance de cette classe et possède différentes propriétés constituant la structure des fichiers IMD.bases.list et IMD.desc.ed. Ces propriétés constituent la composante statique de la classe. Chaque shell de traitement général est équivalent à une "méthode*" que l'on peut appliquer sur cet objet. L'ensemble de ces méthodes constitue la composante dynamique de la classe.

D'autre part, il existe une notion d'héritage liée à la description des bases. Chaque base déclarée comme provenant de AIRS (`<base code=INOR from=AIRS>`) hérite des propriétés de la description générique (`<generic type=base code=AIRS>`), que sa propre description peut venir compléter ou remplacer.

Enfin la notion de polymorphisme est aussi présente puisque chaque shell de traitement constituant une méthode peut effectuer des opérations différentes en fonction de la structure de la base documentaire utilisée.

5.5 Problèmes rencontrés.

La principale difficulté a consisté à paramétrer des traitements généraux en tenant compte de la disparité de certaines bases par rapport au modèle générique décrit dans le fichier "IMD.desc.ed" (Figure 11) :

```
<generic type=base code=AIRS>
       <title>doc/TITR#
       <author>doc/AUTE/e#
       <index code=AUTE>
             <path>doc/AUTE/e#
             <minibibTag>AU
       </index>
       <index code=TITR text=EN>
              <path>doc/TITR
       </index>
       <index code=DESC lexicon=yes>
              <reject level=assoc>DE.reject.table
              <path>doc/DESC/e#
       </index>
 </generic>
```

Figure 11 : Fichier IMD.desc.ed, type générique.

En effet au départ de la restructuration, le problème ne semblait pas se poser. Toutes les bases avaient rigoureusement la même structure :

- Un champ de mot clé : DESC
- Un champ auteur : AUTE
- Un champ titre : TITR
- Un séparateur d'occurrence identique : " , " pour les champs multivalués (auteurs, mots-clés)

Nous pouvions donc facilement systématiser la prise en compte de ces éléments. Cette généralisation simplifiait énormément l'architecture de l'application. Avec les nouvelles bases à intégrer, nous sortions de ce cadre initial car celles-ci présentaient des structures différentes.

5.5.1 Le nom des champs.

Certaines bases ne présentaient pas les champs standards initialement utilisés pour générer le serveur. Pour garder une application homogène, nous avons analysé ces bases afin de sélectionner des champs intéressants correspondants à ceux du modèle (c'est- à-dire DESC, AUTE, et TITR). En accord avec Françoise Grandjean nous avons décidé d'indexer les champs suivants :

- Base NFOR : ORGA (Organisme à la place de Auteurs).
- DISC (Disciplines à la place de Mots clés).
- SIGLE (à la place du Titre).
- Base INOR : ORG_ORIG (Organisme d'origine pour Auteurs).
  THEME_FR (pour mots clés).
  PAYS_FR (pour titre).
- Base PDOC : AUTEUR (à la place de AUTE).
  DE1 (à la place de DESC).
  TITRE (à la place de TITR).

Il a donc fallu utiliser les informations contenues dans ces champs pour générer le serveur en les intégrant dans les intitulés classiques (DESC, AUTE, TITR). A ce niveau, la modularisation des différentes composantes de l'application était indispensable . C'est dans le fichier de description des bases, " IMD.desc.ed ", que l'on va indiquer les champs que l'on désire indexer ainsi que leurs chemins d'accès dans les notices (Figure 12, exemple pour la base INOR).

```
<base code=INOR from=AIRS>
     <title>doc/PAYS_FR#
     <authors>doc/ORG_ORIG#
     <input type=command>bidon
     <index code=AUTE>
        <path>doc/ORG_ORIG#
        <sep>;
      </index>
     <index code=DESC>
        <path>doc/THEME_FR/e#
        <sep>;
      </index>
 </base>
```

Figure 12 : Fichier IMD.desc.ed, base INOR.

On construira ici un index DESC contenant les occurrences du champ THEME_FR. Pour que ces particularités soient prises en compte par les *shells* qui génèrent les *shells* de préparation des bases (GenereShell.sh) et le fichier de *makefiles* (GenereMakeFile.sh), nous utilisons des variables d'environnement prenant leur valeur dans le fichier de description en fonction de la base considérée.

Exemple d'application 3 :

```
Cat $SERVER_ROOT/IMD.desc.sg | SgmlSelect -s ✂
server/base@code=INOR/index@code=DESC/path# -p @s1 | awk -F/ '{print $2}'
```

Cette commande permet de sélectionner le nom du champs à indexer sous DESC pour la base INOR dans le fichier IMD.desc.sg (le caractère ✂ signifie que la ligne de commande est coupée pour les besoins de la rédaction).

La difficulté provient de la coexistence de deux informations, d'une part le nom du champ que l'on veut indexer (qui permet de le retrouver dans la notice d'origine et d'afficher correctement celle-ci par l'intermédiaire du serveur), et d'autre part le code de cet index pour le serveur qui permet une homogénéité apparente de l'application. Ces deux informations étant utilisées à différents endroits, il faut prendre garde à bien utiliser celle qui convient à chaque opération (et particulièrement dans les fichiers de *makefiles**), ce qui souligne une fois encore la nécessité de la modularisation de l'application.

5.5.2 Les séparateurs d'occurrences.

Pour la disparité des champs, la restructuration établie permettait de remplacer facilement le contenu d'un champ par un autre car cela ne nécessitait pas de traitements supplémentaires (juste une sélection appropriée de l'information en fonction de l'opération réalisée). Pour les séparateurs d'occurrences, le problème rencontré est plus gênant. Lorsque la base à traiter utilise le " ; " comme séparateur d'occurrences, il se pose un gros problème à cause du caractère spécifique de ce " ; " dans le codage SGML des caractères. Or le prétraitement des bases s'effectue sur des données au format SGML. Si nous utilisons le point-virgule comme séparateur, tous les mots contant des caractères spéciaux sont coupés.

Nous avons donc rajouté une balise `<sep>` dans le fichier de description qui signifie le séparateur d'occurrence utilisé. Lorsque ce séparateur est un " ; ", le shell de prétraitement le remplace par une " , ", ce qui revient à traiter un cas particulier pour le traitement de certaines bases. Ce traitement est réalisé pour les bases INOR et PDOC, possédant des champs de la forme : CHAMPS : terme1 ; terme2 ; terme3 ; terme4 …

Exemple d'application 4 :

```
DESCSEP=`cat $SERVER_ROOT/IMD.desc.sg | SgmlSelect -s ✂
Server/base@code=$name/index@code=DESC/sep# -p @s1`
```

Cette commande, placée dans une boucle de traitement sur la liste des bases, permet de sélectionner le séparateur d'occurrences des descripteurs pour la base $name et de l'affecter à la variable DESCSEP. Si le séparateur est un " ; ", la commande " `sed 's/ ; / , /g'` " permet alors de remplacer les " ; " par des " , " avant de poursuivre les traitements.

5.6 Perspectives d'évolution.

5.6.1 Le fichier de description (IMD.desc.ed).

Deux évolutions sont envisagées concernant ce fichier :

- Il serait intéressant de le compléter par une balise " shell ", contenant les différentes lignes de commande utilisées dans les shells de prétraitement des bases (Figure 13, exemple pour INOR). Cela demanderait de fournir d'avantage d'informations dans le fichier de description, mais permettrait de générer plus facilement des shells de prétraitement propres à chaque base.

```
cat  /applis/dps/INRS/IMD.test/Text/INOR.BIB.0  \
   | removeCR \
   | AirsToEd  \
   | SgmlCharSetTr -f ed \
   | MiniBibFromEd -T \
   | sed 's/ ; / , /g'| SgmlTextProc -P strtok -is "," -F first doc/THEME_FR -R -
G THEME_FR -E e  \
   |    SgmlTextProc    -P    tableReplace    -F    all    doc/DESC/e#    -R    -t
/applis/dps/INRS/IMD.test/DE.syn.table \
   | sed 's/ ; / , /g' | SgmlTextProc -P strtok -is "," -F first doc/ORG_ORIG -R -
G ORG_ORIG -E e  \
   | DamHfdBuild -h /applis/dps/INRS/IMD.test/Server/INOR.bib
```

Figure 13 : shell de prétraitement de la base INOR.

- Pour mener à terme la modularisation, il faudra enfin diviser le fichier IMD.desc.ed en plusieurs parties, une contenant la description générique, et une pour la description de chaque base. Le fichier IMD.bases.list a d'ailleurs été construit en vue d'accueillir dans une balise " desc " la localisation du fichier de description propre à chaque base.
  Cela doit pouvoir se faire sans affecter le lien d'héritage entre la partie générique et celles propres à chaque base. L'évolution de DILIB va dans ce sens et la version v0.3 devrait permettre cet aboutissement.
  Ainsi chaque base de la liste sera associée à un fichier de données et à un fichier de description.

5.6.2 Gestion des erreurs.

Incontestablement, il reste un gros travail à faire concernant la gestion des erreurs lors du déroulement du shell de génération du serveur. En effet, les différents shells lancés ne réalisent aucun test d'existence des informations nécessaires à leur exécution, ce qui rend très délicate la gestion des erreurs. Il faudra donc modifier tous les shells pour qu'ils effectuent des tests systématiques avec des sorties immédiates au cas où des informations sont manquantes.

5.7 Bilan des restructurations.

L'objectif de la restructuration était de faciliter l'ajout de nouvelles bases dans l'application, ainsi que sa mise à jour. Pour l'utilisateur, cela est totalement transparent, mais la comparaison des protocoles de mise à jour avant et après les modifications permet de mieux comprendre le bien fondé de ces restructurations.

5.7.1 Ajout d'une base.

Avant la restructuration.

Auparavant, l'importation d'une nouvelle base dans l'application nécessitait de modifier un à un sept fichiers différents : (Annexe 9) :

- Import d'un fichier contenant les données de la base dans le répertoire " Text ". Cela n'est plus nécessaire, les données sont prises là où elles sont stockées.

- Ecriture d'un *shell* de prétraitement et d'organisation des données de la base, à intégrer dans le répertoire " Prog ". Cela n'est plus nécessaire car ces shells sont générés par le " GenereShell.sh " et placés automatiquement dans le répertoire " Prog ".

- Ajout du nom de la base dans le fichier " liste_bases ". Les données de ce fichier et celles du fichier " BASE.DECLARATION " sont rassemblées dans le fichier " IMD.bases.list ".

- Complétion du fichier " BASE.DECLARATION " (nom de la base, chemin d'accès aux données, intitulé de la base).

- Complétion du fichier IMD.index.mk.

- Complétion du fichier IMD.bib.mk. Les données contenues dans les deux fichiers de *makefile* (IMD.index.mk et IMD.bib.mk) sont maintenant rassemblées en un seul fichier (IMD.bases.mk) généré lors du shell de création du serveur.

- Complétion du fichier IMD.desc.ed.

Après la restructuration.

Grâce à la factorisation des traitements au sein de l'application, l'ajout d'une base se fait maintenant en complétant simplement deux fichiers qui regroupent toutes les informations spécifiques aux différentes bases, nécessaires pour effectuer les traitements généraux.

- Complétion du fichier IMD.bases.liste

- Complétion du fichier IMD.desc.ed

5.7.2 Tableau récapitulatif.

Le tableau de la Figure 14 permet un comparatif rapide des deux protocoles de mise à jour en soulignant les tâches effectuées automatiquement.

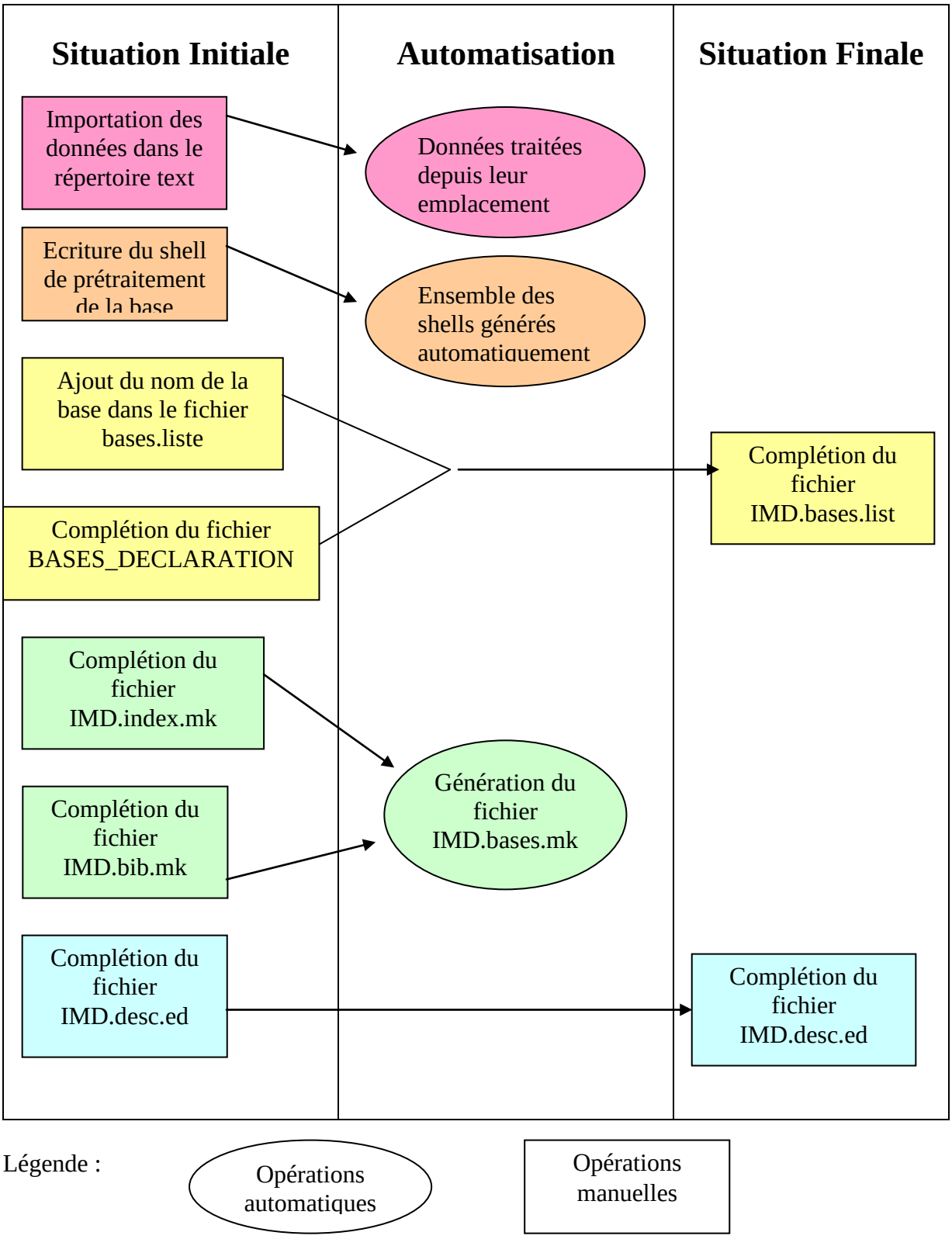

Figure 14 : Comparatif des protocoles d'ajout d'une base.

5.7.3 Mise à jour de l'application.

La restructuration établie permet aussi de faciliter le lancement du serveur, et sa mise à jour (prise en compte des nouvelles notices pour les bases déjà implémentées). Cela marque un grand pas vers l'automatisation de la mise à jour. Avant les modifications, la génération du serveur se faisait en trois étapes :

- Lancement du *shell* d'initialisation (initialisation des variables d'environnement et création de quatre répertoires Text, Prog, bin, Server)

- Import des données des différentes bases dans le répertoire Text, et import des *shells* de prétraitement de chaque base dans le répertoire Prog.

- Lancement du *shell* de génération

Le répertoire Text n'étant plus utilisé, et le répertoire Prog étant alimenté automatiquement lors du lancement du shell de génération, il n'y a plus de raisons de réaliser le lancement en deux étapes. D'autre part le fait d'aller chercher les données sur le serveur AIRS de l'INRS permet à la fois une économie des ressources (pas de duplications inutiles), et une mise à jour facilitée puisqu'il suffit de relancer le serveur pour prendre en compte les nouvelles notices (ce qui nécessitait auparavant de réimporter les données mise à jour dans le répertoire Text).

Nous avons donc fait un grand pas vers l'automatisation.

Remarque :

Les restructurations mises en place pour l'application IMD ont demandé plus de temps que celui prévu au départ pour travailler sur cette application. Il a fallu environ un mois pour les mettre en place alors que cela ne devait initialement pas durer plus de trois semaines. Le temps passé sur cette première partie m'a cependant permis de bien me familiariser avec DILIB et les travaux sur l'application " Transcriptome " s'en sont trouvés grandement facilités. En effet la première partie du stage sur l'application IMD m'a permis d'étendre mes connaissances des commandes UNIX*, des commandes propres à DILIB et de la structure des applications de DILIB. Cela m'a permis de régler beaucoup plus facilement les problèmes liés aux attentes de Bertrand Rihn pour l'évolution de l'application " Transcriptome ".

6 L’application " Transcriptome ".

6.1 Origine de l’application.

6.1.1 L’amiante et le mésothéliome.

De par ses qualités d’isolant et ses propriétés de résistance, l’amiante fut largement utilisée au cours des cinquante dernières années dans l’industrie et le bâtiment. Certaines études ont permis de mettre en évidence le caractère cancérogène des fibres d’amiante. Il a été prouvé que leur inhalation était responsable chaque année en France de nombreuses atteintes pulmonaires (cancer du poumon, fibroses pulmonaires, cancer de la plèvre). Le mésothéliome est le cancer de la plèvre (enveloppe du poumon) résultant en général d’une exposition professionnelle à l’amiante.

6.1.2 L'expression Génique.

L'ensemble des gènes* est défini par le génome*. Dans une cellule, un certain nombre de gènes sont activé en fonction du type de la cellule et de son environnement. Cet état d'activation des gènes est appelé l'expression des gènes. Les gènes exprimés donnent lieu à la synthèse d'ARN messagers* leur correspondant, cette étape s'appelle la transcription. L'ensemble des transcrits (ARN messager) est appelé transcriptome*. Ces ARN messagers seront ensuite traduit en protéines qui formeront le protéome*.

6.1.3 La technique des puces à ADN.

Afin de mieux comprendre la pathogénie du mésothéliome pleural humain (cancer de la plèvre) et de décrire plus précisément ses mécanismes moléculaires, l’équipe de Bertrand Rihn (INRS) a entrepris l’étude des gènes impliqués dans ce cancer [7].

Différentes techniques de biologie moléculaire ont été utilisées à cet effet, en particulier la technique des puces à ADN (Figure15). Cette technique récente permet de tester l’état d’activation de milliers de gènes simultanément alors que les techniques antérieures ne permettaient qu’une étude parcellaire portant sur une dizaine de gènes.

L’expérimentation a consisté à comparer l’expression de 7000 gènes de cultures de cellules saines et de cellules malignes de la plèvre par quantification des ARN messagers.

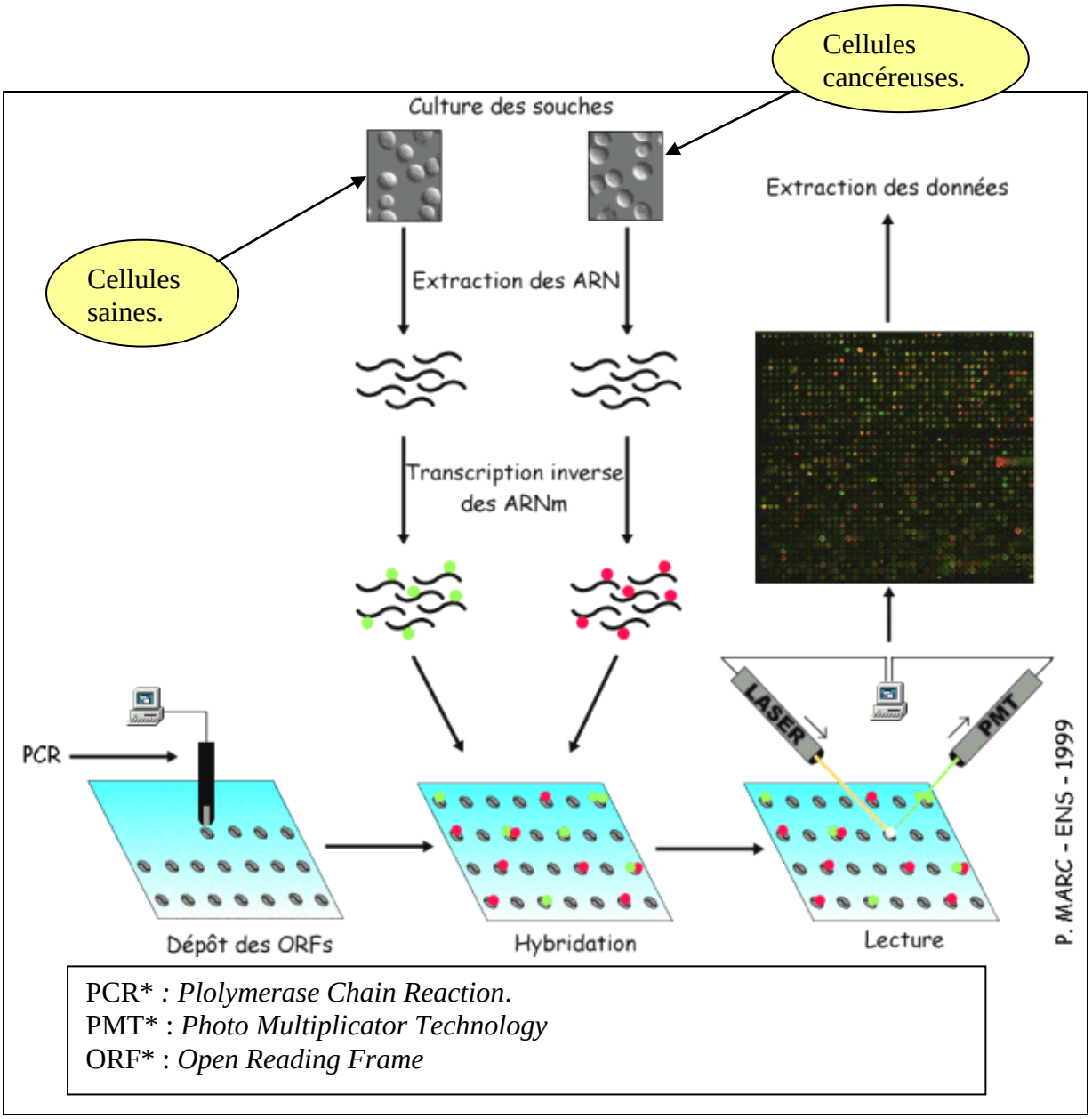

Figure 15 : Protocole expérimental de la technique des puces à ADN.[8]

Etape 1 : Extraction de l'ARN messager* (ARNm).

L'ARN messager est extrait des cellules souches saines et des cellules cancéreuses. Plus un gène est exprimé, plus la quantité d'ARN messager* correspondant est importante.

Etape2 : Transcription inverse* de l' ARN messager.*

La transcrition inverse* consiste à recréer une séquence d'ADN complémentaire* à partir d'ARN messager. Cet ADN complémentaire est ensuite marqué par des fluorochromes différents pour les cellules saines (marquage vert) ou cancéreuses (marquage rouge).

Etape 3 : Hybridation de l'ADN complémentaire* (ADNc)

L'ADN complémentaire marqué est ensuite introduit dans des puits où différentes séquences d'ADN connues ont été placées au préalable. Si la séquence d'ADN du puit est complémentaire d'une séquence d'ADNc introduite dans ce même puit, il y a hybridation.

Etapes 4 : Expression différentielle.

L'ADN complémentaire étant marqué, il est possible de quantifier l'hybridation par des mesures de fluorescence. On obtient ainsi une fluorescence d'intensité variable en fonction du degré de marquage (vert pour les cellules saines, rouge pour les cellules cancéreuses). Il est ainsi possible de quantifier l'expression des gènes dans les cellules cancéreuses par rapport à leur expression dans les cellules saines.

Le rapport de l'expression d'un gène dans les cellules cancéreuses à celle dans les cellules normales est appelé l'expression différentielle du gène. Si l'expression du gène est supérieure dans les cellules cancéreuses, l'expression différentielle est affectée d'un signe plus, d'un signe moins dans le cas contraire. On estime que l'expression différentielle d'un gène est significative si elle est supérieure à deux en valeur absolue (ce qui représente environ 400 gènes sur 7000 dans le cas de cette étude).

6.1.4 Du Transcriptome au corpus bibliographique.

Les résultats de cette étude ont été rassemblés dans un tableau permettant de visualiser, entre autres, le niveau d'expression de chaque gène dans les cellules saines et cancéreuses de la plèvre (tableau Figure 16). Ces résultats des expériences sur les puces à ADN* permettent également de comprendre certains mécanismes génétiques de la transformation cancéreuse des cellules normales de la plèvre en cellules malignes de mésothéliome. Cette étude a par ailleurs permis d'expliquer la relative résistance du mésothéliome à la chimiothérapie et à la radiothérapie.

Afin d'exploiter au mieux les résultats de l'étude, il a été envisagé une analyse bibliométrique réalisée avec la plate-forme DILIB et exploitant des données bibliographiques issues de *Pubmed* en relation avec les gènes mis en évidence par l'expérimentation. L'utilisation d'outils infométriques tels que DILIB permet d'analyser des corpus bibliographiques importants qui ne sont pas exploitables manuellement. Cela est d'autant plus intéressant en génomique compte tenu de l'étendue des bases de données dans ce domaine.

L'application mise en place à l'INRS et décrite dans ce rapport contient deux serveurs d'investigation, un pour les gènes sur-exprimés (expression différentielle > 1.9), et un pour les gènes sous-exprimés (expression différentielle < -1.6). Cette application de DILIB a été nommée " application Transcriptome ".

Expression Différentielle	Molécule pour laquelle code le gène	**Numéro d'accès dans la base Genbank**
32,2	plasminogen activator inhibitor, type II (arginine-serpin)	**M31551**
19,2	fatty acid binding protein 5 (psoriasis-associated)	**AA972250**
18,6	deiodinase, iodothyronine, type II	**AF007144**
11,9	serum-inducible kinase	**AF059617**
11,3	ESTs	**N35555**
10,6	Human EV12 protein gene	**M55267**
9,1	annexin A1	**X05908**
8,1	ornithine decarboxylase	**M81740**
7,3	pentaxin-related gene, rapidly induced by IL-1 beta	**M31166**
7,2	integrin, alpha 6	**X53586**
6,3	deiodinase, iodothyronine, type II	**AF093774**

Figure 16 : Exemple de quelques gènes sur-exprimés dans le mésothéliome.

6.2 Les objectifs.

Les modifications apportées sur l'« application Transcriptome » concernent deux axes principaux :

- La rédaction d'une procédure utilisateur.
- L'optimisation du serveur en matière d'analyse de l'information.

6.2.1 La procédure utilisateur.

Le but de cette procédure est de rendre autonome l'utilisateur afin qu'il puisse lui même générer une nouvelle application lui permettant d'exploiter d'autres résultats de recherche. Pour cela, il est nécessaire d'automatiser en partie l'extraction des corpus et le tri des index comme nous le verrons dans les pages suivantes.

6.2.2 L'optimisation du serveur.

L'extraction des corpus bibliographiques s'effectue à partir des numéros d'accession *Genbank* des gènes sur-exprimés ou sous-exprimés dans les expériences sur les puces à ADN* réalisées à l'INRS. Ces numéros sont fournit par le fabricant des puces à ADN (ici la société *Incyte Pharmaceuticals*). Il était souhaitable de pouvoir relier chaque gène exprimé différentiellement dans les puces à ADN (et donc chaque numéro d'accession *Genbank*) aux mots clés associés dans les références *Medline* et chaque mots clés au(x) gène(s) associé(s).

D'autre part il fallait mettre en place un outil simple de comparaison des vocabulaires des deux corpus (gènes sur-exprimés et gènes sous-exprimés).

6.3 Extraction des corpus bibliographiques.

6.3.1 Schéma initial.

Le schéma initial d'extraction des corpus était constitué de deux étapes (Figure 17) :

- Interrogation de *Genbank*.
- Liens vers *Pubmed*.

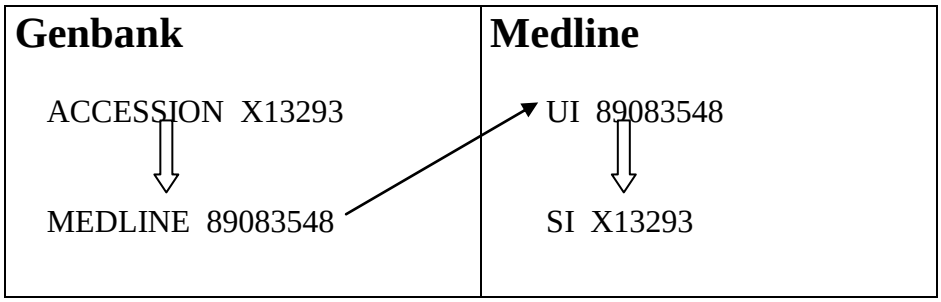

Figure 17 : Schéma d'extraction des références bibliographiques.

Cette méthode de recherche ne permettant pas d'obtenir des corpus bibliographiques assez conséquents, une étape intermédiaire est rajoutée, exploitant l'option d'interrogation *nucleotide neighbors* de *Genbank*. Cela permet d'obtenir les notices *Genbank* de gènes considérés comme voisins des gènes recherchés dans la base *Genbank*. Chacune de ces notices pouvant faire référence à des notices *Medline*, cela permet d'étendre le champ d'investigation des recherches (Figure 18).

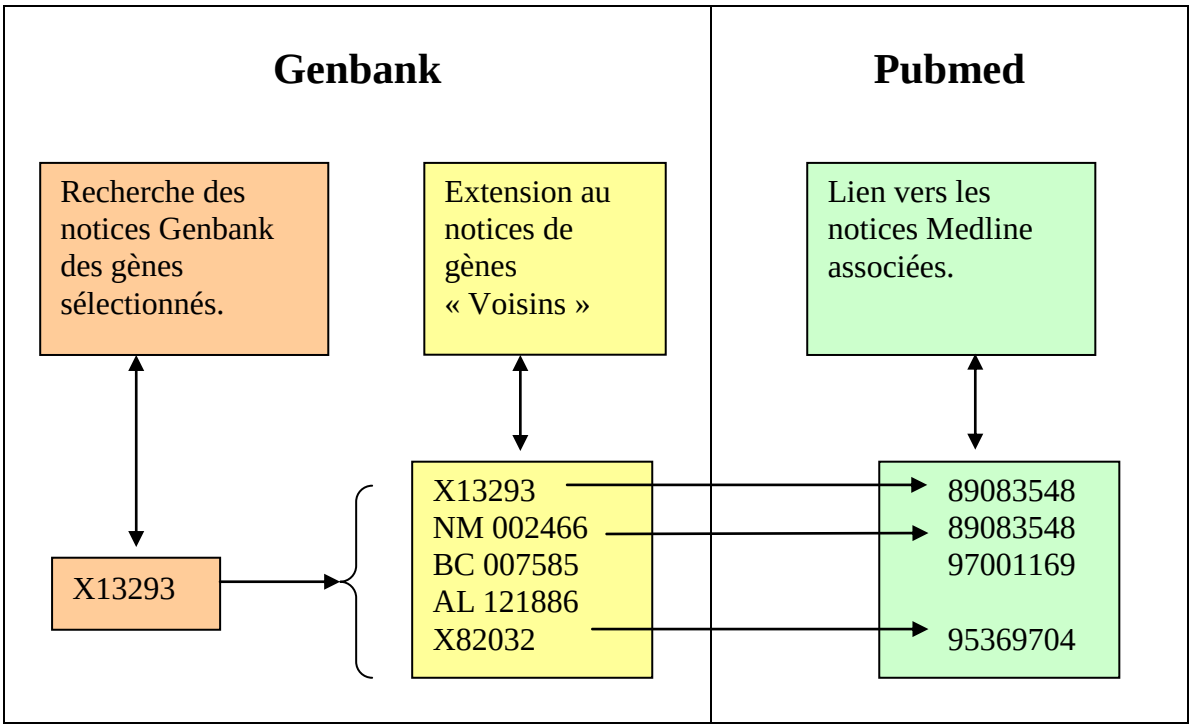

Figure 18 : Exemple d'extraction de notices medline en rapport avec le gène X13293

Ce schéma montre bien que nous obtenons ainsi trois références *Medline* à partir du gène X13293, au lieu d'une seule.

6.3.2 Nouveau schéma d'extraction.

L'élargissement des recherches par les *nucleotide neighbors* supprime la possibilité de pouvoir relier les mots clés des références *Medline* aux numéros *Genbank* de départ. Pour mieux le comprendre, reprenons l'exemple de la Figure 18. Le gène X82032 permet d'élargir la recherche à la notices *Medline* 95369704. Seulement rien ne permet de relier cette notice au gène X13293. En effet nous ne savons pas comment sont générés ces liens de " voisinage" qui eux, ne figure pas dans les notices *Genbank*. Cet intermédiaire a donc été abandonné d'autant plus que nous n'avions aucune assurance sur la pertinence scientifique de ces liens, ce que nous a confirmé Bertrand Rihn. Nous avons donc adopté un schéma d'extraction plus simple, permettant d'intégrer plus facilement cette étape préalable à la génération du serveur dans une procédure utilisateur. La recherche est effectuée directement dans le champs *Secondary Source Identifier* (SI) de *Medline* qui contient les numéros d'accession *Genbank* sur le site. Cela permet de recueillir plus de notices intéressantes qu'en effectuant la recherche dans *Genbank* (Figure 19).

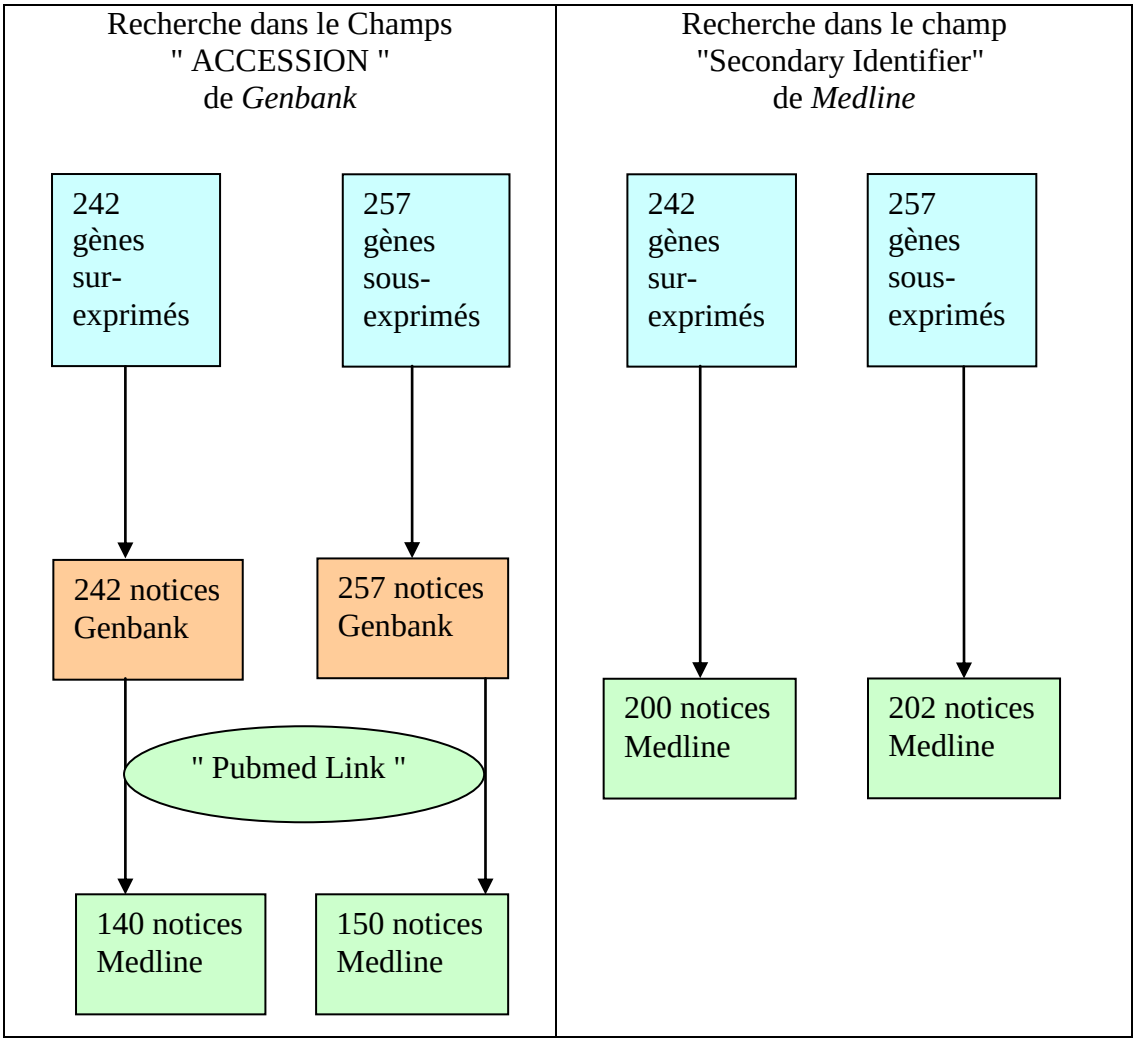

Figure 19 : Comparaison recherche directe et indirecte.

6.4 Procédure utilisateur.

Ce mode d'emploi (Annexe 10) doit permettre à un utilisateur n'ayant pas de connaissance particulière de DILIB de générer son propre serveur d'investigation. Néanmoins une connaissance des commandes de base UNIX* est requise pour être autonome.

6.4.1 Génération des requêtes et extraction des corpus.

Comme nous l'avons vu dans le paragraphe précédent, le schéma d'extraction des corpus bibliographiques a été simplifié. Cette simplification nous a ensuite permis de semi-automatiser la recherche des notices pour que cette étape préalable ne constitue pas un obstacle pour l'utilisateur.

Pour cela nous avons écrit un *shell* qui génère les requêtes utilisées pour l'extraction des deux corpus à partir de parties du tableau des résultats de puces à ADN* (Figure 20).

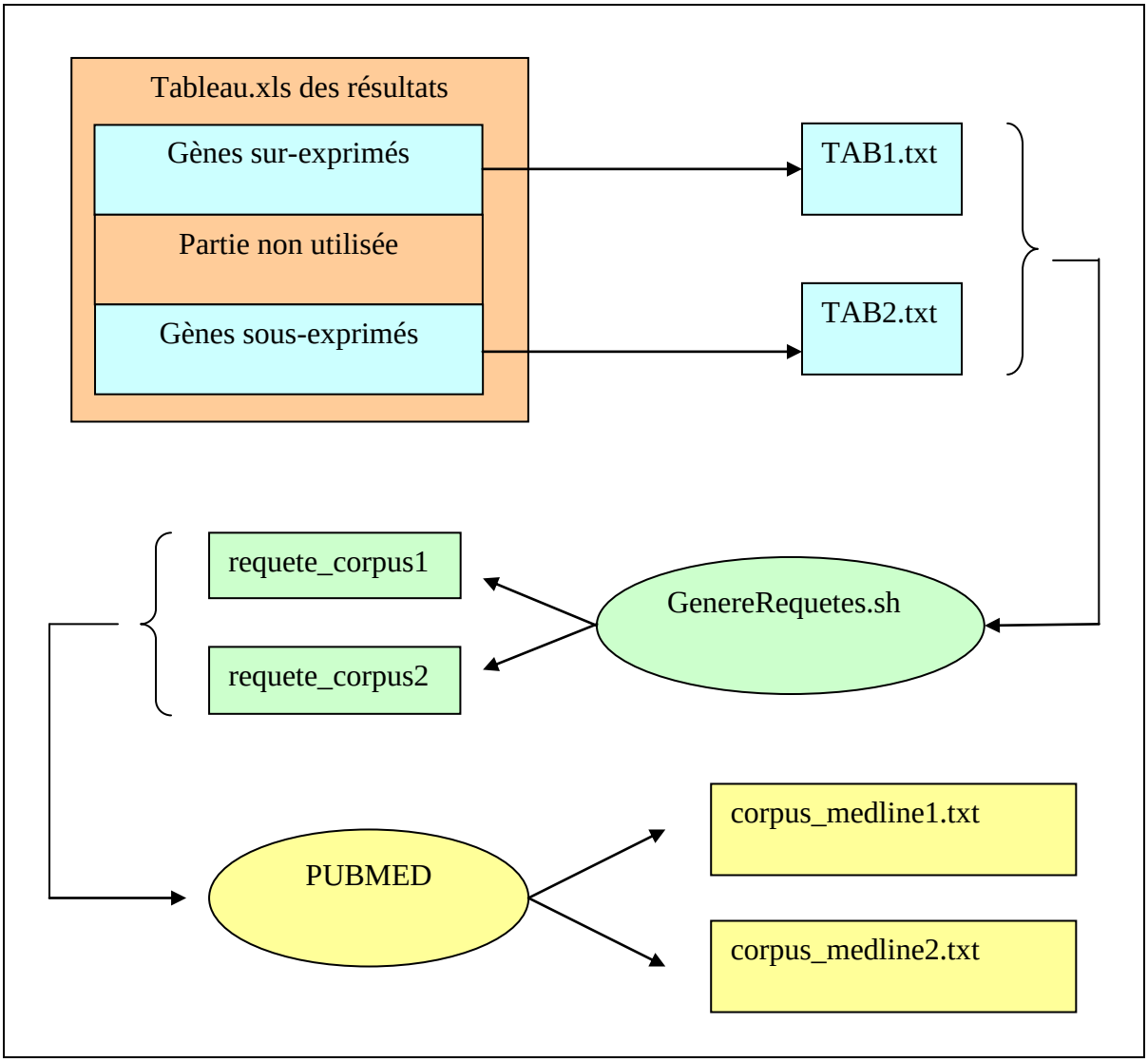

Figure 20 : Méthode d'extraction des notices Medline.

Le *shell* "GenereRequetes.sh " (Annexe 11) crée les requêtes à partir des fichiers TAB1.txt et TAB2.txt. Pour les utiliser, il suffit de faire un "copier-coller " depuis les fichiers " requete_corpus1 " et " requete_corpus2 ", vers le site de recherche (*Pubmed*).

Notons que cette opération demande environ 15 minutes, là où l'ancienne procédure nécessitait une à deux journées de travail. En effet la recherche s'effectuait de façon parcellaire par tranche de vingt gènes. Il fallait donc écrire de façon manuelle une vingtaine de requêtes soumises une à une, et sauvegarder les résultats de chaque requête avant de tout concaténer pour avoir un corpus complet [1].

6.4.2 Tri des index.

Une fois le serveur d'investigation généré, il est nécessaire de trier les différents index afin de ne garder que les termes intéressants. Pour ne pas prendre en compte les termes " vides ", trop génériques, ou simplement jugés sans intérêt lors de la génération du serveur, DILIB offre la possibilité d'utiliser des tables de rejet (ou anti-dictionnaires*) déclarées dans le fichier de description (Figure 21). La base de code " gene1 " correspond aux gènes sur-exprimés et la base " gene2 " aux gènes sous-exprimés.

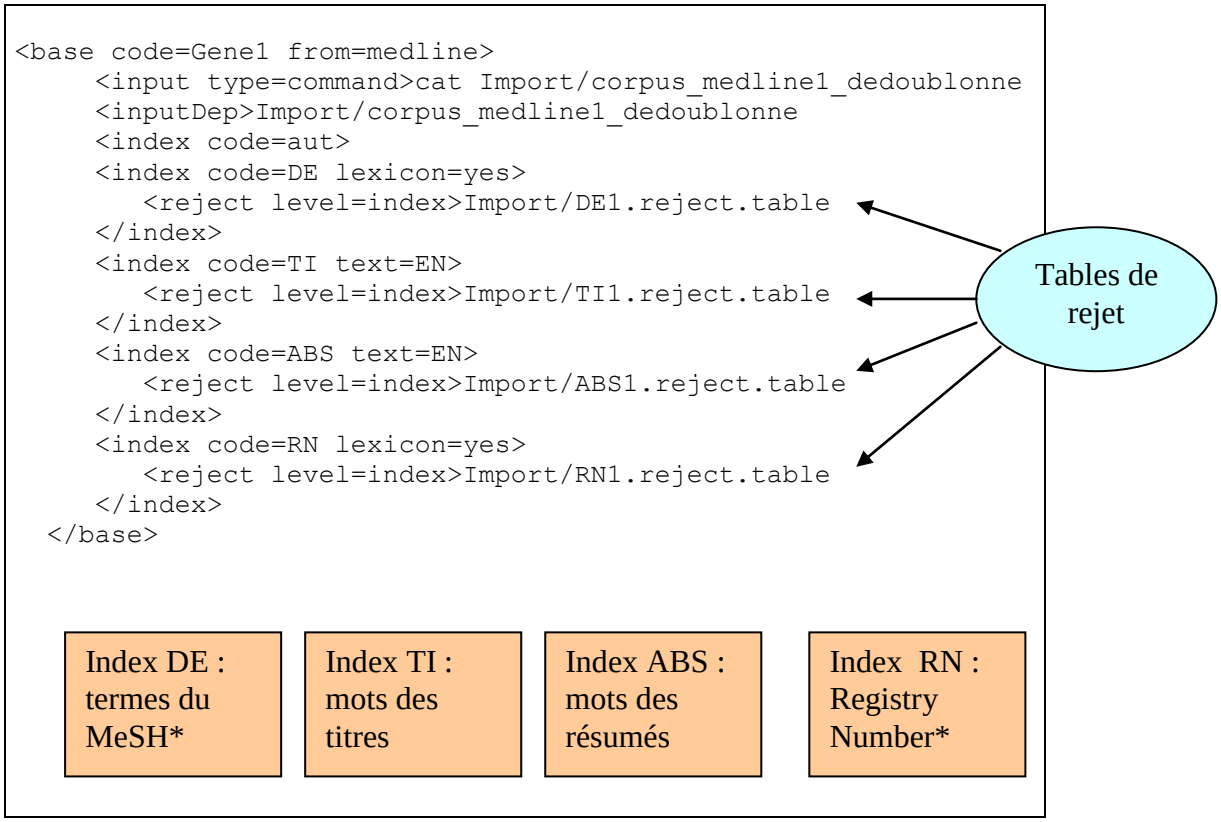

Figure 21 : Extrait du fichier de description pour le corpus lié aux gènes sur-exprimés.

Il est possible d'alimenter ces anti-dictionnaires* directement avant de lancer la création du serveur. Il est cependant difficile d'être exhaustif en procédant de la sorte et il est en général nécessaire de refaire un tri manuel après la génération. Pour que ce tri ne s'avère pas trop fastidieux pour l'utilisateur, nous avons écrit un *shell* qui permet de rendre cette opération semi-automatique (Figure 22). Il est possible d'utiliser ce programme pour trier les quatre principaux index et alimenter les tables de rejet correspondantes :

- termes du Medical Subject Headings (MeSH*), alimentation des tables de rejet "DE1.reject.table" et "DE2.reject.table",
- mots des titres, alimentation des tables de rejet "TI1.reject.table" et "DE2.reject.table",
- mots des résumés, alimentation des tables de rejet "ABS1.reject.table" et "ABS2.rejct.table",
- *registry number*, alimentation des tables de rejet "RN1.rejet.table" et "RN2.reject.table".

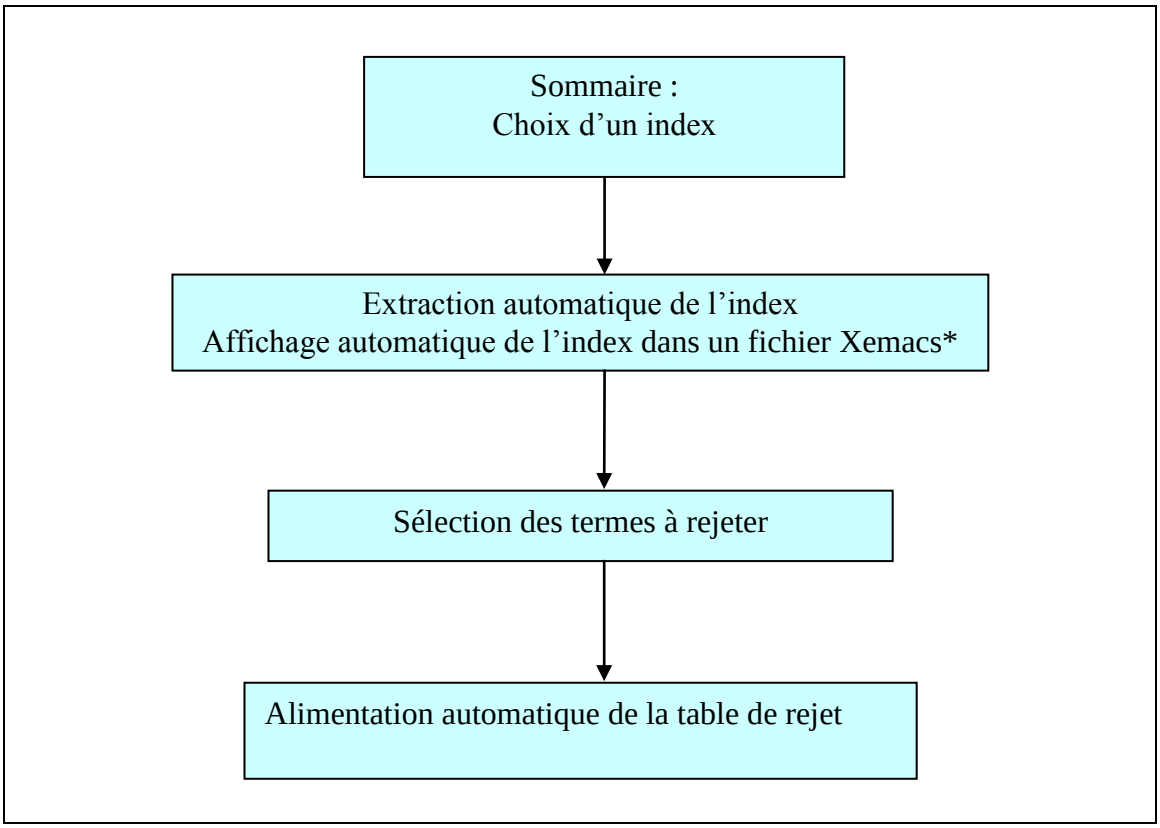

Figure 22 : Déroulement du Shell « ExtractIndex.sh ».

Ce *shell* en appelle en fait trois autres qui constituent chacun une partie du *shell* initial.

Première étape :

Elle correspond au *shell* "ExtractIndex1.sh " (Annexe 12). Celui-ci permet de créer une interface permettant à l'utilisateur de sélectionner l'index qu'il désire trier et de l'extraire de la structure HFD* vers un fichier temporaire.

- Extraction de l'index.

```
DamCat ../Server/$INDEX|SgmlSelect -s idx/kw# -p @s1|sed 's/^*//g' ✂
| sort  -u > index1
```

La commande " DamCat " permet d'accéder à tous les fichiers d'une structure HFD*. La commande " SgmlSelect " permet de sélectionner uniquement les termes de l'index (rappelons que les fichiers HFD* contiennent aussi les occurrences des termes et le numéro des notices auxquelles il est lié).

- Génération du fichier à modifier.

```
comm -23 index1 $TABLE | sed 's/^/ /g'  > index.tmp
comm -12 index1 $TABLE | sed 's/^/x /g'  >> index.tmp
comm -13 index1 $TABLE | sed 's/^/x /g'  >> index.tmp
cat index.tmp | sort -u > index
```

Les commandes " comm " permettent de comparer le fichier contenant les mots de l'index avec la table de rejet de cet index afin de signaler les termes déjà rejetés. Ceci permet notamment de relancer le *shell* de tri plusieurs fois sur le même index sans avoir à le trier à nouveau entièrement. Les mots déjà présents dans la table de rejet sont précédés d'une croix dans le fichier résultant.

Seconde étape :

Le *shell* " ExtractIndex2.sh " sert uniquement à afficher le fichier résultant de l'extraction sous Xemacs* sans intervention de l'utilisateur. Le *shell* initial a été séparé en trois parties uniquement pour rendre cette étape indépendante. Nous laissons ainsi la possibilité d'utiliser un autre éditeur de texte. Dans la liste affichée à l'utilisateur, tous les termes sont précédés d'une tabulation. Il lui suffit alors de cocher les termes qu'il désire éliminer. Par la suite, on envisagera une évolution consistant à utiliser un formulaire HTML pour réaliser cette étape.

Troisième étape :

Lorsque l'utilisateur ferme le fichier Xemacs*, le dernier shell " ExtractIndex3.sh " est lancé (Annexe 13) . Celui-ci permet d'ajouter les termes sélectionnés par l'utilisateur dans la table de rejet, puis d'effacer les fichiers temporaires pour ne pas encombrer l'application. La table de rejet est alors enrichie. Cependant le résultat n'est pas encore visible sur le serveur car les tables de rejet sont prises en compte lors de sa génération. Il est donc nécessaire de relancer le serveur si l'on veut voir apparaître les modifications (Figure 23).

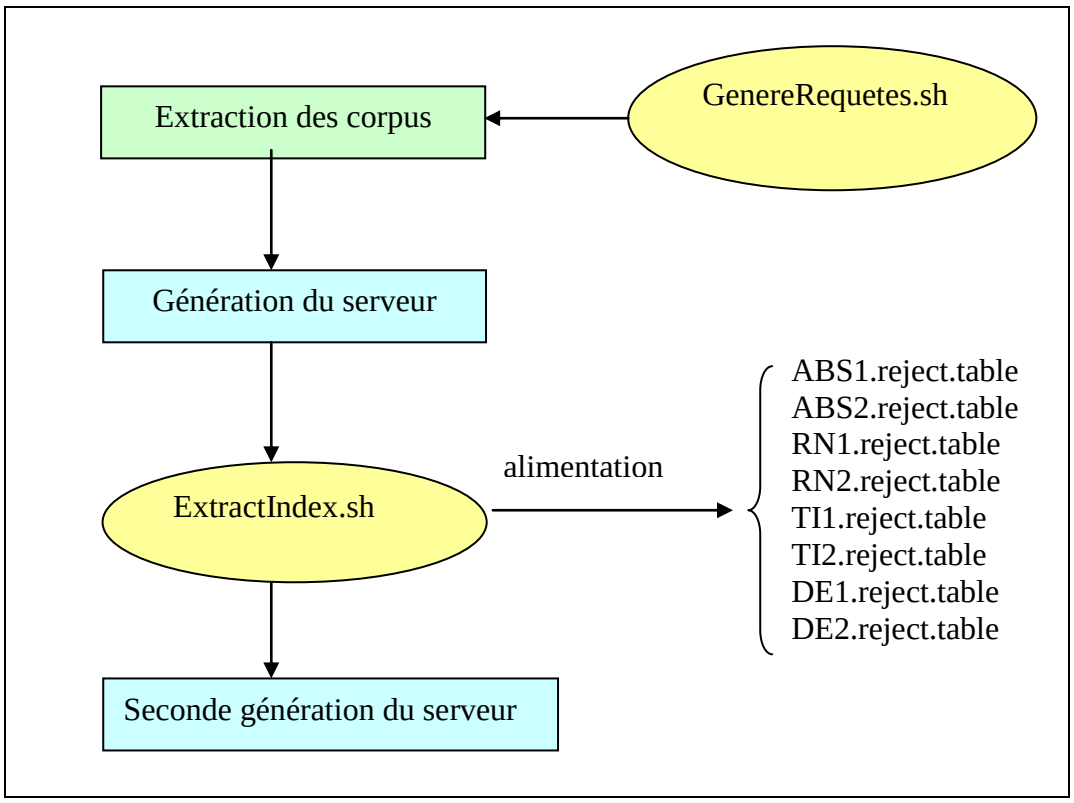

Figure 23: Récapitulation de la procédure utilisateur.

6.5 Modifications apportées au niveau du serveur.

6.5.1 Champs indexés.

Dans l'application d'origine, cinq champs des notices *Medline* étaient indexés lors de la génération du serveur (Auteurs, mots du Titre, mots du Résumé, termes du MeSH*, *Registry Number*). Afin d'étoffer l'application, nous en avons ajouté cinq autres (titre, Numéros *Genbank*, Numéros *Medline*, Affiliation et Journal). D'autre part, plusieurs champs ont été modifiés afin de faire apparaître sur le serveur d'investigation des fonctionnalités intéressantes pour l'analyse des informations.

Index des titres (tit) :

Cet index a été créé afin d'avoir une liste de l'ensemble des documents utilisés pour générer le serveur. Il contient en fait les informations de deux champs des notices *Medline*. En effet, chaque titre est précédé de la date d'introduction du document dans *Medline* (champ EDAT des notices). La date étant placée devant le titre et l'affichage sur le serveur se faisant par ordre numérique inverse, cela permet de faire apparaître les documents les plus récents en premier. Lors d'une mise à jour de l'application on peut alors rapidement identifier les nouveaux documents.

- Sélection des deux champs.

```
cat Import/corpus_medline1.sg | SgmlSelect -s medline/EDAT# -s ✂
medline/TI# -p @s1 -p @s2 > Import/tab1
```

La commande "SgmlSelect" permet, à partir des notices au format SGML, de créer un fichier « tab1 » en deux colonnes, l'une contenant la date, l'autre contenant le titre.

- Création de la table de remplacement.

```
cat Import/tab1 | awk -F\t '{printf "%s\t(%s) %s\n", $2,$1,$2}' ✂
> Import/tab11
```

La commande "awk" génère la table de remplacement permettant de rajouter la date devant les titre dans le fichier " tab11 ". Ce fichier est créé à partir du fichier " tab1 ".

- Prise en compte de la table de remplacement.

```
cat Import/corpus_medline1.sg | SgmlTextProc -P tableReplace -F ✂
all medline/tit# -R -t Import/tab11 > corpus_medline1_dedoublonne
```

La commande "SgmlTextProc" permet de prendre en compte la table de remplacement.

Index des numéros *Genbank*.

Le contenu du champ *Secondary Source Identifier* (SI) a été modifié pour les gènes du tableau des résultats de puces à ADN*. Pour ces gènes, nous avons ajouté l'expression différentielle ainsi que la protéine pour laquelle il code (information fournie par le fabricant des puces à ADN utilisées). Pour réaliser cette opération, nous avons créé une table de remplacement (Figure 24) selon le même principe que pour les titres.

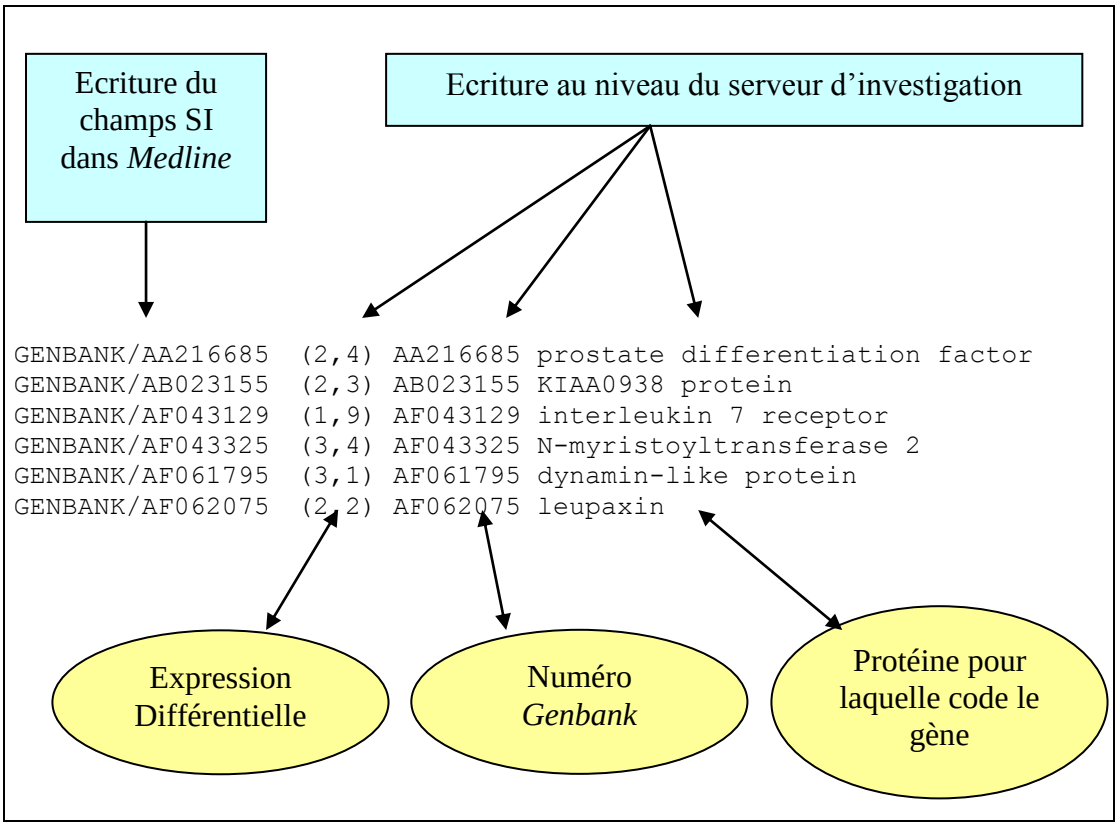

Figure 24 : Table de remplacement pour les numéros Genbank

Index des mots du MeSH*.

Les termes du MeSH sont contenus dans le champs MH des notices *Medline*. Ce champs MH est utilisé pour constituer l'index "descripteur" (DE) du serveur. Le champ MH des notices *Medline* contenant les termes du MeSH* est particulier (Figure 25). Chaque occurrence de ce champ est en fait un suite de termes incluant des termes génériques (*Subheadings*). Si cette présentation particulière est utile pour faire des recherches sur Medline, elle n'est pas très appropriée à l'établissement d'un serveur d'investigation réalisant des associations et des *clusters* sur les termes spécifiques.

```
<DE>
    <e>Aorta&sol;drug effects/metabolism/pathology</e>
    <e>Blotting, Northern</e>
    <e>Cell Division/drug effects</e>
    <e>Cell Movement/*drug effects</e>
    <e>Cells, Cultured</e>
    <e>Comparative Study</e>
    <e>Cytokines/metabolism/*pharmacology</e>
    <e>*Gene Expression/drug effects</e>
</DE>
```

Figure 25 : Structure de l'index des mots du MESH.

Afin de pouvoir exploiter tous les termes contenu dans cet index, nous les avons ramenés à des "unitermes " (figure 26).

```
<DE>
    <e>Aorta&sol;drug effects</e>
    <e>metabolism</e>
    <e>pathology</e>
    <e>Blotting, Northern</e>
    <e>Cell Division</e>
    <e>drug effects</e>
    <e>Cell Movement</e>
    <e>drug effects</e>
    <e>Cells, Cultured</e>
    <e>Comparative Study</e>
    <e>Cytokines</e>
    <e>metabolism</e>
    <e>pharmacology</e>
    <e>*Gene Expression</e>
    <e>drug effects</e>
</DE>
```

Figure 26 : Structure de l'index des termes du MESH après scission.

Cette séparation des termes du MESH est réalisé par un programme que nous ajoutons dans la commande "MedlineFromWww " chargé de passer les notices du formats Web au format SGML utilisé par DILIB (figure 27). La commande modifiée s'appelle " MedlineFromWwwSplit ", elle intègre un le programme de scission " SplitDEFromEd.pl ".

```
#!/bin/sh

. $DILIB_CONFIG

SgmlCharSetTr -f iso8859        \
 | MacNewLine                   \
 | MedlineWww1                  \
 | MedlineWww2                  \
 | sed  's/&sol;/\//g'          \
 | Import/SplitDEFromEd.pl             \
 | MiniBibFromEd
```

Figure 27 : Modification locale de la commande " MedlineFromWww ".

La scission s'effectuant sur le caractère « / », il est nécessaire de rétablir ce caractère codé par " &sol ; " en SGML avant d'appliquer cette commande. C'est le rôle de la commande " sed " qui la précède. La dernière modification a consisté à supprimer les " * " précédant certains termes du MESH* afin de ne pas perturber les classements alphabétiques. Pour cela nous avons utilisé une table de remplacement comme pour les deux index précédents.

6.5.2 Tri des index.

Dans le cadre d'une problématique particulière, tous les termes des index ne sont pas pertinents, ce qui rend les index inutilement volumineux. Pour rendre la navigation sur le serveur plus ergonomique et la recherche d'informations plus efficace et plus rapide il est nécessaire d'effectuer un tri des index (Figure 28) . Ce tri à été réalisé par Alain Zasadzinski, ingénieur documentaire spécialisé en biologie moléculaire (INIST), et Sébastien Vachenc, étudiant de maîtrise en stage dans le laboratoire de Bertrand Rihn.

Leur intervention a permis de tester à la fois la procédure utilisateur (Annexe 10) et le programme de mise à jour des tables de rejet.

	INDEX	Nombre de termes avant le tri	Nombres de termes après le tri
Corpus gènes sur-exprimés	Mots des résumés	672	359
	Termes du MESH	828	540
	Registry Number	508	465
	Mots des titres	885	494
Corpus gènes sous-exprimés	Mots des résumés	618	408
	Termes du MESH	774	580
	Registry Number	455	431
	Mots des titres	818	566

Figure 28 : Nombres de termes par index avant et après le tri.

Remarque :

Comme nous l'avons vu dans la procédure utilisateur, les termes rejetés servent à alimenter des tables de rejet spécifiques pour chaque champ *Medline*. Ces tables peuvent par la suite être prise en compte directement lors d'une mise à jour ou lors de la génération d' une nouvelle application dans le même domaine de recherche, ce qui permet de limiter les interventions manuelles.

D'autre part, il est possible selon le même principe d'alimenter des tables de synonymes afin d'éviter les redondances sur le serveur. En effet, il est fréquent qu'un terme scientifique décrivant une entité ou un phénomène biologique possède un ou plusieurs synonymes.

6.5.3 Les croisements entre les différents index.

Afin d'améliorer les fonctionnalités du serveur d'investigation généré, nous avons introduit des croisements systématiques entre les différents index. Cela permet par exemple, lorsque l'on sélectionne un mots clé, d'avoir accès à des listes des termes associés provenant des autres index (Figure 29).

Figure 29 : Exemple des termes associés disponibles pour un terme du MeSH.

Pour intégrer de tels croisements au serveur il faut compléter deux fichiers :

- Le fichier de description (Annexe 14).
  Dans ce fichier, il faut paramétrer les "cross", c'est-à-dire les croisements d'index que l'on désire effectuer, et cela dans la déclaration de chaque index. Puisque nous désirons réaliser ces croisements sur les deux corpus, nous les déclarons dans la description générique (Figure 30). Chaque base faisant référence à ce type générique héritera de la déclaration des croisement d'index à effectuer. Ces croisements permettent d'associer un terme d'un index aux occurrences des autres index de l'application.

```
<generic type=base code=medline>
     <title>medline/TI#
     <author>medline/AU/e#
     <index code=DE lexicon=yes>
        <replace>Import/DE.syn.table
        <reject level=index>Import/DE.reject.table
        <path>medline/DE/e#
        <cross>DE
        <cross>aut
        <cross>TI
        <cross>tit
        <cross>RN
        <cross>ABS
        <cross>SI
        <cross>UI
        <cross>AD
        <cross>TA
     </index>
</generic>
```

Figure 30 : Fichier "Genome.desc.ed", déclaration des cross.

Cette déclaration dans le fichier de description permet de générer les listes de termes associés. Pour pouvoir y accéder sur le serveur, il faut paramétrer l'affichage des liens vers ces listes.

- Le dictionnaire.

Pour que l'affichage des liens vers ces croisements soit effectif, il faut les déclarer dans les fichiers " Genome.fr.dict " et " Genome.EN.dict " (Figure 31).

```
Gene1/DE/RN/crossAssoc/longName
<--#echo key=medline/DE/RN/crossAssoc/longName -->
Gene1/DE/UI/crossAssoc/longName
<--#echo key=medline/DE/UI/crossAssoc/longName -->
Gene1/DE/AD/crossAssoc/longName
<--#echo key=medline/DE/AD/crossAssoc/longName -->
Gene1/DE/TA/crossAssoc/longName
<--#echo key=medline/DE/TA/crossAssoc/longName -->
```

Figure 31 : Extrait du fichier " Genome.FR.dict ", déclaration des " cross ".

Sans cela, des données nécessaires à l'exécution des CGI* seraient manquantes et l'affichage ne serait pas réalisé.

Remarque :

Les croisements réalisés par DILIB ne prennent pas en compte les tables de rejet ni les tables de synonymes, ce qui peut occasionner du " bruit " (informations non pertinentes) lors de la consultation des listes de termes associés.

Ces tables devraient être prises en compte par les croisements d'index dans les prochaines améliorations de la version v0.3 de DILIB.

6.5.4 Comparaison du vocabulaire des deux corpus.

Bertrand Rihn souhaitait réaliser une analyse comparative des vocabulaires des deux bases. Plus précisément, il souhaitait pouvoir identifier sur le serveur les termes communs dans les deux bases et les termes propres à chaque base, en particulier pour les termes du *MeSH**, les numéros *Genbank** et les *Registry Number**.

- Extraction des index des deux bases (exemple pour les termes du MESH).

```
DamCat /applis/dps/INRS/Genome/Server/Gene1.DE.i.hfd | SgmlSelect ✂
-s idx/kw# -p @s1|sed 's/^*//g' | sort -u > Import/DE1

DamCat /applis/dps/INRS/Genome/Server/Gene2.DE.i.hfd | SgmlSelect ✂
-s idx/kw# -p @s1|sed 's/^*//g' | sort -u > Import/DE2
```

La commande d'extraction est la même que celle utilisée dans le *shell* de mise à jour des tables de rejet. Elle utilise la commande " Damcat " combinée avec la commande " SgmlSelect ".

- Comparaison des deux index (exemple pour les termes du MeSH).

```
comm -12 Import/DE1 Import/DE2 > Import/DEcommuns
```

- Génération d'une table de marquage des termes communs.

```
cat Import/DEcommuns | awk -F\t '{printf "%s\t%s(c)\n", $1,$1}' ✂
> Import/DEcommuns.table
```

Pour marquer les termes communs, nous avons utilisé des tables de remplacement. Le principe est simple, on remplace tous les termes communs par le même terme additionné d'un (c).

Le *shell* " GenereTableMotsCommuns.sh " permet de créer les tables de marquage des termes pour les trois index concernés (Annexe 15).

- Prise en compte de la table

Comme toutes les autres modifications du contenu des index exposées au paragraphe 5.5.1, la table de remplacement est prise en compte dans l'exécution du *shell* " TraitementPrelim.sh ", en utilisant la commande " SgmlTextProc ", option " tableReplace " (Annexe 16).

Remarque :

Ce marquage modifie le contenu des index. Afin de ne pas perturber l'établissement des tables de rejet, il est préférable de l'effectuer en dernier lieu une fois le tri sur les index terminé (Figure 32). Pour alimenter les tables de marquage des mots communs, il faut se placer dans le répertoire " Import " et lancer le *shell* " GenereTableMotsCommuns.sh ". Les tables de marquage sont alors alimentées. Elles seront prises en compte dès que l'on relance le serveur.

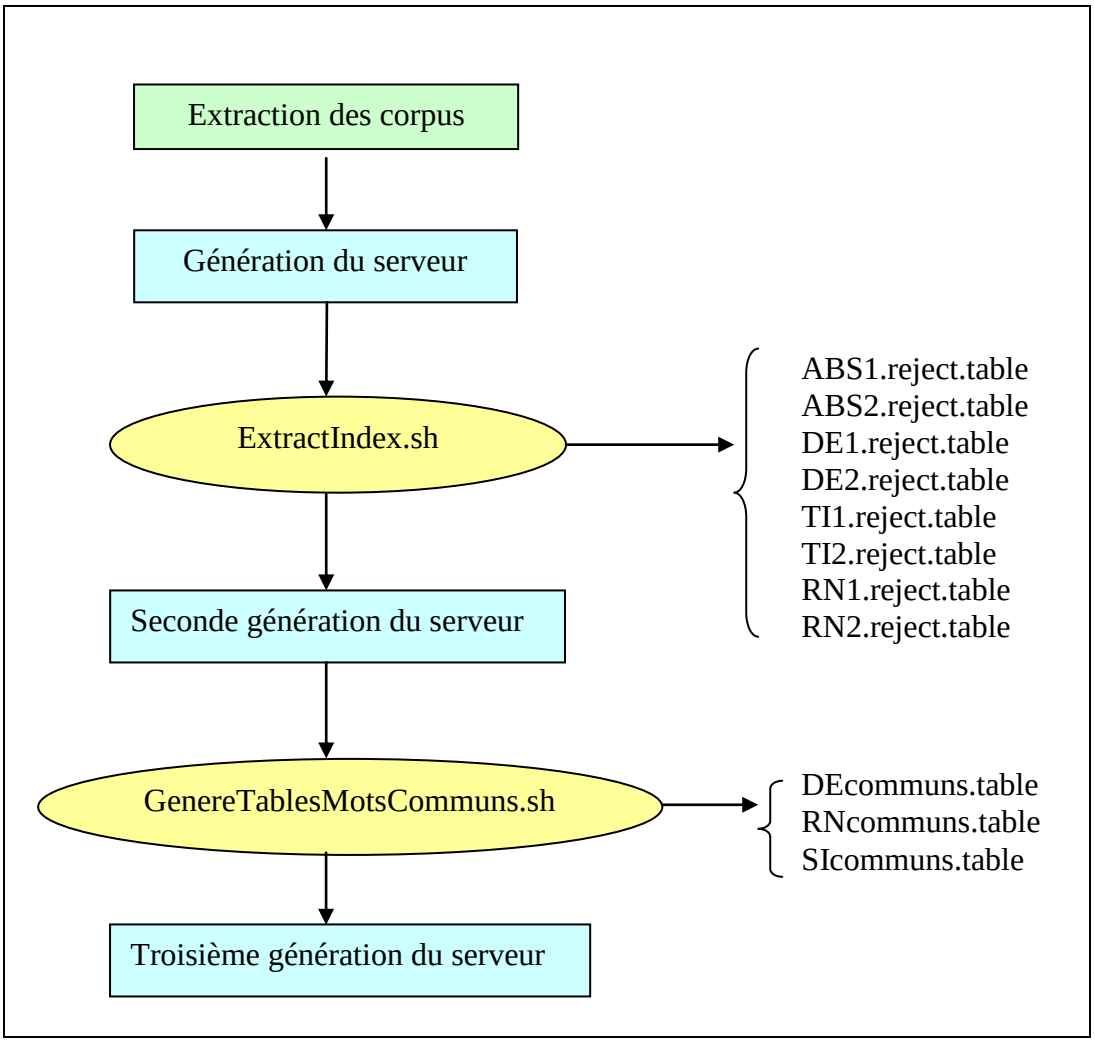

Figure 32 : Récapitulation de la procédure utilisateur avec ajout du marquage.

Remarque :

Nous utilisons ici une démarche itérative. La première génération permet d'extraire des index. Ces index sont utilisés pour générer des tables de rejet prises en compte lors de la seconde génération. La seconde génération permet d'obtenir des index pertinents que nous utilisons afin de générer des tables de synonymes permettant le marquage des termes communs lors de la troisième génération.

6.6 Perspective d'évolution : passage en version v0.3

Pour obtenir les fonctionnalités désirées sur le serveur, nous avons utilisé beaucoup de tables de remplacement. Celles ci sont appliquées directement sur le fichier contenant les notices au format SGML*. Or ce fichier sert à l'affichage du contenu des notices sur le serveur. Les notices *Medline* visualisables sur le serveur ne sont donc plus conformes. Cela ne pose pas de problèmes pour une utilisation interne mais devient gênant si l'on envisage de mettre l'application en ligne. Il faudrait donc pouvoir afficher des notices non modifiées, tout en réalisant des modifications au niveau des index du serveur. Une des solutions envisageable pour y parvenir serait de suivre simplement l'évolution de DILIB. En effet DILIB est en évolution constante et de nouvelle fonctionnalités intégrées à la version v0.3 devraient permettrent de résoudre le problème.

6.6.1 Traitement des termes du *MESH.*

La structure initiale du champ MH des notices d'origine permet d'optimiser les recherches dans *Medline*. C'est l'information la plus importante à restituer lors de l'affichage d'une notice par l'intermédiaire du serveur d'investigation. La version v0.3 de DILIB intègre de nouvelles options pour les fonctions " MedlineToPascal " et " SgmlAddNode ".

- La fonction " SgmlAddNode " permet de dupliquer facilement un champ des notices sous un nouveau nom. On pourra alors modifier ce champ et s'en servir pour créer les index du serveur, tout en conservant intact le champ initial.

- La fonction " MedlineToPascal " permet de ne garder que la première partie du contenu du champ MH (termes du MeSH*) de *Medline*, c'est-à-dire celle qui nous intéresse pour créer l'index DE du serveur.

En combinant ces deux fonctionnalités (Figure 33), nous obtenons la modification désirée dans le champ dupliqué (Figure 34), ce qui permettra de pouvoir disposer d'un index intéressant sur le serveur, sans pour cela toucher à l'intégrité des notices *Medline* affichées.

```
cat corpus_medline1.txt
            | MedlineFromWww
            | SgmlSelect -g medline/DE -p @g1 -p @0
            | MedlineToPascal -k DE/e
            | SgmlAddNode -n ED -A -p medline/DE
            | SgmlCut 1
```

Figure 33 : Nouvelle méthode de traitement pour les termes du MESH en version v0.3.

```
<medline>
...
<DE>
    <e>Chromosomes, Human, Pair 12</e>
    <e>DNA Primers&sol;chemistry</e>
    <e>High Mobility Group
Proteins&sol;*genetics&sol;metabolism</e>
    <e>Human</e>
    <e>Immunoenzyme Techniques</e>
</DE>
<ED>
    <e>Chromosomes, Human, Pair 12</e>
    <e>DNA Primers</e>
    <e>High Mobility Group Proteins</e>
    <e>Human</e>
    <e>Immunoenzyme Techniques</e>
</ED>
...
</medline>
```

Figure 34 : Résultat obtenu après le traitement de la Figure 33.

Nous pouvons remarquer que le champ DE a été dupliqué en un champ ED, lequel a subi les modifications générées par la commande " MedlineToPascal " option " -k ".

6.6.2 Cas des autres champs.

Toutes les autres modifications résultent de l'utilisation de tables de remplacement. Pour que ces modifications n'apparaissent pas à l'affichage des notices, il suffit de dupliquer tout les champs concernés et d'appliquer les tables de remplacement sur les doubles des champs originaux (Figure 35 ci dessous).

```
cat corpus_medline1.txt
            | MedlineFromWww
            | SgmlSelect -g medline/SI# -p @g1 -p @0
            | SgmlAddNode -n SI2 -A -p medline/SI
            | SgmlTextProc –P tableReplace –F all medline/SI2# -R –t table
            | SgmlCut 1
```

Figure 35 : Exemple de duplication du champ SI avant modifications.

Lors de l'affichage de la notice, les deux champs seront visualisés, le champ tel qu'il est dans les notices *Medline*, et son double intégrant les modifications apportées.

6.7 Exploitation du serveur.

Outre les statistiques sur le nombre de documents par terme pour les termes du MeSH et les *Registry Number* que nous ne développerons pas dans ce rapport (Annexes 17 et 18), le serveur a été principalement utilisé pour tenter de faire le lien entre certains gènes anormalement exprimés et différentes protéines dont la fonction est bien établie. Cette analyse s'est orientée suivant deux axes principaux :

- un axe général concernant diverses protéines liées à des gènes exprimés différentiellement* dans les cellules cancéreuses,
- un axe plus ciblé concernant des protéines impliquées dans le processus de mort programmée des cellules (apoptose).

6.7.1 Recherche sur certaines protéines liées à des gènes exprimés différentiellement dans les cellules cancéreuses.

Cette recherche concerne des protéines ou des enzymes ayant un rôle bien défini codées par des gènes exprimés différentiellement (Annexe 19)

- *cyclin dependent kinase* (cdk), enzyme régulant le cycle cellulaire,
- *integrin*
- *laminin*
- *fibronectin*

  } , adhésion et reconnaissance moléculaire,

- *interleukin*, système immunitaire.

Le rapport de Diplôme d'Etude Approfondies de Steve MOHR " Etudes des transcriptomes de cellules humaines mésothéliales et de mésothéliome " [9] notifie les protéines codées par des gènes exprimés différentiellement dans les cellules de mésothéliome. L'analyse des protéines et des gènes associés fournis par le serveur permet d'approfondir les recherches pour tenter d'élucider la biologie du mésothéliome.

6.7.2 Recherche sur le registre " APOPTOSE ".

Ce registre concerne des familles de protéines ayant un rôle dans le processus d'apoptose* (Annexe 20) :

- *Caspase*, protéine intervenant dans la régulation de l'apoptose,
- *Myc*, gène exprimé dans les cellules en apoptose.

Pour ce registre, la démarche est différent. Elle consiste à interroger le serveur pour retrouver des informations sur des protéines impliquées dans le processus d'apoptose. Le serveur permettra ensuite de relier ces protéines à des gènes exprimés différentiellement dans les cellules cancéreuses afin de mieux comprendre la transformation cancéreuse des cellules.

6.7.3 Intérêt du serveur d'investigation DILIB, et ses limites.

La recherche bibliographique automatisée s'impose de par la nature même des expérimentations sur le génome qui mettent en jeu des milliers de gènes et par l'abondance des données bibliographiques dans ce domaine. Ce type de serveur permet d'analyser très rapidement l'ensemble des connaissances publiées sur un sujet d'étude en mettant en évidences les relations entre les concepts biologiques. La navigation hypertexte est une composante essentielle dans la visualisation de ces relations.

Ces informations, souvent inaccessibles par une recherche documentaire classique car noyées dans la masse d'informations bibliographiques, peuvent permettre aux chercheurs de proposer de nouveaux axes de recherche. D'ailleurs, le foisonnement d'applications dans le domaine de la génomique ([10], [11], [12], [13]) témoigne de l'intérêt qu'elles peuvent susciter.

6.8 Devenir de l'application.

6.8.1 Applications multi-base.

Il existe de nombreuses bases de données bibliographiques et factuelles* mettant à disposition des informations diverses sur les gènes et les protéines. Une réflexion est en cours à l'INIST en vue de réaliser un serveur d'investigation multi-base permettant d'intégrer à la fois des bases de données sur les gènes et les protéines afin de pouvoir relier des informations souvent parsemées et difficiles à mettre en corrélation. Un tel serveur d'investigation multi-base est un outil d'analyse très intéressant et peu commun car la plupart des outils d'analyse infométrique ne permettent pas d'exploiter l'annotation* des bases de données factuelles en génomique.

6.8.2 Mise en ligne.

L'INIST et l'INRS projettent de mettre l'application en démonstration sur Internet dans les mois à venir. Cela permettra de présenter le serveur d'investigation généré par DILIB et ses fonctionnalités. Cette mise en ligne sera accompagnée par une publication de Bertrand Rihn sur l'utilisation de ce serveur et les informations qui en ressortent dans une revue spécialisée de génomique. L'application devrait de plus être présentées aux directeurs du Généton et d'Infobiogen avant la fin de l'année.

7 Conclusion

Tout d'abord, ce stage m'a permis de mieux comprendre le rôle de l'INRS et de découvrir l'étendue de ses activités de recherche.

Le travail de restructuration sur l'application IMD m'a permis d'approfondir considérablement mes connaissances de l'environnement UNIX et de langages de script tels que le " *shell* " ou le " *C-shell* ". En effet, comme nous l'avons vu dans ce rapport, la restructuration établie a nécessité l'écriture de plusieurs scripts *shell.* D'autre part ce travail m'a permis de bien comprendre l'architecture d'une application de DILIB, savoir faire essentiel à la mise en place d'une procédure de mise à jour quasi-automatisée de l'application IMD.

Le développement de l'application " Transcriptome " m'a permis d'acquérir des connaissances de base en génomique. C'est un domaine des sciences très intéressant que je ne connaissait pas et que le stage m'a permis de découvrir. Le serveur Tanscriptome est une application concrète du concept en pleine expansion de " génomique et traitement informatique ", et participer à ce type d'entreprise est très captivant.

Outre les fonctionnalités d'analyse fournies par le serveur, c'est la démarche même de construction de l'application qui est intéressante. Le serveur est construit à partir d'informations sélectionnées en relation directe avec les résultats d'une expérience. C'est une démarche assez novatrice en ce qui concerne l'utilisation de DILIB qui pourrait laisser entrevoir bon nombre d'applications dans tous les domaines de la recherche.

BIBLIOGRAPHIE

1 INRS
http://www.inrs.fr

2 Documentation INRS (2001). Faits et chiffres 2000.

3 INIST
http://www.inist

4 Fascicule de présentation de l' INIST (2001). L'accès à l'information scientifique et technique.

5 NEDELCOT A (2000). Mise à jour et installation de DILIB v0.21, son utilisation dans les applications " IMD " et " Genome ". Rapport de stage de DESS Information Scientifique et Technique, Nancy.

6 DUCLOY J (2001). Plateforme et bote à outils DILIB .
http://www.loria.fr/projet/dilib

7 RIHN B., MOHR S., McDOWEL S.A., BINET S., LOUBINOUX J., GALATEAU G., LEIKAUF K. and G.D. (2000). Differential gene expression in mesothelioma. FEBS Letters, 480 pp.

8 INFOBIOGEN.
http://infobiogen.fr

9 MOHR S (2000). Etude des transcriptomes de cellules humaines mésothéliales et de mésothéliome. Rapport de DEA Biologie Moléculaire et cellulaire, Université Louis Pasteur, Strasbourg.

10 MASYS D.R.(2001). Linking microarray data to the literature. Nature genetics, vol.28 9-10

11 ACHARD F., VAYSSEIX G., BARILLOT E. (2001). XML, bioinformatics and data integration. Oxford University Press.

12 RISLER Jean-loup (2001).Génomique et informatique, USTL-Conférence du mardi 24 Juillet 2001.

13 TSAKONAS ARTAMIS S. (2001). A l'aube de la biologie informatique, leçon inaugurale du samedi 28 avril 2001 au Collège de France.

GLOSSAIRE

ADN : Acide DesoxyriboNucléique. Enchainement de nucléotides formant deux brins complémentaires antiparalléles dépositaires de l'information génétique.

ADNc : Acide DesoxyriboNucléique Complémentaire. ADN simple brin obtenu par rétrotranscription à partir d'ARNm.

AIRS : AIRS est un système de recherche documentaire qui permet de classer, d' indexer et de retrouver des documents en fonction de leur contenu. AIRS Web permet un accès aux données sur Internet et Intranet.

Annotation des bases de données : Ajout de champs dans les références bibliographiques permettant de renvoyer à des informations complémentaires dans d'autre bases de donées.

ARNm : Acide Ribonucléique Messager. Copie complémentaire de l'ADN qui spécifie la séquence d'acide aminés d'une protéine.

Base de données AIRS : Base de données gérée par le logiciel AIRS qui est un système de gestion de bases de données documentaires.

CGI : *Common Gateway Interface.*

CIRIL : Centre Inter-universitaire de Ressources en Informatique de Lorraine.

Cluster : Ensemble de mots ayant des associations fortes, référant à un thème commun.

CNAM : Caisse Nationale d'Assurance Maladie.

CNAMTS : Caisse Nationale d'Assurance Maladie des Travailleurs Salariés.

CRAM : Caisse Régionale d'Assurance Maladie.

DEA : Diplôme d'études Approfondies.

DESS : Diplôme d'Etude Supérieures Spécialisées.

DTD : *Definiton Type Document*. C'est l'ensemble des règles, déterminées par une application, qui appliquent le langage SGML au balisage des documents d'un type particulier. Elle définit en cela le vocabulaire et la grammaire du balisage.

DILIB *: Documentation and Information LIBrary*. Plate-forme documentaire développée par l'INIST servant à générer des serveurs d'investigation.

Expression des gènes : Les gènes sont activés ou non selon un schéma génétique préétabli. C'est ce programme qui permet, par exemple, à un globule rouge de produire de l'hémoglobine. Un sous-ensemble activé du génome est responsable de l'aspect et de la fonction caractéristique des 200 types cellulaires différents qui composent le corps humain. C'est aussi la cause et/ou la conséquence de la transformation maligne des cellules. C'est cette même expression des gènes qui gouverne la naissance, le devenir et la mort des cellules.

Fichier inverse : Fichier d'indexation facilitant la gestion des relations entre les termes d'un index et les notices qui les contiennent.

Fichier d'association : Fichier permettant la mise en relation de termes co-occurrents d'un même document.

Format AIRS : Format des notices contenues dans les bases de données gérées par le logiciel AIRS.

Format "ed" : Format XML simplifié ne contenant pas toutes les balises de fin. C'est un format pivot pour DILIB. Le passage systématique par ce format permet d'utiliser un programme unique "minibibFromEd" pour convertir différent format au format XML. Cela est intéressant car le format "ed" est facile à obtenir à partir de tout les formats de notices bibliographiques.

FRANCIS : Base de données de l'INIST. C'est la principale base de données européenes sur les sciences humaines et sociales avec près de 2.5 millions de référebces bibliographiques depuis 1972.

Gène : région d'ADN qui contrôle un caractére héréditaire précis correspondant habituellement à un ARN unique.

Génome humain : Ensemble des gènes (environ 35 000) localisés sur les 23 paires de chromosomes de l'homme. Il comprend la totalité de l'information génétique appartenant à une cellule ou à un organisme.

HFD : *Hierarchical File organisation for Documentation.* Cette structure est utilisée par DILIB pour stocker les fichiers inverses et les fichiers d'association. Cette strcture permet de traiter en standard jusqu'à un million de références bibliographiques réparties en 100 répertoires de 100 fichiers contenant chacun 100 références.

HTML : *HyperText Markup Language*.Il s'agit d'un "langage à balises" (format ASCII), contenant des instructions entre les balises (*tags*) qui sont délimitées entre crochets. Ce langage permet de coder une page à l'aide de commandes de mise en forme. Ces dernières sont ensuite interprétées par un navigateur (*browser*) et apparaissent sur l'écran de l'ordinateur.

http : *HyperText Transfert Protocol*. Ce protocole permet à un serveur de communiquer avec un ou plusieurs clients sous la forme de requête et de réponse.

IMD : Application de DILIB exploitant les bases de données du centre de l'INRS DE Vandoeuvre-lès-Nancy.

INALF : Institut National de la Langue Française, cet institut à été renommé ATILF (Analyse et Traitement Informatique de la Langue Française) au début de cette année.

INIST : INstitut de l'Information Scientifique et Technique.

INRS : Institut National de Recherche et de Sécurité.

ISO : *International Standard Organization.* Cet organisme international établi des normes qui permettent de certifier la qualité de certain produits ou appareils.

IST : Information Scientifique et Technique.

LORIA : Laboratoire Lorrain de Recherches en Informatique et ses Applications.

Makefiles : Fichier utilisés lors de la génération d'une application de DILIB permettant de générer le serveur conformément au paramétrage établi dans le fichier de description.

MeSH : *Medical Subject Headings*. Thésaurus des mots clés de la base *Medline faisant* référence dans le domaine biomédical

Mésothéliome : Tumeur faite d'une prolifération des cellules de la plèvre. Cette tumeur, de la plèvre, toujours maligne, est habituellement en relation avec une exposition à l'amiante.

Méthode : Nom donné à une procédure applicable aux différentes instances d'une classe d'objet dans le modèle objet.

Modèle murins transgénique : Espèce de souris transgénique utilisées en laboratoire.

Navigateur (ou *browser*) : Logiciel de visualisation qui permet de consulter des pages au format HTML sur Internet ou pour un intranet.

ORF : *Open Reading Frame :* cadre de lecture ouvert, cela correspond à la séquence d'ADN d'un gène qui est exprimé, c'est à dire la partie transcrite en ARN messager.

PASCAL : Base de données multidisciplinaire et multilingue de l'INIST. Cette base de donée signale près de 14 millions de références bibliographiques en sciences, technologie et médecine depuis 1973.

PCR : *Polymerase Chain Reaction.* Réaction *in vitro* permettant d'amplifier des régions spécifiques d'ADN grâce à des cycles multiples de polymérisation de l'ADN, chacun suivi d'un traitement thermique bref pour séparer les brins complémentaires.

PMT : *Photo Multiplicator Technology.* Technique utilisée pour la lecture des résultats dans les expériences de puces à ADN.

Plèvre : Enveloppe du poumon lui permettant de glisser lors de la respiration dans la cage thoracique et cible de l'action toxique de l'amiante.

Protéome : Ensemble des protéines synthétisées à partir des ARN messagers.

Puces à ADN : Technique d'hybridation permettant une analyse génomique comparative de l'expression d'un grand nombre de *patterns* de mRNA. Immobilisés sur un support solide (matrice), des oligonucléotides (simples brins) spécifiques de différents gènes ou ADNc connus constituent les sondes dont le rôle est de détecter des cibles marquées complémentaires, présentes dans le mélange complexe à analyser (ARNm extraits de cellules, tissus ou organismes entiers et convertis en ADNc). Les sondes sont soit greffées sur le support, soit synthétisées *in situ* (unité d'hybridation = plot).
Les signaux d'hybridation sont détectés selon le type de marquage, radioactivité ou fluorescence, par mesure radiographique ou par fluorescence, et quantifiés.

Relation interne : Association entre deux mots-clés appartenant au même cluster.

Relation externe : Association entre deux mots-clés appartenant à des clusters différents.

Registry Number : Index des noms de substances et de leur numéro de nomenclature internationnale lorsque celui ci existe (RN pour les produits chimiques, EC pour les enzymes …).

Secondary Source Identifier (SI) : Intiutlé du champs Medline contenant les numéros d'accès vers d'autres bases de données (par exemple Genbank, SwissProt …).

Serveur : Ordinateur qui met ses ressources à la disposition d'autres ordinateurs sous la forme de services, qui peuvent être : Espace disque, Information, Base de données, Traitements automatisés.

SGML : *Standard Generalized Markup Language.*

Shell : Langage de script utilisé sous Unix. Ce langage permet d'ecrire des petits programme appelés script shell.

Texto : Logiciel de traitement de base de données.

Transcription : Synthèse d'ARNm à partir du brin codant d'ADN.

Transcrition inverse : Synthèse d'un brin d'ADN complémentaire à partir d'un ARN.

Transcriptome : C'est l'ensemble des transcrits (ARN messagers) issus de la transcription des gènes exprimés.

Unix : Système d'exploitation multi-utilisateur.

Variable de script et variables d'environnement : Une variable est définie par un nom et sert à mémoriser une information, représentée par une chaîne de caractère. Dans une variable de script le programmeur range une information qui sera utilisée dans les prochaine lignes de commande. Les variables sont aussi utilisées pour définir l'environnement d'execution des commandes .

Windows : Système d'exploitation très répandu sur les ordinateur personnels.

Web : Abréviation de World Wide Web, c'est un ensemble site informatique de type multimédia interconnectés pour constituer une hyper-base de données mondiale accessible en utilisant un navigateur (Netscape Communicator, Internet Explorer)

Xemacs : Editeur de texte utilisé sous unix.

XML : Ext*ensible Markup Language.* Language de balisage établi pour répondre au besoin d'élargir la nature des documents à échanger sur le web, de faciliter l'interopérabilité entre applications, et de permettre des description plus précises.

Liste des Figures

Résumé :	L'institut National de Recherche et de Sécurité utilise la plate-forme documentaire *Documentation and Information Library* (DILIB) pour mettre son fond documentaire à disposition des chercheurs par son Intranet. Cette plate-forme développée à l'Institut National de l'Information Scientifique et Technique (INIST) permet de générer à partir d'un corpus bibliographique un serveur d'investigation offrant des fonctionnalités d'analyses infométriques. La première partie du stage a consisté à automatiser en partie une application de DILIB exploitant les bases de données internes de l'INRS afin de simplifier la procédure de mise à jour du serveur. La seconde partie a été consacrée à l'optimisation d'une application de DILIB destinée à l'exploitation de corpus documentaires issus de Medline, en relation directe avec des résultats d'expérimentation sur les puces à ADN menées au laboratoire de cancérologie de l'INRS.

Mots-clés :	IST, DILIB, Génome, Transcriptome, Information Scientifique et Technique, format, base de données, mésothéliome, SGML, XML, infométrie, bibliométrie.