Gestion de connaissances personnelles et contextuelles, et respect de la vie privée

Fabien Gandon, Norman Sadeh

▶ **To cite this version:**

Fabien Gandon, Norman Sadeh. Gestion de connaissances personnelles et contextuelles, et respect de la vie privée. 15eme Journées Francophones d'Ingénierie des Connaissances, Conference (IC'2004), May 2004, Lyon, France. hal-00374538v2

HAL Id: hal-00374538

https://inria.hal.science/hal-00374538v2

Submitted on 7 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Gestion de connaissances personnelles et contextuelles, et respect de la vie privée

Fabien L. Gandon[1,2], Norman M. Sadeh[1]

[1] School of Computer Science - Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891, USA
http://www-2.cs.cmu.edu/~sadeh/
[2] Equipe ACACIA - INRIA
2004 Route des Lucioles, 06902 Sophia Antipolis
http://www-sop.inria.fr/acacia/

Résumé : Nous montrons comment les mécanismes de confidentialité peuvent évoluer pour exploiter les modèles orientés ontologie et suivre le déploiement de Webs sémantiques. Testée dans le projet *my*Campus, l'architecture distribuée que nous présentons ici repose sur des règles pour le Web sémantique et le respect de la confidentialité ainsi que des services Web pour représenter les ressources personnelles et publiques. Des agents logiciels gèrent les échanges, notamment les agents *e-Wallets* responsables des ressources des utilisateurs.

Mots-clés : connaissances, contexte, confidentialité, Web sémantique.

1 Tension entre connaissance du contexte et confidentialité

Au cœur de la gestion des connaissances, les problématiques les plus étudiées sont le captage, la mémorisation et la diffusion des connaissances. De même, les technologies telles que le Web sémantique mettent l'accent sur des solutions techniques aux problèmes d'interopérabilité. Parallèlement, le développement des services Web, la multiplication des appareils en réseaux et leur déploiement dans une informatique mobile (Sadeh, 2002), ambiante et ubiquitaire accroissent considérablement les connaissances et les services disponibles en ligne.

Beaucoup de personnes parlent de ces futurs réseaux où les connaissances formalisées nous permettront d'identifier et d'accéder aux informations et aux services de façons beaucoup plus efficaces. Mais qui accepterait de se confier à de tels réseaux et services si les mécanismes de sécurité et de confidentialité ne suivent pas la même évolution? Qui utiliserait un web sémantique qui puisse être employé et abusé, par exemple pour du marketing non sollicité ou autre profilage indiscret?

Peu de personnes parlent de futurs réseaux où des connaissances formalisées permettront aussi de cacher ou de restreindre l'accès à ces services et à ces informations. Pourtant, cela doit être abordé rapidement, ou les webs sémantiques et leurs services risquent de ne jamais recevoir la confiance dont ils ont besoin pour fonctionner à pleine puissance. Nous montrons comment une approche ontologique, améliore la gestion des connaissances privées et de la confidentialité. La modélisation

orientée ontologie permet des inférences pour contrôler l'accès et ajuster la précision des réponses, voire même mentir lorsque cela s'avère moins dangereux que de refuser de répondre. Pour cela, nous détaillerons l'approche suivit dans le projet *my*Campus.

1.1 *my*Campus : système ouvert d'accès mobiles aux services en ligne

Les mécanismes présentés dans les sections suivantes ont été testés dans le projet *my*Campus : un environnement ouvert basé sur les technologies des systèmes multi-agents et du Web sémantique pour implanter une plateforme d'accès mobiles à des services en ligne (Sadeh *et al.*, 2003) (Gandon & Sadeh, 2004). Ces services peuvent utiliser les informations contextuelles et personnelles des utilisateurs et l'architecture assure le respect de leur vie privée. La première version a été validée début 2003 sur le campus de Carnegie Mellon où l'environnement était accessible à travers des agendas électroniques connectés et localisés à travers le réseau sans-fil de l'université. L'expérience de 3 jours a impliqué 11 utilisateurs choisis pour couvrir un large spectre de profils et libres de se connecter depuis n'importe où sur le campus et d'utiliser les deux premiers services développés : le Concierge (recommandation de restaurants) et le Messager (filtrage et routage des messages). Les utilisateurs devaient en plus maintenir leur agenda à jour de façon à assurer une bonne connaissance du contexte tout au long de l'expérience. L'évaluation a montré une acceptation globale positive et la connaissance du contexte a montré une amélioration systématique des résultats par rapport à des profils d'utilisateur statiques. Par exemple, pour l'agent de filtrage des messages, dans 70% des cas la meilleure décision ne peut être prise sans connaissance du contexte (Gandon & Sadeh, 2004b).

Comme le montre la figure 1, la plateforme *my*Campus est un environnement multi-agents où, avec le temps, les utilisateurs souscrivent à différents types d'*agents-services*. Ces agents répondent à une attente et aident à différentes activités (ex : établir une réunion, filtrer des messages, préparer un voyage). Pour opérer, chaque agent va avoir besoin de différentes informations sur son propriétaire et éventuellement sur d'autres utilisateurs. L'accès à une information sur un utilisateur est contrôlé par son *e-Wallet* et les règles de confidentialité qu'il contient. Les agents ne sont pas limités aux ressources personnelles des utilisateurs et accèdent également à des services Web publics, à des documents Web, des ontologies et des descriptions du Web sémantique, *etc.* Dans *my*Campus, ceci inclut l'accès à une variété de ressources et services tels que des services Web de localisation, de météorologie, *etc.*

Le *e-Wallet* archive les connaissances statiques au sujet de l'utilisateur (ex : nom, téléphone, *etc.*). Il sait aussi obtenir plus de connaissances en faisant appel à une variété de ressources (ex : agenda électronique, localisation sur le réseau sans-fil, *etc.*), chacune représentée par un service Web. Cette connaissance est stockée sous forme de règles qui lient différents types de connaissances contextuelles à une ou plusieurs invocations possibles de services. Ces règles permettent au *e-Wallet* d'identifier et d'activer les ressources les plus appropriées pour répondre à une question sur son propriétaire (ex : accéder au calendrier pour découvrir sa disponibilité, utiliser le réseau sans-fil pour le localiser). Des règles de confidentialité, enregistrées dans le *e-Wallet*, sont personnalisées par l'utilisateur et assurent que

chaque type d'information est uniquement révélé aux interlocuteurs autorisés à y accéder dans le contexte courant. Elles ajustent aussi l'exactitude ou l'inexactitude des informations fournies selon les préférences de révision spécifiées par l'utilisateur (ex : ne pas révéler la salle où je suis, mais simplement le bâtiment).

Dans *my*Campus l'accès au système se fait à partir d'agendas électroniques reliés au réseau sans-fil de l'université de Carnegie Mellon. Cependant, cette architecture fonctionne tout aussi bien pour des scénarios avec des postes fixes et plus généralement dans des environnements où les utilisateurs se connectent par différents canaux et dispositifs d'accès. Les informations sur le dispositif et le canal peuvent être traitées comme des attributs du contexte et rendues disponibles via le *e-Wallet*.

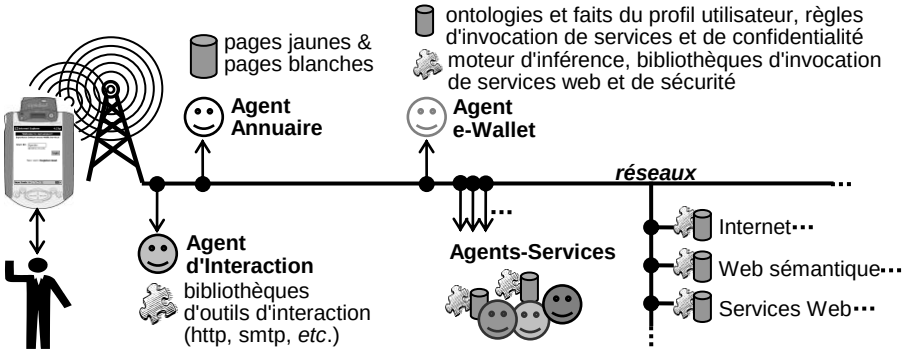

Fig. 1 – Un environnement ouvert pour l'accès mobile aux services en ligne.

Nous avons suivi une architecture (FIPA, 2002) incluant en particulier des *Agents annuaires* permettant à un agent/utilisateur à la recherche d'un service d'obtenir une liste d'agents potentiellement fournisseurs de ce service. Des *Agents d'Interaction* sont responsables des échanges avec l'utilisateur. Cela inclut la gestion des sessions de connexion et des interactions avec les autres agents. Même si nous nous concentrerons sur des scénarios impliquant des utilisateurs individuels, cette architecture s'étend à des scénarios où les utilisateurs sont des organismes entiers ayant chacun un ou plusieurs *e-Wallet*.

La deuxième version du système *my*Campus a été terminée et démontrée fin 2003 sur des scénarios illustrant l'intérêt du *e-Wallet*. A titre d'exemple, un service de cartographie donne à l'utilisateur une carte des environs en utilisant les réponses de son *e-Wallet* : dans le cas où l'utilisateur est disposé à révéler son code postal la carte est plus précise que lorsqu'il est seulement disposé à révéler la ville la plus proche. Plusieurs autres services ont été développés et testés (Gandon & Sadeh, 2004b).

1.2 Travaux connexes sur le contexte et la confidentialité

De plus en plus, les concepteurs de logiciels cherchent à améliorer la facilité et le confort d'utilisation en capturant le contexte dans lequel leurs utilisateurs opèrent afin d'y adapter le comportement des applications. Les contributions au développement d'applications conscientes du contexte sont nombreuses avec parmi les pionnières : Active Badge qui réoriente les appels téléphoniques en fonction de votre localisation

(Want *et al.*, 1995) et ParcTab qui fournit par le biais d'agendas électroniques la liste des ressources à proximité (ex : une imprimante, une note électronique virtuellement laissée dans une pièce, *etc.*) (Schilit, 1995).

Les applications utilisant le contexte sont souvent liées aux sources d'informations contextuelles au cœur même de leur code ; en conséquence, leur maintenance reste coûteuse. Les praticiens du domaine ont réalisé qu'il fallait séparer les mécanismes d'acquisition d'informations contextuelles des services susceptibles de les utiliser. Ainsi, la bibliothèque d'outils de (Dey *et al.*, 2000) permet de rapidement intégrer de façon modulaire des sources d'informations contextuelles tout en les isolant des applications consommatrices. Cette approche est semblable à notre notion de *e-Wallet*, mais nous allons plus loin en intégrant des descriptions sémantiques des ressources personnelles qui permettent ainsi une meilleure interopérabilité et des contrôles plus fins à un niveau sémantique. Ces modèles plus riches sont essentiels pour permettre la découverte et l'accès automatique aux ressources personnelles des utilisateurs par des agents logiciels. Notre architecture permet une intégration en temps réel des ressources personnelles et publiques pour une utilisation immédiate par des agents logiciels sensibles au contexte. Notre implantation transforme ces ressources en services Web identifiables et invocables par des agents logiciels.

Une architecture ouverte pour l'accès à des informations et des services personnels pose immédiatement le problème de la sécurité et de la confidentialité. Les utilisateurs doivent pouvoir contrôler qui a accès à leurs ressources personnelles et dans quel contexte. La notion de *e-Wallet* comme présentée dans le passeport .NET de Microsoft se limite à stocker un nombre limité d'informations sans offrir une réelle flexibilité pour en définir les règles d'accès. Ainsi, les utilisateurs peuvent uniquement indiquer s'ils sont disposés ou non à partager une partie de leur profil avec tous les services proposés sans pouvoir distinguer entre les différents services. Notre *e-Wallet* permet aux utilisateurs de spécifier l'accès à n'importe laquelle de leurs ressources personnelles pour n'importe quel service.

Le *e-Wallet* s'inscrit dans la continuité des efforts récents pour développer des langages plus riches, tels que APPEL (P3P, 2002), permettant de capturer les préférences des utilisateurs en matière de confidentialité. Nous permettons aux ontologies d'être intégrées à ce domaine en les utilisant pour la description des règles de confidentialité qui peuvent alors mobiliser une grande variété d'attributs. Cela permet, en outre, aux utilisateurs d'indiquer des règles de révision par lesquelles ils contrôlent le niveau d'exactitude (ou d'inexactitude) avec lequel leurs informations sont révélées. Vous pouvez accepter de dire à vos collègues dans quelle pièce vous êtes pendant les heures de bureau, ou vous contenter de confirmer que vous être dans l'enceinte de votre lieu de travail, ou même simplement donner le nom de la ville où vous êtes. Cela inclut également des scénarios où vous pourriez vouloir feindre d'être dans un endroit, alors qu'en réalité vous êtes ailleurs ; dans certains cas, refuser de répondre à une question éveille une suspicion toute aussi dangereuse que la réponse à cette question. Un employé de banque qui aurait accès à la salle des coffres mais refuserait de donner sa localisation lorsqu'il est dans cette salle éveillerait rapidement les soupçons par ses refus ponctuels. Si au lieu de refuser il répond qu'il est dans un bureau adjacent, il devient beaucoup plus difficile de savoir qu'il a accès aux coffres.

Enfin, la communauté s'intéressant à la sécurité a développé de puissants langages pour décrire les politiques d'accès tels que (SAML, 2003), (XACML, 2003) et EPAL (Schunter & Powers, 2003), mais ces langages ne tirent pas profit des efforts du Web sémantique comme (OWL, 2003). Nous utilisons OWL pour représenter nos ontologies, les connaissances contextuelles (ex : lieux, activités, les relations au sein d'une organisation, *etc.*) et les préférences de confidentialité qui leurs sont associées. Nous suivons aussi l'évolution des Services Web Sémantiques dans OWL-S et SWS [1] pour automatiser la découverte et l'accès aux services personnels et publics.

2 Le *e-Wallet* : un accès sécurisé aux ressources personnelles

2.1 Le concept de *e-Wallet* et ses fonctionnalités

Le *e-Wallet* est donc un élément central de notre architecture Web sémantique pour la prise en compte du contexte et le respect de la vie privée. Il représente *une interface sémantique unifiée et sécurisée pour les ressources personnelles d'un utilisateur* et permet ainsi aux agents-services de les mobiliser. La connaissance d'un utilisateur gérée dans son *e-Wallet* peut se diviser en quatre catégories :

1. *La connaissance statique*. Cette connaissance inclut typiquement des connaissances indépendantes du contexte (ex : nom, courriel, *etc.*).
2. *La connaissance dynamique*. C'est la connaissance sensible au contexte de l'utilisateur (ex : en conduisant, je ne veux pas recevoir de messages).
3. *Les règles d'invocation de services*. Ces règles permettent d'intégrer au *e-Wallet* des ressources d'information externes, personnelles (ex : agenda) ou publiques (ex : météorologie). Elles ajoutent au *e-Wallet* les capacités d'un annuaire sémantique de services qui peuvent être dès lors automatiquement identifiés et invoqués pour traiter des requêtes. Dans un premier temps, chaque ressource personnelle est transformée en un Service Web Sémantique (ex : une règle peut indiquer qu'une question au sujet de l'activité courante de l'utilisateur peut être répondue en invoquant préalablement le service Web correspondant à son agenda Outlook). Dans un deuxième temps, des règles d'invocation de services relient les types de connaissances contextuelles et des services Web disponibles pour obtenir ses connaissances. Plusieurs règles peuvent fournir la même information, par exemple si le portable de l'utilisateur est allumé et sur le réseau sans-fil c'est une façon de le localiser sinon, son agenda peut suggérer une réponse (ex : une salle de réunion). Enfin, pour répondre à des questions sur l'utilisateur des services publics peuvent également être nécessaires. Par exemple, une question comme "l'utilisateur est-il dans un endroit ensoleillé" exigera typiquement la localisation de l'utilisateur et l'accès à un service public de météorologie.
4. *Préférences de confidentialité*. Ces préférences explicitent quelles informations l'utilisateur est disposé à révéler, à qui et sous quelles conditions. Ces préférences se divisent en deux catégories :

[1] respectivement http://www.daml.org/services/owl-s/1.0/ et http://www.w3.org/2002/ws/swsig/

a. *règles de contrôle d'accès* : ces règles disent qui peut voir quelle information et sous quelles conditions (ex : "ma localisation est accessible à mes collègues seulement pendant les jours de travail entre 8H et 20H")
b. *règles de révision* : souvent les préférences de confidentialité d'un utilisateur ne sont pas booléennes mais impliquent différents niveaux d'exactitude ou d'inexactitude. Une règle de *révision par abstraction* permet de contrôler le niveau de détails d'une réponse en fonction du contexte (ex : indiquer que vous êtes en ville sans donner le lieu exact). Une règle de *révision par falsification* est utilisée dans des scénarios où l'utilisateur ne veut pas que l'on sache qu'il cache une information et préfère fournir une réponse fausse (ex : un utilisateur peut ne pas vouloir indiquer son véritable courriel à un service par crainte de recevoir des publicités non sollicitées).

Tous les types de connaissances présentés ci-dessus (règles y comprises) sont représentés en OWL. Ils requièrent un certain nombre d'ontologies appropriées, par exemple : ontologies des attributs contextuels, des ressources personnelles, ainsi que pour des connaissances de domaine comme les types de cuisines et les préférences culinaires ou les types de messages et les préférences de filtrage

2.2 Un exemple de scénario d'interrogation du *e-Wallet*

Avant de descendre dans les détails techniques du *e-Wallet*, un scénario nous aidera à exemplifier les étapes principales du traitement d'une requête. Imaginons qu'une question soit soumise par un service invoqué par un utilisateur (Norman) au *e-Wallet* d'un autre utilisateur (Fabien) requérant l'endroit où se trouve actuellement ce dernier. Les étapes principales du traitement d'une requête sont les suivantes :

1. *Déclarer le contexte de la requête* : dans un premier temps, un maximum de faits décrivant le contexte de la requête sont affirmés *i.e.* ils sont chargés dans le moteur d'inférence du *e-Wallet* pour être éventuellement utilisés par les inférences qui traiteront la requête. Dans notre scénario, un exemple est que "l'expéditeur de la requête est Norman" ou que "la requête est arrivée à 15H34".
2. *Déclarer les besoins élémentaires en information et les autorisations nécessaires* : ici la requête est scindée en (a) une conjonction de besoins élémentaires en information, ex : "l'endroit où se trouve Fabien" ; et (b) la nécessité d'obtenir une autorisation d'accès pour chacun de ces besoins en information. Le processus d'autorisation est ensuite effectué en deux temps aux étapes 3 et 6.
3. *Pré-vérification des autorisations* : un premier contrôle est effectué pour voir si la question est acceptable dans la limite des connaissances disponibles à ce stade. Dans notre exemple, le *e-Wallet* cherche à vérifier que Norman peut demander à localiser Fabien. Le *e-Wallet* de Fabien inclut une règle de confidentialité indiquant que ses collègues de travail peuvent connaître le bâtiment où il se trouve, lorsqu'il est sur le campus. Le *e-Wallet* cherche alors à prouver que Norman est effectivement un collègue de Fabien en utilisant une base de connaissances sur l'organisation. S'il prouve le contraire, l'*e-Wallet* rejette la question. Dans notre exemple Fabien est un collègue de Norman et la procédure

continue car à ce stade, et parce qu'il n'a pas encore déterminé si Fabien est sur le campus, le *e-Wallet* n'a aucune raison de refuser la requête.

4. *Faire appel à la base de connaissances locale du e-Wallet* : certaines requêtes utilisent des faits de la base de connaissance locale au *e-Wallet*, qui contient des connaissances statiques (nom, courriel, *etc.*) et des connaissances sensibles au contexte ("je ne veux pas de messages quand je conduis"). Dans notre exemple particulier, une telle connaissance n'est pas utile.
5. *Faire appel à des services personnels ou publics* : quand la connaissance locale n'est pas suffisante pour répondre à une question, le *e-Wallet* se tourne vers ses règles d'invocation de services pour identifier les ressources externes qui pourraient l'aider à répondre. Ceci peut impliquer d'accéder à une ou plusieurs ressources personnelles de l'utilisateur (ex : son agenda) et/ou à un ou plusieurs services publics auxquels il puisse faire confiance. Dans notre exemple, le campus où travail Fabien a un réseau sans-fil qui permet la localisation des appareils connectés. Cette fonctionnalité est invoquée par le *e-Wallet* pour obtenir la localisation de Fabien et ajouter ce nouveau fait à sa base de connaissances.
6. *Post-vérification des autorisations* : avec cette nouvelle connaissance, le *e-Wallet* peut maintenant finir de vérifier que la question est autorisable. Dans notre exemple, on permet aux collègues de Fabien de voir sa localisation uniquement lorsqu'il il est sur le campus. En supposant que Fabien soit sur le campus, la demande est maintenant considérée comme permise.
7. *Application des règles de révision* : la localisation par réseau sans-fil peut avoir renvoyé la salle exacte où se trouve Fabien cependant, comme mentionné en 3, Fabien n'est disposé à révéler que le bâtiment où il se trouve. Cette dernière condition est capturée par le *e-Wallet* sous la forme d'une règle de révision qui n'autorise que la connaissance du bâtiment à être utilisée pour la résolution de la question. L'application de cette règle implique typiquement d'accéder à des ontologies décrivant les concepts de salles et de bâtiments ainsi que des annotations au sujet du campus où Fabien travaille.
8. *Réponse* : la question ayant été traitée, une réponse est produite ("Fabien est dans le bâtiment Borel") et envoyée au service invoqué par Norman.

2.3 Architecture interne du *e-Wallet*

Nous avons conçu et implanté une architecture en trois couches (figure 2) :

- La *couche noyau* inclut un méta-modèle de OWL et maintient la connaissance statique (indépendante du contexte) et les préférences dynamiques (dépendantes du contexte). Ces connaissances sont obtenues en chargeant des descriptions disponibles au sujet de l'utilisateur ainsi que les ontologies appropriées et en appliquant un algorithme de chaînage-avant pour compléter la base et éviter de devoir inférer les mêmes faits à plusieurs reprises. Dans cette couche, la connaissance est représentée en utilisant le modèle de triplets RDF classique.
- La *couche service* complète la connaissance du noyau avec des règles d'invocation de services qui décrivent une correspondance entre des types de connaissances et

des services externes permettant de les obtenir. Celles-ci sont représentées par des règles de chaînage-arrière qui ne sont donc invoquées qu'en cas de nécessité. La connaissance dans cette couche est représentée en utilisant un type de triplets particulier appelé "triplet de service". Les triplets résidents dans la couche de service peuvent résulter de la migration d'un triplet de la couche noyau ou de l'activation d'une règle d'invocation. La migration se fait par chaînage-arrière.

- Dans la couche externe ou *couche de confidentialité*, les faits sont représentés par un autre type de triplets appelé "triplet autorisé". Lors de la réception d'une requête, le *e-Wallet* la décompose en un certain nombre de "besoins en triplets autorisés" *i.e.* seuls des triplets autorisés peuvent être utilisés pour répondre aux questions. Ces triplets sont produits par les règles d'autorisation et de révision à partir des triplets de service : les règles de confidentialité sont appliquées en chaînage-arrière et mettent en correspondance des besoins de triplets autorisés et des besoins de triplets de service ; une fois produits (par accès aux connaissances internes ou aux services externes), ces triplets sont révisés avant d'être recopiés dans la couche autorisée.

En résumé, à travers un mécanisme de chaînage-arrière, les "besoins en triplets autorisés" d'une question engendrent des "besoins en triplets de service et en triplets du noyau", résultant soit (a) en la génération des triplets autorisés qui peuvent être retournés en réponse à la question soit (b) en la levée d'une exception, si la question est interdite. A travers ces trois types de triplets, la sécurité repose directement sur les mécanismes de typage du langage d'implantation.

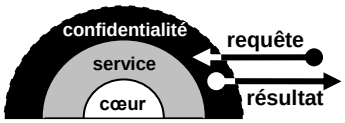

Fig. 2 – Les couches du *e-Wallet*

2.4 Implantation du *e-Wallet*

L'implantation du *e-Wallet* repose sur JESS (Friedman-Hill, 2003), un moteur de règles en chaînage-avant et son langage CLIPS. Ce moteur permet aussi le chaînage-arrière en réifiant les "besoins en faits" comme des faits eux-mêmes, qui peuvent à leur tour déclencher des règles en chaînage-avant chargées de répondre à ces besoins.

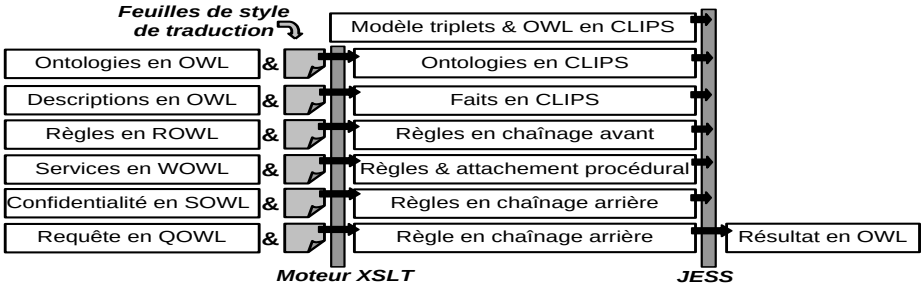

Fig. 3 – Architecture du *e-Wallet* pour la traduction et traitement.

Comme le montre la figure 3, la base de connaissances du *e-Wallet* est initialisée avec : (a) un modèle des triplets (RDF, 1999), (b) un modèle des triplets spécialisés utilisés pour créer les couches du *e-Wallet* et les règles de migration entre la couche noyau et la couche service, et (c) un méta-modèle de OWL. Dans un deuxième temps,

les connaissances additionnelles sont chargées en traduisant des descriptions OWL en des faits et règles CLIPS pour JESS. Les descriptions en OWL incluent des ontologies et leurs instances qui sont transformées en triplets classiques, des règles en chaînage-avant utilisées pour compléter la base (ex : définitions, préférences dynamiques) ainsi que des règles d'invocation de services et des règles de confidentialité en chaînage-arrière. Les langages ROWL, WOWL, SOWL et QOWL sont des extensions de OWL pour décrire respectivement, des règles de déductions de faits nouveaux, des règles d'invocation de services, des règles de confidentialité et des requêtes. Toutes les traductions sont des feuilles de styles (XSLT, 1999). Le système a été conçu pour rester efficace : la résolution des requêtes s'interrompt dès qu'une violation d'autorisation peut être prouvée ; la couche service s'assure que la connaissance n'est pas disponible dans le noyau avant d'appeler un service ; *etc*.

Pour représenter nos règles, nous avons proposé et implanté un langage basé sur RDF-S/OWL pour décrire des clauses de Horns avec variables. Comparé à RuleML (Boley *et al.*, 2003), nous n'avons pas voulu réifier les relations et les rôles de leurs arguments. Les relations étant toujours binaires en RDF, nous sommes restés sur le modèle des triplets et la syntaxe XML de RDF avec sa syntaxe de typage classique. Les seules extensions sont de nouvelles balises pour décrire les règles et un espace de nommage (*namespace*) spécial pour identifier des variables. Nous suivons de près les développements de RuleML, mais nous nous focalisons sur des règles dédiées à OWL. Nous pourrions facilement utiliser XSLT pour implanter une traduction entre notre représentation de règles en OWL et une représentation en RuleML.

Le modèle des triplets RDF est défini comme un modèle de faits non ordonnés et utilisé dans des règles de chaînage-avant. Le méta-modèle de OWL est alors décrit par une liste de faits non ordonnés instanciant ce modèle de faits non ordonnés et la sémantique des propriétés du modèle est traduite par des règles (Gandon & Sadeh, 2004). Nous nous sommes limités aux aspects de OWL-Lite utiles à nos scénarios d'application. Le code et les résultats obtenus sur la base de tests officielle sont disponibles en ligne[2]. La figure 4 donne en exemple un triplet typant la relation d'équivalence entre deux propriétés comme étant une relation symétrique (si $p_1 \Leftrightarrow p_2$ alors $p_2 \Leftrightarrow p_1$) et une règle décrivant sa sémantique (si $p_1 \Leftrightarrow p_2$ alors $p_1(s,o) \Rightarrow p_2(s,o)$).

```
(triple    (predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type")
           (subject   "http://www.w3.org/2002/07/owl#equivalentProperty")
           (object    "http://www.w3.org/2002/07/owl#SymmetricProperty") )

(defrule equivalent-property (declare (salience 100))
   (triple  (predicate "http://www.w3.org/2002/07/owl#equivalentProperty")
            (subject   ?p1)
            (object    ?p2))
   (triple (predicate p1?) (subject ?s) (object ?o))
   =>
   (assert (triple (predicate p2?) (subject ?s) (object ?o))) )
```

Fig. 4 – Déclarer en CLIPS la symétrie des relations et sa sémantique en OWL

Les triplets issus des ontologies et des descriptions sont aussi représentés par des faits non ordonnés. Des règles du domaine en ROWL sont utilisées pour la complétion de la base. Une règle définit par exemple les collègues comme des membres de la même équipe. Cette règle permet de représenter et d'interpréter des préférences

[2] http://mycampus.sadehlab.cs.cmu.edu/public_pages/OWLEngine.html

sensibles au contexte telles que "mes collègues peuvent voir ma localisation quand je suis au travail". Le moteur d'inférence est utilisé pour compléter la base en appliquant toutes ces règles et le résultat est gardé en mémoire, fournissant un point de repli et évitant ainsi de répéter cette complétion à chaque requête.

La figure 5 montre une règle d'invocation de services permettant d'accéder à Pocket Outlook sur l'agenda électronique de l'utilisateur. Ces règles peuvent avoir jusqu'à trois sous-parties : la balise *outpout* décrit la connaissance que ce service peut produire ; la balise *precondition* décrit la connaissance requise pour appeler le service ; la balise *call* décrit comment faire appel au service Web et avec quels paramètres.

```
<wowl:ServiceRule>
  <wowl:output>
    <mc:Person rdf:ID="&variable;#person">
      <mc:has_activity rdf:resource="&variable;#activity" /> </mc:Person>
  </wowl:output>
  <wowl:precondition>
    <mc:Person rdf:ID="&variable;#owner">
      <mc:PDA_endpoint>&variable;#endpoint</mc:PDA_endpoint> </mc:Person>
  </wowl:precondition>
  <wowl:call>
   <wowl:Service wowl:name="call-web-service">
     <wowl:qname>http://mycampus/PDAService#</wowl:qname>
     <wowl:endpoint>&variable;#endpoint</wowl:endpoint>
     <wowl:method>GetCurrentWeekAppointments</wowl:method>
     <wowl:user_id>&variable;#owner</wowl:user_id>
   </wowl:Service>
  </wowl:call>
</wowl:ServiceRule>
```

Fig. 5 – Règle d'invocation d'un service pour accéder à l'agenda de l'utilisateur.

La figure 6, montre une règle de confidentialité donnant l'accès en lecture à la localisation de l'utilisateur mais limitant la précision de la réponse à la présence ou l'absence sur le campus. Ces règles peuvent avoir jusqu'à trois sous-parties : la balise *target* décrit le type de connaissance auquel cette règle s'applique ; la balise *check* décrit les conditions requises pour accorder l'accès ; la balise *revision* décrit les révisions à apporter avant que la connaissance ne soit recopiée dans la couche autorisée. Un système équivalent a été implanté pour les accès en écriture.

```
<sowl:ReadAccessRule>
  <sowl:target>
    <mc:Person rdf:about="&variable;#owner">
      <mc:location rdf:resource="&variable;#location"/> </mc:Person>
  </sowl:target>
  <sowl:check>
   <rowl:And>
     <rowl:condition>
       <mc:E-Wallet>
         <mc:owner rdf:resource="&variable;#owner"/> </mc:E-Wallet>
     </rowl:condition>
     <rowl:condition>
       <mc:Place rdf:about="http://www.cmu.edu">
          <mc:include rdf:resource="&variable;#location" /> </mc:Place>
     </rowl:condition>
     <rowl:not-condition>
       <qowl:Query>
         <qowl:sender rdf:resource="&variable;#owner" /> </qowl:Query>
     </rowl:not-condition>
   </rowl:And>
  </sowl:check>
  <sowl:revision>
    <mc:Person rdf:about="&variable;#owner">
      <mc:location rdf:resource="http://www.cmu.edu"/> </mc:Person>
  </sowl:revision>
</sowl:ReadAccessRule>
```

Fig. 6 – Règle de confidentialité limitant la localisation à la présence sur le campus.

3 Synthèse et discussion

Nous avons présenté une architecture Web sémantique permettant l'accès à des connaissances personnelles et contextuelles tout en respectant la confidentialité. L'élément central présenté ici est le *e-Wallet* permettant la découverte et l'accès automatisés aux ressources personnelles d'un utilisateur et le respect des règles de confidentialité. Les ressources personnelles et publiques sont implantées sous la forme de services Web. Des règles d'invocation de services basées sur des ontologies permettent au *e-Wallet* d'identifier dynamiquement les ressources susceptibles de l'aider à répondre à une requête. De plus, le *e-Wallet* maintient et utilise des préférences de confidentialité et des règles de révision permettant d'ajuster l'exactitude ou l'inexactitude des réponses fournies et ceci selon le contexte de chaque requête. Nous avons décrit l'architecture en trois couches du *e-Wallet* et son implantation reposant sur JESS, OWL-Lite et XSLT. Nous avons évoqué les évaluations menées sur le campus de Carnegie Mellon et le système conclue actuellement la deuxième itération de son cycle développement par prototypes.

Les mécanismes de sécurité et de confidentialité courants ignorent la richesse des descriptions des ressources que propose la vision d'un Web Sémantique. La communauté de la représentation des connaissances a su montrer comment de nouvelles inférences de recherche, de classification, de composition, *etc.* pouvaient utiliser ces connaissances pour rendre les outils plus intelligents dans leur manipulation du Web. De la même façon, il est important de voir que d'autres tâches, telles que la sécurité et la confidentialité, peuvent elles aussi bénéficier d'inférences basées sur des ontologies et que cela est même nécessaire à la viabilité d'un paysage d'informations aussi riche dans ses représentations et dynamique dans ses évolutions.

On peut craindre des systèmes exploitant la connaissance du contexte qu'ils deviennent des systèmes de surveillance des individus. C'est pourquoi notre approche distribuée, basée sur les *e-Wallets*, impose au cœur même de l'architecture que le contrôle des accès aux ressources personnelles reste lui-même entre les mains de leurs propriétaires. Il ne s'agit pas ici de mettre en ligne toutes les ressources personnelles d'un utilisateur ; il s'agit simplement de permettre aux utilisateurs d'ouvrir un accès aux ressources nécessaires à des services qui leurs sont utiles et ceci en leur permettant de contrôler cet accès au même niveau de modélisation que les échanges effectués *i.e.* le niveau des connaissances et leur représentation orientée ontologie. C'est, à notre avis, une condition *sine qua non* d'une automatisation de l'accès et de la composition de services à l'échelle du World Wide Web. Dans cet article, nous avons montré comment une approche ontologique, peut permettre d'améliorer les systèmes de sécurité et de gestion de la confidentialité. De nouvelles inférences ont été intégrées aux mécanismes de sécurité, pour contrôler l'accès en utilisant des règles très expressives et ajuster la précision des réponses, voire même mentir lorsque cela s'avère moins dangereux que de refuser de répondre.

Nous avons clairement positionné le principe du *e-Wallet* par rapport aux travaux existants en montrant ses innovations: une intégration en temps réel de sources d'informations personnelles et publiques grâce à leurs descriptions sémantiques ; une spécification des accès aux ressources personnelles pour chaque service ; un contrôle

de l'exactitude et de la précision des réponses exploitant pleinement l'expressivité des ontologies mobilisées. Ce nouveau concept de *e-Wallet* pose cependant un certain nombre de questions : Comment contrôler les répercussions et la diffusion des révisions apportées aux réponses? Comment assurer la cohérence des réponses après recoupement? *etc.* Enfin, un autre défi demeure, héritage de toutes les approches utilisant des représentations riches : réconcilier l'expressivité des langages avec les exigences ergonomiques des utilisateurs finaux (Gandon & Sadeh, 2004b). Nous expérimentons actuellement différentes approches pour l'édition et l'apprentissage des préférences utilisateurs afin d'améliorer les interfaces actuelles.

Remerciements : DAML Initiative, Air Force Research Lab., Defense Advanced Research Project Agency, IBM, HP, Symbol, Boeing, Amazon, IST Program.

Références

BOLEY, GROSOF, TABET & WAGNER (2003) RuleML DTDs, The Rule Markup Initiative RuleML, http://www.dfki.uni-kl.de/ruleml/indtd0.8.html

DEY, SALBER, FUTAKAWA & ABOWD (2000) An Architecture to Support Context Aware Computing, Tech. Report GIT-GVU-99-23. Georgia Institute of Technology, November

FIPA (2002) Specifications http://www.fipa.org/repository/fipa2000.html

FRIEDMAN-HILL (2003): Jess in Action: Java Rule-based Systems, Manning Publications Company, June, ISBN 1930110898, http://herzberg.ca.sandia.gov/jess/

SAML (2003) OASIS: Security Assertion Markup Language, Technology Reports, April 14 http://xml.coverpages.org/saml.html

XACML (2003) OASIS: Extensible Access Control Markup Language, Technology Reports, March 28, http://xml.coverpages.org/xacml.html

GANDON & SADEH (2004) Semantic Web Technologies to Reconcile Privacy and Context Awareness, Web Semantics Journal. Vol. 1, No. 3, voir aussi CMU-CS-03-211 Tech. Report, http://reports-archive.adm.cs.cmu.edu/anon/2003/abstracts/03-211.html School of Computer Science, Carnegie Mellon University

GANDON & SADEH (2004b), Connaissance du Contexte et Privauté dans les Accès Mobiles: une Approche Web Sémantique et Multi-agents, Journées Francophones: Mobilité et Ubiquité

SADEH, CHAN, VAN, KWON & TAKIZAWA (2003) Creating an Open Agent Environment for Context-aware M-Commerce, Agentcities: Challenges in Open Agent Environments, Ed. Burg, Dale, Finin, Nakashima, Padgham, Sierra, and Willmott, LNAI, Springer Verlag, pp.152-158

SADEH (2002) m-Commerce: Technologies, Services and Business Models, Wiley

SCHILIT (1995) A System Architecture for Context-Aware Mobile Computing, Ph.D. Thesis, Columbia University.

SCHUNTER & POWERS (2003) The Enterprise Privacy Authorization Language (EPAL 1.1), IBM Research Laboratory, http://www.zurich.ibm.com/security/enterprise-privacy/epal/

WANT, HOPPER FALCAO & GIBBONS (1992): The Active Badge Location System, ACM Transactions on Information Systems 10(1) 91-102.

OWL (2003) Web Ontology Language, Working Draft http://www.w3.org/TR/owl-ref/

P3P (2002) Platform for Privacy Preferences, Recommendation, http://www.w3.org/TR/P3P/

XSLT (1999) XSL Transformations Version 1.0, Recommendation, http://www.w3.org/TR/xslt

RDF (1999) Resource Description Framework, http://www.w3.org/TR/REC-rdf-syntax/