Un protocole pour la négociation des politiques de privacy

Amina Mekki Mokhtar, Kheira Bekara, Maryline Laurent

Rapport de recherche 09-010 LOR

Amina MEKKHI MOKHTAR, Kheira BEKKARA, Maryline LAURENT
(Télécom SudParis, département LOR - Samovar, UMR 5157)

Un protocole pour la négociation des politiques de privacy

Table des matières

Introduction

L'augmentation du nombre de fraudes sur internet dues aux vols des données personnelles ont alerté les internautes sur la nécessité de respect de la vie privée. Ils sont de plus en plus nombreux à être préoccupés par l'utilisation de leurs informations personnelles ce qui a poussé les site internet à publier des politiques de respect de la vie privée.
Ces politiques de confidentialité sont publiées grâce à la Platform for privacy Preferences (P3P) et au langage XML conçu pour spécifier comment les sites internet utiliseront les données recueillies lorsque les utilisateurs accèdent à ces site. Généralement, ils publient leur politique P3P dans un endroit précis dans leur serveur. Lorsqu'un client visite un site internet, un logiciel inclus dans son navigateur examine la politique et la compare aux préférences que l'utilisateur a déjà configurées. Si la politique et les préférences concordent, le logiciel approuve la transaction et l'utilisateur peut continuer à naviguer normalement. Dans le cas où le logiciel détermine qu'il y a incompatibilité entre la politique du site et les préférences de l'utilisateur la transaction est alors interrompue.
Cette approche *take it or leave it* est beaucoup trop rigide. Aussi, chaque utilisateur a sa propre vision de la privacy et il se peut qu'il soit prêt à être plus flexible en ce qui concerne ses préférences. Les utilisateurs ont, en général, un ensemble de préférences idéales qu'il voudrait absolument préserver et un ensemble de préférences qu'il pourrait concéder contre un minimum de protection. Du côté serveur, un site internet possède un ensemble d'informations qu'il voudrait collecter et utiliser d'une certaine manière. Cependant il serait disposé à en collecter moins et à les utiliser de façon plus prudente, si l'utilisateur en exprime l'envie.
Introduire un processus de négociation est une façon de rendre les échanges entre les sites internet et les utilisateurs plus flexible. De nombreux protocoles de négociation de politiques de privacy ont vu le jour. Néanmoins, il

existe des limitations dans ces travaux comme des échanges infinis lors de la négociation ou une négociation portée seulement sur les données recueillies et non pas sur les politiques associées. Le but de notre travail est de proposer un protocole qui traiterait ces limitations.

Nous proposons un protocole de négociation de politique de vie privée qui se termine au bout d'un nombre limite de rounds. Le serveur commence par faire une proposition au client, ce dernier soit l'accepte soit fait une proposition à son tour. Si au bout de quelques échanges aucun compromis n'a été trouvé, le serveur a recours à l'ontologie de son domaine afin de remplacer les données qui posent problème et essayant ainsi d'éviter l'échec de la transaction.

Ce rapport est organisé comme suit : la chapitre 1 présente quelques généralités et définition sur le respect de la vie privée et les outils utilisés pour. Le chapitre 2 expose l'état de l'art sur la négociation des politiques de respect de vie privée. Le chapitre 3 définit les messages échangés entre les deux parties, sous quelles conditions s'effectue l'envoie et la stratégie de négociation. Ce document est clos par une conclusion et des perspectives.

Chapitre 1

Généralités sur la vie privée (Privacy)

1.1 L'identité numérique

Avec l'évolution d'internet et l'augmentation des services proposés en ligne, l'utilisateur se retrouve à gérer différents profils (ensemble d'informations qu'il a saisies telles que son nom, son mail etc ...) sur les blog, réseaux sociaux, forums, sites web etc... cette représentation numérique des informations connues sur une personne est appelée l'identité numérique (ou identité virtuelle) [1].

Ces identités virtuelles propagent des informations personnelles appelées "Personnaly Identifiable Information ou PII". En 2006, l'Office du E-Government et de la Technologie de l'Information des USA a émis un Mémorandum pour les chefs officiers des services d'information, le mémo inventait le concept PII et le définissait comme étant "n'importe quelle information sur un individu détenue par une agence, incluant des informations sur l'éducation, transactions financières, historique médical, criminel ou professionnel et les informations qui peuvent être utilisées pour distinguer ou tracer les individus comme leurs noms, numéro de sécurité social, date et lieu de nais-

sance, nom de jeune fille de la mère, données biométriques etc... elle inclut toute autre information qui est liée ou corrélée à un individu [2].

1.2 Gestion de l'identité numérique

Face à l'explosion de l'utilisation des services sur internet, les utilisateurs se retrouvent à manipuler de multiples identités numériques ; sans pour autant pouvoir les gérer. En effet, ils ont du mal à contenir la propagation de leurs attributs tels que pseudo, noms, mots de passe etc... De plus l'augmentation des fraudes sur internet fait que les utilisateurs ont de moins en moins confiance dans les services en ligne tels que le e-commerce. Pour palier à cela, la gestion d'identité ou IdM (Identity Management) est apparue. L'IdM est la gestion du cycle de vie de l'identité au sein d'un domaine. Elle recouvre plusieurs dimensions : techniques (système de gestion des identités), légales (protection des données), judiciaires (usurpation d'identité), sociales (protection de la vie privée) etc... On peut la définir comme étant le processus utilisé pour créer et supprimer une identité numérique d'un utilisateur et gérer ses accès aux ressources électroniques telles que les réseaux, les fichiers ou les services [3].

1.3 Fédération de gestion des identités

La FIM (ou Federation Identity Management) quant à elle permet aux utilisateurs d'un domaine d'accéder de manière fiable et transparente aux données et aux systèmes d'un autre domaine. Elle permet d'augmenter la sécurité et de réduire les risques en permettant à une organisation d'identifier et d'authentifier une seule fois l'utilisateur pour ensuite utiliser cette identité à travers de multiples domaines. Elle peut améliorer le respect de la vie privée en permettant à l'utilisateur de contrôler quelles informations sont partagées, ou en limitant l'impact du partage. Le marché de l'identity management a

explosé ses dernières années et des standards sont apparus comme Liberty Alliance [1], OpenID [2], InfoCard [3] etc...

1.4 Fédération de cercles de confiance ou FC^2

1.4.1 Les cercles de confiance

Ces dernières années, le e-commerce a vu un développement fulgurant dû au fait que les gens ont de plus en plus confiance dans la sécurisation des paiements en ligne. Certains web services ont décidé de créer des liens de confidentialité ; une sorte de partenariat aussi appelé *cercle de confiance*. Un utilisateur qui accède à un web service dans un cercle de confiance peut alors accéder aux autres web services du même cercle sans authentification supplémentaire. Les cercles de confiance sont naturellement composés de web services de même nature.

1.4.2 FC^2

La Fédération de Cercles de Confiance ou FC^2 [4] est un projet recherche et développement français qui a débuté en 2007. Il regroupe 20 membres de différents domaines : petites et moyennes entreprise (NTX Research, CEV Group, ...) Grandes entreprises (EADS, Orange LABS, ...) Universités et laboratoires : CNAM, TELECOM & Management SudParis, ...). Il a pour objectif de mettre en place un nouveau modèle permettant d'assurer l'inter-opérabilité entre des cercles de confiance hétérogènes par exemple bancaire, gouvernemental, télécommunication (cf. figure n°1.1) . Le projet FC^2 permet une coopération entre différents fournisseurs de service basée sur le partage

1. http ://www.entrouvert.com/fr/identite-numerique/liberty-alliance
2. http ://openid.org/home ?aspxerrorpath=/
3. http ://informationcard.net/
4. http ://www.fc2-consortium.org/

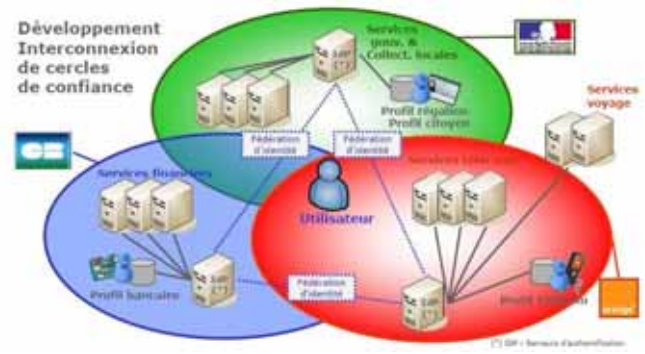

FIGURE 1.1 – Fédération des cercles de confiances FC^2.

d'attributs avec une forte contrainte de respect de la vie privée de l'utilisateur [4]. Le modèle adopté devra être interopérable avec la majorité des environnements de fédérations d'identité courantes (Liberty Alliance, InfoCard, higgins [5], etc...) [5].

La fédération est composée de deux entités principales : l'Identity Provider (IdPs) qui gèrent les identités individuelles et les Services Providers (SPs) qui offrent les services aux utilisateurs[1] . FC^2 propose une approche centrée sur l'utilisateur. Elle lui permet de configurer lui même ses préférences quant à l'utilisation de ses attributs. En effet un "FC^2 identity selector" est installé chez l'utilisateur et lui permet d'accéder aux différents cercles de confiance. Il gère les information-cards [6] générées par chaque IdP de chaque cercle de confiance. L'utilisateur contrôle son identité, supervise la propagation de ses informations. Il choisit de donner ou pas l'accès d'un SP à l'infocard qu'il lui demande.

5. http ://www.eclipse.org/higgins/

6. Ce concept fut introduit par microsoft en 2005. Elles permettent de donner la gestion de l'identité à l'utilisateur. Ses infocards sont stockées dans son ordinateur. Chaque carte représente une de ses identités numériques et contient un ensemble d'attributs qui le représentent (nom, date de naissance...) [4]

1.5 Définition de la Privacy

Depuis le début nous parlons de *privacy* ; mais que veut dire cette notion au juste ? La privacy signifie littéralement le respect de la vie privée. Selon [6], la privacy est *"la possibilité des individus à contrôler la collecte, l'utilisation et la transmission de leurs informations par une tierce personne"*. Les PII sont particulièrement sensibles et combinées à d'autres données peuvent être utilisées pour construire les profils des individus, leurs habitudes ainsi que leurs préférences.

Aussi, le professeur Steven.M.BELLOVIN de l'Université de Columbia souligne que l'envahissement de la vie privée ne se limite pas qu'au monde numérique mais qu'il touche de plus en plus le monde physique. Il donne comme exemple dans [7] l'utilisation des caméras dans les péages des autoroutes afin de récolter les plaques minéralogiques des voitures, leurs horaires de passage ainsi que les radiographies des véhicules. Ces informations stockées doivent servir à des raisons de sécurité nationale mais peuvent faire l'objet d'utilisations tierces.

La privacy est donc un enjeu majeur. Dans le e-commerce par exemple, en 2003 72% des internautes abandonnaient leurs achats en ligne lorsqu'ils devaient saisir des informations personnelles et en 2001 la méfiance des internautes a coûté à l'économie américaine plus de 15 milliards de dollars [6].

De ce fait, des efforts considérables ont été accomplis que ce soit dans le monde industriel ou en recherche, et parmi eux l'élaboration du standard *P3P*.

1.6 Langages associés à la privacy

1.6.1 P3P

Platform for Privacy Preferences ou P3P a été développé par The World Wide Web Consortium [7] (W3C). P3P est une recommandation depuis 2002 et a pour but d'informer les internautes sur les pratiques de collecte des données des sites web. P3P a été conçu avec deux objectifs [8] :

- Permettre la transparence de la politique de privacy. Ce qui nécessite que la politique soit décrite dans un format clair et lisible par un humain.
- Permettre aux utilisateurs de savoir quelles données seront collectées par les sites visités, de quelle façon ses données seront utilisées et quels usages de ces données ces internautes accepteront ou refuseront.

La politique de confidentialité d'un site est mentionnée dans le site lui même et été écrite en language naturel que personne ne prenait la peine de lire. De plus, le problème de la compréhension de la langue dans laquelle est écrite la politique se pose. P3P alors permet l'expression de la politique de confidentialité dans un format structuré et normalisé que les agents-utilisateurs [8] obtiendront facilement et interprèteront aisément.

Néanmoins, rien ne garantit que le site internète se comporte coformément à sa politique de confidentialité. Il existe des mécanismes de certification des politiques P3P, c'est à dire une signature électronique de ces politiques par un tiers de confience. Ces autorités garantissent le respect de la politique de confidentialité décrite dans le site, et indémnisent les usagers en cas de manquement. En réalité, peu de sites envoient une politique P3P et les mé-

7. http ://www.w3.org/

8. Un agent utilisateur ou user agent est une application cliente relatif à protocole réseau particulier. Elle peut peut aller du simple navigateur web jusqu'au moteur de rechreche.

canismes de vérification des certificats des politiques ne sont pas encore mis en oeuvre [9].

1.6.2 APPEL

APPEL ou P3P Preference Exchange Language [10] complète le standard P3P. Il décrit un ensemble de préférences de l'utilisateur vis à vis d'une politique de confidentialité. En utilisant ce language, l'utilisateur peut exprimer un ensemble de règles de préférences qui sont utilisées par l'agent-utilisateur de son navigateur pour comparer ses préférences à la politique P3P du site web et décider s'il accepte cette dernière ou pas.

P3P et APPEL expriment tous deux la politique en format XML. *eXtensible Markup Language* est un langage informatique de balisage. Il sert à stocker et transférer des données structurées de manière arborescente. Il est extensible car il permet à l'utilisateur de définir lui même des balises.

1.6.3 Le codage XML des politiques

W3C définit un standard pour exprimer la syntaxe et la sémantique des politiques P3P.
Les politiques doivent être placées dans un élément <POLICIES >. Voir le lexique pour plus de détails.

9. $http://fr.wikipedia.org/wiki/Platform_for_privacy_preferences$
10. http ://www.w3.org/TR/P3P-preferences/

Chapitre 2

Etat de l'art sur la négociation des politiques de privacy

2.1 Introduction

L'augmentation du nombre d'usurpations d'identités et la perte de données personnelles a alerté les gens sur la nécessité d'une politique de confidentialité. Pour cela, de plus en plus de sites web publient leur politiques de sécurité en utilisant P3P ; ils la stocke dans un endroit connu afin que les user agents puissent la récupérer et la comparer avec les préférences de l'utilisateur qu'il a préalablement configurées. Si la politique du site web concorde avec les préférences de l'utilisateur alors le user agent va approuver la transaction sinon un message d'erreur s'affichera chez le client. Ce dernier a le choix alors soit d'accepter le contrat du site web et continuer sa transaction, soit de refuser et la transaction sera alors interrompue. Si ce consommateur est vraiment intéressé par le service, il acceptera sûrement de divulguer les informations demandées. Cette approche de "take it or leave it" est trop rigide.
L'utilisateur est en géneral prêt à faire des concessions. Il a un ensemble de préférences *idéales* et d'autres *suffisantes* qu'il est prêt à divulguer contre

un minimum de confidentialité. Du côté serveur, le site web est aussi prêt à collecter moins d'informations et à les utiliser avec plus de précautions seulement si l'utilisateur demande de telles protections [9]. Un processus de Négotiation peut alors rentrer en jeu mais l'utilisateur n'a pas forcément envie de négocier à chaque fois qu'il effectue une transaction ; c'est pour celà qu'une automatisation est nécessaire. Néanmoins la version courante de P3P n'inclut délibérément aucun processus de négociation[privacy policy nego]. Le premier draft de la spécification de P3P incluait un mécanisme de négociation qui a été supprimé pour laisser place à une simple comparaison afin d'alléger l'implémentation et permettre une adoption plus rapide du protocole [10].
Les chercheurs ont eu alors un intérêt particulier pour la négociaion ; elle fut étudiée dans différentes disciplines.

2.2 Comparaison simple basée sur les agents utilisateurs

2.2.1 Evaluation globale de la privacy

Un premier travail fut effectué par AT&T. Ils ont implémenté une sorte d'outil d'aide dans le navigateur Internet Explorer appelé *Privacy Bird*[1]. Ce logiciel incorpore une icone en forme d'oiseau à droite de la barre d'outils du navigateur (cf. figure n°2.1). Les utilisateurs peuvent configurer Privacy Bird avec leurs propres préférences en utilisant une interface graphique ou en important des fichiers APPEL. A chaque fois que l'utilisateur visite un site web, Privacy Bird recherche les politiques P3P et les compare aux préférences de l'utilisateur. Si la politique de sécurité du site web concorde avec les préférences utilisateur alors une icone sous forme d'un oiseau vert heureux

1. http ://www.privacybird.org/

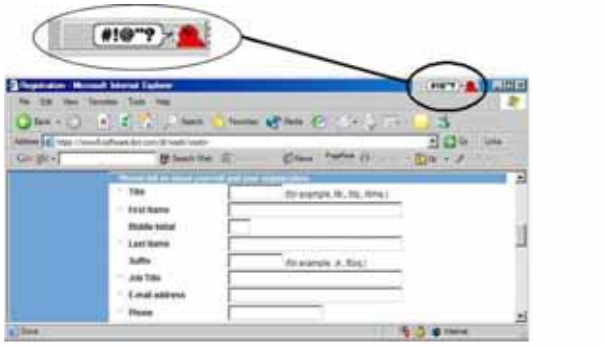

FIGURE 2.1 – Privacy Bird indiquant un conflit.

apparait. Dans le cas contraire c'est une icone sous forme d'un oiseau rouge et en colère qui apparait. Dans le cas où le site web n'a pas de politique P3P ce sera un oiseau jaune incertain.
Néanmoins, il reste difficile à l'utilisateur de déterminer la cause du conflit en cas d'échec de la comparaison. Aucune information n'est donnée quant aux champs de données qui ont causé problème. De ce fait l'utilisateur ne sachant pas où réside le souci risque fortement de fermer la page web et abandonner la transaction.

2.2.2 Evaluation détaillée

Afin de mieux localiser le problème, [11] propose *The Integrated Privacy View*. Elle permet de visualiser la conformité des préférences utilisateurs et des politiques de privacy devant chaque champ où l'utilisateur doit saisir une information (cf. figure n°2.2) . Ce module se compose de trois parties : une extension P3P qui permet de lier les élements de la politique à ceux qui leur correspondent dans la page HTML ; un mécanisme qui permet d'intercepter les pages web contenant cette extention et un nouveau user agent qui permet de tester la conformité entre la politique de privacy du site web et les préfe-

FIGURE 2.2 – Visualisation de la conformité au niveau de chaque champs.

rences utilisateur et cela individuellement pour chaque champs de saisie de la page web. Il insère par la suite au niveau de chaque champs un indicateur visuel en forme de visage vert heureux ou rouge en colère. De plus, une fenêtre pop up s'affiche lorsqu'on passe la souris sur le visage indiquant une courte description sur la conformité ou bien des informations supplémentaires sur la cause du conflit.

2.2.3 Evaluation des résultats d'une recherche dans un navigateur

Dans [12], les auteurs proposent un *Privacy Bird Search Engine* qui permet aux utilisateurs d'effectuer des recherches sur internet tout en donnant des informations sur les politiques de privacy des résultats de la recherche. Privacy Bird Search Engine contient quatre composants principaux : *un module d'acquisition des politiques*, *un module d'intégration à Google*, *un évaluateur APPEL* et *un cache*. L'utilisateur saisit sa recherche dans une interface qui ressemble à l'interface Google grâce au module d'intégration à Google. Ce dernier va soumettre la recherche au moteur de recherche Google et va récupérer les résultats de la recherche. Pour chaque lien il va rechercher sa

politique de sécurité dans sa base de données des politiques si elle existe. Dans le cas contraire il va la récupérer directement du site web grâce au module d'acquisition des politiques puis la stocker dans la base de données. Les politiques sont évaluées grâce à l'évaluateur APPEL qui va comparer toutes les politiques aux préférences utilisateurs et retourner les résultats au module d'intégration à Google qui affichera à l'utilisateur le résultat de sa recherche et placera à côté de chaque lien un oiseau rouge, jaune ou vert (cf. figure n°2.3) . En ce qui concerne le cache, il permet de maintenir à jour la base de données des politiques dès que celles ci sont dépassées et en même temps scane internet à la recherche de nouvelles politiques P3P.
Google peut renvoyer un nombre incroyablement grand de liens ; les auteurs ne donnent aucun détail quant à la charge de travail que nécessite le traitement de chaque lien retourné ni sur la capacité de stockage du cache.

2.3 Négociation

2.3.1 Les travaux d'HP

Les auteurs de [13] ont proposé une méthodologie qui permet à différents Services Provider (SP) d'une même FIM de partager des PII sans passer par l'utilisateur et cela tout en respectant ses préférences. Leur approche permet de comparer directement les politiques de privacy des SP avec les préférences utilisateurs. Pour cela, ils ont introduit la notion de *policy subsumption*[2]. Elle donne aux SP une grande autonomie du fait qu'ils peuvent grâce à elle spécifier les préférences utilisateur et les comparer avec les politiques d'autres SP si besoin est.
Les auteurs ont défini de manière formelle comment exprimer la politique de sécurité en utilisant *les préférences utilisateur*, *les obligations* :les conditions qui doivent être vérifiées pour que les PII puissent être utilisées et *les règles*

2. du verbe *subsumer* : établir une relation hiérarchique entre des concepts proches. Equivalente à l'impliquation en logique : http ://fr.wiktionary.org/wiki/subsumer

FIGURE 2.3 – Résultats de la recherche et visualisation de la conformité de chaque site web.

de partage des données : qui spécifient comment les PII doivent être manipulées. La notion de subsumption fut exprimée en utilisant les ontologies (voir définitton page 27).
Ils ont proposé deux algorithmes : le premier permet de comparer la compatibilité de deux politiques de sécurité de deux SP. Il ne prend pas en considération les préférences du client comme par exemple les obligations (qui sont une instance de ses préférences). C'est une sorte de haut niveau de compatiblité. Comme résultat, l'algorithme fournit l'ensemble des conditions que doivent vérifier les préférences du client pour que les PII puissent être partagées entre les deux SP. Le second algorithme effectue une comparaison plus fine en utilisant des préférences utilisateurs instanciées. En sortie, nous avons soit un échec alors le SP ne peut pas partager ces PII avec le second SP ; soit un succès.

2.3.2 Basée sur des échanges simples

– Langendörfer et Bennicke proposent dans [6] un processus de négociation où ils ne différencient pas le client du serveur. Les deux parties de la négociation ont un ensemble de demandes et peuvent chacun leur tour faire une proposition. Soient A et B les négociants. A effectue la première proposition à B qui contient un sous ensemble de ses demandes. B reçoit cette proposition et la compare avec ses demandes à lui. Si aucun conflit n'apparait, B accepte la proposition. Si A a d'autres demandes il les propose séquentiellement à B. Les rôles peuvent changer au cours de la négociation car B peut faire une proposition à A à son tour. Quand l'une des deux parties a épuisé ses demandes elle envoie le message *finish*. La négociation est terminée une fois que les deux parties ont envoyé le message *finish*.
  Les auteurs ont aussi introduit des messages bien spécifiques ; en effet lorsque B reçoit une proposition il peut soit accepter, soit accepter avec condition, soit rejeter avec une contre proposition, soit rejeter avec une

description de la cause du problème, soit rejeter définitivement la proposition de A.
Au cours de la négociation, lorsque l'une des deux parties reçoit une proposition, peut la mettre de côté et envoyer sa propre proposition. Ces reports peuvent à la longue mener à une impasse. De plus ce processus de négociation pause un problème de terminaison. De plus des informations sensibles peuvent être véhiculées dans la description du rejet.

– Dans [14] les auteurs proposent tout d'abord de représenter un contrat comme étant deux catégories distinctes : un *contract proposal* qui est la proposition de contrat émise par le SP et l'ensemble des *préférences* du client. Ils ont défini un contract proposal comme étant un ensemble de *paragraphs* et les préférences comme étant un ensemble de *directives*. Chaque paragraphe et directive se compose d'un ensemble de *statements*. Chaque statement est composé à son tour d'un ensemble de *subjects* groupés en *categories*. Un contract proposal n'est accepté que si tous les paragraphes le sont. Pour qu'un paragraphe soit accepté, il faut qu'il soit en accord avec toutes les directives.
Ils ont défini une relation appelée *legal relation* et qui permet de statuer sur l'acceptation d'un contract proposal. Cette relation entre un paragraph et une directive peut prendre 5 valeurs : NOT AFFECTED, POSSIBLY PERMITTED,DEFINITLY PERMITTED, POSSIBLY PROHIBITED, DEFINITELY PROHIBITED. Ils ont aussi défini syntaxiquement un paragraph et une directive comme suit : $paragraph = (purposes_{\S}, recipients_{\S}, retention_{\S}, datagroup_{\S}, rewards_{\S})$; $directive = (purposes_D, recipients_D, retention_D, datagroup_D, rewards_D)$. La *legal relation* $\boldsymbol{R}$ pour un paragraph et les préferences s'exprime comme étant : *paragraph R preferences = (paragraph R directive*$_1$*)*$^\circ$*...*$^\circ$*(paragraph R directive*$_{n-1}$*)*$^\circ$*(paragraph R directive*$_n$*)* avec $^\circ$ étant une relation de *concaténation*.

De la même manière la legal relation entre un paragraph et une directive fut exprimée comme suit : $paragraphRdirective = ((([purpose_{\S}Rpurpose_D]\bullet [recipient_{\S}Rrecipient_D])\bullet[retention_{\S}Rretention_D])\bullet[datagroup_{\S}Rdatagroup_D])$ avec $\bullet$ étant une seconde relation de concaténation.
Les auteurs ont décrit des tableaux et des règles définissant la relation **R** ainsi que les relations de concaténation $^\circ$ et $\bullet$ néanmoins nous déplorons le manque d'explication des définitions. De plus, il existe deux spécifications différentes : une pour le serveur et une pour le client ; cette dernière n'a pas été présentée.

2.3.3 Basée sur la notion d'utilité et de bénéfice

Dans [10], en plus de la rigidité du concept "take it or leave it" des échanges courants, les auteurs mettent le doigt sur une autre lacune qui est *"one-size-fits-all"*, c'est à dire que le service web présente une politique de sécurité unique à tous ces clients, alors qu'ils ont des préférences et des habitude différentes. Pour palier à cela, ils ont défini la notion de *privacy dimension*. Il considère la politique de sécurité du site comme étant un ensemble multi-dimensionnel de concepts. Pour chaque dimension, différents niveaux de divulgation de données existent correspondant à la volonté de l'utilisateurs de révéler ses données. Basés sur la sémantique P3P, tous les élements non optionnels peuvent être négociés ; les auteurs ont choisi de négocier le nombre d'attributs à partager.
Ils ont défini une fonction d'utilité U pour l'utilisateur qui dépend de plusieurs paramètres tels que : le niveau de sensibilité des données, la dimension de privacy, le seuil minimum des données nécessaires etc... Notons l'introduction de la notion de *bénéfices* qui peut être soit une réduction par exemple soit un bénéfice non matérialisable. Le serveur entame la négociation en donnant une offre de base consistant en un petit ensemble de données et peu de bénéfices. Dans le cas où l'utilisateur accepte, une autre offre lui sera pro-

posée, avec plus de données et par conséquent plus de bénéfices. A chaque étape l'utilisateur peut soit accepter et passer à l'étape suivante, soit annuler et la transaction échouera, soit confirmer et la présente offre sera définitive. Le but étant de maximiser la fonction d'utilité de l'utilisateur. L'implémentation de ce processus fut effectuée en utilisant les mécanismes d'extension de P3P. L'extension de négociation est considérée comme obligatoire et est référencée dans le fichier adéquat. De ce fait seuls les user-agents capables d'interpréter cette extension peuvent effectuer la négociation.

2.3.4 Basée sur la théorie des jeux

David Ahn et haifei Li proposent dans [15] un processus de négociation en deux phases : la phase de pré-négociation qui permet d'éliminer tous les éventuels contrats qui ne seraient pas *pareto-optimal*[3] et la phase de marchandage ou de négociation à proprement dite.
Lors de la première phase l'algorithme renvoie l'ensemble des préférences (coté client et coté serveur) pareto-optimales, ordonnées des plus préférables aux moins préférables. Dans la seconde phase, les deux négociants associent à chaque préférence un compteur qui équivaut au nombre de fois qu'il proposera cette préférence à son adversaire. Soient N_1 et N_2 les deux négociants. N_1 commence par envoyer la préférence la plus élevée sur sa liste issue de la première étape et décrémente le compteur associé. N_2 reçoit la proposition et vérifie si elle est égale à sa préférence la plus élevée. Si oui il accepte la proposition, sinon il envoie à N_1 une nouvelle proposition (la première dans son ordre de préférences) et décrémente le compteur qui lui est associé. Vu que la liste des préférences est limitée et que les compteurs associés aussi, ce processus finit après un certain nombre d'échanges.
Cette approche parait intéressante du fait que le processus termine et par l'in-

3. est un terme économique qui signifie : *un état dans lequel on ne peut améliorer le bien être d'un individu sans détériorer celui d'un autre : http ://fr.wikipedia.org/wiki/Optimum_de_Pareto.* Dans notre cas cela signifie que la négociation est effectuée sans privilégier un négociant par rapport au second

troduction du concept de "compteur". En revanche, aucune distinction n'est faite entre le client et le SP et la négociation est appliquée pour négocier les données échangées et non pas les politiques.

2.3.5 Basée sur la notion de récompenses

Dans [16], la négociation se fait non seulement sur les données échangées mais aussi sur la récompense offerte en échange de la divulgation de ces données. Une fonction d'utilité a été introduite (côté client et serveur) pour chaque statement de la politique P3P qui représente l'utilité du serveur à obtenir cette information (parallèlement l'utilité du client à dévoiler l'information). De la même manière une fonction d'utilité fut introduite pour chaque récompense. Ces deux fonctions serviront à calculer l'utilité d'une offre. Une offre consiste en un statement et à la récompense qui lui est associée. Le SP et le client définissent chacun de leur coté un seuil ; ils n'accèpteront une offre que si son utilité est supérieure à ce seuil. Ils définissent aussi un tableau qui donne pour chaque combinaison possible de données divulguées l'utilité correspondante.
Les auteurs ont établi un ensemble de contraintes et de règles à suivre lors de la négociation afin que le protocole converge. Le serveur entame la négociation en envoyant au client la combinaison ayant le maximum d'utilité. Si elle maximise aussi l'utilité coté client, ce dernier l'accèpte sinon il lui envoie une offre qui maximise son utilité à lui. Et ainsi de suite jusqu'à ce qu'ils trouvent un terrain d'entente.
Le modèle proposé introduit de manière très pertinente la notion de récompense au sein du processus de négociation, en revanche l'expressivité de la politique de sécurité et des préférences utilisateurs reste pauvre. Là aussi, la négociation se base sur les données échangés et non pas les politiques associées. De plus, définir les tableaux d'utilité reste à l'appréciation du négociant ; aucune méthode n'est proposée.

2.3.6 Négociation en 3 rounds

D.Walker propose dans [9] une stratégie de négociation très interessante qui s'effectue en trois rounds seulement. Dans chaque round une partie fait une proposition à la seconde et celle ci décide de l'accepter ou de la rejeter. Le serveur effectue la première proposition en demandant toutes les données ainsi que leur politique. Le client choisit d'accepter ou de refuser en envoyant une contre proposition. Cette contre proposition contiendra ses préférences à lui. Le serveur à ce moment choisira à son tour d'accepter cette proposition ou de la refuser en envoyant sa meilleure offre (ou *best offer*). C'est alors que le client peut soit accepter et la négociation terminera avec succes soit refuser et la négociation échouera.
L'auteur a défini un formalisme inspiré de P3P pour exprimer la politique de sécurité. Il a aussi défini un modèle de préférences client qui contient : une fonction d'utilité qui servira à évaluer les termes de la politique négociés et trois DAGs (Directed Acyclic Graphs) un associé à la politique "recipient", un à la politique "retention" et un à la politique "purpose". Ils aideront à partager les données en trois ensembles : *Ideal* pour les données et les politiques dont la divulgation ne le gène pas ; *Acc* pour ceux qu'il ne voudrait pas partager mais qu'il est prêt à faire des concessions et *Unacc* pour ceux qui ne cédera sous aucun pretexte.
Et de la même manière un modèle de préférences serveur qui contient aussi une fonction d'utilité. De plus les politiques coté serveur sont distiguées en deux groupes : les politiques *required* sont celles nécessaires à la transaction sinon celle ci échouera et *preferred* sont celles qu'il aimera voir présentes dans la politique finale mais dont l'abscence n'entrave pas le succes de la transaction.
L'auteur a aussi établie un ensemble de règles et de contraintes que la négociation doit suivre afin que le résultat de la négociation soit pareto-optimal. Par ailleurs, il a sécurisé les échanges entre les deux parties en signant les messages et en incluant des timestamp et des identificateurs. De plus il a

représenté de manière claire la politique de sécurité et les préférences utilisateur. En fin la négociation finit obligatoirement au bout de trois rounds.

2.4 Négociation basée sur les ontologies

2.4.1 Introduction

Malgré tous ces efforts fournis, la privacy dans le web rencontre encore de nombreux problèmes dûs à la nature ouverte du web et la fuite d'informations sensibles lors des transactions. P3P et APPEL définissent des plateformes qui permettent de comparer les préférences utilisateurs à la politique proposée par le service web. Néanmoins ils ne permettent pas à l'utilisateur de contrôler leurs informations une fois celles ci saisies [17]. De plus ils n'ont pas l'expressivité nécessaire pour s'assurer que les données sont manipulées en conformité avec la politique de sécurité du SP [18]. Il est donc évident qu'une sémantique formelle de plus haut niveau s'impose. Les chercheurs en sécurité ont fait plusieurs propositions qui ont mené à l'apparition de la *Privacy Ontology*.

2.4.2 Qu'est ce qu'une ontologie ?

Il existe plusieurs définitions de l'ontologie dans la littérature ; l'équipe de Stanford [] la présente comment étant *une description formelle et explicite des concepts d'un domaine, les propriétés de chaque concept qui décrivent les fonctions et les attributs du concept et les restrictions sur ces attributs.* Une ontologie combinée aux instances des classes constituent ce qu'on appelle **une base de connaissances**. D'une autre façon, l'ontologie décrit un vocabulaire commun pour des chercheurs ou des utilisateurs qui ont besoin de partager des informations dans le même domaine.

2.4.3 Pourquoi utiliser une ontologie ?

Parmi les raisons qui pousseraient au developpement d'une ontologie :
- Partager une compréhension commune d'une structure d'information.
- Permettre la réutilisation des connaissances du domaine.
- Classification et organisation des ressources de données.
- Application des politiques de privacy.
- etc...

2.4.4 Les ontologies et leur application à la privacy

Les auteurs de [17] ont proposé une ontologie pour la privacy en mettent en place différents concepts. Leur ontologie se compose de plusieurs parties : une partie pour la gestion des ressources, une pour la gestion des identités des différentes entités et une autre pour gérer l'accès des entités aux ressources. Ils ont par la suite dérivé de cette ontologie une ontologie plus spécialisée en rapprochant les concepts qu'ils ont défini de ceux du e-commerce. De plus ils se sont inspirés de P3P pour définir des concepts comme *Statement*, *Recipient*, *Retention* et *Purpose* qu'ils ont reliés avec les concepts d'identité et de ressource pour définir une sorte de politique de privacy abstraite.
Ces travaux offrent une vision détaillée de ce que pourrait être une ontologie de privacy et peuvent servir comme première base pour des travaux sur l'utilisation des ontologies dans le e-commerce.

Dans [18] on propose de combiner des ontologies à des règles afin de préserver la privacy. L'ontologie proposée contient elle aussi plusieurs parties : une pour représenter la structure de données des utilisateurs, une pour catégoriser les utilisateurs et leur appartenance à telle ou telle organisation et une pour les types de données (elle décrit la hiérarchisation des classes, les propriétés des profils utilisateurs et les traces des transactions) et enfin une partie qui décrit le futur usage des données. Ils présentent deux scénarios

où ils utilisent des règles spécifiques pour raisonner et manipuler les données utilisateurs avec les droits requis et en respectants certaines conditions.

Afin de renforcer P3P et élargir son utilisation, les auteurs de [19] proposent une nouvelle plateforme basée sur deux concepts : le premier étant d'adopter Rei [4] comme langage de la politique pour sa structure extensible, sa sémantique en logique du premier ordre et son itégration de notions telles que les permissions, les inhibitions, les obligations etc... Le second étant de proposer un mécanisme d'évaluation des sites web basé sur les ontologies. Il permettra aux utilisateurs d'exprimer leurs connaissances et leurs opinions relatifs aux sites internet ce qui rassurera les utilisateurs quant à l'interaction avec un site particulier. Le résultat de l'évaluation servira comme condition à l'agent utilisateur pour permettre le déroulement d'une transaction ou pas.

Les ontologies furent utilisées dans des contextes très différents afin d'assurer la privacy, en revanche il n'existe pas de travaux qui décrivent de manière claire l'utilisation des ontologies lors du procesus de négociation. Des projets sont en cours comme l'utilisation du langage Rei pour la négociation de politiques mais aucun n'a encore vraiement abouti.

4. http ://rei.umbc.edu/

Chapitre 3

Le protocole de négociation

3.1 Introduction

Dans ce chapitre nous proposons un protocole de négociation qui termine en un nombre fini de rounds soit avec un succes soit avec un echec. Nous avons tenté de faire en sorte que le protocole soit équitable c'est à dire qu'il ne privilégie aucune des deux parties. On suppose l'existence d'une infrastructure PKI (Public Key Infrastructure) pour assurer l'authenticité. Les mechanismes qui mettent en place et qui gèrent les politiques P3P, la détection de violation de ces politiques ainsi que les mesures légales à entreprendre en cas de violation ne font pas partie de ce travail.

3.2 Les politiques de sécurité

Notre protocole s'inspire du travail de D.Walker [9] pour exprimer les politiques de sécurité.
Les politiques se composent d'éléments atomiques appelés *data element* et de *practice tags*. Un *data element* fait référence à une information relative à un individu (par exemple une adresse mail). Ces data elements sont organi-

sés hiérarchiquement en *data categories* qui représente un ensemble de data elements. Dans ce qui suit, on utilise des éléments et des catégories définis par W3C considérés comme un standard.
En plus de déclarer les types de données colléctés, les entités doivent spécifier comment ces données seront manipulées. Pour cela, les *data elements* sont associés aux *practice tags*. Il existe trois types de practice tags : *recipient*, *retention* et *purpose*. Recipient tags spécifient les entités qui auront acces à la donnée, retention tags spécifient combien de temps la donnée sera stockée et purpose tags spécifie comment la donnée sera utilisée. On suppose que les trois ensembles disjoints $RecipientTags$, $RetentionTags$ et $PurposeTags$ contiennent tous les recipient, retention et purpose tags possibles.
Formellement, la politique est un ensemble de statements $P = \{S_1, ..., S_n\}$ tel que chaque S_i est un tuple de la forme $S_i = (D_i, Rec_i, Ret_i, Pur_i)$ avec $D_i \subseteq AllData$, $Rec_i \subseteq RecipientTags$, $Ret_i \subseteq RetentionTags$, $Pur_i \subseteq PurposeTags$.

3.3 Les préférences du client

Le client, via l'agent utilisateur qui se trouve dans son navigateur, pourra définir grâce à une interface l'ensemble de ses préférences. Il pourra associer à chaque donnée qu'il divulguera les préférences quant à son utilisation. Pour chaque donnée il choisira de placer les politiques correspondantes dans un des deux ensembles : *Ideal* ou *Acc*. Ideal contiendra les politiques qui ne posent aucun souci à l'utilisateur. Acc contiendra les politiques que l'utilisateur aimerait ne pas permettre mais dont il pourra faire la concession. Les politiques restantes seront considérées automatiquement comme inacceptables et seront placées dans l'ensemble *Unacc*.
Ideal est l'ensemble des statements S_i^I tels que $S_i^I = (D_i, Rec_i^I, Ret_i^I, Pur_i^I)$ avec $D_i \subseteq AllData$, $Rec_i^I \subseteq RecipientTags$, $Ret_i^I \subseteq RetentionTags$, $Pur_i^I \subseteq$

$PurposeTags$.
Acc est l'ensemble des statements S_i^A tel que $S_i^A = (D_i, Rec_i^A, Ret_i^A, Pur_i^A)$ avec $D_i \subseteq AllData$, $Rec_i^A \subseteq RecipientTags$, $Ret_i^A \subseteq RetentionTags$, $Pur_i^A \subseteq PurposeTags$.
Unacc est l'ensemble des statements S_i^U tel que $S_i^U = (D_i, Rec_i^U, Ret_i^U, Pur_i^U)$ avec $D_i \subseteq AllData$, $Rec_i^U \subseteq RecipientTags$, $Ret_i^U \subseteq RetentionTags$, $Pur_i^U \subseteq PurposeTags$.
Les ensembles *Ideal*, *Acc* et *Unacc* sont disjoints deux à deux.

3.4 Les préférences du serveur

Se basant sur la spécification P3P, chaque data element lui est associé un tag *optionnal*. S'il est égal à *no* le serveur considère la donnée et la politique associée comme obligatoire (mandatory) et s'il est égal à *yes* le serveur considère la donnée et la politique associée comme optionnelle (optionnal). Deux ensembles alors se dégagent :
Mand est l'ensemble des statements S_i^M tel que $S_i^M = (D_i, Rec_i^M, Ret_i^M, Pur_i^M)$ avec $D_i \subseteq AllData$, $Rec_i^M \subseteq RecipientTags$, $Ret_i^M \subseteq RetentionTags$, $Pur_i^M \subseteq PurposeTags$.
Opt est l'ensemble des statements S_i^O tel que $S_i^O = (D_i, Rec_i^O, Ret_i^O, Pur_i^O)$ avec $D_i \subseteq AllData$, $Rec_i^O \subseteq RecipientTags$, $Ret_i^O \subseteq RetentionTags$, $Pur_i^O \subseteq PurposeTags$.
L'ensemble *Mand* contient les données ainsi que les politiques qui doivent être présents dans la politique finale sinon la négociation échoue. L'ensemble *Opt* contient quant à lui les données ainsi que les politiques que le serveur aimerait voir présents dans la politique finale. Leur absence ne met pas en périle le succes de la négociation.

3.5 Spécification du protocole

3.5.1 Préambule

La négociation consiste en plusieurs rounds. Durant chaque round un des deux participants fait une proposition et l'autre participant prend la décision d'accepter ou de rejeter cette proposition. La première proposition vient du serveur et contient sa politique en entier, ses alternatives et une variable qui indique qu'en cas d'échec de la comparaison si le serveur est prêt à négocier ou pas. Le client va de son coté comparer la politique envoyée par le serveur avec l'ensemble de ses préférences. Si pour chaque donnée demandée par le serveur, les politiques associées ne font pas partie des préférences inacceptables du client alors la comparaison aura réussie et le client accepte alors la proposition. Si en revanche il existe une seule donnée pour laquelle les politiques ne concordent pas alors la comparaison échoue. A ce moment le client consulte la variable incluse dans le premier message, si la variable ne permet pas une négociation alors l'échange s'arrête et la transaction est annulée. Dans le cas contraire le processus de négociation peut commencer.

La négociation va consister en un ensemble de négociations simultanément de politiques de chaque data element effectuées en parallèle. Si toutes les négociation réussissent alors les résultats sont combinés pour produire la politique finale. Si une des négociations échoue alors toute la négociation échoue et la transaction sera alors annulée.

3.5.2 Les échanges de messages

Les différents rounds de la négociation se présentent comme suit :

Round 1 le client initie le premier round en se connectant au serveur

et en demandant sa politique de privacy. Le serveur répond en envoyant sa politique par défaut dans le message $Proposal_1$. Il envoie également ses *alternatives* et la variable *N*. Si la comparaison réussie alors le client envoie le message $Accept_1$ sinon si elle échoue le client va, selon la valeur de la variable *N*, savoir si le serveur désire négocier. Si oui le Round 2 commence sinon la transaction est annulée et le client envoie le message *Reject*.

Round 2 si une seule des politiques ne concorde pas et si le serveur accepte de négocier, on va commencer à négocier simultanément les politiques de chaque data element. Le client commence le second round en envoyant la liste de ses politiques idéales dans le message $Proposal_2$. Le serveur va comparer si l'ensemble des politiques obligatoires pour un data element D_i est inclus dans l'ensemble des politiques idéales du client pour le même data element D_i. Si oui alors le serveur envoie le message $Accept_2$ sinon le round 3 commence.

Round 3 le serveur va alors utiliser la politique idéal du client envoyée dans $Proposal_2$ pour former sa contre proposition. Il va envoyer au client un message $Proposal_3$ qui contient les politiques idéales du client ainsi que ses politiques à lui obligatoires relatives au data element D_i.
Le client de son coté va tester si aucune des politiques envoyées dans $Proposal_3$ n'appartient à ses politiques inacceptables. Si oui alors il va envoyer le message $Accept_3$. Sinon s'ils existent des politiques incluses dans l'ensemble inacceptable alors il va les remplacer grâce aux alternatives ; et les politiques non incluses dans l'ensemble inacceptable seront alors acceptées et stockées afin de former la politique finale.

Dans le cas où le client arrive à trouver une alternative aux politiques qui posent problème alors il va remplacer chaque politique et pour chaque politique remplacée vérifier si elle n'est pas incluse dans l'ensemble inacceptable

relatif au même data element. Si c'est le cas alors la politique est acceptée sinon elle devra être remplacée par une seconde alternative, etc... jusqu'à épuisement de toutes les alternatives proposées par le serveur. Si le client a épuisé toutes les alternatives sans arriver à un accord alors le client envoie au serveur une proposition $Proposal_4$ qui contient le data element D_i pour lequel une entente n'a pas été trouvée et le round 4 commence. Sinon si le client arrive à un accord avant l'épuisement des alternatives alors il envoie le message $Accept_4$.

Round 4 le serveur reçoit à travers le message $Proposal_4$ le data element D_i qu'il doit remplacer. On suppose que du coté serveur il existe une ontologie qui conceptualise son domaine d'application. Le Serveur va alors grâce à un raisonneur, par exemple RQL (the RDF Query Langage), inférer sur l'ontologie et récupérer un data element équivalent à D_i et l'envoyer ainsi que la politique qui lui est associée dans le message $Proposal_5$.
Le client va tester de son coté si la politique du nouveau data element n'appartient pas à ce qu'il considère lui comme inacceptable. Si c'est la cas alors il envoie un message *Accept* et la négociation aura alors *réussie*. Sinon il envoie le message *Reject* et la négociation aura définitivement *échouée*.

3.5.3 La stratégie de négociation

Lors de la formulation des messages, pour le bon déroulement de la négociation et afin de préserver le caractère équitable des échanges, les deux négociants doivent suivrent une certaine stratégie. Dans ce qui suit on décrit un ensemble de règles qui doivent être respéctées :

Règle n°1 Le serveur envoie dans le premier round sa politique par défaut sans distinction entre les éléments obligatoires et ceux optionnels.
$Proposal_1 = \{\sum D_i \; tel \; que \; D_i = (\{Rec_i^M \bigcup Rec_i^O\}, \{Ret_i^M \bigcup Ret_i^O\}, \{Pur_i^M \bigcup Pur_i^O\})$;

alternatives ;N}

Règle n°2 Le client va alors comparer la politique envoyée dans le message précédent à ses propres préférences :
$\forall D_i \in$ Unacc, si $D_i \in Proposal_1$ alors : $((Rec_i^U \bigcap (Rec_i^M \bigcup Rec_i^O)) = \varnothing$ et $(Ret_i^U \bigcap (Ret_i^M \bigcup Ret_i^O)) = \varnothing$ et $(Pur_i^U \bigcap (Pur_i^M \bigcup Pur_i^O)) = \varnothing$) celà signifie qu'aucune des politiques envoyées par le serveur ne fait partie des politiques inacceptables du client. Dans ce cas, ce dernier envoie le message $Accept_1$

Règle n°3 Sinon il va tester la variable N, si $N = False$ alors il envoie le message *Reject* sinon si $N = True$ il entame le second round.

Règle n°4 Le client envoie la liste de ses préférences idéales dans $Proposal_2$: $Proposal_2 = Ideal = \sum S_i^I$ tel quel $S_i^I = (D_i, Rec_i^I, Ret_i^I, Pur_i^I)$

Règle n°5 A ce stade, la négociation s'effectue individuellement pour chaque data element D_i. Le serveur va tester, si toutes ses préférences relatives à D_i, qu'il considère comme obligatoires font parti de la proposition du client (concernant toujours D_i) :
Si $(Rec_i^M \subseteq Rec_i^I)$ et $(Ret_i^M \subseteq Ret_i^I)$ et $(Pur_i^M \subseteq Pur_i^I)$ alors le serveur accepte la proposition du client et envoie le message $Accept_2 = (D_i, Rec_i^M, Ret_i^M, Pur_i^M)$.

Règle n°6 Si la condition ci dessus n'est pas vérifiée alors le serveur envoie une contre proposition $Proposal_3$ qui contient les politiques idéales du client ainsi que les politiques obligatoires du serveur relatives à D_i :
$Proposal_3 = \{(D_i, \{Rec_i^I \cup Rec_i^M\}, \{Ret_i^I \cup Ret_i^M\}, \{Pur_i^I \cup Pur_i^M\})\}$

Règle n°7 Le client va tester si aucune des politiques reçues dans $Proposal_3$ ne fait partie de ses politiques inacceptables :

$Proposal_3 \bigcap Unacc(D_i) = ?$ cela revient à tester $Mand(D_i) \bigcap Unacc(D_i) = ?$ vu que *Ideal* et *Unacc* sont disjoints par définition.

Si $Mand(D_i) \bigcap Unacc(D_i) = \varnothing$ c'est à dire : $(Rec_i^M \bigcap Rec_i^U = \varnothing)$ et $(Ret_i^M \bigcap Ret_i^U = \varnothing)$ et $(Pur_i^M \bigcap Pur_i^U = \varnothing)$ alors le client envoie le message $Accept_3$ tel que :
$Accept_3 = (D_i, Rec_i^M, Ret_i^M, Pur_i^M)$ les politiques relatives à D_i feront partie des politiques *Acc* et *Ideal* du client.

Règle n°8 Dans le cas $Mand(D_i) \bigcap Unacc(D_i) \neq \varnothing$ alors soit :
$A = \{(D_i, Rec_i, Ret_i, Pur_i)\}$ tel que : $(Rec_i = Rec_i^M \bigcap Rec_i^U)$ et $(Ret_i = Ret_i^M \bigcap Ret_i^U)$ et $(Pur_i = Pur_i^M \bigcap Pur_i^U)$. A contient de ce fait les politiques qui sont d'un côté obligatoires pour le serveur et d'un autre impossibles à concéder pour la client.

En revanche, les politiques $(Rec_i^M - Rec_i)$, $(Ret_i^M - Ret_i)$ et $(Pur_i^M - Pur_i)$ relatives au data element D_i sont acceptées. Elle appartiennent par déduction à l'ensemble *Ideal* et/ou à l'ensemble *Acc*. Elles doivent être stockées afin de produire la politique finale.

Règle n°9 Chaque politique du data element D_i de A devra être remplacée selon les alternatives proposées par le serveur.
Soit $A' = (D_i, Rec'_i, Ret'_i, Pur'_i)$ les nouvelles politiques associées à D_i.
Le client va tester : si $Rec'_i \bigcap Rec_i^U = \varnothing$ alors la nouvelle politique *Recipient* est acceptée ; elle est stockée afin de former la politique finale. Sinon le client doit trouver une nouvelle alternative à cette politique. Le même travaille est effectué pour Ret'_i et Pur'_i.
Si avant d'avoir épuisé toutes les alternatives on réussit à obtenir : $(Rec'_i \bigcap Rec_i^U = \varnothing)$ et $(Ret'_i \bigcap Ret_i^U = \varnothing)$ et $(Pur'_i \bigcap Pur_i^U = \varnothing)$ alors le client envoie le message $Accept_4$:

$Accept_4 = \{(D_i, \{(Rec_i^M - Rec_i) \bigcup Rec_i'\}, \{(Ret_i^M - Ret_i) \bigcup Ret_i'\}, \{(Pur_i^M - Pur_i) \bigcup Pur_i'\})\}$.
Sinon il envoie au serveur le message $Proposal_4 = D_i$ qui contient le data element pour lequel aucune politique commune n'a pu être trouvée.

Règle n°10 Le serveur doit remplacer le data element D_i par un autre data element équivalent sémantiquement au sein de son ontologie. Soit D_i' le nouveau data element, le serveur va récupérer la politique de privacy associée à D_i' pour former le message $Proposal_5$:
$Proposal_5 = (D_i', Rec_i', Ret_i', Pur_i')$.

Règle n°11 Le client va tester : si $(Rec_i' \bigcap Rec_i^M = \varnothing)$ et $(Ret_i' \bigcap Ret_i^M = \varnothing)$ et $(Pur_i' \bigcap Pur_i^M = \varnothing)$ alors il envoie le message *Accept* et la négociation aura alors réussie pour le data element D_i initial. Dans le cas contraire, il envoie le message *Reject* la négociation aura échouée pour D_i et la transaction sera interrompue.

Remarque La négociation s'effectue au complet sur la partie *Mandatory* (obligatoire) de la politique. Quant à la partie *optional* (optionnelle) elle se déroulera de la même manière sauf qu'elle ne prendra en compte ni les ontologies ni les alternatives.
Au niveau du round 3 coté client, les politiques optionnelles seront comparées aux politiques inacceptable du data element en question. Soit $B = \{(D_j, Rec_j, Ret_j, Pur_j)\}$ tel que : $(Rec_j = Rec_j^O \bigcap Rec_j^U)$ et $(Ret_j = Ret_j^O \bigcap Ret_j^U)$ et $(Pur_j = Pur_j^O \bigcap Pur_j^U)$
Si $(Ret_j = \varnothing)$ alors cette politique est acceptée sinon seules les politiques $\{Rec_j^O - Rec_j^U\}$ seront acceptées et le reste sera définitivement refusé. Idem pour Rec_j et Pur_j.

3.5.4 Exemple de négociation

Nous allons prendre comme exemple une transaction pour l'achat en ligne d'un téléphone mobile avec abonnement. L'exemple va porter sur la négociation des politiques de deux data elements *le numéro de carte de crédit (ou N˚CC)* et *l'adresse mail (ou @mail)*. La négociation concernera la partie **obligatoire** de la politique de privacy.

Les politiques

La politique obligatoire du serveur concernant les deux data element est comme suit :
$Mandatory = \{(N^{\circ}CC, \{ours\}, \{no-retention\}, \{current\}),$
$(@mail, \{ours, delivery\}, \{indefinitly\}, \{current, telemarketing, admin\})\}$.

La politique optionnelle du serveur est :
$Optionnal = \{(N^{\circ}CC, \{\}, \{business-practices\}, \{pseudo-analysis\}),$
$(@mail, \{same, other-recipient\}, \{\ \}, \{tailoring\})\}$.

La politique que le client consoit d'accorder sans aucun souci est :
$Ideal = \{(N^{\circ}CC, \{ours\}, \{no-retention\}, \{current\}),$
$(@mail, \{ours, delivery, same\}, \{no{-}retention, business{-}practices\}, \{current\})\}$.

La politique qu'il n'aimera ne pas attribuer mais qu'il concedera ni nécessaire : $Acc = \{(N^{\circ}CC, \{\ \}, \{business-practices\}, \{\ \}),$
$(@mail, \{delivery, other{-}recipient\}, \{stated{-}purpose\}, \{admin, develop\})\}$.

La politique dont il ne peut faire concession est :
$Unacc = \{(N^{\circ}CC, \{delivery, same, unrelated, other{-}recipient, public\}, \{stated{-}purpose, legal{-}requirement, indefinitly\}, \{admin, develop, tailoring, pseudo{-}analysis, pseudo-decision, individual-analysis, contact, historical\}),$

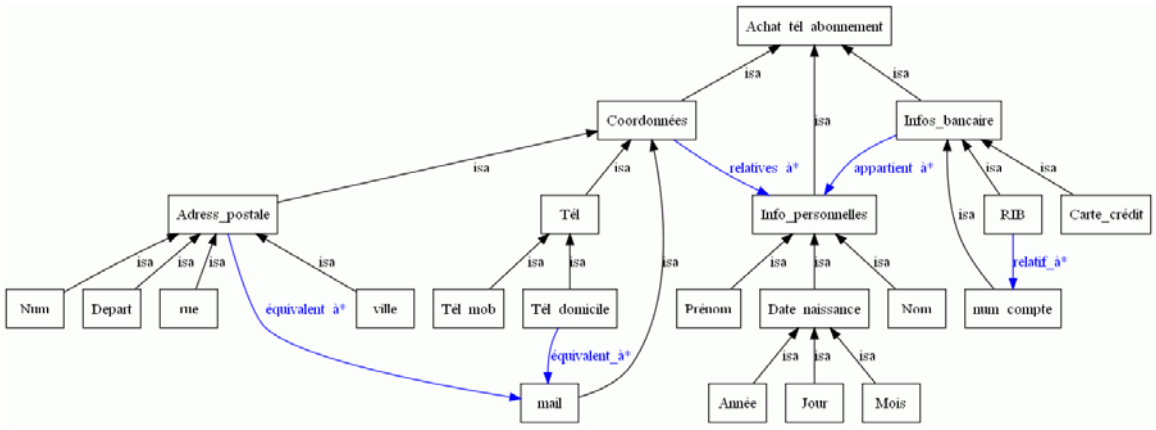

FIGURE 3.1 – L'ontologie du serveur pour l'achat de téléphone mobile en ligne.

$(@mail, \{unrelatted, public\}, \{legal-requirement, indifinitly\}, \{tailoring, pseudo-analysis, pseudo-decision, individual-analysis, contact, telemarketing, other-purpose, historical\})\}$.

Les alternatives expriment d'autres possibilités de politiques que le serveur est prêtes à accepter. Pour les besoins de notre exemple, supposons que pour le data element *@mail*, le serveur propose deux alternatives pour la *retention*, une seule alternative pour *purpose* et aucune pour *recipient*.
Si $D_i = @mail$ alors
Pour *Retention* :

- $Alternative_1 : Ret'_i \leftarrow \{legal - requirement\}$,
- $Alternative_2 : Ret'_i \leftarrow \{stated - purpose\}$.

Pour *Purpose* :

- $Alternative_1 : Pur'_i \leftarrow \{contact\}$.

Soit en (cf. figure n° 3.1) un exemple de l'ontologie du serveur qui défini-

rait l'achat de téléphone mobile en ligne. L'exemple a été généré par Protégé [1].

La négociation

Round 1 Le serveur envoie sa politique dans $Proposal_1 = \{(N^{\circ}CC, \{ours\}, \{no-retention, business-practices\}, \{current, pseudo-analysis\}),$
$(@mail, \{ours, delivery, same, other-recipient\},$
$\{indefinitly\}, \{current, telemarketing, admin, tailoring\}), Alternatives, N = True\}$. (Règle N°1)
$\exists\ (D_i = N^{\circ}CC \in Unacc)\ et\ (D_i \in Proposal_1)$ et $\{current, pseudo-analysis\} \bigcap \{admin, develop, tailoring, pseudo-analysis, pseudo-decision, individual-analysis, contact, historical\} \neq \varnothing$. La comparaison a échouée et $(N = True)$
alors on entame le second round (règle N°2 et N°3).

Round 2 Le client envoie la liste de ses idéaux dans $Proposal_2 = \{(N^{\circ}CC, \{ours\}, \{no-retention\}, \{current\}),$
$(@mail, \{ours, delivery, same\}, \{no-retention, business-practices\}, \{current\})\}$.
(Règle N°4)

- Négociation des politiques du data element N°CC :
  Le serveur va tester : $(\{ours\} \subseteq \{ours\})$ et $(\{no-retention\} \subseteq \{no-retention\})$ et $(\{current\} \subseteq \{current\})$ alors le serveur envoie $Accept_2 = \{(N^{\circ}CC, \{ours\}, \{no-retention\}, \{current\}$ (Règle N°5)

- Négociation des politiques du data element @mail :
  Le serveur va tester : $(\{ours, delivery\} \subseteq \{ours, delivery, same\})$ et $(\{indefinitly\} \nsubseteq \{no-retention, business-practices\})$ et $(\{current, telemarketing, admin\} \nsubseteq \{current\})$. Le serveur envoie

1. Protégé est un éditeur d'ontologie et un environnement pour les bases de connaissances. Il a été créé à l'université de Sandford ; il est gratuit et à code source libre. http ://protege.stanford.edu/

$Proposal_3$. (Règle N°6)

Round 3 $Proposal_3 = \{(@mail, \{ours, delivery, same\},$
$\{indefinitly, no{-}retention, business{-}practices\}, \{current, telemarketing, admin, \})\}$

Le client va tester : $(\{ours, delivery, same\} \bigcap \{unrelatted, public\} = \varnothing)$ et $(\{indefinitly, no - retention, business - practices\} \bigcap \{legal - requirement, indifinitly\} = \{indefinitly\})$ et $(\{current, telemarketing, admin, \} \bigcap \{tailoring, pseudo{-}analysis, pseudo{-}decision, individual{-}analysis, contact, telemarketing, other{-}purpose, historical\} = \{telemarketing\})$.
Soit $A = (@mail, \{\ \}, \{indefinitly\}, \{telemarketing\})$ le client va remplacer ces politiques par les alternatives proposées par le serveur dans $Proposal_1$. En revanche les politiques {ours,delivery,same} pour les recipients et {current,admin,} pour purpose sont acceptée et stockées. (Règle N°8)

Selon les alternatives citées plus haut : $A' = (@mail, \{\ \}, \{legal - requirement\}, \{contact\})$
Le client va tester si les nouvelles politiques ne sont pas inacceptables pour lui : $\{legal{-}requirement\} \bigcap \{legal{-}requirement, indifinitly\} = \{legal{-}requirement\}$ et $\{contact\} \bigcap \{tailoring, pseudo{-}analysis, pseudo{-}decision, individual{-}analysis, contact, telemarketing, other{-}purpose, historical\} = \{contact\}$ alors le client devra chercher de nouvelles possibilités. Or il n'existe plus d'alternative possible pour la politique *purpose* du data element *@mail* alors le client envoie $Proposal_4 = \{@mail\}$. (Règle N°9)

Round 4 Soit (cf. figure n°3.2) une partie de l'ontologie définie pour la transaction d'achat de téléphone en ligne. Nous remarquons qu'il existe des relations *d'équivalence* entre certains concepts. Un raisonneur comme RQL

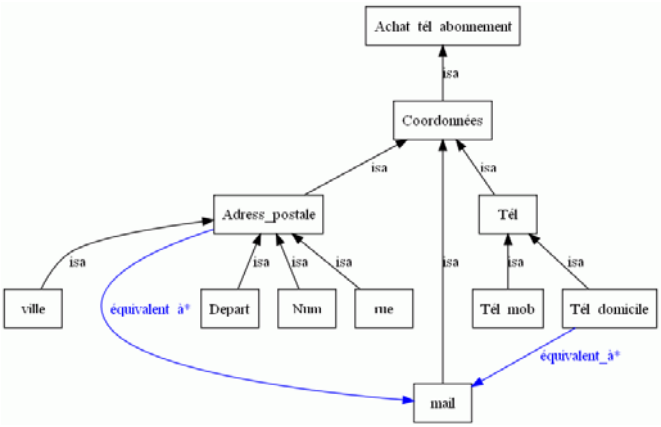

FIGURE 3.2 – Partie de l'ontologie relative aux coordonnées.

peut requêter l'attribut équivalent à *@mail* dans notre cas le résultat serait Tél domicile.

Soient les politiques obligatoires relatives au *Tél domicile* : $\{delivery, same\}$ pour recipient, $\{stated-purpose, business-practices\}$ pour retention, $\{current, admin\}$ pour purpose ; et soit les politiques inacceptables du client relatives au *Tél domicile* : $\{other-recipient, unrelated, public\}$ pour recipient, $\{current, admin\}$ pour retention et $\{telemarketing, contact\}$ pour purpose.

Le client va tester : $\{delivery, same\} \bigcap \{other-recipient, unrelated, public\} = \varnothing$ et $\{stated-purpose, business-practices\} \bigcap \{current, admin\} = \varnothing$ et $\{current, admin\} \bigcap \{telemarketing, contact\} = \varnothing$ alors il envoie le message ***Accept*** (Règle N°11). La négociation a **réussie** et la politique finale est $\{(N°CC, \{ours\}, \{no-retention\}, \{current\}), (Tél\ domicile, \{delivery, same\}, \{stated-purpose, business-practices\}, \{current, admin\})\}$.

Conclusion

Le protocole proposé est un protocole de négociation des politiques de privacy relatives aux informations recueillies lors d'une navigation et qui termine au bout d'un certain nombre d'échanges. Il propose l'utilisation d'un certain nombre d'alternatives, proposées par le serveur, afin de permettre d'autres éventualités au cas où une politique d'une certaine donnée ne concorderait pas ; donnant ainsi une chance supplémentaire à la négociation de réussir. L'ontologie du domaine du serveur est utilisée comme ultime recours afin de remplacer une information sur laquelle aucune entente n'a pu être trouvée, par une autre information équivalente.
Afin de spécifier leurs préférences, les utilisateurs devront avoir des connaissances préalables sur la syntaxe d'une politique de privacy ; un minimum de pré requis s'avéreront nécessaires.
Le développement d'une ontologie complète d'un domaine du serveur et son exploitation par un raisonneur permettrait de mettre en évidence la plus-value de ce travail. Il serait aussi nécessaire de s'appliquer à la sécurisation des messages échangés grâce à l'ajout de time stamps, d'identifiants et de signatures.

Bibliographie

[1] Maknavicius Maryline and Bekara Kheira. Enabling user privacy in a federated environment. In *en cours de soumission*, 2009.

[2] L. Wilbanks. The impact of personally identifiable information. *IT Professional*, 9(4) :62–64, July-Aug. 2007.

[3] Joe Carmichael and Alfonso Gutierrez. Identity management whitepaper : An introduction to the terms and concepts of identity management. *Best Practice Reports*, 10 2005.

[4] Maknavicius Maryline and Bekara Kheira. Enabling collaboration between heterogeneous circles of trust through innovative identity solutions. In *en cours de soumission*, 2009.

[5] Davoux Alexis, Defline Jean-Christophe, Francesconi Ludovic, Maknavicius Maryline, Bekara Kheira, Gola Romain, Lezoray Jean-Baptiste, and Etchebarne Vincent. Federation of circles of trust and secure usage of digital identity. In *Collaboration and the Knowledge Economy : Issues, Applications, Case Studies, Paul Cunningham and Miriam Cunningham (Eds), IOS Press*, 2008.

[6] M. Bennicke and Langendorfer. Towards automatic negotiation of privacy contracts for internet services. In *Networks, 2003. ICON2003. The 11th IEEE International Conference on*, pages 319–324, Sept.-1 Oct. 2003.

[7] Steven M. Bellovin. The puzzle of privacy. *Security & Privacy, IEEE*, 6(5) :88–88, Sept.-Oct. 2008.

[8] Klaus Kursawe, Gregory Neven, and Pim Tuyls. Private policy negotiation. In *In Financial Cryptography 2006, volume 4107 of LNCS*, pages 81–95, 2006.

[9] D.D. Walker, E.G. Mercer, and K.E. Seamons. Or best offer : A privacy policy negotiation protocol. In *Policies for Distributed Systems and Networks, 2008. POLICY 2008. IEEE Workshop on*, pages 173–180, June 2008.

[10] Sören Preibusch. *Implementing Privacy Negociation In E-Commerce*, pages 604–615. Springer-Verlag, 2006.

[11] With Fine-Grained Policy and Stephen E. Levy. Improving understanding of website privacy policies. In *In WWW 05 : Proceedings of the 14th international conference on World Wide Web*, pages 480–488. ACM Press, 2005.

[12] Simon Byers, Lorrie Faith Cranor, Dave Kormann, and Patrick Mcdaniel. Searching for privacy : Design and implementation of a p3p-enabled search engine. In *In 2004 Workshop on Privacy Enhancing Technologies (PET2004*, pages 314–328, 2004.

[13] Anna Squicciarini, Marco Casassa Mont, Abhilasha Bhargav-Spantzel, and Elisa Bertino. Automatic compliance of privacy policies in federated digital identity management. In *POLICY '08 : Proceedings of the 2008 IEEE Workshop on Policies for Distributed Systems and Networks*, pages 89–92, Washington, DC, USA, 2008. IEEE Computer Society.

[14] M. Maaser and P. Langendoerfer. Automated negotiation of privacy contracts. In *Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International*, volume 1, pages 505–510 Vol. 2, July 2005.

[15] Haifei Li, D. Ahn, and P.C.K. Hung. Algorithms for automated negotiations and their applications in information privacy. In *e-Commerce Technology, 2004. CEC 2004. Proceedings. IEEE International Conference on*, pages 255–262, July 2004.

[16] Scott Buffett, Keping Jia, Y Liu, Bruce Spencer, and Fang Wang. Negotiating exchanges of p3p-labeled information for compensation. *Computational Intelligence*, 20 :663–677, 2004.

[17] M. Hecker, T.S. Dillon, and E. Chang. Privacy ontology support for e-commerce. *Internet Computing, IEEE*, 12(2) :54–61, March-April 2008.

[18] Yuh-Jong Hu, Hong-Yi Guo, and Guang-DeLin. Semantic enforcement of privacy protection policies via the combination of ontologies and rules. In *Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC '08. IEEE International Conference on*, pages 400–407, June 2008.

[19] P. Kolari, Li Ding, G. Shashidhara, A. Joshi, T. Finin, and L. Kagal. Enhancing web privacy protection through declarative policies. In *Policies for Distributed Systems and Networks, 2005. Sixth IEEE International Workshop on*, pages 57–66, June 2005.

[20] Ninghui Li, Ting Yu, and Annie I. Antón. A semantics-based approach to privacy languages. Technical report, in IEEE Symposium on Security and Privacy, 2003.

[21] Cheonshu Park and Joochan Shon. A study on the web ontology processing system. In *Advanced Communication Technology, 2005, ICACT 2005. The 7th International Conference on*, volume 2, pages 1035–1038, 0-0 2005.

[22] D.Z.G. Garcia and M. Toledo. A web service privacy framework based on a policy approach enhanced with ontologies. In *Computational Science and Engineering Workshops, 2008. CSEWORKSHOPS '08. 11th IEEE International Conference on*, pages 209–214, July 2008.

[23] B. Sarder and S. Ferreira. Developing systems engineering ontologies. In *System of Systems Engineering, 2007. SoSE '07. IEEE International Conference on*, pages 1–6, April 2007.

[24] L.A.F. Martimiano, M.R.P. Goncalves, and E. dos Santos Moreira. An ontology for privacy policy management in ubiquitous environments. In *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, pages 947–950, April 2008.

[25] Anna C. Squicciarini, Abhilasha Bhargav-spantzel, Alexei Czeskis, and Elisa Bertino. Traceable and automatic compliance of privacy policies in federated digital identity management.

[26] Fabien Gandon and Norman Sadeh. Gestion de connaissances personnelles et contextuelles, et respect de la vie privée. In Nada Matta, editor, *Ingénierie des Connaissances*, pages 5–16, Lyon France, 05 2004. Presses universitaires de Grenoble.

[27] Freedman Michael, Nissim Kobbi, and Benny Pinkas. *Advances in Cryptology*, chapter Efficient Private Matching and Set Intersection, pages 1–19. Springer-Verlag, 2004.

Table des figures

Lexique

1. RDF : pour Resource Description Framework est un vocabulaire XML enrichi qui définit une syntaxe et une sémantique pour exprimer les ontologies.

2. Timestamp : est une séquence de caractères qui sert à situer un événement dans le temps. Il contient en général une date et une heure.

3. W3C définit un standard pour exprimer la syntaxe et la sémantique des politiques P3P.

   – L'élément <POLICIES > : il rassemble une ou plusieurs politiques P3P dans le même fichier. Cette caractéristique permet d'optimiser les performances du réseau et du stockage. Cet élément est l'élément racine des fichiers de politiques.

   – L'élément <POLICY> : il contient une politique P3P complète. Chaque politique doit contenir un seul élément <POLICY>. L'élément <POLICY> doit contenir un élément <ENTITY> identifiant l'entité légale exposant les pratiques de confidentialité contenues dans la politique. En outre, l'élément <POLICY> doit contenir un élément <ACCESS> , un ou plusieurs éléments <STATEMENT> , un élément <DISPUTES-GROUP> , un schéma de données P3P et une ou plusieurs extensions.

     <POLICY>

     name : le nom de la politique, utilisé comme identificateur de fragments pour appeler la politique.
     discuri : l'adresse URI de la déclaration de confidentialité en langage

naturel.
opturi : l'adresse URI des instructions permettant aux utilisateurs d'accepter ou de refuser l'utilisation des données les concerant.
xml :lang : la langue dans laquelle la politique est exprimée.

- L'élément <ENTITY> : donne un description précise de l'entité légale qui expose ses pratiques de confidentialité.

- L'élément <ACCESS> : offre la possibilité à un individu de consulter les données identifiées et adresser ses questions et ses inquiétudes au fournisseur de services.

- L'élément <DISPUTES> : une politique devrait contenir un élément <DISPUTES-GROUP> contenant à son tour un ou plusieurs éléments <DISPUTES>. Ces éléments décrivent les procédures de résolution de litiges à observer en cas de contestation des pratiques de confidentialité.

- L'élément <REMEDIES> : chaque élément <DISPUTES> devrait contenir un élément <REMEDIES> qui définit les réparations possibles en cas de violation de la politique.

Les déclarations : décrivent les traitements appliqués aux types de données spécifiques.

- L'élément <STATEMENT > : c'est une sorte de conteneur regroupant un élément <PURPOSE>, un élément <RECIPIENT>, un élément <RETENTION>, un élément <DATA-GROUP> et en option un élément <CONSÉQUENCE> ainsi qu'une ou plusieurs extensions. Toutes les données référencées par l'élément <DATA-GROUP> sont manipu-

lées conformément aux divulgations présentes dans les autres éléments contenus par la déclaration.

- L'élément <PURPOSE> : tout élément <STATEMENT> doit contenir un élément <PURPOSE> qui exprime une ou plusieurs intentions concernant la collecte ou l'utilisation des données. Les sites doivent classer leurs pratiques vis à vis des données dans l'une ou plusieurs des catégories d'intention suivantes :
  - <**current**/> : les renseignements peuvent servir au fournisseur de service afin d'achever le processus pour lequel ceux-ci ont été fournis.
  - <**admin**/> : administration du site web et du système.
  - <**tailoring**/> : les renseignements peuvent servir à l'ajustement ou la modification du contenu ou de l'aspect du site.
  - <**pseudo-analysis**/> : les renseignements peuvent servir à la création ou l'élaboration d'un enregistrement concernant un individu (ou un ordinateur) particulier. Ce profile servira à déterminer les habitudes les centres d'interêts ou les autres caractéristiques d'un individu afin de prendre une decision l'affectant directement.
  - <**individual-analysis**/> :les renseignements peuvent servir à déterminer les habitudes, les centres d'interêts ou les autres caractéristiques des individus et être combinées avec des données identifiées dans un but de recherche, d'analyse ou d'études.
  - <**contact**/> : contacter des visiteurs pour la commercialisation de services ou produits.
  - <**historical**/> : les renseignements peuvent être archivés ou stockés à des fins de conservation des antécédents sociaux comme éxigé par une loi existante.
  - <**telemarketing**/> : contacter des visiteur pour la commercialisation de services ou de produits par téléphone.

- L'élément <RECIPIENT> : définit l'entité légale ou le domaine , hormi le fournisseur de services et ses agents, où les données peuvent être distribuées. Il doit contenir un ou plusieurs des destinataires suivants :

  - <**ours**/> : nous même et/ou des entités agissant en tant qu'agents ou les entités pour le compte desquelles nous agissons comme tel.
  - <**delivery**/> : service de livraison suivant peut être d'autres pratiques.
  - <**same**/> : personnes morales suivant nos pratiques.
  - <**other-recipient**/> : personnes morales suivant des pratiques différentes.
  - <**unrelated**/> : les personnes morales dont les pratiques vis à vis de l'utilisation des données sont inconnues du fournisseur de service original.
  - <**public**/> : les tribunes publiques, telles que les annuaires publics, les annuaires de CD-ROM commerciaux.

- L'élément <RETENTION> : définit le type de politique de rétention en vigueur. Il doit contenir un ou plusieurs des éléments suivants :

  - <**no-retention**/> : les renseignements sont conservés pour la brève durée nécessaire à leur utilisation au cours d'une seule opération en ligne. Les renseignements doivent être détruits immédiatement après cette opération et ne doivent pas apparaître dans un journal, ni être archivés ou stockés d'une manière quelconque.
  - <**stated-purpose**/> : les renseignements sont conservés pour satisfaire à l'intention déclarée. Les renseignements doivent être détruits le plus tôt posible.

- <**legal-requirement**/> : comme exigé par la loi ou en responsabilité selon les lois en vigueur : les renseignements sont conservés pour satisfaire à une intention déclarée mais la période de rétention est plus importante du fait d'une obligation légale ou d'une reponsabilité légale.
- <**business-practices**/> : selon les pratiques commerciales du fournisseur de service : les renseignements sont conservrées sous couvert des pratiques commerciales déclarées par les fournisseurs de services. Les sites doivent avoir une politique de rétention établissant un calendrier de destruction.
- <**indefinitly**/> : les renseignements sont conservés pour une durée indéterminée. Cette option reflète l'absence d'une politique de rétention.

- L'élément <DATA-GROUP et DATA> : tout élément <STATEMENT> doit contenir au moins un élément<DATA-GROUP> qui contient à son tour un ou plusieurs élément <DATA> Les éléments <DATA> servent à décrire les types de données collectées par un site.
  <DATA> : décrit les données à transférer. Doit contenir un élément <OPTIONAL> qui indique si le site impose ou non aux visiteurs de soumettre cet élément de données pour accéder à une ressource ou achever une transaction. La veuleur "no" signifie que l'élément de donnée n'est pas optionnel (donc obligatoire).

- Les catégories et l'élément <CATEGORIES> : les catégories sont des éléments dans les éléments de données offrant aux agents utilisateurs des indications concernant les usages prévus des données. Les catégories sont essentielles et facilitent la mise en oeuvre et l'utilisation des agents utilisateurs P3P. Elle permettent aux utilisateurs d'exprimer des préférences et règles plus générales pour échanger leurs données.

– <**physical**/> : ce sont les coordonnées physiques.
– <**online**/> : ce sont les coordonnées en ligne.
– <**purchase**/> : ce sont les informations d'achat.
– <**financial**/> : ce sont les informations financières.
– etc ....