METIORE-WISP : UNE PLATEFORME POUR LA RECHERCHE COLLABORATIVE D'INFORMATION DU VEILLEUR

Philippe KISLIN(*), **Amos DAVID(*)**
Philippe.Kislin@loria.fr, Amos.David@loria.fr

(*) Laboratoire LORIA
Université de Nancy 2
Campus Scientifique, BP 239
54506 Vandoeuvre-lès-Nancy Cedex
FRANCE.

Mots clefs :
Intelligence économique, recherche d'information, veille, veilleur, décideur, système de recherche d'information, information hétérogène, modélisation du problème de recherche d'information du veilleur, travail collaboratif.
Keywords:
Economic intelligence, information retrieval, watch, watcher, decision-maker, IRS system, heterogeneous information, modelisation of the watcher's information search problem, collaborative work.

Résumé

L'activité de veille, au sein du processus d'Intelligence Economique, est principalement une activité de résolution de problème informationnel. Cette activité de veille est réalisée collaborativement par le veilleur qui doit localiser, surveiller et mettre 'en valeur' l'information stratégique et par le décideur qui doit formuler des demandes informationnelles aussi précises que possible afin que les résultats présentés par le veilleur soient utiles pour la prise de décision. Afin d'optimiser la traduction du problème décisionnel en problème informationnel, nous avons développé un modèle ainsi qu'un prototype qui l'instancie. Ces outils vont permettre de créer une interface de suivi et de communication pour la résolution collaborative des problèmes du décideur et du veilleur, de mémoriser l'intégralité du processus de veille et d'assurer une traçabilité des éléments informationnels pour en favoriser la réutilisation future.

1 Introduction

Décider dans un contexte d'Intelligence Economique consiste à choisir la solution qui paraît la plus adaptée à un problème décisionnel et à un moment donné parmi plusieurs alternatives disponibles. Pour cela, il est nécessaire que l'entreprise puisse disposer d'informations et d'outils, c'est-à-dire d'un ensemble de moyens pour juger, interpréter et évaluer la situation. Nous proposons que cette évaluation soit réalisée conjointement par deux acteurs, le décideur et le veilleur, qui s'engagent tous deux dans un projet collaboratif de résolution de problèmes, décisionnels pour l'un et informationnels pour l'autre. En effet, l'environnement socio-économique de l'entreprise, caractérisé par une accélération de plus en plus importante des cycles de renouvellement et d'adaptation des produits, exige une nécessaire efficacité des moyens engagés. De plus, cette forte mouvance impose trop souvent au décideur d'agir dans l'urgence et de décider dans l'incertitude. Dans ce contexte incertain et dans le cadre de nos travaux, nous pouvons envisager l'Intelligence Economique comme une méthodologie de compréhension et de résolution de ces problèmes, appréhendés le plus souvent dans l'urgence, abordés selon la confrontation d'un double point de vue, celui du décideur et celui du veilleur sur l'environnement et dont les activités de recherche de solutions en collaboration en constituent la finalité. Cet autre point de vue apporté par le veilleur, permet au décideur d'obtenir une approche complémentaire du problème à traiter, une vision plus globale et plus complète, donc de rendre plus efficiente la recherche de la solution. Afin de favoriser la traduction du problème décisionnel en problème informationnel, nous proposons, dans cet article, un outil informatique qui va servir à supporter à la fois le processus de traduction et la recherche d'information. Cet outil s'appuie sur le modèle WISP, un modèle du problème informationnel du veilleur que nous avons créer pour favoriser cette traduction et l'interopérabilité de divers composants informatiques qui vont permettre de mémoriser, d'annoter et de réutiliser les activités de recherche du veilleur. Cet outil va servir de plus d'interface de communication entre les deux acteurs, facilitant le suivi et la supervisation des recherches par le décideur.

2 Le modèle WISP : un outil pour favoriser la traduction du problème décisionnel en problème informationnel

Le modèle de recherche d'information du veilleur WISP (ou « *Watcher-Information-Search-Problem* ») [1] [2] que nous proposons est constitué d'une collection de vingt-sept éléments interreliés correspondants aux différents 'objets' manipulés par le veilleur durant tout le processus de veille. Ces objets, vont servir à la fois de conteneurs et de liens, et permettent au veilleur d'organiser et de coordonner les différentes étapes du processus de veille, depuis l'enregistrement de la demande jusqu'à la présentation des résultats. Chacun de ces objets comporte de nombreux attributs (intitulé, date, référence,…) qui les caractérisent et qui favorisent la traçabilité des informations, le suivi des activités et la réutilisation de ces informations et connaissances (figure 1).

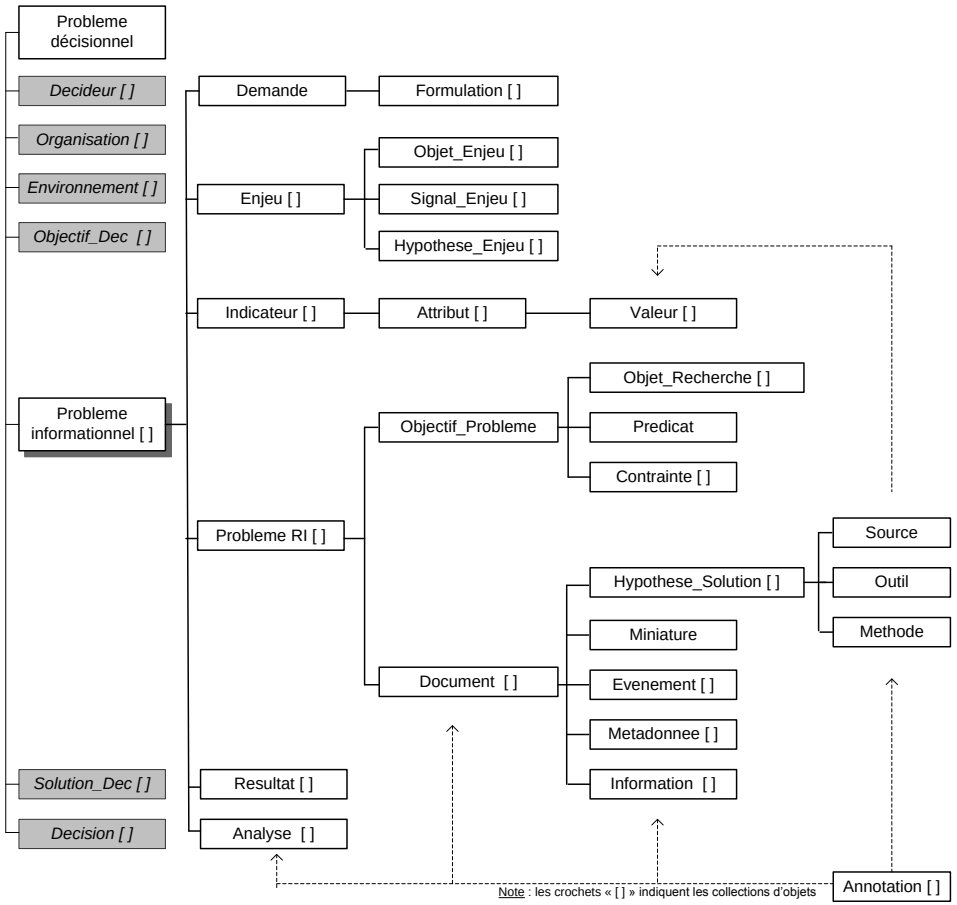

Figure 1 - Présentation générale du modèle WISP

Bien que ce modèle soit présenté sous la forme d'organigramme, le WISP n'est pas figé (« wisp » en anglais, c'est le 'brin' (d'herbe ou d'amour), ou la 'mèche' (de cheveux), mais aussi le fragment, l'indication ou la trace ; il est adaptable, évolutif et permet l'ajout à tout moment d'éléments complémentaires, comme les annotations par exemple. Les crochets '[]' qui suivent certaines étiquettes indiquent que l'élément est constitué d'une collection, c'est-à-dire d'un ensemble d'objets. Par exemple, à l'élément <Demande> *est associée une collection d'éléments* <Formulation> *qui vont correspondre à l'enregistrement des différentes formulations et reformulations de la demande produites par le décideur et par le veilleur.*

Ces éléments du modèle sont à la fois renseignés par le décideur (en grisé sur le schéma) et par le veilleur. Les éléments <*Décideur*>, <*Organisation*>, <*Environnement*>, <*Objectif Décisionnel*> vont contenir les descriptions de ces différents paramètres du contexte décisionnel de la demande auxquelles pourront

s'ajouter si nécessaire les références à des problèmes antérieurs ou à d'autres problèmes à traiter en parallèle. Ces éléments ont pour but d'aider le décideur à rendre plus explicite son besoin et à formuler une demande informationnelle en exacte adéquation avec celui-ci. Les différentes informations nécessaires au renseignement de ces éléments seront collectées à l'aide d'entretiens guidés, en réalisant des suivis de veille et des recherches ponctuelles dont les axes seront notifiés dans le modèle WISP. Enfin, en aval du processus informationnel, les éléments <*Solution Décisionnelle*> et <*Décision*> vont recevoir respectivement l'éventail des alternatives et des solutions envisagées, puis en dernier lieu, la solution retenue. Certains de ces paramètres inclus dans ces éléments seront à destination unique du décideur, d'autres pourront être communiqués et servir de support aux échanges avec le veilleur.

Les autres éléments du modèle (en blanc sur le schéma) vont servir à caractériser le problème informationnel. Seront ainsi mémorisés :

- La demande et ses différentes reformulations proposées par le veilleur et le décideur, dont seront extraits les mots-clés ;
- Les indicateurs informationnels, qui sont dérivés des thèmes ou des mots clés de la demande et dont les attributs seront déterminés par le veilleur et évalués par le décideur. Présentés sous la forme de couples (indicateurs, propriétés), ils vont constituer une 'interprétation' validée de la demande ;
- Les problèmes de recherche d'information destinés à valuer les attributs des indicateurs, pour lesquels les objectifs de recherche sont explicitement formulés ;
- Les outils, sources et méthodes employés pour la résolution de ces problèmes de recherche ;
- Les documents trouvés par le veilleur associés à leurs métadonnées, aux actions réalisées (clics sur liens, sélections d'information, utilisations de formulaires, …) et aux annotations produites.

Enfin les éléments <*Résultat*> et <*Analyse*> contiendront respectivement les produits informationnels et les analyses réalisées à la fois, sur ces derniers et sur le déroulement du processus. Tous les éléments du modèle sont datés et comportent chacun un identifiant unique. Ils vont permettre la traçabilité et l'évaluation des éléments informationnels tout au long du processus de veille. Les éléments rattaché à l'élément <*Problème de Recherche d'Information*> seront supportés par le prototype Metiore, qui grâce à ses facilités d'automatisation, permettra d'associer des données aux paramètres de ces éléments de façon transparente pour le veilleur.

3 Présentation du prototype Metiore

La plateforme Metiore (ou «*Multimedia coopErative InformaTIOn Retrieval SystEm*») était initialement un système de recherche coopérative d'information qui fut développé par David [3] puis adapté par Bueno [4] [5] dans le cadre de ses travaux sur la modélisation de l'utilisateur et la personnalisation de l'information. Ce prototype, permettait à l'origine de naviguer et d'interroger des bases de données comportant des informations structurées issues de sources homogènes comme des références bibliographiques par exemple. Nous avons donc souhaité faire évoluer ce prototype de manière à ce qu'il puisse traiter en plus des notices bibliographiques, des informations multisources et hétérogènes : pages web, documents PDF, XML et multimédias. Nous avons gardé son appellation d'origine en référence à leurs auteurs, mais nous l'avons cependant complètement réécrit et redéployé avec un environnement de développement plus adapté et plus performant pour pouvoir y implanter le modèle WISP et ses données. Cette plateforme utilise l'interopérabilité entre applications qui consiste à faire communiquer différents modules (sous la forme de composants 'COM','ActiveX' et assemblages .NET) et de les encapsuler dans des conteneurs au sein d'une seule interface informatique. Ainsi, nous avons doté Metiore d'un navigateur web s'appuyant sur le composant Internet Explorer dont nous interceptons les informations affichées et les événements de l'utilisateur afin notamment de les envoyer à un autre module chargé de les gérer et de les enregistrer dans une base de données. En réalité, nous avons instancié plusieurs composants 'navigateurs' (jusqu'à trois dans la fenêtre principale) de manière à ce que le veilleur puisse extraire des éléments de la page web en cours de visualisation par glisser-déposer, puis qu'il puisse modifier ces extraits dans un autre afficheur et navigateur HTML (figure 2). Nous avons ainsi

créé une fonctionnalité qui n'existe pas dans les navigateurs actuels : la possibilité de recomposer une page au format HTML, en récupérant des éléments textuels et graphiques issus de plusieurs autres, de modifier ces éléments et d'y ajouter des informations sous la forme d'annotations personnelles.

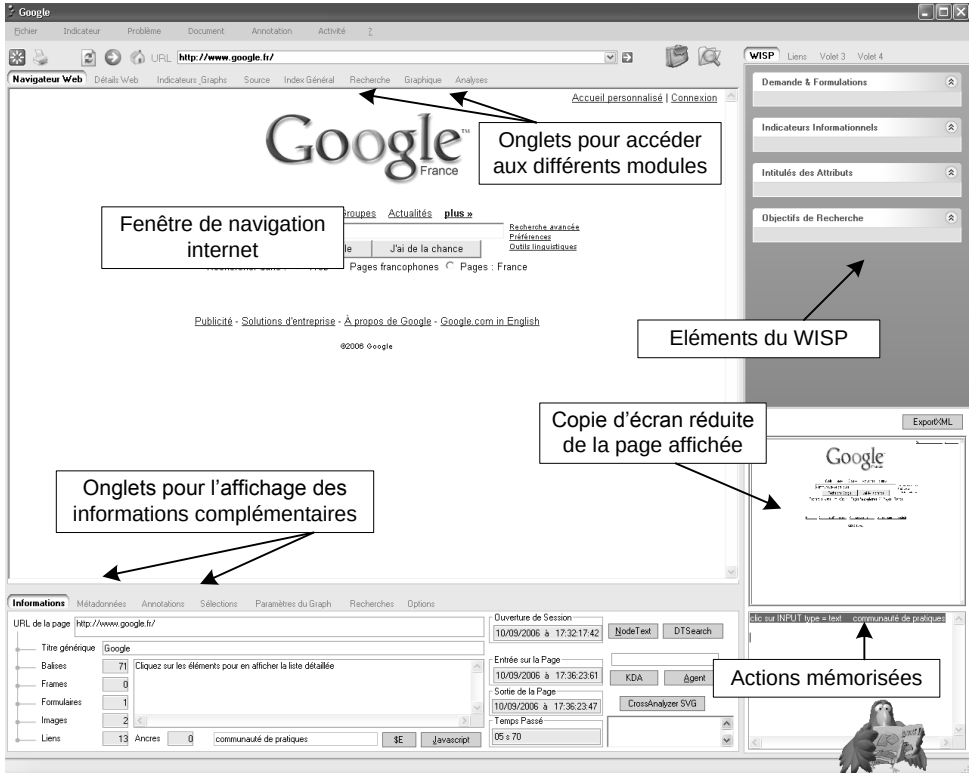

Figure 2 - L'interface principale de METIORE

Nous voyons sur cette copie d'écran, tout en bas, les informations relatives à la page affichée : l'URL de la page, son titre, les collections d'éléments (le nombre de liens, d'images,…). En cliquant sur ce nombre, l'utilisateur peut afficher par exemple la liste des liens, y accéder directement, etc. Nous pouvons lire également dans ce même cadre, les dates et heures de connexion et d'accès à la page, le temps passé et les actions réalisées sur celle-ci.

3.1 Architecture

Le système que nous proposons est composé de plusieurs modules (figure 3) qui sont regroupés autour d'un entrepôt de données. Ces modules vont communiquer avec l'utilisateur par l'intermédiaire d'une interface unique :

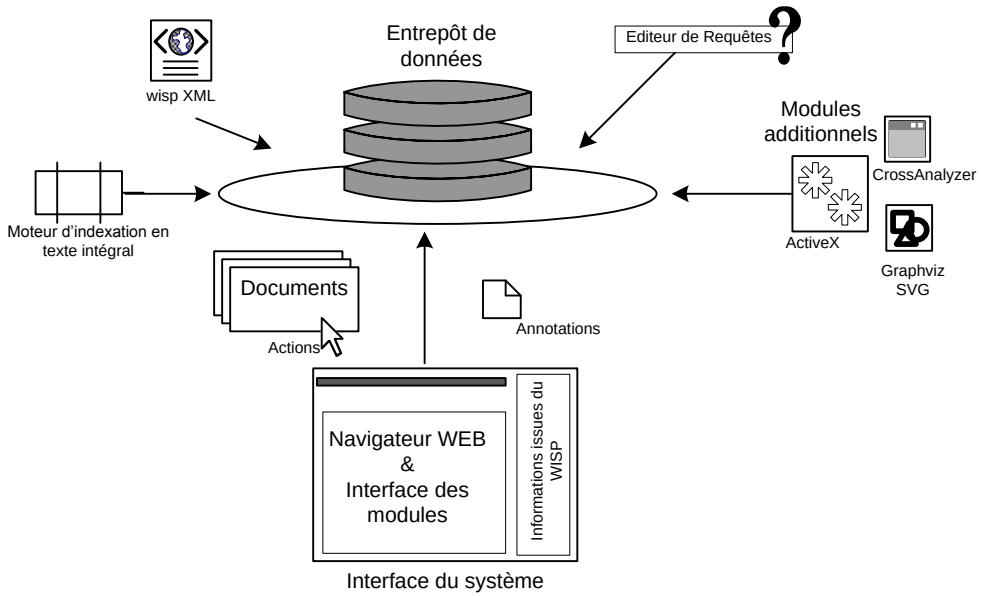

Figure 3 - Le schéma global de l'architecture de METIORE

- Un navigateur web pour l'affichage des pages et de toutes les informations issues de l'entrepôt ;
- Un module permettant d'enregistrer en tâche de fond des actions réalisées par l'utilisateur sur les documents affichés par le navigateur (comme les clics, les sélections, la saisie de formulaire, etc.) la copie iconographique (copie d'écran) de chacune des pages rencontrées lors des recherches ;
- Un module de traitement des pages web pour la visualisation et la complétion des métadonnées existantes ;
- Un module d'affichage des éléments du modèle WISP (les demandes et ses formulations, les objectifs de recherches, les indicateurs informationnels, etc.) ;
- Un module pour gérer le glisser-déposer d'information des pages vers l'espace des annotations et permettant ainsi de les intégrer aux différents éléments du WISP (et principalement de les relier aux attributs des indicateurs) ;
- Un moteur d'indexation en texte intégral pour l'indexation des documents produits par l'utilisateur ;
- Un éditeur de requêtes et de rapports permettant d'interroger et d'afficher les données de l'entrepôt selon diverses présentations ;
- Des modules additionnels comme un moteur de cooccurrences (*CrossAnalyzer*) et un module graphique (*Graphviz*) pour l'affichage de certains éléments du modèle sous la forme d'organigrammes (comme les indicateurs et leurs attributs par exemple) et qui peuvent opérer, comme les autres modules sur toutes les informations mémorisées.

- Un Système de Gestion de Base de Données Relationnelle (ou SGBDR) constituant l'entrepôt de données sur lequel il sera possible d'effectuer des requêtes SQL et de générer un ou plusieurs documents au format XML.

Cette interface est suffisamment souple pour gérer le « glisser-déposer-annoter » et ainsi facilite le travail de collecte et de traitement du veilleur.

3.2 Utilisation du modèle objet de document

Pour pouvoir utiliser des éléments d'une page HTML, encore faut-il pouvoir les identifier et accéder à ses différentes parties. Pour optimiser le traitement des pages web, les concepteurs les ont considérées comme des agrégats d'éléments composés eux-mêmes d'autres éléments appelés objets. Ainsi, un objet peut en contenir un autre, qui peut à son tour en contenir plusieurs, etc. Une page web peut être alors définie comme une collection structurée d'objets et présentée sous la forme d'un arbre. Pour accéder au contenu de ces objets, il suffit de connaître leur organisation hiérarchique. Au sein de cette arborescence d'objets appelée DOM (ou modèle objet de document), chaque objet 'parent' est connecté à son ou ses 'enfants' par l'intermédiaire d'une association qu'il suffit de repérer. Par exemple, si nous voulons accéder au contenu d'un champ de texte d'un formulaire, nous écrirons : (*Window->Document->Forms[]->Elements[]->Textarea->Text*) et si nous désirons obtenir toutes les images ou tous les liens d'une page, nous pourrions écrire : (*Window->Document->Images[] ou Window ->Document->Links[]*).

Cet usage est assez simple puisqu'il suffit de faire des références aux objets ou aux collections d'objets qui nous intéressent en indiquant leurs arborescences respectives. Ces objets possèdent également des propriétés et interceptent des événements (clic-souris sur un bouton, survol d'un lien,…) qu'il est possible à la fois de consulter et de modifier par programmation. Nous pouvons alors, par l'intermédiaire de l'interface DOM, manipuler nos documents issus du web et les associer aux éléments de notre modèle. Etant donné que nous avons conçu le WISP avec une structure 'objet' similaire (Figure 1) nous pouvons faire 'communiquer' les éléments de ces deux modèles entre eux :

Probleme_informationnel []->indicateur []->attribut []->Valeur = Window->Document->body->innerHTML

(Dans cet exemple, nous renseignons la valeur d'un attribut d'un indicateur avec le contenu textuel complet de la page web courante affiché par l'utilisateur.)

Il est ainsi possible d'afficher tout élément du WISP dans un navigateur et vice-versa, de mémoriser les objets de la page web et de les associer à tout élément de notre modèle.

3.3 Stockage des données

Afin de mémoriser à la fois les éléments du modèle WISP renseigné par le veilleur, et les références des documents affichés dans le navigateur, nous avons créé plusieurs tables dans une base de données Hyperfile. Chacune de ses tables va contenir un objet du modèle : la demande, ses formulations, les objectifs de recherche d'information, les indicateurs, leurs attributs, etc. Bien que la base Hyperfile puisse contenir plusieurs gigaoctets de données, nous ne pouvons pas y insérer directement les pages web dans leur intégralité. Techniquement, c'est assez délicat à réaliser, car lorsque le navigateur sauvegarde une page web, il sépare les images et les autres éléments multimédias comme les séquences vidéo, les animations au format 'Flash'(SWF) du contenu textuel en les enregistrant dans des dossiers différents. De plus, une procédure de sauvegarde automatique en texte intégral des pages nécessite plusieurs dizaines de secondes par page visitée, temps qui peut s'accroître très rapidement si la connexion internet est mauvaise et si l'ordinateur n'est pas équipé d'un processeur véloce. En outre, d'un point de vue cognitif, si le fonctionnement du logiciel n'est pas transparent pour le veilleur, il risque d'entraver sa réflexion et d'entraîner de trop nombreux arrêts dans sa recherche, empêchant

la fluidité de son raisonnement. Nous avons donc opté de ne sauvegarder que les métadonnées de la page (par extraction directe), le descriptif des actions réalisées par l'utilisateur, une copie d'écran du document, et les annotations dans la base de données Hyperfile. Le contenu des pages sera quant à lui enregistré sous un format compressé (MHT) dans un répertoire sur le disque dur de l'ordinateur du veilleur par l'intermédiaire d'une action de validation de sa part (Figure 4).

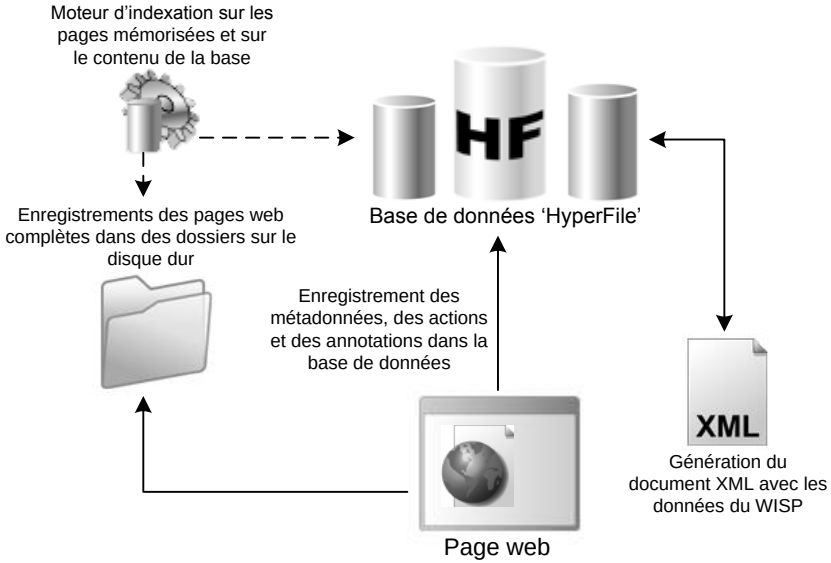

Figure 4 - Les différents lieux de stockage des données de METIORE

Comme toute séquence de recherche entraîne de nombreux aller et retour entre les pages (explorer un document depuis un lien dans un moteur de recherche, revenir à celui-ci, visiter d'autres pages, changer de requêtes, retourner sur des pages précédentes,…), nous aurions vite saturé la mémoire sur le disque dur si nous n'avions pas opté pour une solution 'débrayable' où le veilleur va lui-même indiquer à Metiore les pages à sauvegarder. De plus, comme ces pages seront indexées par le moteur d'indexation, nous aurions, en plusieurs exemplaires, certaines d'entre elles (sans compter toutes les fenêtres publicitaires qui s'y seraient rajoutées et que le navigateur ne peut bloquer) rendant son usage peu efficient.

Enfin, la particularité du modèle WISP est d'être adaptable. Or, une base de données nécessite de définir au préalable, c'est-à-dire lors de sa création, une structure rigide des données. Définir le modèle d'une base de données est une activité de conception qui exige de bien cerner tous les besoins et leurs contraintes ainsi que spécifier les exigences pour chacune des données à mémoriser. Si de nouveaux champs doivent être ajoutés ultérieurement, il faut alors recréer une nouvelle structure de données et redéployer la base, activités qui ne peuvent être réalisées directement par l'utilisateur. Pour pouvoir ajouter autant d'éléments que nécessaires à notre modèle, nous avons opté pour une sauvegarde du WISP dans un fichier au format XML. Ce document présente également une structure hiérarchique arborescente des éléments d'information.

3.4 Accès aux données

Etant donné que nous avons trois lieux différents de stockage de l'information dans Metiore (une base de données, les informations propres au modèle WISP dans un fichier XML, les pages visitées ainsi que tous les documents produits par l'utilisateur dans des répertoires sur le disque dur), nous avons développé deux protocoles d'accès aux données : l'un pour la base, l'autre pour les fichiers sur le disque.

Accéder aux données de la base est relativement simple à réaliser : à l'aide de requêtes SQL, nous pouvons extraire des informations des différents champs des tables de données et les faire afficher par notre navigateur. La base de données Hyperfile, comme la grande majorité des bases existantes accepte ce langage d'interrogation facile d'utilisation. Nous pouvons ainsi créer des requêtes adaptées pour générer des états afin d'afficher les informations à travers des gabarits différents (rapports, listing d'enregistrements, tableaux de contingence, etc.) et les transmettre au décideur selon le format préalablement retenu et le modifier à tout moment si nécessaire. Un autre intérêt de ces requêtes est qu'elles vont permettre au veilleur d'effectuer des analyses sur le processus de recherche d'information et d'ainsi créer des indicateurs statistiques à la fois sur les éléments du modèle (par un accès au document XML) et sur les données de la base. Par exemple, comme le temps passé sur chaque page survolée est mémorisé, il est possible de cumuler ces temps et d'ainsi connaître le temps consacré à la résolution d'un problème de recherche particulier. Un autre indicateur statistique pourrait consister à calculer le pourcentage d'utilisation d'une source d'information particulière par rapport au nombre de documents retenus.

Pour pouvoir extraire des informations des fichiers stockés sur le disque, nous avons utilisé un moteur d'indexation qui a été implanté dans Metiore de la même manière que le navigateur web, c'est-à-dire sous la forme d'un composant 'ActiveX'. Ce moteur d'indexation, comme tout moteur, ne peut avoir accès qu'aux documents numériques dont il connaît l'emplacement et ne peut traiter que les éléments textuels de ceux-ci. Bien évidemment, le veilleur, pour résoudre un problème de recherche d'information, n'utilise pas que des informations issues des pages web visitées. Il a recours à beaucoup d'autres sources d'information comme les documents produits en interne par les différents services de l'entreprise. Si ces documents existent sous forme numérique, le navigateur étant capable d'afficher tout type (ou presque) de document électronique par l'ajout de 'plug-ins ' (les documents PDF, les documents produits par la suite Office,…), il suffira que le veilleur y accède par l'intermédiaire du navigateur pour permettre au système de les mémoriser. Si une partie de ces documents n'existent que sous forme papier, l'utilisateur devra alors les scanner au préalable s'il souhaite en obtenir une copie numérique ou réaliser pour chacun d'eux, une description de ces documents (une notice catalographique en quelque sorte) selon le format de métadonnées que nous avons retenu (DCMI vers 1.1). Le champ 'adresse (URI)' du document ne sera pas, dans ce cas, une adresse web ou un chemin d'accès sur le disque, mais contiendra sa localisation physique. De même, si des informations proviennent de sources informelles et orales, le veilleur devra transcrire ces informations dans un ou plusieurs documents, comme s'il s'agissait d'établir le compte rendu écrit d'une réunion. Il sera ensuite référencé dans la base Hyperfile, associé à un élément du WISP et pourra ainsi être traité comme tout autre document numérique.

Ces différentes conversions auront donc pour but que le moteur d'indexation puisse indexer les documents en intégralité ou le cas échéant leur fiche descriptive respective associée à leur localisation. En implémentant ce moteur (figure 5), nous avons prévu que l'utilisateur puisse paramétrer lui-même l'accès aux différents répertoires contenant ces fichiers et lancer des réindexations à tout moment de son choix.

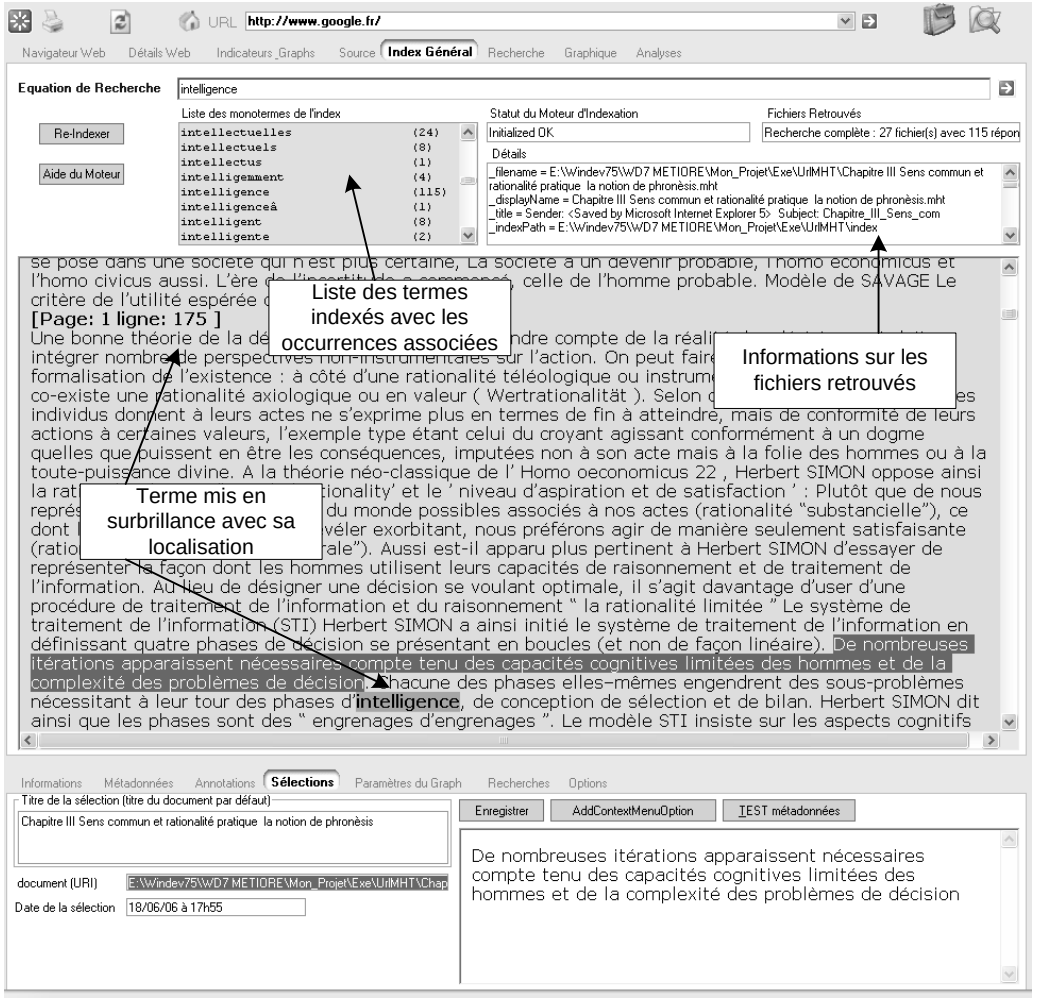

Figure 5 - L'implémentation du moteur de recherche dans METIORE

Note : apparaît ici, une sélection de l'utilisateur opérée sur le résultat de la recherche affiché. (le document présenté est extrait de [Gueorguieva V., La connaissance de l'indéterminé. Le sens commun dans la théorie de l'action, Thèse en sociologie, Université Laval, Août 2004].

3.5 Sélection des informations

Pour extraire des informations des pages, il suffit au veilleur de sélectionner à l'aide de souris l'information l'intéressant et de la « glisser-déposer » dans l'onglet de sélection. Il peut s'il le souhaite compléter cette information, car le champ est directement éditable puis la mémoriser avant de l'enregistrer (figure 6).

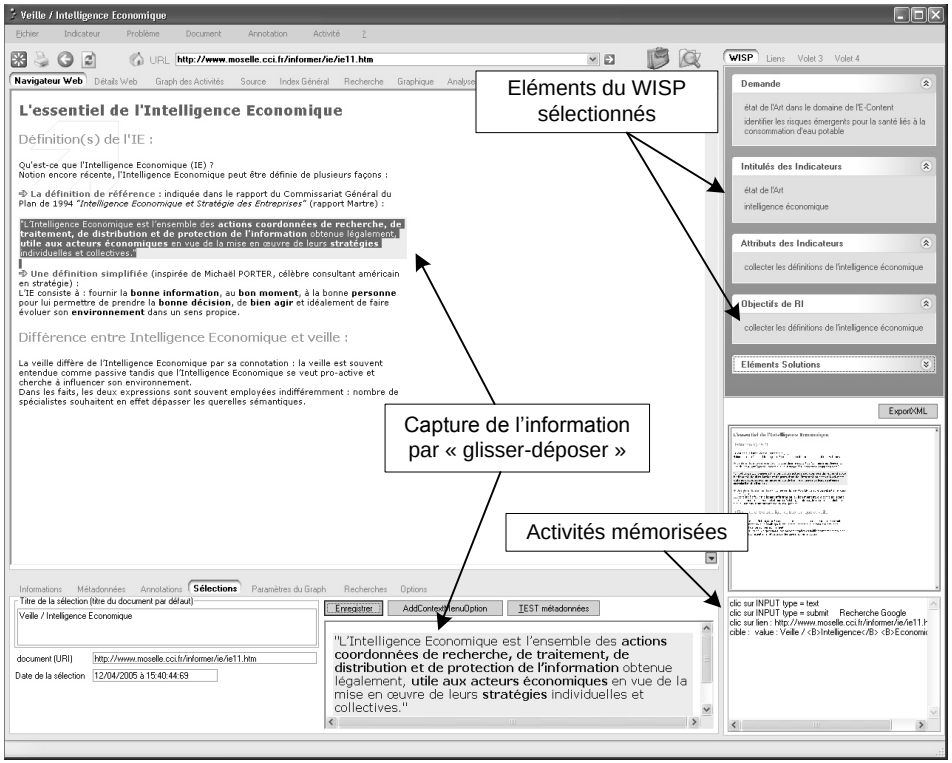

Figure 6 – La sélection d'information dans METIORE

3.6 Renseignement des métadonnées

Généralement, dans les navigateurs web, les métadonnées des pages web ne sont pas directement accessibles à l'utilisateur. Elles sont avant tout destinées à la description de leur contenu aux moteurs de recherche pour augmenter la précision de leur indexation. Elles figurent dans le code source de la page, et sont la plupart du temps, renseignées par le concepteur du site. Dans Metiore, le veilleur peut à tout moment accéder aux métadonnées de la page sur laquelle il se trouve juste en activant l'onglet adéquat. Celles-ci sont extraites du code source par le système de manière à ce que le veilleur puisse les consulter, les modifier et les compléter si besoin est (figure 7). Elles seront ensuite mémorisées dans la base de données.

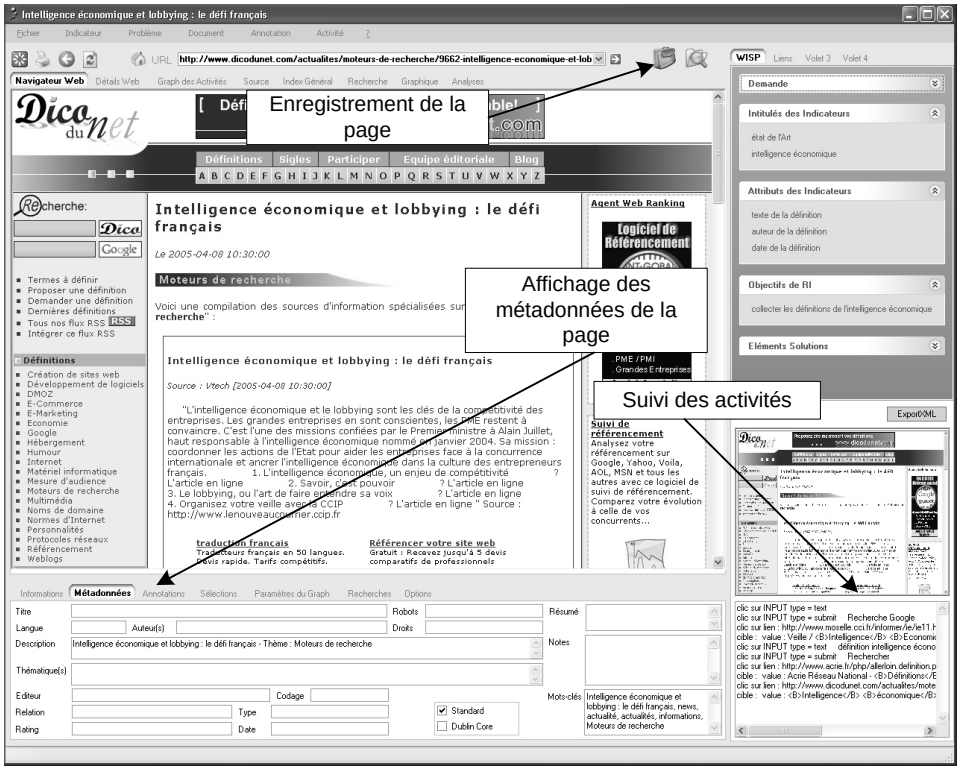

Figure 7 - L'accès aux métadonnées

Le Prototype va dans un premier temps détecter si les métadonnées de la page sont au format 'standard' ou au format 'Dublin-Core'. Il va dans un second temps, extraire ces métadonnées du code source de la page et les afficher dans le formulaire pour que l'utilisateur puisse en plus de les consulter, les modifier et les compléter si nécessaire. Si l'utilisateur souhaite sauvegarder la page entière, il peut le faire par un clic sur l'icône 'cartable' en haut près du champ d'adresse. La page sera alors compressée (MHT) et enregistrée en tâche de fond dans le répertoire défini par l'utilisateur (onglet 'options'). Les métadonnées seront quant à elles sauvegardées dans la base de données.
Note : nous voyons dans le suivi des activités, les actions du veilleur qui ont été mémorisées, les adresses des sites visitées, les mots-clés des requêtes, et le nom des objets qui ont reçu les clics souris.

3.7 Accès aux documents mémorisés et à l'historique des activités du veilleur

Le veilleur peut accéder aux documents mémorisés par l'intermédiaire du module de recherche (figure 5) ou par la formulation de requêtes sur la base de données. Il peut également retrouver les documents en consultant l'historique des activités (figure 8). L'historique permet au veilleur de visualiser sa séquence de recherche comme s'il se repassait une bande vidéo et de sélectionner tout élément informationnel pour l'afficher, le modifier ou l'effacer.

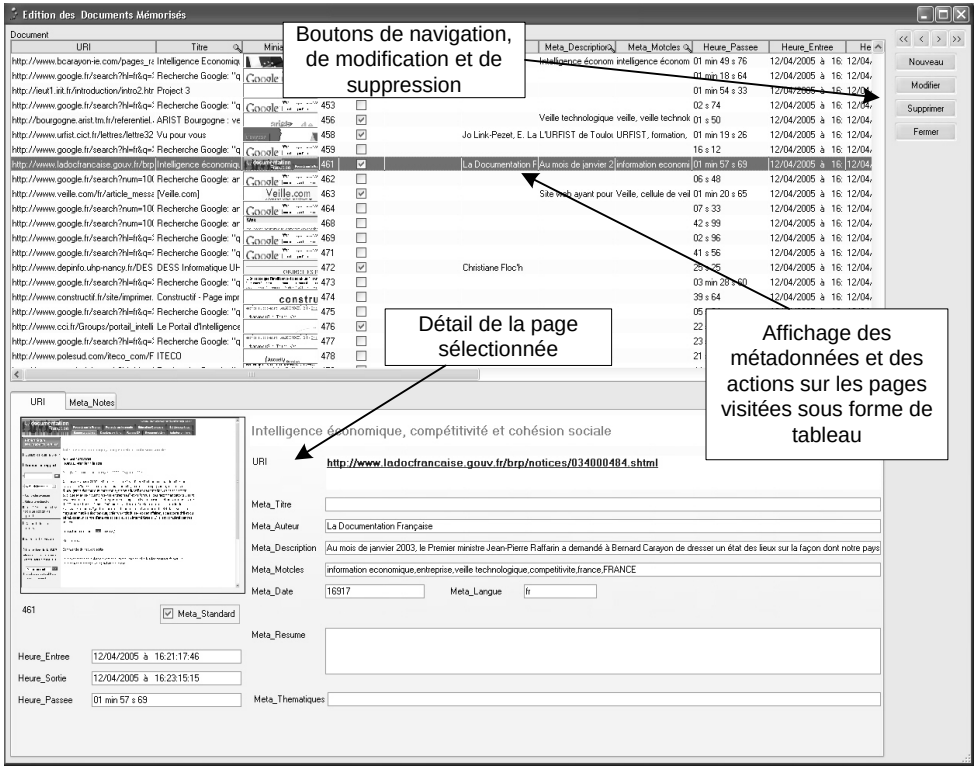

Figure 8 - L'accès aux documents mémorisés dans METIORE

3.8 Modules additionnels : Moteur d'analyse de cooccurrences et cartographie d'information

Le prototype Metiore, dans sa version actuelle, intègre également deux modules complémentaires. Ceux-ci, à la différence des autres modules, ne sont pas d'un usage direct et nécessitent quelques opérations de post-traitement. Ils seront utiles au veilleur et au décideur pendant la phase d'analyse afin de leur apporter d'autres représentations sur les informations collectées et sur le déroulement du processus.

La notion de cooccurrence que nous employons ici, fait référence au phénomène général par lequel des mots sont susceptibles d'être utilisés dans un même contexte [7]. Elle se caractérise par « *la coprésence de notions, mots ou de toute autre régularité lexicale, syntaxique ou sémantique à l'intérieur d'une unité de contexte définie* » [8]. Rechercher les cooccurrences correspond alors d'une manière très synthétique à identifier quels sont les mots qui « *s'attirent les uns les autres* » [9]. L'objectif sera donc de faire émerger, au sein d'un corpus textuel particulier, les associations de termes (ou concomitances lexicales) en utilisant la statistique

textuelle. Si des associations peuvent être prévisibles par le veilleur sur un domaine connu (comme dans celui de l'enseignement où la présence du mot 'étudiant' peut impliquer de trouver également celui de 'professeur'), dans un domaine qu'il ne maîtrise pas, l'analyse des fréquences de cooccurrence va permettre au veilleur de découvrir quels sont les termes qui se rencontrent le plus souvent. Cette découverte de termes et d'associations de termes va donc favoriser la création de sens et de connaissances.

Le moteur de cooccurrence que nous avons appelé '*CrossAnalyzer*' peut traiter tout type d'information (et pas uniquement des références bibliographiques comme à l'origine). Il a été écrit en langage C et sous la forme d'un module indépendant (une DLL ou librairie dynamique). Ce module peut être ainsi utilisé par le prototype, mais également par d'autres applications comme un logiciel documentaire ou un serveur web. Il permet de faire toute opération de comptage d'occurrences simples (le nombre de fois qu'un terme apparaît dans un document, qu'un auteur est cité dans un corpus par exemple) mais également des calculs d'occurrences plus complexes et ce, jusqu'à trois champs distincts (une analyse de cosignatures d'articles par date, un dénombrement des cooccurrences de mots-clés par auteur, etc.). Pour pouvoir être traités par le *CrossAnalyser*, les documents doivent comporter des éléments identifiés (sous la forme de balises XML) Si ce n'est pas le cas, le veilleur devra explicitement indiquer au moteur d'analyse quels sont les contenus qui seront associés aux éléments à analyser. De plus, comme le modèle WISP est au format XML, le *CrossAnalyzer* peut donc analyser toutes les informations de celui-ci. Il est ainsi possible de réaliser des opérations de dénombrement sur tous ses éléments : calcul de fréquences de cooccurrences des termes de la demande avec ceux des objectifs de recherche, des termes contenus dans les annotations pour un auteur en particulier, etc. Une autre analyse que nous avons envisagée est d'analyser les cooccurrences entre les résultats d'un moteur de recherche et les termes de la requête qui lui est transmise. De cette façon, le veilleur peut découvrir quels sont les termes qui sont associés avec ceux de sa requête (en plus des cooccurrences 'prévisibles') pour produire d'autres requêtes et surtout enrichir ses connaissances sur le domaine.

Le second module que nous avons implanté dans Metiore permet non seulement d'obtenir des représentations graphiques des résultats issus du *CrossAnalyzer* mais également des informations extraites du modèle WISP. Pour développer ce module, nous avons utilisé la bibliothèque « *Graphviz* » de AT&T qui se présente tout comme le moteur de cooccurrence, sous la forme d'une librairie dynamique. *Graphviz* implémente un langage de script très facile d'utilisation (le langage DOT) permettant de générer des graphes et de les afficher selon divers formats d'image. En outre, le veilleur n'a pas à se préoccuper de l'agencement des termes à l'intérieur du graphe puisque ce sont les algorithmes du *Graphviz* qui vont le gérer automatiquement. Metiore peut ainsi, grâce à ce langage, générer des représentations cartographiques de l'information contenue dans un texte en le passant en paramètre au module *Graphviz* (figure 9)

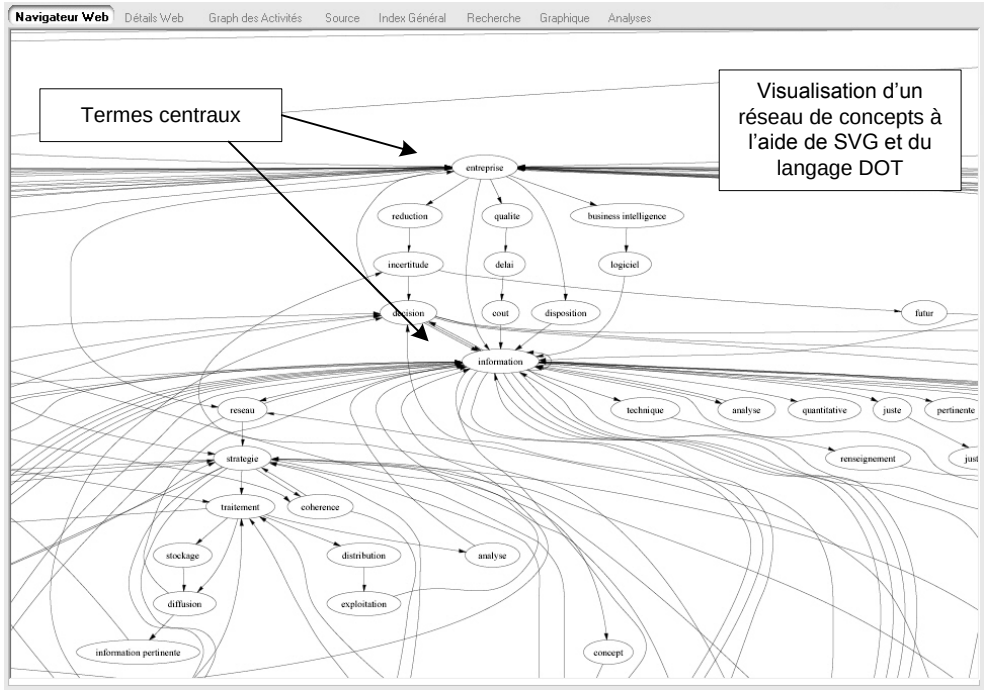

Figure 9 - Visualisation graphique d'informations issues d'une page web (Extrait)

Le module Graphviz nécessite néanmoins quelques prétraitements (remplacer les voyelles accentuées, supprimer la ponctuation et les espaces superflus,…), la plupart de ceux-ci étant réalisés par METIORE avant le passage du texte en paramètre. Cependant, il reste encore de nombreux bruits dans le document que METIORE ne peut pas traiter. Nous voyons néanmoins apparaître ici les termes centraux ('entreprise', 'information') qui sont reliés par un grand nombre de flèches, mais également d'autres terme périphériques comme 'décision', 'stratégie', 'traitement' notamment. La page affichée ayant été obtenue avec la requête 'intelligence économique', l'utilisateur peut ainsi obtenir une cartographie des termes du domaine. Notons ici, que les mots vides ont été supprimés par l'analyseur.

Parmi les différents formats d'images générés par *Graphviz*, nous avons retenu le format SVG (*Scalar Vector Graphic*) qui présente la particularité de coder les éléments d'une image dans un document dérivé de XML. Les avantages que nous avons perçus sont nombreux :

- Les graphiques SVG sont affichés dans tout navigateur web, simplement en ajoutant le plug-in correspondant. Les graphiques créés sont dynamiques et interactifs (ajouts de liens, d'animations). Ils sont en outre redimensionnables sans perte de qualité puisqu'ils sont définis de façon vectorielle ;
- Le SVG étant sous forme textuelle, son contenu peut être directement indexé par Metiore et le veilleur peut ainsi retrouver une information dans une image, ce qui n'est pas envisageable avec la plupart des autres formats d'image généralement utilisés ;

Enfin, le SVG étant structuré selon un formalisme XML, tous les éléments et tous les attributs du document SVG sont accessibles par l'intermédiaire du DOM. Il est donc possible d'associer des éléments du WISP et de les afficher directement en SVG (figure 10).

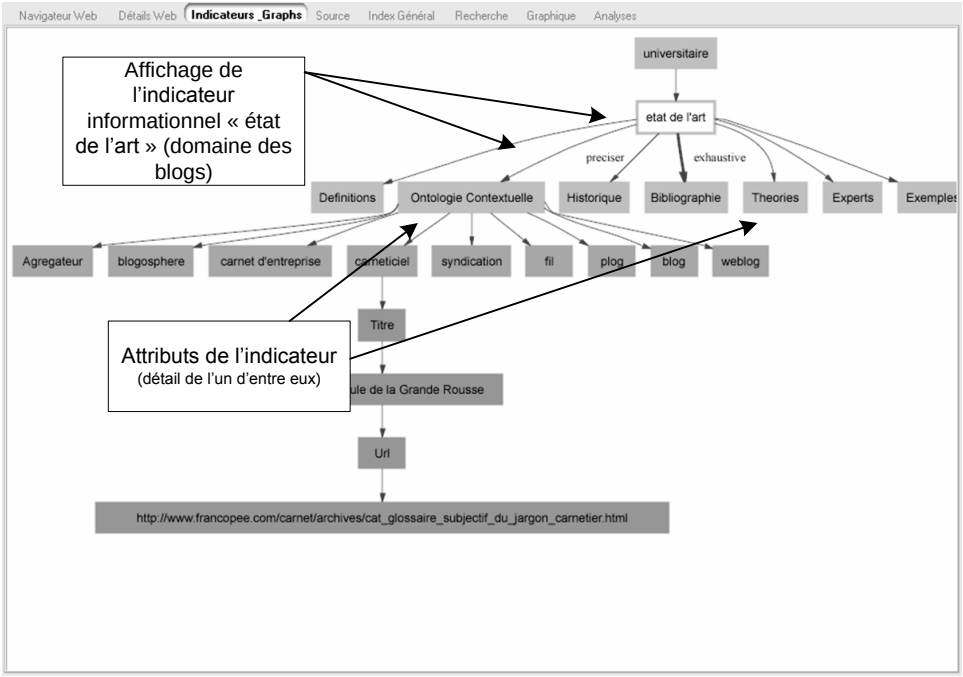

Figure 10 - Affichage de quelques éléments du WISP sous forme graphique

Nous apercevons ici les attributs de l'indicateur « Etat de l'art » affichés sous la forme d'un graphique SVG grâce au module Graphviz. Pour des raisons de lisibilité, nous n'avons développé qu'un seul attribut (ici Ontologie Contextuelle) et les références du document où l'utilisateur a retenu le terme'carneticiel'.

4 Conclusion

Nous avons développé dans cet article, les différentes fonctionnalités du prototype Metiore. Cet outil informatique permet d'assister le veilleur dans ses activités de recherche d'information tout en permettant au décideur de superviser et d'intervenir à tout moment sur l'intégralité du processus de veille. Il implante le modèle WISP dont la finalité est de favoriser la traduction du problème décisionnel en problème informationnel et d'optimiser les délais d'accès aux sources pertinentes et aux documents mémorisés.

Parmi les évolutions possibles de notre prototype, nous réfléchissons à l'utilisation de divers modules complémentaires qui permettraient :

- D'ajouter des annotations directement dans la page affichée par le navigateur : celles-ci n'étant actuellement traitées par le prototype qu'au sein des sélections de texte ;
- D'éditer des cartes conceptuelles au format SVG afin de pouvoir les indexer directement sans avoir à réaliser des opérations de post-traitement ;
- De formuler les objectifs de recherche et les requêtes à l'aide d'un logiciel de dictée vocale afin d'alléger les saisies de texte au clavier qui peuvent être quelquefois pénibles et contraignantes. Quelques essais ont déjà été réalisés avec des produits du marché et les résultats obtenus sont très encourageants. De plus, nous sommes actuellement en train d'expérimenter les différentes possibilités d'utilisation des annotations vocales au sein de notre prototype ;
- Enfin de bloquer l'affichage des fenêtres publicitaires qui surgissent à n'importe quel moment dans le navigateur et qui nécessitent de la part de l'utilisateur un important nettoyage les historiques de recherche dans la base de données.

5 Bibliographie

[1] KISLIN P., DAVID A., *De la caractérisation de l'espace-problème décisionnel à l'élaboration des éléments de solution en recherché d'information dans un contexte d'Intelligence Economique: le modèle WISP*, IERA'2003, Nancy: INIST, 14-15 avril 2003.

[2] KISLIN P., *Les activités de recherché d'information du veilleur dans le contexte d'IE : le modèle WISP*, Paru dans : Organisation des connaissances dans les systèmes d'informations orientés utilisation : contexte de veille et d'intelligence économique, Amos David (dir), Nancy : PUN, 97-118, 2005.

[3] DAVID A., *Modélisation de l'utilisateur et recherche coopérative d'information dans les systèmes de recherche d'informations multimédia en vue de la personnalisation des réponses*, HDR : SIC, Université de Nancy II, 1999.

[4] BUENO D., DAVID A., *Metiore: To Personalized Information Retrieval System, International UM'2001*, 8th Conference one To use Modelling, Sonthofen, Germany, 2001.

[5] BUENO D., *Recomendaciòn personalizada de documentos en sistemas de recuperaciòn de la informaciòn basada en objetivos*, Thèse en informatique, Espagne : Université de Malaga, 2003.

[6] DAVID A., THIERY O., *Prise en compte du profil de l'utilisateur dans un Système d'Information Stratégique, VSST'2001, Système d'Information Elaborée, Bibliométrie, Linguistique, Intelligence Economique, Barcelone, Espagne, 15-19 octobre 2001*.

[7] LEBRATY J.F., *Nouvelles technologies de l'information et processus de prise de décision : modélisation, identification et interprétation*, Thèse en Sciences de Gestion, Université de Nice Sophia-Antipolis, octobre 1994.

[8] LAFON P., *Dépouillements et Statistiques en Lexicométrie*, Paris : Slatkine-Champion, 1984.

[9] MARTINEZ W., *Contribution à une méthodologie de l'analyse des cooccurrences lexicales multiples dans les corpus textuels*, Thèse en Sciences du Langage, Université de la Sorbonne nouvelle, Paris 3, 2003.