Modèle générique de profils pour la personnalisation de l'accès à l'information

Pascaline Tchienehom

Modèle générique de profils pour la personnalisation de l'accès à l'information

Pascaline Laure Tchienehom

IRIT, 118 route de Narbonne, 31062 Toulouse cedex 4, France
Pascaline.Tchienehom@irit.fr

RÉSUMÉ. *L'hétérogénéité des techniques d'accès à l'information a soulevé le problème de la définition d'un modèle homogène qui leur servirait de base. Dans cet article, nous proposons un modèle générique de profil qui permet de décrire la structure et la sémantique générale de tout type de profil d'information ou d'usager pour l'accès à l'information. Nous construisons également un graphe sémantique de profils qui combine des instances de profils ainsi que des classes (génériques) du modèle générique. L'exploitation de ce graphe, via la définition de règles d'inférence, nous permet de déduire les couples d'éléments de profils qui soient appariables. Ces couples d'éléments appariables sont ensuite utilisés par la méthode d'appariement que nous proposons et qui nous permet de calculer un degré de ressemblance entre deux profils.*

ABSTRACT. *Heterogeneity of information access techniques has raised the problem of defining a homogeneous model, which would be used as a basis for them. In this article, we propose a profile generic model, which describes the general structure and semantics of any information or user profile type for information access. We also build a profiles semantic graph, which combines profiles instances as well as (generic) classes of the generic model. The exploitation of this graph, via the definition of inference rules, enables us to deduce couples of profiles elements, which we can match. These couples of elements that can be matched are then used by the proposed matching method, which calculates a degree of resemblance between two profiles.*

MOTS-CLÉS : *Profils, sémantique, personnalisation, accès à l'information.*

KEYWORDS: *Profiles, semantics, personalization, information access.*

Catégorie : Jeune Chercheur

1. Introduction

La pertinence de l'information dans les techniques d'accès à l'information a conduit à la définition de différents modèles de description d'informations et d'usagers. Ces modèles sont de différents types (parties de document, documents, collections de documents, thèses, articles de journaux, utilisateur individuel, groupe d'utilisateurs, etc.). En recherche comme en filtrage d'information, on peut restituer des informations selon la granularité des usagers (utilisateur individuel ou groupe d'utilisateurs) et/ou selon la granularité des informations (parties de document, documents, collections de documents). Ainsi, un document non pertinent peut éventuellement contenir des parties qui soient pertinentes pour les besoins de l'utilisateur. De même, dans un groupe d'usagers les attentes de chaque individu du groupe ne sont pas forcément les mêmes. Le but est de découvrir des collections spécialisées dans des domaines spécifiques, de découvrir de nouvelles informations ou de retrouver toutes les unités d'informations pertinentes pour un besoin donné tout en s'adaptant aux caractéristiques de chaque usager ou groupe d'usagers. Il existe une multitude d'approches d'accès à l'information qui tentent de résoudre ces problèmes. L'hétérogénéité de ces approches et des modèles sous-jacents donne une plus grande envergure aux problématiques relatives : à la définition de modèles génériques pour la conception de ces systèmes, à l'interopérabilité entre différents modèles et applications.

Dans cet article, nous nous intéressons à la définition d'un modèle générique de profil qui permet de décrire tout type de profil pour l'accès à l'information. La particularité de ce modèle générique est liée à l'aspect sémantique qui y est intégré et qui va permettre la coopération et l'interopérabilité entre différents modèles de profils. Ce modèle générique de profil décrit la structure et le contenu des profils mais également la sémantique qui y est associée. Nous construisons également un graphe sémantique qui combine des instances de profils ainsi que des classes (génériques) du modèle générique de profil. L'objectif de ce graphe sémantique est d'identifier les couples d'éléments (ou d'attributs) descriptifs de profils qui sont appariables car ayant la même sémantique ou des sémantiques compatibles, ceci au travers de règles d'inférence. Par la suite, nous définissons une méthode d'appariement de profils qui utilise les résultats des règles d'inférence définies pour calculer un degré de ressemblance entre deux profils. Enfin, des expérimentations menées sur les collections « Agence Télégraphique de Suisse 94 » (ATS 94), « Le Monde 94 » (LeMonde 94) et « Los Angeles Times 94 » (LaTimes 94) de la campagne d'évaluation « CLEF 2001 », nous permettent d'analyser la méthode d'appariement proposée pour l'accès personnalisé à l'information.

2. Etat de l'art

Les techniques d'accès à l'information permettent à un individu ou à un groupe d'individus d'obtenir les informations dont il a besoin. On peut subdiviser ces techniques en deux groupes : la technique du pull qui a besoin d'une requête

explicite d'un individu et la technique du push qui permet de renvoyer des informations aux usagers sans une demande explicite de leur part.

La Recherche d'Information (RI), qui est une technique du pull, est basée sur l'expression d'un besoin utilisateur à travers une requête formulée dans un langage plus ou moins structuré (Baeza-Yates *et al*., 1999). Cependant, l'intention réelle de l'utilisateur n'est pas toujours évidente dans sa façon de formuler ses requêtes et cela peut générer des ambiguïtés au niveau du sens des mots. Plusieurs solutions existent qui permettent de préciser le sens d'une requête à travers des reformulations de requêtes basées sur : les jugements utilisateurs (Boughanem *et al*., 1999), la notion de contexte (Bottraud *et al*., 2003), (Pitkow *et al*., 2002) ; etc.

Le Filtrage d'Information (FI), qui est une technique de push, est une tâche relativement passive (Belkin *et al*., 1992) pour l'usager qui ne formule pas explicitement son besoin à travers une requête comme en RI. En Filtrage d'Information, on utilise une représentation de l'utilisateur que l'on appelle profil utilisateur pour lui renvoyer de l'information. Il existe plusieurs méthodes de filtrage (Montaner *et al*., 2003) : le filtrage cognitif ou basé sur le contenu qui utilise la description du contenu des informations pour déterminer à quels profils utilisateurs elles correspondent (Pazzani *et al*., 1996), (Korfhage, 1997) ; le filtrage social ou collaboratif qui utilise les jugements utilisateurs sur un ensemble d'informations pour effectuer des recommandations (Goldberg *et al*., 1992), (Konstan *et al*., 1997) ; le filtrage démographique qui se base sur les données démographiques des usagers (âge, profession, etc.) pour faire des recommandations (Krulwich, 1997). Ces approches de filtrage ne sont pas exclusives et différentes méthodes hybrides ont été développées (Good *et al*., 1999), (Pazzani, 1999), (Balabanovic *et al*., 1997).

Les méthodes d'accès à l'information sont basées sur la description de données qu'elles manipulent et qui sont appelées *profils*. Le profil d'un objet est un ensemble de caractéristiques qui permet de l'identifier ou de le représenter. Les profils utilisés dans les techniques d'accès à l'information sont de nature variée (profil utilisateur, profil de document, etc.) et leur structure peut-être composée d'un ou de plusieurs éléments descriptifs : centres d'intérêts ou mots clés, données démographiques, préférences utilisateurs, métadonnées de documents, etc. Les modèles existants de profils sont généralement basés sur la structure et sur une sémantique implicite, ce qui pose le problème de la coopération de profils décrits par des structures et taxonomies (noms d'attributs) différentes. Pour pallier cela, on a besoin de modèles de profils : génériques (Kobsa, 2001) ; sémantiques (Dolog *et al.*, 2003) ; extensibles, flexibles, ré-utilisables et interopérables (Berners-Lee *et al.*, 2001). Notre contribution s'inscrit dans ces contextes. La particularité du modèle générique de profils que nous proposons est liée à la sémantique qui y est intégré et qui permet de décrire aussi bien la structure que le contenu d'un profil.

3. Définition de profils pour un accès personnalisé à l'information

Dans cette section, nous proposons un modèle générique de profil en UML pour la description de la structure, du contenu et de la sémantique de tout type de profils. La combinaison d'instances de ce modèle à travers un graphe sémantique va permettre d'identifier automatiquement les couples appariables d'attributs descriptifs de profils. Ce graphe sémantique est décrit avec une approche orientée

logique de description (Cullot *et al*., 2003) via le formalisme RDF/RDFS/OWL qui nous fournit un cadre formel pour expliciter nos règles d'inférence. Par la suite, nous utilisons le résultat de nos règles d'inférence pour décrire une méthode d'appariements de profils différents. Cette méthode va mesurer un degré de similarité (et non une valeur binaire de similarité) entre deux profils pour un ensemble d'attributs donné. Nous faisons également une analyse de la méthode proposée au travers d'expérimentations menées sur les collections «ATS 94», «LeMonde 94» et «LaTimes 94» de la campagne « CLEF 2001 ».

3.1. *Modèle générique de profil*

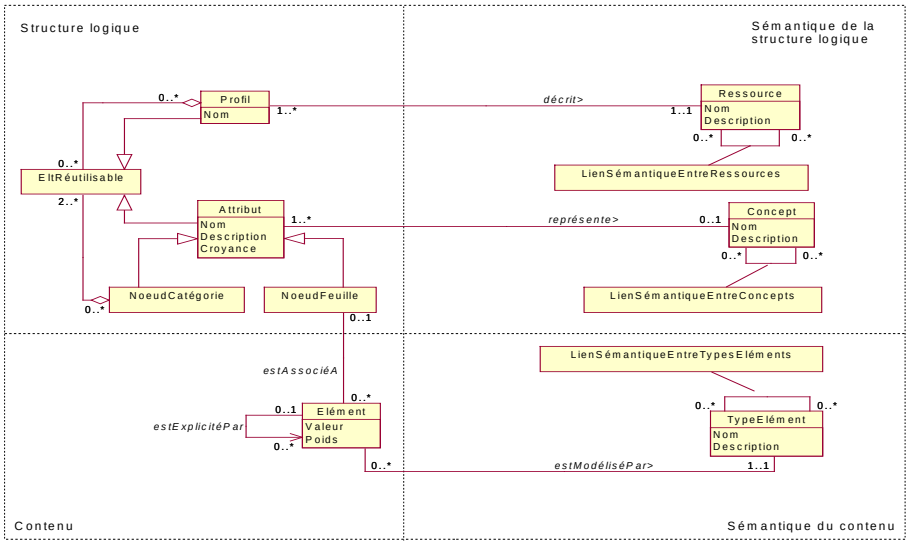

FIG 1 – Modèle générique de profil

Le schéma de la figure *FIG 1* présente notre modèle générique de profil. Il résulte de l'analyse de différents systèmes de recherche et de recommandation afin d'en déduire un modèle général. Les systèmes existants sont conçus pour atteindre des objectifs particuliers en fonction des spécificités propres de leur contexte : recommandation de pages web en fonction des signets (Rucker *et al*., 97), filtrage de mails (Goldberg *et al*., 92), commerce électronique (Cho *et al*., 02), etc. Contrairement à ces systèmes, notre modèle est assez général pour être utilisé dans différentes applications.

Le modèle générique de profil de la figure *FIG 1* peut-être subdivisé en quatre niveaux : *la structure logique du profil*, *le contenu du profil*, *la sémantique de la structure logique* et *la sémantique du contenu*.

La structure logique présente la structure générale d'un profil. Cette structure est sous la forme d'une hiérarchie d'éléments réutilisables (classe *EltRéutilisable*) permettant de caractériser un profil. Cette hiérarchie est un arbre dont les noeuds intermédiaires sont soit des profils, soit des catégories d'éléments de profils (classe *NoeudCatégorie* : par exemple l'attribut *préférences utilisateurs* peut-être composé

des attributs *langue, taille et date*.) et les feuilles (classe *NoeudFeuille*) sont tout simplement des attributs auxquels on peut affecter des valeurs. Un élément réutilisable peut donc être : un *Profil*, un *NoeudCatégorie* ou un *NoeudFeuille*.

Le contenu des feuilles d'un profil (classe *Elément*) sont des listes de couples *valeur-poids.* Ces listes peuvent contenir un seul couple *valeur-poids* (attribut de type monovalué comme la *taille* d'un document) ou plusieurs couples *valeur-poids* (attribut de type multivalué comme les *mots clés* d'un document).

De façon générale, les profils dérivés du modèle générique peuvent être :

- *réutilisables et partageables* : un sous arbre d'un profil peut avoir la structure d'un autre profil existant provenant éventuellement d'une autre application. Par exemple, un profil utilisateur peut-être composé : de ses différents profils d'usage (ou profils court terme) et/ou de profils qui le décrivent dans un contexte environnemental particulier (poste de travail du bureau, poste de travail de la maison, téléphone portable, etc.) ;

- *multi facettes* : les profils peuvent être analysés sous différents angles (ensemble d'attributs, sous profils). Ainsi, chaque profil ou attribut ou combinaison de profils ou d'attributs de profils peut constituer une facette de l'utilisateur ;

- *adaptables et évolutifs* : tous les attributs d'un profil donné ne sont pas forcément renseignés. Un profil peut être partagé et enrichi par différentes applications qui utilisent tout ou partie du profil. De plus, nos profils peuvent être modifiés et peuvent évoluer dans le temps.

L'intérêt de l'utilisation d'un modèle générique pour définir un type de profil donné est que la structure de base qu'il propose peut être utilisée par tout type d'application afin de définir tout type de profils (Chevalier *et al*., 2004), (Tchienehom, 2004). La figure *FIG. 2* présente des instances de notre modèle générique de profils décrivant principalement la structure logique (ou taxonomie) et le contenu d'un profil individuel d'utilisateur et d'un profil d'information.

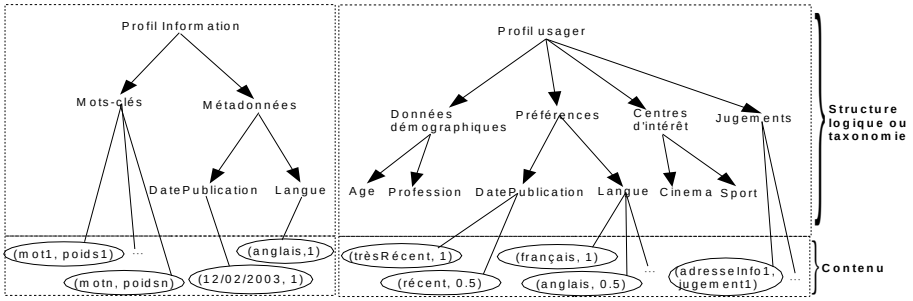

FIG. 2 – Exemples de profils d'information et d'usager : structure et contenu

Notre modèle générique va également nous permettre d'expliciter la sémantique de la structure logique d'un profil et de son contenu. *La sémantique de la structure logique* de notre modèle générique explicite ce que représente un profil ainsi qu'un attribut de ce profil. La sémantique d'un *profil* est la description d'une ressource (information ou usager) dans un contexte donné. Ainsi, les profils peuvent être relatifs aux utilisateurs (individu ou groupe), aux informations mises à disposition (parties de documents, documents, collections, etc.), etc. Notons qu'un profil peut être également : de court terme (profils construits sur une période courte) ou de long

terme (profil construit sur une période relativement importante) (Widyantoro et al. 99), positif ou négatif (Hoashi et al. 00). La figure *FIG. 3* illustre des instances de types de profils avec les liens sémantiques (instances de la classe d'association *LienSémantiqueEntreRessources*) qui les relient.

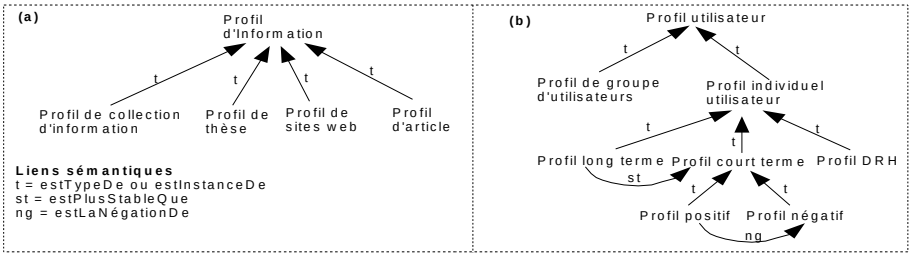

FIG. 3 – Exemples d'instances de la sémantique de ressources

La sémantique d'un *attribut* va permettre d'expliciter la caractéristique que représente l'attribut. La figure *FIG. 4* illustre un exemple d'instance de la sémantique d'attributs de profil.

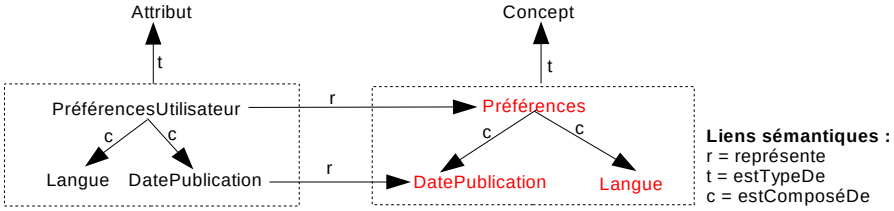

FIG. 4 – Exemple d'instance de la sémantique d'attributs de profil

La sémantique du contenu d'un profil permet d'expliciter le modèle de représentation ou type (instance de la classe *TypeElément*) des éléments de contenu (*cf.* par exemple les types de données de XMLSchéma). La figure *FIG. 5* illustre des exemples d'instances de la sémantique d'éléments de contenu associés aux attributs feuilles : *DatePublicationArticle* et *PréférencesUtilisateurEnDate-Publication*. Ces deux attributs ne sont pas représentés dans le même référentiel (ou espace vectoriel) mais il serait cependant intéressant de pouvoir déduire que l'on peut tout de même les comparer en faisant une transformation de format de représentation (passer du type JJ/MM/AAAA au type AAAA) et un changement de base (de l'espace vectoriel) de représentation. Cet exemple montre l'intérêt qu'il y a à expliciter la sémantique des contenus d'attributs.

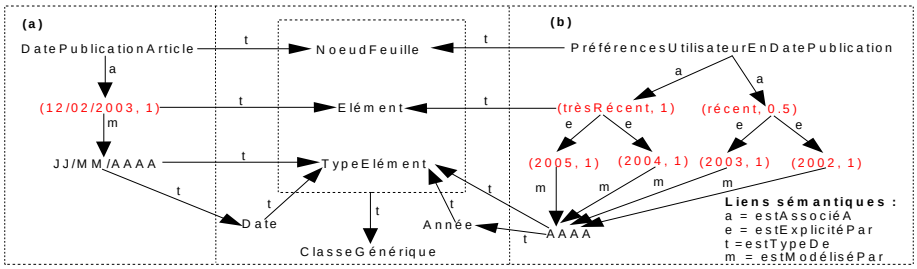

FIG. 5 – Exemples d'instances de la sémantique du contenu d'attributs feuille

Notons que l'organisation des différents attributs par catégorie (classe *NoeudCatégorie*) d'attributs permet de regrouper les attributs similaires dans une même classe et de définir ainsi une nomenclature (ou taxinomie) de ces attributs de façon, généralement, indépendante des ressources (*cf.* les métadonnées du Dublin Core par exemple). Cette taxinomie est déjà une façon de définir la sémantique ou du moins une partie de la sémantique des attributs d'un profil. Cependant cette sémantique est limitée et est fortement liée à l'application qui crée le profil. Ceci rend difficile la coopération entre profils décrits par des taxinomies différentes. Pour affiner cette sémantique et permettre une coopération optimale entre profils nous avons ajouté à notre modèle une dimension sémantique qui permet de s'abstraire d'une structure logique prédéfinie lors de la description ou de la comparaison de profils. Nous avons également explicité la sémantique du contenu des profils.

A partir du modèle générique et sémantique de profil proposé, nous pouvons dériver la structure de différents profils en appliquant des règles de décomposition sur des attributs mais également nous pouvons dériver une sémantique plus fine des profils, des attributs et de leur contenu. Ceci va faciliter la coopération entre profils différents afin de déduire automatiquement des couples d'attributs de même sémantique ou de sémantiques compatibles via certaines transformations.

Nous avons choisi un langage orienté description qui va nous fournir un cadre formel pour combiner des instances de notre modèle générique de profil et formaliser des règles de déduction automatique de couples d'attributs appariables, c'est-à-dire de sémantiques compatibles, entre deux structures logiques de profils disjointes. Cet aspect de la *sémantique* est décrit et illustré dans la section suivante.

3.2. *Graphe sémantique de profils pour la détermination d'attributs appariables*

Pour apparier deux profils différents décrivant des instances de ressources différentes dans des taxinomies (noms d'attributs) différentes, il faut pouvoir déterminer les couples d'attributs feuilles appariables entre ces profils. Pour cela, il faut définir la sémantique de chaque attribut feuille et celle de leur contenu ainsi que des règles permettant de déduire ces couples. La sémantique des attributs feuilles va expliciter la caractéristique représentée par l'attribut tandis que la sémantique de son contenu va décrire le modèle de représentation de ce dernier. Nous avons utilisé la notion de *triplet RDF* pour expliciter formellement les règles d'inférence automatique de paires d'attributs appariables. Pour cela, nous avons construit un graphe sémantique qui combine des instances de profils dérivés du modèle générique de la figure 1. Toute relation sémantique dans notre graphe est donc définie sous la forme d'un triplet de la forme : *[sujet, prédicat, objet]*.

La figure *FIG 6* présente un extrait de description de la sémantique de certains attributs de profils d'information et d'utilisateur. Cet extrait met en exergue l'intérêt du web sémantique pour la description (Dolog et al. 2003) et surtout l'appariement de profils. Ce graphe peut-être vu comme une ontologie de tâche pour l'appariement de profils. Les ressources de notre graphe sémantique sont :

- des types prédéfinis de concepts décrivant la caractéristique représentée par un attribut ;
- des classes représentant des attributs descriptifs et le contenu des profils (structure logique et contenu) ;

- des classes donnant des informations supplémentaires sur le modèle de représentation du contenu des attributs feuilles comme : des instances de la classe *TypeElément* (unité de mesure utilisée pour l'évaluation de l'attribut : nombre de termes ou d'octets pour l'attribut *taille* par exemple ; format de représentation de la valeur de l'attribut : texte, numérique, formats de date ; etc.), le référentiel de représentation ou espace vectoriel (liste des valeurs des éléments de contenu d'un attribut feuille donné), etc.

Pour décrire les relations sémantiques, nous avons dû définir certains prédicats comme : *représente* (noté *r*) pour expliciter la caractéristique représentée par un attribut, *estComposéDe* (noté *c*) pour traduire le lien de composition (entre attributs et/ou profils), *estExplicitéPar* (noté *e*) pour définir le référentiel d'un élément de contenu, etc. Ces prédicats sont complétés par des prédicats RDF/RDFS/OWL qui permettent de typer les différents éléments d'un triplet (*rdf :type* pour subsumer une classe*, rdfs :subClassOf* pour subsumer toutes les instances d'une classe, etc.) et définir des contraintes sur des *classes génériques* (*owl :disjointWith* pour traduire la disjonction entre deux classes, etc.). Des exemples de triplets de la figure *FIG 6* sont par exemple : *[Cinéma, représente, Sujet], [DatePublication, rdf :type, Concept], [2003_ID, estModéliséPar, AAAA]*, etc.

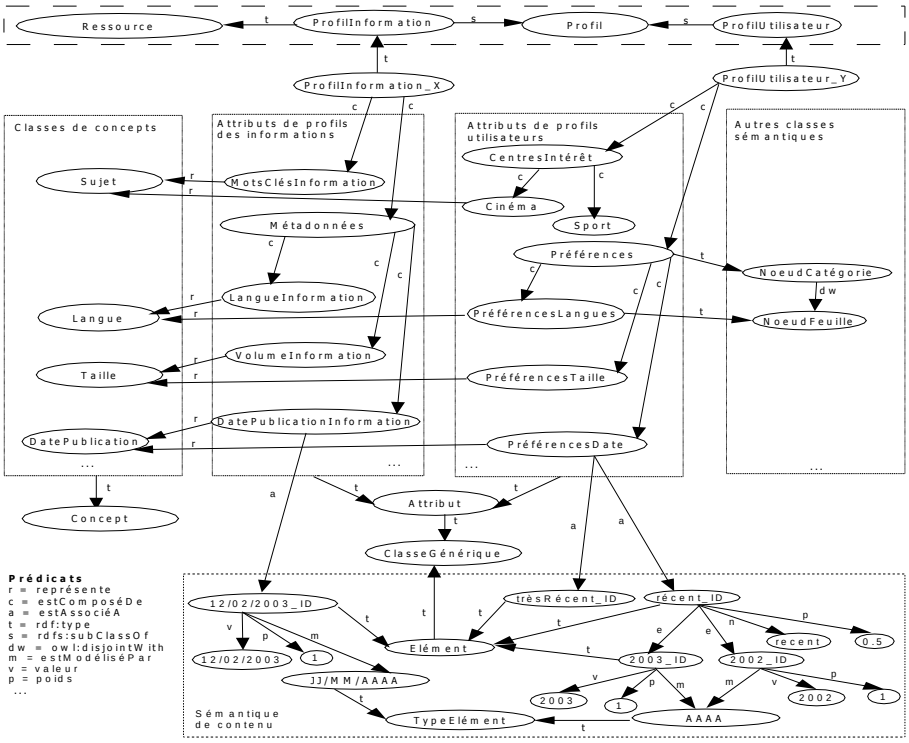

FIG 6 – Extrait d'un graphe sémantique combinant des instances de profils

Soit A l'ensemble des attributs de profils, *C* l'ensemble prédéfini de concepts, *G* l'ensemble des triplets du graphe sémantique, *E* l'ensemble des instances de la classe *Elément, T* l'ensemble des instances de la classe Type*Elément* et *P*

l'ensemble des prédicats, deux attributs feuilles sont appariables si les règles suivantes sont vérifiées :

1. *Règle nécessaire :* les deux attributs doivent être des *NoeudFeuille* et doivent décrire le même concept (Sujet, Langue, Taille, etc.) c'est-à-dire qu'ils sont reliés à la même classe sémantique de type *concept* par le prédicat « *représente »* (noté *r*). Ainsi, dans la figure *FIG 6*, les attributs *MotsClésInformation* et *Cinéma* sont appariables tandis que les attributs *PréférencesLangues* et *VolumeInformation* ne le sont pas. Formellement, la règle nécessaire pour pouvoir envisager un appariement entre deux attributs *x* et *y*, notée *Règle_Nécessaire(x,y),* s'écrit :

Soient x, y ∈ A, x et y sont appariables ssi, [x,rdf :type,NoeudFeuille] et [y,rdf :type,NoeudFeuille] ∈ G et ∃ a∈ C : [x,r,a] et [y,r,a] ∈ G

2. *Règle nécessaire et suffisante :* pour effectuer un appariement entre deux attributs, il faut vérifier que ces attributs ont les mêmes *liens sémantiques* ou *prédicats* (en terme de nombre et de type) qui partent de ces attributs vers les mêmes classes sémantiques. Si les liens sémantiques sont les mêmes mais ne sont pas toujours définis vers les mêmes classes, il faut vérifier si l'objet des triplets des attributs est une instance de la classe *Elément*. Si c'est le cas, il faut rechercher la sémantique (classe *TypeElément*) des instances de la classe *Elément* et vérifier qu'il s'agit de la même classe ou qu'il existe une règle de transformation entre les instances de la classe *TypeElément ainsi trouvé*. Par exemple dans la figure 6, pour pouvoir apparier l'attribut *DatePublicationInformation* et l'attribut *PréférencesDate* il faut qu'il existe une règle permettant de passer de la classe JJ/MM/AAAA à la classe AAAA. Par la suite, il faut vérifier la dimension (nombre de valeurs associées au contenu de l'attribut) de chaque contenu d'attribut ainsi que son référentiel (liste des valeurs). Si le référentiel n'est pas le même c'est-à-dire qu'il y a une disjonction entre les listes de valeurs, il faut effectuer un changement de référentiel (changement de base de l'espace vectoriel), du référentiel de dimension la plus petite vers celui de dimension la plus élevée. S'il y a inclusion de référentiel, il faut juste effectuer un changement de dimension, de la dimension la plus petite à la dimension la plus grande.

Formellement, la règle nécessaire et suffisante pour pouvoir effectuer un appariement entre deux attributs *x* et *y*, notée *Règle_Nécessaire_et_Suffisante(x,y),* s'écrit :

Soient *x*, *y* ∈ A, *x* et *y* sont appariables si et seulement si, *∀ [x,p,a]∈ G, ∃ [y,p,b]∈ G : p ∈ P* et :

1. *a=b*
2. ou si *a, b ∈ E* alors *recherche(a ,a1) et recherche(b, b1)* qui renvoient *a1* et *b1 ∈ T* tels que s'il existe *RègleTransformation(a1, b1) alors*
   a. si *A1* (liste des valeurs de l'attribut x) et *B1* (liste des valeurs de l'attribut y) sont disjoints, il faut effectuer un changement de référentiel. Par exemple, dans la figure 6, il faudra exprimer le contenu de l'attribut *DatePublicationInformation* (initialement exprimé dans le référentiel « *12/02/2003* ») dans le référentiel de l'attribut *PréférencesDate* qui est : « *trèsRécent », « récent »*.
   b. si $A1 \subset B1$ ou inversement, il faut juste effectuer un changement de dimension

où recherche(a ,a1) est une méthode permettant d'obtenir la classe *a1* instance de la classe *TypeElément* des éléments *a* et *b,* et *RègleTransformation(a1, b1)* une règle permettant la transformation d'un triplet *[x1,rfd :type,a1]* en un triplet *[x1,rdf :type,b1]* ou inversement.

Notons que la méthode *recherche(a ,a1)* avec $a \in E$ est définie récursivement comme suit :

a. *soit* $\exists$ *[a,estModéliséPar,a1]* et *[a1,rdf :type,TypeElément]* $\in G$

b. *soit si* $\exists$ *[a,estExplicitéPar,v1]* $\in G$ avec $v1 \in E$ *alors recherche(v1,a1)*

Pour apparier deux attributs, il faut vérifier la cohérence de leur sémantique (caractéristique représentée, sémantique du contenu). A cet effet, nous avons identifié des règles de transformations pour des attributs qui n'auraient pas exactement la même sémantique (modèle de représentation par exemple). Parmi ces règles, nous pouvons citer : la transformation d'attributs *monovalués* en attributs *multivalués* pour le changement de dimension, le changement de référentiel, la transformation *de dates* en divers « formats de date », etc.

L'intérêt d'un graphe sémantique d'attributs de profils comme celui de la figure *FIG 6* est qu'il permet de donner les noms que l'on souhaite aux attributs de nos profils sans perturber l'appariement. Il suffit de rattacher ces attributs aux classes permettant de préciser leur sémantique. De plus, on pourra aussi procéder à l'appariement de profils issus de différentes applications et/ou décrits par des taxinomies différentes. Notre graphe sémantique va pouvoir également être manipulé (mises à jour, interrogations, etc.) à travers des langages d'interrogation RDF (Haase et al. 2004). Pour déterminer les couples d'attributs appariables, on peut utiliser un *parseur* ou *analyseur* RDF qui, étant donné un document RDF, renvoie tous les triplets *[sujet, prédicat, objet]* de ce document. C'est l'ensemble des triplets obtenus qui est analysé afin de déterminer les couples d'attributs appariables.

Dans la section suivante, nous décrivons notre méthode d'appariement de profils et analysons la qualité de cette méthode aux travers d'expérimentations.

3.3. *Appariement de profils*

Les profils peuvent être décrits par plusieurs attributs. L'appariement de profils est lié aux appariements des attributs feuilles décrivant ces profils. La combinaison de certains de ces appariements va permettre de sélectionner les résultats correspondant aux utilisateurs ou tout simplement de ré-ordonnancer ces résultats. Pour apparier des attributs de profils différents, il faut que ces attributs aient des sémantiques compatibles (type des valeurs des attributs, modèle de représentation, caractéristique représentée, etc.). Notre approche de combinaison d'appariements de profils, que nous décrivons dans la section suivante, va permettre de calculer un degré de ressemblance d'une information à un usager (individu ou groupe) pour un ensemble d'attributs donné.

Nous avons classifié les appariements d'attributs feuilles de profils comme suit :

1. *appariement de type booléen :* cet appariement est utilisé lorsque les attributs à apparier sont mono-valués. Le résultat de ce type d'appariement est binaire ;

2. *appariement de type RI :* ce type d'appariement est utilisé lorsque au moins l'un des attributs à apparier est multi-valué. Dans ce cas, on représente les différents attributs à apparier dans un même espace vectoriel dont la dimensionnalité est donnée par la taille du vocabulaire (ensemble des valeurs décrivant les attributs). A chaque vecteur de valeurs ou termes, noté par exemple $d=(t_1,t_2,...,t_n)$, est associé un vecteur de poids réel ou booléen, noté $p=(w_{d,t1},w_{d,t2},...,w_{d,tn})$ qui permettra de calculer un degré de similarité entre attributs à apparier avec la formule du *cosinus* par exemple ;

Comme exemples de ce type d'appariement, on peut citer : l'appariement entre des mots-clés d'un document et une requête où les poids sont généralement calculés avec les formules de *tf* et *tf.idf*, l'appariement entre la langue d'un document et les langues préférées de l'utilisateur., etc. Les tableaux 1 et 2 illustrent respectivement des appariements de type RI dans le cas d'un changement simple de dimension et d'un changement de base de l'espace vectoriel pour les attributs *langue* et *date* respectivement.

Attribut *langue*				
Base	**Français (t_1)**	**Anglais (t_2)**	**Espagnol (t_3)**	**Similarité du cosinus**
Poids du Document : p_d	$w_{d,t1}$=0	$w_{d,t2}$=1	$w_{d,t3}$=0	$sim(p_d,p_u)$= 0.4364
Préférences utilisateur : p_u :	$w_{u,t1}$=1	$w_{u,t2}$=0.5	$w_{u,t3}$=0.25	

TAB 1 – Appariement avec changement de dimension

Attribut *date*					
Base u : t_i	**trèsRécent (t_1)**		**récent (t_2)**		**Similarité**
Base d : v_i	**2005 (v_1)**	**2004 (v_2)**	**2003 (v_3)**	**2002 (v_4)**	
poids p_d* dans *d	*0*	*0*	*1*	*0*	
poids p_u* dans *d	*1*	*1*	*1*	*1*	
poids p_d* dans *u	$w_{d,t1}$=0		$w_{d,t2}$=1		$sim(p_d,p_u)$= 0,4472
poids p_u* dans *u	$w_{u,t1}$=1		$w_{u,t2}$=0.5		

TAB 2 – Appariement avec changement de base

Le problème qui se pose est de pouvoir évaluer la similarité entre deux profils lorsque ces derniers sont décrits par plusieurs attributs. Nous proposons, pour cela, une méthode qui combine différents appariements effectués sur les attributs feuilles de ces profils.

3.3.1 Méthode de combinaison d'appariements

Pour combiner différents types d'appariements, ces derniers doivent tout d'abord être effectués séparément. Ainsi, chaque résultat d'appariement (ou combinaison de résultats d'appariements) va représenter un facteur potentiel de sélection ou d'ordonnancement des informations. On peut donc décrire les informations recherchées sous forme de listes de facteurs (ou appariements) notées : $a=(f_1,f_2,...,f_n)$. A chaque liste d'appariements décrivant une information va correspondre un vecteur de résultats de ces appariements effectués entre des couples

d'attributs appariables de profils. Ainsi, soit *u* et *d* des profils à apparier, le vecteur des résultats des appariements entre ces profils est noté $p_{d,u}=(v_{d,u,f1},v_{d,u,f2},...,v_{d,u,fn})$. Une sous-liste de *a* peut-être utilisée pour la sélection (on la note a_s) et une autre (ou la même) pour l'ordonnancement (on la note a_o) des informations. Un exemple de liste de facteurs peut être : *correspondance aux besoins de l'usager* (sujet du document, des granules, de la collection), *compatibilité aux préférences en langue de l'usager*, etc.

De plus, à chaque utilisateur ou groupe d'utilisateurs ou pour l'ensemble de la population des usagers, on va associer un vecteur de poids pour une liste de facteurs donnée a' (a' pouvant être a_s ou a_o). Ce vecteur de poids est noté $p_{a',x}=(w_{f1},w_{f2},...,w_{fn})$ et décrit le pouvoir discriminant (ou l'importance) des facteurs les uns par rapport aux autres. Ainsi, w_{fj} est le poids ou l'importance du facteur f_j. Afin de calculer les valeurs des w_{fj}, des ordres de préférences doivent être donnés pour tous les éléments du vecteur $p_{a',x}$. Considérons les ordres de préférences d'un utilisateur, pour une liste de facteurs donnée, définit dans le tableau *TAB 3*. La méthode de calcul des éléments du vecteur $p_{a',x}$ est donnée par : $\alpha_i=\beta\sum_{j>i}\alpha_j$

où α_1 et β sont pré-définis et α_i représente le poids des facteurs d'ordre de préférence *i* (car plusieurs facteurs peuvent avoir le même ordre de préférence). Ainsi, si on a *k* ordres de préférences, on aura *k-1* équations à *k-1* inconnues à résoudre. Pour déterminer les α_i on peut utiliser la méthode du pivot de Gauss. Notons que si on souhaite que les α_i appartiennent à l'intervalle [0,1], α_1 doit être fixé à 1.

Vecteur de facteurs *a'*	f_1	f_2	f_3	f_4	f_5	…	f_n
Ordres de préférences *i*	1	1	2	3	3	…	k
Poids des facteurs $p_{a',x}$	$W_{f1}=\alpha_1$	$W_{f2}=\alpha_1$	$W_{f3}=\alpha_2$	$W_{f4}=\alpha_3$	$W_{f5}=\alpha_3$	…	$W_{fn}=\alpha_k$

TAB 3 – Ordres de préférences et poids des facteurs de sélection ou d'ordonnancement des informations

On peut donc calculer un *poids de sélection* p_s et/ou un *poids d'ordonnancement* p_o pour chaque information. Ce poids sera une fonction de $p_{d,u}$ et $p_{a',x}$. Ces poids peuvent être évalués à l'aide de la formule de la moyenne pondérée qui est définie, dans ce contexte, comme suit : $f(p_{d,u},p_{a',x})=\frac{\sum_j v_{d,u,fj}.w_{fj}}{\sum_j w_{fj}}$.

L'utilisation d'une moyenne pondérée, plutôt que celle d'une somme pondérée, permet de normaliser la valeur des poids de sélection ou d'ordonnancement à l'intervalle des valeurs possibles pour les résultats d'appariements $v_{d,u,fj}$. Ceci permettra de comparer les appariements ou combinaisons d'appariements entre eux.

Pour la sélection des informations, il sera nécessaire de définir un seuil pour décider si la correspondance d'une information à un utilisateur est assez significative. Pour cela, on peut analyser la distribution des scores (poids de

sélection ou d'ordonnancement des informations) et définir ce seuil par expérimentations.

Notons que la liste complète de tous les appariements possibles *a* est généralement déterminée en fonction de l'ensemble des attributs de tous les profils définis pour une application donnée. Les sous-listes a_s et a_o et les ordres de préférences de leurs facteurs sont déterminés soit manuellement par les utilisateurs, soit par l'application qui les évalue pour les différents usagers. Notons également que l'appariement d'un nœud intermédiaire (*Profil ou NoeudCatégorie*) d'une structure de profil donné correspond à la combinaison des appariements de ces nœuds feuilles.

Étant donné que tous les attributs feuilles d'un profil ne sont pas toujours renseignés par des valeurs, il va se poser le problème des valeurs nulles et de leur gestion dans la restitution d'informations adaptées aux usagers.

Influence des valeurs nulles

Tous les attributs feuilles des profils ne sont pas toujours renseignés. Il se pose donc le problème de la gestion des valeurs nulles. En cas de valeur nulle, deux choix s'offrent :

- considérer la valeur nulle comme un zéro (ce qui correspond au pire des cas) et dans ce cas, le dénominateur des formules de poids reste le même ;
- ne tenir compte que des valeurs renseignées et dans ce cas, le dénominateur de la formule du poids ne prend pas en compte le poids des facteurs pour lesquels la valeur (résultat d'appariement) est nulle.

Si p_p est le poids obtenu en remplaçant la valeur nulle par la valeur 0 et p_n le poids obtenu en tenant compte uniquement des valeurs renseignées, on aura toujours : *numérateur*(p_p)=*numérateur*(p_n) *et dénominateur*(p_p)>*dénominateur*(p_n).

On aura donc toujours $p_p < p_n$. Ceci signifie qu'en tenant compte uniquement des valeurs renseignées, on augmente les chances de restituer un document dont certains attributs ne seraient pas renseignés.

Nous décrivons, dans la section suivante, les expérimentations que nous avons menées sur la combinaison de profils avec les collections «ATS 94», «LeMonde 94» et «LaTimes 94». du programme CLEF.

3.3.2 Expérimentations

Le tableau *TAB 4,* présente une description des différentes collections utilisées pour nos expérimentations. Notons que tous les attributs descriptifs des documents des collections (titre, date, auteur, etc.), fournis par la DTD de ces collections, ne sont pas forcément tous renseignés et ne portent pas toujours le même nom ou ne sont pas représentés de la même façon, d'où l'intérêt de la définition d'un graphe sémantique pour identifier les attributs appariables et effectuer correctement les différents appariements moyennant certaines transformations.

Collections	***ATS 94***	***LeMonde 94***	***LaTimes 94***
taille en octets	82.1 Mo	156 Mo	420 Mo
nombre de documents	42 710	44 008	112 967
langue	Français	Français	Anglais

TAB 4 – Description des collections ATS 94, LeMonde 94 et LaTimes 94

Nous avons utilisé les profils de trois utilisateurs décrivant leurs préférences en *langue, taille et date* pour les informations à leur restituer. Pour chaque profil, on a exécuté les 50 requêtes de la campagne *CLEF 2001*. Le but est de mesurer qualitativement la divergence ou la convergence de nos résultats par rapport aux résultats qui seraient obtenus en se basant uniquement sur l'appariement relatif à la requête formulée et au contenu des documents (*sujet du document*).

Avec la description des documents et des utilisateurs nous avons identifié une liste d'appariements. Cette liste, avec les ordres de préférences considérés, est présentée dans le tableau *TAB 5*.

Appariements	*Sujet du document (1)*	*Langue (2)*	*Taille (3)*	*Date (4)*	*Sujet du Titre (5)*	*Sujet de la collection (6)*
Préférences	1	2	3	4	5	6

TAB 5 – Ordres de préférences des facteurs de sélection ou d'ordonnancement

La figure *FIG 7* présente une étude comparative des 30 premiers documents obtenus pour nos différents utilisateurs sur les 50 requêtes de la campagne CLEF 2001. Sur l'axe des abscisses, les étiquettes représentent respectivement les appariements ou combinaisons d'appariements suivants : sujet du document (*F1*) ; sujet du document et langue (*F12*) ; sujet du document, langue et taille (*F123*) ; sujet du document, langue, taille et date (*F1234*) ; sujet du document, langue, taille, date et sujet du titre du document (*F12345*) ; sujet du document, langue, taille, date, sujet du titre du document et sujet de la collection à laquelle appartient le document (*F123456*). Les différentes figures (*FIG 7* (a), (b) et (c)) représentent respectivement les courbes de moyenne : *rappel-précision moyen des utilisateurs*, *rappel-précision par utilisateur*, *des moyennes de variation de rang* (sur les 30 premiers résultats) pour les 50 requêtes de la collection CLEF 2001. La variation de rang est la différence entre la position d'un document dans l'ensemble des 30 premiers résultats de l'appariement *F1* et sa position dans l'ensemble des 30 premiers résultats d'un autre appariement (*F12*, *F123*, etc.), s'il y est toujours présent.

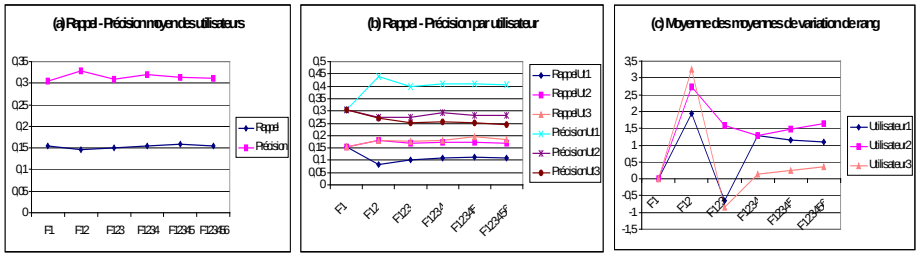

FIG 7 – Résultats qualitatifs des recherches des différents utilisateurs avec les poids des différents appariements calculés pour $\beta=2$ et $\alpha_1=1$ en fonction du tableau TAB 6

Analyse des résultats

Les résultats obtenus avec nos trois utilisateurs montrent que la combinaison de différents appariements dans l'ensemble des 30 premiers résultats fait varier les mesures de rappel et de précision des résultats, calculés relativement à l'appariement *F1* mais agit surtout sur l'ordonnancement des résultats. Cependant, la figure *FIG 7* (a) montre que la précision est généralement au-dessus de la précision initiale (*F1*) tandis que pour le rappel ce n'est pas toujours le cas. En général, on peut dire que

même s'il arrive de perdre en rappel ou en précision (car la courbe de précision n'est pas toujours croissante), cette perte est généralement faible. On constate que les différences de rappel par rapport au rappel initial (*F1*) varient entre –7,4E-03 et 5,7E-03 tandis que les différences de précision varient entre 6,4E-03 et 2,3E-02. Cette faible variation s'explique par la méthode de calcul de poids des appariements (les α_i) qui permet de maximiser les facteurs les plus importants.

L'ordre des résultats dans cet ensemble peut être modifié de façon significative, par rapport à l'ordre qui existait dans les résultats de l'appariement classique de RI (noté ici *F1*), afin de correspondre davantage aux préférences de l'utilisateur. Ainsi, on note des variations de rang pouvant aller jusqu'à des moyennes d'environ 2 à 4 positions parmi les 30 premiers documents. On peut noter que lorsque la courbe des variations de rang croît, cela signifie qu'un nombre important de documents voit leur position augmenter dans le classement des résultats et lorsqu'elle décroît, cela signifie plutôt qu'un nombre important de documents voit leur position dans le classement diminuer. En résumé, la prise en compte de différents appariements permet d'ajuster les résultats aux « préférences » de l'utilisateur tout en veillant à ce que l'on ne perde pas trop en précision et en rappel.

4. Conclusion

Dans cet article, nous proposons une modèle générique de profil qui permet de décrire la structure, le contenu et la sémantique de tout type de profils afin d'optimiser la coopération entre ces derniers. Nous construisons également un graphe combinant deux instances de profils et qui permet de déduire les couples d'attributs appariables entre ces instances via des règles d'inférence. Enfin, nous définissons une méthode d'appariement qui permet de calculer un degré de ressemblance entre deux profils.

Les premières expérimentations nous ont permis d'évaluer qualitativement l'intérêt de notre approche, cependant il reste à la valider dans un cadre applicatif réel.

En termes de perspectives, nous comptons donc étendre nos expérimentations et procéder à une validation de nos propositions dans un cadre concret d'application d'accès à l'information (au sein d'une communauté par exemple) pour mesurer la satisfaction réelle des utilisateurs à travers leur feedback. Nous envisageons également concevoir un outil d'aide à la construction de profils (structure, contenu et sémantique) en exploitant les contraintes explicitées dans le modèle générique.

Bibliographie

Baeza-Yates R., Ribeiro-Neto B., *Modern Information Retrieval*. First edition, Addison Wesley, ISBN 0-201-39829-X, 1999.

Balabanovic M., Shoham Y., Fab : Content-Based, Collaborative Recommendations. *Communications of the ACM*, vol. 40, n°. 3, p. 66-72, 1997.

Belkin N. J., Croft W. B., Information Filtering and information Retrieval : Two Sides of the same Coin? *Communications of the ACM,* vol. 35, n° 12, p. 29-38, 1992.

Berners-Lee T., Hendler J., Lassila O., The semantic web. *Scientific American*. 2001.

Boughanem M., Chrisment C., Soulé-Dupuy C., Query modification based on relevance backpropagation in adhoc environment. *Information Processing \& Management Journal*, Elsevier Science, vol. 35, p. 121-139, 1999.

Bottraud J. C., Bisson G., Bruandet M. F., An Adaptative Information Research Personnal Assistant. *In proceedings of Workshop AI2IA (Artificial Intelligence, Information Access and Mobile Computing) IJCAI'03,* 2003.

Chevalier M., Soulé-Dupuy C., Tchienehom P. L., A profile-based architecture for a flexible and personalized information access. *IADIS International Conference*, vol 2, p. 1017-1022, IADIS - ISBN 972-99353-0-0, 2004.

Cho Y. H., Kim J. K., Kim S. H., A personalized recommender system based on web usage mining and decision tree induction. *Expert System with Applications*, vol. 23, n° 3, p. 329-342, 2002.

Cullot N., Parent C., Spaccapietra S., Vangenot C., Des SIG aux ontologies géographiques, *Revue internationale de géomatique,* 2003.

Dolog P., Nejdl W., Challenges and benefits of the Semantic Web for User Modelling. *In proceeding of AH'03*, 2003.

Goldberg D., Nichols D., Oki B. M., Terry D., Using Collaborative Filtering to weave an Information Tapestry. *Communications of the ACM, Information Filtering*, vol. 35, n° 12, p. 61-70, 1992.

Good N., Schafer J., Konstan J., Borchers A., Sarwar B., Herlocker J., Riedl J., Combining Collaborative Filtering with Personal Agents for Better Recommendations. *In Proceedings of AAAI,* vol. 35, p. 439-446, AAAI Press, 1999.

P. Haase, J. Broekstra, A. Eberhart, R. Volz, A comparison of RDF Query Languages. *In proceedings of the third International Semantic Web Conference ISWC'04*, 2004.

Hoashi K., Kazunori M., Naomi I., Hashimoto K., Document filtering Method using non-relevant information profile. *In proceedings of the 23rd Annual International ACM-SIGIR Conference on research and development in information Retrieval*, p. 176-183, 2000.

Kobsa A., Generic User Modelling Systems. *User Modelling and User-Adapted Interaction*, vol. 11, p. 49-63, 2001.

Konstan J. A., Miller B. N., Maltz D., Herlocker J. L., Gordon L. R. and Riedl J., Grouplens: Applying Collaborative Filtering to Usenet News. *Communication of the ACM*, vol. 40, n°3, p. 77-87, 1997.

Korfhage R.R., *Information storage and retrieval*. Wiley computer publishing, ISBN 0-471-14-338-3, 1997.

Krulwich B., LifeStyle Finder : Intelligent User Profiling Using Large-Scale Demographic Data. *AI Magazine*, vol.18, n° 2, p. 37-45, 1997.

Montaner M., Lopez B., Rosa J. L. D. L, A Taxonomy of Recommender Agents on the Internet, *Artificial Intelligence Review*, vol. 19, pages 285-330, Kluwer Academic Publishers, 2003.

Pazzani M., Muramatsu J., Billsus D., Syskill & Webert : Identifying interesting web sites, *In Proceedings of the Thirteenth National Conference on AI*, p. 54-61, 1996.

Pazzani M., A Framework for Collaborative, Content-Based and Demographic Filtering, *Artificial Intelligence Review,* 1999.

Pitkow J., Schütze N., Cass T., Cooley R., Turnbull D., Edmonds A., Adar E. and Breuel T., Personalized Search : A contextual computing approach may prove a breakthrough in personalized search efficiency, *Communications of the ACM*, vol. 45, No. 9, p. 50-55, 2002.

Rucker J., Polanco M. J., Siteseer : Personalized Navigation for the Web, *Communications of the ACM*, vol. 40, n° 3, p. 73-75, 1997.

Tchienehom P., Architecture de Recherche et de Recommandation d'Information à base de Profils : définitions, acquisitions et usages de profils, *22ième Congrès National Inforsid'04*, p. 143-159, 2004.

Widyantoro D. H., Ioerger T. R., Yen J., An Adaptative Algorithm for learning Changes in User Interests. *In Proceedings of the Eighth International Conference on Information and Knowledge Management (CIKM'99)*, p. 405-412, New York, ACM Press, 1999.