Rapport de stage - septembre 2002

DESS Ingénierie documentaire

Réorganisation du système de gestion de la documentation au Centre de Ressources Informatiques de Haute-Savoie

BIZET Delphine

Sous la direction de Jean-Claude Fernandez
Directeur scientifique du Centre de Ressources Informatiques de Haute-Savoie

enssib
école nationale supérieure des sciences de l'information et des bibliothèques

Réorganisation du système de gestion de la documentation au Centre de Ressources Informatiques de Haute-Savoie

Résumé :

Le développement du centre de ressources informatiques de Haute-Savoie nécessite une réorganisation de son système de gestion de documentation dans le but de l'améliorer et de le rendre plus homogène.

En premier lieu, une analyse de l'existant a permis d'établir un bilan général.

Dans un second temps, des améliorations ont pu être proposées ; une solution a été adoptée et la mise en œuvre de l'outil a pû commencer.

Descripteurs :

Gestion électronique document, GED, GEIDE, Système gestion électronique document, Document électronique, Langage XML

Abstract :

The developpement of the Haute-Savoie Data Processing Resources Centre needs a reorganization of its electronic data management in order to improve it and to make it more homogeneous.

At first, a study of the existing tool has made it possible to draw up a general idea.

Then, improvements could have been suggested, a solution was adopted and the operation of the tool has begun.

Keywords :

Electronic document management, EDM, Electronic document, XML language, Extensible Markup Language

Remerciements

Je tiens à remercier toute l'équipe du Centre de Ressources Informatiques de Haute-Savoie pour sa collaboration et son aide.

Sommaire

Introduction

Ce stage a été réalisé à Archamps, au Centre de Ressources Informatiques (CRI), durant une période de 4 mois, commençant le 3 juin et finissant le 27 septembre 2002.

Ayant effectué un stage volontaire au CRI après une maîtrise d'informatique, j'ai été informée par Jean-Claude Fernandez, directeur scientifique du CRI, des besoins du CRI en matière de documentation; il a donc été convenu dès l'été 2001 d'effectuer le stage sanctionnant la fin du DESSID au CRI; par la suite, j'ai rencontré Jean-Claude Fernandez à deux reprises en décembre 2001 et mars 2002 pour avoir des précisions sur ce stage et la mission proposée.

L'objectif général de faire évoluer le système actuel de gestion de documentation pour l'adapter aux nouvelles technologies et aux nouveaux besoins.

Dans cette perspective, j'ai été intégrée au projet nommé *cridoc*. Le groupe de projet *cridoc* est constitué de deux personnes en plus de moi-même : Sébastien Delcroix, responsable des développements et de la documentation et Grégory Duchatelet, développeur.

Dans un premier temps, il s'agissait d'apporter des propositions d'améliorations concernant certains aspects spécifiques. Ces différentes études devaient aboutir à une réflexion sur le système de gestion de la documentation dans sa globalité et à des propositions de solutions pour améliorer et homogénéiser le système.

Suite à ces propositions, une réflexion commune a abouti au choix d'une solution. Par la suite, j'ai participé à la programmation de cette solution.

Dans une première partie nous évoquerons le Centre de Ressources Informatiques (CRI) de Haute-Savoie et verrons que la gestion de la documentation n'est qu'une activité annexe.

Nous aborderons ensuite l'analyse de l'existant. Nous verrons les différentes études que j'ai réalisées sur des points précis à améliorer. Puis, nous verrons les propositions faites pour améliorer le système dans sa globalité.

Nous conclurons en évoquant la mise en œuvre de la solution adoptée et de quelle manière j'y ai participé.

Le Centre de Ressources Informatiques de Haute-Savoie

1. Présentation du Centre de Ressources Informatiques

Le Centre de Ressources Informatiques (CRI) [1] a été créé à l'initiative du Conseil Général de Haute-Savoie [2] et de l'Agence Economique Départementale [3]. C'est une structure unique en son genre sur le territoire français.

Le CRI poursuit une politique de déploiement des technologies de l'information en Haute-Savoie en direction des catégories d'utilisateurs relevant du service public (le Conseil Général, les communes, groupements ou syndicats de communes, les villes, les établissements scolaires et organismes liés à l'éducation, les hôpitaux, les organismes liés à la recherche, le tourisme). L'ensemble constitue un véritable réseau des services publics parmi les plus développés au niveau national.

Au 15 mai 2002, on comptait 697 établissements scolaires connectés, 153 organismes publics et 850 institutions.

Le CRI est situé à Archamps, près de Genève. Cette structure se compose de douze personnes : un directeur scientifique, un responsable développements et documentation, deux développeurs, un responsable technique réseaux, un technicien réseaux, un technicien haut-débit, une « webmaster », un responsable technique systèmes, un technicien systèmes, deux cyber secrétaires.

2. Les missions du Centre de Ressources Informatiques

Les missions du Centre de Ressources Informatiques sont de plusieurs ordres :

2.1. *Information et Sensibilisation*

Le CRI organise diverses manifestations (séminaires, universités d'été, expositions, démonstrations de solutions et produits, etc.), destinées à sensibiliser l'ensemble des acteurs économiques de la Haute-Savoie sur l'intérêt qui existe pour un organisme, public ou privé, à s'approprier les technologies d'information et de communication.

Le CRI participe à des colloques et salons au niveau national et international. Durant l'été 2002, trois représentants du CRI se sont rendus aux Rencontres Mondiales du Logiciel Libre (RMLL) qui se sont déroulées à Bordeaux.

2.2. *Formation*

Les formations organisées par le CRI visent en général à doter les organismes de compétences qui leur permettent à leur tour de dispenser les mêmes formations en interne. Pour des organismes plus petits, des formations sont assurées directement à destination des utilisateurs finaux.

2.3. *Aide aux utilisateurs*

Une aide est fournie aux organismes de telle façon qu'ils puissent assurer eux-mêmes une "hot line" de premier niveau permettant de répondre aux appels à dépannage lancés par leurs membres. Si un dépannage n'est pas résolu à ce niveau, le CRI se tient à la disposition des organismes pour prendre le relais du dépannage.

Une aide sous forme de conseils sur les équipements et les solutions à mettre en place est également disponible auprès du CRI.

2.4. *Etude des besoins*

Des comités techniques dans lesquels sont représentés les groupes d'utilisateurs et le CRI, permettent d'identifier les besoins des utilisateurs, de décider de solutions répondant à ces besoins et d'organiser la diffusion des solutions et des savoir-faire.

2.5. *Déploiement de solutions*

Le CRI prend en charge une exploitation toujours plus importante avec le nombre croissant de ses utilisateurs, en assurant la disponibilité de services toujours plus nombreux, le plus souvent 24h/24 pendant toute l'année.

2.6. *Recherche et Développement*

Le CRI assure un travail permanent et important de veille technologique, à partir duquel il fait des choix et construit les solutions à venir en intégrant le plus possible des produits existants sur le marché, et en les adaptant à ses utilisateurs.

2.7. *Services proposés par le CRI*

Parmi les services liés à Internet, on retrouve la majorité des services proposés par la plupart des fournisseurs d'accès "commerciaux" (accès au réseau Internet, accès à un compte email, utilisation de listes de diffusion, accès au services de News, hébergement de sites web, protection des intrusions en provenance du réseau Internet par un filtrage du trafic entrant et sortant du réseau géré par le CRI).

De plus, le CRI propose des services spécifiques, tels que la consultation des mails à travers une interface web, le filtrage des sites web, la réservation et l'hébergement de noms de domaine.

Analyse de l'existant

Comme nous venons de l'exposer, le Centre de Ressources Informatiques de la Haute-Savoie est un organisme proposant de nombreux services, et qui par conséquent doit gérer une quantité importante de documents. Paradoxalement, la gestion de la documentation est une activité annexe : elle occupe une personne à mi-temps ; cependant, elle apparaît comme un maillon essentiel au bon fonctionnement de l'ensemble de l'organisme.

1. Les différents types de documents utilisés au CRI

Pour réorganiser le système documentaire, j'ai commencé par établir une liste exhaustive des différents types de documents utilisés au CRI. Notons que la quasi-totalité de ces documents existe sur support électronique. Pour établir cette liste, j'ai été aidé de Sébastien Delcroix qui a développé les différentes DTD XML [12] relatives aux documents produits en interne. Le CRI utilise neuf types de documents internes :

- compte-rendu de réunion
- mini-howto
- rapport
- spécification technique
- spécification fonctionnelle
- spécification générale
- documentation utilisateur
- documentation technique
- procédure

Chacun de ces types de documents possède une DTD spécifique.
En outre, il sera nécessaire de prendre en compte des documents ne possédant pas de DTD spécifique tels que les documents administratifs, les cours en ligne et les revues de presse.

Les documents administratifs tels que les plannings, les commandes sont actuellement gérés par les deux secrétaires.
Les cours en ligne sont ceux dispensés par les membres du CRI à ses utilisateurs ; ils sont gérés par une même personne et sont accessibles par identification sur un site Web spécifique.
Il en est de même pour le périodique électronique maintenu par le CRI ; le site web est en revanche accessible à tous à l'adresse http://reseaux74.cri74.org
On évalue ce corpus à deux cent documents environ.

Concernant la documentation externe, plusieurs types de documents sont à prendre en considération :

- site Web
- livre
- périodique électronique ou E-zine
- document électronique :
  - documentation de développement
  - source de développement
  - document téléchargé
  - cours en ligne (autres que ceux dispensés par les membres du CRI)

La bibliothèque du CRI compte 150 ouvrages. Les livres sont considérés comme documentation externe du fait de leur mode de publication (chaque ouvrage est indexé grâce à un formulaire de publication de la même manière que tout document externe).

Cette distinction entre documentation interne et externe est importante car le système de documentation que j'ai eu à analyser traite différemment ces deux collections de documents.

2. La gestion de la documentation interne

La documentation interne est gérée par chaque auteur (chaque membre du CRI) sur son espace privatif. L'espace privatif est un espace réservé et donc inaccessible à toute autre personne.

Le CRI possède un intranet. Chaque rubrique de l'intranet est un lien hypertexte permettant d'obtenir le document. La plupart des documents accessibles via l'intranet est enregistrée sur l'espace privatif de l'auteur.

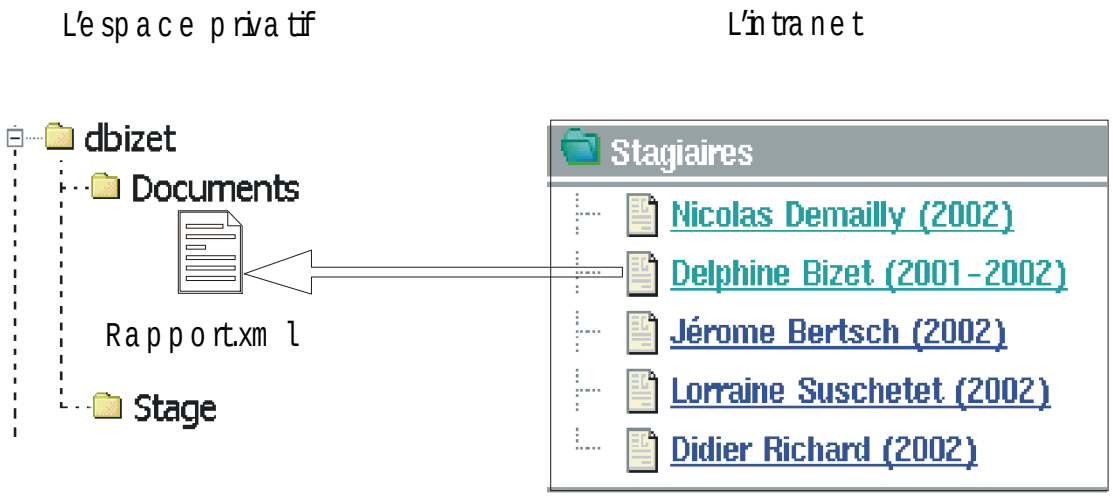


Schéma 1

Après connexion à l'intranet, si l'utilisateur clique sur le lien hypertexte « Delphine Bizet », il obtient un document nommé *rapport.xml* qui se trouve physiquement sous l'espace privatif de son auteur.

Un autre aspect à prendre en compte est qu'il n'existe aucun système de classement de ces documents. Comme les documents n'ont aucun lien entre eux, il est cohérent qu'il n'y ait pas de système de classification. En revanche, il faudra résoudre ce problème si l'on veut gérer la documentation interne comme tout autre document.

Concernant les formats de ces documents, la quasi-totalité d'entre eux est produite au format XML (eXtensible Markup Language).

3. La gestion de la documentation externe

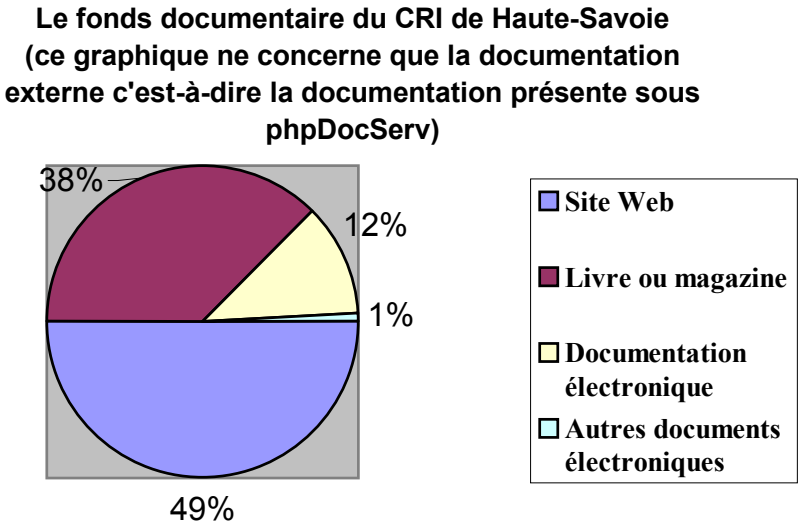
Actuellement, un système développé par Sébastien Delcroix et Jérôme Tamiotti en 2000 est disponible. Il permet de gérer un corpus de documents de supports divers. Ce système se nomme phpDocServ .

3.1. Présentation de phpDocServ

phpDocServ est un serveur de documents : le principe est de mettre en ligne la connaissance, aussi hétérogène soit-elle. Pour cela, les concepteurs ont déployé un intranet permettant de rassembler toute l'information voulue, de la standardiser et de la présenter de manière à en rendre l'accès le plus simple et le plus rapide possible.

Ce serveur peut être schématisé selon un modèle producteur / consommateur d'informations. Un des objectifs poursuivis par les concepteurs de l'outil était de pouvoir modéliser le plus grand nombre de supports d'information (du site web au CD-ROM, en passant par le livre, etc.), afin d'élargir au maximum le flux entrant d'informations. Pour les autres types de supports, on ne conserve que leurs caractéristiques à travers des notices ou fiches descriptives.

3.2. Le fonds documentaire



Plus de 60% des documents publiés sur phpDocServ sont des documents sur supports électroniques.
On évalue ce corpus à 400 documents.

3.3. Le mode de publication d'un document

Lorsqu'un membre du CRI veut diffuser un document, il doit s'identifier au niveau de l'intranet de phpDocServ, puis renseigner les différents champs du formulaire de publication. Ce formulaire, une fois validé, constituera la fiche descriptive du document ; c'est cette fiche qui sera enregistrée sur le serveur de documents. Il est aussi possible de télécharger le document sur le serveur ; dans ce cas, le document aura donc deux représentants sur le serveur : la fiche descriptive et le document lui-même.

La production d'informations se traduit donc par la publication d'une fiche descriptive et éventuellement, la publication du document source lui-même.

La personne qui publie la ressource devient propriétaire de cette ressource.

Comme tous les types de supports ne peuvent être décrits de manière unique, le formulaire possède un certain nombre de propriétés communes à

tous les supports ; la seconde partie du formulaire est spécifique à chaque support.

Voici comment se présente une fiche de publication d'un site web :

Après validation par l'administrateur, la notice est enregistrée dans la base de données. Cette base de données est constituée de plusieurs tables ; chaque table correspond à un type de support.

Par exemple, si l'on remplit le formulaire ci-dessus et que l'administrateur valide les données relatives au nouveau site web, les données seront enregistrées dans la table « bookmark ».

3.4. La modification ou la suppression d'un document

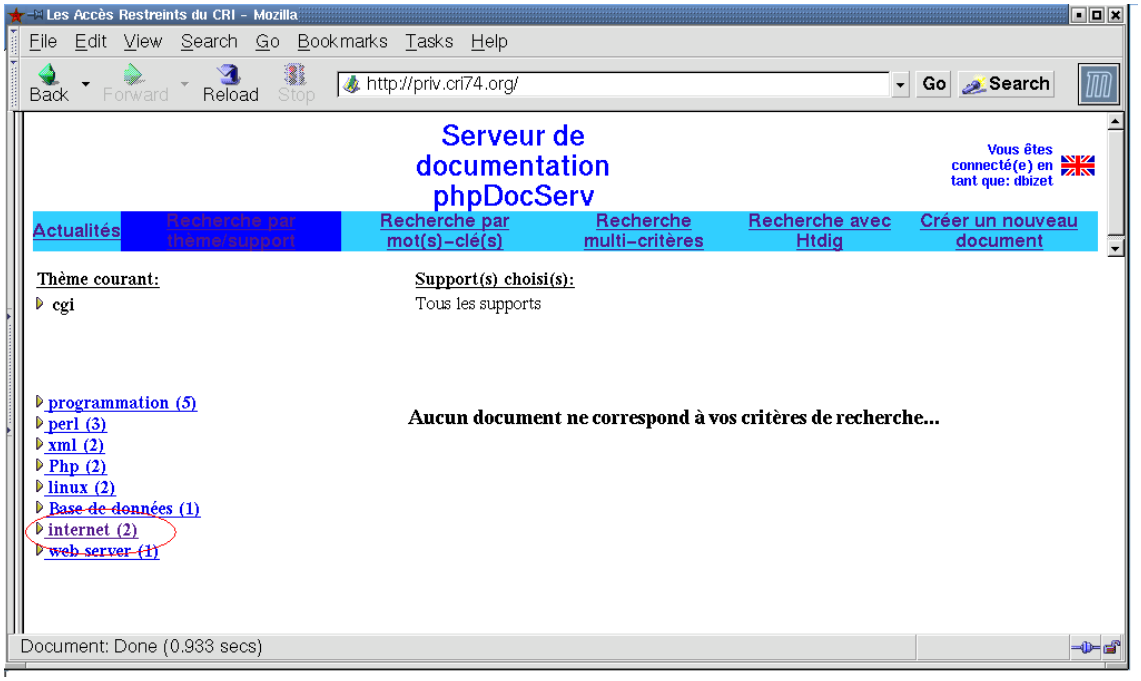
Toute modification ou suppression concernant les caractéristiques d'une ressource se fait via le même formulaire que la phase de publication.
Le propriétaire peut modifier les informations qu'il a entrées lors de l'indexation du document ; lorsqu'il valide le formulaire, les nouvelles valeurs des champs sont prises en compte.
Le propriétaire peut également choisir de supprimer le document.

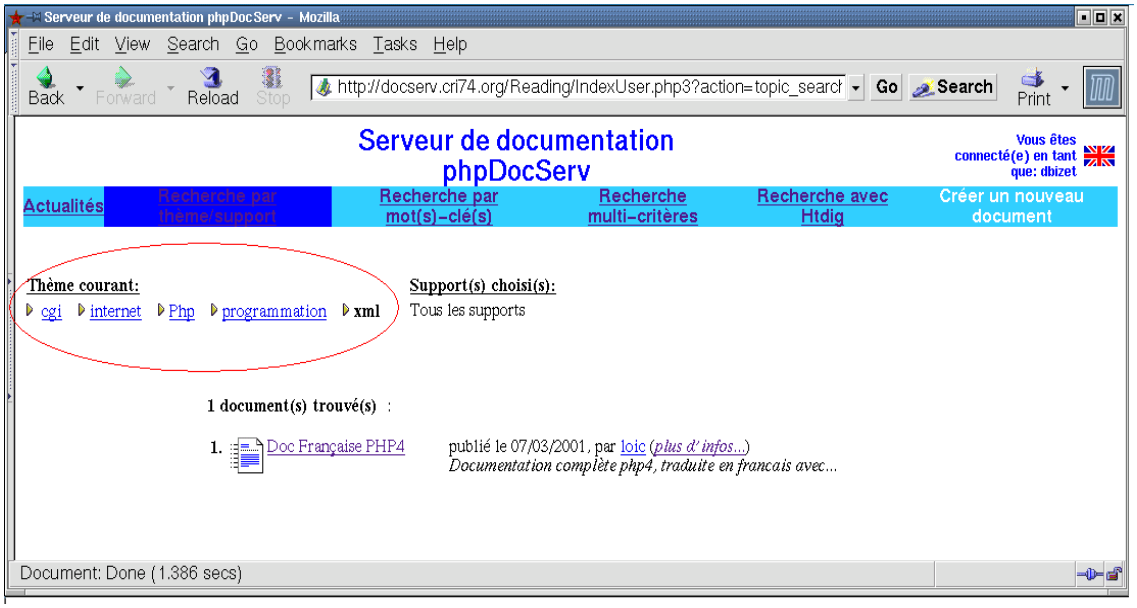
3.5. Recherche d'un document

La consommation d'informations (c'est-à-dire la consultation) se fait par différentes méthodes de recherche sur cette fiche descriptive. Il est possible de faire une recherche par mots-clés, une recherche thème/support ou une recherche multi-critères. Il est également possible de réaliser une recherche sur le texte intégral grâce au moteur de recherche htdig.
Quel que soit le type de recherche utilisé, celle-ci se fait grâce à un moteur de recherche interne développé au sein du CRI.

3.6. Le système de classification

Sous phpDocServ, chaque document est indexé grâce à un ou plusieurs thèmes : quand le propriétaire (celui qui crée la fiche descriptive du document) remplit le formulaire de publication, il doit associer des thèmes (au maximum cinq) permettant de décrire son document.
La liste de thèmes est gérée dynamiquement dans la mesure où il est possible d'entrer un nouveau thème dans la liste ; seule la validation par l'administrateur est requise.
L'association de thèmes lors de l'indexation d'un document engendre la création de liens entre chacun de ces thèmes. Les thèmes associés les uns aux autres constituent un ensemble de chemins ; il faudra passer par l'un de ces chemins pour accéder au document.





Cet exemple illustre la manière d'accéder à un document. L'utilisateur a cliqué sur le thème « cgi », puis, parmi la liste obtenu, il a choisi le thème « internet » et ainsi de suite…On voit qu'un seul document est indexé par les cinq thèmes « cgi », « internet », « php », « programmation » et « xml ». Ce document aurait pu être retrouvé en cliquant sur « internet » puis « programmation », « php », « xml » et « cgi ». Par conséquent, ce document est accessible par 5*4*3*2*1 soit 120 chemins possibles.

En revanche, le choix de seulement quatre de ces cinq thèmes n'aurait pas permis d'obtenir directement le document intitulé « Doc Française PHP4 ».

4. Bilan de l'analyse de l'existant

4.1. Le fonds documentaire

- Les documents concernent différents domaines. Le contenu de certains documents peut être très précis.
- Les documents se trouvent sur divers supports, en majorité électronique.
- Les documents internes sont au format XML mais les autres documents (notamment les documents téléchargés) ont des formats très divers.
- Le fonds documentaire contient 500 documents environ dont 400 sont des documents téléchargés ou des notices ; celles-ci peuvent contenir un lien hypertexte permettant d'accéder au document téléchargé sur le serveur.

4.2. Les outils

Deux outils bien distincts permettent la gestion du fonds documentaire :

- L'espace privatif permet à l'auteur de gérer sa propre documentation au sein de son arborescence de fichiers
- Le serveur de documents phpDocServ permet de gérer la documentation externe et les livres de la bibliothèque.

On peut comparer l'espace privatif à un coffre-fort : le propriétaire du coffre range ses documents comme bon lui semble à l'intérieur du coffre ; il est le seul à posséder la clé du coffre ; s'il ouvre le coffre avec cette clé, il pourra alors modifier le rangement des documents, en enlever…

PhpDocServ, quant à lui, peut être comparé à une étagère, sur laquelle chaque personne autorisée peut venir déposer ou consulter des ouvrages divers.

4.3. L'indexation

L'espace privatif ne contient aucun système de classification des documents. C'est la structure de l'intranet qui permet de faire le lien avec un document enregistré sous l'espace privatif de son auteur.

Sous phpDocServ, les documents sont indexés grâce à 5 thèmes au maximum. La classification est une classification « à plat » c'est-à-dire qu'il n'existe aucune règle régissant l'importance des thèmes. Lorsque l'indexation est effectuée, chaque thème est lié aux autres. Ce principe permet de distinguer des chemins différents pour accéder à chaque document ; il faut utiliser un de ces chemins pour retrouver le document ; en pratique, on crée artificiellement le chemin en cliquant successivement sur chaque thème : chaque fois que l'utilisateur clique sur un thème, il obtient la liste des thèmes qui lui sont rattachés, la liste des documents correspondants à cette indexation et le nombre de documents concernés.

Propositions d'améliorations

Pour pouvoir être retrouvé le plus rapidement possible, un document doit être enregistré de manière pertinente. Je me suis donc attachée à étudier le mode de classification des documents.
Lorsqu'une classification pertinente a été établie, il est nécessaire d'utiliser une base de données efficace pour stocker l'ensemble des documents ou des caractéristiques de ces documents, ceci pour permettre un accès rapide au document.
Enfin, dans le cycle de vie du document, une phase est l'exportation : les métadonnées permettent d'apporter des informations sur les données d'un document, et ainsi le rendre plus facilement exportable.

J'ai réalisé une série de propositions concernant certains points précis :

- Le plan de classification
- La base de données
- L'intégration des métadonnées
- La gestion de la documentation interne
- La gestion de la documentation externe

1. Le plan de classification

Une des questions principales est d'élaborer un système de classification plus efficace que celui de phpDocServ en intégrant la gestion des documents internes.
C'est pourquoi j'ai réfléchi aux différents moyens d'améliorer cette classification. J'ai tiré des conclusions de l'utilisation de phpDocServ et des remarques constructives des utilisateurs, ce qui m'a permis de proposer un nouveau mode de classification.

1.1. Utilité d'un thésaurus

En premier lieu, je me suis interrogée sur l'utilité d'un thésaurus. A la vue du nombre de documents et de la fréquence d'utilisation du serveur de documents, plusieurs points plaident en défaveur de l'utilisation d'un thésaurus : tout d'abord, vu la précision et la complexité des documents gérés, il aurait fallu créer ce thésaurus en interne ; ensuite, même en supposant pouvoir créer ce thésaurus, il aurait fallu le mettre à jour régulièrement ; de plus, le nombre de documents est relativement restreint ; en conséquence, l'utilisation d'un thésaurus aurait été lourde et n'aurait vraisemblablement eu qu'un faible rendement par rapport au travail à réaliser.

1.2. Système d'indexation

La classification par thèmes étant jugée trop ouverte, j'ai envisagé de proposer une classification qui laisserait moins de libertés à l'utilisateur : une classification hiérarchique. L'idée est d'associer un document non plus à cinq thèmes mais à une rubrique, un sujet et de un à trois thèmes.

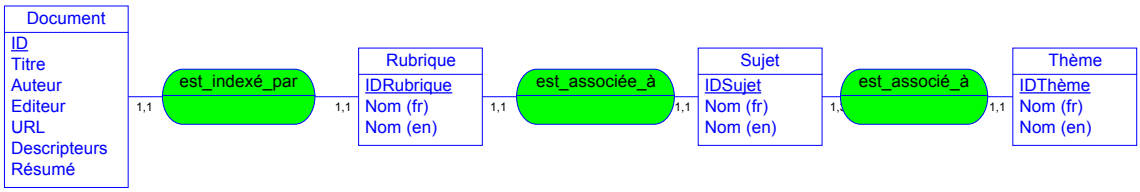
Voici les principales règles adoptées :

- Les rubriques sont au nombre de trois : *admin* qualifie tous les documents administratifs (cette rubrique concerne principalement le travail des deux cyber-secrétaires) ; dans la rubrique *cri_tech* se trouveront tous les documents internes créés par les membres du CRI et concernant les missions du CRI ; enfin, la rubrique nommée *tech* concerne les documents actuellement sur le serveur de documents phpDocServ.
- La liste des rubriques ne peut être modifiée par l'utilisateur
- Une rubrique est associée à un ou plusieurs sujets.
- Un sujet peut être associé à plusieurs thèmes
- Un thème peut être associé à plusieurs sujets

- A un document est associée une rubrique et une seule, un sujet et un seul, et de un à trois thèmes.

Voici le modèle conceptuel de données illustrant ce principe d'indexation :

Modèle Conceptuel de Données



Cette solution offre l'avantage de diminuer le nombre de chemins possibles pour accéder à un même document : si vous décrivez un document grâce à 5 thèmes différents, vous avez, comme nous l'avons vu, 120 chemins possibles pour aboutir au document.

En imposant le choix de deux de ces thèmes (ici la rubrique et le sujet), il n'y a plus que 6 solutions possibles (3*2*1) ; le système est donc moins lourd. En contrepartie, il faut que le système soit efficace : la liste de thèmes devra contenir des termes très précis.

1.3. Fichier de classification

Le système de classification doit être associé au document. Deux méthodes sont possibles : soit la classification est intégrée dans l'en-tête du document, soit la classification est enregistrée dans un fichier séparé, associé à la ressource.

La question est donc de savoir si l'on modifie la DTD XML de chaque type de document pour y intégrer des balises de classification. Cette solution aurait été idéale : elle aurait permis de gérer dans un même fichier le document et les thèmes qui lui sont associés ; toutefois, une raison essentielle a fait que cette solution n'a pas été retenue : depuis début 2002 et le basculement en XML, de nombreux documents ont été réalisés ; avant la mise en place du futur système de gestion de documentation, de nombreux autres documents seront créés. Il aurait donc fallu entièrement modifier ces fichiers pour y intégrer les nouvelles balises. Pour ne pas avoir à réaliser ce travail fastidieux, on crée un fichier de classification associé au document (qu'il ait été créé depuis quelques mois ou nouveau) contenant les

balises de classification. Le nom du fichier de classification doit respecter une règle : il porte le nom du fichier initial, auquel on ajoute le suffixe *_classification* ; par exemple, si le fichier initial est *monfichier.xml*, le fichier de classification associé portera le nom *monfichier_classification.xml*.

2. La base de données

Parmi les différentes bases de données qui pourraient être utilisées, on a le choix entre une base SQL de type PostGreSQL ou MySQL, et une base de données XML. Une base de données XML serait idéale car rendrait homogène l'ensemble du système. Pour cette étude, j'ai été aidée d'un autre stagiaire, Nicolas Demailly, qui a testé les différentes bases de données dans le but de déterminer laquelle ou lesquelles serai(en)t la(les) mieux adaptée(s) à notre système. Son travail a abouti à la conclusion qu'aucune base de données XML ne donne satisfaction. Il a testé notamment les bases de données relationnelles natives XML et Open Source nommées DBDOM et eXist. Ces bases de données sont implémentées en Java ; les spécifications du langage sont publiques (Open Source) mais les machines virtuelles Java ne sont pas libres ; c'est la raison principale qui a fait que les bases de données implémentées en Java ont été écartées.

Le principe adopté est de vider la base à chaque mise à jour et de la regénérer entièrement. A priori, il semble plus facile de mettre à jour régulièrement la base de données et de ne prendre en compte que les modifications depuis la dernière mise à jour; dans la pratique, il est délicat de faire cette différence entre les deux mises à jour. L'expérience montre qu'il est préférable de regénérer entièrement la base de façon à garder une homogénéité des données et à réduire au maximum le risque d'erreurs.

En outre, la quantité de données n'étant pas très importante, la différence de performance entre les deux méthodes possibles est négligeable.

En revanche, avec l'augmentation du corpus, il faudra vraisemblablement imaginer une autre méthode de stockage et de visualisation des données. Une base de données XML serait la solution idéale; comme nous l'avons vu, à l'heure actuelle, aucune base XML ne répond aux besoins; le monde XML étant d'une grande

évolutivité, il est tout à fait envisageable d'espérer avoir à notre disposition une base XML adéquate dans les années à venir.

3. L'intégration des métadonnées

Dans l'objectif d'exporter un document, le CRI souhaite que ce document corresponde aux normes en vigueur concernant les métadonnées.
Comme nous l'avons vu, le CRI possède des DTD XML pour chaque type de document interne. Pour que le document XML créé soit valide par rapport à cette DTD, l'auteur doit renseigner un certain nombre de balises. L'ajout de métadonnées aurait constitué un travail supplémentaire pour l'auteur. Ce constat a abouti à l'idée de créer un outil permettant de remplir automatiquement un certain nombre de balises. En effet, les informations contenues dans les balises de métadonnées et celles contenues dans les balises définies par la DTD sont souvent des informations redondantes.
Le choix s'est porté sur le standard de métadonnées Dublin Core [6] qui est un standard parmi les plus développés et les plus utilisés. Dans l'avenir, le système pourra être adapté à tout autre standard de métadonnées.
J'ai comparé les balises de métadonnées recommandées par le Dublin Core et les balises propres au CRI. J'ai établi une liste de balises qui pouvaient être automatiquement renseignées à partir des données de l'auteur et une liste de balises Dublin Core ne correspondant à aucune balise CRI ; ces dernières devront donc être renseignées par l'auteur lui-même.
Le bilan de cette comparaison est très satisfaisant : en effet, la quasi-totalité des balises Dublin Core correspondent à des balises du CRI ; elles peuvent donc être générées automatiquement ; seules quatre des quinze balises Dublin Core n'ont pas d'équivalent au niveau des balises CRI. De même, certaines balises CRI n'ont pas d'équivalent parmi les balises Dublin Core.
Le bilan de cette comparaison est disponible à l'annexe 4.

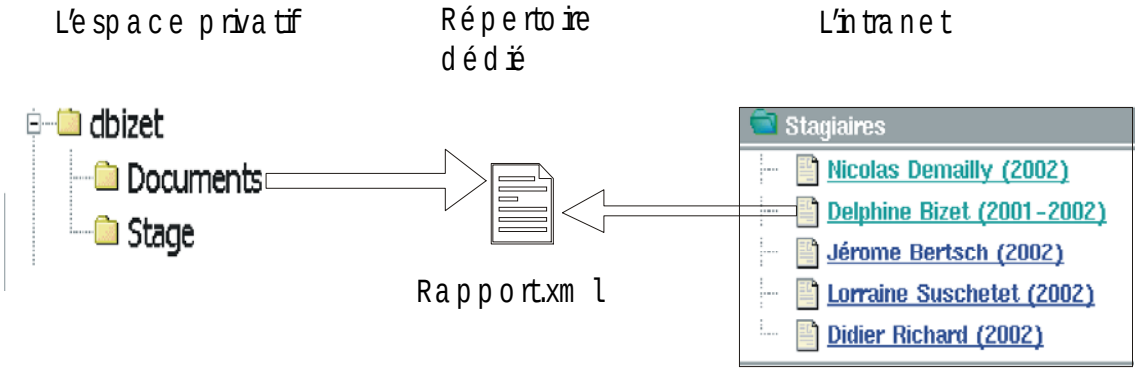
4. La gestion de la documentation interne

Comme nous l'avons vu, l'intranet est composé de pointeurs vers les documents présents sur l'espace privatif de l'auteur. C'est le principal inconvénient de ce système : en effet, en cas de modifications de l'arborescence des fichiers au sein de l'espace privatif, les liens entre l'intranet et le document sont irrémédiablement rompus. Comme le document est inaccessible à toute autre personne que le titulaire du compte utilisateur, il n'est pas possible de résoudre aisément le problème.

La solution la plus évidente mais la plus complexe à réaliser est de modifier les liens vers les documents : les documents doivent être enregistrés ailleurs que dans l'espace privatif de l'auteur ; un pointeur permet de pointer de l'espace privatif vers le document, un autre pointeur permet à l'intranet de pointer vers le même document. Pour casser un lien, il faudra alors que le document soit supprimé du serveur.

Par contre, cette solution implique une gestion rigoureuse des droits d'accès au niveau du document.

Le schéma 1 montre la situation actuelle. Voici une représentation des améliorations envisagées :



Lors de la mise en œuvre, on envisagera un délai conséquent pour que l'ensemble du corpus soit regroupé sous un répertoire dédié.

5. La gestion de la documentation externe

L'analyse de phpDocServ a constitué une étape importante de mon stage. L'objectif est de déterminer les points forts et les points faibles de ce système. Bien que le système soit moins utilisé qu'à ses débuts (principalement pour des raisons de manque de temps), il est performant ; le désir est de le faire évoluer, non de repenser entièrement le système.

5.1. Avantages de phpDocServ

L'utilisateur du système a la possibilité d'entrer de nouveaux thèmes pour indexer son document ; le fait que le plan de classification soit ouvert est un aspect qui doit être reconduit dans le nouveau système.

La recherche en texte intégral se fait grâce au moteur htdig ; elle est très performante.

5.2. Inconvénients de phpDocServ

PhpDocServ possède une classification à plat ; en conséquence, les problèmes de bruit et de silence sont devenus ingérables avec l'augmentation du nombre de documents présents sur le serveur.

Par ailleurs, nous en sommes arrivés à des aberrations : par exemple, des thèmes identiques se retrouvent dans la liste de thèmes; la seule différence entre eux est un espace ou une majuscule lors de l'ajout d'un thème…Autant dire que la recherche d'un document peut s'avérer difficile. Ceci est dû à un manque de rigueur du groupe des administrateurs de phpDocServ. Certaines règles, notamment syntaxiques, auraient dû être adoptées dès le départ et les administrateurs auraient dû se montrer plus rigoureux sur le respect de ces règles, soit en les faisant scrupuleusement respecter aux utilisateurs, soit en corrigeant les erreurs de ces utilisateurs avant de valider les données.

Le principe de la recherche par mots-clés sera reconduit. En revanche, la recherche actuelle s'effectue sur la quasi-totalité des champs de la fiche descriptive (pas seulement le champ mots-clés). Il faudra donc réduire le nombre de champs concernés pour améliorer ce type de recherche.

La réflexion concernant chacun de ces modules a abouti à une réflexion sur le système de gestion de documentation dans sa globalité.

Propositions de solutions de réorganisation

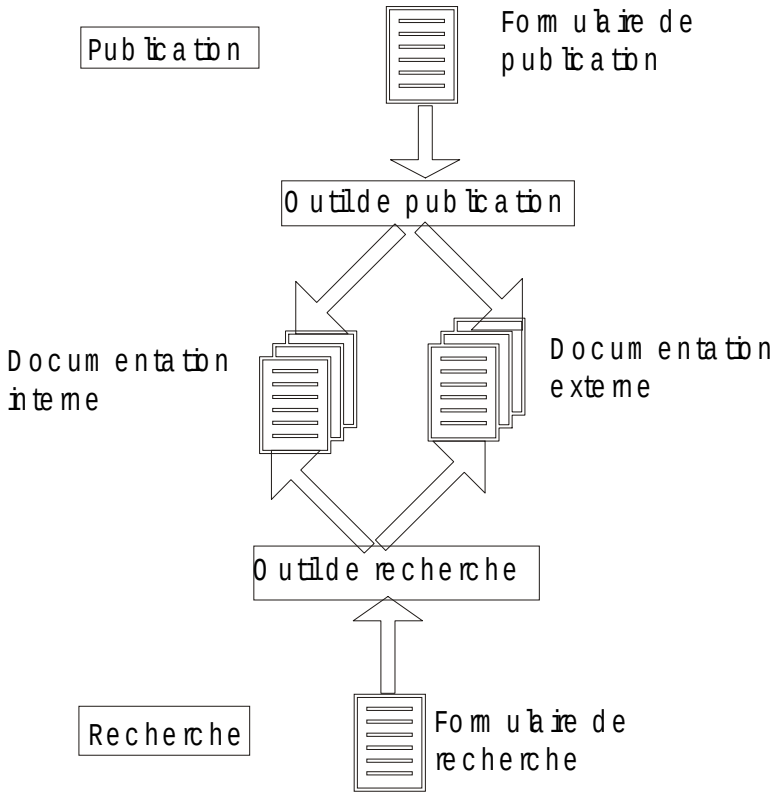
Il est nécessaire de proposer des solutions plus globales pour prendre en compte les modifications induites par l'élargissement du fonds documentaire à la documentation d'origine diverse.
Les réunions régulières, la plupart informelles, les différentes études menées et les informations que j'ai pu recueillir auprès des utilisateurs m'ont permis de proposer deux solutions :

1. Première solution : Ajout d'outils de publication et de recherche

1.1. Présentation de la solution

Cette solution consiste à laisser la documentation enregistrée sous phpDocServ ; les seuls documents déplacés seraient les documents internes qui seraient enregistrés sous un répertoire dédié. Nous ajouterions des outils en amont et en aval de la base de données. Par conséquent, un outil permettrait l'alimentation de la base et un moteur permettrait la recherche de données parmi l'ensemble de documents.
Le regroupement de toute la documentation interne permettrait de résoudre le problème de liens.
Le plan de classification s'appliquerait aux documents externes seulement.
Concernant le moteur de recherche, l'utilisateur entre une requête et la recherche portera automatiquement sur la (les) collection(s) concernée(s) par la requête.
Par conséquent, l'outil joue le rôle d'interface entre les producteurs et la base d'une part, et entre la base et les consommateurs d'autre part.
En outre, les droits d'accès seraient gérés au niveau de chaque document de la collection.

Notons que la documentation interne et externe constituent ici deux collections bien différentes.



1.2. Avantages

Le principal avantage est d'avoir un outil de production unique : tous les documents sont enregistrés au même endroit par l'auteur ; c'est l'outil qui se charge de dispatcher les documents soit vers le répertoire réservé aux documents internes, soit vers le répertoire des documents externes.

Par exemple, si l'utilisateur veut entrer une notice décrivant un site Web, il enregistre les différents champs de la notice au niveau du formulaire ; il valide la notice ; ensuite, l'outil se charge de stocker cette ressource sous le répertoire spécifique aux documents externes.

Cette solution permet d'avoir un système de classification pour chaque outil ; selon le type de documents, on pourra avoir une classification adaptée.

Lors de la recherche, l'outil saura dans quel répertoire aller chercher le document demandé ; par exemple, si l'utilisateur choisit le type « URL », l'outil effectuera la recherche dans le répertoire contenant les documents externes.

Par conséquent, la recherche sera a priori plus rapide car elle ne s'effectuera que sur une partie seulement du corpus.
Enfin, cette solution engendre peu de modifications au niveau de l'utilisation des outils. Les usagers devront seulement modifier le mode de publication de leurs documents personnels, ceux-ci devant être enregistrés dans un répertoire spécifique.

1.3. Inconvénients

L'inconvénient majeur est que la documentation reste éparpillée, ce qui engendre un risque de perte d'informations et un risque de redondance d'informations.
De plus, à terme, le risque est grand d'avoir deux systèmes de classement différents pour chaque outil, ce qui les rendraient difficilement compatibles.
Enfin, les outils qui sont évoqués ici seraient à programmer en interne car ils devront s'adapter aux spécificités du CRI. Le coût en moyens humains n'est pas à négliger.

2. Seconde solution : Création d'un serveur de documents

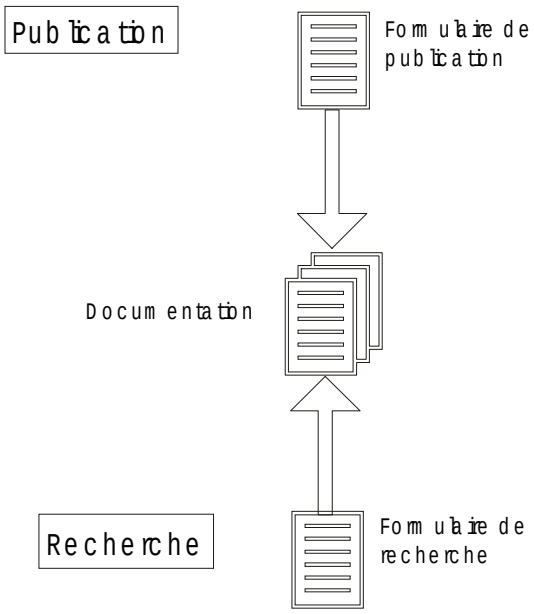
2.1. Présentation de la solution

Le principe, ici, est différent : on regrouperait tous les documents, quelle que soit leur origine, sous le même répertoire ; c'est ce répertoire qui ferait le lien entre les producteurs et les consommateurs d'informations. Les producteurs publieraient leurs documents sous ce répertoire; les informations seraient transférées vers une base de données unique ; les outils de recherche attaqueraient la base unique pour accéder et retrouver les documents.
Depuis l'espace privatif, il sera possible d'accéder aux documents présents physiquement sur le serveur.

Les droits d'accès seront gérés au niveau de l'espace privatif : certains utilisateurs pourront seulement consulter (ils auront alors le statut d' « anonymous »), d'autres pourront consulter et gérer leurs documents.

L'espace privatif constituera le point d'accès unique à la base, aussi bien pour la gestion des documents (publication, modification, suppression) que pour la consultation.

Lors de la connexion à l'intranet de l'espace privatif, on peut envisager la génération automatique d'un menu (propre à chacun selon les documents qu'il gère et selon ses droits d'accès) permettant d'accéder au corpus de documents accessibles.



2.2. Avantages

Les producteurs produisent à un seul endroit grâce à un formulaire unique.

Le moteur de recherche attaque une seule base.

Le système de classification ne s'applique qu'à un corpus de documents.

Les droits d'accès sont facilement gérables au niveau de l'espace privatif car l'accès à son espace privatif nécessite une identification préalable. On pourra donc établir une liste des utilisateurs ayant seulement un droit en lecture et une liste des utilisateurs ayant un droit en lecture et en écriture.

La gestion des droits d'accès au niveau de l'espace privatif permet de créer des vues adaptées au profil de l'utilisateur. On peut alors envisager la création d'un menu propre à chacun, qui serait généré automatiquement à chaque connexion à son espace privatif.

Cette solution pourrait reposer sur la plus grande partie des concepts qui régissent actuellement le serveur de document phpDocServ. Quelques adaptations seraient à effectuer au niveau du système de classification.

Cette solution engendre, à terme :

- La disparition du serveur de documentation phpDocServ dans son état actuel
- Le transfert de la documentation stockée sous phpDocServ et sous les différents espaces privatifs vers le nouveau répertoire dédié à la documentation
- La mise en place d'une procédure unique de publication des documents
- La création d'une nouvelle base de données à partir des fichiers enregistrés sous ce répertoire
- La création d'un moteur de recherche attaquant la "nouvelle" base

2.3. Inconvénients

Le principal inconvénient est que le système de classification devra être pertinent et évolutif pour s'adapter à l'augmentation du nombre de documents au fil des ans.

Mise en œuvre

Lors de ce stage, j'ai non seulement pu proposer des solutions pour améliorer le système, mais j'ai aussi été intégrée au choix d'une solution et à la mise en oeuvre de celle-ci. En effet, j'ai programmé quelques fonctionnalités du serveur de documents. Il a été décidé de commencer par la gestion de la documentation interne car c'est ce point qui pose le plus de problèmes et nécessite le plus de travail.

Certains tâches de programmation ont été laissés de côté ; elles seront réalisées ultérieurement par les développeurs du CRI. Il est prévu que l'application soit opérationnelle au début 2003.

1. Le choix du langage de programmation

Le langage utilisé est le langage XML. La politique du CRI en matière de logiciels est depuis toujours d'utiliser si possible des logiciels libres; c'est un principe très ancré. On utilise des logiciels propriétaires seulement si aucun logiciel libre sur le marché ne correspond aux besoins.
Dans cette optique, le langage XML est non seulement libre, mais il offre des possibilités évidentes en matière de documentation électronique.

1.1. Les avantages de XML

- XML est un standard ouvert ; il est dérivé d'un standard solide : SGML.
- Le standard XML a été établi par le W3C (World Wide Web Consortium)
- XML permet de distinguer la structure du document de son contenu et de sa présentation : les feuilles de style XSL contrôlent la présentation des documents XML; les feuilles de style sont des fichiers distincts des

documents XML; ceci implique la possibilité de créer différentes vues sur un même document sans changer le contenu de celui-ci. En outre, le contenu peut être présenté sous différentes formes à différents utilisateurs.

- XML fournit une description plus précise du contenu du document en acceptant un ensemble de balises extensibles : les implémenteurs XML peuvent définir leurs propres ensembles de balises.
- Le contenu et la structure d'un document XML est définie dans une grammaire. La grammaire décrit les balises valides, les attributs (caractéristiques des balises telles que l'identifiant) et autres contenus pour le document XML. Lors de la création d'un document, l'auteur est responsable de la conformité du fichier à la grammaire.
- XML facilite l'échange de documents entre utilisateurs et applications : un de ses avantages est de transmettre le contenu des documents sources en format de sortie tels que HTML, PDF ou PS.
- La recherche dans un document XML est plus facile car la structure et le sens du contenu sont identifiés (comme défini dans la grammaire).
- Le code XML est lisible sur n 'importe quel éditeur de textes.

1.2. Pourquoi ne pas avoir choisi d'autres DTD classiques telles que DocBook

DocBook est par définition une application SGML, comme le HTML. Mais il existe aussi une version XML de DocBook. La DTD DocBook a été créée pour les documentations techniques liées à l'informatique ; elle convient toutefois aussi pour des domaines moins spécifiques.

Nous aurions pu utiliser cette version pour notre application ; mais cette DTD est beaucoup trop lourde à utiliser pour des documents simples (les notices que nous enregistrons par exemple). Par contre, il n'est pas impossible de transformer les documents XML du CRI en DocBook.

1.3. Les outils utilisés

Le CRI utilise différents outils pour programmer en XML : l'éditeur de texte Xemacs, le processeur xsltproc, les librairies XML et des librairies Perl. Je n'évoquerai que les trois premiers outils car je n'ai pas eu l'occasion d'utiliser des librairies Perl.

- Xemacs est un éditeur, disponible sous Linux et Windows, sous licence GPL, est particulièrement adapté à la programmation XML du fait de son mode SGML. L'avantage d'Xemacs est qu'il reconnaît automatiquement certains types de fichiers comme les .html ou .xml et met en évidence les mots-clés associés.
- Le programme xsltproc est proposé par la distribution Debian. Il s'appuie sur la bibliothèque de fonctions libxml écrite en langage C. Ce programme permet de réaliser des transformations XSL.
- libXML est une bibliothèque XML. C'est un analyseur ou parser : un analyseur est un composant logiciel permettant d'accéder simplement aux données encapsulées dans un fichier XML. C'est donc le composant de base de toute application XML.Il permet entre autre de valider des documents. Cette bibliothèque est également sous licence GPL.

Tout ne peut être programmer en XML ou en scripts shell ; des langages tels que Perl ou php devront être utilisés ; cette étape de programmation ne me concerne pas.

2. Mes réalisations

La phase de programmation que j'ai effectuée concerne la gestion des métadonnées et le système de classification des documents.

Comme nous l'avons vu, le CRI possède un système de classification propre. Dans le but d'exporter facilement les documents, il est nécessaire de respecter le standard de métadonnées Dublin Core.

Les objectifs à respecter sont divers :

- Le fichier de classification généré devra comporter deux types de données :
  - les données de classification du CRI
  - les données Dublin Core
- Le second objectif est de minimiser au maximum le travail de la personne qui publie le document : dans la mesure du possible, on utilisera les données définies comme obligatoires pour générer les données du fichier de classification.
- Que le document soit d'origine interne ou externe, la même opération devra permettre de générer le fichier de classification

J'ai donc créé une feuille de style XSLT permettant de générer un fichier XML contenant toutes les balises nécessaires à la classification du document (les balises propres au CRI et les balises Dublin Core)

Un document XML comporte des balises qui doivent être renseignées par son auteur ; le principe est de récupérer parmi ces données les données relatives à la classification et de les recopier dans les balises du fichier de classification généré.

Par exemple, la DTD définit une balise *<title>* ; l'auteur doit obligatoirement renseigner cette balise, par exemple *<title>Comprendre XML</title>* ; la valeur de la balise sera recopiée pour être insérée dans la balise *<dc:title>*, qui est une balise définie par le W3C.

Après la génération du fichier, nous aurons donc le document XML contenant entre autre la balise de titre *<title>Comprendre XML</title>* d'une part, et le fichier de classification contenant entre autre la balise *<dc:title>Comprendre XML</dc :title>*.

Les problèmes se posent quand les balises ne correspondent pas : par exemple, le CRI possède une balise *<license>* et une balise *<copyright>* ; le standard ne définit qu'une seule balise *<dc:rights>* destinée à recevoir les informations relatives à la licence et au copyright du document. Dans ce cas, et comme les éléments de métadonnées sont répétables, nous avons décidé de créer deux balises *<dc :rights>*, l'une étant renseignée par la balise *<license>*, l'autre étant renseignée par la balise *<copyright>* ; en cas d'exportation du document, toute la difficulté sera de pouvoir extraire les bonnes informations des bonnes balises.

A l'heure actuelle, j'ai terminé la feuille de style et le script permettant de générer le fichier de classification. Le script shell permet d'appliquer la transformation définie par la feuille de style à un ensemble de fichiers XML.
Avant d'effectuer la génération, on vérifie que le document possède bien un identifiant unique ; si l'identifiant est erroné ou absent, le système en génère un automatiquement et s'en sert lors de la génération.
Les documents qui seront enregistrés dans la base de données doivent pouvoir être facilement retrouvés ; par conséquent, la condition sine qua none de leur transfert vers la base de données est qu'ils possèdent bien une classification adéquate permettant d'effectuer une recherche sur différents champs.
Par conséquent, avant le transfert vers la base, on réalise une double vérification :

- On vérifie que le fichier de classification associé au–dit document existe bien.
- On vérifie ensuite que les balises nécessaires à la classification au sein du CRI soient bien renseignées.

3. La suite du projet docServ

Différentes tâches restent à effectuer :

- La création de fichiers XML contenant les données actuellement enregistrées dans la base de phpDocServ.
- La création d'une base de données à remplir avec les données extraites de ces fichiers XML.
- La programmation du système permettant la publication de tout document.
- La programmation du moteur de recherche.

Les trois dernières tâches sont de la programmation. Elles seront effectuées par les membres du CRI ; quant à moi, je m'attacherai, durant ce dernier mois de stage, à effectuer la tâche consistant à intégrer les données de la base dans des fichiers XML.

Conclusion

La réorganisation du système documentaire a permis de réfléchir à différents points cruciaux pour améliorer le système.
Les améliorations proposées seront mises en œuvre.
La mise en œuvre de la solution adoptée a commencé ; d'ores et déjà, il est possible d'associer à tout document XML un fichier contenant les informations relatives à sa classification.
Jusqu'à la fin du stage, je m'attacherai à réaliser le transfert des données contenues dans la base de phpDocServ en documents XML.
Après mon départ, un échéancier sera mis en place pour planifier les tâches restantes. Les développeurs du CRI vont poursuivre la programmation de l'application ; celle-ci devrait être opérationnelle au début de l'année 2003.

Glossaire

AFUL

L'AFUL est l'Association Francophone des Utilisateurs de Linux et des logiciels libres. C'est une association loi 1901 dont l'objectif principal est de promouvoir, directement ou indirectement, les logiciels libres et en particulier les systèmes d'exploitation libres dont le plus connu est le système Linux.

Analyseur XML

Outil analysant et décodant les balises d'un document XML afin de permettre à l'application utilisant cet analyseur de traiter "facilement" des données au format XML

Balise / Tag / Markup

Permet de délimiter des données dans les langages de balisage tels que HTML ou XML.

DocBook

DocBook est un modèle de documentation technique actuellement maintenu par un comité technique du consortium OASIS Open. Son utilisation s'avère particulièrement intéressante dès qu'il s'agit de créer des documents techniques ou des articles.

La recommandation DocBook est issue du monde de l'informatique et de l'électronique. Du coup, il existe beaucoup d'éléments permettant de coder des informations liées à ce domaine.

Les modèles issus de DocBook permettent d'ajouter ses propres éléments dans le modèle de base. De même, si des objets ne sont pas utiles, ils peuvent être aisément supprimés. Du coup, beaucoup d'industriels se sont emparés de ce modèle et l'ont adapté à leurs propres besoins.

DocBook est à la base une application SGML. Il existe une version XML. Contrairement au HTML, DocBook ne fournit pas d'informations concernant la

mise en page. Ainsi, il est possible de convertir les documents créés en formats tels que le postscript, le pdf, le html…

Dublin Core

Raccourci utilisé pour Dublin Core Metadata Initiative

Dublin Core Metadata Initiative (DCMI)

Ensemble de métadonnées défini en 1995 par le NCSA (National Center for Supercomputing Applications) et l'OCLC (Online Computer Library Center) réunis au siège de l'OCLC à Dublin, Ohio.

DTD (Document Type Definition – Définition de Type de Document ou Déclaration de Type de Document)

Une DTD est une définition formelle des éléments, de la structure et des règles pour un type donné de document XML. La DTD peut être stockée au début du document ou dans un fichier séparé. Les règles définissent généralement le nom et le contenu de chaque balise et le contexte dans lequel elles doivent exister. Cette formalisation des éléments est particulièrement utile lorsqu'on utilise de façon récurrente des balises dans un document XML.

Elément

Unité de donnée ou de métadonnée

E-zine (electronic magazine)

Magazine électronique, généralement gratuit. On parle aussi de webzine ou de e-journal.

Freeware

Programme que tout le monde peut utiliser et distribuer sans payer de droits.

Htdig ou Ht://Dig

Ht://Dig est un logiciel d'indexation et de recherche de documents par mots-clefs à l'intérieur d'un site web donné. C'est typiquement le genre de logiciel qui est utilisé lorsqu'un site web offre des fonctionnalités du type "rechercher sur ce site".

Ht://Dig se distingue des autres logiciels du même genre en étant un logiciel libre distribué sous licence GPL (GNU General Public Licence) et aussi en étant l'un des plus performants. Il est de ce fait le moteur de recherche favori des distributions de Linux et des sites dédiés au logiciel libre.

Hyperlien

Mot, phrase ou graphique permettant de se référencer à un autre document ou partie de document.

Hypertexte

Document contenant des hyperliens vers d'autres documents. En sélectionnant un lien, le document référencé est automatiquement affiché.

HTML HyperText Markup Language

Langage destiné à décrire les pages Web et les documents hypertextes. HTML dérive de SGML (Standard Generalized Markup Language). Après les ébauches de Tim Berners-Lee entre 1991 et 1993, les premières versions normalisées (HTML 2.0 et 3.0), apparaissent en 1995

Intranet

Réseau interne à une entreprise, qui fonctionne et s'utilise comme Internet. Ce n'est ni plus ni moins qu'un réseau local doté d'outils de navigation.

Java

Java est un langage de programmation orienté objet proche du C++. Java est un langage évolué, robuste et sécurisé qui s'est très vite imposé comme le langage de prédilection pour les applications "Internet" et distribuées.

Licence GPL

Le code source de Linux est accessible gratuitement, ce qui fait que ce système peut être compilé sur d'autres plates-formes que le PC. Afin de permettre la distribution de programmes exempts de droits, la fondation FSF (Free Software Foundation, Fondation pour les logiciels libres) a développé un projet nommé

GNU. Les utilitaires GNU sont soumis à une licence d'utilisation de Linux expliquant les dispositions légales vis-à-vis de l'utilisation, la distribution et la modification de Linux : la licence GPL (General Public Licence),.
Voici à titre indicatif quelques aspects de cette licence permettant de la comprendre:

- la licence permet la modification du programme original, et sa diffusion (sous licence GPL)
- la licence autorise la vente du logiciel freeware (gratuit...) sous sa forme originelle ou modifiée, à condition que le vendeur autorise la diffusion (même gratuite) de ce logiciel
- la licence autorise l'utilisation du logiciel à des fins lucratives (permettant des bénéfices)
- les logiciels sous la licence GPL appartiennent à leurs auteurs et personne ne peut s'approprier une partie ou l'intégralité des droits d'auteur
- la licence n'implique aucune forme de rémunération des auteurs

Cette licence est parfois appelée *copyleft*, par analogie au copyright.

Linux

Système d'exploitation de type Unix développé sous forme de logiciel libre. Distribué gratuitement le plus souvent. Le noyau de Linux a été développé par Linus Torvalds à l'Université d'Helsinki. Linus B.Thorvald est l'inventeur de ce système d'exploitation entièrement gratuit. Au début des années 90, il voulait mettre au point son propre système d'exploitation pendant ses loisirs. Amusées par cette initiative, de nombreuses personnes ont contribué à aider Linus Thorvald à réaliser ce système, si bien qu'en 1991 une première version du système a vu le jour. C'est en mars 1992 qu'a été diffusée la première version ne comportant quasiment aucun bug.

Logiciel libre

D'après les statuts de l'AFUL, sont considérés comme libres les logiciels disponibles sous forme de code source, librement redistribuables et modifiables,

selon des termes proches des licences « GPL », « Berkeley » ou « artistique »et plus généralement des recommandations du groupe « Open Source ».
Un logiciel libre n'est pas forcément gratuit. L'ambiguïté provient de l'expression d'origine, « free software », puisqu'en américain « free » signifie aussi bien « libre » que « gratuit ». Dans la pratique, nombre de logiciels libres se trouvent gratuitement sur certains sites web. Des versions payantes, mais souvent très bon marché, sont commercialisées par des entreprises sous forme de CD-ROM, avec notice complète, et contrat d'assistance à l'installation ou contrat de maintenance. Les sociétés MandrakeSoft (France), RedHat (USA), Caldera (USA) et SuSE (Allemagne) distribuent ainsi différentes version de Linux, le logiciel libre le plus connu.

Machine virtuelle Java ou Interpréteur Java
Couche logicielle présente dans tous les navigateurs Java, dans laquelle tourne l'application sous forme de code objet issu du compilateur Java, et qui traduit ce code objet en code exécutable par l'ordinateur client.
Cette solution logicielle permet de faire tourner des applications Java sur n'importe quelle plate-forme, pourvu que le navigateur Web utilisé soit compatible avec Java.
La machine virtuelle immunise l'ordinateur client contre les virus et les bugs. En effet, une attaque n'affectera finalement que la machine virtuelle, l'environnement cible étant protégé.

Métadonnée
Une métadonnée est littéralement une « donnée sur une donnée ». Plus précisément, c'est un ensemble structuré d'informations décrivant une ressource quelconque. Les ressources décrites par des métadonnées ne sont pas nécessairement sous forme digitale: un catalogue de bibliothèque ou de musée contiennent aussi des métadonnées décrivant les ressources que sont les ouvrages de la bibliothèque ou les objets du musée. Une métadonnée peut être utilisée à des fins diverses:

- la description et la recherche de ressources

- la gestion de collections de ressources
- la préservation des ressources

Les métadonnées sont utilisées dans les systèmes de gestion de contenu pour éditer, gérer, rechercher, réutiliser, diffuser, publier de multiples contenus (textes, images, vidéo…)

Les informations contenues dans les métadonnées sont utilisées par les moteurs de recherche pour assurer le repérage des sites. Un métamoteur de recherche compte sur un outil appelé « spider » pour circuler parmi les sites et repérer des pages qu'il pourra ensuite indexer de manière automatique.

Moteur de recherche

Système d'interrogation de bases de données. Un robot parcourt les documents présents sur le web pour les indexer sur des serveurs. Lors d'une recherche sur un moteur, l'internaute lance une requête sur les bases de données de ces serveurs qui contiennent des millions de pages web

MySQL

Gestionnaire de base de données relationnelle fonctionnant notamment sur les serveurs Apache. S'exploite souvent avec PHP.

NCSA - National Center for Supercomputing Application

Centre spécialisé dans la recherche en communication, situé à l'Université de l'Illinois

News

Articles publiés dans un newsgroup

Newsgroups

Forum de discussion permettant l'échange d'informations sur un même thème

Open Source

Logiciel dont le code, mis à la disposition de tous, peut être modifié librement (il est ouvert). La plupart des logiciels Open Source sont des logiciels libres, mais néanmoins l'usage de certains peut être limité, voire payant.

Parser

Un parser est un logiciel spécialisé dans la reconnaissance des balises dans un document. Un parser qui lit une DTD et vérifie et rapporte les erreurs de balisage est un parser valide.

PDF - Portable Document Format

Les documents enregistrés au format PDF peuvent être visualisés dans leur exacte mise en page quel que soit l'ordinateur sur lequel on les consulte. Pour lire les fichiers PDF, il faut télécharger Acrobat Reader. Pour créer des fichiers PDF, il faut acheter Adobe Acrobat ou un autre générateur de fichiers PDF.

PHP - Hypertext Preprocessor

Langage de programmation interprété. Développé initialement pour les serveurs Web tournant sur Linux et disponible maintenant sur tous les systèmes. Le code PHP s'exécute sur le serveur et le navigateur Web ne reçoit que le résultat de l'exécution. Les pages Web écrites en PHP ont l'extension php, php3 ou phtml. Initialement PHP signifiait Personal Home Page.

PS - PostScript

Langage de description de pages.

Resource Description Framework (RDF)

Le Resource Description Framework est un cadre pour les métadonnées; il permet l'interopérabilité entre les applications qui échangent sur le Web des informations déchiffrables par les machines.

Dans la plupart des cas, le RDF sert de cadre aux métadonnées Dublin Core, c'est-à-dire que les balises Dublin Core sont encapsulées dans les balises RDF.

Serveur

a) Logiciel permettant à un ordinateur d'offrir des services à d'autres ordinateurs connectés (les clients).
b) Ordinateur sur lequel tourne le logiciel serveur.

SGML (Standard Generalized Markup Language)

SGML est un outil de gestion de documents mais peu utile à la diffusion des données; il sert essentiellement à séparer l'information de sa présentation.

SQL (Simple Query Language)

Langage de description de requêtes dans une base de données.

Robot / Spider

Module d'un moteur de recherche qui parcourt le web et l'internet afin d'en mémoriser les documents (URL, mots clés, corps du texte) pour alimenter l'index des moteurs.

Unicode

Unicode spécifie un numéro unique pour chaque caractère, quelle que soit la plate-forme, quel que soit le logiciel et quelle que soit la langue.
Grâce à Unicode, un seul logiciel ou site Internet peut satisfaire simultanément et sans modification les demandes de plusieurs plate-formes, langues et pays. Unicode permet aussi à des logiciels de provenance variée d'échanger des caractères sans pertes de données.
Le Consortium Unicode est une organisation sans but lucratif, ayant pour mission de développer, étendre et promouvoir Unicode.

XML (eXtensible Markup Language)

XML est un standard récent (adopté en 1998 par le W3C) qui s'appuie sur un standard solide (SGML) dont les origines remontent à 1960.
Ce métalangage (c'est-à-dire un langage servant à décrire d'autres langages) permet de définir des représentations de données ou d'informations structurées, et ce de façon ouverte et indépendante des plates-formes ou logiciels. Ainsi, XML n'est pas

un format de données, mais bien une norme qui permet de définir des formats de données.
La norme XML permet de modéliser l'information de façon précise, souple et efficace.
La norme XML définit très peu de choses. Elle fournit une syntaxe de base pour représenter de l'information électronique, sans plus. A partir de cette syntaxe de base, des outils peuvent être développés pour traiter ces informations électroniques.
Par ailleurs, XML étant une norme récente, elle s'appuie sur des développements récents de l'informatique. L'un de ces développements est le standard Unicode, qui vise à définir un jeu de caractères universel pour l'ensemble des écritures (actuelles et passées) du monde. Il est donc aisé de construire des systèmes documentaires multilingues en XML.
Enfin, XML a donné naissance à différentes normes satellites pour effectuer différents traitements sur l'information. Par exemple, XSLT permet la transformation des documents, XSL-FO le formatage et l'impression, Xlink les liens hypertextes, XQL permet le traitement des requêtes. En conséquence, l'information et les traitements sont représentés de façon normalisée.

XML Path :
XPath est une syntaxe standardisée pour localiser des parties d'un document XML.

XSL eXtensible Stylesheet Language - Langage de feuille de style extensible
Langage de présentation. Une feuille de style XSL permet de mettre en page et de reformater (ou réordonner) le document XML auquel elle est associée.

XSLT eXtensible Style Language Transformation
Métalangage pour la définition de règle de filtrage de contenus XML en vue de présenter ces contenus (ajouter d'information de style pour composer un document) ou de convertir la structure de ces contenus (échange de données structurées).

Document XML « valide »

Un document XML est dit "valide" lorsqu'il respecte les contraintes spécifiées dans sa DTD (s'il se contente de respecter la syntaxe XML, il est dit "bien formé").
Comme son nom l'indique, un analyseur "validant" effectue des contrôles lors de l'analyse du document et valide celui-ci en fonction de sa DTD (mais au prix d'un surcoût de traitement).

W3C, World Wide Web Consortium
Organisme de proposition et de normalisation des technologies, protocoles et langages du Web. A son actif, la standardisation de DOM, HTML, HTTP, XML, XSL, etc.
Site du W3C : www.w3.org

Bibliographie

Le Centre de Ressources Informatiques de Haute-Savoie

[1]**Centre de Ressources Informatiques de Haute-Savoie**. *Site du Centre de Ressources Informatiques de Haute-Savoie*, [En ligne]. http://www.cri74.org/ (Page consultée le 10 septembre 2002)

[2]**Conseil Général de Haute-Savoie**. *Site du Conseil Général de Haute-Savoie*, [En ligne]. http://www.cg74.fr/ (Page consultée le 10 septembre 2002)

[3]**Agence Economique Départementale**. *Site de l'Agence Econimique Départementale*, [En ligne]. http://www.hautesavoie.com/ (Page consultée le 10 septembre 2002)

Le serveur de documentation phpDocServ

[4] **Centre de Ressources Informatiques de Haute-Savoie.** *Le serveur de documents phpDocServ*, [En ligne]. http://www.phpdocserv.org/ (Page consultée le 10 septembre 2002)

[5]*Site de téléchargement de phpDocServ* : ftp://ftp.cri74.org/phpDocServ/ (Site consulté le 10 septembre 2002)

Le standard de métadonnées Dublin Core

[6]**Dublin Core Metadata Initiative (DCMI)** , [En ligne]. http://dublincore.org/ (Page consultée le 10 septembre 2002)

[7]**NCSA (National Center for Supercomputing Applications)** , [En ligne]. http://www.ncsa.uiuc.edu/ (Page consultée le 10 septembre 2002)

[8]**OCLC (Online Computer Library Center),** [En ligne]. http://www.oclc.org/home/ (Page consultée le 28 août 2002)

[9]**Anne-Marie Vercoustre,** *Eléments de métadonnées du Dublin Core, Version 1.1: Description de Référence*.[En ligne] http://www-rocq.inria.fr/~vercoust/METADATA/DC-fr.1.1.html (Page consultée le 28 août 2002)

[10]**Diane Hillmann, traduction de Guy Teasdale***, Guide d'utilisation du Dublin Core*; [En ligne] http://www.bibl.ulaval.ca/DublinCore/usageguide-20000716fr.htm (Page consultée le 28 août 2002)

[11] http://www.gnu.org/philosophy/free-sw.fr.html (Page consultée le 28 août 2002)

XML

[12] *W3C (World Wide Web Consortium)*.[En ligne] http://www.w3c.org (Page consultée le 28 août 2002)

[13] **W3C**, *Extensible Markup Language (XML)* [En ligne] http://www.w3.org/XML/ (Page consultée le 28 août 2002)

[14] Extensible Markup Language (XML) 1.0 (Second Edition) - W3C Recommandation du 6 Octobre 2000 [En ligne] http://www.w3.org/TR/REC-xml (Page consultée le 28 août 2002)

[15] Extensible Stylesheet Language (XSL) Version 1.0 - W3C Recommandation du 15 Octobre 2001 [En ligne] http://www.w3.org/TR/xsl/ (Page consultée le 28 août 2002)

[16] *Xml.fr.org* [En ligne] http://xmlfr.org/ (Page consultée le 28 août 2002)

[17] **W3C***, XML en 10 points*. [En ligne] http://www.w3.org/XML/1999/XML-in-10-points.fr.html (Page consultée le 28 août 2002)

[18] *L'altruiste - Le langage XML*.[En ligne] http://www.laltruiste.com/coursxml/sommaire.html (Page consultée le 28 août 2002)

[19] *Le site de l'AFUL*; [En ligne] http://www.aful.org/index.html (Page consultée le 28 août 2002)

[20] *Recommandantion du W3C du 6 Octobre 2000 - Extensible Markup Language (XML) 1.0 (Second Edition)* ; [En ligne] http://www.w3.org/TR/REC-xml (Page consultée le 28 août 2002)

[21] **ADLER Sharon, BERGLUND Anders**, *Recommendation du W3C du 15 Octobre 2001 (Version 1.0)* ; [En ligne] http://www.w3.org/TR/xsl/ (Page consultée le 28 août 2002)

[22] *Le site de Unicode Consortium* [En ligne] http://www.unicode.org/unicode/consortium/consort.html (Page consultée le 28 août 2002)

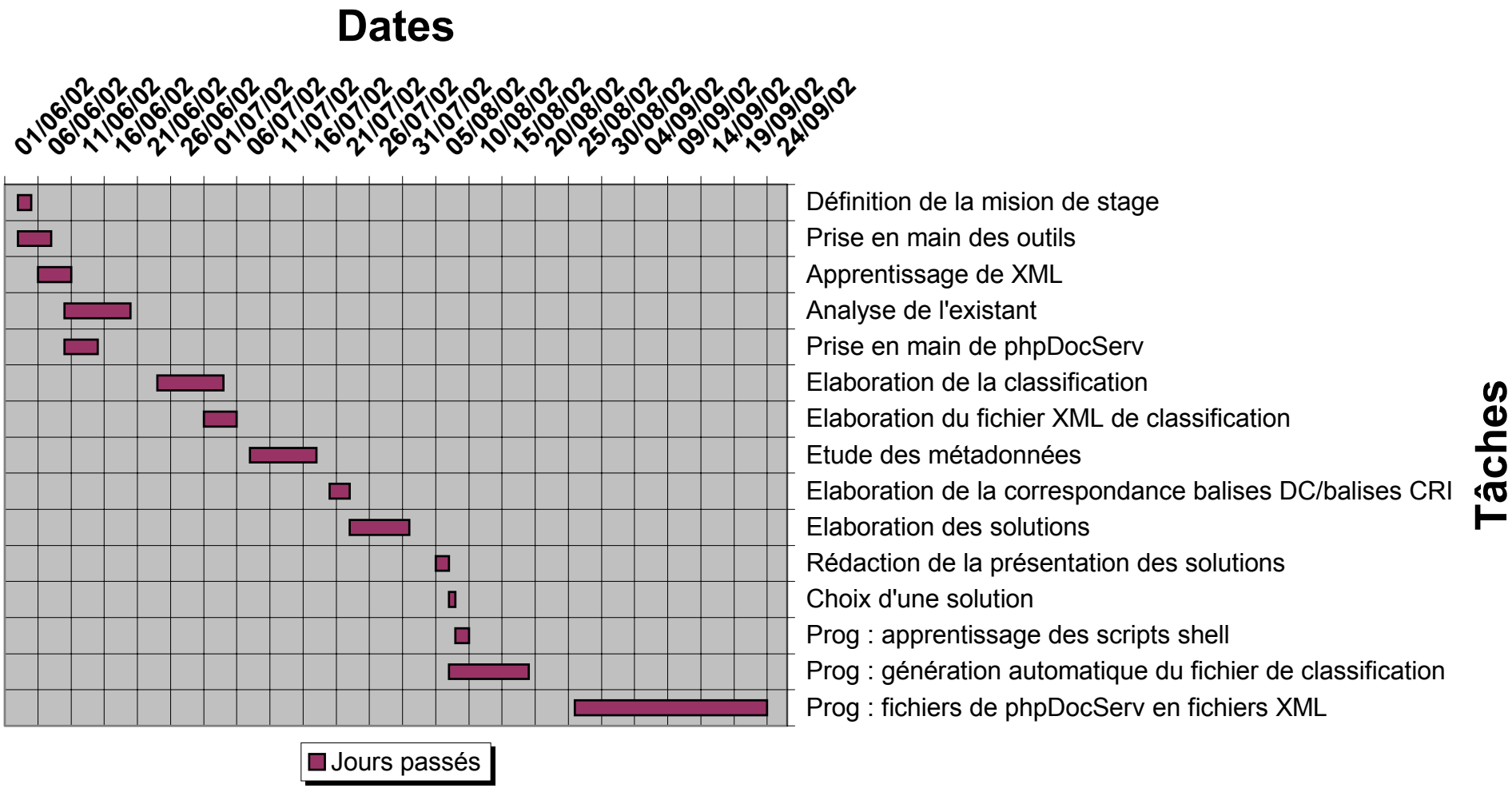
[23] *Le site des Rencontres Mondiales du Logiciel Libre (RMLL)* **:** http://lsm.abul.org/home.php3?langnew=fr

[24] **AMANN Bernd, RIGAUX Philippe**, *Comprendre XSLT*, Paris : O'REILLY, 2002

Table des annexes

Annexe 1 : Diagramme de Gantt

Diagramme de Gantt



Annexe 2 : Le standard de métadonnées Dublin Core

Dublin Core est le standard de métadonnées le plus répandu et le plus avancé pour la description des ressources internet.
Défini en 1995 à Dublin en Ohio par l'OCLC (Online Computer Library Center) et le NCSA (National Center for Supercomputing Applications), le développement du Dublin Core est assuré par le DC Directorate, supervisé par le OCLC (Office of Research and Special Projects).
Les autorités définissent 15 éléments ; ces éléments doivent être fournis par le producteur de la ressource ; ces éléments on trait :

- au contenu de la ressource : Title, Description, Subject, Source, Coverage, Type, Relation
- à la propriété intellectuelle : Creator, Contributor, Publisher, Rights
- à la version : Date, Format, Identifier, Language

Chaque élément est optionnel.
Chaque élément est répétable.
Le Dublin Core est extensible : chaque élément peut être enrichi d'attributs ; on parle alors de Dublin Core qualifié.
Le Dublin Core est multidisciplinaire.
Le Dublin Core est international : il supporte actuellement plus de 20 langues.

Le Dublin Core a été proposé pour faciliter la recherche de ressources peu complexes. Le Dublin Core ne prétend pas répondre aux besoins et à la complexité de tous les métiers. C'est pourquoi, dans certains domaines, des champs additionnels ou des schémas complémentaires sont nécessaires pour décrire correctement des structures spécifiques.

Annexe 3 : Comparaison des éléments Dublin Core et des balises propres au CRI

Balises DC	Balises CRI	Signification
dc:title	<info><title>Titre de la ressource</title></info>	Titre
dc:title alternative	<info><subtitle> Sous-titre_de_la_ressource </subtitle></info>	Sous-titre (éventuel)
dc:date created	<info><version date="date_de_création_de_la_version"></version/></info>	Date de création de la version
dc:contributor	<info><version><author><lastname> Nom_de_l'auteur_de_la_version </lastname></author></version></info>, <info><version><author><firstname> Prénom_de_l'auteur_de_la_version </firstname></author></version></info>	Nom de l'auteur de la version
dc:creator	<info><authors><author><lastname> Nom_de_l'auteur_du_document </lastname></author></authors></info>, <info><authors><author><firstname> Prénom_de_l'auteur_du_document </firstname></author></authors></info>	Nom de l'auteur du document
dc:rights	<info><copyright> Informations_sur_le_copyright </copyright></info>, <info><license> Informations_sur_la_licence </license></info>	Informations sur le droit de copyright et informations sur la licence
dc:description abstract	<info><abstract>Résumé</abstract></info>	Résumé du document
dc:subject	<info><keywords><keyword> Mots-clés </keyword></keywords></info>	Mots-clés

Annexe 4 : La DTD du projet de serveur de documents docServ [extrait]

```
<?xml version="1.0" encoding="iso-8859-1" ?>

<!ENTITY % coredoc "info,preamble?,chapter+,bookmarks?,glossary?,biblio?">
<!ELEMENT report (%coredoc;)>
  <!ATTLIST report id ID #IMPLIED>
<!ELEMENT info (title, subtitle?, logo?, version, authors, copyright*, license?, abstract?, keywords?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT subtitle (#PCDATA)>
<!ELEMENT authors (author+)>
<!ELEMENT version (author?,comment?)>
  <!ATTLIST version release CDATA #REQUIRED
                    date CDATA #REQUIRED>
<!ELEMENT author (firstname?, lastname?, email?, corp?)>
  <!ATTLIST author authorid IDREF #IMPLIED>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

```
<!ELEMENT corp (#PCDATA)>
<!ELEMENT keywords (keyword+)>
<!ELEMENT keyword (#PCDATA)>
<!ELEMENT chapter (title,(section|para)+)>
  <!ATTLIST chapter id ID #IMPLIED>
<!ELEMENT section (title,(subsection|para)+)>
  <!ATTLIST section id ID #IMPLIED>
<!ELEMENT subsection (title,(subsubsection|para)+)>
  <!ATTLIST subsection id ID #IMPLIED>
<!ELEMENT subsubsection (title,para+)>
  <!ATTLIST subsubsection id ID #IMPLIED>
<!ELEMENT para (%data;)*>
<!ELEMENT url EMPTY>
  <!ATTLIST url href CDATA #REQUIRED
                name CDATA #IMPLIED
                bookmark          (yes|no)          "yes">
```

Annexe 5 : Le plan de classification [Extrait]

Technique

- Programmation
  - C
  - C ++
  - Emacs
  - Java
  - Perl
  - PHP4
  - SQL
  - XML
  - …
- SGBD
  - PostgreSQL
  - MySQL
  - Oracle
  - Ms Access
  - …
- Sécurité
  - Virus
  - Cryptage
  - Anti-virus

- Pare-feu
- FAQ

Réseau

- Sauvegarde
- Serveur
- Cluster
- Informations
- Haute-disponibilité
- Ethernet
- Stockage

Système

- Linux
- UNIX
- Windows

Messagerie

- SYMPA
- Postfix
- Outlook

Internet

- ZOPE
- FAQ
- HOWTO

Administratif

Infos générales

- Plannings
- Procédures de commandes
- Infos
- Inventaire

FAE

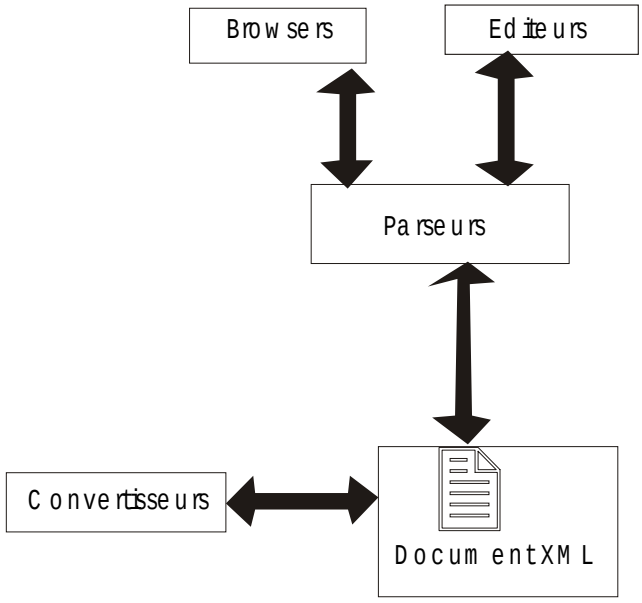
- Formateurs
- Elèves

- Plannings
- Cours

Technique CRI

- Stats CRI
  - Stats Web
  - Stats FTP
  - Stats proxy
  - Stats réseau
  - Stats connexions
- Réseau
  - News
  - Câblage
  - Electricité
  - Infos
  - Routeurs
  - Noms de domaine
  - Site d'Archamps
  - Constructeurs
  - Etat du réseau
  - Clusters
  - Adresses MAC
- R&D
  - Doc technique
  - Packages Debian
  - Etude
  - Développements
  - Doc sur les développements
  - Infos
- PingOO
  - Développements PingOO
  - Doc PingOO

Annexe 6 : Les outils XML



Le **browser** permet la visualisation d'un document XML et la navigation à l'intérieur de ce document.

L'**éditeur** permet la création interactive de documents XML.

Le **parseur** effectue l'analyse syntaxique du document XML et enregistre la construction de la structure du document XML (définie par la DTD).

Le **convertisseur** (aussi appelé **générateur**)permet de traduire automatiquement en XML des documents aux formats hétéroclites. Cet outil est utilisé pour traiter de grands volumes de données.

Exemples d'outils :

- Browsers : **Internet Explorer** (*Microsoft*), **Mozilla** *(The Mozilla Team)*
- Editeurs : **Xemacs, Visula XML** *(Pierre Morel),* **Amaya** *(Worl Wide Web Consortium)*

- Parseurs : **XML Parser for Java** (*IBM*), **libxml** *(Daniel Veillard)*, **XDOM** *(Open XML)*, **Xerces C++** *(The Apache XML Project)*
- Convertisseurs : **XML Generator** (*DataChannel*), **DB2XML** *(Volker Turau)*, **Some2XML** *(Paul Tchistopolskii)*

Annexe 7 : La structuration d'un document XML

Un document XML est structuré en 3 parties :

La première partie, appelée **prologue** permet d'indiquer la version de la norme XML utilisée pour créer le document (cette indication est obligatoire) ainsi que le jeu de caractères (en anglais encoding) utilisé dans le document (attribut facultatif, ici on spécifie qu'il s'agit du jeu ISO8859-1, jeu LATIN, pour permettre de prendre en compte les accents français). Ainsi le prologue est une ligne du type :
<?xml version="1.0" encoding="ISO8859-1"?>
Le prologue se poursuit avec des informations facultatives sur des instructions de traitement à destination d'applications particulières.

Le second élément est une déclaration de type de document (à l'aide d'un fichier annexe appelé DTD - Document Type Definition)

```
<!DOCTYPE identite SYSTEM "/usr/share/tools/dtd/identite.dtd" [
<!ENTITY identite (nom, prenom, date_naissance?)>
<!ENTITY nom (#PCDATA)>
<!ENTITY prenom (#PCDATA)>
<!ENTITY date_naissance (annee, mois)>
<!ENTITY annee (#PCDATA)>
<!ENTITY mois (#PCDATA)>
]>
```

Et enfin la dernière composante d'un fichier XML est l'arbre des éléments comme par exemple :

<identite>
<nom>Valjean</nom>
<prenom>Jean</prenom>
<date_naissance>
<annee>1842</annee>
<mois>Juillet</mois>
</date_naissance>
</identite>