

cismef.pdf

Hervé Ly-Sunnaram
logoASI.pdf

Rapport de stage de spécialité

Refonte de la plateforme d'indexation de documents médicaux

Adaptation à une nouvelle architecture de Base de Données

Table des matières

Remerciements

Je souhaiterais tout d'abord remercier Stefan Darmoni et Benoit Thirion, tous deux coordinateurs du CISMeF, pour m'avoir accueilli chaleureusement au sein de leur équipe.

Je voudrais ensuite remercier mon tuteur de stage, Badisse Dahamna, pour toute l'aide qu'il m'a apporté et tout le temps qu'il m'a consacré pour mener à bien mon stage de spécialité. De plus, je souhaiterais remercier les autres membres de l'équipe technique du CISMeF, Ivan Kergourlay et Julien Grosjean entre autres, pour avoir pris de leur temps pour répondre à mes questions et Gaétan Kerdelhué, documentaliste, pour ses explications sur l'utilisation de la Plateforme d'Indexation.

Enfin, je voudrais remercier Pascal Mazon, sans qui ce stage ne se serait pas aussi bien déroulé.

Glossaire

ASI Architecture des Systèmes d'Information

CHU Centre Hospitalier Universitaire

CISMeF Catalogue et Index des Sites Medicaux de langue Française

INSA Institut National des Sciences Appliquées

LOM Learning Object Metadata

MeSH Medical Subject Headings

MVC Modèle-Vue-Contrôleur

PIC Projet INSA Certifié (projet industriel à mi-temps effectué par une équipe de 4-9 élèves-ingénieurs)

POO Programmation Orientée Objet

Thésaurus "Type de langage documentaire qui consiste en une liste de termes sur un domaine de connaissances reliés entre eux par des relations synonymiques, hiérarchiques et associatives"
[Wikipédia : Thésaurus]

Chapitre 1

Introduction

Le département Architecture des Systèmes d'Information (ASI) de l'INSA de Rouen offre une formation qui est consolidée par deux stages. Ce rapport constitue une synthèse du travail effectué lors de mon stage de spécialité réalisé au sein de l'équipe du CISMéF du CHU de Rouen entre le 28 Juin et le 22 Août 2010.

Le stage de spécialité permet de découvrir d'une part le monde de l'entreprise dans un des domaines de spécialité du département et permet d'appliquer d'autre part les connaissances acquises au cours du cursus ASI.

Étant membre de l'équipe Projet INSA Certifié (PIC) CHU, le premier semestre de ce PIC m'avait déjà permis de travailler avec le CISMéF sur la conception d'une nouvelle architecture de base de données. Or, ce stage se situe dans la continuité directe de cette conception. En effet, il consistait à faire migrer l'un des outils du CISMéF, à savoir la Plateforme d'Indexation de ressources, sur cette nouvelle architecture de base de données. Il est à noter que j'ai effectué ce stage en équipe avec Pascal Mazon, lui aussi membre de l'équipe PIC CHU 2010.

Ce rapport se divisera en 3 parties. Dans un premier temps, je donnerai une description du CISMéF, de ses objectifs et de quelques uns de ses outils. S'en suivra une présentation du travail effectué lors de ce stage de huit semaines. Enfin, je conclurai sur ce qui a été réalisé et sur les améliorations possibles à apporter.

Chapitre 2

Présentation du CISMeF

2.1 Le projet CISMeF

Avec l'afflux d'informations disponibles sur Internet, le personnel médical a besoin d'avoir à sa disposition des outils fiables lui permettant de rechercher et trouver des ressources pertinentes de manière efficace sur Internet. C'est dans ce but que le projet Catalogue et Index des Sites Médicaux de langue Française (CISMeF) a débuté en 1995. Ce projet est né suite au constat du manque de catalogues de ressources médicales francophones. Il avait donc comme objectif premier l'indexation de ressources francophones dédiées à la santé.

Là où les moteurs de recherche tels que Google ou Yahoo indexent leurs ressources de manière automatique en effectuant des indexations plein texte qui se basent à la fois sur les méta-données indiquées dans la ressource et sur le contenu lui-même, la majorité des ressources indexées par le CISMeF sont quant à elles étudiées et étiquetées par des documentalistes professionnels à l'aide de descripteurs issus de thésaurus reconnus tel que le thésaurus Medical Subject Headings (MeSH). Par ce biais, nous obtenons une indexation de qualité selon un vocabulaire normalisé spécifique aux ressources médicales qui permet par la suite d'effectuer des recherches bien plus précises et pertinentes.

2.2 Le Laboratoire du CISMeF

Le CISMeF est situé dans les locaux du Centre Hospitalier Universitaire de Rouen. L'équipe du CISMeF fait partie de l'équipe Traitement de l'Information en Biologie et Santé (TIBS) du Laboratoire LITIS EA 4108.

L'organigramme ?? représente la hiérarchie de l'équipe du CISMeF. Elle est coordonnée par Stephan Darmoni, professeur d'Informatique et médecin à l'Université de Rouen, et Benoit Thirion, conservateur des bibliothèques. Cette équipe regroupe différentes professions nécessaires au développement, à la maintenance et à l'alimentation en données du Systèmes d'Information du CISMeF. Ainsi, elle est composée actuellement de trois ingénieurs de recherche (Badisse Dahamna, Ivan Kergourlay et Julien Grosjean), deux documentalistes de la Bibliothèque Médicale (Catherine Letord et Gaétan Kerdelhué) ainsi que plusieurs doctorants du LITIS (Saoussen Sakji et Tayeb Merabti)[?]. Par ailleurs, cette équipe est régulièrement renforcée par des médecins et des stagiaires ainsi que par les PIC de l'INSA de Rouen.

Le financement est issu principalement de fonds publics : Ministère de la Recherche, Université Médicale Virtuelle Francophone, Agence Nationale de la Recherche[?]. Il est de l'ordre de 200k€ annuels. Ce financement est variable selon les projets en cours.

2.3 Les outils du CISMeF

Le **portail du CISMeF** [?] regroupe plusieurs outils mis à la disposition du personnel de santé, des familles et des patients.

En 2000, le CISMeF a publié l'outil **Doc'CISMeF**. Il permet d'effectuer des recherches dans la base qui contient maintenant près de 50000 ressources indexées manuellement. La recherche

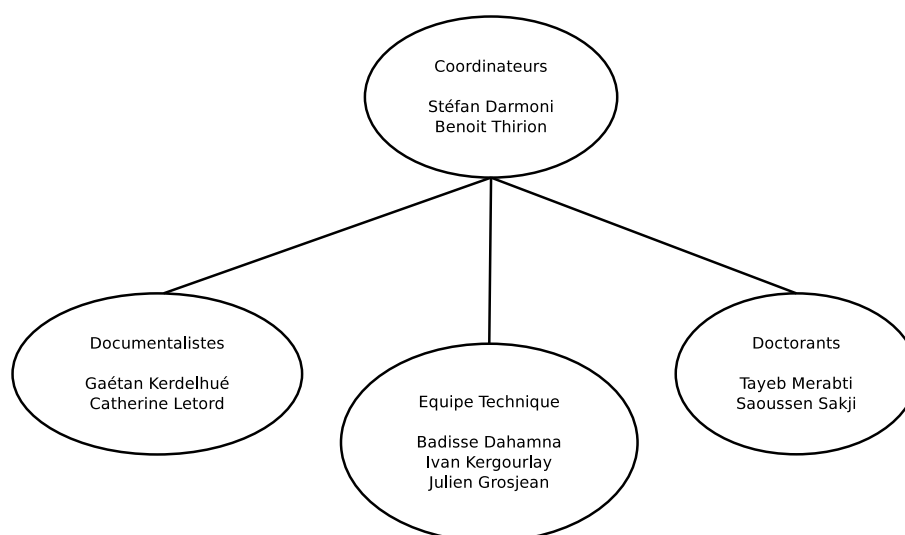


FIGURE 2.1 – Organigramme de l'équipe du CISMeF

s'effectue via des mots-clés ou via une grammaire plus spécifique utilisant les opérateurs booléens et des codes suffixes. Par exemple, pour rechercher le mot clé "asthme" dans un titre, il suffit de suffixer ".ti" à asthme (asthme.ti).

Par ailleurs, il propose quatre outils de recherche supplémentaires :

- Recherche suivant un index alphabétique basé sur les descripteurs de la terminologies MeSH,
- Recherche suivant un classement thématique basé sur des métatermes Catalogue et Index des Sites Medicaux de langue Française et des descripteurs MeSH,
- Recherche suivant un type de ressource basé sur les types de publication et les descripteurs MeSH,
- Recherche via le Portail Terminologique de Santé. Cet outil permet d'effectuer une recherche selon un descripteur issu d'une terminologie et d'obtenir toutes les informations relatives à ce descripteur : description et définition, hiérarchie, ressources liées au descripteur.

En matière de qualité de l'information, le site du CISMeF a reçu la certification du Health On the Net (HON) Code qui est une certification basée entre autres sur l'éthique du site (confidentialité vis à vis des informations personnelles des visiteurs, identité des administrateurs du site), sur la transparence des financements et sur le professionnalisme vis à vis de l'information contenue dans le site (avis justifié et critique sur les traitements médicamenteux par exemple). Par ailleurs, le site du CISMeF suit aussi les critères de la qualité de l'information de la santé dictés par le Net Scoring. Parmi ceux-ci, il y a les aspects déontologiques, la crédibilité de la ressource ou encore la véracité du contenu.

En ce qui concerne les outils internes, le CISMeF est doté d'une Plateforme d'Indexation de ressources qui a été réalisée lors de la campagne 2005-2006 des PIC INSA Rouen. Cette plateforme est utilisée principalement par les documentalistes de la Bibliothèque Médicale. Elle leur permet d'ajouter différents types de métadonnées aux ressources internet. Parmi celles-ci, il y a 10 métadonnées issues du Dublin Core (titre, description, auteur, ...) et 11 métadonnées issues de la catégorie "Educational" du LOM [?].

2.4 Méthodologie de l'Indexation des Ressources

La détermination des ressources à indexer est exclusivement manuelle. Les documentalistes travaillant au CISMeF effectuent une veille sur les sites institutionnels, gouvernementaux ou encore les sites des professionnels de la santé afin d'identifier les ressources destinées en tout pre-

mier lieu aux professionnels de la santé, en particulier les recommandations. Ils recensent aussi les cours universitaires destinés aux étudiants en médecine et enfin les ressources pratiques pour les patients et les familles des patients.

Le CISMeF dispose de trois niveaux d'indexation pour une fiche. Les ressources dites "N1" sont celles qui sont indexées totalement manuellement. Elles représentent plus de 50% des ressources disponibles via les recherches. Les ressources "N2" sont indexées automatiquement grâce à un algorithme de sac de mots et cette indexation automatique est ensuite vérifiée à la main par les indexeurs. Enfin, les ressources "N3" sont les ressources moins importantes (cours sur un sujet déjà traité de multiples fois par exemple). Elles sont indexées automatiquement et non vérifiées ensuite.

Lors d'une recherche via **Doc'CISMeF**, les ressources "N1" sont affichées en premier car elles sont les plus importantes et pertinentes, suivent les ressources "N2" puis enfin les "N3".

2.5 Les réalisations du CISMeF

Grâce à son savoir-faire en matières de recherches d'informations à l'intérieur de terminologies, le CISMeF a réalisé plusieurs moteurs de recherche. Parmi ceux-ci, il y a bien entendu **Doc'CISMeF** [?], le moteur de recherche du CISMeF, le moteur de recherche **KISMeF** [?] dédié aux ressources concernant les cancers. Le CISMeF a également réalisé, entre autres, le **site des recommandations du VIDAL** [?], qu'il héberge et maintient sur ses serveurs.

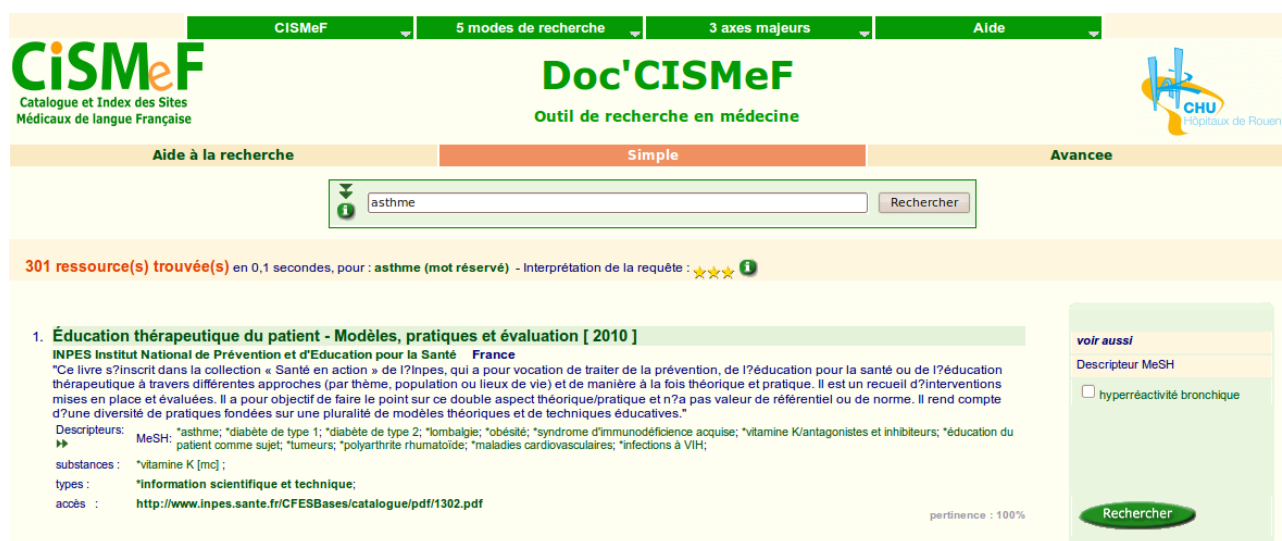


FIGURE 2.2 – Capture d'écran du moteur de recherche Doc'CISMeF

Chapitre 3

Présentation du sujet de stage

Le CISMeF met à la disposition du public des outils de recherche de ressources médicales. Ces ressources sont indexées manuellement par des documentalistes du domaine de la santé travaillant au CISMeF.

Afin d'indexer ces ressources, les documentalistes utilisent une Plateforme d'Indexation qui a été réalisée par la campagne PIC 2006. Cette plateforme dépend d'une architecture de base de données que le personnel technique du CISMeF estime lourde à maintenir. En effet, une fiche descriptive d'une ressource internet est liée au minimum à 8 autres entités (parmi lesquelles un éditeur, un auteur, un descripteur, ...) et ces liaisons impliquent pour la plupart la création de nouvelles tables. Si nous souhaitions ajouter une nouvelle entité ou une nouvelle liaison, cela se traduirait par la modification du modèle physique de la base en premier lieu puis par la modification de la Plateforme d'Indexation en second lieu. L'équipe technique souhaite s'affranchir de ces modifications de structure par la mise en oeuvre d'une architecture dite "générique". Par ailleurs, le CISMeF souhaite rendre son Système d'Information pluridisciplinaire et multilingue.

Ressources Terminologies Sources Autres **Editeur : 32** Profil : Indexeur Déconnexion

Accueil > Liste des Editeurs

N° 32

Editeur

URL Editeur

Sous-Titre

GIP UMF ☐

Liste des Questionnaires pour cet éditeur

	N° Fiche	Titre
<input type="checkbox"/>	100	Prévention de la carie denta
<input type="checkbox"/>	191	Dépistage des infections à c
<input type="checkbox"/>	449	Dépistage et le diagnostic p
<input type="checkbox"/>	455	Santé des autochtones (La)
<input type="checkbox"/>	526	Dépistage de l'infection par
<input type="checkbox"/>	587	Dépistage des troubles visu

Annuler Appliquer les modifications

Plan du site - © 2006 CHU de Rouen - CISMeF

FIGURE 3.1 – Capture d'écran de la Plateforme d'Indexation du CISMeF avant migration

Dans cette optique de pluridisciplinarité, de multilinguisme et de généricité, le CISMeF a demandé à l'équipe du premier semestre du PIC CHU 2010 de concevoir un modèle de base de données générique et de migrer les données existantes sous cette nouvelle architecture. Cette nouvelle architecture à structure fixe permet d'insérer n'importe quel type d'entité et d'y associer n'importe quel type de données. Ce nouveau modèle est constitué de 8 entités qui permettent de stocker la structure des entités que l'on souhaite insérer (type des attributs et des relations, ordre d'affichage, ...) et de stocker les données sous forme d'attributs et de relations.

L'objectif de ce stage, effectué en équipe avec Pascal Mazon, était de procéder d'une part à

la migration de la Plateforme d'Indexation sur la nouvelle architecture de base de données et d'autre part à son amélioration . La plateforme, une fois migrée, devait intégrer l'ensemble des fonctionnalités précédentes et par ailleurs ajouter la gestion du multilinguisme. De plus, la gestion des ressources devait s'effectuer de manière générique. C'est à dire que l'ajout d'une nouvelle entité (une fiche concernant le droit par exemple) devait être géré automatiquement. Toutes les informations structurales sont définies par des entrées dans la base de données, par exemple le fait qu'un auteur ait un nom et un prénom et qu'il est lié à des ressources.

Chapitre 4

Travail effectué

Pour parvenir à réaliser les différents objectifs demandés dans le cahier des charges (cf Annexe ??), nous avons dû tout d'abord, Pascal et moi, nous familiariser avec l'environnement de travail et les technologies utilisées. Nous avons tous deux travaillé sur le système d'exploitation Windows XP avec l'IDE Eclipse et nous avons à notre disposition un gestionnaire de versions afin de pouvoir travailler en parallèle. L'application utilise la technologie J2EE avec notamment comme conteneur Web un serveur Apache Tomcat et comme conteneur d'applications un serveur JBoss. Le Système de Gestion de la Base de Données du CISMeF est Oracle XE.

Étant donné que nous devons adapter une application, la première étape de notre travail fut de comprendre son fonctionnement. Ce sera l'objet de la première partie de ce chapitre. La seconde partie sera consacrée à l'explication du nouveau modèle de base de données afin de comprendre les répercussions sur l'application. La troisième partie concerne notre démarche vis à vis des modifications qui ont été apportées à l'application.

4.1 Analyse de la Plateforme d'Indexation existante

La Plateforme d'Indexation est une application qui a été réalisée par la campagne PIC 2006. L'utilité première de cette plateforme est de permettre aux documentalistes de gérer les fiches descriptives de ressources Internet du Système d'Information. Les indexeurs peuvent donc en créer, leur assigner des métadonnées (titre, auteur, descripteurs . . .) ou les supprimer.

Mais cette plateforme est également utilisée pour maintenir l'ensemble des terminologies de descripteurs implantées dans le Système d'Information du CISMeF. Ce sont ensuite les descripteurs de ces différentes terminologies qui sont utilisés pour indexer les ressources.

Ces deux utilisations, qui paraissent différentes l'une de l'autre, sont en fait gérées de la même façon par la Plateforme d'Indexation. Cette application a en effet été conçue pour manier des types d'entités de manière générique. J'entends par type d'entité une fiche, un auteur, un descripteur. . . . La gestion générique intervient à tous les niveaux de l'application. Quelque soit l'action que l'on souhaite effectuer, par exemple un affichage, une modification, une création ou une suppression d'entité, l'application utilise le même processus pour chaque type d'entité. Ainsi, la modification d'un descripteur utilisera les mêmes mécanismes que la modification d'une fiche.

Cette généricité au niveau du traitement implique donc une certaine complexité qu'il nous a fallu appréhender et comprendre pour mener à bien notre projet. La première partie de notre travail fut donc d'effectuer une rétro-ingénierie sur cette application. En effet, bien que nous ayons eu à notre disposition la documentation du PIC 2006, celle-ci ne s'avérait pas assez détaillée.

Nous avons distingué 5 modules dans cette application. Il nous fallait donc savoir comment ces modules interagissaient entre eux afin de déterminer leur fonction au sein de l'application. Ces interactions sont résumées dans la figure ??.

En étudiant les fonctionnalités, nous avons ainsi pu retrouver une architecture Modèle-Vue-Contrôleur au sein de l'application.

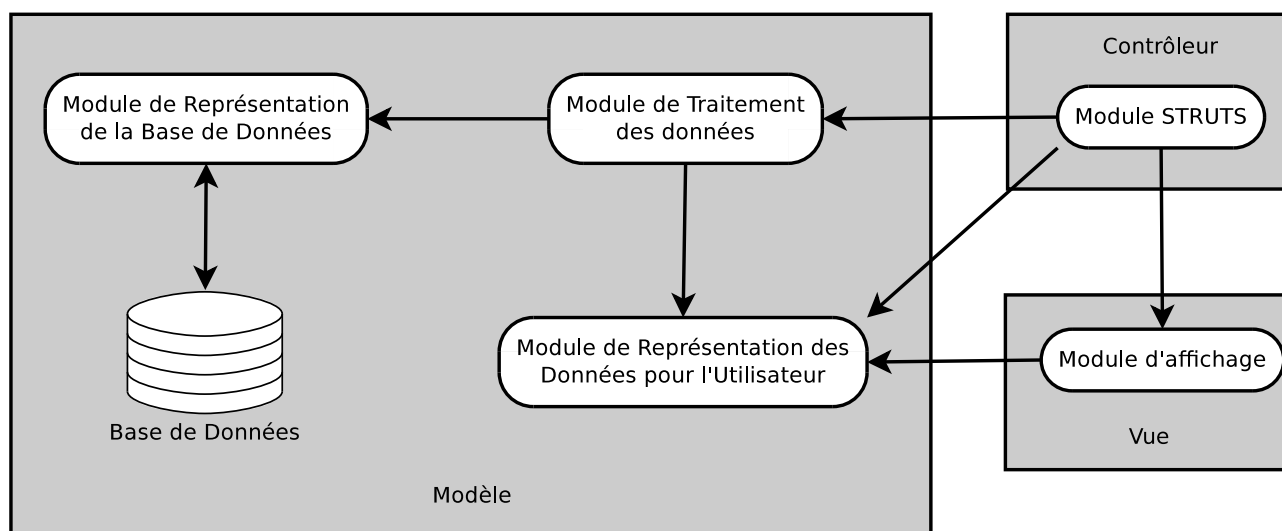


FIGURE 4.1 – Diagramme d'interaction entre les différents modules de l'application

4.1.1 La Vue

Le module dédié à la Vue contient toutes les pages JSP qui sont interprétées par le serveur Tomcat. Ce module d'affichage contient deux fonctionnalités principales :

- L'affichage en liste qui permet de réunir plusieurs résultats après une recherche par exemple,
- L'affichage en fiche qui permet d'obtenir toutes les informations d'une fiche descriptive de ressource, d'un auteur, d'un descripteur. . .

4.1.2 Le Contrôleur

Le module Struts est la partie Contrôleur. Il contient les méthodes qui sont appelées via le fichier de configuration de Struts. Il redirige vers la Vue ou fait appel au traitement des données de la partie Modèle. Étant donné que nous avons déjà vu ce framework en cours, la compréhension de ce module fut assez simple.

4.1.3 Le Modèle

La partie Modèle est quant à elle divisée en 3 modules. Cette partie a été la plus difficile à appréhender du fait de la généricité de l'application. Il faut savoir que l'application possède ici 2 types de représentation de la donnée : une représentation pour la base et une pour le reste de l'application.

L'ancien Modèle de base de données

L'ancien modèle de base de données est un modèle relationnel classique d'une base. Toutes les entités et associations utilisées par l'application (une fiche, un auteur, un descripteur, . . .) y ont été repartitionnées pour déterminer le modèle physique. De ceci a ensuite été tiré les tables représentant la donnée, TB_FICHES pour une fiche avec les champs titre, url, description, par exemple, et les tables permettant de lier une entité à une autre, TB_AUTEURS_FICHE pour lier un auteur à une fiche.

La représentation de la base de données

Le module qui correspond à la représentation de la base de données utilise la technologie EJB 2.0, en particulier les EJB Entité. Cette ancienne version des EJB s'avérait plus difficile à comprendre et à utiliser que la version 3.0 que nous avons vu en cours. Il y avait autant d'EJB que de tables utilisées : un pour la table TB_FICHE, un pour la table TB_DESCRIPTEURS etc. . .

Ce module est réutilisé par le module Traitement des Données pour répercuter sur la base les modifications faites par l'utilisateur. Cependant, ce ne sont pas les EJB qui sont utilisés par la Vue pour obtenir les données. Il existe en fait un autre module qui représente la donnée côté utilisateur.

La représentation des données côté utilisateur

La représentation des données côté utilisateur contient toute l'intelligence de l'application. Elle permet de créer les entités d'un formulaire sous forme d'objet ayant une structure générique.

Le diagramme de classe ?? est un diagramme simplifié des classes présentes dans ce module.

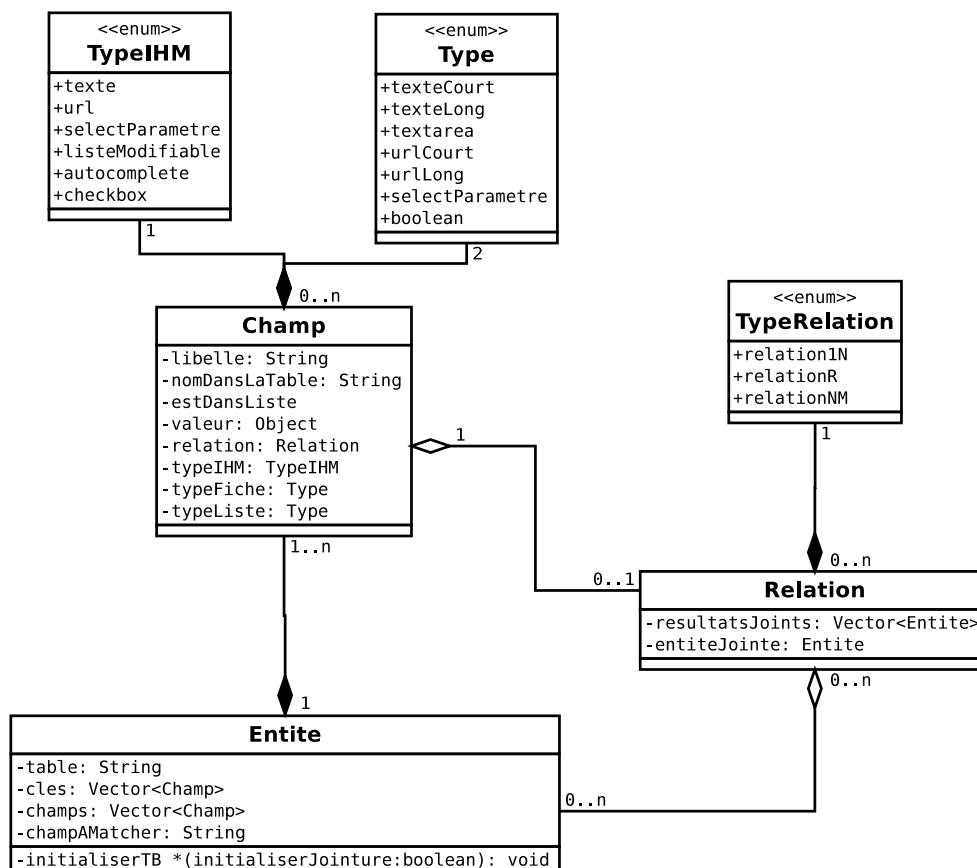


FIGURE 4.2 – Diagramme de classe du module de représentation des données côté utilisateur

Un objet de classe Entite est la représentation objet d'un formulaire pour une entité donnée. Pour chaque champ d'un formulaire, il y aura un objet Champ correspondant. Par exemple, si une fiche a un titre et une URL, l'attribut *champs* de l'objet Entite contiendra 2 objets Champ. Ces deux objets Champ auront alors comme libellés respectifs "Titre" et "URL". L'attribut *cles* permet d'identifier de manière unique une entité. Cet attribut est en fait un vecteur constitué d'objets Champ. Un objet Entite possède aussi le nom de la table à laquelle il correspond. C'est à dire que pour avoir un objet Entite relatif à la table "TB_FICHES", il faut insérer dans l'attribut *table* la valeur "TB_FICHES".

Pour construire le squelette d'une entité avec l'ensemble de ses champs et relations, la classe Entite est munie d'autant de méthodes privées "initialiserTB_*" qu'il y a de tables existantes

déclarées en EJB.

clé ayant pour libellé "N°"
et pour valeur "100"

champ ayant pour libellé "Titre"
et pour valeur "prévention de la carie dentaire"

N°	100		
Titre	Prévention de la carie dentaire		
Sous-Titre			
URL(s)	Ordre	URL	Type accès
	1	http://www.ctfphc.org/French_Text/DENTAL.htm	

champ ayant pour libellé URL(s)
et contenant une Relation avec l'entité URL

FIGURE 4.3 – Décomposition de l'interface en clés et champs

La classe Champ permet de représenter les champs d'un formulaire. Pour être plus explicite, je vais me baser sur la figure ?? qui représente une partie de l'interface d'édition d'une ressource Internet. Un objet de classe Champ est composé d'un attribut *libelle* qui correspond au libellé affiché sur l'interface, d'un attribut *nomDansLaTable* qui permet de faire la correspondance entre ce champ de l'objet Entite et la base de données ainsi que d'un attribut *valeur* ou d'un attribut *relation*. L'attribut *valeur* est renseigné lorsque le champ est un type simple (titre, sous-titre dans ce cas ci) tandis que l'attribut *relation* est renseigné lorsque le champ concerné est en relation avec une autre entité (ici, il y a une relation avec des objets Entite de type URL). L'attribut *typeIHM* est utilisé par le module d'affichage pour déterminer le bon type d'affichage correspondant à ce champ dans la partie Vue. Les attributs *typeFiche* et *typeListe* sont quant à eux utilisés pour affiner l'affichage selon l'interface sur laquelle l'utilisateur se trouve. Ainsi, le titre pourra être affiché avec un *texteLong* dans l'interface Fiche et avec un *texteCourt* dans l'interface Liste.

La classe Relation permet de lister l'ensemble des entités en relation avec l'entité mère. Elle dispose d'un attribut *entiteJointe* qui contiendra le squelette d'un objet Entite du type de l'entite jointe. L'attribut *resultatsJointes* est utilisé pour contenir toutes les entités existantes en base qui sont liées à l'entité mère.

Le Traitement des Données

Ce module est utilisé pour effectuer l'ensemble des interactions entre les EJB et les objets Entite de l'application. Il permet notamment de remplir les entités avec les données issues de la base, de modifier les données en base grâce aux données issues du formulaire ou encore de supprimer des entités de la base.

La correspondance entre les EJB et les objets Entite est illustrée par la figure ??.

L'attribut *table* de l'objet Entite permet de déterminer l'EJB qui sera utilisé. L'attribut *nomDansLaTable* de chaque clé ou champ de cet objet Entite permet de faire la correspondance vis à vis du champ de l'instance de l'EJB adéquate. Les valeurs de chaque clé ou champ de l'objet Entite sont ensuite copiées dans l'instance de l'EJB aux champs correspondants.

4.2 Le nouveau modèle générique de base de données

Le nouveau modèle de base de données a été conçu lors de la campagne PIC 2010. Il se veut générique, multilingue et multidisciplinaire. Je vais me baser dans cette partie sur le schéma ??, qui est un schéma simplifié, pour expliquer les principes de ce modèle générique. En effet, il est important de comprendre ce modèle pour comprendre les modifications qui ont été faites.

./img/interactionEJBModele.pdf

FIGURE 4.4 – Correspondance entre les EJB et les objets de classe Entité

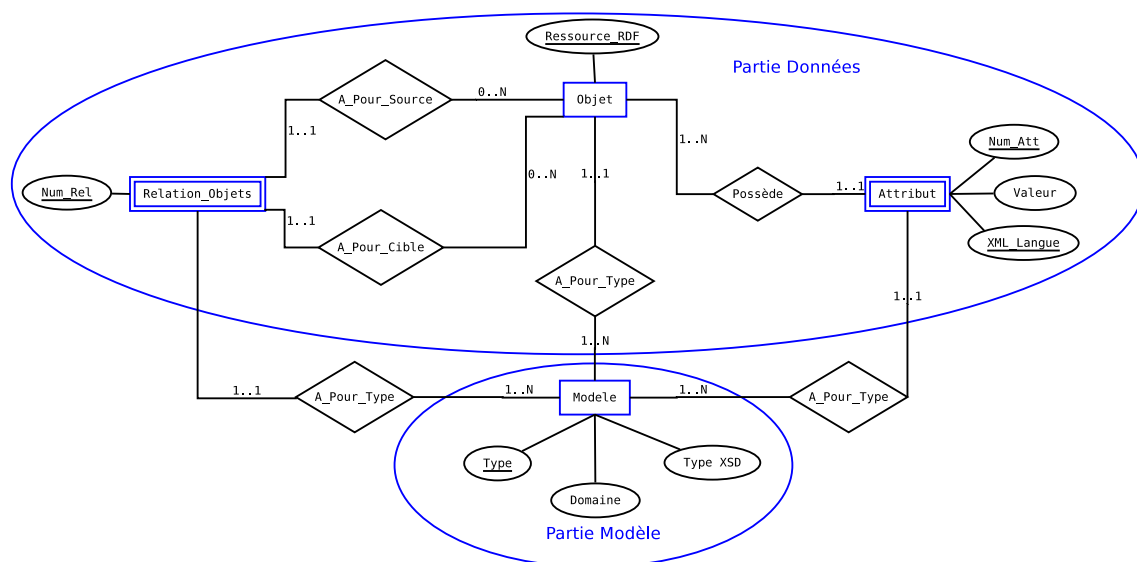


FIGURE 4.5 – Nouveau Modèle de Base de Données simplifié

Ce modèle générique peut se décomposer en deux parties : la partie Modèle et la partie Données.

4.2.1 La partie Modèle

Cette partie permet de déclarer tous les types qui seront réutilisés par la partie Données. Les types déclarés ne sont pas des types de données tels que les chaînes de caractère ou les entiers mais les types sémantiques. Par exemple, s'il faut insérer dans la partie Données une fiche, il y sera déclaré un type Fiche dans la partie Modèle. De même, pour insérer une métadonnée de type Titre dans la parties Données, un type Titre sera déclaré dans la partie Modèle.

Aucun objet, attribut ou relation ne pourra être utilisé tant que son type n'est déclaré dans la partie Modèle. Ceci permet de garder une intégrité au niveau de la partie Données pour ne pas

insérer n'importe quoi.

4.2.2 La partie Données

La partie Données s'inspire de notions de la Programmation Orientée Objet (POO) et du web sémantique, en particulier la notion de triplet RDF.

Elle est décomposée en 3 parties. La notion de POO se retrouve dans les entités Objet et Attribut. Pour insérer une nouvelle entité dans la structure générique, il faudra, comme il l'a été dit précédemment, déclarer les types sémantiques utilisés puis exploser l'entité en attributs. Par exemple, pour insérer une fiche qui contient un titre et une url, il y aura 2 attributs (titre et fiche) insérés dans l'entité Attribut. Pour ne pas perdre le lien entre la fiche et ses 2 attributs, il sera déclaré dans l'entité Objet une entrée avec un identifiant RDF_RESSOURCE. Les 2 attributs auront donc pour Ressource_RDF celle qui aura été déclarée dans l'entité Objet.

La notion de triplet RDF est retrouvée lorsqu'il faut lier 2 entités. En effet, un triplet RDF permet de lier 2 entités grâce à l'association {sujet, prédicat, objet}. Le sujet est la ressource à décrire (la liaison "A pour Source"), le prédicat est le type de la relation (la liaison "A pour Type") et enfin l'objet est l'autre ressource (la liaison "A pour Cible"). Ainsi, pour lier une fiche à son auteur, on aura l'association {identifiant_fiche, type_aPourAuteur, identifiant_auteur}.

4.3 Les modifications apportées à l'application

Une fois le fonctionnement de l'application assimilé, nous avons commencé à adapter les modules les plus bas-niveaux. Nous avons donc modifié en premier lieu le module de représentation de la base de données puis nous avons adapté le module de représentation des données côté utilisateur. Ensuite, nous avons modifié le module de traitement des données et nous avons terminé par la modification du contrôleur et de l'affichage en parallèle pour vérifier le bon fonctionnement de l'application.

4.3.1 Adaptation du module de représentation de la base de données

Huit classes EJB ont été développées pour notre nouveau modèle. Dans la partie ??, il n'a été donné que 4 entités du nouveau modèle de base de données pour simplifier la compréhension mais le modèle en compte en fait 8 au total. La difficulté principale de cette première adaptation provenait de la technologie utilisée : les EJB 2.0. Nous avons utilisé, tout comme le PIC 2006, le moteur XDoclet qui nous permettait de générer l'ensemble des fichiers nécessaires aux EJB Entité à partir d'un fichier source annoté.

4.3.2 Adaptation du module de représentation de la données côté utilisateur

Suite à cette première brique, nous avons adapté le module de représentation des données côté utilisateur. Cette partie s'avérait la plus complexe à réaliser étant donné que la structure de nos entités a été explosée en attributs et en relations.

Il s'avère que le modèle générique de la base dispose de 2 entités dans la partie Modèle qui permettent de stocker les informations concernant la structure des entités. Étant donné que ces informations n'étaient pas présentes au départ, nous avons donc dû créer un panel de procédures stockées pour les générer. Au final, nous pouvions par exemple savoir qu'une fiche a un titre, une url et une relation avec un auteur. L'ordre des champs était également stocké.

À partir de ces informations, nous avons créé un objet de classe Entité ayant le même squelette que l'application originelle. Nous n'avons pas effectué une refonte totale de l'application

car la brique d'affichage pouvait être quasi-totalement réutilisée moyennant un delta de modification. À ce stade, nous pouvions donc créer une fiche descriptive de ressource ayant la même structure que l'ancienne fiche.

Le cahier des charges nous demandait également d'implanter une composante multilingue à l'application. Or, il s'avère que ce module permet de générer les entités mais aussi toutes les informations des champs des entités. Nous avons donc ajouté une information à la classe Champ pour spécifier si un attribut pouvait être multivalué ou non, et le cas échéant s'il acceptait le multilinguisme.

Par ailleurs, la base de données contient l'ensemble des libellés des champs selon différentes langues. Notre adaptation permet selon une langue d'interface demandée par la Vue de changer la langue des libellés facilement. De même, il suffit de changer un attribut du modèle en base pour rendre des champs visibles ou non, multivalués ou non ou encore affichés dans une liste. Ces demandes de fonctionnalités faites par notre tuteur permet à l'équipe technique du CISMeF de ne plus avoir à modifier le code source de l'application. Tout est géré via la base de données.

Ce module est celui sur lequel l'application se repose pour générer les squelettes d'objet Entite qui sont réutilisés par l'ensemble des autres modules. Nous avons donc dû résoudre un problème de performance due à la sollicitation continue de ce module. En effet, il utilise des données contenues dans la base de données. Nous avons donc mis en place un cache qui stocke les squelettes des objets Entite. Ainsi, une fois qu'un squelette est chargé une première fois, si il est demandé un objet de même type, c'est le cache qui est sollicité et non la base. Ce système a réussi à réduire significativement le temps de réponse de l'application.

4.3.3 Adaptation du Module de Traitement des Données

Le module de Traitement des Données est celui qui a subi le plus de modifications. En effet, étant donné la grande différence entre l'ancien et le nouveau modèle de base de données, le remplissage, la modification ou encore la suppression des données ne pouvaient pas s'effectuer de la même manière.

Nous avons dû concevoir un système qui nous permettait la réutilisation des briques d'affichage via les objets Entite. Nous y sommes parvenus en ajoutant un niveau dans les objets Entité initiaux.

Ceci peut se schématiser par la figure ??.

Ajouter un palier lors du remplissage des données donnait deux avantages. Tout d'abord, la modification du module d'affichage se trouvait très réduite. En effet, pour afficher le titre par exemple, nous n'affichions plus la valeur du champ titre elle-même mais l'objet Entite Attribut contenu dans le champ titre.

Le second avantage réside dans le fait que les identifiants des entités Attribut sont conservés. Nous pouvons donc récupérer leurs instances EJB comme cela était fait dans l'ancienne version de l'application et ainsi modifier les valeurs en base.

4.3.4 Adaptation du Contrôleur

La partie contrôleur n'a pas subi de modification majeure. Nous avons dû bien entendu l'adapter aux nouveaux squelette des objets Entite et effectuer quelques modifications pour la recherche d'éléments.

4.3.5 Adaptation de la Vue

L'affichage devait rester quasi identique par rapport à l'application précédente. Nous avons donc modifié que très peu les feuilles de style. Cependant, nous avons dû ajouter la gestion des



FIGURE 4.6 – Modification au niveau du remplissage des données

champs multilingues et adapter la récupération des données qu'il faut afficher à notre nouvelle construction d'objets Entite.

La figure ?? est un aperçu de la plateforme avant adaptation et la figure ?? est un aperçu après adaptation.

Certains champs ont été ajoutés dans l'interface notamment le champ de relation entre une institution et un auteur. Ils peuvent être facilement retirables en effectuant une modification dans un attribut de la partie Modèle de la base de données. Ici, nous donnons la possibilité d'assigner à cette institution un auteur d'une fiche.

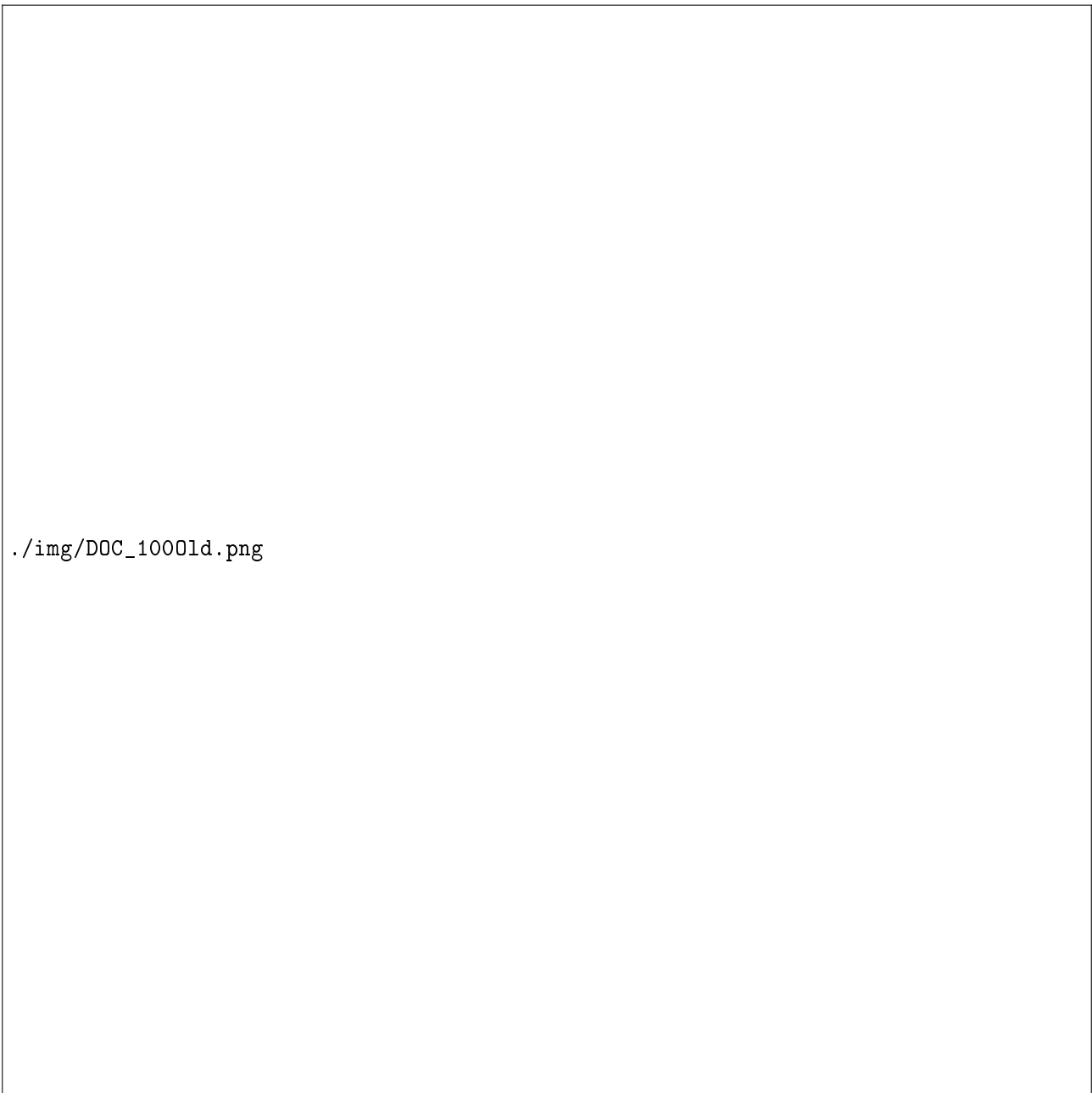


FIGURE 4.7 – Plateforme d'Indexation avant modification

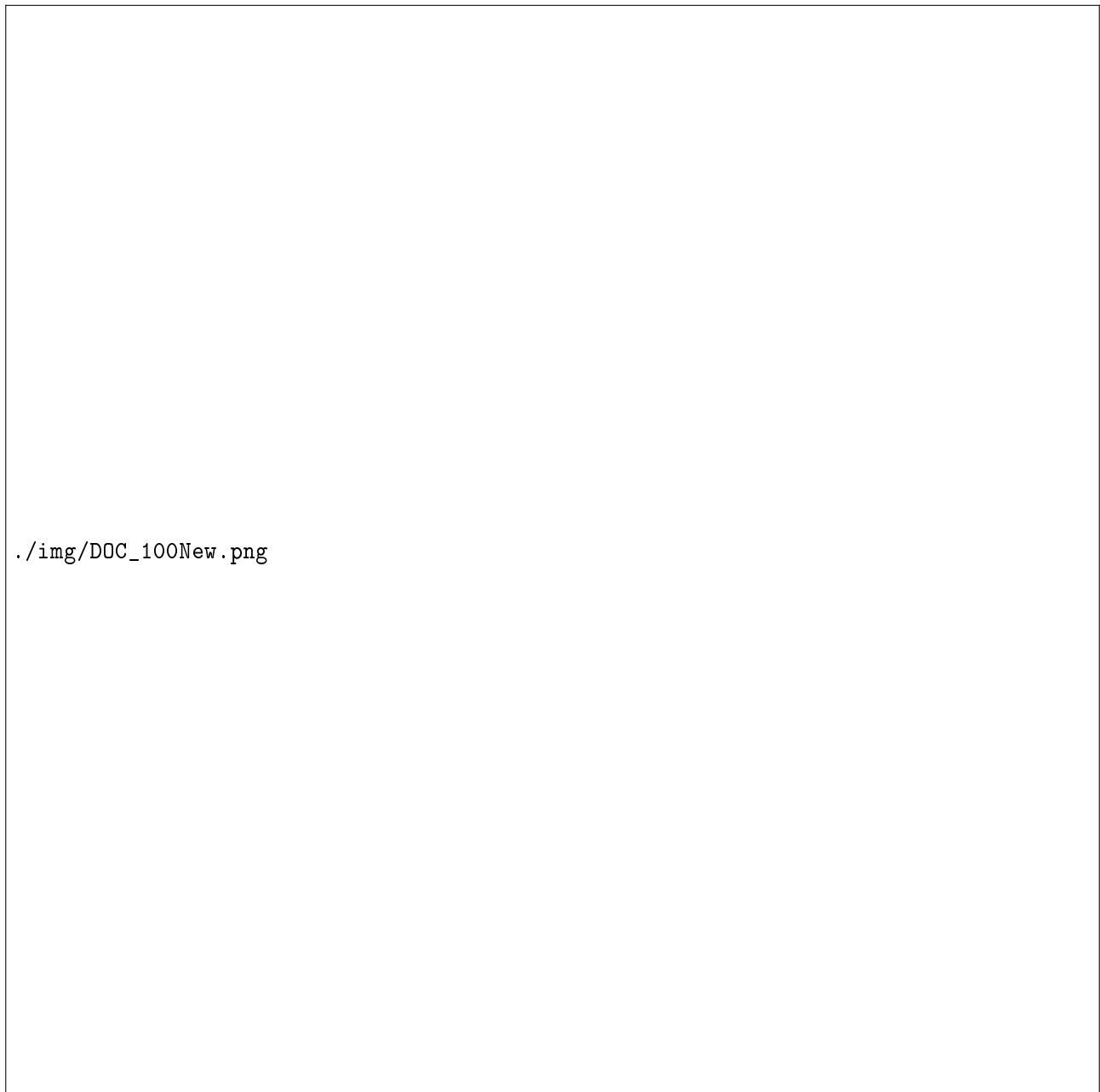


FIGURE 4.8 – Plateforme d'Indexation après modification

Chapitre 5

Conclusion

La conclusion sera scindée en trois parties. Dans un premier temps, je présenterai ce qui a été réalisé vis à vis du cahier des charges. Dans un second temps, je donnerai ce qu'il reste à faire et les évolutions possibles et enfin je tirerai un bilan personnel vis à vis de ce stage.

5.1 Les réalisations

Nous avons un cahier des charges détaillé des fonctionnalités à implanter (cf Annexe ??). Le cahier des charges était divisé en 3 parties et ce stage nous a permis de réaliser entièrement la partie A, qui concernait la migration de la plateforme sur la nouvelle base de données. L'application est entièrement fonctionnelle et est maintenant en test pour être mise en production en Janvier 2011. Les performances, malgré le modèle, n'ont pas été réellement altérées et les temps de réponse s'avèrent équivalents à ceux de l'ancienne application.

Nous avons également réalisé certaines améliorations de la partie B, notamment la gestion de l'interface multilingue dynamiquement et l'ajout de liens pointant vers d'autres outils du CISMeF.

Par ailleurs, nous avons maintenu la documentation et l'avons adaptée aux changements que nous avons apportés. Cependant, nous avons manqué de temps pour la rendre plus complète.

5.2 Les améliorations à effectuer

La plupart des améliorations à effectuer sont situées dans la partie B de l'Annexe ?? . Nous n'avons ainsi pas eu le temps de générer le menu de l'application via la base de données ou encore migrer les fonctionnalités d'un client Virtual Basic sur la plateforme. Certains affichages de l'interface restent également à améliorer.

5.3 Bilan personnel

Sur le plan technique, ce stage m'a permis de me rendre compte de la difficulté de reprise d'une application lorsque peu de documentation est disponible. Effectuer une rétro-ingénierie complète s'avère être néanmoins une très bonne expérience pour comprendre les mécanismes d'une application malgré son coût énorme en temps. Je me suis également rendu compte que les Technologies Web sont certes intéressantes mais assez répétitives. Aussi je souhaiterais ne pas continuer à travailler dans ce secteur.

Sur le plan personnel, bien que je travaille depuis quelques années assez fréquemment avec Pascal, ce stage m'a permis encore une fois d'expérimenter la dynamique et le travail en équipe. Ceci est un réel plus et je ne pense pas que ce stage aurait été aussi bien réussi sans cela.

Chapitre 6

Bibliographie

Bibliographie

- [1] Site Internet du CISMeF, <http://www.chu-rouen.fr/cismef/cismef.html>
- [2] Composition de l'équipe du CISMeF, <http://www.chu-rouen.fr/cismef/equipecismef.html>
- [3] Financeurs publics et Partenaires du CISMeF, <http://www.chu-rouen.fr/cismef/partenairecismef.html>
- [4] Portail du CISMeF, <http://www.chu-rouen.fr/cismef/>
- [5] Moteur de recherche Doc'CISMeF, <http://doccismef.chu-rouen.fr/servlets/Simple>
- [6] Moteur de recherche KISMeF (ressources concernant les cancers), <http://doccismef.chu-rouen.fr/servlets/KISMeF>
- [7] Site des Recommandations Vidales, <http://www.vidalrecos.fr>

Annexe A

Cahier des charges du stage

Résumé : Ce stage a été réalisé par Hervé Ly-Sunnaram et Pascal Mazon dans le laboratoire du CISMeF situé au CHU de Rouen entre le 28 juin et le 22 Août 2010. Le CISMeF a pour but de fournir au personnel médical des outils permettant de retrouver efficacement des ressources internet dédiées au domaine médical. Pour indexer l'ensemble de ces ressources, le personnel du CISMeF utilise une Plateforme d'Indexation qui a été réalisée par la campagne PIC 2006. Elle permet d'indexer les ressources grâce à des descripteurs issus de terminologies médicales telle que celle du MeSH.

Ce stage avait pour sujet la migration de la Plateforme d'Indexation de ressources vers un nouveau modèle de base de données. La particularité de cette migration est l'utilisation d'un modèle dit générique, multilingue et multidisciplinaire qui a été conçu lors de la campagne PIC 2010. Au terme de cette migration, la Plateforme devait devenir multilingue et gérer l'ensemble des entités contenues dans la base de manière générique.

Abstract : This internship was carried out by Hervé Ly-Sunnaram and Pascal Mazon. It took place at the research lab of the CISMeF (french acronym for Catalog and Index of French Language Health Resources on the Internet) from the 28th June to the 22th August 2010. The objective of the CISMeF is to provide tools for health professionals that will help them to find internet ressources efficiently. The staff of the CISMeF uses an Indexation Tool which was realized by the PIC 2006. This Tool indexes Internet Resources thanks to descriptors coming from medical terminologies such as the MeSH terminology.

The aim of this internship was to migrate the Indexation Tool to a new database model. The special feature of this database model is the fact it is generic, multidisciplinary and that it accepts any languages. This database model was realized by the PIC 2010. At the end of this internship, the Indexation Tool was accepting whatever languages and managing datas generically.

couvertureBis2.pdf



INSA de Rouen
Avenue de l'Université - BP 08
76801 Saint-Etienne-du-Rouvray
Tél : 02 32 95 97 79
<http://asi.insa-rouen.fr>