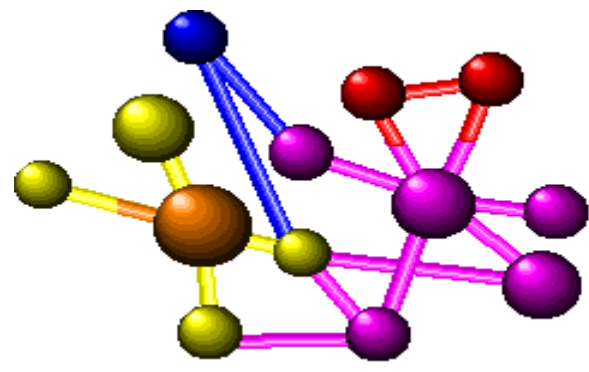
 <p>pêche écologique en Guinée</p> <p>B7-6200/99-03/DEV/ENV Rapport¹</p>	Implémentation d'une capacité d'apprentissage à un centre d'informations sur le secteur des pêches en Guinée	Rédacteur : Cheikh BA	Date création 07.10.02	Référence 52.004R/A
			Dernière modif. 02/11/02 23:24	Page 1 / 14

Diffusion :

- 1. affichage, M.Fofana
 - 2. documents du projet
- date de diffusion(s) : 15.10.02
 Dernière impression le : 02/02/03 15:48

Remerciements



Je voudrais particulièrement remercier mon responsable de stage, Mr Jean Le Fur pour la proposition du stage, son accueil et sa disponibilité malgré toutes ses contraintes de temps .

Mes remerciements vont aussi à Mr Philippe Reitz qui n'a pas hésité à être mon co-encadreur pour ce stage. Ses conseils sur les directions à prendre et ses suggestions de lecture m'ont été d'un très grand intérêt.

Enfin un grand remerciement à l'équipe informatique et au personnel du CNSHB pour leur gentillesse et leur aide, et grâce à qui j'ai eu un séjour agréable.

Sommaire

Remerciements 1

Sommaire..... 1

Introduction 2

Présentation..... 2

Le Centre d'information..... 3

Mécanismes de capture..... 7

Exploitations 12

¹ **Diplôme d'Etudes Approfondies «Informatique »**, Académie de Montpellier, Université de Montpellier II, Sciences et Techniques du Languedoc, , Date de soutenance : le 1^{er} Juillet 2002, Tuteurs de stage: Jean Le Fur (IRD-Conakry), Philippe Reitz(LIRMM).

Conclusion.....	14
Références bibliographiques	14
Annexes	14

Introduction

L'utilisation régulière de certains environnements ou interfaces cache souvent l'existence de nouvelles informations qui restent pour la plupart du temps « inconnues ». Il est donc nécessaire, pour faire surgir ces informations, de doter ces environnements d'un mécanisme d' « apprentissage » basé sur certaines techniques.

Le travail s'inscrit dans un projet qui vise à la conception d'un centre d'information(CI) sur les pêches en Guinée :un CI capable d'informer sur les questions actuelles, sans remettre en cause les approches antérieures ni les informations qui avaient alors été stockées ;en permettant de nouveaux points de vue et le stockage de nouveaux type d'informations.

Le CI se présente comme un site Internet générant du contenu de façon dynamique grâce à la technologie Servlet de Java.

Une information données peut être accédée à travers un sommaire ou par le choix d'un critère. Une fois l'information sélectionnée, elle est affichée avec les descripteurs qui la caractérisent. L'utilisateur peut alors naviguer d'une information à l'autre en sélectionnant les descripteurs qui l'intéressent.

En fonction des utilisateurs, des chemins sont plus empruntés que d'autres, certaines informations se correspondent, certains items prennent plus de poids, d'autres moins. . .

Il s'agira donc de voir comment, tout au long des différents sessions de navigation, repérer et constater certains effets ; ensuite de réfléchir sur des stratégies d'exploitation.

Présentation

L'IRD et le contexte du développement en Guinée

L'IRD est un établissement public à caractère scientifique et technologique qui, placé sous la tutelle des ministres chargés de la Recherche et de la Coopération, conduit des recherches sur les milieux intertropicaux depuis plus de cinquante ans. Ses recherches scientifiques sont centrées sur les relations entre l'homme et son environnement dans les régions tropicales et méditerranéenne, dans la perspective d'un développement durable de ces régions. L'Institut est présent en France métropolitaine, dans les territoires et départements d'Outre mer et dans plus de 40 pays du monde tropical dont la Guinée où j'ai effectué une partie de mon stage.

La Guinée est une république démocratique située en Afrique de l'Ouest. Ancienne colonie française indépendante depuis 1958, après une longue période de dictature sous Sékou Touré, le pays se mobilise dès 1984 pour donner un nouvel élan vers la voie démocratique en s'ouvrant au monde et en instaurant le multipartisme et les élections libres. Economiquement, la Guinée joue un rôle actif dans la production des minerais avec ses réserves de bauxite qui sont parmi les plus riches du monde et ses gisements de fer, de diamant, d'or et de l'uranium.

Toutefois, les risques de marginalisation subsistent toujours face à la globalisation de l'économie et la Guinée demeure un Etat de droit au service du développement.

Le CNSHB et le projet PEG

Dans ce contexte, l'IRD mène des projets de recherche en coopération avec des institutions locales en y apportant un fort appui technique, et c'est dans une de ces institutions, le CNSHB (Centre National des Sciences Halieutiques de Boussoura), que j'ai travaillé pendant mon séjour d'un mois à Conakry. Le CNSHB est un établissement public à caractère scientifique, placé sous la tutelle du Ministère chargé de la pêche. Sa mission est de contribuer au développement de la pêche maritime en Guinée par une meilleure connaissance et évaluation des ressources halieutiques. En effet, la pêche artisanale représente une des principales activités du pays : elle est pratiquée par environ 10000 pêcheurs principalement au moyen des pirogues dont la moitié sont équipées de moteur, et chaque année plusieurs dizaines de tonnes sont pêchées dont quelques milliers exportées.

Dans ce cadre, le projet PEG (Pêche Ecologique Guinéenne) qui a débuté en septembre 2000 et qui est mené par l'équipe de l'IRD et du CNSHB a pour objectif d'établir les modalités d'un développement durable de la pêche maritime guinéenne. Ce projet est fondé sur l'usage respectueux des écosystèmes marins afin d'éviter la surexploitation et donc l'extinction des ressources halieutiques guinéennes. Il consiste notamment à préserver les capacités productives de l'écosystème et à développer au sein du secteur une connaissance sur les effets de l'activité humaine.

Les principaux volets d'activité de ce projet sont l'acquisition de la connaissance sur le secteur de la pêche et la modélisation de l'écosystème marin et de l'exploitation guinéenne, qui sera restituée sous forme d'un logiciel de simulation.

Le Centre d'information

Le centre d'information s'adresse aussi bien aux chercheurs qu'aux différents acteurs de la pêche (pêcheurs, mareyeurs, gestionnaires...) donc à des utilisateurs de profils très différents. De même, les informations qui composent le centre sont très hétérogènes par leur nature, leur forme, leur mode de représentation et leur complexité. Par conséquent, il :

- dispose d'une architecture générique pour héberger les informations,
- permet l'intégration de toute nouvelle information et
- permet la restitution de l'information sous une forme adaptée.

Il se présente donc sous la forme d'un site Web dynamique relié à une base de données MySQL. La base stocke des pointeurs sur des fichiers qui contiennent l'information proprement dite ainsi qu'une série de descripteurs (dimension, source, mots-clés...) permettant de caractériser l'information. Chaque information de la base est identifiée par sa clé primaire qui est son numéro ID.

L'extraction des données contenues dans la base et la génération des pages HTML correspondantes se sont faits dans un environnement Java, combiné aux technologies XML et XSLT suivant le schéma suivant :

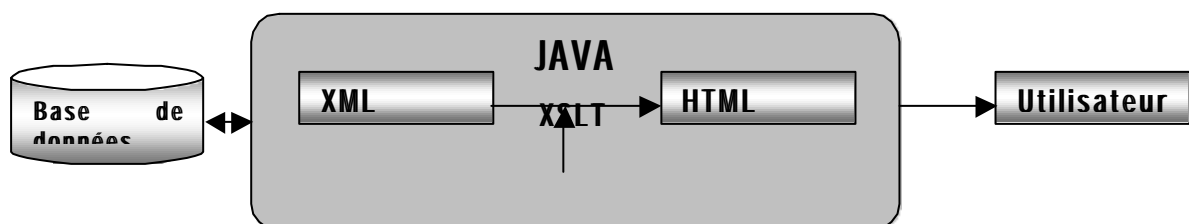
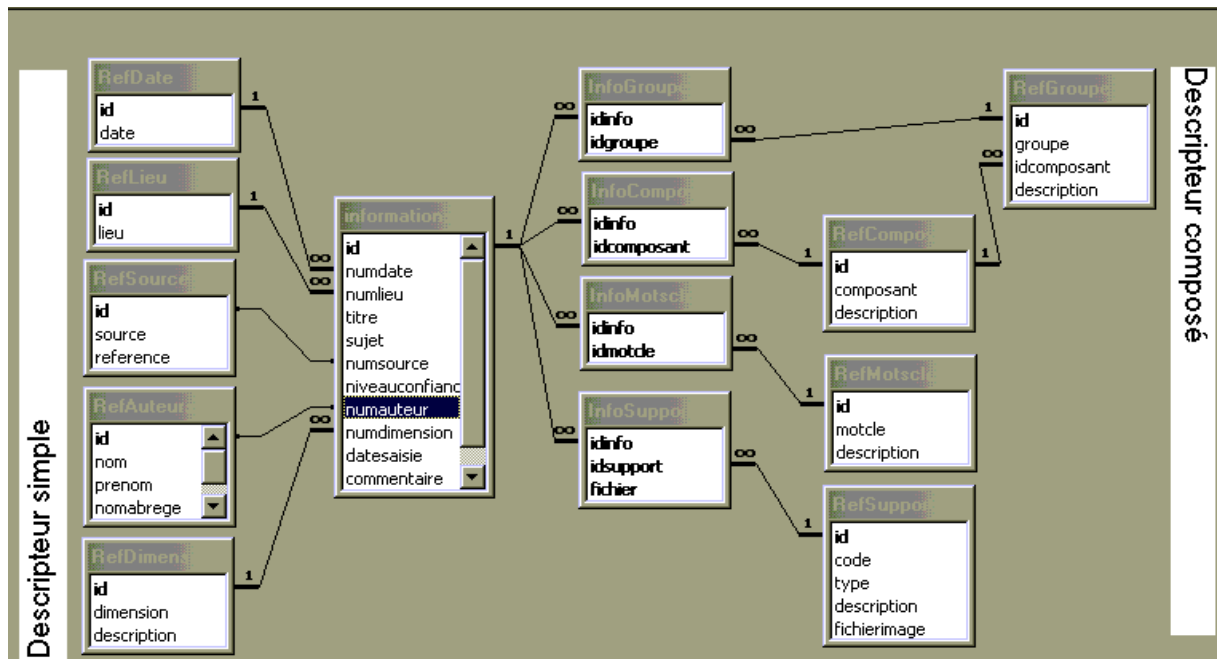


Schéma de génération de l'information

Modèle conceptuel de données

La structure de la base de données est présentée sur la figure suivante :



Quand java accède à la base, il réifie les éléments sans modification de la structure présentée (une table < > une classe).

Les choix technologiques

Une des préoccupations majeures pour concevoir un système d'information est son interopérabilité avec d'autres applications qui peuvent évoluer de manière très distincte. Les choix technologiques doivent donc répondre à ce besoin d'interopérabilité. C'est le cas du couple Java-XML/XSLT qui satisfait les critères de facilité d'utilisation, d'évolutivité et d'ouverture aux standards Web.

- Le XML

Le langage XML (Extensible Markup language) issu du SGML (Structured Generalized Markup Language) est un métalangage de définition de types de documents. Dans ce sens, il permet de fixer les règles de syntaxe de tout nouveau type de documents tout comme le SGML qui a défini le type de document HTML.

Le XML a l'avantage d'abolir les limites du HTML tout en évitant la trop grande complexité du SGML qui a freiné son développement. En effet, alors que le texte d'un document HTML repose sur des données non structurées qui ne peuvent être traitées par un programme, un document XML présente une structure intermédiaire entre celle rigoureuse d'une base de données et celle inexistante d'un texte. Il fournit ainsi des données aptes à être traitées.

Cette extensibilité du XML s'explique par l'introduction d'un nouveau type de balises appelées des balises sémantiques. Celles-ci ne donnent ni indication de structure ni celle de formatage mais sont accessibles aux scripts et aux programmes. Elles permettent ainsi l'application d'autres technologies telles que le HTML dynamique ou les feuilles de style. De cette manière, le XML accroît les possibilités de créer des descriptions adaptées aux besoins et de générer des pages plus interactives.

- Le XSLT

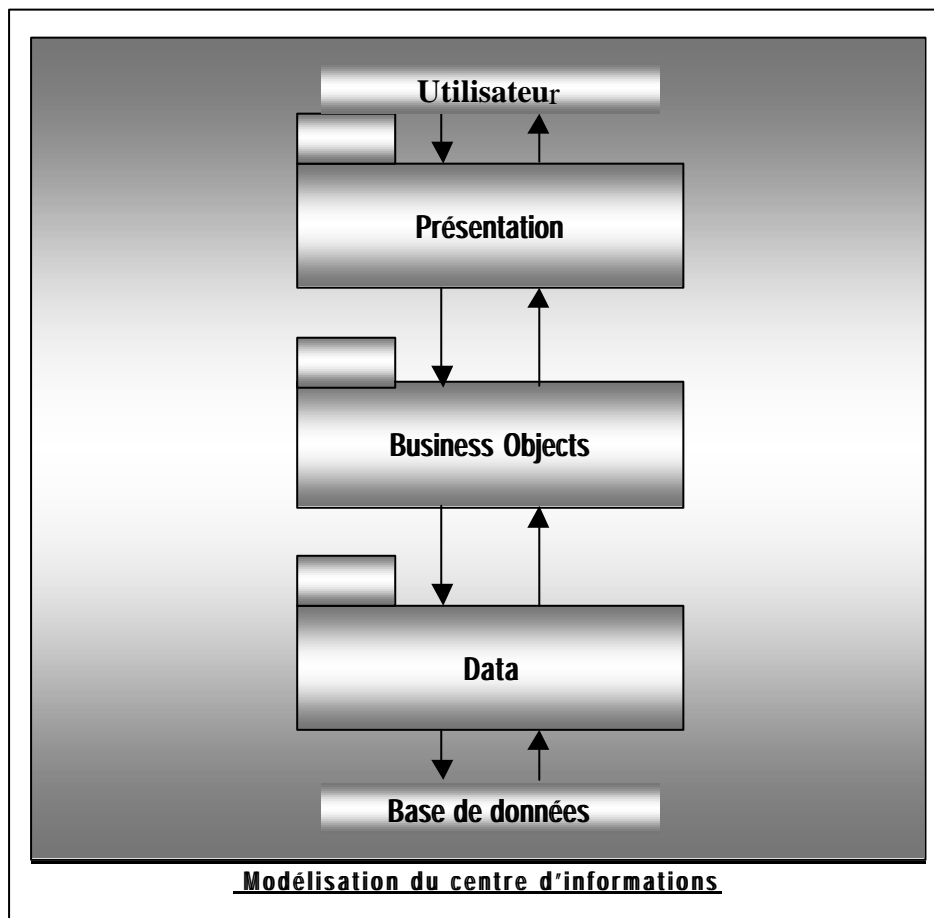
L'idée d'écarter les balises de formatage dans un document XML a consolidé son association avec le langage XSL (Extensible Stylesheet Language) qui permet gérer l'affichage des données. Le rôle du langage XSLT (XSL Transforms) est alors de transformer des documents en appliquant les feuilles de style XSL. Dans notre cas le programme XSLT contient des instructions de transformation d'un document XML en un document HTML.

Le document XML généré à partir de la base de données et contenant l'information sera le résultat d'un programme. Celui-ci sera envoyé à un processeur XSLT afin d'être converti en document HTML selon les instructions d'un document XSLT associé. Lorsque le document HTML est reçu par le navigateur, celui-ci ne sait pas que le document original est un document XML. Dans ce modèle, la transformation se fait du côté serveur et le navigateur affiche simplement le document HTML.

On aurait pu envoyer le document XML accompagné du document XSLT au navigateur, ce qui lui présente l'avantage de posséder en mémoire les deux documents XML et HTML et donc de pouvoir transformer le document XML plusieurs fois. Toutefois tous les navigateurs ne sont pas forcément capables de réaliser la transformation et ne possèdent pas un processeur XSLT. Notre modèle utilisera donc juste la fonction principale du navigateur qui est d'interpréter HTML.

L'approche objet

Un objet est une représentation abstraite d'un monde et possède un sens précis dans le contexte du problème. Il est par conséquent défini par un état caractérisé par les valeurs de ses attributs et des comportements qui sont décrits par ses méthodes.



La modélisation objet a pour but de décrire ces objets en représentant leurs relations dans le système. Cette approche de la modélisation prend son intérêt par rapport à la programmation traditionnelle dans la mesure où elle correspond étroitement au monde réel et permet la décomposition du problème en sous-problèmes. Le modèle objet est ainsi plus portable et plus souple pour les modifications.

L'application qui va de l'extraction des données à la génération des pages HTML a été

structurée en trois couches. Les couches représentent un ensemble de classes en Java et se communiquent entre-elles à travers des méthodes publiques d'une classe. En effet, dans la modélisation objet, on distingue la

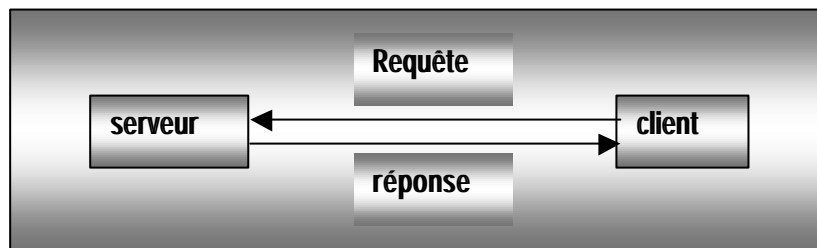
classe d'objet qui décrit un groupe d'objets ayant les mêmes propriétés et le même comportement et l'instance d'objet qui représente un objet particulier.

a. La couche Présentation

Elle représente l'interface avec l'utilisateur. Elle est composée de deux sous-couches :

- une permettant la création des documents XML à partir des objets générés par la couche Business Objects,
- une permettant la transformation d'un document XML en HTML en appliquant la feuille de style XSLT et l'envoi au navigateur des documents HTML ainsi générés à l'aide des Servlets.

Une Servlet est un programme en Java qui, exécuté au sein d'un serveur Web, permet de gérer la relation requête-réponse du client et du serveur :



b. La couche Business Objects

C'est dans cette couche que sont définis et créés les objets Java qui seront manipulés par le programme. Les principaux objets manipulés sont instanciés par les classes :

- **Information** qui comporte des variables d'instances privées représentant tous les descripteurs de l'information et des méthodes publiques permettant d'y accéder,
- **ListeInformations** qui comporte une collection d'informations ayant en commun les paramètres du descripteur et qui possède en méthode publique un itérateur permettant d'accéder à cette liste d'informations,
- **DescripteurSimple** et **DescripteurComposite** qui implémentent l'interface Descripteur. Un descripteur est dit simple lorsqu'à un type du descripteur on peut associer un contenu unique (ex : *support, dimension, date_info...*) et il est qualifié de composite lorsqu'à un type du descripteur on peut associer plusieurs contenus (ex : *mots_cles, composants*).

Un objet descripteur comporte donc en attributs privés une chaîne de caractères représentant le type du descripteur et une chaîne de caractères ou une collection représentant le ou les contenus du descripteur selon qu'il soit simple ou composite.

c. La couche Data

C'est dans cette couche que s'effectue la connexion avec la base de données et l'extraction de l'information requêtée sous un format générique. Cette information est ensuite envoyée à la couche supérieure Business Objects qui crée l'objet approprié.

Les interactions entre les classes de ces trois couches représentent la structure statique du système conçu.

Mécanismes de capture

Lors de la navigation, l'utilisateur est confronté à plusieurs types d'écrans qui correspondent à autant de niveau d'inspection du CI. Si l'on multiplie les utilisateurs, des « chemins » sont plus souvent empruntés que d'autres, certaines informations se correspondent, prennent plus de poids, d'autres moins etc.

Ainsi un premier objectif serait de pondérer les items afin d'obtenir un indicateur de la configuration informative du CI. En d'autres termes, identifier des ensembles de descripteurs très pertinents, pertinent, peu pertinent, pas pertinent. Ce qui équivaut à une simple pondération de l'information lorsqu'elle est « sollicitée », laquelle pondération impliquant celle de tous les items en « relation » avec l'information proprement dite (voir le modèle conceptuel de données du CI).

Pour ce faire, il suffit simplement d'ajouter un attribut **nb_acces** à la table **informations** et d'insérer un code d'incrémentation dans la méthode **GetObjetInformation()** de la classe **MyDataFactory**.



id
numdate
numlieu
titre
sujet
numsource
niveauconfiance
numauteur
numdimension
datesaisie
commentaire
nb_acces

Cette connaissance ne sera pas utilisée par l'utilisateur mais pourra servir par exemple à orienter la construction du CI de la façon la plus efficace (retrait ou modification de descripteurs).

Un deuxième objectif serait de pouvoir mettre en place une fonctionnalité nous permettant lors des navigations des différents utilisateurs, de capter les liens entre les informations visualisées. Ce qui donnera l'opportunité de discuter sur le tissu de relations entre informations ainsi créé.

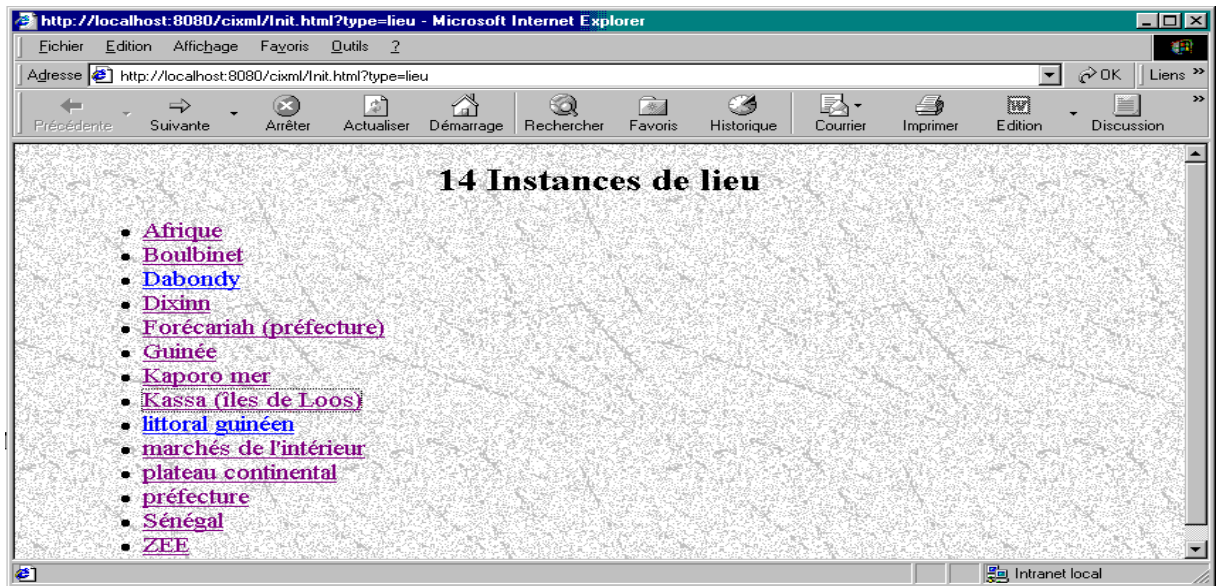
La mise au point de cette fonctionnalité sera un peu différente selon qu'il s'agisse de la version « stand alone » ou de la version à publier sur Internet.

Version « Stand Alone »

C'est la version sur laquelle les travaux de test se sont faits tout au début. Elle est quasiment mono-utilisateur.

La navigation se fait par une série d'appels de servlets java (au nombre de trois) générant dynamiquement des pages web après interrogation d'une base de données.

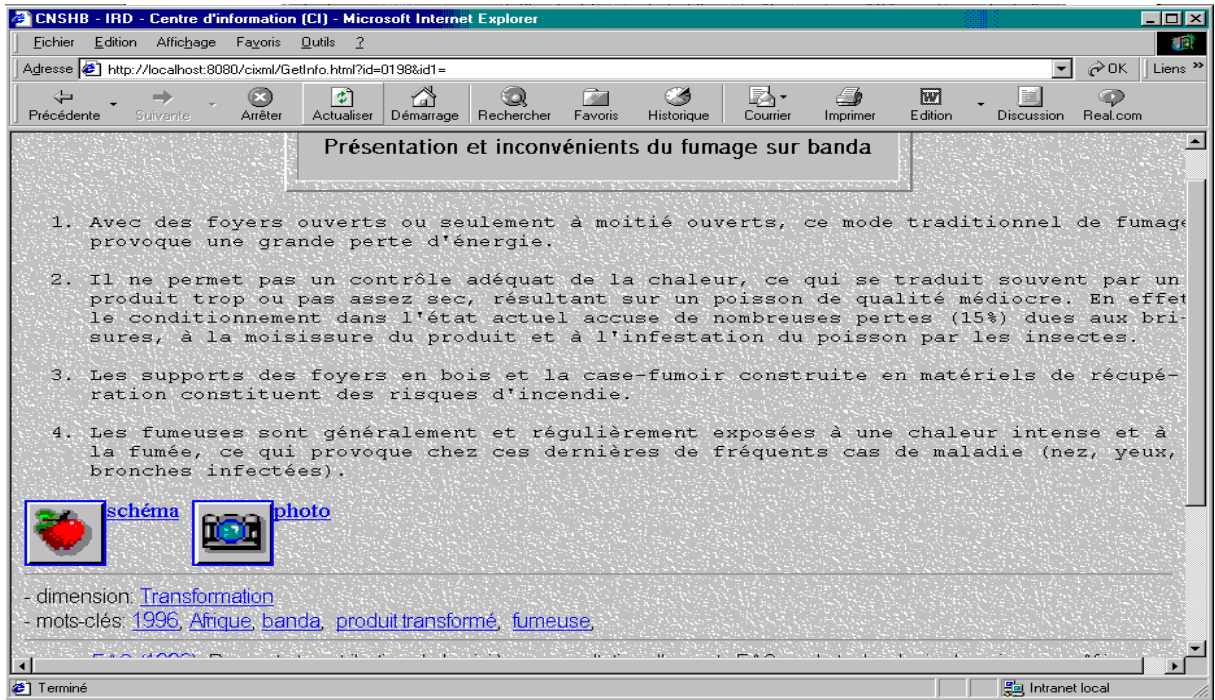
- **Init.html** (alias de la servlet **HttpGetServletInit**) qui prend en paramètre un type de descripteur et renvoie une page web contenant la liste des instances du descripteur. Cette servlet est appelée en inscrivant par exemple « Init.html?type=lieu » directement dans la zone de texte du browser réservée aux URLs.



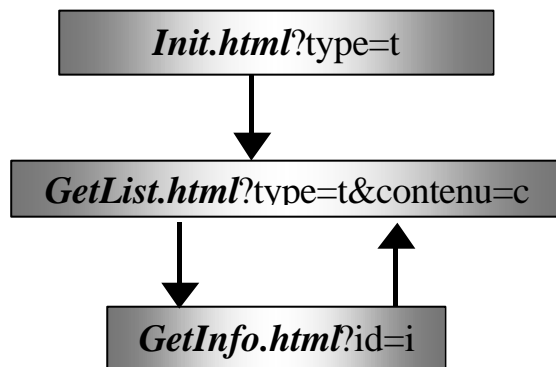
- **GetList.html** (alias de la servlet **HttpGetServletListInformations**) qui prend en paramètres le type et le contenu d'un descripteur puis affiche la liste des informations associées à ce contenu. Elle est appelée en cliquant soit sur un lien de la page générée par **Init.html**, soit sur un lien de la page générée par **GetInfo.html**.



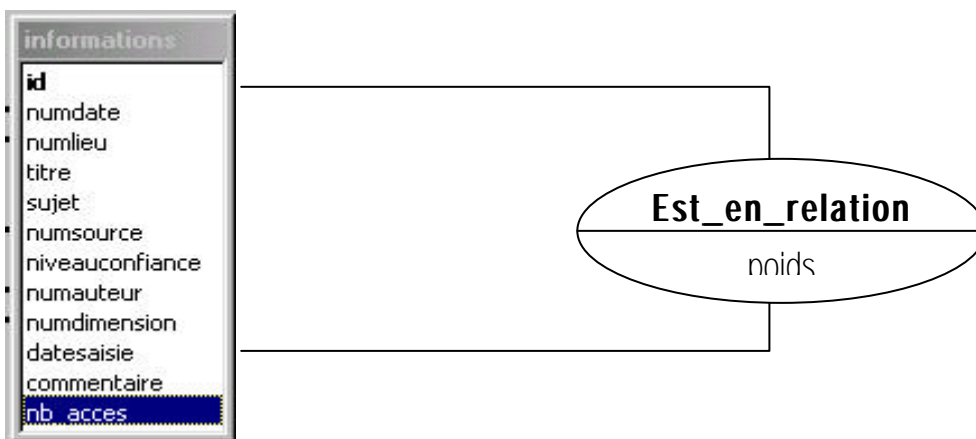
- **GetInfo.html** (alias de la servlet **HttpGetServletInformation**) qui prend en paramètre une id puis affiche la l'information et toutes les instances de descripteurs associées.



Ainsi le déroulement e la navigation peut être schématisé de la manière suivante :



« Une information I1 est en relation avec une autre information I2 s'il a existé un certain utilisateur U qui a visualisé I2 immédiatement après avoir visualisé I1 ». Ces liens vont être pondérés pour servir d'éléments d'analyse. Le modèle conceptuel suivant a été utilisé pour le stockage en dur :



Ce qui amène à créer une nouvelle table **lien_infos** dans la base.

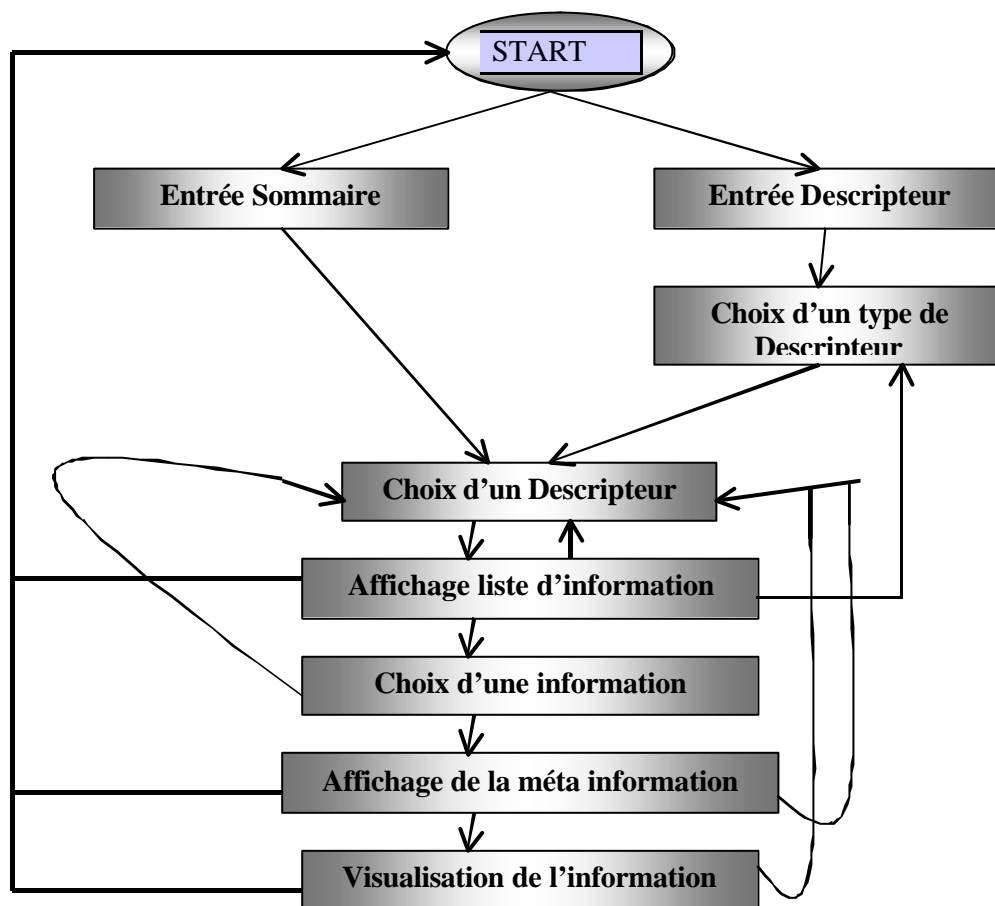
Lorsque l'utilisateur visualise une information, l'id est conservée jusqu'au prochain affichage d'une information. L'idée retenue consiste à considérer l'id comme argument supplémentaire d'appel des servlets **GetList** et **GetInfo**. Il a fallu aussi faire quelques modifications dans les méthodes générant les documents XML et dans les fichiers de formatage XSL. L'enregistrement d'un lien dans la base se fait par appel de la méthode **set_lien(String,String)** juste au moment de l'affichage d'une information .

lien_infos
Id1
Id2
...

Cependant, les possibilités offertes aux utilisateurs sont un peu limitées. En effet après la visualisation d'une information I, l'utilisateur ne pourra accéder qu'aux informations en rapport avec les seules instances de descripteurs associées à I. D'où l'intérêt de la version combinant des pages statiques aux pages dynamiques, offrant plus de flexibilité à la navigation et une bien meilleure présentation.

Version à publier

La navigation se passe différemment par rapport à la version « stand alone » ; Son déroulement peut être schématisé de la manière suivante :



L'algorithme reste le même que précédemment si l'utilisateur se limite seulement aux liens pointant vers les servlets. Mais cette éventualité est rare.

Vu le déroulement de la navigation, l'enregistrement des liens devient un peu plus complexe. En effet, le problème apparaît lorsque par exemple, après avoir visualisé une information, l'utilisateur passe par un jeu de clics sur les pages statiques avant d'accéder à une autre information. De ce fait, l'id ne pourra pas être conservé par les passages de paramètres (les pages statiques ne recevant pas d'argument). Il faut donc conserver l'id en dur quelque part.

Puisqu'une servlet peut être appelée par plusieurs utilisateurs en même temps, il est nécessaire de trouver un mécanisme de marquage d'id pour ne pas avoir des liens erronés.

Exemple de lien erroné :

Temps1 temps2

User1 I1 → création du lien (I1,I2) :erroné !

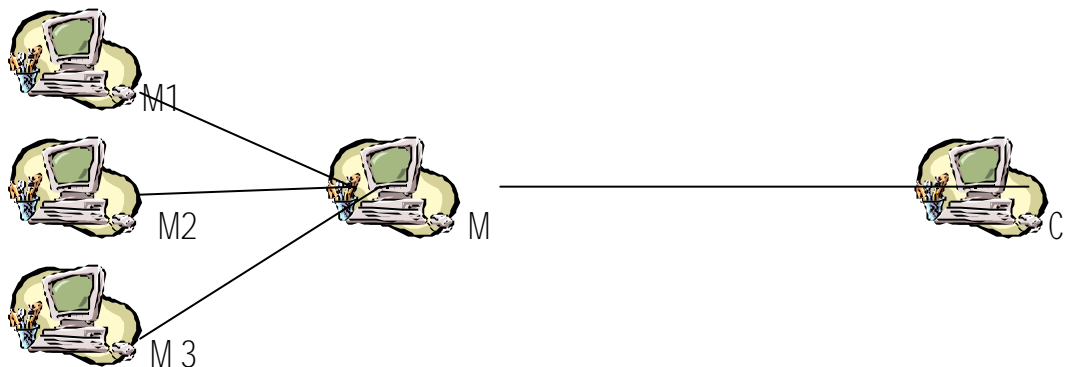
User2 I3

Une idée serait donc d'utiliser l'algorithme suivant :

1. création d'une table **tempo(id,ip,date)** ;
2. Si l'utilisateur accède à une information, on vérifie si son ip se trouve dans la table tempo.
3. Si c'est le cas, **tempo.id** est en relation avec cette information, on change **tempo.id** par l'id de la dernière information visualisée ;
4. Sinon, on ajoute le tuple (id,ip) dans **tempo**.

Cependant, il y a des situations que cette algorithme ne gère pas.

Supposons le cas de figure suivant :



Intranet

- Les machines M1, M2, M3 ne sont pas visibles de l'extérieur, donc possèdent la même adresse IP si jamais elles font des requêtes à la machine CI. L'algorithme liera deux informations même si ces dernières sont visualisées par des personnes différentes.
 - Même cas de figure si par exemple deux personnes accèdent chacun à une information sur la même machine.
- Une autre consisterait à utiliser une primitive offerte par Java, à savoir les *cookies*. Ils permettent de stocker temporairement un couple se « String » sur la machine cliente. Si un utilisateur visualise une information, l'id est stockée sur son poste. Ainsi, au prochain affichage d'une information, l'id est récupéré, le lien entre ces deux informations est créé ou pondéré. Ceci permet de lever l'ambiguïté sur les adresse IP ;

Mais cette possibilité n'a pas été bien accueillie vu que ce n'est pas toutes les machines qui acceptent l'utilisation des cookies.

Finalement, la solution adoptée est celle consistant à enregistrer l'id, l'IP et la date dans la base.

Reste maintenant à voir, après ces procédures de capture, comment les restituer, les exploiter, en tirer profit.

Exploitations

Le choix des captures supplémentaires(c'est à dire autres que le fichier des logs) s'explique par le fait que ces derniers nous limitent aux seuls traitements statistiques sur les fréquences des pages visitées.

Il m'a été proposé, pour une toute première fonctionnalité, de se focaliser sur les relations entre informations, créées lors des différentes navigations.

Une étude de l'état de l'art dans le domaine avait été faite pour voir s'il y avait d'éventuels algorithmes qui pourraient être appliqués à notre problème.

Concernant l'exploitation, s'agira-t-il d'un apprentissage au sens commun du terme ou d'un « CI mining » ?

1. Apprentissage :

« apprendre c'est identifier dans un espace d'hypothèses, l'hypothèse telle que le taux d'erreur et la probabilité d'erreur soient minimaux »[2].

Les nouveaux environnements interactifs sur lesquels l'utilisateur agit sont devenus les meilleurs moyens de communication entre l'homme et la machine.

On constate que leur utilisation s'accompagne de la répétition de nombreuses actions. D'où l'intérêt florissant des recherches sur ces types d'interfaces afin d'assister l'utilisateur. Il faut noter que ce type d'étude se focalise sur les habitudes individuelles :

- dire que sous Unix, 70% des commandes saisies sont des répétitions. [Greenberg et Written, 1993]
- dire que certains menus sont plus utilisés que d'autres(dans MSWord par exemple)
-

suppose que les suggestions devront se faire en fonction de l'utilisateur même. Les actions ou plus généralement les séquences d'action que l'utilisateur d'un environnement interactif est amené à répéter peuvent être différentes d'un utilisateur à l'autre, d'une session à une autre ou évoluer au cours d'une même session.

Cependant, cette approche reste est un peu différente de la nôtre. Il s'agit plutôt d'un « apprentissage par masse » basé sur une étude globale. En effet, vu la fonction du CI, vu les réalité du domaine, il est difficile d'apprendre les habitudes d'un utilisateur.

*« l'habitude est une tendance à répéter un acte que l'on a déjà accompli plusieurs fois »**Aristote**.*

Un même utilisateur aura du mal à parcourir à chaque fois les mêmes chemins pour retrouver les mêmes informations ;puisque l'information est supposée acquise au bout d'un petit nombre de visualisation.

Donc reconnaître les traces d'un utilisateur pour lui faciliter la tâche lors de sa prochaine navigation n'a pas lieu d'être.

L'apprentissage(au sens commun du terme), à mon avis serait de trouver d'éventuelles classes d'informations pour pouvoir faire à l'utilisateur des suggestions « pertinentes » de liens vers d'autres informations.

Explication :

Le CI peut être utilisé par différents types d'utilisateurs :pêcheurs, mareyeurs, modélisateur... Les informations aussi sont de plusieurs types. Il se pourrait qu'il y ait, pour un certain type d'utilisateur, des types particuliers d'informations qui les intéressent. Donc, après un certain nombre de connexions à la base, il y a des chances qu'il y ait émergence d'un phénomène : l'apparition de classes d'utilisateurs.

Le résultat d'un apprentissage serait donc de faire correspondre un utilisateur à une classe, puis de lui suggérer des liens.

2. Mining :

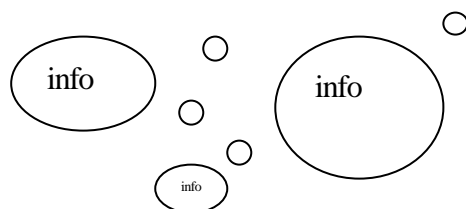
« on pourra définir le data mining comme une démarche ayant pour objectif de découvrir des relations et des faits, nouveaux et significatifs, sur de grands ensembles de données. »[3].

Exemple : Dans un super marché, si par exemple le fait de savoir que beaucoup de gens achètent de l'huile après avoir acheté du poisson, peut être utilisé pour définir les emplacements de ces produits au niveau des rayons.

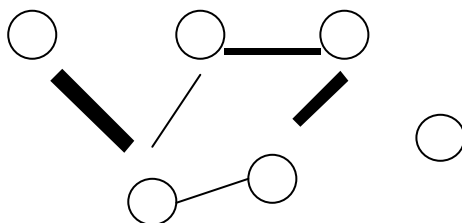
Le CI, fait dans le cadre d'un projet en cours, vient à peine d'être créé. L'ensemble des informations stockées peuvent en cacher d'autres qui sont « invisibles à l'œil nu », c'est à dire nécessitent des techniques particulières pour les faire émerger. Un exemple de ce type serait la détection des classes d'utilisateurs.

Finalement, ces deux aspects ont pu être combinés pour les procédures d'exploitation.

- il y a un mécanisme de pondération des informations à chaque fois qu'elles sont sollicitées. Ainsi, à long terme certaines informations pourront devenir « très lourdes », « lourdes », « peu lourdes »... Ce résultat sera intéressant pour les modélisateurs.



- A chaque fois que deux informations sont visualisées, un lien se crée automatiquement. Nous aurons donc un graphe de liens entre informations, et l'apparition d'un circuit pourra par exemple être considéré comme une classe d'informations. L'étude de ces classes est nécessaire pour l'analyse des classes d'utilisateurs.
- A partir du moment où les classes d'informations ont commencé à émerger, une assistance est envisageable. Si à un moment quelconque d'une navigation, le moteur arrive à faire correspondre un utilisateur à une classe, il peut s'en suivre des suggestions de liens vers des informations qui l'intéressent le « plus ». Ceci peut se faire en se basant sur les pourcentages de poids des liens entre les informations.



- Si jamais l'émergence des classes n'a pas encore eu lieu, la suggestion pourra se faire en fonction des statistiques globales :
 - o soit on lui suggère les n premières informations les plus visualisées ;
 - o soit les n plus visualisées ayant un lien avec l'information même.

Conclusion

Au cours de ce stage au CNSHB, dans le projet PEG, j'ai mis au point des algorithmes de stockage d'informations sur les navigations ; ceci afin de pouvoir observer le déroulement de certains faits.

Malheureusement, le temps ne nous a pas permis de tester les procédures d'exploitation proposés. Le CI est destiné à une utilisation non restreinte, à la portée de tous, puisqu'il sera déployé sur le réseau Internet, cause pour laquelle l'émergence de certains phénomènes peut tarder à avoir lieu.

En retour, si jamais il arrive qu'il y ait assez de « vrais » navigateurs, c'est à dire des navigateurs s'intéressant au domaine, rien que la classification sera d'une très grande utilité.

L'inconvénient majeur c'est qu'il peut y avoir déjà une classe d'utilisateurs qui existe réellement et qui ne pourra jamais connu par le CI : il s'agit de certains acteurs de la pêche pour qui ce sera difficile, voir même impossible de consulter le CI.

Nous nous sommes seulement arrêtés à l'étude de liens entre utilisateurs ; ceci est bien sûr très important, mais une perspective serait de s'intéresser aux chemins (courts ou longs) parcourus, dont les nœuds ne sont pas seulement des informations, mais des informations et des descripteurs.

Références bibliographiques

1. Misako Ito (2000) « Implémentation d'un centre d'information à l'IRD de Guinée » Rapport de stage.
2. Jean-David Ruvini « Assistance à l'utilisation d'un environnement interactif : Apprentissage des habitudes de l'utilisateur » ; Thèse de Doctorat de l'Université de Montpellier II. 2000
3. José Luis Cabral de Moura, Borges « A data mining model to capture user navigation patterns » Department of computer science, University College, London July 2000

Annexes

Sont présentés en annexes les diagrammes de classes simplifiés du CI et de la couche Business object