Annotation et recommandation collaboratives de documents selon leur qualité

Laure Berti-Equille

IRISA, Campus de Beaulieu
35042 Rennes, France
Laure.Berti-Equille@irisa.fr

RÉSUMÉ. *Au centre des problématiques de la recherche et du filtrage d'information, la pertinence des réponses fournies par un système conditionne son efficacité. Notre article étend l'évaluation de la pertinence à la prise en compte de la qualité des documents dans un contexte d'annotation et de filtrage par collaboration. Nous proposons une modélisation des documents qui inclut des méta-données décrivant la qualité des documents selon différents critères objectifs et subjectifs. Les critères subjectifs sont définis et évalués par un groupe d'annotateurs collaborant dans le but d'atteindre un consensus sur l'appréciation des documents. Lors du filtrage ou d'un épisode de recherche d'information, la sélection des documents est menée à la fois sur le contenu objectif des documents et sur les aspects qualitatifs. Nous proposons d'associer au calcul de pertinence, un algorithme de recommandation des documents permettant une sélection multicritère et adaptative des documents qui prenne en compte le problème de la révision dynamique des annotations. Pour valider nos propositions, nous avons développé un serveur de documents XDARE (XML-Documents Annotation and Recommendation Environnement) permettant l'annotation et la recommandation collaboratives de documents XML pour une approche mixte de recherche et de filtrage d'information.*

ABSTRACT. *As a key problem of the information retrieval and filtering, the relevance of the results provided by a system conditions its effectiveness. Our article extends the evaluation of information relevance to the taking into account of document quality in a context of collaborative annotation and filtering. We propose a document modelling which includes metadata describing quality according to various objective and subjective criteria. The subjective criteria are defined and evaluated by a group of annotators collaborating with the aim of reaching a consensus on the annotation of documents. The selection of documents is carried out simultaneously on the objective contents of documents and on the qualitative aspects evaluated by the annotators. We propose an algorithm for recommending documents, that is, adaptively selecting documents with taking into account the problem of dynamic revision of annotations. To validate our proposals, we developed the document server XDARE (XML-Documents Annotation and Recommendation Environment) for collaborative annotation and recommendation of XML documents.*

MOTS-CLÉS: *qualité de document, annotation, recommandation, recherche, filtrage d'information.*

KEYWORDS: *document quality, annotation, recommendation, information retrieval, filtering.*

1. Introduction

Orthogonales, la recherche et le filtrage d'information diffèrent de par leur finalité respective, leur mode d'utilisation, leurs modèles théoriques sous-jacents, ou encore le mode d'expression de ce qui est (ou n'est pas) recherché [BC92]. Néanmoins, ces deux problématiques se rejoignent sur le problème de la pertinence des réponses renvoyées par leurs systèmes [Sch91][Sch93].

Dans le cas de la recherche d'information, la méthode générale consiste à formuler, en un épisode de recherche unique, l'expression de ce qui est recherché dans le langage de requête du système, puis de la comparer aux termes d'indexation représentant les textes. La technique de recherche est basée soit sur le principe d'un appariement exact (modèle booléen), soit sur celui d'un appariement optimal (modèle de type espace vectoriel ou probabiliste) avec pondération (en *tf.idf*, par exemple) des termes recherchés avec les mots-clés des textes indexés. Il s'agit de proposer au final un ordonnancement des textes selon une probabilité de pertinence par rapport à la requête. Malgré les nombreuses techniques d'expansion automatique des requêtes [XC96], cette méthode connaît des problèmes d'adéquation entre l'expression des besoins en information et les requêtes (en particulier, de par les contraintes posées par le langage de requêtes), et ce,. Les textes retrouvés sont ensuite évalués par l'utilisateur. Cette évaluation entraîne souvent une modification de la requête et son raffinement progressif par un processus de « *relevance feed-back* » (réinjection de nouveaux critères de pertinence découverts lors de l'étape précédente).

Dans le cas du filtrage d'information, l'utilisateur exprime ce qui est recherché et ce qui ne l'est pas dans un flux dynamique d'informations en utilisant une représentation de ses centres d'intérêt relativement stables sur le long terme (profil). La technique de filtrage est basée sur des modèles de graphes d'inférence ou des arbres de décision bayésiens pour le calcul de probabilités sur la comparaison d'une nouvelle information avec plusieurs profils d'utilisateurs. La méthode repose sur l'hypothèse que les profils sont des spécifications correctes des centres d'intérêt ou des préférences des utilisateurs.

Deux notions de pertinence se sont jusqu'alors classiquement distinguées [Den97] : la pertinence par rapport à un sujet *(relevance to a subject)* et la pertinence pour l'utilisateur. La première repose sur la relation thématique *(topicality)* entre l'information recherchée et l'information résultat : « orientée système », elle dépend directement de la correspondance faite par le système entre les termes de la requête et les termes qui indexent le document. La seconde, « orientée utilisateur » repose sur la décision que prend l'utilisateur d'accepter ou de rejeter l'information présentée. Différents facteurs de pertinence (regroupés en 24 catégories) ont été identifiés dans [Sch91] [Sch93] [Bar94]. Les indices de pertinence traditionnellement utilisés sont de nature thématique. Et il est fréquent que le critère thématique soit modélisé de manière quantitative, par comptage des occurrences du terme de la requête dans le document indexé et sans nuances qualitatives.

Or, les caractéristiques qualitatives des documents permettraient, si elles étaient exploitées, de raffiner la qualité des réponses fournies par le système de recherche ou de filtrage d'information, notamment, en considérant et pondérant des critères relatifs à la qualité des documents (fraîcheur d'un texte, fiabilité ou précision du contenu, par exemple).

Notre approche consiste à étendre l'évaluation de la pertinence d'un document par la prise en compte de sa qualité dans un contexte d'annotation et de filtrage par collaboration. Nous proposons une modélisation des documents incluant des méta-données représentant la qualité des documents sous différents critères objectifs et subjectifs. Les critères objectifs peuvent être quantitatif et calculés par des méthodes statistiques. Les critères subjectifs sont définis et évalués par un groupe d'annotateurs collaborant dans le but d'atteindre un consensus sur l'annotation des documents. La sélection des documents est menée à la fois sur le contenu objectif des documents comme sur les aspects qualitatifs évalués par les annotateurs. Notre objectif est de proposer une recommandation multicritère des documents, c'est-à-dire raffiner la sélection classique des documents en exploitant les méta-données qui leur sont associées, soit automatiquement, soit en tant que résultat d'une collaboration entre individus. Dans le cadre d'un filtrage social, nous considérons que la révision dynamique des appréciations est un phénomène naturel à prendre nécessairement en compte dans la conception d'un système. A partir de ces spécifications, nous avons développé le système *XDARE (XML-Documents Annotation and Recommendation Environment)* pour l'annotation et la recommandation de documents XML.

La suite de l'article s'organise de la façon suivante : la section 2 présente un état de l'art sur la modélisation de l'utilisateur, les systèmes de recherche, de filtrage et de recommandation d'information. La section 3 décrit la modélisation des documents et de leurs méta-données de qualité. La section 4 présente les principales opérations mises en œuvre dans le processus de recommandation des documents par collaboration dynamique. La section 5 décrit les fonctionnalités et l'architecture du système *XDARE* et illustre notre approche par un exemple d'annotation et de recommandation collaboratives de documents XML. Enfin, la section 6 conclut l'article en présentant nos perspectives de recherche et de développement.

2. État de l'art

2.1. Modélisation de l'utilisateur : profils et préférences

En filtrage d'information, très tôt est apparu le constat de l'insuffisance des modèles d'utilisateur jugés trop canoniques pour une communauté d'utilisateurs très hétérogène : des modèles d'utilisateur individualisés ont donc été préconisés [Ric83]. [BT94] ont analysé les problèmes de modélisation de l'utilisateur. Ils ont proposé l'architecture générale d'un système à base de connaissances : le système *IR-NLI II*

comprenant une interface experte d'assistance à l'utilisateur pour la recherche d'information en ligne *UM-tool*. Le projet *IFT (Information Filtering Tool)* [ADT97] construit et gère la représentation des préférences de l'utilisateur pour le filtrage d'information en utilisant des techniques de modélisation de l'utilisateur. Ce projet comprend trois prototypes : *IFTool*, *PIFT* et *iFWeb*. *IFTool* et *PIFT* sont caractérisés par deux algorithmes d'appariement utilisés pour évaluer la pertinence de nouveaux documents avec le modèle de préférences de l'utilisateur (*UMT : User Modeling Tool)*. Le système *ifWeb* [AT97], dédié à la navigation sur l'Internet, est un prototype d'agent intelligent basé sur le modèle utilisateur. Il permet d'assister l'utilisateur dans sa navigation sur le Web en utilisant le *feed-back* implicite (positif et négatif) qui permet de mettre à jour ou de raffiner la modélisation de l'utilisateur. Les stratégies de navigation autonome mises en oeuvre dans *ifWeb* sont basées sur la notation du degré pour lequel un lien hypertextuel est considéré comme prometteur pour accéder à d'autres documents pouvant être pertinents par rapport aux centres d'intérêt de l'utilisateur.

En 1995, Yao [Yao95] a décrit de manière formelle les jugements de l'utilisateur et la notion de préférences pour la représentation, l'interprétation et la mesure de la pertinence des documents. Il utilise une échelle ordinale de pertinence et suggère une nouvelle méthode de mesure de performances pour les SRI basée sur la distance entre l'ordonnancement de l'utilisateur et celui fait par le système.

De nombreux systèmes intègrent aujourd'hui les techniques d'apprentissage automatique pour la création de profils d'utilisateurs pour le filtrage. Avec son système de filtrage de News, *NewWeeder*, [Lan95] compare les performances des techniques d'apprentissage aux techniques traditionnellement utilisées pour le filtrage utilisant la formule *tf.idf*. Un vecteur de fréquence de termes est d'abord créé sur les nouveaux articles puis, utilisé pour entraîner un algorithme d'apprentissage basé sur le principe de la longueur de description minimale (*MDL : Minimum Description Length)*. Les résultats ont indiqué que les techniques d'apprentissage présentent beaucoup d'avantages pour l'identification d'articles de *News* intéressants.

[RB96] présente un mécanisme pour l'acquisition non intrusive des préférences de l'utilisateur pour le filtrage d'information dans le contexte d'un service multimédia interactif de télé-shopping et de vidéo à la demande *(VOD : Video On Demand)*. Les préférences de l'utilisateur sont apprises implicitement par le système par observation des habitudes de sélection ou de réactions à un choix (sollicitées ou volontaires). Selon les différents attributs proposés pour les films (genre, producteur, directeur, année...), deux profils de préférences sont composés : l'un positif, avec une note entre 0 et 10 (de l'indifférence au grand enthousiasme) et l'autre négatif, avec une note de -10 à 0 (du rejet total à l'indifférence). Chaque note possède un degré de certitude qui est construit sur la base des sélections de l'utilisateur. La création des profils est menée selon une approche d'heuristiques statistiques qui crée un profil stéréotype et un profil de l'utilisateur dont les goûts sont similaires.

2.2. Agents assistants et systèmes de recommandation collaboratifs

De nombreux systèmes utilisant des agents ont été développés pour le filtrage des informations telles que les *mails* ou les articles *Usenet* [Bac92] [EBGP96] [PE97]. Ils utilisent également des techniques d'apprentissage ou d'adaptation [Bac92] pour améliorer la qualité de leur assistance au cours du temps. Si l'agent est un assistant, la connaissance du domaine d'application et/ou de l'utilisateur lui est nécessaire et deux approches sont traditionnellement utilisées pour lui fournir cette connaissance: (1) la première méthode, la plus commune nécessite que les utilisateurs fournissent leurs propres règles en utilisant un langage de scripts (voir les systèmes *Information Lens System* [MGT+87] et *IRA* (*Information Retrieval Agent*) [Voo94]), (2) la seconde méthode utilise les techniques traditionnelles d'ingénierie de la connaissance pour identifier la connaissance de fond sur l'application et celle sur l'utilisateur.

Une solution alternative est de construire un profil qui reflète les préférences de l'utilisateur quand il utilise une application particulière (un *browser*, par exemple). Une multitude de mécanismes d'apprentissage ont été utilisés par les agents pour induire de tels profils : des algorithmes génétiques ou encore des algorithmes d'induction de règles symboliques ont notamment été employés par les agents filtrant les *News*. [HKW95] présentent un système de filtrage cognitif (*CIFS : Cognitive Information Filtering System*) permettant la sélection de messages électroniques. Il utilise une population d'agents en compétition avec une adaptation des algorithmes génétiques pour l'apprentissage du *feed-back* et du comportement de l'utilisateur. Des approches basées sur les instances (dont celle du système *MAGI -Mail Agent Interface-* [EBGP96]) permettent de dériver une classification en comparant une nouvelle instance (*mail*, *news* ou document) avec les instances précédemment classifiées [PE97].

Parallèlement, les systèmes de recommandation collaboratifs [RV97] représentent une alternative intéressante aux techniques de recherche d'information traditionnelles, car ils travaillent en identifiant les plus proches voisins d'un utilisateur dans l'espace de ses précédentes appréciations (par exemple, appréciation d'un site Web, d'un domaine musical ou cinématographique, etc.). Ils lui recommandent finalement ce que ces voisins ont évalué comme étant "le meilleur". Dans cette tendance, *Fab* [Bal97] est un service de recommandation de pages *Web* basé à la fois sur le contenu des pages et sur une recommandation collaborative entre utilisateurs. Plutôt que les mesures de rappel et de précision utilisées classiquement en recherche d'information ou bien celles utilisées en apprentissage pour la classification, *Fab* utilise la mesure de performance basée sur la distance normalisée développée par Yao [Yao95] qui distingue les jugements relatifs des jugements absolus.

Par opposition, *PICS (Platform for Internet Content Selection)* [RM96] est une infrastructure qui associe des labels au contenu des documents disponibles sur l'*Internet*. A l'origine conçu pour aider les parents et les professeurs à contrôler la navigation de leurs enfants/élèves sur le *Web*, il permet d'affecter n'importe quel

critère sur des labels que le système interprète pour autoriser ou bloquer l'accès à un document. Complémentaire à la recommandation, l'intérêt de cette approche est que la structure d'un document peut être enrichie de labels qui conditionneront sa (non-)diffusion.

Le Tableau 1 présente des systèmes développés pour la recherche (SRI) et le filtrage (SFI) d'information ainsi que ceux employant des agents collaboratifs (Agents).

SYSTÈME	*TYPE*	*MODÈLE*	*MODE D'INTERACTION AVEC L'UTILISATEUR ET PARTICULARITÉS*
INQUERY [Cro95] [XC96]	SRI	Modèle probabiliste	Requêtes et expansion de requêtes
Système TREC classique	SRI	Modèle probabiliste	Requêtes et expansion de requêtes
Datacycle [HGLW87]	SFI		Requêtes floues + SQL
InRoute [Cal96]	SFI	Modèle probabiliste	Réseau de requêtes spécifiées par l'utilisateur
PFIT, *ifWeb* [ADT97][AT97]	SFI	Réseaux bayésiens	Dialogue basé sur des requêtes
Information Lens System [MGT+87]	SFI		Construction par l'utilisateur de templates à partir de règles
Syskill & Webert [PMB97]	FI Agents	Classification bayésienne	Classification Apprentissage de profils concernant l'intérêt que porte un utilisateur à certaines pages Web suggestion de liens
Amalthaea [Mou96]	FI Agents Discovery Agents	Modèle économique	Ecosystème d'agents
FAB [RV97][Bal97]	FI Agents		Service de recommandation hybride

Tableau 1. *Agents, systèmes de recherche(SRI) et de filtrage d'information (SFI)*

De ces différents travaux de recherche, nous avons retenu plusieurs principes qui guident notre approche :

- pour améliorer la qualité d'un résultat, il faut évaluer la qualité des documents et informations donnés en entrée au système et l'exploiter dans le traitement de la requête,
- il est nécessaire que la définition de la qualité des documents et des informations soit flexible et l'utilisation de labels et de méta-données permettent cette flexibilité,
- notre approche se veut mixte et modulaire, c'est-à-dire applicable à la fois à la recherche et au filtrage d'information.

3. Modélisation des documents et des méta-données

3.1. Documents et méta-données

Nous modélisons les document selon la structure suivante (Figure 1) : un document se compose d'une à plusieurs version(s) constituée(s) chacune d'une à plusieurs parties (appelées block) caractérisées par des mots-clés éventuels, un contenu (value) et un ensemble de méta-données (metadataset). Un ensemble de méta-données peut être associé à un document, une version ou une partie de document.

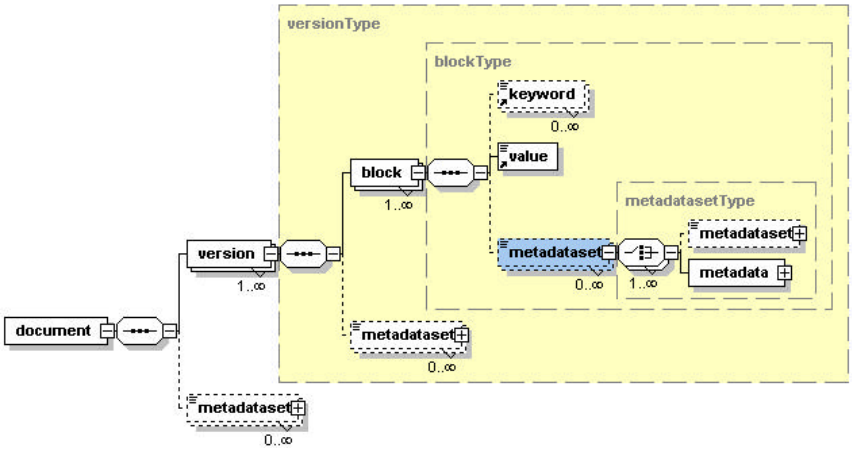

Figure 1. *Schéma XSD représentant la structure d'un document*

Sur la base de plusieurs travaux de spécification de méta-données tels que : (i) *Bib-1* [NISO01], décrivant l'ensemble des attributs bibliographiques de la norme *Z39.50* (ANSI/NISO) (ii) *Dublin Core* [WGMD95], système développé, à l'origine, pour décrire, au moyen de 15 attributs, les caractéristiques essentielles des documents en-ligne (iii) *STARTS* [GGM97], système utilisé, notamment pour l'annotation collaborative de documents et son modèle de méta-données, *MBasic-1*, reprenant un certain nombre d'attributs de *Bib-1* et de *GILS* (*Government Information Locator Service*) [Chr96], nous proposons un ensemble d'attributs (Tableau 2) pour caractériser les documents.

Nous distinguons les méta-données associées à tout ou partie du document selon deux catégories :

- les méta-données indépendantes du contenu qui capturent les informations telles que la date de modification, la localisation d'un document,

- les méta-données dépendantes du contenu : comme, par exemple, la taille d'un document. Cette catégorie se subdivise elle-même en deux :
  - les méta-données basées directement sur le contenu telles que se définissent les descripteurs d'images ou les indices *full-text*, les fichiers inversés et les vecteurs d'indices d'un document,
  - les méta-données descriptives de contenu comme les annotations décrivant le contenu d'un texte par un ensemble de mots-clés.

Intitulé	***Label***	***Définition***
Identificateur du document	DID	Chaîne de caractères unique non nulle utilisée pour identifier le document
Titre du document	TITLE	Intitulé du document attribué par son auteur ou son éditeur
Niveau de confidentialité	CONFIDENTIALLEVEL	Entier compris entre 0 et 3 signifiant respectivement diffusion publique du document (0), restreinte (1), confidentielle (2), secrète (3)
Auteur, Créateur ou Producteur	CREATOR	Personne ou organisation responsable de la création initiale du document, de son contenu (par exemple les auteurs du texte, artistes ou photographes pour les images)
Editeur ou émetteur	PUBLISHER	Organisme responsable de la diffusion du document
Date de création	CREATIONDATE	Date à laquelle le document a été créé (JJ-MM-AAAA)
Date de publication	PUBLICATIONDATE	Date à laquelle le document a été mis à disposition (JJ-MM-AAAA)
Catégorie du document	CATEGORY	Catégorie de document (article de recherche, rapport technique, ...)
Type du document	DOCTYPE	Booléen signifiant que le document est composite (1) ou standard (0), c'est-à-dire composé (ou non) de parties avec différents niveaux de confidentialité
Type de fichier	MIMETYPE	Chaîne de caractères représentant le type MIME du document
Taille du document	SIZE	Taille du document en Koctets
Rayonnement hypertextuel	LINKAGE	Ensemble de liens référencés dans tout ou partie du document
Langue	LANGUAGE	Langue du contenu du document

Tableau 2. *Attributs caractérisant un document*

3.2. Qualité de documents

La qualité d'un document est mesurable en combinant des mesures quantitatives (à partir des informations liées directement ou indirectement au contenu du document) et des mesures qualitatives évaluées de façon subjective par un (ou plusieurs) relecteur(s).

Comme l'illustre la Figure 2, le type de l'ensemble des méta-données associées à un document (metadatasetType) peut être un ensemble de méta-données (metadataset) ou une méta-donnée (metadata) qui se compose d'un critère, d'une notation datée représentant l'évaluation du critère faite par un générateur (generator) pouvant être un humain (annotator) ou un programme (program) et d'un commentaire éventuel (comment). Un consensus peut être éventuellement calculé pour un critère et une date donnés si plusieurs notations ont été proposées. L'instanciation des critères de qualité d'un document peut être basée sur une adaptation de la liste non exhaustive de critères donnée dans [Ber99] ou parmi les propositions de [WSF95] [Ric96] [CPPR98]. Le projet *DESIRE* [HBP00] a également produit une liste détaillée de critères de qualité à utiliser pour la sélection des ressources du *Web* répartis en différentes catégories : 1) les critères liés à la politique de diffusion et à la portée de la ressource, 2) les critères liés au contenu, 3) les critères liés à la forme, 4) les critères de gestion de la collection de documents.

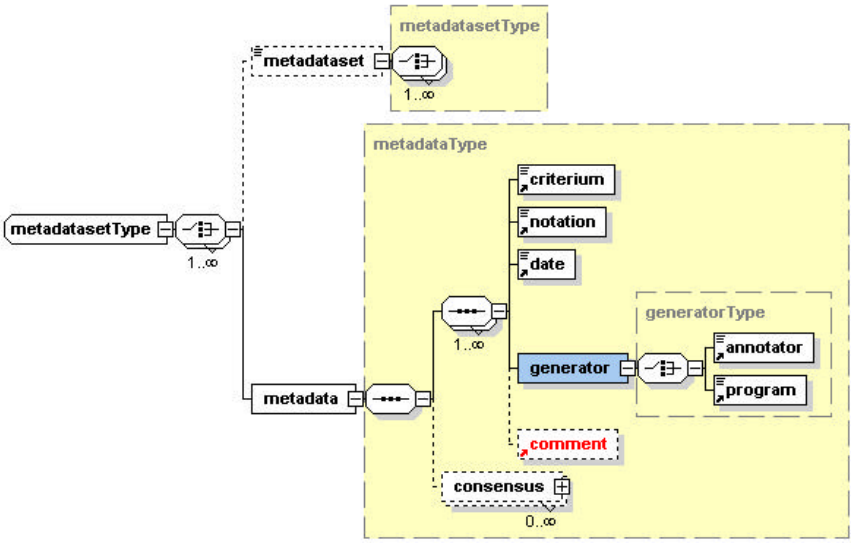

Figure 2. *Schéma XSD représentant les méta-données associées à un document*

Par exemple, dans la catégorie les critères liés au contenu, ceux qui concernent la validité du document sont : l'autorité et la réputation de ses auteurs, la teneur, l'exactitude, l'organisation du document et plus généralement de la ressource hypermédia. Pour évaluer la validité d'un document, une série de questions, de conseils et de contrôles est proposée dans *DESIRE* et peut être employée pour discerner si la ressource répond bien au critère de validité : jusqu'à quel degré la teneur de l'information est-elle valide ? Pourquoi l'information est-elle présentée ? Quelle était la motivation du producteur d'information quand il l'a rendue disponible ? Avait-il un motif caché ? La teneur de la ressource est-elle vérifiable ? etc.

Ces choix de modélisation pour les documents, les méta-données et la qualité ont pour objectif de : 1) permettre une définition rigoureuse mais flexible de chaque dimension de la qualité d'un document, ainsi que son mode de mesure sur la base d'une méthode scientifique programmable, 2) pouvoir réutiliser les standards de méta-données proposés dans la littérature incluant notamment des éléments de qualité, 3) proposer une assistance à l'utilisateur pour qu'il définisse et évalue lui-même la qualité des documents qu'il manipule en lui fournissant, notamment, une bibliothèque de programmes pour la génération de méta-données, pour le calcul des différents critères de qualité ou pour celui d'un consensus.

Si la définition de la qualité d'un document reste flexible, nous avons choisi, pour les besoin de notre application, de la décomposer selon quatre catégories de méta-données [Ber99] adaptées au filtrage collaboratif de documents :

- la première catégorie (ManagementMD) regroupe les méta-données associées à la gestion et l'administration des documents par le système. Elle comprend des critères mesurables tels que la confidentialité, la sécurité, l'accessibilité.
- la seconde catégorie (RepresentationMD) regroupe les méta-données associées à la représentation du document par le système. Dans notre contexte, la technique d'indexation est une méta-donnée, de même que le fait d'utiliser (ou non) les méta-données pour indexer les documents.
- la troisième catégorie (ContentMD) regroupe les méta-données associées directement au contenu du document (la langue utilisée dans le texte, le nombre de liens hypertextuels, ...).
- la quatrième catégorie (RelativeMD) regroupe les méta-données liées à la qualité relative des documents par rapport au contexte d'usage des documents (selon un référentiel temporel ou applicatif), ou par rapport au référentiel social, culturel, cognitif ou affectif (appréciations, préférences) des utilisateurs.

Chaque catégorie regroupe un ensemble de critères objectifs mesurables que nous avons définis et implantés au sein d'un serveur de documents. Les critères subjectifs de qualité d'un document sont évalués à une date donnée par un (ou plusieurs) annotateur(s). Le résultat de ce jugement est stocké sous la forme d'une méta-donnée liée au document comme le présente l'exemple de la Figure 3. Dans cet exemple, les méta-données associées au document XML "MonArticle" décrivent trois

critères représentant la qualité du document : son originalité, son impact industriel et scientifique et sa présentation. Ces critères ont été évalués par deux annotateurs A1 et A2 à différentes dates (voir la sortie XSL synthétisant les méta-données de qualité).

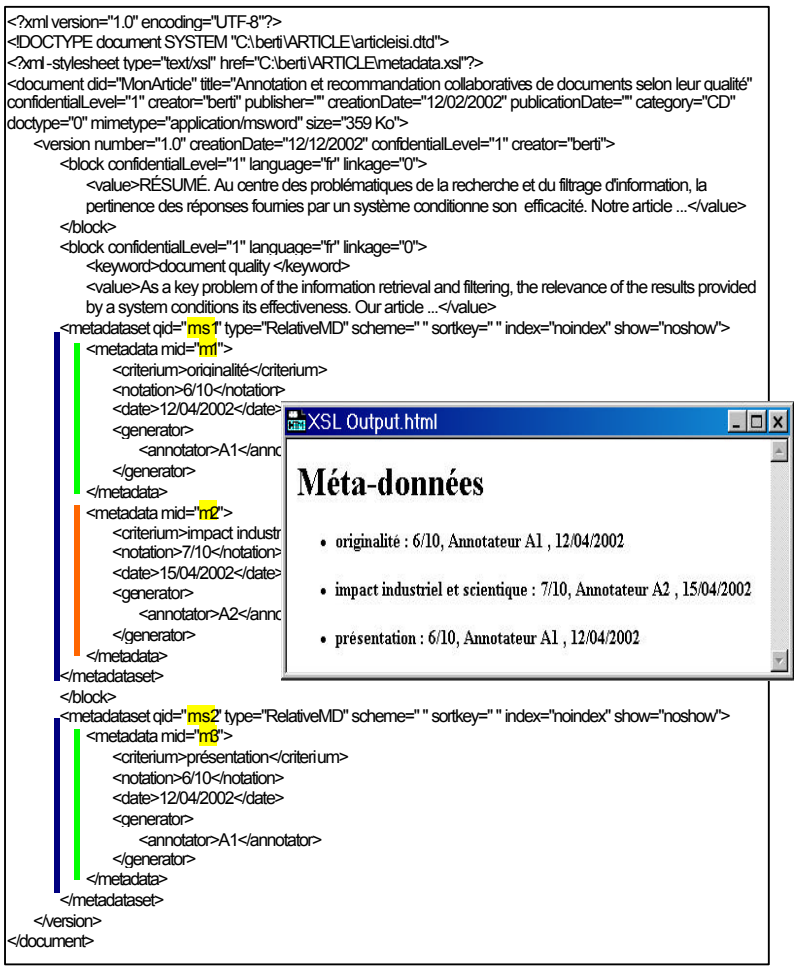

Figure 3. *Exemple de document avec les méta-données représentant sa qualité*

Notre objectif est ici de proposer à l'utilisateur (annotateur ou utilisateur en recherche d'information) les moyens de spécifier, choisir et configurer lui-même un ensemble cohérent de méta-données (critères objectifs et subjectifs) reflétant sa "vision" personnelle (ou "corporative") de la qualité d'un document quel que soit le

mode d'accès à l'information par recherche ou par filtrage. Le calcul d'un score de pertinence thématique par rapport aux mots-clés ou à un profil est un exemple de critères objectifs et quantitatifs. Les appréciations des documents par collaboration d'individus évaluateurs constituent des méta-données subjectives. C'est sur la base d'une combinaison de ces deux types de méta-données que seront sélectionnés les documents à la fois les plus pertinents et de meilleure qualité, c'est-à-dire selon la définition ISO8402 : "ayant l'ensemble des caractéristiques qui leur confère (ainsi qu'au service de RI ou FI), l'aptitude à satisfaire les besoins exprimés ou potentiels des utilisateurs à une date donnée".

3.3. Processus d'annotation et de recommandation collaboratives

Le processus d'annotation et de recommandation collaboratives des documents peut être décomposé en 6 opérations (Figure 4) : 1) la génération de méta-données (annotating), 2) l'indexation des documents (indexing), 3) l'appariement entre la requête (ou le profil) et les représentations des documents indexés (matching), 4) la fusion des documents et/ou annotations (merging), 5) l'ordonnancement des documents (scoring) et 5) la révision et mise à jour des méta-données (refreshing). Chacune de ces étapes constitue une opération élémentaire (ou primitive) menée sur les documents, et elle peut être représentée par un graphe orienté avec, en entrée, la collection de documents et, en sortie, un ensemble de méta-données associé et la liste des documents recommandés. La combinaison de ces opérations élémentaires s'inscrit à la fois dans un processus de recherche d'information ou dans celui de filtrage, comme l'indique la Figure 4.

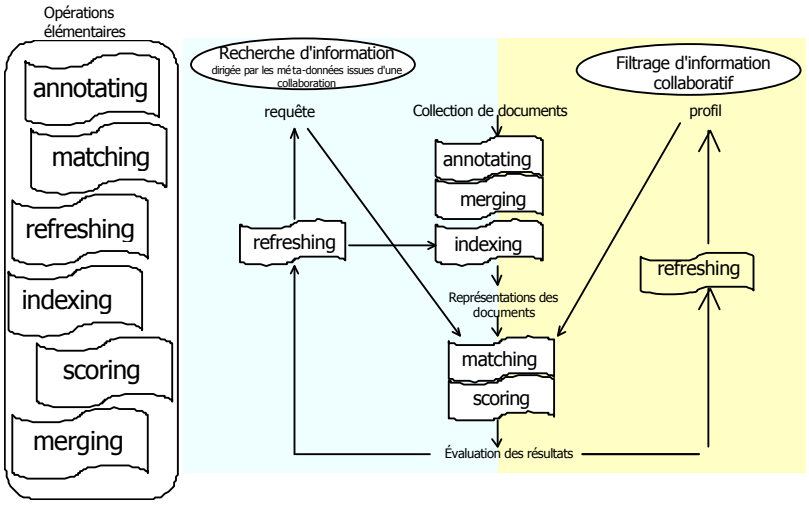

Figure 4. *Opérations pour la recommandation collaboratives en RI et FI*

Pour préserver la modularité de notre approche, nous proposons un langage déclaratif extensible pour spécifier ces opérations élémentaires. La déclarativité de notre langage s'inspire de SQL pour garantir un déploiement et une maintenance faciles. Néanmoins, notre langage de spécification n'est pas complètement déclaratif puisque du code en langage procédural est utilisé dans certaines des opérations. Le langage peut donc être étendu pour permettre que : i) les opérations proposées puissent invoquer des fonctions externes spécifiques pouvant être ajoutées à la bibliothèque prédéfinie de fonctions, ii) les opérations puissent être combinées afin d'obtenir un programme de recommandation plus complexe, iii) l'ensemble des fonctions et algorithmes proposés peut être étendu si besoin.

Notre langage de spécification permet à l'utilisateur, pour son épisode de recherche ou au cours du filtrage des informations, d'associer implicitement ou explicitement l'exécution d'un programme de recommandation particulier à ses exigences en termes de qualité de résultats attendue (profil qualité).

```
<programme de recommandation de documents> ::=
        (<déclaration des types complexes>)*
        (<codage des fonctions et algorithmes>)*
        (<déclaration des documents en entrée >)+
        (<définition des opérations élémentaires >)+

(<définition des opérations élémentaires >)::=
        <CREATE> (<ANNOTATING> | <MERGING> | <SCORING> |
                        <MATCHING> | <REFRESHING> | <INDEXING>)
        (<FROM> <INPUT RAW DOCUMENT FLOW> +) |
                (<USING> <INPUT DOCUMENT FLOW>) |
                (<ON> <INPUT RAW DOCUMENT FLOW> [ <REFERENCING> <DOC_IDENTIFIER>]
        (<FOR EACH> (<ATTRIBUTE NAME> | <ELEMENT NAME>)
        [<WHEN> <CONDITION> ])
        [<LET> ( <VARIABLE> = <EXPRESSION>(, <EXPRESSION>)*)+]
        [<BY> <ALGORITHM> | <FUNCTION>]
        [<WHERE> <PREDICATE> ( <AND> | <OR> <PREDICATE>)*]
        ( { <SELECT> (<VARIABLE> | <ATTRIBUTE NAME> | <ELEMENT NAME>)+
        [<INTO> <OUTPUT DATA FLOW>]
        (<CONSTRAINT> (<EXPRESSION> <FUNCTIONAL_DEPENDENCY> <EXPRESSION>) |
                (<CHECK> <EXPRESSION>))*
        } )*
```

Figure 5. *Syntaxe d'un programme de recommandation*

Le processus d'annotation et de recommandation peut être représenté sous la forme d'un diagramme d'état-transition UML pour une collection de documents convertis en XML (Figure 6). Celle-ci est d'abord annotée selon les spécifications de méta-données retenues en mode RI et FI. Les méta-données peuvent être révisées et mises à jour régulièrement selon la politique de rafraîchissement définie. Les documents sont au final recommandés selon leur pertinence thématique et sur leur qualité. Les documents sont indexés. L'indexation peut être complétée par l'ajout de méta-données et des résultats de recommandation.

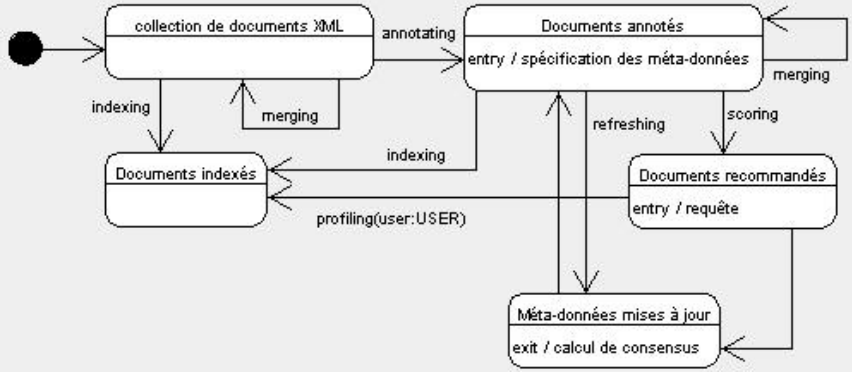

Figure 6. *Processus d'annotation et de recommandation centré document*

4. Recommandation de documents XML

Dans cette section, nous décrivons les principales fonctions d'un système mixte de recherche et de filtrage d'information incluant l'annotation et la recommandation des documents selon leur qualité. Ces fonctions correspondent aux opérations élémentaires citées précédemment. Nous nous intéresserons ici en particulier aux opérations d'indexation des documents, de fusion, de recommandation et de révision.

Dans notre application, nous manipulons des documents préalablement convertis en documents XML et XHTML, rendus valides pour la DTD donnée en Figure 11. Le langage *XPath* définie par le *W3C* permet de décrire un modèle de chemin dans l'arbre d'un document XML et de sélectionner l'ensemble des nœuds que l'on peut atteindre en suivant, à partir d'un nœud donné, tous les chemins conformes à ce modèle. Nous employons l'arbre d'un document XML construit de la façon suivante:

- il possède une racine qui a pour fils unique l'élément racine du document,
- une feuille est associée à chaque attribut et à chaque texte,
- à chaque élément est associé un nœud non terminal qui a pour fils les nœuds associés à ses composants immédiats (attributs, textes et éléments).

4.1. Indexation des documents

On souhaite retrouver des documents par le noms de leurs éléments, de leurs attributs, par leur structure ou leur contenu. Pour cela, le système doit fournit des mécanismes afin de mener efficacement les opérations suivantes :

- pour un nom d'élément donné, trouver une liste d'éléments ayant le même nom groupés par les documents auxquels ils appartiennent,
- pour un nom d'attribut donné, trouver une liste d'attributs ayant le même nom groupés par les documents auxquels ils appartiennent,
- pour un élément donné, trouver ses fils et parent (ou attributs). Pour un attribut donné, trouver son élément de parent.

La structure d'index du système comprend de trois composants principaux : l'index des éléments, l'index des attributs et l'index des structures (Figure 7).

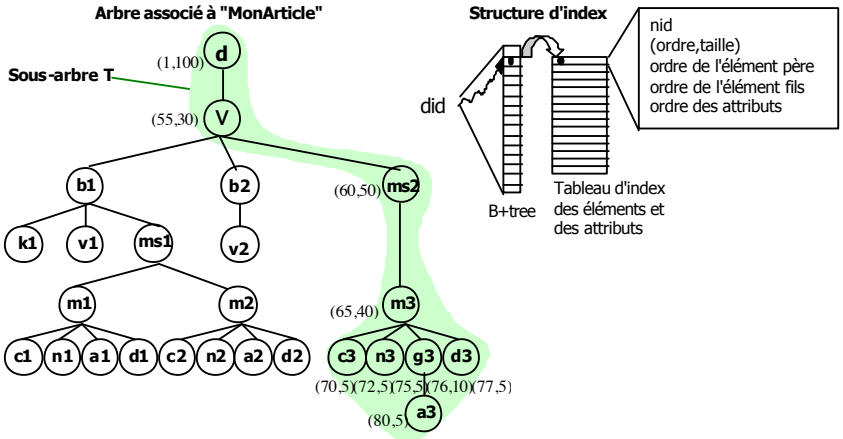

Tableau des éléments, attributs et valeurs pour le sous-arbre T associé à "MonArticle"

Noeud	*Type*	*Nom*	*Attribut*	*Valeur*
d	*élément*	document	did="MonArticle" title="Annotation et recommandation collaboratives de documents selon leur qualité" confidentialLevel="1" creator="berti" publisher="" creationDate="12/12/2002" publicationDate="" category="CD" doctype="0" mimetype="application/msword" size="359 Ko"	
v	*élément*	version	number="1.0" creationDate="12/12/2002" confidentialLevel="1" creator="berti"	
ms2	*élément*	metadataset	qid="ms2" type="RelativeMD" scheme=" " sortkey=" " index="noindex" show="noshow"	
m3	*élément*	metadata	mid="m3"	
c3	*élément*	criterium		présentation
n3	*texte*	notation		6
g3	*élément*	generator		
d3	*texte*	date		12/04/2002
a3	*texte*	annotator		A1

Figure 7. *Structure et indexation d'un document XML*

Deux autres composants sont l'index des noms pour stocker les chaînes de caractères et la table des valeurs. On considère que toutes les valeurs définies dans les documents XML sont des chaînes de caractères de longueur variable. Toutes les chaînes de caractères distinctes sont rassemblées dans l'index des noms, implanté sous forme d'un arbre B+. Chaque chaîne de caractères est identifiée de façon unique par un identifiant de nom (nid) dans l'index des noms. L'utilisation de l'index des noms réduit au minimum le stockage et les coûts en éliminant les chaînes de caractères dupliquées et des comparaisons de chaînes. Pour la même raison, toutes les valeurs de chaînes (c.-à-d. valeur d'attribut et valeur des textes) sont collectées dans la table de valeurs. Un élément ou un attribut peut être identifié de façon unique par son ordre et l'identifiant du document did dans le système. L'index des éléments, l'index des attributs et l'index des structures sont les trois index permettant d'assurer les fonctionnalités énumérées ci-dessus.

L'index des éléments et l'index des attributs sont implantées sous forme d'arbres B+ en utilisant les identifiants de noms (*nid*) comme clefs. Chaque entrée dans un nœud feuille pointe sur un ensemble des enregistrements de longueur constante pour des éléments (ou des attributs) ayant une chaîne de caractère identique, groupés selon le document auquel ils appartiennent . L'index des éléments nous permet de trouver rapidement tous les éléments avec la même chaîne de caractères, ce qui est une des opérations les plus communes pour des requêtes de type expression régulière de chemin. Chaque enregistrement d'élément comprend un couple *<ordre;taille >* défini dans [LM01] de la façon suivante (voir le sous-arbre T dans la Figure 7) :

soient x et y, x est nœud père de y si :

$$taille(x) = ?_y\ taille(y)$$

$$[ordre(y),ordre(y)+taille(y)] \supseteq [ordre(x),ordre(x)+taille(x)]$$

4.2. Fusion d'annotations

Pour l'opération de fusion d'annotations (merging), nous employons la comparaison d'arbres ordonnés étiquetés. La structure des documents peut être décomposée de telle sorte que les étiquettes des arbres représentent : (1) soit des éléments de structure (section, paragraphe, sous-section, titre, figure, lien) (2) soit des méta-données. De cette façon, cette représentation par arbre permet de ne considérer que les aspects structuraux d'un document ou bien sa méta-description. Les éléments de deux documents, de deux versions peuvent être fusionnés et synchronisés si besoin, de même que leurs annotations si plusieurs annotateurs ont procédés à des différentes annotations. Nous employons pour cette tâche l'algorithme 3DM [Lin01] afin d'offrir différents niveaux de fusion des versions de documents et de fusion de leurs méta-données.

4.3. Calcul de score et recommandation des documents

L'objectif de la recommandation des documents selon leur qualité est d'exploiter les méta-données et d'utiliser les évaluations des différents critères pour proposer une sélection des documents qui répondent le mieux à la qualité requise par l'utilisateur lors de son épisode de recherche ou au cours du filtrage d'information. Par exemple, l'utilisateur peut privilégier la fraîcheur d'un document à sa précision.

Il est souvent nécessaire de prendre en compte l'importance relative des critères de qualité les uns par rapport aux autres et, s'il y a lieu, de leur affecter une pondération qui sera utilisée dans la stratégie de recommandation. Dans ce cas, les critères de qualité ne sont pas interchangeables et l'utilisateur devra attribuer un poids à chacun des critères selon l'importance qu'il lui accorde.

L'opération de recommandation (scoring) consiste à calculer, pour chaque document, un score de qualité relativement à une requête et à la collection de documents thématiquement pertinents. Notre objectif est de développer une bibliothèque de fonctions de calcul de scores. A ce titre, nous avons implanté les modèles de calcul décrits dans [FC94] en les adaptant pour la recommandation des documents tels que :

- *l'affectation linéaire de poids (AL)* : cette méthode permet le calcul d'un score de qualité pour chaque document tel que :

$$Scoring(D_i) = \sum_{k=1}^{L} W_k N_{ik} \text{ où } \begin{cases} W_k \text{ est le poids du } k_{ème} \text{ critère} \\ N_{ik}, \text{ l'évaluation du } k_{ème} \text{ critère pour le } i_{ème} \text{ document } D \end{cases}$$

- *l'élimination par aspects (EA) :* cette méthode classe les critères par importance, puis élimine les documents ayant le plus mauvais score de qualité pour le critère le plus important. L'opération est répétée jusqu'à ce qu'il ne reste plus qu'un seul document pour la recommandation ou bien que tous les critères aient été considérés.
- *la méthode d'Anderson (AND)* [And90] : cette méthode utilise trois matrices indiquant chacune la situation relative des documents les uns par rapport aux autres :
  - la matrice contenant la concordance des évaluations, c'est-à-dire qui considère la pondération des critères pour la notation desquels "un document i est meilleur qu'un document j",
  - la matrice de discordance qui considère la pondération des critères pour la notation desquels "un document i est moins bon qu'un document j",
  - la matrice de magnitude des évaluations qui considère "de combien un document i est meilleur qu'un document j ".
- *la méthode de Subramanian et Gershon (SG) :* sur le même principe que la méthode d'Anderson, la pondération des critères de qualité se fait lors de la construction des matrices de concordance et de discordance.

La liste des méthodes utilisables pour la sélection n'est pas exhaustive et la flexibilité de notre approche permet l'ajout d'autres fonctions à la bibliothèque, le choix et l'utilisation de différentes fonctions de sélection pour la recommandation. Celles-ci peuvent être couplées aux calculs classiques de pertinence. Enfin, la recommandation peut être menée sur tout ou partie des documents (recommandation de *blocks* en exploitant les méta-données qui leur sont particulièrement associées). A partir des modèles de calcul de score et pour déterminer quel est le "meilleur" document, nous proposons l'algorithme de recommandation suivant (Figure 8) :

```
Procedure Recommandation(Q : requête, D : collection_Documents, m : methode_Calcul)
/* m : une des méthodes par methode_Calcul = {AL, EA,AND,SG}  */

Déclaration
Soit acc[di] l'accumulateur de score pour chaque document di de la collection
Début
Initialisation des accumulateurs à 0
Pour chaque document
        Retrouver l'ensemble des critères de qualité {c1, c2,..., cl} pour le document di ayant la notation
        respective {n1, n2,..., nl} /* l : nombre de critères décrivant la qualité le document di */
        Pour chaque critère de qualité ck ayant la notation nk (avec k = 1,..., l)
                Faire
                  Début
                        Retrouver l'ensemble des documents de la collection {dck,1 , dck,2,... ,dck,mk}
                                        pour lesquelles le critère ck a été noté
                        /* mk : nombre de documents pour lesquels le critère ck a été noté */
                        Pour chaque document dck,j (avec j = 1,..., mk)
                                Faire
                                  acc[dck,j] = acc[dck,j] + Scoring(dck,j, ck, Q, m)
                        /* Scoring(dck,j, ck, Q, m) : fonction donnant le score de qualité du document dck,j
                        pour le critère de qualité ck la requête Q et la méthode de calcul m*/
                  Fin
Tri des accumulateurs de score par ordre décroissant
Affectation des valeurs de scores de documents par ordre selon les scores des accumulateurs
Fin.
```

Figure 8. *Procédure principale pour l'algorithme de recommandation de documents*

La fonction $\mathrm{Scoring}(s_{ck,j},\ c_k,\ Q,\ m)$ est calculée pour chaque document de la collection sur une requête à partir d'un des modèles de calculs présentés précédemment (affectation linéaire de poids *AL*, élimination par aspects *EA*, méthode d'Anderson *AND*, de Subramanian et Gershon *SG*).

4.4. Révision des notations jusqu'à la convergence de l'annotation collaborative

Le problème de la mise à jour des méta-données (refreshing) s'inscrit dans une problématique de rafraîchissement et de maintien de la cohérence (et de la synchronisation) entre les données et leurs méta-données. Le risque que les méta-données soient rapidement "décorrélées" des documents conditionne la validité des résultats de recherche et l'intérêt de l'usage des méta-données. Par souci de clarté terminologique, nous préférons employer les termes de "rafraîchissement" pour des méta-données dont la génération ou le calcul est automatisable et le terme "révision" pour des méta-données subjectives impliquant l'appréciation d'un individu.

Cette section présente complètement une des approches qui permet d'atteindre un consensus sur la notation d'un critère de qualité. La proposition de [DeG74] a été ici adaptée dans le cadre de l'annotation collaborative de documents pour la raison suivante : l'approche est facile à appréhender et relativement simple à implanter, puisque chaque notation associée à un critère est datée. L'étude de la stabilité (ou convergence des jugements par collaboration) peut être menée automatiquement.

Notre objectif est de proposer différents mécanismes de révision par appel à la bibliothèque de fonctions destinées à refléter au mieux l'évolution dynamique de l'annotation par collaboration. Cette approche est transposable au filtrage d'information dans la mesure où l'utilisateur peut réviser son profil qualité, par exemple, selon la confiance qu'il accorde à telle ou telle source ou annotateur.

Deux cas de figure se présentent lors de la révision des notations d'un critère :

1. un seul annotateur est en charge de la notation du critère et revoit successivement sa dernière évaluation en fonction d'un apport extérieur de connaissance (en l'occurrence d'un arrivage de nouvelles informations).
2. plusieurs annotateurs chargés de la notation du critère sont amenés à revoir leur propre jugement en prenant en compte les jugements respectifs des autres experts.

Dans le cas (1), on atteint une convergence de la notation du critère de qualité quand la variation des notations est faible. On note $n_i(c, t_o)$ la notation assignée par l'annotateur *Ai* au critère *c* à la date t_o. L'état de convergence sur la notation du critère *c* par l'annotateur *Ai* est atteint à la date t_m ($t_o < t_1 <...< t_m$) telle que :

$$\frac{?\, n_i(c,t_m)}{?\, t} \approx 0$$

Dans le cas (2), pour atteindre une convergence des notations successivement révisées pour un critère de qualité, les trois étapes suivantes peuvent être proposées : (1) la pondération des jugements menées par l'ensemble des annotateurs, (2) la révision des notations, (3) l'itération du processus de révision, (4) la détermination d'un état de convergence des jugements.

4.2.1. Pondérations des jugements

Si l'on considère un groupe de *k* annotateurs de documents. On suppose que chacun de ces *k* annotateurs affecte une notation pour le critère *c*. La notation du critère *c* peut être considérée comme une variable arbitraire. La valeur de la notation du critère *c* est supposée appartenir à un espace *N* tel que $N = [0,1]$. $n_1(c, t_o)$, $n_2(c, t_o)$...,$n_k(c, t_o)$ sont des notations sur *N* représentant respectivement les appréciations *a priori* du critère *c* par chacun des *k* annotateurs à la date t_o.

Hypothèse 1 : Si un annotateur *Ai* est mis au courant à la date t_1 ($t_0 < t_1$) de la notation $n_j(c, t_o)$ ($j \neq i$) de chacun des autres annotateurs du groupe, on peut penser qu'il lui sera naturel de réviser son jugement et sa première notation. Dans le cas d'un processus d'évaluation

collaboratif, il intégrera les informations, opinions et jugements du reste du groupe. Si on suppose que l'annotateur *Ai* révise sa notation de cette manière, alors on peut supposer que sa notation révisée à la date t_1 est une combinaison linéaire des notations faites à la date t_o en incluant la sienne.

D'autres modèles sont ici envisageables et peuvent être déterminées (voire "appris") par le système lors de la révision effective de l'annotation par les annotateurs. On note $p_{ij}(c, t_1)$ avec $i = 1,...,k$ et $j = 1,...,k$, le poids qu'affecte l'annotateur *Ai* à la notation faite par l'annotateur *Aj* lors de la révision de son jugement à la date t_1. Considérons les deux hypothèses suivantes :

Hypothèse 2 : $p_{ij}(c, t_1) \geq 0$

Hypothèse 3 : $\sum_{j=1}^{k} p_{ij}(c,t_1)=1$

Les poids $p_{i1}(c, t_1),...,p_{ik}(c, t_1)$ sont des constantes positives choisies par l'annotateur *Ai* avant de connaître les notations faites par les autres annotateurs. Ces constantes reflètent la confiance relative qu'accorde *a priori* l'annotateur *Ai* aux opinions des autres annotateurs en s'incluant dans le groupe. Par exemple, il peut considérer que l'annotateur *Aj* est mieux placé que lui pour évaluer le document sur le critère *c* ou que l'annotateur *Aj* a accès à une plus grande quantité d'informations pour évaluer ce critère, donc il affecte un poids élevé $p_{ij}(c,t_1)$ à la date t_1.

4.2.2. Révision des notations

Si, à la date t_1, l'annotateur *Ai* est informé des notations antérieures $n_1(c, t_o),..., n_k(c, t_o)$ affectées au critère *c* par les autres annotateurs du groupe, on suppose qu'il révisera sa propre évaluation $n_i(c, t_o)$ telle que d'après l'hypothèse 1 :

$$\text{à la date } t_1,\ n_i(c,t_1)=\sum_{j=1}^{k} p_{ij}(c,t_1).n_j(c,t_0)$$

Soient :

$P(c,t_1)$, la matrice $k \times k$ des éléments $p_{ij}(c, t_1)$ avec $i = 1,...,k, j = 1,...,k$ dont la somme de chaque ligne égale *1*, (d'après l'hypothèse 3)
$N(c, t_o)$, le vecteur colonne $k \times 1$ des notations des *k* annotateurs à la date t_o dont la transposée est $N(c, t_o)^T = [n_1(c, t_o),..., n_k(c, t_o)]$ et
$N(c, t_1)$, le vecteur colonne $k \times 1$ des notations des *k* annotateurs à la date t_1 dont la transposée est : $N(c, t_m)^T = [n_1(c, t_1),..., n_k(c, t_m)]$

on a donc :

$$N(c, t_1) = P(c,t_1).N(c, t_o)$$

4.2.3. Itération du processus de révision

Si chaque annotateur est informé des notations antérieures des autres annotateurs, alors leurs notations seront révisées de $n_1(c, t_o),..., n_k(c, t_o)$ en $n_1(c, t_1),..., n_k(c, t_1)$. De la même manière, si un annotateur Ai change sa notation de $n_i(c, t_o)$ en $n_i(c, t_1)$ et qu'il est informé que les $(k-1)$ autres annotateurs ont également changé leurs notations, il peut souhaiter réviser, à la date t_2, sa dernière évaluation telle que :

$$n_i(c, t_2) = \sum_{j=1}^{k} p_{ij}(c, t_2).n_j(c, t_1)$$

Le processus se réitère et, après m révisions des notations, on note $n_i(c, t_m)$, la notation attribuée par l'annotateur i. Soit $N(c, t_m)$ le vecteur colonne $k \times 1$ dont la transposée est :

$N(c,t_m)^T = [n_1(c, t_m),..., n_k(c, t_m)]$

on a : $N(c, t_2) = P(c,t_2).N(c, t_1) = P(c,t_2).P(c,t_1).N(c, t_o)$

et plus généralement :

$$N(c, t_m) = P(c,t_m).N(c, t_{m-1}) \text{ pour } m = 2,3...$$

4.2.4. Convergence des notations

Les notations des k annotateurs convergent si et seulement s'il existe une évaluation $n(c)$ telle que :

$$\lim_{tm \to +\infty} n_i(c, t_m) = n(c) \quad pour\ i = 1,..., k$$

c'est-à-dire, si tous les éléments du vecteur colonne $N(c, t_m)$ convergent vers la même limite quand t_m tend vers l'infini.

Soit $p_{ij}(c,t_m)$, l'élément de la ligne i et colonne j de la matrice $P(c,t_m)$, la convergence est atteinte si et seulement s'il existe un vecteur $\pi = (\pi_1, ..., \pi_k)$ tel que :

$$\text{pour } i = 1,...,k \text{ et } j = 1,...,k, \lim_{tm \to +\infty} p_{ij}(c, t_m) = p_j$$

Si la convergence est atteinte, alors la notation commune pour chacun des k annotateurs sur le critère c sera :

$$n(c) = \sum_{i=1}^{k} p_i.n_i(c, t_m)$$

La convergence des notations peut être exploitée pour le filtrage par collaboration, car les utilisateurs peuvent vouloir imposer un seuil de tolérance dans la dispersion des jugements préalablement à la présentation des résultats. L'intérêt de notre approche est de pouvoir implanter différents modèles de révision et permettre une configuration du système selon le contexte "social" ou "consensuel" ou "corporatif" du filtrage par collaboration.

5. XDARE

XDARE (*XML-Document Annotation and Recommendation Environment*) est un serveur dédié à l'annotation et à la recommandation collaboratives de documents XML. Il permet la production, la gestion partagée des documents XML sécurisés selon différents niveaux de confidentialité.

5.1. Fonctionnalités et architecture de l'environnement XDARE

Les principales fonctionnalités qui font de *XDARE* un système mixte sont : la gestion partagée de documents XML (et de leurs méta-données), la gestion des utilisateurs (et de leurs profils thématique et qualité), le filtrage et la recherche d'information. Le serveur permet à des personnes identifiées par leurs *login* et mot de passe de produire et consulter différentes versions de documents au moyen de commandes propres à un protocole de communication développé au-dessus de IP. Chaque document converti en XML ou XHTML possède un degré de confidentialité de 0 à 3 (respectivement, "diffusion publique", "diffusion restreinte", "diffusion confidentielle", "secret"). Chaque membre (en tant qu'utilisateur identifié et connu du serveur) possède un niveau d'habilitation ("non habilité", "habilitation restreinte", "habilité confidentiel", "habilité secret") qui lui donne certains droits d'accès selon le degré de confidentialité des documents (en mode création, modification, annotation, suppression et consultation). Par exemple, un membre non habilité a le droit de consulter et annoter des documents à diffusion publique mais il n'a pas de droit en modification ; un membre à habilitation restreinte a le droit de créer, consulter, mettre à jour, annoter et supprimer des documents à diffusion publique et restreinte, mais il n'a aucun droit sur les documents aux niveaux supérieurs de confidentialité. Deux types de documents peuvent être créés (voir l'attribut DOCTYPE dans le Tableau 2) :

- les documents standards dont le degré de confidentialité est le même pour tout le contenu du document,
- les documents composites possédant plusieurs degrés de confidentialité correspondant à certaines parties du texte : ainsi, selon le niveau d'habilitation du membre consultant, tout ou partie du document sera (ou non) visible, "annotable" et/ou modifiable.

En fonction des degrés de confidentialité des documents et des niveaux d'habilitation des membres, le serveur permet, outre les opérations classiques de gestion des documents, la génération de méta-données. Comme l'illustre la Figure 9, l'architecture du système XDARE se compose de trois couches logicielles au-dessus de plusieurs systèmes de stockage des données. Les méta-données sont initialement spécifiées par un annotateur à partir d'une collection de documents bruts préalablement convertis en XML ou XHTML (par le module externe XML Converter), stockés dans une base et indexés selon la technique décrite précédemment (Indexer).

L'annotation peut être réalisée par l'éditeur Amaya et en particulier la partie client d'Annotea [AM01]. L'appel à l'outil 3DM assure la fusion des versions de documents et des annotations. Les méta-données sont instanciées soit par l'annotateur soit par génération automatique à partir de la bibliothèque de fonctions. Les méta-données sont ensuite utilisées par l'optimiseur pour établir un plan optimal de construction d'un résultat à une requête soumise par l'utilisateur en mode recherche ou filtrage d'information. Pour répondre aux requêtes et assurer la recommandation des documents, l'ordonnanceur invoque des programmes de la bibliothèque d'algorithmes et stocke tous les scores et profils (thématique et qualité) dans un SGBD relationnel (Oracle 8) dont les données sont exploitées par une technique d'apprentissage basé sur la logique inductive (PLI Profiler: Aleph 3).

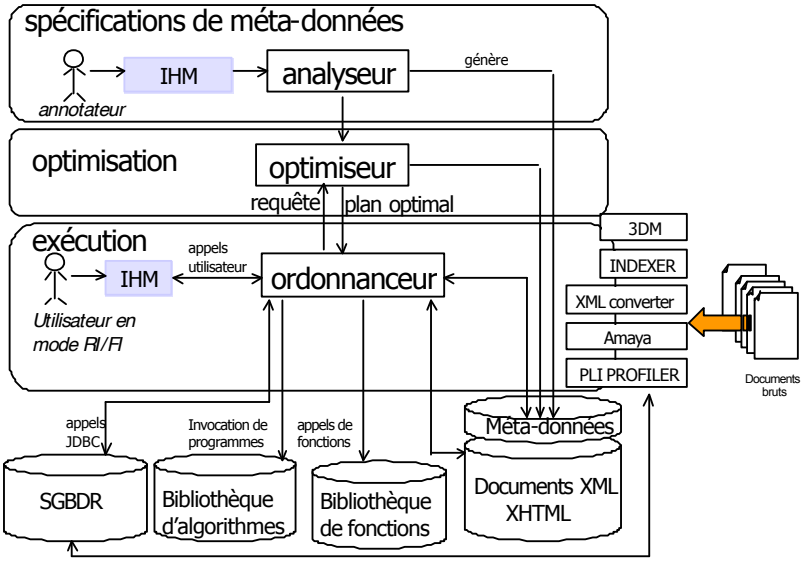

Figure 9. *Architecture du serveur de documents XDARE*

5.2. Analyse et conception

Dans le cadre d'une application orientée utilisateur tel que *XDARE*, il nous a semblé préférable d'utiliser l'*API DOM* (*Document Object Model*) pour accéder aux documents XML. Le protocole *DOM* convertit un document XML en une collection d'objets qu'il est ensuite possible de manipuler (et d'indexer) par leur structure arborescente. La conception du serveur *XDARE* s'articule en 6 packages (Figure 10). Plus précisément, le package *Document* utilise deux *design pattern* pour le traitement

des documents XML. Le *design pattern Factory* permet de générer un *parser* avec des caractéristiques particulières pour parcourir le document. Le *design pattern Builder* génére, dans le protocole *DOM*, la structure arborescente du document XML.

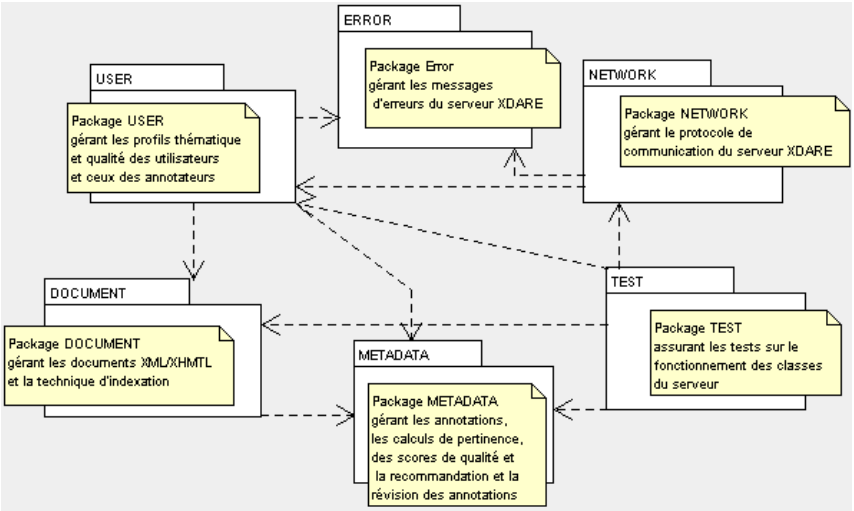

Figure 10. *Les packages de XDARE*

Le serveur ne traite que des documents valides respectant la DTD donnée en Figure 11. Les méthodes assurant la gestion d'un document XML (parcours d'un document, lecture d'informations, ajout de données,…) sont assurées au niveau de la classe DocumentMonitor (voir Figure 14 en Annexe).

6. Exemple d'application

Considérons l'exemple d'un processus de relecture et de sélection d'articles scientifiques pour une revue et, en particulier, les trois scénarii suivants :

6.1. Scénario 1 : Annotation et sélection de documents

Lors du processus de sélection des articles soumis au comité de lecture de la revue, trois rapporteurs relisent, évaluent et commentent plusieurs articles parmi l'ensemble des soumissions. Chaque article est noté de 0 à 10 sur les 5 critères suivants : originalité, adéquation aux thèmes, qualité scientifique et technique, présentation, impact pratique et industriel. Des commentaires peuvent être associés à

chaque article, soit sur sa globalité ou, soit sur certaines parties du texte (Figure 12). Les annotations et commentaires sont, par la suite, fusionnés pour constituer les notifications d'acceptation/refus aux auteurs. La sélection multicritère des articles se base sur plusieurs stratégies prenant en compte une pondération des critères (par exemple, la présentation de l'article est jugée moins importante que la qualité scientifique et technique, l'adéquation aux thèmes et l'originalité).

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT document (version+, metadataset*)>
<!ATTLIST document
   did CDATA #REQUIRED
   title CDATA #REQUIRED
   confidentialLevel (0 | 1 | 2 | 3) #REQUIRED
   creator CDATA #REQUIRED
   publisher CDATA #IMPLIED
   creationDate CDATA #REQUIRED
   publicationDate CDATA #IMPLIED
   category (CD | RP | RT | O | C) #REQUIRED
   doctype (0 | 1) #REQUIRED
   mimetype CDATA #REQUIRED
   size CDATA #REQUIRED>
<!ELEMENT version (block+, metadataset*)>
<!ATTLIST version
   number CDATA #REQUIRED
   creationDate CDATA #REQUIRED
   confidentialLevel (0 | 1 | 2 | 3) #REQUIRED
   creator CDATA #REQUIRED>
<!ELEMENT block (keyword*, value, metadataset*)>
<!ATTLIST block
   confidentialLevel (0 | 1 | 2 | 3) #REQUIRED
   language (fr | en) #REQUIRED
   linkage CDATA #REQUIRED>
<!ELEMENT keyword (#PCDATA)>
<!ELEMENT value (#PCDATA)>
<!ELEMENT metadataset (#PCDATA | metadataset | metadata)*>
<!ATTLIST metadataset
   qid IDREF #REQUIRED
   type (ContentMD | ManagementMD | RepresentationMD | RelativeQualityMD) #REQUIRED
   scheme CDATA #REQUIRED
   sortkey CDATA #IMPLIED
   index (index | noindex | asparent) "asparent"
   show (noshow | show | inherit) "inherit">
<!ELEMENT criterium (#PCDATA)>
<!ELEMENT notation (#PCDATA)>
<!ELEMENT annotator (#PCDATA)>
<!ELEMENT program (#PCDATA)>
<!ELEMENT comment (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ENTITY % annot "criterium, notation, date, generator ">
<!ELEMENT consensus (%annot;, comment*)>
<!ELEMENT metadata ((%annot;)+, comment*, consensus*)>
<!ATTLIST metadata
   mid CDATA #REQUIRED>
<!ELEMENT generator (annotator | program)>
```

Figure 11. *DTD des documents gérés par le serveur XDARE*

6.2. Scénario 2 : Révision des annotations

Dans ce scénario, l'affectation d'un article à un rapporteur est faite de manière arbitraire. Pour chaque article, trois annotateurs sont chargés de la notation des 5 critères. Ils identifient au préalable leur niveau de compétence les uns par rapport aux deux autres rapporteurs pour chaque article qu'ils sont en charge de relire et de rapporter. Par la suite, lors de la sélection finale des articles, ils sont amenés à revoir leur propre jugement en prenant en compte les jugements respectifs des deux autres annotateurs selon leur domaine de compétence respectif.

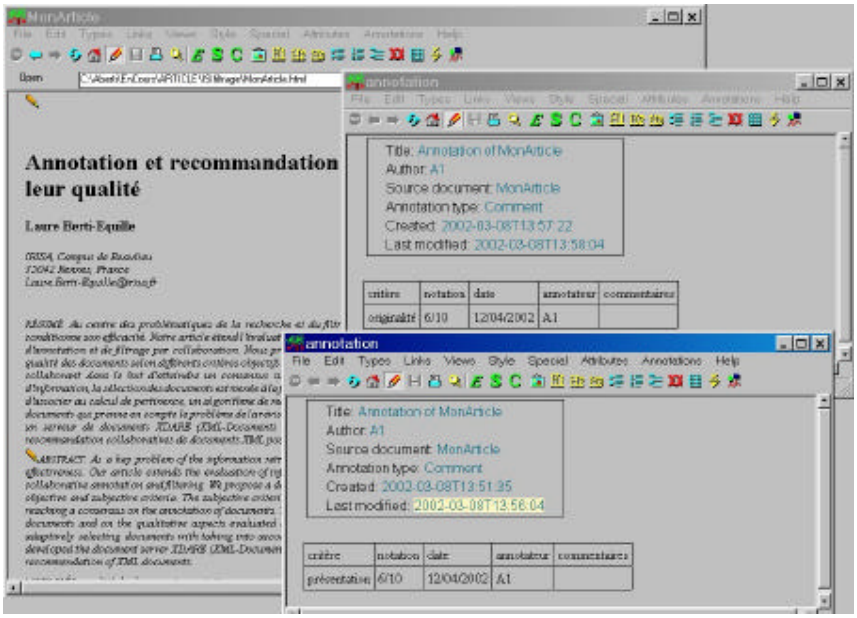

Figure 12. *Copie d'écran de l'annotation de MonArticle avec Amaya [AM01]*

6.3. Scénario 3 : Recherche de documents basé sur la qualité

Dans ce scénario, les articles ont été annotés et un utilisateur recherche les meilleurs articles traitant de qualité de document (Figure 13).

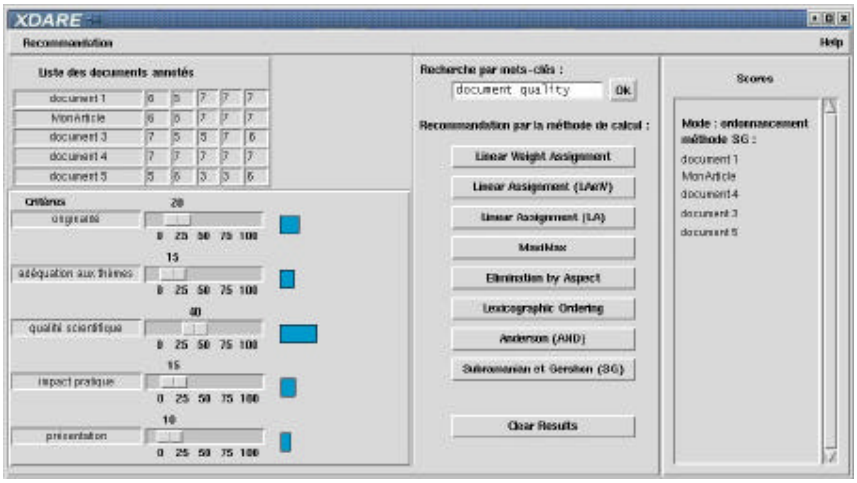

Figure 13. *Copie d'écran de XDARE pour la recommandation en mode recherche*

7. Conclusion

La spécification et l'exploitation de méta-données décrivant la qualité des documents peuvent améliorer l'efficacité des systèmes de recherche et de filtrage d'information. Une requête dans un épisode de recherche ponctuelle peut inclure des exigences sur la qualité du résultat retourné par le système. Les profils d'utilisateurs peuvent également inclurent des exigences de qualité de résultats et raffiner la pertinence thématique du contenu. Dans un contexte collaboratif, nous proposons une modélisation des documents qui inclut des méta-données décrivant la qualité des documents selon différents critères objectifs et subjectifs. Les critères subjectifs peuvent être définis et évalués par un groupe d'annotateurs collaborant pour atteindre un consensus sur l'annotation des documents. Une sélection des documents peut ensuite être menée à la fois sur le contenu des documents comme sur les aspects qualitatifs évalués par le groupe d'annotateurs. Nous avons proposé un ensemble d'opérations pour mener un processus de recommandation des documents prenant en compte le problème de révision dynamique des annotations. Celles-ci sont applicables à la fois en RI et en FI et ont été implantées, au sein du serveur de documents *XDARE* en tant que modules faisant appel à deux bibliothèques d'algorithmes et de fonctions en cours d'extension pour l'annotation et la recommandation collaboratives de documents XML. Nos perspectives de recherche concernent maintenant : 1) l'étude de stratégies de rafraîchissement de tout type de méta-données associées aux documents, 2) l'apprentissage des profils (thématique et qualité) par induction (module PLI profiler), 3) l'évaluation et la

validation du système *XDARE* dès que le développement de ce dernier module sera achevé.

8. Références

[ADT97] F.A. Asnicar, M. DiFant, C. Tasso. User model-based information filtering. Proc. of the 5th Congresso della Associazione Italiana per l'Intelligenza Artificiale, 1997.

[AM01] Amaya, W3C's Editor/Browser, http://www.w3.org/Amaya/

[And90] E. Anderson. Choice models for evaluation and selection of software packages. J. of Management Information Systems, 6:123-138, 1990.

[AT97] F.A. Asnicar, C. Tasso. ifWeb : a prototype of user model-based intelligent agent for document filtering and navigation in the World Wide Web. Proc. of the 6th Intl. Conf. on User Modeling (UM'97), 1997.

[Bac92] P.E. Baclace. Competitive agents for information filtering. Com. of the ACM, 35(12):50, 1992.

[Bal97] M. Balabanovic. An adaptive Web page recommendation service. Proc. of the 1st Intl. Conf. on Autonomous Agents, 1997.

[Bar94] C.L. Barry. User-defined relevance criteria : an exploratory study. J. of the American Society for Information Science, 45(3):135-141, 1994.

[BC92] N.J. Belkin, W.B. Croft. Information filtering and information retrieval : two sides of the same coin ? Com. of the ACM, 35(12):29-38, 1992.

[Ber99] L. Berti. Qualité de données multi-sources et recommandation multi-critère. Thèse de Doctorat, Université de Toulon, 1999.

[BT94] G. Brajnik, C. Tasso. A shell for developing non-monotonic user modeling systems. Intl. J. of Human Computer Studies, 40:31-62, 1994.

[Cal96] J. Callan. Document filtering with inference networks. Proc. of the 19th Annual Intl. ACM SIGIR Conf., p. 262-269, 1996.

[Chr96] E. Christian. ISO 9000. Application Profile for the Government Information Locator Service (GILS), 1996 http://www.usgs.gov/public/gils , http://www.gils.net/

[CPPR98] S. Calabretto, J.M. Pinon, L. Poullet, M.A. Richez. De la qualité de l'information à la qualité de la documentation. Document Numérique, 12(1):37-52, 1998.

[Cro95] W.B. Croft. Effective text retrieval based on combining evidence from the corpus and users. IEEE Expert, 1995.

[DeG74] M. DeGroot. Reaching a consensus. J. of American Statistical Association, 69:118-121, 1974.

[Den97] N. Denos. Modélisation de la pertinence en recherche d'information : modèle conceptuel, formalisation et application. Thèse de Doctorat, Université Joseph Fourier, Grenoble I, 1997.

[EBGP96] P. Edwards, D. Bayer, C.L. Green, T.R. Payne. Experience with learning agents which manage Internet-based information. Proc. of the AAAI Spring Symposium on Machine Learning for Information Access, 1996.

[FC94] C. Fritz, B. Carter. A classification and summary of software evaluation and selection methodologies. Technical report, Mississippi State University, 1994.

[GGM97] L. Gravano, H. Garcia-Molina. Merging ranks from heterogeneous internet sources. Proc. of VLDB'97, p. 196-205, 1997.

[HBP00] D. Hiom, M. Belcher, E. Place. People Power and the Web: Building Quality Controlled Portals, TERENA Networking Conference, 2000. http://www.desire.org/

[HGLW87] G.E. Herman, G. Gopal, K.C. Lee, A. Weinrib. The datacycle architecture for very high throughput database systems. Proc. of the ACM SIGMOD, 1987.

[HKW95] M. Höfferer, B. Knaus, W. Winiwarter. An evolutionary approach to cognitive information filtering. Proc. of the ACM SIGIR Conf. on Research and Development in Information Retrieval, p. 1-15, 1995.

[Lan95] K. Lang. Newsweeder : Learning to filter news. Proceedings of the 12th Intl. Conf. on Machine Learning, 1995.

[Lin01] T. Lindholm. The 3DM merging and differencing tool for XML documents, M.Sc. Thesis, Helsinki University of Technology, Sept. 2001.

[LM01] Q. Li, B. Moon. Indexing and querying XML data for regular path expressions, Proc. of VLDB Conf., 2001.

[MGT+87] T. Malone, K. Grant, F. Turbak, S. Brobst, M. Cohen. Intelligent information sharing systems. Com. of the ACM, 30(5):390-402, 1987.

[Mou96] A. Moukas. Amalthaea : Information discovery and filtering using a multi-agent evolving ecosystem. Proc. of PAAM, 1996.

[NISO01] Z39.50: Maintenance agency, attribute set Bib-1, 2001. ftp://ftp.loc.gov/pub/z3950/defs/bib1.txt , http://www.niso.org/standards/index.html

[PE97] T.R. Payne, P. Edwards. Interface agents that learn: An investigation of learning issues in a mail agent interface. Applied Artificial Intelligence, 11(1):1-32, 1997.

[PMB97] M. Pazzani, J. Muramatsu, D. Billsus. Syskill & Webert : identifying interesting Web sites. Proc. of the 13th National Conf. on Artificial Intelligence, 1997.

[Ric83] E. Rich. Users are individuals : individualising user models. Intl. J. of Man-Machine Studies, 18:199-214, 1983.

[Ric96] M.A. Richez. Propositions pour la conception et la mise en oeuvre d'un système d'aide à l'élaboration et au partage de documents composites. Thèse de Doctorat, INSA Lyon, 1996.

[RM96] P. Resnick, J. Miller. PICS : Internet access controls without censorship. Com. of the ACM, 39(10):87-93, 1996. http://wwww.w3.org/PICS

[RV97] P. Resnick, H. Varian. Recommender systems. Com. of the ACM, 40(3):56-89, 1997.

[Sch91] L. Schamber. Users' criteria for evaluation in a multimedia environment. Proc. of the American Society for Information Science (ASIS'91), p. 126-133, 1991.

[Sch93] L. Schamber. Relevance and information behavior. ARIST, 29:3-48, 1993.

[Voo94] E.M. Voorhees. Software agents for information retrieval. Proc. of the 1994 Spring Symposium of Softwre Agents, p. 126-129, 1994.

[WGMD95] S. Weibel, J. Godby, E. Miller, R. Daniel. OCLC/NCSA metadata workshop. Report. http://dublincore.org/, 1995.

[WSF95] R.Y. Wang, V.C. Storey, C.P. Firth. A framework for analysis of data quality research. IEEE TKDE, 7(4):623-638, 1995.

[XC96] J. Xu, W.B. Croft. Query expansion using local and global document analysis. Proc. of the 19th Annual Intl. ACM SIGIR Conf. on Information Retrieval, p. 4-11, 1996.

[Yao95] Y.Y. Yao. Measuring retrieval effectiveness based on user preference of documents. J. of the American Society for Information Science, 46(2):133-145, 1995.

Annexes

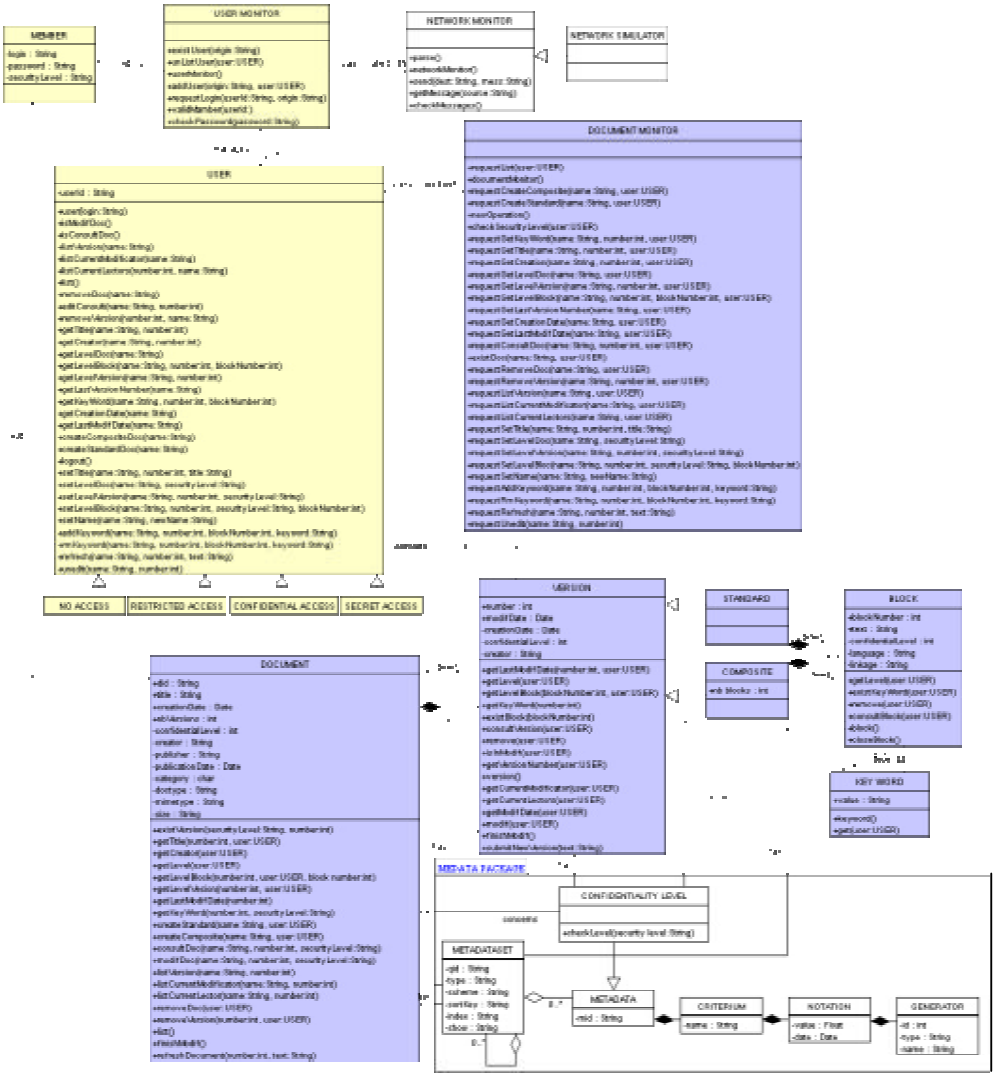

Figure 14. *Extrait du diagramme de classes UML de conception du serveur XDARE (packages User, Document et Metadata)*