Interopérabilité de Profils pour l'Accès à des Ressources

Max Chevalier[1,2], Chantal Soulé-Dupuy[1,3], Pascaline L. Tchienehom[1,3]

[1]IRIT, 118 route de Narbonne, F-31062 Toulouse cedex 4
[2]IUT A, Université Paul Sabatier, 133B avenue de Rangueil, F-31077 Toulouse
[3]Université Toulouse 1, Place Anatole France, F-31042 Toulouse Cedex

{Max.Chevalier,Chantal.Soule-Dupuy,Pascaline.Tchienehom}@irit.fr

RÉSUMÉ. *Dans cet article nous appliquons une sémantique interprétable par machine à la définition de profils pour l'accès à des ressources. Le but est d'expliciter la sémantique des éléments descriptifs d'un profil et de proposer ou d'utiliser des moyens de déduction automatique d'informations sémantiques implicites. De plus, la sémantique décrite n'est pas limitée à un cadre applicatif prédéfini. Elle doit permettre de faire interopérer des modèles de profils issus d'applications différentes. Pour cela, nous proposons un modèle de profil qui possède une double dimension : générique et sémantique. Nous définissons également une méthode d'analyse d'instances de profils basée sur leur sémantique afin de déduire automatiquement les couples d'éléments de sémantique compatible (appariables) qui existent entre deux profils que l'on souhaite comparer.*

ABSTRACT. *This paper deals with a machine understandable semantics for profiles definition in resources access. The goal is to clarify the semantics of descriptive elements of profiles and to define or use means for an automatic deduction of implicit semantic information. Moreover, the defined semantics is not limited to a particular application. It should allow interoperability between profiles models coming from different applications. For this purpose, we propose a profile model which guarantees a double dimension: generic and semantic. Moreover, we define an analysis method based on semantics in order to infer elements pairs of compatible semantics that exist between two profiles that one wishes to compare.*

MOTS-CLÉS : *Interopérabilité, profil, accès à des ressources, modèle générique, sémantique.*

KEYWORDS: *Interoperability, profile, resources access, generic model, semantics.*

Catégorie : Chercheur

1. Introduction

Les développements que connaissent aujourd'hui Internet, et en particulier le World Wide Web, mais également les intranets et tous les environnements numériques de travail conduisent à la mise à disposition d'une masse sans cesse croissante de ressources. Une ressource dans ce contexte peut être de nature très variée comme un moyen matériel ou logiciel, une information (document par exemple), une collection d'informations, un usager, etc. Les besoins d'interopérabilité ou de coopération entre ressources, dans les applications d'accès à ces ressources, mettent en évidence un certain nombre de problèmes liés en particulier à l'hétérogénéité de leurs descriptions. Par exemple, un fournisseur de services Y doit envoyer une information I sous forme de MMS au dispositif mobile d'un utilisateur X. Pour cela, il devrait pouvoir déterminer si le dispositif mobile de ce dernier possède les applications requises à la réception et à la visualisation de MMS. Dans le cas contraire, il devrait transmettre l'information sous-forme de SMS par exemple. Dans ce cadre, une solution serait de trouver une manière homogène de décrire le besoin d'un côté et les ressources de l'autre par des métadonnées susceptibles de permettre un appariement. Or, la diversité et l'hétérogénéité des ressources qui sont amenées à coopérer engendrent nécessairement une grande disparité dans la façon de les décrire. Il faut donc être en mesure d'exploiter cette disparité en analysant et en interprétant les descriptions de ressources.

Dans un contexte collaboratif par exemple, on peut être amené à rechercher des utilisateurs similaires pour un partage d'informations en fonction du contenu et/ou des jugements effectués sur ces derniers. De plus, on peut souhaiter échanger ces informations via des dispositifs mobiles (PDA, téléphone portable, etc.) ou non. Pour cela, on doit généralement combiner différentes descriptions d'utilisateurs, d'informations et de dispositifs d'échange, de communication ou de stockage qui vont constituer ainsi les ressources clés de l'application. L'application doit manipuler ces différentes ressources (usagers, informations, matériels, logiciels, etc.) pour répondre de façon personnalisée ou non aux demandes de chaque usager ou groupe d'usagers. Les applications d'accès à des ressources sont donc confrontées aux problèmes d'intégration (ou coopération) de ressources hétérogènes pour la réalisation d'une tâche spécifique.

Ainsi, afin de faciliter la coopération, et donc la description de ressources hétérogènes, il est nécessaire de définir des modèles de ressources qui aient à la fois des propriétés *d'extensibilité, de flexibilité, de ré-utilisabilité et d'interopérabilité*. Pour cela, une sémantique doit être associée à la description des ressources. Cette sémantique doit permettre de faire coopérer des modèles différents de façon cohérente, en se basant sur des langages de métadonnées. Par exemple, des utilisateurs qui s'intéressent à des « informations récentes » peuvent avoir chacun leur propre définition de cette notion. Ainsi, les résultats des recherches de ces derniers doivent être fortement liés à l'interprétation de la sémantique qu'ils y auront associée.

Dans le but de pallier l'hétérogénéité de description des ressources, des solutions ont été élaborées et consistent, en général, à définir des modèles génériques dont le but est d'offrir un référentiel et cadre homogène de description de ressources. Il

n'en reste pas moins que ces référentiels doivent être exploitables, pouvoir évoluer et s'adapter.

L'objectif de cet article est donc de définir un cadre de description et d'exploitation de profils basés sur la sémantique pour pallier ce problème d'hétérogénéité. Après un état de l'art (section 2) sur les modes d'accès et de description de ressources, nous présentons dans la section 3, notre modèle de description de ressources (ou modèle de profil) qui possède une double dimension : *générique et sémantique*. L'aspect générique permet d'avoir un modèle sous-jacent homogène et la sémantique, quant à elle, va atténuer les disparités qui subsistent au niveau des instances de profils pour une plus grande interopérabilité de ces derniers au sein d'applications. Notons que la sémantique est définie de façon générique pour pouvoir être instanciée sur des exemples de profils. Par la suite, dans la section 4, nous définissons une méthode d'exploitation de profils basée sur l'analyse de la sémantique, pour l'appariement de profils dans les processus d'accès à des ressources.

2. Revue de littérature

L'accès à des ressources est une vision large de l'accès à l'information (Baeza-Yates *et al.*, 1999) où une ressource peut-être étendue à toutes sortes de personnes, choses ou actions : usager individuel ou groupe d'usagers ; usager dans un contexte environnemental donné (situation géographique, environnement matériel et logiciel) ; documents mis à disposition (articles, thèses, etc.) ; collections ou parties de documents ; composants matériels ou logiciels ; etc.

De nombreuses applications d'accès à l'information existent pour aider l'utilisateur à trouver des informations qui sont mises à sa disposition. Ce sont principalement des applications de recherche (Pitkow *et al.*, 2002), (Boughanem *et al.*, 1999) ou filtrage (Montaner *et al.*, 2003), (Pazzani, 1999) d'informations. Ces méthodes sont basées sur des modèles sous-jacents de ressources qu'elles manipulent et que nous appelons profils. Ces modèles sont hétérogènes et, en général, dans les techniques traditionnelles d'accès à des ressources, la sémantique de ces profils est considérée comme étant implicite (ou elle est fortement liée à l'application qui les a définis). Ainsi, les profils sont généralement décrits par des modèles : de types *attribut-valeur* (Schilit *et al.*, 1994) où les attributs ne sont pas structurés les uns par rapport aux autres ; basés sur la définition *d'une structure logique* des attributs et *d'un contenu associé* (Bouzeghoub *et al.*, 2005), (Chang *et al.*, 2004). Ces deux types de modèles ne donnent aucune information explicite sur la sémantique des profils et ceci limite voire rend impossible toute coopération entre profils issus d'applications différentes. Pour résoudre les problèmes d'hétérogénéité et d'interopérabilité de ressources, on a besoin de modèles extensibles, flexibles, ré-utilisables et interopérables (Berners-Lee *et al.*, 2001). Ces propriétés peuvent être obtenues avec des modèles génériques (Kobsa, 2001) et sémantiques (Dolog *et al.*, 2003) de profils. Par exemple, il existe des modèles pour la description sémantique de contexte utilisateur à travers les capacités de leurs dispositifs de connexion comme : CC/PP (Composite Capability/Preference Profiles) (Klyne *et al.*, 2004) et CSCP (Comprehensive Structured Context Profiles) (Buchholz *et al.*, 2004). L'idée

de base est d'utiliser un langage ou des langages de métadonnées qui vont servir de ponts entre deux descriptions de ressources différentes.

La particularité du modèle que nous proposons est qu'il n'est pas conçu pour une classe prédéfinie de ressources. De plus, nous définissons aussi une méthode d'exploitation flexible de profils pour l'accès à des ressources basée sur l'analyse de la sémantique de ces profils qui permet de déduire automatiquement les couples d'éléments de sémantique compatible (similaire) entre des profils que l'on souhaite comparer. Cette recherche de similarité permet de s'affranchir d'une similarité implicite (définie a priori) mais également elle permet de réduire les incohérences et les silences liés à une recherche de similarité basée uniquement sur la structure logique, surtout si l'on se projette au niveau inter-applications.

3. Modélisation de profil

Dans cette section, nous décrivons notre modèle générique et nous présentons également une instanciation du modèle qui illustre et explicite ses avantages.

3.1. *Modèle générique de profil*

Afin de définir des profils qui soient extensibles, flexibles, ré-utilisables et inter-opérables, nous proposons un modèle générique de profil qui intègre une dimension sémantique. Il a été conçu pour la description de ressources non pré-définies.

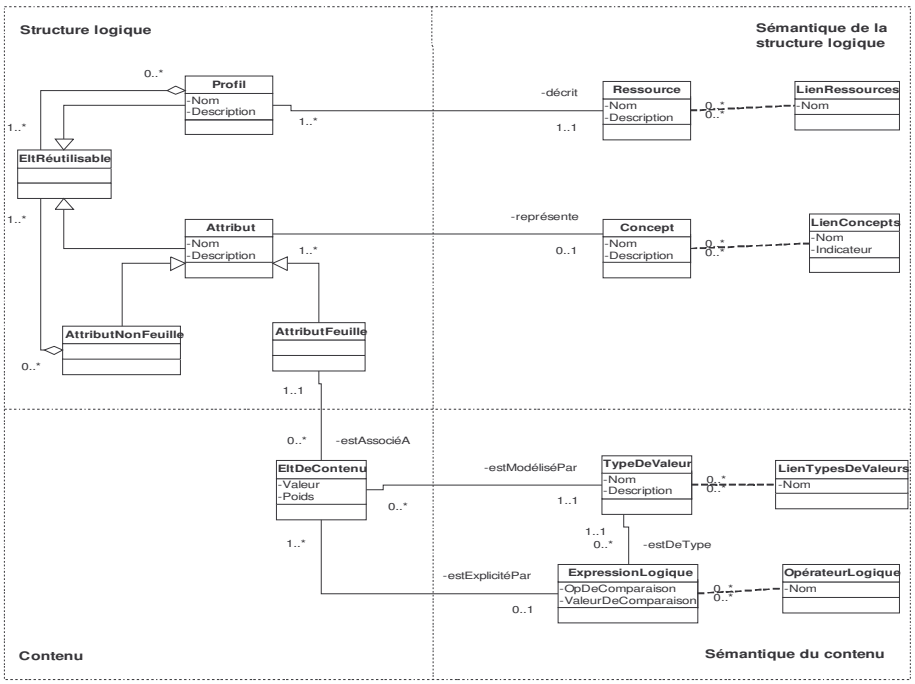

Figure 1. *Modèle générique de profil*

La *figure 1* présente le modèle générique de profil proposé. Il est décrit en utilisant UML (User Modelling Language) (Muller et al., 2000) car ce langage nous fournit un degré d'abstraction intéressant pour décrire des modèles génériques. Notre modèle résulte de l'analyse de différents types de modèles génériques de profils (Chang *et al.*, 2004), (Schilit *et al.*, 1994). Contrairement à ces systèmes, notre modèle est assez général pour décrire différentes classes de profils tout en assurant une séparation claire entre la structure logique, le contenu et la sémantique des profils. Ainsi, le modèle générique de la *figure 1* est subdivisé en quatre niveaux : *la structure logique du profil, le contenu du profil, la sémantique de la structure logique du profil et la sémantique du contenu du profil*.

3.1.1. *Structure logique*

La structure logique présente la structure générale d'un profil. Cette structure est sous la forme d'une hiérarchie d'éléments réutilisables (instances de la classe *EltRéutilisable*) permettant de caractériser un profil. Cette hiérarchie est un arbre dont les noeuds sont soit des profils (instances de la classe *Profil*) existants, soit des attributs (instances de la classe *Attribut*) qui décrivent les caractéristiques d'un profil de l'arborescence (ancêtre de type « profil », le plus proche de l'attribut dans l'arborescence). Il existe deux types d'attributs : les attributs pouvant être des noeuds intermédiaires, qui sont des instances de la classe *AttributNonFeuille* de la hiérarchie du profil, et qui permettent de représenter des catégories ou classes d'attributs (par exemple, l'attribut *préférencesUtilisateurMobile* peut être composé des attributs *formatMessage*, *langue, taille et date*) ; et les attributs qui sont des feuilles (instances de la classe *AttributFeuille*) de la structure logique et auxquels on peut affecter des éléments de contenu (ou valeurs). Un élément réutilisable peut donc être : *un Profil, un AttributNonFeuille ou un AttributFeuille*.

3.1.2. *Contenu*

Le contenu d'un profil (instances de la classe *EltDeContenu*) prend la forme de listes de couples *valeur-poids*. Ces listes peuvent contenir un seul couple *valeur-poids* (attribut de type monovalué comme *la taille d'un écran*) ou plusieurs couples *valeur-poids* (attribut de type multivalué comme les différentes *applications* d'un dispositif mobile). La *valeur* ici est le contenu réel de l'attribut et le *poids* est une donnée numérique qui décrit à quel point la *valeur* représente l'attribut. Par exemple, si un utilisateur préfère recevoir des MMS plutôt que des SMS sur son téléphone portable, on devrait définir une pondération qui représente cette préférence.

3.1.3. *Sémantique associée à la structure logique*

La sémantique de la structure logique de notre modèle générique explicite ce que représente un profil ainsi qu'un attribut de ce profil. Ainsi :

- la sémantique d'un profil est la description d'une ressource (information ou usager, par exemple) 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, thèses, etc.), etc. L'intérêt de cette classe (classe *Ressource*) est qu'elle permet la ré-utilisation de structures génériques existantes pour la description de modèles spécifiques de profils ;

- la sémantique d'un attribut, quant à elle, va permettre d'expliciter la caractéristique générique que représente l'attribut (instance de la classe *Concept*). Ainsi, des attributs vont être reliés à des concepts génériques qui sont généralement issus de langages de métadonnées (standards ou normes) existants comme : Dublin Core, etc. Par exemple, l'attribut *langue* d'une information pourra être relié à la métadonnée du Dublin Core *dc:language*.

3.1.4. *Sémantique associée au contenu*

La sémantique du contenu d'un profil permet d'expliciter le modèle de représentation ou type de données (instance de la classe *TypeDeValeur*) des éléments de contenu (*cf.* par exemple, les types de données de XMLSchema). A titre d'exemples, on peut citer les types : entier, chaîne de caractères, date, etc. ; les types dérivés comme les *patterns* pour décrire : des numéros de sécurité sociale, des codes postaux, des dates, etc.

La sémantique de contenu va permettre également de préciser le sens d'un élément de contenu donné au travers d'expressions logiques. Par exemple, on va pouvoir exprimer le fait qu'un usager s'intéresse à des informations publiées après une certaine date et avant une autre date donnée. On va donc pouvoir combiner plusieurs expressions logiques via les opérateurs logiques : ET, OU.

La sémantique est représentée dans notre modèle générique par les classes : *Ressource, Concept, TypeDevaleur, ExpressionLogique* ainsi que par les classes d'associations (*LienRessource, LienConcepts, LienTypesDeValeurs et OpérateurLogique*) qui les relient et qui explicitent le lien sémantique (*subsomption, égalité, équivalence, etc.*) qui existe entre des instances des classes précédemment citées. Cette sémantique est basée sur des langages de métadonnées (Dublin Core, RDF, RDFS, OWL, XMLSchema, etc.) et va définir ainsi un *langage pivot* d'interprétation de ressources les unes par rapport aux autres.

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 pour définir différentes classes de profils. Des instances de parties du modèle générique proposé sont présentées dans (Tchienehom, 2005). Par contre, la sémantique du contenu dans le présent article est décrite différemment et permet ainsi de prendre en compte des valeurs floues, par l'usage d'opérateurs de comparaison et d'opérateurs logiques. Des instances de la prise en compte de ces valeurs floues sont décrites dans (Chevalier *et al.*, 2005). Dans la section suivante, nous décrivons les caractéristiques d'instances complètes de profils et nous présentons également les spécificités d'une interopérabilité de profils basée sur la partie sémantique du modèle générique.

3.2. *Instanciation et interopérabilité de profils*

UML est un langage semi formel qui nous a permis d'avoir un meilleur rendu visuel de notre modèle générique. Par contre, pour décrire les instances de ce modèle générique nous avons choisi les formalismes RDF/RDFS/OWL qui sont des langages formels qui s'inspirent de la logique de description. Ils sont davantage adaptés pour la description de la sémantique car ils nous fournissent des prédicats de base que nous pouvons ré-utiliser : disjonction (*owl:disjointWith*), équivalence (*owl:equivalentClass*), égalité (*owl:sameClass*), généralisation (*rdfs:subClassOf* et

rdf:type), etc. De plus, l'usage de RDF/RDFS/OWL nous permet, par la suite, de valider expérimentalement notre modèle en utilisant des interfaces de programmation existantes pour le web sémantique qui sont capables d'interpréter et de raisonner sur des représentations RDF/RDFS/OWL. Par ailleurs, notons que UML et RDF ne sont pas des langages disjoints. La brique de base de RDF qui est le triplet *[sujet, prédicat, objet]* existe de façon implicite en UML et dans tous les langages. Ainsi, les associations entre classes et les relations entre une classe et ses propriétés dans un modèle UML peuvent être explicitées par des triplets RDF.

3.2.1. *Instanciation de profils*

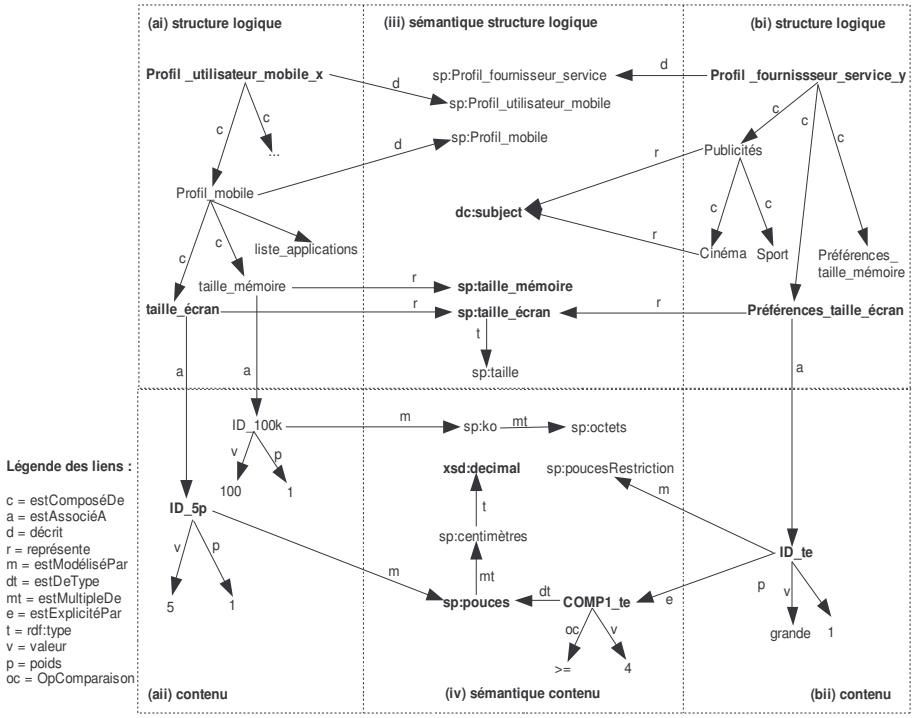

Figure 2. *Profils : structure logique, contenu et sémantique*

La *figure 2* illustre deux profils : le profil d'un utilisateur mobile à travers son dispositif mobile (figure 2a) et le profil d'un fournisseur de services vers des utilisateurs de dispositifs mobiles (figure 2b). Pour chaque profil nous décrivons sa structure logique (figures 2ai et 2bi), son contenu (figures 2aii et 2bii) et la sémantique associée (figures 2iii et 2iv) au travers d'un graphe RDF.

La structure logique et le contenu d'un profil sont toujours décrits sous la forme d'un arbre. La structure de graphe n'est obtenue que lorsque l'on rajoute des éléments de sémantique à la structure logique ou au contenu. Par exemple, on peut avoir plusieurs éléments de la structure logique d'un profil qui vont être rattachés au même élément sémantique. De plus, les éléments de sémantique de la structure logique peuvent être également reliés entre eux. Ainsi dans la *figure 2*, nous avons

les attributs *Publicités* et *Cinéma* du profil *Profil_fournisseur_service_y* qui représentent le même concept générique, issu du Dublin Core, *dc:Subject*.

Dans ces instances, nous montrons également la ré-utilisabilité de métadonnées existantes dans la description d'un profil. Par exemple :

- la ré-utilisation de métadonnées du Dublin Core pour la définition de la sémantique associée aux attributs de la structure logique. Ainsi, les attributs *Publicité* et *Cinéma* représentent le concept *dc:Subject* ;

- la ré-utilisation de métadonnées de *XMLSchema* pour la définition du type des valeurs des éléments de contenu. Par exemple, *xsd:decimal* qui est le type de nombres décimaux est utilisé pour décrire le contenu de l'attribut *taille_écran*.

La ré-utilisation de métadonnées permet de faciliter l'interprétation de modèles car elle limite la re-définition de concepts ou types de données existants et donc réduit également la création de liens sémantiques d'équivalence ou d'égalité entre concepts ou entre types de données. Notons que le préfixe *« sp: »* représente l'espace de noms associé au vocabulaire (noms d'associations, instances de classes et de classes d'associations) de notre modèle générique.

3.2.2. *Interopérabilité de profils*

La *figure 2* illustre également l'interopérabilité de profils basée sur la sémantique de leur structure logique et de leur contenu. Cette interopérabilité est décrite à travers les attributs feuilles : *Préférences_taille_écran* qui décrit les préférences en taille d'écran d'un fournisseur de service donné et l'attribut *taille_écran* qui décrit la taille d'écran d'un dispositif mobile donné. Si l'on regarde la sémantique de ces attributs (sémantique de la structure logique), l'on constate qu'ils représentent le même concept générique *sp:taille_écran* qui est défini dans un espace de noms préfixé par *« sp: »*. Par contre, si on compare la sémantique de leur contenu, elle n'est pas identique. Les préférences en taille d'écrans du fournisseur de services sont décrites sous forme de restrictions (ou contraintes) sur des tailles d'écrans, tandis que la taille d'écran du dispositif mobile est représentée en pouces (unité de mesure des tailles d'écrans). Ce qui serait intéressant ici, c'est de pouvoir analyser cette sémantique et de déduire que l'on va pouvoir comparer ces attributs moyennant des transformations qui auraient été clairement identifiées : transformation de type de données si nécessaire, changement de la combinaison linéaire (modèle vectoriel) de représentation des éléments de contenu. Nous allons définir formellement, dans les sections suivantes, ce principe d'analyse de profils.

En résumé, l'interopérabilité (ou coopération) entre profils est liée à l'analyse de la structure logique, du contenu et de la sémantique des profils ainsi qu'à l'usage de règles d'inférence qui permettent de déduire les éléments de ressemblance qui existent entre des profils. L'optimisation de cette interopérabilité implique :

- une analyse de la sémantique rattachée à la structure logique car des attributs feuilles qui représentent le même concept peuvent être décrits par des éléments de contenu de types différents. Par exemple, la taille mémoire d'un dispositif mobile peut être mesurée en kilo octets tandis que les préférences en taille mémoire d'un fournisseur de service peuvent être décrites en méga octets. Il est donc nécessaire ici de procéder à des transformations de types avant de comparer le contenu de ces deux attributs ;

- une analyse de la sémantique rattachée au contenu des profils à apparier car des attributs feuilles dont les contenus sont de mêmes types ne décrivent pas toujours la

même caractéristique générique. Par exemple, la *langue* d'une information et les *thèmes publicitaires* d'un fournisseur de services peuvent avoir tous les deux leur contenu décrit par des chaînes de caractères. Cependant, l'appariement entre ces deux attributs génèrera une incohérence car ces deux attributs ne décrivent pas des concepts similaires (*dc:language vs dc:subject*, par exemple).

Décrire la sémantique de la structure logique et du contenu des profils est donc une étape nécessaire à une exploitation optimale et cohérente de ces derniers.

4. Exploitation de profils pour l'accès à des ressources

L'exploitation de profils pour l'accès à des ressources est principalement basée sur l'appariement de profils. Pour garantir un appariement cohérent, il faut pouvoir détecter les couples d'éléments de sémantique compatible qui existent entre les profils que l'on souhaite comparer. Pour cela, nous avons défini tout d'abord une architecture générale d'exploitation de profils.

4.1. *Architecture générale d'exploitation de profils*

L'architecture générale d'exploitation de profils sur laquelle nous allons nous baser pour analyser et apparier des profils dans des processus d'accès à des ressources comporte différents types d'informations :

- des espaces de noms existants (dublin core, rdf, rdfs, owl, xmlschema) et un espace de noms associé au vocabulaire de notre modèle générique nommé *SemanticProfile_NameSpace* et préfixé par *« sp: »* ;
- des taxinomies ou ontologies globales (indépendantes des instances de profils) qui aideront à l'analyse d'instances de ressources, concepts, types de valeurs ou valeurs de contenu de profils définies par des métadonnées ou reliées via des métadonnées. Ces ontologies permettront de définir des liens sémantiques (subsomption, égalité, équivalence, synonymie, etc.) entre éléments de profils ou entre métadonnées. Notons que ces ontologies peuvent être le résultat de l'union de plusieurs fichiers d'ontologies *:*
- des instances de profils à apparier ;
- une librairie de règles de transformations de types de données ;
- des index de contenu de profils construits sur les principes classiques d'indexation en recherche d'information (mots clés issus du contenu par exemple).

Dans la section suivante, nous décrivons l'algorithme général d'analyse de profils, afin d'en déduire les éléments de sémantique compatible que l'on va pouvoir comparer. Pour cela, nous avons utilisé l'API java *Jena* développée dans le cadre du web sémantique avec le langage d'interrogation *SPARQL* (Prud'hommeaux *et al*., 2005) qui permet de faire des inférences sur des documents RDF.

4.2. *Détermination d'éléments appariables : algorithme général*

L'étape préalable à l'appariement est celle de la détermination de couples d'éléments de sémantique compatible entre les profils à comparer. La *figure 5* illustre l'algorithme général pour cette analyse avec ses différentes étapes, les paramètres en entrée et ceux obtenus en sortie. Les étapes peuvent être subdivisées

en trois classes : les étapes de parcours de graphe de profils pour la recherche d'éléments spécifiques (étapes 1, 3 et 5), les étapes de vérification de compatibilité entre certains éléments de profils (étapes 2, 4 et 6) et l'étape 7 des transformations.

En entrée de cet algorithme, nous avons les deux profils à comparer et en sortie nous obtenons une liste de couples d'attributs appariables avec pour chacun d'eux, les combinaisons linéaires (ou espaces vectoriels de représentation) du contenu de chacun des attributs du couple. Notons que ces attributs sont des attributs feuilles car ce sont eux qui sont associés à des éléments de contenu. L'appariement d'un profil ou d'attributs non feuilles d'un profil est en fait le résultat d'une agrégation d'appariements d'attributs feuilles qui le composent.

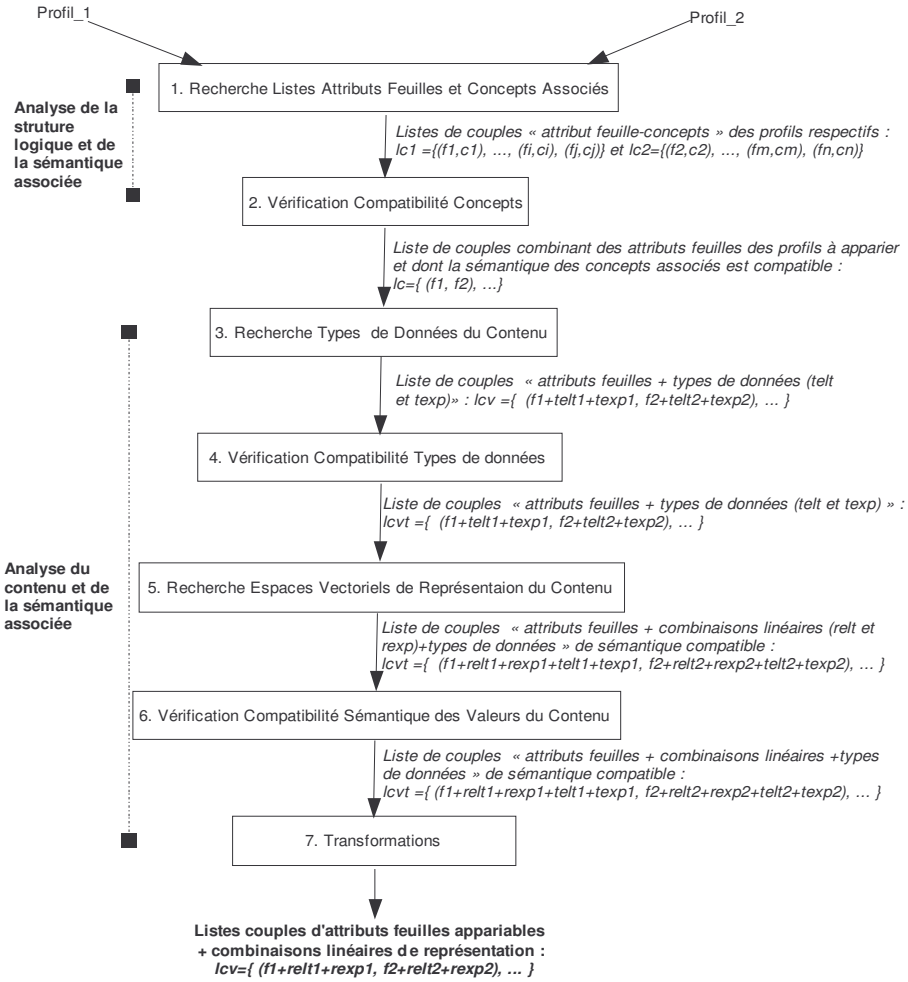

Figure 5. *Algorithme général d'analyse de profils à apparier*

L'analyse de la structure logique et de la sémantique associée permet de déterminer la liste des couples d'attributs feuilles qui représentent des concepts

compatibles. Cette analyse définit la *règle nécessaire* d'appariement entre deux attributs feuilles issus de profils différents. Ensuite, cette règle est complétée par l'analyse du contenu de ces attributs et de la sémantique associée qui permet de s'assurer de la compatibilité sémantique effective (suffisante) des couples d'attributs feuilles des profils à apparier. Les combinaisons linéaires des attributs feuilles obtenues au terme de l'algorithme seront utilisées pour mesurer un degré de similarité pour chaque couple.

4.2.1. *Analyse de la structure logique et de la sémantique associée*

L'analyse de la structure logique et de la sémantique associée consiste à :

1. *déterminer la liste des attributs feuilles et des concepts associés* pour chacun des profils que l'on souhaite apparier ;

2. *vérifier la compatibilité des concepts* associés à tout couple possible d'attributs feuilles des profils à comparer. Lorsque les concepts rattachés aux attributs feuilles sont différents (proviennent de différents espaces de noms ou ont des noms différents), on peut vérifier s'ils sont équivalents ou identiques. Pour cela, on peut utiliser une requête SPARQL similaire à celle du *tableau 1*.

```
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT  ?c1  ?c2
FROM <Concepts.rdf>
WHERE {
        ?c1 rdf:type sp:Concept .
        ?c2 rdf:type sp:Concept .
{ ?c1 owl:equivalentClass ?c2 } UNION { ?c1 owl:sameAs ?c2 } .
FILTER (  (?c1=<cf1> || ?c1=< cf2 >) && (?c2=< cf1> || ?c2=< cf2 >)  ) .
}
```

Tableau 1. *Vérification de la compatibilité entre concepts*

Notons que les propriétés *owl:equivalentClass* et *owl:sameAs* sont définies comme étant symétriques et que c_{f1} et c_{f2} sont les noms de concepts trouvés pour des attributs feuilles *f1* et *f2* des profils respectifs *profil_1* et *profil_2* que l'on souhaite comparer. La symétrie va permettre de déduire automatiquement les triplets : c_2 *owl:equivalentClass* c_1 et c_2 *owl:sameAs* c_1 même si ces derniers n'existent pas explicitement dans le fichier *concepts.rdf*. De plus, le mot clé *UNION,* qui permet de définir une alternative, est employé ici car entre deux concepts donnés, on peut avoir l'une ou l'autre des propriétés *owl:sameAs* (égalité) ou *owl:equivalentClass* (équivalence). Le mot clé *FILTER* permet lui d'effectuer une sélection de triplets.

4.2.2. *Analyse du contenu et de la sémantique associée*

Pour l'analyse du contenu et de la sémantique associée, nous considérons que tous les éléments de contenu d'un attribut feuille sont de même type, c'est-à-dire qu'ils sont tous modélisés par la même instance de la classe *TypeDeValeur*. De la même manière, toutes les instances de la classe *ExpressionLogique* qui explicitent les éléments de contenu d'un attribut donné sont également du même type de données. Ceci s'explique par le fait qu'un attribut feuille est un élément de description élémentaire et donc possède, de ce fait, un contenu homogène. Ainsi, l'analyse du contenu et de la sémantique associée à un couple d'attributs feuilles

nécessairement compatibles (c'est-à-dire que la sémantique de leur concept est compatible) consiste à :

1. *déterminer les types de données associés* à chaque attribut feuille d'un couple nécessairement appariable : l'un est défini par le type des valeurs des éléments de contenu *t_elt_f* et l'autre par le type des valeurs de comparaison des expressions logiques associées *t_exp_f*, si elles existent ;

2. *vérifier la compatibilité des types de données associés* à chaque attribut feuille d'un couple nécessairement appariable : cette vérification consiste à parcourir la librairie des règles de transformations de types de données et vérifier s'il existe, pour deux types de données t_{f1} et t_{f2} des attributs feuilles *f1* et *f2* respectifs, une méthode permettant de transformer une valeur de type t_{f1} en une valeur de type t_{f2} ou inversement. Le nom de la méthode est une concaténation des noms des deux types de données. Notons que l'on peut vérifier cette compatibilité en analysant et en interprétant les liens sémantiques qui existent entre les types de données. Cependant, cette alternative peut s'avérer coûteuse et inutile si la méthode de transformation n'a pas été définie ;

3. *déterminer les combinaisons linéaires* (espaces vectoriels) de représentation des contenus associés à chaque attribut feuille d'un couple nécessairement appariable : l'une est définie par les valeurs des éléments de contenu *v_elt_f* et l'autre par les valeurs de comparaison des expressions logiques associées *v_exp_f*, si ces dernières existent ;

4. *vérifier la compatibilité de la sémantique des valeurs des éléments de contenu ou d'expressions logiques associées* à chaque attribut feuille d'un couple nécessairement appariable : cette vérification permettrait d'augmenter la flexibilité de l'analyse en utilisant la taxinomie globale *ValeursContenu.rdf*, par exemple. Un exemple de ce type de vérification est décrit dans le *tableau 2*. Notons que les propriétés *sp:estUneTraductionDe*, *sp:estUnSynonymeDe* sont définies comme étant symétriques. De plus, val_1 et val_2 sont deux valeurs des éléments de contenu (ou d'expressions logiques) associées aux attributs feuilles *f1* et *f2* des profils *profil_1* et *profil_2* respectivement. L'intérêt de cette vérification est qu'elle permet de définir un espace vectoriel de représentation minimum et *sémantiquement commun* aux attributs à comparer et permet ainsi d'obtenir un résultat de similarité qui traduit plus fidèlement la ressemblance entre les attributs. Par exemple, on peut avoir les valeurs *fr* et *français* qui représentent la même sémantique si *fr* est une abréviation de *français*. Il serait intéressant, dans ce contexte, de représenter ces deux valeurs par le même terme (traduction de la même sémantique).

```
PREFIX sp: <http://www.irit.fr/SIG/.../SemanticProfile/>
SELECT ?v1 ?v2
FROM <ValeursContenu.rdf>
WHERE {
{ ?v1, sp:estUneTraductionDe, ?v2 }
UNION { ?v1, sp:estUnSynonymeDe, ?v2 }
UNION { ?v1, sp:estUneAbréviationDe, ?v2 } .
FILTER (?v1 = <val1> && ?v2 = <val2>) .
}
```

Tableau 2. *Vérification de la compatibilité des valeurs de contenu d'attributs feuilles*

5. *Identification des transformations* à effectuer sur le contenu pour l'appariement : la vérification de la compatibilité de type de données de contenu d'attributs peut conduire à un certain nombre de transformations nécessaires, afin de pouvoir effectuer l'appariement. Par exemple, pour apparier les attributs *taille_mémoire* dont le contenu est en kilo octets et *Préférences_taille_mémoire* dont le contenu est en méga octets, il est nécessaire d'avoir une règle qui permette de passer de la classe du type *kilo octets* à celle du type *méga octets* ou inversement.

Par ailleurs, il faut, en général, vérifier l'espace vectoriel de représentation du contenu de chaque attribut feuille (liste des valeurs de ces éléments) et la dimension de cet espace vectoriel (nombre d'éléments de contenu associés à un attribut feuille). Il peut y avoir soit une *disjonction*, soit une *inclusion* ou un *recouvrement* entre valeurs (ou vecteurs, ou termes) des espaces vectoriels des attributs feuilles à comparer. Afin de procéder à l'appariement, il peut être nécessaire d'effectuer :

- un changement de combinaison linéaire de représentation par *extension des vecteurs de base* afin de représenter les attributs à comparer dans la même dimension ;

- un changement de combinaison linéaire de représentation par *changement des vecteurs de base* si l'un des attributs feuilles à comparer peut être représenté dans différents espaces vectoriels. C'est le cas des attributs ayant des éléments de contenu explicités via des expressions logiques. Dans ce cas, le type des valeurs des éléments de contenu et des valeurs de comparaison de leurs expressions logiques sont généralement différents.

Un exemple de changement d'espace de représentation avec changement des vecteurs de base et extension de la dimension de l'espace vectoriel est illustré dans la section suivante.

4.2.3. *Exemple de transformations de représentation de contenu pour l'appariement*

Soient les deux attributs feuilles de sémantique compatible suivants :

- *Préférences_taille_écran* dont le contenu est *{(normale, 0.5), (grande,1)}*. Avec la valeur *normale* qui représente la restriction aux tailles d'écrans inférieures à *4 pouces* et la valeur *grande* qui représente la restriction aux tailles d'écrans supérieures ou égales à *4 pouces* ;

- *taille_écran* dont le contenu est *(5, 1)*.

Les résultats de l'analyse de contenu donnent deux combinaisons linéaires pour ces deux attributs : l'une associée aux valeurs des éléments de contenu (noté a_{elt}) et l'autre associée aux valeurs des expressions logiques (noté a_{exp}). Ainsi :

- pour l'attribut *Préférences_taille_écran* (noté a_{fs}), on obtient :

$$a_{fsexp} = 1.\overrightarrow{LT4} + 1.\overrightarrow{GE4}$$

$$a_{fselt} = 0{,}5.\overrightarrow{normale} + 1.\overrightarrow{grande}$$

- pour *taille_écran* (noté a_m), on obtient $a_{mexp} = 1.\overrightarrow{GE4}$ après conversion de type de données, et après changement d'espace de représentation on a :

$a_{mexp} = 0.\overrightarrow{LT4} + 1.\overrightarrow{GE4}$ (extension des vecteurs de base)

$a_{melt} = 0.\overrightarrow{normale} + 1.\overrightarrow{grande}$ (changement des vecteurs de base)

Notons que *GE* remplace le symbole « >= » et *LT* le symbole « < ».

Les deux attributs *Préférences_taille_écran* et *taille_écran* seront appariés en utilisant leur représentation relative aux éléments de contenu a_{fselt} et a_{melt} et en appliquant, par exemple, la mesure du cosinus. Une méthode de calcul de poids d'appariements (pouvant être utilisée également pour le calcul de poids d'attributs) et une méthode d'agrégation d'appariements d'attributs pour comparer deux profils sont décrites et évaluées dans (Tchienehom, 2005).

4.3. *Evaluation de l'algorithme d'analyse de profils*

Nous avons évalué l'algorithme d'analyse de profil proposé par rapport à des méthodes d'analyse qui seraient basées sur un modèle de profils de type *attribut-valeur* ou de type *structure logique*.

Pour cela, nous avons défini 10 profils utilisateurs qui décrivent leurs centres d'intérêts, leurs préférences (en dates, langues ou tailles de documents) et leurs données démographiques (nom, sexe, profession). Les profils utilisateurs sont décrits par une structure logique, un contenu et une sémantique et sont composés d'une moyenne de : 55,63 triplets, 27,63 ressources (ou sujets de triplets) différentes et 9,95 prédicats différents. Nous avons également utilisé des documents de la campagne d'évaluation CLEF 2001 qui contient les articles de l'année 1994 des journaux : Agence Télégraphique Suisse (*ATS*), Le Monde (*LeMonde*) et Los Angeles Times (*LaTimes*). Le *Tableau 3* décrit quelques propriétés de ces collections.

Collections	*ATS 94*	*LeMonde 94*	*LaTimes 94*
Taille de la collection	82.1 Mo	156 Mo	420 Mo
Nombre de documents	43 178	44 013	113 005
Langue des documents	Français	Français	Anglais

Tableau 3. *Description des collections ATS 94, LeMonde 94 et LaTimes 94*

La description RDF de la structure logique de ces documents respecte la DTD (Document Type Definition) de leur collection respective. Le contenu a été extrait des documents et la sémantique a été définie en analysant des exemples de documents des différentes collections. Les DTDs de ces trois collections sont différentes et décrivent différents aspects des documents (identification, titre, auteurs, date, paragraphe, etc.). Cependant la sémantique de certains éléments de structure logique de DTD différente est similaire. Par exemple, les auteurs d'articles sont représentés par les mots : *AU* pour la collection SDA, *AUTHOR* pour la collection LeMonde et *BYLINE* pour la collection LaTimes. Ces éléments de structure logique peuvent représenter le concept *dc:creator* du Dublin Core, par exemple.

La base de profils obtenue est assez hétérogène pour mener nos expérimentations. Les expérimentations ont consisté à détecter le nombre de couples d'attributs feuilles de sémantique compatible parmi ces profils. Pour cela, nous avons appliqué notre algorithme (nommé *SemanticProfile*). Ensuite, nous avons considéré le cas du modèle *attribut-valeur* (listes des attributs feuilles des profils de la base) où les attributs de sémantique similaire ont le même nom. Enfin, nous avons

considéré le cas de modèle de profils basé sur une *structure logique* uniquement (structures logiques des profils de la base uniquement) où les attributs de sémantique similaire ont le même nom mais également le même chemin d'accès dans la structure logique. Le *tableau 4* illustre les résultats obtenus pour les différentes méthodes.

Modèles de profils	*SemanticProfile*	*Attribut valeur*	*Structure logique*
Nombre moyen de couples d'attributs de sémantique compatible	4,84	0,92	0,79

Tableau 4. *Résultats de détection automatique d'attributs de sémantique compatible*

On remarque que notre algorithme donne de meilleurs résultats et est donc plus adapté dans un contexte hétérogène. La sémantique ajoutée agit comme une partie partageable entre profils et permet ainsi, grâce à des inférences, de détecter des couples d'attributs qui n'ont pas le même nom mais qui partagent la même sémantique, ce qui n'est pas possible avec les modèles de type *attribut-valeur* ou *structure logique*. Par ailleurs, les attributs qui portent le même nom n'ont pas forcément la même sémantique. Ainsi, grâce à la double analyse de la sémantique de la structure logique et de la sémantique du contenu, notre approche garantit des résultats cohérents. Notre méthode permet donc, dans un contexte hétérogène, de réduire le silence informationnel et les incohérences qui sont généralement inhérentes aux méthodes classiques de représentation et d'exploitation de profils.

5. Conclusion et discussions

Dans cet article nous avons proposé un modèle générique de profils et une méthode d'analyse de ces profils pour en faciliter l'appariement. L'intérêt de ces propositions repose sur l'explicitation et l'usage de la sémantique des éléments descriptifs des profils. Elles offrent ainsi une plus grande flexibilité pour la description et l'appariement de profils. Nous avons testé l'algorithme d'analyse de profils pour l'appariement proposé avec l'API java *Jena* pour le web sémantique. Les résultats montrent que nous obtenons de meilleurs résultats, dans un contexte hétérogène, en comparaison aux méthodes classiques basées sur des modèles de profils de type *attribut-valeur* ou *structure logique*. De plus, l'API Jena permet de combiner les aspects de programmation structurelle à des aspects de raisonnement qui permettent d'inférer des informations implicites sur des documents RDF via l'usage du langage d'interrogation *SPARQL*.

Pour les travaux futurs, nous envisageons de définir des moyens de prise en compte de liens sémantiques pas toujours symétriques dans les vérifications de compatibilité, comme les liens de spécialisation ou généralisation avec les prédicats *rdf:type* et *rdfs:subClassOf* (qui sont des liens transitifs). Par exemple, le remplacement de la valeur *véhicule* par la valeur *voiture* ou inversement devrait sans doute impliquer des modifications de poids pour ces valeurs. De même, la prise en compte de ces liens (spécialisation/généralisation) au niveau des instances de concepts représentés par des attributs, nécessite de définir une procédure particulière

de raisonnement pour leur interprétation dans l'analyse de la sémantique de profils que l'on souhaite apparier.

Bibliographie

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

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.

Bouzeghoub M., Kostadinov D., Personnalisation de l'information: aperçu de l'état de l'art et définition d'un modèle flexible de profils. *COnférence en Recherche d'Informations et Applications (CORIA'05)*, p.201-218, 2005.

Buchholz S., Hamann T., Hubsch G., Comprehensive Structured Context Profiles (CSCP): Design and Experiences. I*n Proceedings of the Workshop on Context Modeling and Reasoning (CoMoRea'04)*, p. 43-47, 2004.

Chang B., Kesselman J., Rahman R., Document Object Model (DOM) Level 3 Validation Specification Version 1.0. *W3C Recommendation*, http://www.w3.org/TR/2004/REC-DOM-Level-3-Val-20040127/, 2004.

Chevalier M., Soulé-Dupuy C., Tchienehom P. L., Profiles Semantics and Matchings Flexibility for Resource Access. *International IEEE conference on Signal-Image Technology & Internet-based Systems (SITIS'05)*, 2005.

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

Klyne G., Raynolds F., Woodrow C., Ohto H., Hjelm J., Butler M. H., Tran L., editors,Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. *W3C Recommendation*, http://www.w3.org/TR/CCPP-struct-vocab/, 2004.

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

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.

Muller P. A., Gaertner N., *Modélisation objet avec UML.* Deuxième édition, Eyrolles, ISBN 2-212-09122-2, 2000.

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.

Prud'hommeaux E., Seaborne A., SPARQL Query Language for RDF. *W3C Working Draft*, http://www.w3.org/TR/2004/rdf-sparql-query/, 2005.

Schilit B. N., Adams N. L., Want R., Context-aware computing applications. *In IEEE Workshop on Mobile Computing Systems and Applications*, 1994.

Tchienehom P. L., Modèle générique de profils pour la personnalisation de l'accès à l'information, *23ième Congrès National Inforsid'05*, p. 269-284, 2005.