AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur : ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite de ce travail expose à des poursuites pénales.

Contact : portail-publi@ut-capitole.fr

LIENS

Code la Propriété Intellectuelle – Articles L. 122-4 et L. 335-1 à L. 335-10
Loi n°92-597 du 1er juillet 1992, publiée au *Journal Officiel* du 2 juillet 1992
http://www.cfcopies.com/V2/leg/leg-droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 1 Capitole (UT1 Capitole)*

Présentée et soutenue le *12/09/2014* **par :**

Arnaud Oglaza

Système d'aide à la décision pour la protection des données de vie privée

JURY

Jean-Paul Arcangeli	Maître de Conférences, Université Paul Sabatier	Président du Jury
Daniela Grigori	Professeur d'université, Université Paris Dauphine	Rapporteur
Nora Cuppens-Boulahia	Directeur de recherches, Technopôle Brest-Iroise	Rapporteur
Sophie Chabridon	Maître de Conférences, Télécom Paris Tech	Examinateur
Pascale Zaraté	Professeur d'université, Université Toulouse 1 Capitole	Directeur de thèse
Romain Laborde	Maître de Conférences, Université Paul Sabatier	Encadrant de thèse

École doctorale et spécialité :
MITT : Domaine STIC : Réseaux, Télécoms, Systèmes et Architecture
Unité de Recherche :
IRIT - Institut de Recherche en Informatique de Toulouse (UMR 5505)
Directeur(s) de Thèse :
Pascale Zaraté et *Romain Laborde*
Rapporteurs :
Daniela Grigori et *Nora Cuppens-Boulahia*

Liste des publications

Articles de revues internationales

— **Sophie Chabridon, Romain Laborde, Thierry Desprats, Arnaud Oglaza, Pierrick Marie, Samer Machara Marquez**. *A survey on addressing privacy together with quality of context for context management in the Internet of Things. Dans : Annals of Telecommunications,* **Springer**, Numéro spécial *Privacy-aware electronic society,* Vol. 69 N.1, p. 47-62, février 2014.

Conférences et workshops internationaux

— **Arnaud Oglaza, Pascale Zaraté, Romain Laborde**. *KAPUER : A Decision Support System for Protecting Privacy* (regular paper). Dans : *Group Decision and Negotiation (GDN 2014), Toulouse, France, 10/06/2014-13/06/2014*, Pascale Zaraté, Gregory Kersten, Jorge Hernandez, (Eds.), **Springer**, LNBIP 180, p. 100-107, juin 2014.

— **Arnaud Oglaza, Romain Laborde, Pascale Zaraté**. *Authorization policies : Using Decision Support System for context-aware protection of user's private data* (regular paper). Dans : *IEEE International Symposium on UbiSafe Computing, Melbourne (Australia), 16/07/2013-18/07/2013, (Eds.)*, **IEEExplore digital library**, p. 1639-1644, juillet 2013.

Conférences et workshops nationaux

— **Jean-Paul Arcangeli, Amel Bouzeghoub, Valérie Camps, Sophie Chabridon, Denis Conan, Thierry Desprats, Romain Laborde, Emmanuel Lavinal, Sébastien Leriche, Hervé Maurel, Mohamed Mbarki, André Péninou, Chantal Taconet, Pascale Zaraté, Raja Boujbel, Léon Lim, Samer Machara Marquez , Pierrick Marie, Clément Mignard, Arnaud Oglaza, Sam Rottenberg**. *Projet INCOME : INfrastructure de gestion de COntexte Multi-Echelle pour l'Internet des Objets (short paper). Dans : Conférence Francophone sur les Architectures Logicielles (CAL 2014), Paris, 10/06/2014-11/06/2014, (Eds.)*, **ENSEEIHT**, (en ligne), juin 2014.

— **Arnaud Oglaza, Pascale Zaraté, Romain Laborde**. *Système d'aide à la décision pour la protection des données de vie privée de l'utilisateur*. Dans : *ROADEF-15ème congrès annuel de la Société française de recherche opérationnelle et d'aide à la décision*. (2014,Février)

Table des matières

Introduction

« I think it's fair to say that personal computers have become the most empowering tool we've ever created. They're tools of communication, they're tools of creativity, and they can be shaped by their user. »

Bill Gates

Peu d'entre nous peuvent se targuer de pouvoir passer une journée complète sans avoir été en contact avec un quelconque réseau. Nous rentrons dans une ère hyper-connectée. Toutes sortes d'objets du quotidien sont devenus des mini-ordinateurs et peuvent se connecter à Internet. On a pris l'habitude des téléphones, mais aujourd'hui ce sont des objets plus insolites qui se retrouvent connectés comme des ampoules [1], des lunettes [2] ou même des fourchettes [3]. L'extension d'Internet à tous ces objets connectés porte un nom : l'Internet des Objets (*Internet of Things* - IoT).

L'objectif principal de tous ces objets est de nous simplifier la vie, de pouvoir être joignable n'importe où ou encore d'automatiser des tâches de notre vie de tous les jours. Mais la multiplication des objets connectés combinée à l'augmentation des débits a aussi entraîné la prolifération des transferts de données sur les réseaux. Ainsi, il est aujourd'hui possible de trouver tout et n'importe quoi sur Internet. Mais il y a un inconvénient majeur, certaines de ces données nous concernent. Notre nom, nos photos, notre localisation et toutes les données nous concernant peuvent être transférées et se retrouver dans les mains de tout un chacun.

Le respect de la vie privée a depuis toujours concerné le public. Certaines personnes comme les médecins doivent même prêter serment avant de pouvoir exercer afin de garantir la confidentialité de leurs futurs patients. Que ce soit au niveau national ou international, de nombreuses lois encadrent le droit à la vie privée. Mais l'avènement de l'informatique a changé la donne. S'il y a quelques dizaines d'années, s'éloigner de la société permettait de

1. http ://meethue.com/fr-fr/
2. http ://www.google.com/glass
3. http ://www.hapi.com/fr/products-hapifork.asp

s'isoler et de retrouver son intimité, aujourd'hui ce n'est plus le cas. A tout moment, nous pouvons être localisés par notre smartphone, être identifiés sur une photo sur un réseau social, voire même filmés dans la rue par une caméra de surveillance. L'arrivée de certains périphériques connectés a fait scandale. Les Google Glass, lunettes connectées sont sûrement le périphérique ayant le plus défrayé la chronique. L'implantation d'une caméra sur ces lunettes est vue comme un moyen de surveillance et donc un dispositif anti vie privée. Les détracteurs de ces lunettes sont nombreux, une étude menée en 2014 par Toluna [4] montre que 72% des américains sondés refuseraient d'acheter des Google Glass car ils considèrent qu'elles portent atteinte à la vie privée.

Nous devons pouvoir contrôler la divulgation de nos données privées et sensibles. Les problèmes de contrôle d'accès sont d'actualité depuis que l'informatique existe. Que ce soit pour des organisations ou des particuliers, des équipes de chercheurs ont toujours essayé de trouver des méthodes de plus en plus poussées pour protéger efficacement les données plus ou moins sensibles des utilisateurs. Mais avec l'Internet des Objets, c'est un nouveau défi qui est apparu. Comme le souligne L'union Internationale des Télécommunications (*International Telecommunications Union* - ITU) dans son rapport sur l'IoT [37], le respect de la vie privée est crucial pour contrôler cet environnement complexe et en perpétuel mouvement. Les échanges de données sont incessants et se font sans que leurs propriétaires soient au courant. Il faut donc se poser la question de qui au final contrôle les données collectées par tout ce qui nous entoure.

Les méthodes de contrôle d'accès permettent de réguler efficacement la divulgation de données mais pour être mises en place, elles exigent certaines compétences qui ne sont pas accessibles au grand public. Comment faire alors pour permettre à tout le monde de protéger ses données de vie privée sur des appareils connectés ? C'est ce problème en particulier qui nous intéresse : trouver un moyen pour permettre à quiconque d'écrire sa politique d'autorisation pour ses données de vie privée sans avoir besoin de compétences techniques.

L'écriture d'une politique d'autorisation n'est pas quelque chose de facile. Il faut entre autre connaître un langage précis et la structure à utiliser. Il n'est pas envisageable de laisser une personne non experte du domaine écrire sa propre politique d'autorisation sans aide. Mais dans le cadre d'utilisation de l'Internet des Objets, il n'est pas non plus possible d'avoir un expert derrière chaque personne pour aider à cette écriture. Une solution aurait pu être de créer des politiques d'autorisation par défaut et chaque personne aurait pu choisir celle se rapprochant le plus de sa vision de la protection des données de vie privée. Mais chaque individu est unique et a une vision différente de la vie privée et c'est une chose trop importante pour avoir une protection imprécise.

L'évolution de l'informatique, que ce soit au niveau matériel ou logiciel, a permis le développement de nombreux domaines. Parmi eux, celui de l'aide à la décision. Un système d'aide à la décision assiste un utilisateur dans ses prises de décision complexe en apprenant au préalable son comportement. L'utilisation de ce type de système pour aider les propriétaires de périphériques informatiques pouvant contenir des données sensibles est tout à fait pertinente. Les aider à comprendre ce qu'il se passe, les aider à contrôler leurs données pour finalement protéger leur vie privée.

4. https ://us.toluna.com/

L'utilisation d'un système d'aide à la décision permet de comprendre ce que veut l'utilisateur, d'apprendre ses préférences. A partir de ses préférences, le système peut proposer à l'utilisateur des règles permettant de protéger certaines de ses données. L'ensemble des règles acceptées par l'utilisateur forme sa politique d'autorisation. Ainsi l'utilisateur écrit sa politique d'autorisation simplement, sans avoir besoin de connaissances techniques.

Des travaux liés à un projet : INCOME

Cette thèse s'articule autour du projet INCOME (*INfrastructure de gestion de COntexte Multi-Échelle pour l'Internet des Objets*) [5]. L'objectif général d'INCOME est de fournir des solutions logicielles et intergicielles génériques pour la gestion de contexte multi-échelle. INCOME cible le niveau infrastructure pour de nouvelles applications grand public consommatrices d'informations de contexte d'un haut niveau d'abstraction, obtenues après traitement et filtrage de nombreuses informations de contexte directement issues de l'environnement de l'utilisateur ou de l'Internet des objets. Les solutions apportées par le projet faciliteront le développement et le déploiement de ces applications construites au dessus de l'Internet des objets, d'infrastructures ambiantes et mobiles et de nuages informatiques. Au sein de ce projet, la tâche 4 s'occupe de la qualité de contexte et du respect de la vie privée. C'est à l'intérieur de cette tâche que s'intègrent nos travaux. Le but étant de pouvoir protéger la vie privée des utilisateurs et qu'ils puissent contrôler leurs données.

Organisation du document

Les trois premiers chapitres de ce document dressent un état de l'art des différents domaines abordés. Le chapitre 1 concerne la protection de la vie privée, de sa définition aux différentes techniques permettant de l'assurer. Le chapitre 2 concerne la gestion des autorisations. Plusieurs modèles de contrôle d'accès sont détaillés ainsi que les solutions existantes pour mettre en œuvre une politique d'autorisation. Le chapitre 3 concerne l'aide à la décision. Il présente ce vaste domaine et les méthodes existantes. C'est dans ce chapitre que sont introduits les systèmes d'aide à la décision et plus particulièrement les systèmes de recommandation.

Le chapitre 4 présente notre modèle multi-critère mis au point pour la prise en compte des préférences utilisateurs utilisé dans le système Kapuer. Conçu pendant la thèse, Kapuer en est la contribution principale. Le chapitre 5 présente une implémentation de Kapuer. Ce prototype a été réalisé pour un smartphone ou une tablette basée sur le système d'exploitation Android. Finalement, le chapitre 6 présente les évaluations faites de Kapuer. Nous avons effectué plusieurs tests à partir d'un simulateur, les différents résultats obtenus sont analysés dans ce chapitre.

5. www.anr-income.fr

1 Protection de la vie privée

« Vous n'avez déjà plus aucune vie privée - autant l'accepter. »
Scott McNealy, PDG de Sun Microsystems - 1999

TOUTE personne a certains aspects de sa vie qu'elle ne souhaite pas partager avec tout le monde. On parle alors de vie privée. La protection de la vie privée est un droit universel mais n'est pas forcément aisé à mettre en œuvre. Dans ce chapitre, nous allons commencer par définir le concept de vie privée puis nous verrons comment la protection de la vie privée est assurée juridiquement pour finir par voir différentes techniques permettant d'améliorer cette protection de la vie privée en informatique et comment elles pourraient nous aider.

1.1 Définition

Bien que l'on entende beaucoup parler de problèmes liés à la protection de la vie privée depuis quelques années, les premières préoccupations dues à ces problèmes remontent à la fin du 19ème siècle. Le juge américain Thomas Cooley a été le premier à décrire la vie privée comme étant le droit de s'isoler ("*the right to be let alone*") [78]. Déjà à l'époque, les auteurs parlaient "de nombreux appareils mécaniques donnant la possibilité de proclamer sur les toits ce qui se chuchote dans l'intimité", faisant référence par exemple aux premiers appareils photos disponibles pour le grand public et permettant de prendre quiconque en photo sans forcément avoir son consentement.

A l'heure actuelle, les nombreuses avancées technologiques ont modifié cette vision de la vie privée. L'utilisation d'Internet et de tous les appareils connectés à notre disposition ont fait évoluer les interactions sociales permettant très facilement à une personne de prendre contact avec quelqu'un qu'elle n'a jamais vu et qu'elle ne verra peut être jamais. Par exemple, si deux personnes rentrent en contact pour effectuer une transaction, chacune d'entre elles doit obtenir des informations sur l'autre afin d'avoir confiance l'une en l'autre et que la transaction se passe correctement. Il est alors impossible pour une personne de préserver sa vie privée en restant totalement isolée comme c'était encore le cas au 19ème siècle [45].

Une autre définition, datant des années soixante, convient plus à notre vision actuelle de la protection de la vie privée. Alan Westin a défini la vie privée comme "le droit d'une personne, d'un groupe ou d'une institution de déterminer eux mêmes quand, comment et à quelle ampleur des informations les concernant peuvent être communiquées aux autres" [80]. Cette définition se concentre sur les informations caractérisant la personne et non plus sur l'isolement. Selon Westin, si l'isolement peut conduire à la protection des informations personnelles, personne ne veut se couper totalement des autres car le besoin de participer à la société est aussi fort que celui de se protéger des autres. La vie privée est d'ailleurs fortement liée à la société. Selon Moore, "le besoin de vie privée est un besoin créé socialement. Sans société, il n'y aurait pas besoin de vie privée" [52]. La protection de la vie privée porte donc sur le contrôle qu'a une personne sur ses informations personnelles.

Désormais, la conception de la vie privée pour une personne est d'être dans un état de protection et de contrôle sans devoir chercher à atteindre cet état. Une personne ne va pas penser à sa vie privée tant que cette dernière n'est pas menacée. Marx a défini quatre conditions ou frontières qui, si elles sont franchies, peuvent menacer la vie privée d'une personne [50] :

— une frontière naturelle qui empêche la présence, les sentiments ou les émotions d'être perçus par un des cinq sens. Cela peut être des limitations physiques comme une porte, un mur ou des vêtements, mais aussi des expressions du visage pour masquer une émotion.
— une frontière sociale impliquant que des personnes comme un docteur ou un avocat respectent le serment de confidentialité qu'ils ont avec leurs patients ou leurs clients. Cela concerne aussi les secrets entre membres d'une même famille ou de ne pas lire un message dont on est pas le destinataire.
— une frontière temporelle ou spatiale. Les différentes périodes de la vie d'une personne ainsi que ses différents lieux de vie sont censés rester séparés les uns des autres.
— une frontière sur les événements éphémères ou transitoires. Les interactions et les discussions entre plusieurs personnes doivent être éphémères et transitoires et ne doivent pas être enregistrées sans que cela soit précisé.

De son côté, Solove estime qu'une définition de la vie privée n'est pas suffisante et qu'il existe plusieurs formes de vie privée [71]. Il a proposé la taxonomie suivante pour la vie privée incluant une vue d'ensemble des activités pouvant mener à des violations de la vie privée :

— la collecte d'informations. Même si généralement, la collecte d'information se fait avec le consentement de la personne, la collecte d'information cachée ou forcée peut entraîner une surveillance de la personne ou des interrogations sur ses activités amenant une violation de la vie privée.
— l'analyse d'informations. Il peut y avoir violation de vie privée en stockant ou en agrégeant de l'information.
— la dissémination de l'information qui peut amener la rupture de la confidentialité de diverses façons. L'accès à des informations déjà publiques peut être augmenté en la

disséminant à plusieurs endroits. Par exemple un numéro de téléphone peut être une information publique car disponible sur les pages blanches mais au milieu des autres numéros cela ne pose pas de problèmes. Par contre si une personne malveillante se met à diffuser ce numéro un peu partout sur Internet ou dans des lieux publics, le propriétaire du numéro peut être ensuite dérangé régulièrement à cause de cette dissémination.

— l'invasion. S'introduire dans le domicile d'une personne ou l'inonder de courriers est une forme d'invasion violant la vie privée. Interférer dans des décisions personnelles ou de couple est un autre exemple d'invasion violant la vie privée.

Ces quatre groupes d'activités se retrouvent sur la figure 1.1 reprise et traduite de [71].

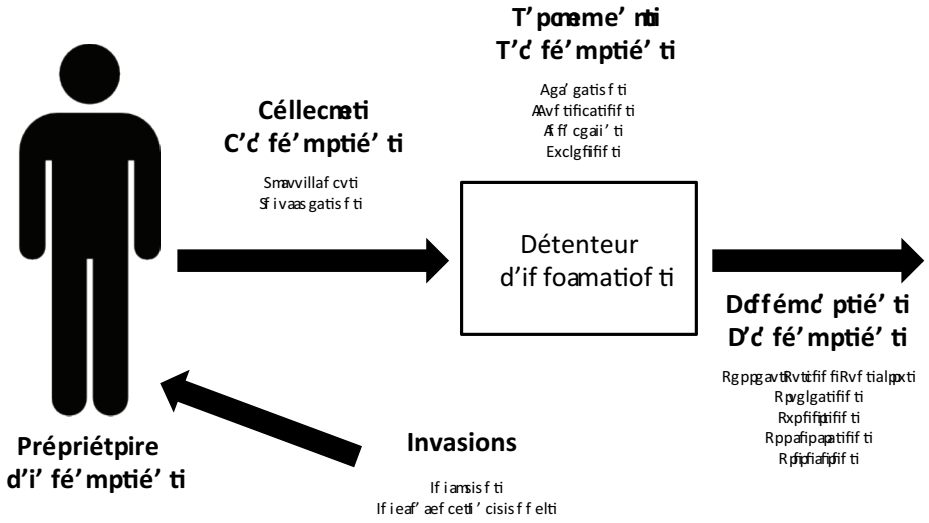


Figure 1.1 — Illustration de la taxonomie de Solove

Il est clair que de nos jours, voir la vie privée comme le droit de s'isoler n'est pas possible. Nous avons besoin d'interagir avec notre société et de divulguer certaines informations. Le tout est d'arriver à limiter cette divulgation à certaines informations et à certaines personnes. Nous cherchons un moyen pour aider un utilisateur à protéger ses données de vie privée. Nous n'allons donc pas traiter tous les groupes d'activité définis par Solove. Notre but est de maitriser la divulgation des données de vie privée, la *collecte d'information* est donc le groupe d'activité qui se rapproche le plus de notre problème.

Nous allons maintenant voir quels sont les moyens pour protéger notre vie privée en commençant par l'aspect juridique.

1.2 La protection de la vie privée d'un point de vue juridique

La notion de droit à la vie privée a donné lieu à de nombreux textes de loi de par le monde. Au niveau international, la vie privée est protégée par l'article 12 de la déclaration universelle des droits de l'homme de 1948 [6] : *Nul ne sera l'objet d'immixtions arbitraires dans sa vie privée, sa famille, son domicile ou sa correspondance, ni d'atteintes à son honneur et à sa*

réputation. Toute personne a droit à la protection de la loi contre de telles immixtions ou de telles atteintes.

Le droit à la vie privée est aussi garanti au niveau européen par l'article 8 de la convention européenne des droits de l'homme [21] : *Toute personne a droit au respect de sa vie privée et familiale, de son domicile et de sa correspondance.*

Au niveau de la France, l'article 8 de la déclaration universelle des droits de l'homme est utilisé pour le droit à la protection de la vie privée. Le code civil français rajoute un texte capital dans l'article 9 : *Chacun a droit au respect de sa vie privée*. De nombreuses informations sont protégées comme par exemple l'état de santé d'une personne qui est couvert par le secret médical. Ces informations protégées se retrouvent dans les catégories suivantes :

— **les coordonnées personnelles** : adresse postale, adresse électronique, numéro de téléphone, etc.
— **la situation patrimoniale et financière** : patrimoine immobilier, revenus perçus, impôts et taxe, etc.
— **la formation** : formation initiale, inscription dans un établissement d'enseignement, diplômes, etc.
— **les numéros d'immatriculation** : INSEE, police d'assurance, immatriculation d'un véhicule particulier, etc.
— **l'appartenance politique ou religieuse** : appartenance politique, usage du pouvoir de suffrage, croyances religieuses, etc.
— **la situation professionnelle** : dates de congés, congés de maternité ou parentale, horaires de travail, etc.
— **la situation médicale** : l'ensemble des informations concernant la personne venues à la connaissance d'un professionnel de santé.

Pour l'état civil d'une personne, alors que la date de naissance, l'âge ou le lieu de naissance par exemple sont des informations protégées, le nom et le prénom d'une personne ne le sont pas. Le décès ne met pas un terme à la vie privée d'une personne mais peut permettre aux proches d'avoir accès à certains documents utiles. Au niveau des archives publiques, des délais sont prévus au delà desquels les documents deviennent librement communicables.

Plusieurs commissions sont chargées de veiller au respect de la vie privée. Parmi elle, la CNIL [1] (Commission Nationale de l'Informatique et des Libertés), est chargée de veiller entre autre à ce que les systèmes informatiques ne portent pas atteinte à la vie privée des citoyens. Une autre commission, la CADA [2] (Commission d'Accès aux Documents Administratifs) a pour objectif de faciliter et de contrôler l'accès des particuliers aux documents administratifs.

Malgré toutes les lois existantes, de nombreuses entreprises cherchent à collecter un maximum de données personnelles sur leurs clients dans le but de mieux les cibler et de faire des profits. L'explosion d'Internet a permis de fonder un nouveau modèle économique basé sur les données personnelles. Leur vente est devenue un nouveau marché représentant

1. www.cnil.fr
2. www.cada.fr

plusieurs milliards d'euros par an et pourrait atteindre 1000 milliards d'euros en 2020 selon une étude du *Boston Consulting Group* [31]. Il est nécessaire qu'un utilisateur de périphériques connectés à Internet possède des outils pour protéger ses informations. Nous allons présenter dans la prochaine section les technologies permettant cette protection.

1.3 Technologies de protection de la vie privée

Nous étudions dans cette section les technologies de protection de la vie privée (*Privacy Enhancing Technology* - PET). Ce concept de PET a été proposé par Cavoukian [16] pour unifier les nouvelles technologies et la protection de la vie privée. De nombreux travaux ont suivi et le concept de PET fait maintenant entièrement partie du vocabulaire lié à la vie privée.

Récemment, une nouvelle approche a été présentée pour tenter de donner plus de contrôle à l'utilisateur : la protection intégrée de la vie privée (Privacy by Design) [17]. Elle repose sur le fait de penser à la protection de la vie privée au moment même de la conception d'un système et non pas de s'en occuper seulement une fois que le système s'exécute. Cette approche part aussi du principe que les lois protégeant la vie privée d'une personne ne sont pas suffisantes pour garantir cette protection. La protection intégrée de la vie privée repose sur sept principes fondamentaux :

— prévoir et prévenir les risques d'incidents liés à la vie privée avant qu'ils ne se produisent.
— protéger l'utilisateur sans qu'il n'ait à s'en soucier ou à agir dans ce sens, la protection de la vie privée est assurée par défaut et non par choix.
— intégrer la protection de la vie privée à la conception et à l'architecture des systèmes informatiques.
— assurer aussi bien la protection de la vie privée et la sécurité. L'un ne doit pas empêcher l'autre.
— protéger les données de vie privée pendant toute leur durée de vie : de leur création à leur destruction.
— garder un fonctionnement transparent aux yeux de l'utilisateur, permettre à tout le monde de voir que le système fait ce qu'il est censé faire afin que l'utilisateur ait confiance dans l'utilisation du système.
— privilégier les intérêts de l'utilisateur et respecter sa vie privée.

Mais ces principes sont rarement appliqués, ce qui est à la base de notre problème. Si les développeurs des systèmes d'exploitation de périphériques mobiles avaient pensé dès la conception aux risques liés à la divulgation des différentes données présentes, le problème de la protection de la vie privée sur ces périphériques serait moins présent. Mais le manque de patrons ou de modèles permettant de mettre en place ces principes à la conception rend la tâche difficile. Pour mieux présenter les diverses solutions amenés par les PET, cette section est organisée en trois parties pour donner trois points de vue différents : la protection de la vie privée en tant que confidentialité des données, en tant que contrôle des données et finalement en tant que transparence autour des données [26].

1.3.1 D'un point de vue confidentialité des données

La confidentialité des données est présente dans certaines technologies de protection de vie privée sous diverses formes. Ces technologies considèrent que l'objectif principal en terme de vie privée est de protéger les données personnelles en évitant leur accès par des personnes non autorisées. Si des données personnelles deviennent publiques, la confidentialité et par conséquent la vie privée est perdue. Trois méthodes permettent de résoudre les problèmes de vie privée via la confidentialité des données. Il s'agit de l'anonymisation des données, de l'anonymisation dans les communications et de la minimisation des collectes de données.

L'anonymisation des données consiste à utiliser des méthodes de cryptographie pour que les données obtiennent certaines des propriétés suivantes définies par [60] :

— des données sont intraçables quand un observateur ne peut pas relier deux données ou deux actions appartenant au même utilisateur.
— une donnée est indétectable quand un observateur ne peut savoir si elle existe ou non.
— une donnée est inobservable quand il est impossible pour un observateur de la discerner parmi un ensemble de données.

L'anonymisation des données est utilisée dans de nombreuses approches comme le k-anonymat [75]. Prenons par exemple une liste de données personnelles où pour chaque entrée, on retrouve l'âge, le sexe et l'adresse de la personne. Cette liste satisfait au k-anonymat si pour chaque entrée, il existe au moins k autres entrées partageant les mêmes valeurs. Pour cela il est possible de grouper certaines données afin d'obtenir des entrées ayant les mêmes valeurs comme par exemple regrouper les âges en dizaines où ne communiquer que la ville de domiciliation au lieu de l'adresse complète. Il est ainsi impossible pour un observateur de différencier les k entrées similaires et l'anonymat est respecté. Cette technique est donc un compromis entre l'anonymat et la précision des informations. La figure 1.2 montre un exemple de k-anonymat. Dans cette exemple, le tableau de gauche présente les données complètes et le tableau de droite après avoir utilisé la technique du k-anonymat. Le code postal, l'âge et la nationalité de chaque personne a subi des modifications pour améliorer la confidentialité des données. Les "*" correspondent à des données supprimées. Dans ce cas là et pour les trois types d'informations modifiées, chaque entrée est similaire à trois autres entrées. On se trouve donc dans un cas de 4-anonymat.

Des variantes du k-anonymat comme la l-diversité ont été proposées pour renforcer encore plus l'anonymisation des données [48]. La l-diversité est plus avancée que le k-anonymat car elle garantit en plus que dans un groupe de k individus, il y aura au moins l valeurs sensibles distinctes. Si on reprend l'exemple de la figure 1.2 en ne gardant que les personnes numérotés de 9 à 12. Les informations sont 4-anonymes et 1-diverse. Donc si on sait qu'une personne est présente dans le listing, on connaîtra sa maladie. Donc bien que le niveau de k-anonymat soit important, si la l-diversité est égale à un, l'anonymat n'est pas garanti.

L'anonymisation traite aussi des communications. Anonymiser les communications re-

	Peu sensible			Sensible
	Codo pottal	Agg	Natigtialaté	Malatiag
1	33400	28	Frç4çç0	P4rumr4m
P	PPPuu	P6	P46lmm	P46um64m6
P	PP4uu	39	P9m4nmmm	Dèèbèu9
4	DD5uu	D5	D9è5èèèu	Dèèbèu9
5	D5DDu	55	Ruuu9	HéHè5u9
9	H5uuu	5H	H9è5èèèu	H59uè 95è9
7	757uu	5u	79è5èèèu	G9%GG9
9	G5uuu	59	G9è5èèèu	G9%GG9
9	G5uGu	GG	EuEè959l	Eè5c99
Ec	E5cEE	E9	E9è5èèèc	Eè5c99
EE	E5cEE	EE	E9è5èèèc	Eè5c99
EE	E5cE5	E9	Icèlè95	l è5c99

	Peu sensible			Sensible
	Codo poatalt	Aggt	Natignalitét	Malatiigt
1t	33***t	<<0t	0t	<ng0mgnigt
2t	22000t	220t	0t	2ng0mgnigt
2t	22000t	220t	0t	Dièéètgt
4t	44000t	440t	0t	4ièéètgt
5t	55000t	>>0t	0t	HéHè0tgt
6t	66000t	660t	0t	6ng0è gnigt
7t	77000t	770t	0t	GriGGrt
8t	88000t	880t	0t	8ri88rt
9t	99090t	90t	0t	9èncrrc
90c	99090c	90c	0c	9èncrrc
99c	99090c	90c	0c	9èncrrc
99c	99090c	90c	0c	9èncrrc

Figure 1.2 — Exemple de l'utilisation de la technique du k-anonymat

vient à protéger les flux de données de façon à ne pas savoir qui parle à qui. Même si le contenu d'une conversation est tenu confidentiel, des informations sensibles peuvent être interceptées par les flux de données. Ces informations peuvent concerner les localisations et les identités des deux parties mais aussi les heures, la fréquence ou le volume des conversations. Fournir des communications anonymes n'est pas aisé étant donné que beaucoup de protocoles de communications utilisent des identifiants uniques [51]. Malgré tout des solutions existent. Mix-network [18] est un protocole de routage utilisant des serveurs intermédiaires pour router des messages de façons aléatoires. A chaque fois qu'un routeur reçoit des messages, ils mélangent leur ordre et les renvoient de façon aléatoire à d'autres serveurs. Chaque serveur ne sait pas d'où vient initialement le message ni le destinataire. Il est ainsi plus compliqué de tracer le trajet d'un message d'un bout à l'autre. L'application utilisant le protocole mix-network la plus connue est Tor (The Onion Router)[29].

Le but de la minimisation des données est de limiter la collecte et le traitement des données personnelles. Cela peut être effectué en utilisant une architecture distribuée et une agrégation des informations avant l'envoi sur un serveur central. Un système centralisé peut en effet poser certains problèmes. Ce système récupère toutes les données de ses utilisateurs, y compris les données relatives à la vie privée des utilisateurs. Une fois sur le serveur central, l'utilisateur n'a plus de contrôle sur ses données qui peuvent être exploitées à des fins commerciales sans qu'il soit au courant. A l'inverse, si les données de l'utilisateur sont stockées en local, l'utilisateur garde un certain contrôle. Par exemple, [11] propose une approche décentralisée pour son système de recommandation. Ce type de système recommande des objets à l'utilisateur en fonction de ses préférences et des utilisateurs ayant un profil similaire. L'utilisation d'un système décentralisé évite le regroupement d'information sur un serveur central mais pour trouver les utilisateurs avec un profil similaire, le système a tout de même besoin d'accéder aux autres profils via un système de pair à pair. Pour éviter d'avoir accès aux profils exacts des autres utilisateurs et ainsi mettre en péril leur vie privée, le système utilise un système d'obfuscation pour cacher le profil exact des utilisateurs. Plus le profil est obfusqué moins les recommandations seront précises, il faut donc trouver le bon compromis pour garantir la protection de la vie privée d'un utilisateur tout en gardant un niveau de recommandation acceptable. D'autres techniques pourraient permettre de conserver la

confidentialité des données stockées sur un serveur distant. Par exemple, le chiffrement homomorphique [56] permet de crypter des données et de continuer à effectuer des traitements sur ces données sans avoir à les déchiffrer. Mais pour l'instant, seules des opérations basiques comme l'addition ou la multiplication sont possibles.

Les techniques apportant de la confidentialité sont intéressantes de notre point de vue. Garder et traiter les données en local évite une dissémination des informations et que les données soient stockées dans des bases de données sans le savoir. De la même façon, l'anonymisation des données peut permettre de ne pas remonter jusqu'à l'identité de leur propriétaire et ainsi garantir une protection. Mais ces techniques doivent être mise en place dès la conception des applications nécessitant l'information. Si un concepteur a développé son application de telle sorte que les informations qu'elle récupère soient envoyées sur un serveur centralisé et traitées sur ce serveur, il n'est pas possible de contourner ce processus et d'effectuer le traitement en local. Anonymiser les données ou minimiser les collectes ne sont donc pas toujours des techniques appropriées. Si une personne doit fournir des données personnelles pour faire fonctionner un service, la protection de ses données peut s'effectuer en contrôlant par exemple à qui l'on divulgue, quand ou comment. Nous présentons dans la prochaine section quelles sont les techniques et méthodes permettant d'acquérir un contrôle sur la divulgation des données.

1.3.2 D'un point de vue contrôle des données

Présenter la vie privée d'un point de vue contrôle des données correspond à présenter les possibilités pour une personne de contrôler ce qui advient avec ses données personnelles et de prévenir les fuites. Il n'est pas possible de s'isoler totalement des autres et la divulgation de certaines informations est nécessaire. Si une personne contrôle ce qu'elle divulgue et ce qu'elle garde pour elle, sa vie privée sera protégée. Cela rejoint la définition de la vie privée de Westin donnée plus haut. En effet, nous allons présenter dans cette section divers mécanismes permettant le contrôle des données personnelles.

Les modèles de contrôle d'accès ont été conçus pour formaliser la façon d'écrire des règles d'autorisation. Nous présenterons de nombreux modèles dans le chapitre suivant mais nous verrons que plus leur pouvoir d'expression augmente, plus la difficulté d'écrire ces politiques augmente aussi. Ce qui n'est pas compatible avec une utilisation grand public. Des langages spécifiques ont été développés pour écrire les politiques d'autorisation. Par exemple, EPAL (*Enterprise Privacy Authorization Language*) est un langage formel conçu pour définir des politiques internes pour les entreprises [5]. Cette entreprise après avoir défini un vocabulaire spécifique à EPAL peut écrire sa politique EPAL selon ses besoins. Bien que cela permette de définir des politiques riches, la granularité d'écriture des règles n'est pas assez fine. XACML [3] est un autre langage permettant d'écrire des politiques d'autorisation. XACML est plus puissant qu'EPAL pour exprimer des politiques de contrôle d'accès mais aussi pour exprimer des politiques de protection de vie privée [77]. Pour rajouter du contrôle sur les données, en plus de gérer les autorisations, il est possible de gérer des obligations pour décrire comment les données doivent être utilisées une fois l'accès autorisé. Une solution pour être sûr que les obligations soient respectées est d'utiliser des politiques

3. https ://www.oasis-open.org/committees/xacml/

collantes (*sticky policies*). Un mécanisme de chiffrement qui supporte qu'une politique soit attachée aux données permet de chiffrer les données selon les détails de la politique. L'accès aux données est ensuite accordé par une autorité de contrôle qui vérifie que la politique est bien respectée pour donner la clé de déchiffrement [59]. Par exemple, le projet PRIMElife utilise l'attachement de politique pour distribuer des politiques de protection de vie privée [27].

Une façon, pour une personne, de contrôler ce qu'elle va partager avec les autres, est d'utiliser un système de gestion d'identité (*Identity Management* - IM). En utilisant un tel système, une personne peut se créer plusieurs identités en se décrivant à partir d'attributs qu'elle accepte de partager. Chaque identité peut ensuite être utilisée en fonction des besoins. Ainsi il est possible de créer une identité révélant plus d'informations à destination d'un cercle d'amis ou de la famille et une autre plus privée destinée au travail. Le système d'IM permet à la personne de s'identifier sur d'autres systèmes en utilisant une identité. Les systèmes d'IM récents séparent deux entités : le fournisseur de service qui va procurer un service à un utilisateur en utilisant des informations le concernant et le fournisseur d'identité qui identifie l'utilisateur et stocke les données correspondant à l'identité de l'utilisateur. De nombreux systèmes utilisent ce mécanisme comme Shibboleth [53] qui autorise les administrateurs du fournisseur d'identité à choisir quels attributs peuvent être partagés selon le fournisseur de services. D'autres systèmes d'IM sont plus connus et utilisés par le grand public comme Facebook Connect (figure 1.3). Ce service permet de se connecter à un nombre très important de sites via son profil Facebook et donc de ne pas avoir à créer un compte sur ces sites. Bien que ce service soit présenté comme un système de gestion d'identité, il ne permet pas à l'utilisateur de contrôler quelles données seront partager avec les autres sites. Pire, il donne à Facebook des informations sur les préférences de l'utilisateur et les sites auxquels il se connecte, informations qui sont ensuite soit utilisées directement par Facebook soit revendues à des annonceurs qui achètent des publicités pour le site [4].

Bien que les systèmes d'IM donnent l'impression de donner le contrôle à l'utilisateur, bien souvent il n'en est rien. Si du côté des fournisseurs d'identités, il est possible de restreindre les données partagées et stockées, ce n'est pas aussi évident pour le fournisseur de services. Ce dernier demande un certain nombre d'informations pour authentifier un utilisateur, si l'utilisateur ne veut pas les donner, il ne pourra pas être authentifié. Il n'y a pas de phase de négociations pour atteindre un niveau satisfaisant les deux parties, c'est tout ou rien. Le contrôle se trouve donc du côté du fournisseur de service.

Dans un environnement distribué et mobile, chaque utilisateur ne peut être accompagné d'un expert en sécurité pour l'aider à protéger ses données de vie privée. Comme nous l'avons déjà vu, il n'est plus possible de rester isolé et d'avoir une politique de non divulgation totale. Pour avoir une vie normale avec notre société, tout un chacun doit partager certaines informations selon ce qu'il désire faire. L'utilisateur a besoin de réguler les informations qu'ils communiquent et ne pas divulguer plus d'informations que nécessaire. Avoir le contrôle sur ses données est une solution évidente permettant de gérer quand et comment les données peuvent être divulguées. Pour autant, cela implique que l'utilisateur est conscient de ce qu'il doit faire pour avoir une réelle protection de la vie privée. Il doit

4. https ://www.facebook.com/about/privacy/your-info#howweuse

Figure 1.3 — Identification via Facebook Connect

donc comprendre pourquoi une personne ou une application veut avoir accès à ses données, dans quel contexte et dans quel but. Avoir juste le contrôle de ses données n'est donc pas suffisant, c'est pourquoi nous allons dans la prochaine section présenter un autre point de vue de la protection des données, la transparence autour des données.

1.3.3 D'un point de vue transparence autour des données

Les outils amenant de la transparence autour des données liées à la vie privée permettent de faire comprendre aux utilisateurs quelles données sont divulguées et comment ces données sont utilisées. Selon Castellucia et al. [15], pour qu'un outil puisse améliorer la transparence des données, il doit posséder aux moins une des quatre caractéristiques suivantes :

1. fournir des informations sur la façon dont les données de l'utilisateur sont collectées, stockées ou analysées.
2. fournir un récapitulatif de quelles données ont été divulguées, à qui elles ont été divulguées et sous quelles conditions.
3. fournir un accès en ligne aux données personnelles et aux informations acquises grâce à leurs traitements et fournir un moyen de savoir si ces traitements respectent les lois sur le respect de la vie privée et les accords de divulgation.
4. fournir un moyen d'éviter d'établir un profil de l'utilisateur, l'aider à savoir comment les données divulguées peuvent le faire rentrer dans une catégorie de profil afin de réduire les risques dans le futur.

La transparence des données est quelque chose de très important. Comprendre ce qu'il advient de ses données est primordial pour arriver à les contrôler correctement. Les tech-

nologies de protection de la vie privée sont inutiles si leurs utilisateurs ne peuvent pas s'en servir efficacement. Les évolutions récentes de l'informatique rendent le besoin de transparence évident pour améliorer la protection de la vie privée qui est encore plus critique. Grâce à l'Internet des objets, considéré comme le futur de notre société électronique ; Internet ne connectera pas seulement les personnes entre elles mais aussi les machines et tous les objets dits intelligent. Le paradigme de communication "N'importe où, n'importe comment, n'importe quand" sera étendu avec l'IoT à "N'importe quoi, n'importe qui, n'importe quel service". Du coup, en plus de devoir contrôler les données personnelles propagées à partir des périphériques utilisés directement, il va falloir aussi contrôler les données produites automatiquement par les objets connectés que l'on possède, ceux qui nous entourent et qui sont présents dans notre environnement quotidien. Ces données issues de l'IoT peuvent être dispersées à travers un immense système distribué en faisant face à des problèmes comme l'hétérogénéité ou le passage à l'échelle. Malgré l'importance de ces problèmes, peu de personnes ont travaillé dessus. Par exemple, Castellucia et al. [15] ont indiqué que pour l'instant, aucun outil existant ne prend en compte la caractéristique 4. alors que son utilité à été démontrée dans le projet FIDIS [35].

De nombreux travaux de recherche ont été menés pour simplifier les interactions entre les utilisateurs et leurs périphériques électroniques pour la sécurité. Par exemple le projet P3P (*Platform for Privacy Preferences*) [23] a défini un standard pour simplifier les politiques de confidentialité des données personnelles sur les sites web afin de permettre aux utilisateurs de mieux comprendre comment les sites web utilisent leurs données. Ces politiques sont ensuite évaluées en fonction des préférences de l'utilisateur par des mécanismes ad hoc. Poursuivant le même but, Inglesant et al. [36] ont proposé un langage naturel contraint pour la spécification de politique d'autorisation. Stepien et al. [73] ont eux travaillé sur des notations non-techniques pour les politiques XACML.

Des approches tentent d'impliquer l'utilisateur dans la gestion de la vie privée. Lederer et al. [46] proposent d'améliorer la compréhension de l'utilisateur sur les implications de la vie privée en lui fournissant des informations en retour. Les miroirs de vie privée [57] permettent aux utilisateurs de mettre en place leur politique de divulgation vis à vis de leurs données personnelles pour ensuite leur montrer comment ces données peuvent être vues par d'autres personnes. Finalement, le projet PrimeLife [79] a publié une analyse donnant un moyen de construire des interactions homme-machine dans le but d'améliorer la protection de la vie privée. Ils ont par exemple défini une liste de 25 questions que le concepteur de l'interface d'un PET doit vérifier pour être sûr de protéger la vie privée de l'utilisateur. Parmi ces questions, il est demandé si les paramètres par défaut favorise la protection de la vie privée, si les utilisateurs peuvent contrôler quelles données peuvent être divulguées ou encore si suffisamment d'informations sont données pour permettre à l'utilisateur de prendre des décision informées.

Rendre un système transparent en terme de gestion des données de vie privée est essentiel pour leur protection. Comment un utilisateur peut-il se protéger correctement s'il n'est pas au courant de ce qu'il advient de ses données ? Pour lui permettre de prendre les bonnes décisions, il faut l'informer aussi bien de ce qu'il se passe sur le système que des risques qu'il peut prendre en faisant telle ou telle action. Ainsi le risque de se tromper est minimisé.

1.4 Conclusion

Ce chapitre a introduit les notions de vie privée et de protection des données de vie privée. Plusieurs aspects sont importants pour avoir une bonne protection. Il faut ainsi tenir compte de la confidentialité des données de la personne, du contrôle qu'elle peut avoir sur ses données et de la transparence à avoir autour de ses données afin qu'elle ait toutes les clés pour comprendre les enjeux et les risques à les divulguer.

Tout cela va nous aider à mieux cerner les enjeux du prochain chapitre destiné à la gestion des autorisations.

2 Gestion des autorisations

« I'm still a hacker. I get paid for it now. I never received any monetary gain from the hacking I did before. The main difference in what I do now compared to what I did then is that I now do it with authorization. »

Kevin Mitnick

NOUS avons vu dans le chapitre précédent qu'avec les pratiques actuelles de l'informatique, un utilisateur avait besoin de divulguer certaines de ses données pour pouvoir utiliser pleinement les services qui s'offrent à lui. Parmi ces données, certaines sont considérées comme personnelles et leur divulgation peut porter atteinte à la vie privée de leur propriétaire. Les outils permettant à l'utilisateur de garantir la protection de sa vie privée doivent fournir de la transparence sur la façon dont les données ont été divulguées, qui les a collectées ou comment elles seront analysées. En plus de cette transparence sur les données, ces outils doivent surtout permettre à l'utilisateur d'avoir le contrôle sur ses données et de pouvoir choisir qui peut y avoir accès, quand et comment. Il est donc question de gestion d'autorisation sur les accès aux données de vie privée.

La gestion des autorisations correspond à spécifier et mettre en œuvre les droits d'accès à des ressources. Autrement dit, "autoriser" revient à définir une politique d'accès. Afin d'exprimer des politiques de contrôle d'accès, de nombreux modèles de contrôle d'accès existent. Nous allons présenter dans la prochaine section certains de ces modèles.

2.1 Modèles de contrôle d'accès pour exprimer une politique

Les modèles de contrôle d'accès ont été conçus pour formaliser l'écriture des règles constituant une politique de contrôle d'accès. Bien qu'il existe de nombreux modèles ayant chacun leurs caractéristiques, il y a une base commune. Tout ces modèles considèrent trois ensembles :

— **l'ensemble des objets O** représentant les ressources ou les services à contrôler.
— **l'ensemble des sujets S** représentant les entités qui veulent exécuter des actions sur les objets. Ces entités peuvent représenter des utilisateurs ou des applications.

— **l'ensemble des droits d'accès R** représentant comment les sujets peuvent accéder aux ressources.

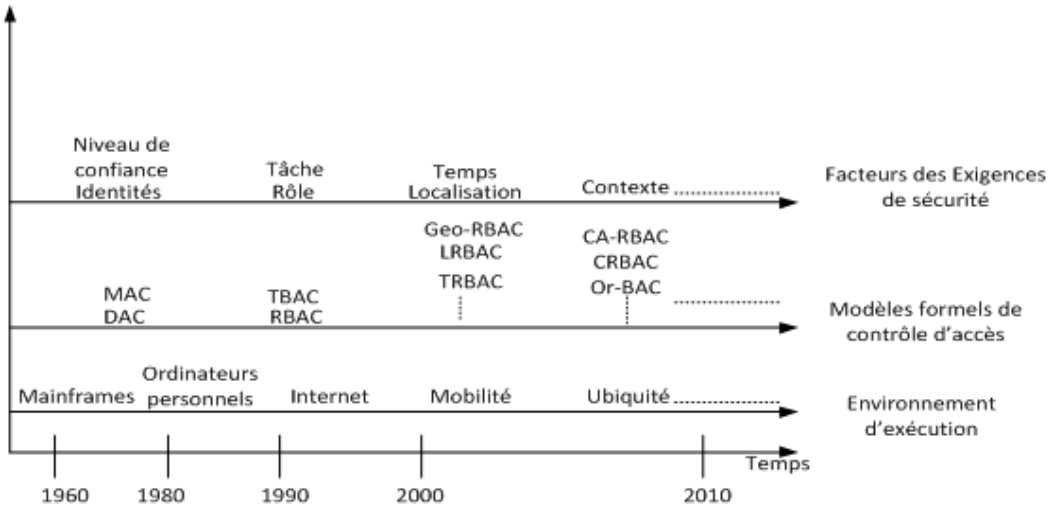


Figure 2.1 — Evolution des modèles de contrôle d'accès tiré de [19]

Écrire une politique de contrôle d'accès revient donc à écrire un ensemble de relations sur O x S x R. Les modèles de contrôle d'accès ont évolué avec le temps pour se conformer à l'évolution des exigences de sécurité et à l'évolution de l'environnement d'exécution. La figure 2.1 tiré de [19] illustre ces évolutions. Nous allons maintenant présenter plusieurs de ces modèles : DAC qui permet à une personne de définir directement qui peut faire quoi sur quel objet, MAC utilise un administrateur pour définir des politiques multi-niveaux, RBAC propose la notion de rôle pour abstraire les sujets selon leur fonction, OrBAC qui centre les politiques d'autorisation sur l'organisation et enfin ABAC, un modèle assez générique utilisant des attributs pour décrire les différentes entités des politiques.

2.1.1 Modèles de contrôle d'accès basé sur les identités

Les modèles de contrôle basé sur les identités (*Identity Based Access Control* - IBAC) limitent l'accès aux objets en se basant généralement sur l'identité de l'utilisateur. Un de ces modèles, le modèle d'accès discrétionnaire (*Discretionary Access Control* - DAC) est présenté par le TCSEC (*Trusted Computer System Evaluation Criteria*) comme "un moyen de limiter l'accès aux objets basés sur l'identité des sujets ou des groupes auxquels ils appartiennent Les commandes sont discrétionnaires car un sujet avec une certaine autorisation d'accès est capable de transmettre cette permission à n'importe quel autre sujet" [39].

La description des règles d'autorisation est basée sur le modèle des matrices d'accès introduit par Lampson en 1971 [44]. On retrouve dans ce modèle le triplet <S,O,R> avec S qui représente l'ensemble des sujets (utilisateurs, applications, processus, etc.), O qui représente l'ensemble des objets (fichiers, table, programme, etc.) et R qui représente les actions des sujets sur les objets (lecture, écriture, exécution, etc.). La figure 2.2 présente un exemple de matrice d'accès.

	manuscrit.txt	petup.exe
JohhJ	lirol	oxécutorl
PiorroP	ProP écriroP	

Figure 2.2 — Exemple de matrice d'accès

Chaque sujet est disposé sur une ligne et chaque objet sur une colonne alors que les actions sont placées dans les cases. Ainsi chaque case permet de définir les actions autorisées par un sujet sur un objet. Dans notre exemple, John et Pierre ont deux autorisations chacun. John peut lire le fichier "manuscrit.txt" et exécuter "setup.exe" et Pierre peut lire et écrire le fichier "manuscrit.txt".

DAC est donc un modèle intéressant car c'est à l'utilisateur de mettre en place sa politique d'autorisation. Dans notre cas, nous ne pouvons pas avoir une personne derrière chaque utilisateur pour écrire la politique d'autorisation. Par contre le fait que l'utilisateur doit écrire des règles pour chaque objet peut rendre le travail très long et pénible sur certains systèmes.

2.1.2 Modèles de contrôle d'accès multi-niveaux

Le problème des modèles DAC est le manque d'abstraction obligeant l'utilisateur à écrire des règles pour chaque objet. Pour amener de l'abstraction et faciliter l'écriture des politiques d'autorisation, des modèles de contrôle d'accès multi-niveaux ont été développés. Parmi eux, les modèles de contrôle d'accès obligatoire (*Mandatory Access Control* - MAC) qui ne laissent plus le contrôle à l'utilisateur. C'est un administrateur qui est chargé d'écrire les politiques d'autorisation. Plusieurs modèles ont été définis, chacun basé sur des propriétés différentes. Par exemple, le modèle de Bell-LaPadula [9] se concentre sur la confidentialité, d'autres sur l'intégrité comme le modèle de Biba [10]. Nous allons présenter plus en détail le modèle Bell-LaPadula qui a été le premier à utiliser cette approche.

Le modèle Bell-LaPadula [9] est un modèle développé pour le département de la défense américaine dans le but de garantir la confidentialité des données. C'est un modèle multi-niveaux basé sur la classification des sujets et des objets. Les sujets et les objets sont classés selon différents niveaux de confidentialité appelés niveaux d'habilitation pour les sujets et niveaux de classification pour les objets. Les classes d'accès de ce modèle sont modélisées par des treillis. Chaque niveau est composé de deux attributs :

— le premier, cl représente un élément parmi une classification ordonnée tel que Top Secret (TS) > Secret (S).
— le second, C représente des catégories décrivant le type d'information. Par exemple défense, nucléaire

L'ensemble des niveaux constitue le treillis et ces niveaux sont reliés par une relation de dominance. Ainsi un niveau $x = (cl, C)$ domine un niveau $y = (cl', C')$ si $cl \geq cl'$ et $C \subseteq C'$.

La figure 2.3 illustre un exemple de treillis. Cet exemple est construit à partir des catégories Défense et Nucléaire et de la classification Top Secret (TS) > Secret (S).

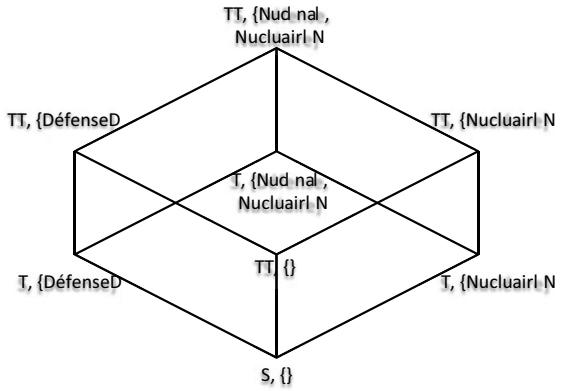


Figure 2.3 — Exemple de treillis

Afin de garantir la confidentialité, deux propriétés doivent être respectées :

— **no read up** un sujet est autorisé à lire un objet seulement si son niveau domine le niveau de l'objet. Un personne avec l'habilitation Secret ne pourra donc pas avoir accès à un objet classé Top Secret.

— **no write down** un sujet est autorisé à modifier un objet seulement si le niveau de l'objet domine son niveau. Une personne avec l'habilitation Top Secret ne peut pas modifier un objet classé Secret. Cela évite que des informations d'un niveau supérieur puisse être accessible à des personnes d'un niveau inférieur.

En utilisant une approche multi-niveaux et en laissant un administrateur gérer les politiques de contrôle d'accès, le modèle Bell-LaPadula résout les problèmes de fuite d'informations des modèles DAC. Néanmoins il présente plusieurs inconvénients. Étant donné qu'un sujet ne peut pas modifier un objet d'un niveau inférieur, cet objet doit être surclassé au niveau du sujet pour pouvoir être modifié. Mais au fil du temps, cela conduit à une surclassification des informations qui deviennent inaccessibles aux sujets de faible niveau. De plus ce modèle est très rigide, et ne permet pas de gérer une exception. Ainsi un sujet d'un niveau ne peut pas accéder exceptionnellement à un objet d'un niveau supérieur. Ainsi, les niveaux permettent d'appréhender plus facilement les règles de contrôle d'accès.

2.1.3 Modèle de contrôle d'accès à base de rôle

Le modèle de contrôle d'accès à base de rôle (*Role Based Access Control* - RBAC) propose un nouveau concept, le rôle, pour structurer les organisations. L'idée vient d'un constat simple : les droits d'accès accordés aux sujets sont souvent liés à leurs rôles dans l'organisation. Par exemple à l'université, les étudiants ont accès seulement aux salles en accès libre alors que les enseignants ont en plus accès à leurs salles de cours et le personnel d'entretien a accès à toutes les salles. Le rôle est donc un moyen d'abstraire une fonction dans une organisation (dans notre exemple il y a trois fonctions : étudiant, enseignant, personnel d'entretien). RBAC est apparu lorsque le nombre d'ordinateurs personnels et donc le nombre d'utilisateurs est devenu trop important pour les gérer un à un. La notion de rôle

permet de créer des catégories dans lesquelles l'administrateur va placer les utilisateurs et ainsi gérer non plus les utilisateurs mais ces catégories. Le groupe ANSI a standardisé le modèle RBAC [65]. On y retrouve une formalisation comprenant les ensembles et relations suivantes :

— l'ensemble des utilisateurs U représentant des personnes ou des processus.
— l'ensemble des rôles R représentant les fonctions d'une organisation.
— l'ensemble des permissions P représentant les accès aux ressources du système.
— l'ensemble des sessions S associant chaque utilisateur avec ses rôles.
— la relation $UA \subseteq U$ x R qui permet d'associer des rôles à un utilisateur.
— la relation $PA \subseteq P$ x R qui permet d'associer des permissions à un rôle.
— la relation $RH \subseteq R$ x R qui permet de créer une hiérarchie de rôle. Cette hiérarchie est construite selon un ordre partiel donc soit r^1 et $r^2 \in R$, si $r^1 \geq r^2$ alors les permissions de r^2 sont aussi des permissions de r^1.
— $S_u = S \rightarrow U$ permet d'établir l'utilisateur d'une session
— $S_r(s) \subseteq \{r \in R | (S_u(s), r) \in UA\}$ permet d'établir l'ensemble des rôles associés à une session s.

Les modèles RBAC peuvent être très riches et complexes mais aussi très simples. Pour permettre de modéliser plus facilement les politiques, quatre sous-modèles ont été définis :

— RBAC_0 aussi appelé *Flat RBAC* reprend les principes essentiels de RBAC. Les utilisateurs sont assignés à des rôles et les permissions sont assignées à des rôles. Chaque utilisateur obtient les permissions associées à son ou ses rôles.
— RBAC_1 ou *Hierarchical RBAC* reprend RBAC_0 et rajoute les hiérarchies de rôles. Chaque rôle peut alors avoir un rôle père et un ou plusieurs rôles fils. Cela permet de rajouter de l'héritage entre les rôles et donc de faire hériter à un droit les permissions d'un autre droit.
— RBAC_2 ou *Constrained RBAC* reprend aussi RBAC_0 en permettant cette fois d'ajouter des contraintes. Par exemple il peut être spécifié qu'un utilisateur ne peut pas prendre deux rôles différents en même temps.
— RBAC_3 est la version la plus complète. Aussi appelé *Symmetric RBAC*, il reprend RBAC_0 et rajoute les hiérarchies de rôles et les contraintes. La figure 2.4 illustre ce modèle.

RBAC et son utilisation du concept de rôle permet de grandement faciliter l'administration des politiques de contrôle d'accès. Le rôle permet de séparer la relation sujet/objet. Avec ce modèle, l'arrivée d'un nouvel utilisateur dans le système n'est pas problématique. Il suffit de lui assigner un ou plusieurs rôles pour qu'il ait directement les accès autorisés à sa fonction. De plus, il est facile de faire évoluer les droits attribués à une fonction. Pour cela, il suffit d'ajouter ou de révoquer des permissions aux rôles associés. Mais RBAC a aussi des inconvénients comme la difficulté de mettre en place des règles dépendantes du contexte de l'utilisateur. Par exemple la règle "les étudiants ont le droit d'accéder seulement à leurs données personnelles" ne peut pas être traitée facilement par RBAC. Une solution serait de

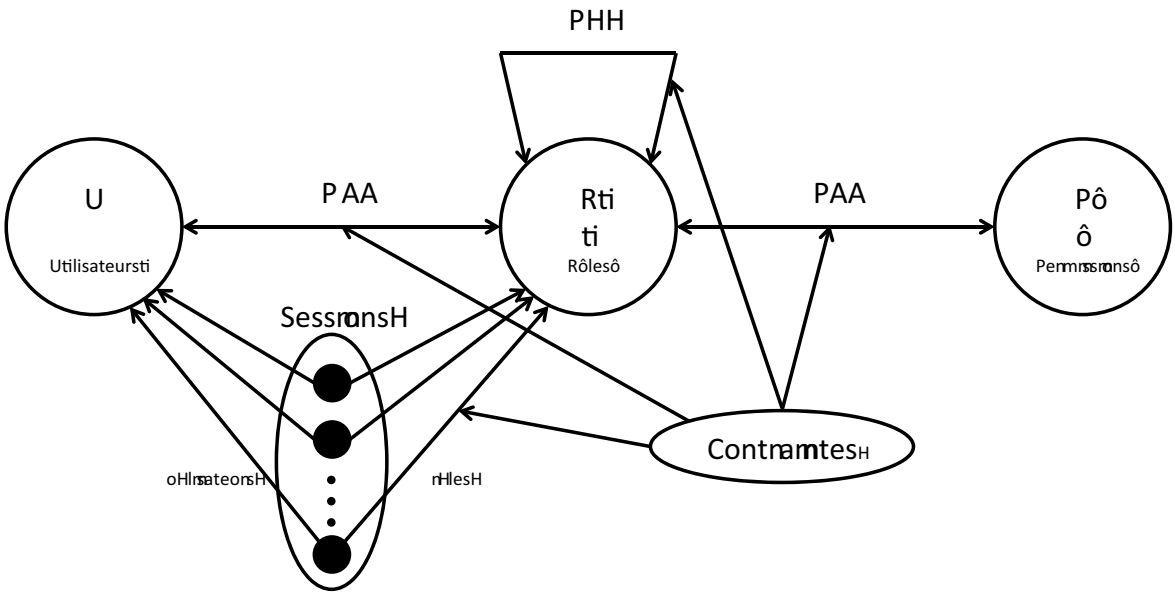


Figure 2.4 — Modèle RBAC

créer un rôle privé à chaque étudiant lui permettant d'accéder à ses données personnelles mais étant donné qu'il y a plusieurs dizaines de milliers d'étudiants dans certaines universités, il n'est pas envisageable de mettre en place une telle solution qui ferait perdre à RBAC sa facilité d'administration.

2.1.4 Modèle de contrôle d'accès basé sur les organisations

Le modèle RBAC permet d'abstraire les politiques de contrôle d'accès en utilisant le concept de rôle qui facilite grandement leur administration. Malgré tout, nous avons vu qu'il n'est pas possible de modéliser avec RBAC des règles du style "un médecin peut avoir accès aux dossiers médicaux de ses patients" où le contexte de la situation rentre en compte. Le modèle de contrôle d'accès basé sur les organisations (*Organization Based Access Control* - OrBAC [1]) permet d'écrire ce genre de règles et ne fait pas que de l'autorisation. Il permet aussi de définir des permissions, des obligations, des recommandations ou des interdictions qui s'appliquent sur une organisation pour contrôler les activités effectuées par des rôles sur des vues [40; 24]. Comme dans RBAC, les rôles permettent d'abstraire les sujets selon leur fonction mais OrBAC introduit deux nouvelles notions d'abstraction :

— les activités permettent d'abstraire les actions à effectuer sur un objet. Par exemple **sélectionner un fichier** ou **lire un fichier** sont deux actions regroupées sous l'activité **consulter**.
— les vues permettent d'abstraire les objets. Par exemple les **radios** d'un patient et ses **résultats sanguins** sont deux ressources regroupées sous la vue **dossier médical**.

1. http ://orbac.org/

Comme son nom l'indique, l'organisation est une notion centrale d'OrBAC. Une organisation peut être vue comme un groupe organisé d'entités actives, ces entités peuvent être des sujets ou des sous-organisations. Plusieurs relations ont été définies utilisant ce concept d'organisation et les notions de sujets, rôles, actions, activités, objets et vues. Ainsi :

- la relation *Empower* relie un sujet, un rôle et une organisation. Par exemple, Empower(hôpital, John, infirmier) indique que le sujet John joue le rôle d'infirmier dans l'organisation hôpital.
- la relation *Use* relie un objet, une vue et une organisation. Par exemple, Use(hôpital, fic_42.txt, fichier_client) indique que l'organisation hôpital utilise le fichier fic_42.txt dans la vue fichier_client.
- la relation *Consider* relie une action, une activité et une organisation. Par exemple, Consider(hôpital, lire, consulter) indique que l'organisation hôpital considère l'action lire comme l'activité consulter.
- la relation *Define* qui relie un contexte, un sujet, une action, un objet et une organisation. Cette relation ajoute la notion de contexte qui permet de définir des contraintes sur le contexte de la situation [24]. Le contexte peut concerner des aspects temporels, spatiaux, etc. L'exemple suivant permet de définir le contexte médecin_traitant : $\forall s \in S, \forall \alpha \in A, \forall o \in O(Define(H, s, \alpha, o, médecin\_traitant) \leftrightarrow nom(o) \in patient(s)$. Cet exemple indique que dans l'hôpital H, le sujet s est dans le contexte *médecin_traitant* si o est le dossier médical d'un patient de s.

Ces notions et ces relations permettent à OrBAC de définir des politiques de sécurité à deux niveaux. D'un côté des politiques de bas niveau en utilisant un sujet, une action et un objet avec la relation Is_permitted(Sujet, Action, Objet). De l'autre côté des politiques abstraites de haut niveau avec la relation Permission(Organisation, Rôle, Activité, Vue, Contexte). Ainsi :

si Permission(Org, Rôle, Activité, Vue, Contexte) ∧
Empower(Org, sujet, Rôle) ∧
Consider(Org, action, Activité) ∧
Use(Org, objet, Vue)
Define(Org, sujet, action, context)
alors Is_permitted(sujet, action, objet)

Cette définition de politique de sécurité à deux niveaux est illustrée par la figure 2.5 reprise et traduite de [25].

Des conflits peuvent apparaître en utilisant ces deux niveaux. Si par exemple il existe deux règles, la première autorise un médecin à consulter le dossier médical de ses patients seulement et la deuxième autorise un chirurgien à consulter les dossiers médicaux d'une personne même si elle ne fait pas partie de ses patients. Un chirurgien étant un médecin, il y a un conflit entre les deux règles. Pour avoir un moyen de savoir quelle règle utiliser, OrBAC utilise un système de priorités permettant dans notre cas de dire que la deuxième règle est prioritaire à la première.

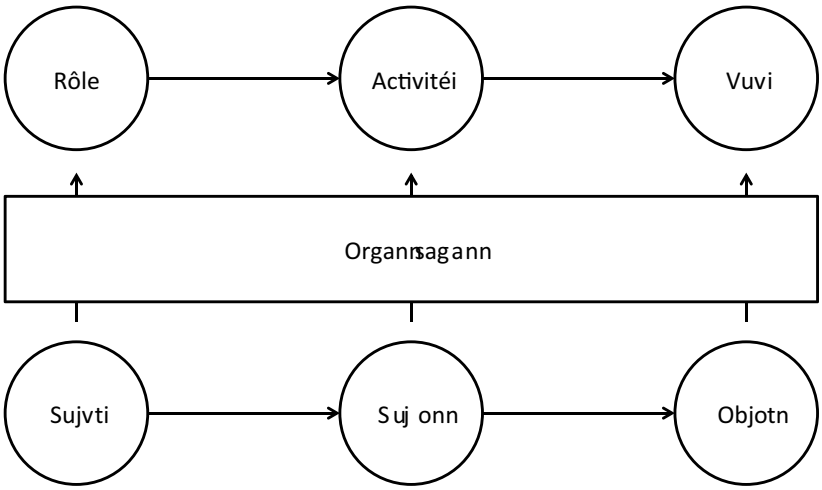


Figure 2.5 — Politique de sécurité OrBAC à deux niveaux

On peut donc voir que OrBAC va plus loin de RBAC. Il ne propose plus seulement de créer avec les rôles des catégories de sujets, il permet de créer des catégories sur les objets et sur les actions. OrBAC offre donc un cadre à l'administrateur lui permettant de créer à sa guise des règles jouant sur l'abstraction des sujets, des objets et des actions.

2.1.5 Modèle de contrôle d'accès basé sur les attributs

Depuis quelques années avec l'avènement d'Internet, nous avons accès à un nombre incalculable d'informations et de données. De plus en plus, des informations sont ajoutées à des données pour en décrire le contenu ou le contexte. Ces informations sont appelées des méta-données. Par exemple, on va ajouter à une photo la date ou les coordonnées GPS de l'endroit où elle a été prise. Utiliser toutes les informations disponibles peut permettre de définir des politiques d'autorisation puissantes et précises, encore faut-il pouvoir les utiliser correctement.

Le modèle de contrôle d'accès basé sur les attributs (*Attribute Based Access Control* - ABAC) permet la construction de politiques en utilisant les attributs des sujets, des objets ou de l'environnement. Un attribut correspond à tout ce qui permet de caractériser une entité et qui peut prendre une valeur. Par exemple, une personne peut avoir comme attribut son nom, sa fonction, son age, sa localisation, etc.. Techniquement, ABAC permet de modéliser n'importe quelle notion à partir des informations disponibles. Par exemple, en ayant des informations sur les fonctions des personnes appartenant à une organisation, ABAC peut utiliser la notion de rôle comme le fait RBAC ou OrBAC. En ayant des informations sur les types de ressources, il est possible de créer des catégories pour abstraire ces ressources selon leur type comme le fait OrBAC avec les vues. Barker définit une catégorie comme n'importe quelle classe ou groupe distinct et fondamental parmi les nombreux qui existent auquel une entité peut être assignée [7]. Mais contrairement aux autres modèles de contrôle d'accès, ABAC ne va pas fournir de cadre à l'administrateur. Il peut ainsi créer tout type de catégories mais ne sera pas guidé dans le choix des catégories à utiliser ni dans leur utilisation. Avec ABAC, l'administrateur va donc pouvoir créer un système de contrôle d'accès selon

ses besoins et les informations dont il dispose.

ABAC est donc un modèle offrant une très grande liberté quant à l'utilisation des informations disponibles et des catégories à utiliser pour abstraire ces informations mais ne propose aucun cadre à l'administrateur. Il faut donc choisir des catégories appropriées aux informations disponibles mais aussi aux besoins du système. Pour notre part, nous cherchons à protéger les données de vie privée de l'utilisateur, nous devons donc utiliser des informations relatives à ce besoin de protection. Nous allons donc maintenant étudier d'autres modèles de contrôle d'accès, spécialisés dans la protection de la vie privée pour voir quelles catégories ils utilisent.

2.1.6 Modèles de contrôle d'accès pour la protection de la vie privée

La protection de la vie privée étant un sujet prenant de plus en plus d'importance, de nombreux travaux ont étudié des modèles de contrôle d'accès mettant en avant cette protection. En effet de nouveaux besoins apparaissent. Par exemple, une personne peut accepter de partager ses données selon l'utilisation qui va en être faite. Ce n'est pas la même chose d'utiliser des données à des fins statistiques et anonymes que de vouloir les revendre à des entreprises voulant étoffer leur base de clients. Savoir pourquoi une entité veut avoir accès à une donnée devient donc un besoin lorsqu'il s'agit de protéger la vie privée d'une personne.

PBAC (*Purpose Based Access Control*) est un modèle de contrôle d'accès basé sur les intentions [13]. L'intention correspond au but, à la raison pour laquelle une entité veut avoir accès à des données. Par exemple, un développeur peut vouloir récupérer l'âge des utilisateurs de son système pour mieux cibler les tranches d'âge les plus importantes ou bien récupérer un maximum d'informations et vendre cette base d'informations à une régie publicitaire. Pour utiliser ce modèle, l'administrateur doit créer un arbre hiérarchique d'intentions (exemple figure 2.6). Les règles d'autorisation associent une ressource avec une intention. Lors de chaque demande d'accès, l'intention liée à la demande est comparée à celle de la règle d'autorisation. Si l'intention de la demande appartient à la branche de celle de la règle d'autorisation, alors la décision de la règle s'applique.

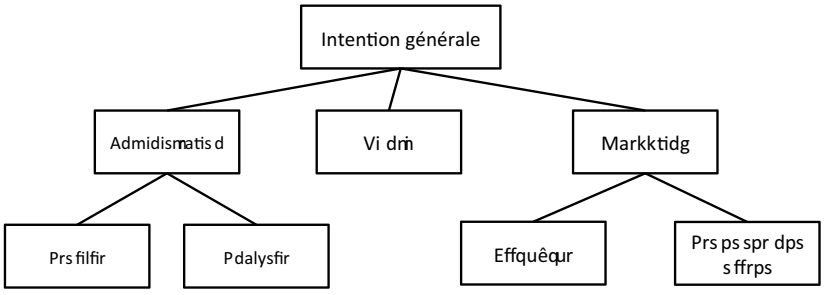


Figure 2.6 — Exemple d'arbres hiérarchique d'intentions

De nombreux modèles utilisent l'intention mais ce n'est pas la seule notion utilisée. Par exemple, P-RBAC (*Privacy-aware Role Based Acces Control*) étend RBAC avec les notions d'intentions, de conditions et d'obligations pour protéger au mieux la vie privée des utilisateurs [58]. Les obligations sont des actions qui doivent être effectuées une fois l'accès à la ressource autorisé. Par exemple, une personne peut autoriser l'accès à ses données mais

seulement si elles sont supprimées une fois utilisées, pour éviter que les données soient conservées. Les conditions sont des pré-requis à vérifier avant d'obtenir des données. Par exemple, une condition peut être de vérifier que la personne soit majeure ou d'avoir le consentement des parents avant d'utiliser ou de divulguer les données. D'autres notions peuvent être utilisées comme par exemple :

— la confiance [76]. Le niveau de confiance dans l'entité qui demande un accès à nos données joue un rôle dans la décision à prendre. Accepter de divulguer des informations à une entité dont on ne fait pas confiance ou qu'on ne connaît pas augmente le risque que ces informations soient utilisées à mauvais escient. Au contraire, plus le niveau de confiance est élevé envers l'entité, plus on sera enclin à dévoiler nos données.
— la rétention de l'information [3]. La durée de stockage peut être un critère important pour choisir si une donnée peut être partagée ou non. Par exemple, si une entreprise veut avoir accès à des informations comme l'âge ou la taille d'une personne pour faire une analyse statistique et supprime les informations individuelles juste après, il est fort possible qu'une personne acceptera de fournir ses informations. Au contraire une entreprise voulant avoir accès à des photos et les stocker infiniment sur leurs serveurs incitera moins une personne à vouloir divulguer ses données.
— Ajam et al. ont étendu OrBAC pour prendre en compte des exigences de protection de vie privée [4]. En plus d'utiliser l'intention comme catégorie, ils ont rajouté le consentement et la précision des objets. L'utilisation du consentement permet de demander l'avis de l'utilisateur avant d'autoriser l'accès. La précision des objets permet quant à elle de jouer sur la précision de l'information divulguée. Par exemple, selon le but de la demande, une information n'a pas besoin d'avoir un niveau de précision maximale. Un exemple de précision est donné dans [4] sur les informations de localisation. Les différents niveaux de précision sont spécifiés par deux paramètres : l'identité de l'utilisateur et un paramètre k venant de l'algorithme de k-anonymat. En jouant sur ces deux paramètres, de nombreux niveaux de précisions peuvent être utilisés et l'utilisateur peut écrire sa politique en choisissant le niveau de précision qui convient selon les autres attributs de chaque règle.

Maintenant que nous avons vu les modèles disponibles pour créer une politique d'autorisation, nous allons voir les techniques existantes pour écrire cette politique d'autorisation.

2.2 Solutions pour mettre en œuvre une politique

Les modèles de contrôle d'accès sont des outils permettant d'exprimer des politiques d'autorisation en étant plus ou moins abstraits et plus ou moins complets. Mais faut-il encore pouvoir écrire ce que l'on souhaite exprimer. La plupart de ces modèles exigent la présence d'un administrateur, d'un expert du domaine, afin de garantir la mise en place des politiques d'autorisation et de les maintenir. Notre problématique nous empêche d'avoir cet expert pour écrire les politiques d'autorisation. Dans notre cas, c'est au propriétaire des données d'écrire sa politique d'autorisation, c'est à dire de spécifier qui va avoir le droit

d'accéder à quelles informations et dans quelles situations. Nous allons présenter dans cette section deux types d'outils permettant de définir une politique d'autorisation : à partir d'une interface graphique ou à partir d'un éditeur textuel.

2.2.1 A partir d'une interface graphique

Les interfaces graphiques permettent d'utiliser des techniques visuelles et des outils d'interaction homme machine qui peuvent aider une personne à définir sa politique d'autorisation. L'objectif principal de l'utilisation d'outils graphiques est la simplification du processus. La majorité des utilisateurs de ces systèmes n'ont pas de connaissances techniques en administration, ils ne sont donc pas capable d'écrire leur politique d'autorisation directement dans un langage prévu à cet effet. A la place, les outils graphiques vont permettre d'apporter plus de simplicité pour informer les utilisateurs sur les différentes options disponibles et leur permettre de faire leur choix. Nous allons présenter deux systèmes : *Privacy Guard Manager* de la distribution CyanogenMod pour le système d'exploitation mobile Android et le système utilisé par le réseau social Facebook.

2.2.1.1 Privacy Guard Manager de CyanogenMod

CyanogenMod [2] est une distribution alternative pour Android basé sur *Android Open Source Project*. Le but de cette distribution est d'apporter plus de performances, de fiabilités, de fonctionnalités et de sécurités que la version constructeur. Une des fonctionnalité ajoutée par CyanogenMod est le Privacy Guard Manager. Cet outil permet de gérer les permissions attribuées à chaque application. Dans Android, à chaque fois qu'un utilisateur veut installer une application, le système le prévient que cette application nécessite un certain nombre de permissions. Pour finaliser l'installation de cette application, l'utilisateur doit accepter de fournir l'ensemble des permissions à l'application, il n'a pas le choix de n'en autoriser qu'un sous-ensemble, voire aucune. Si l'utilisateur refuse, l'installation est suspendue. Cette approche du tout ou rien est un problème pour les utilisateurs. Si quelqu'un veut absolument installer une application mais qu'il considère qu'une ou plusieurs permissions n'ont pas à être utilisées par l'application, il ne peut pas les lui ôter. Ses seuls choix sont soit de refuser de l'installer ou d'accepter mais en lui donnant toutes les permissions demandées.

Privacy Guard Manager permet de passer outre le système d'origine d'Android et de gérer, une fois l'application installée, sa liste de permissions. Ainsi un utilisateur peut installer une application même si celle-ci contient une ou plusieurs permissions qu'il ne souhaite pas lui accorder, après l'installation, en utilisant Privacy Guard Manager, il pourra avoir accès à la liste des applications (Figure 2.7 gauche) et pour chacune d'entre elle la liste des permissions (Figure 2.7 droite).

Un utilisateur peut donc à tout moment modifier l'accès que peut avoir une application sur ses données. Pour chaque permission, en plus d'avoir le choix d'accepter cette permission ou de la refuser, il peut aussi choisir que le système lui demande à chaque fois où l'application veut avoir accès à la ressource en question si elle peut ou non avoir accès. Ainsi

2. http ://www.cyanogenmod.org/

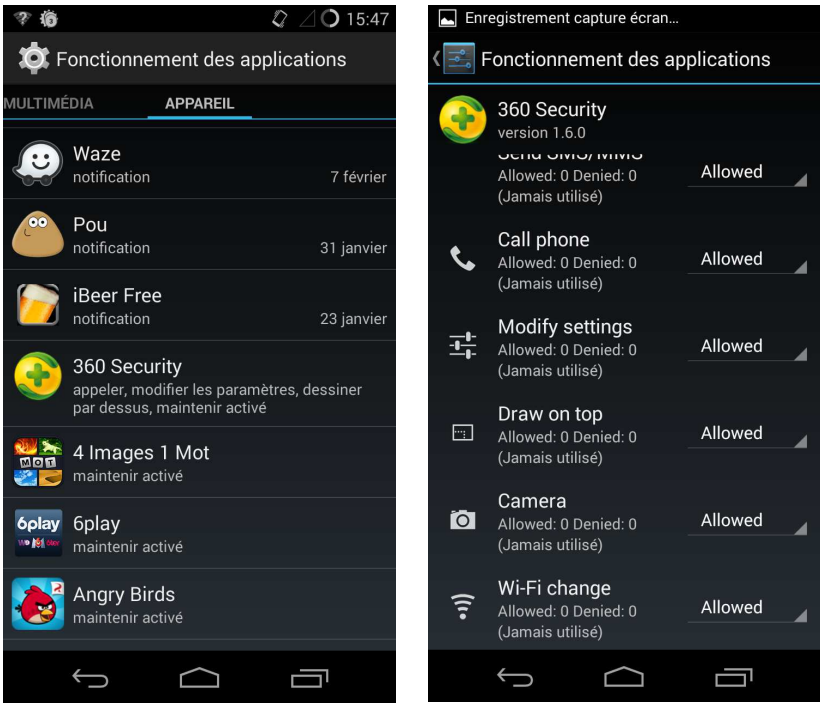


Figure 2.7 — Privacy Guard Manager : liste des applications

Privacy Guard Manager permet à l'utilisateur de gérer sa politique d'autorisation à l'exécution et donc de l'adapter selon la situation en cours.

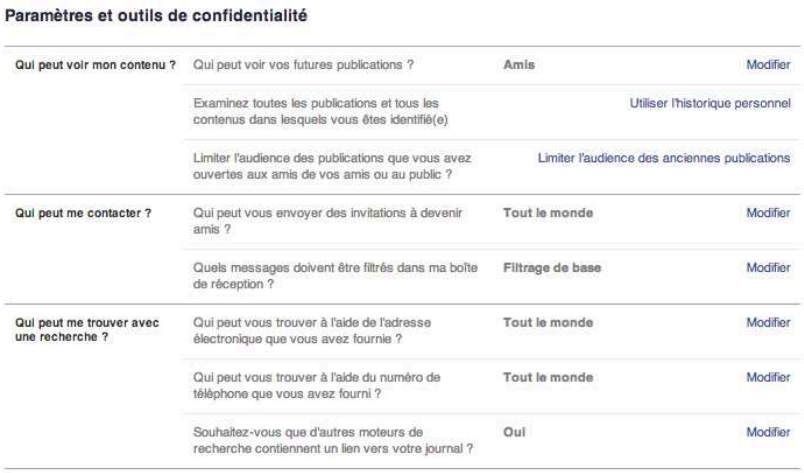
Si Privacy Guard Manager est très simple à utiliser, il demande aussi beaucoup de temps pour mettre en place et maintenir une politique d'autorisation. Selon le recensement effectué par Google en 2013[3], les français ont en moyenne 32 applications installées sur leur smartphone et seulement 3 payantes. De plus, en analysant les permissions des 50 applications gratuites les plus téléchargées, nous avons trouvé que chaque application demande en moyenne 11,4 permissions dont 5,72 ont un impact fort sur la vie privée. Un utilisateur doit donc gérer en moyenne 364 permissions dont 183 ont un impact direct sur sa vie privée. Avec Privacy Guard, l'utilisateur gère une par une les permissions. Imaginons qu'un jour, après avoir mis en place sa politique d'autorisation, il change d'avis et ne veut plus autoriser l'accès aux applications à une certaine ressource. Il va devoir passer toutes les applications en revue pour changer la règle d'accès à cette ressource. De plus à chaque fois qu'il installe une nouvelle application, il doit faire attention à bien aller régler les permissions pour cette application. Ce problème est lié à l'utilisation de l'approche IBAC et montre des difficultés de passage à l'échelle en cas d'utilisation d'un nombre important d'applications. Un autre inconvénient apparaît en utilisant cette approche : gérer les permissions hors de l'utilisation de l'application oblige l'utilisateur à se mettre dans le bon contexte pour savoir ce qu'il veut accorder ou refuser. Cet effort cognitif n'est pas facile et peut amener à des erreurs sur la gestion des permissions. Par exemple une personne ne veut pas qu'une application accède à sa localisation et supprime toutes les permissions correspondantes. Or de temps en temps, elle utilise une application GPS pour trouver des itinéraires. En ayant supprimé la permis-

3. http ://think.withgoogle.com/mobileplanet/fr/

sion liée à la localisation, l'application GPS ne fonctionnera pas correctement et la personne ne fera peut être pas le rapprochement entre la permission supprimée et le non fonctionnement de l'application. Si la gestion de cette permission était intervenue pendant l'utilisation de l'application, la personne aurait compris que dans ce cas là, il fallait laisser l'accès à sa localisation et n'aurait pas eu de problèmes.

2.2.1.2 Facebook

Facebook est le réseau social le plus utilisé au monde. Selon son bilan annuel 2013, la plateforme compte 757 millions d'utilisateurs qui se connectent quotidiennement et 1,23 milliards qui se connectent au moins une fois par mois. Chaque personne peut parler de sa vie, publier des photos, des vidéos, utiliser des applications ou même utiliser son compte pour avoir accès à de nombreux sites (voir section 1.3.2). Le compte d'une personne est donc potentiellement une énorme base d'informations sur sa vie privée auquel tout le monde peut accéder. Pour éviter que n'importe quelle information soit accessible à n'importe qui, Facebook a mis en place un système d'autorisation pour réguler qui a le droit d'accéder à quoi accessible sur le site via des écrans d'options (figure 2.8). Grâce à ces options, l'utilisateur peut décider qui est autorisé à voir ce qu'il publie, qui peut le contacter, qui peut le retrouver grâce à une recherche. Ainsi il est possible de choisir si le contenu ajouté par l'utilisateur doit être vu seulement par lui, par ses amis ou par tout le monde. Il est aussi possible pour certaines options d'utiliser les groupes d'amis pour ne donner l'accès qu'à un certain groupe.



Paramètres et outils de confidentialité

Qui peut voir mon contenu ?	Qui peut voir vos futures publications ?	**Amis**	Modifier
	Examinez toutes les publications et tous les contenus dans lesquels vous êtes identifié(e)		Utiliser l'historique personnel
	Limiter l'audience des publications que vous avez ouvertes aux amis de vos amis ou au public ?		Limiter l'audience des anciennes publications
Qui peut me contacter ?	Qui peut vous envoyer des invitations à devenir amis ?	**Tout le monde**	Modifier
	Quels messages doivent être filtrés dans ma boîte de réception ?	**Filtrage de base**	Modifier
Qui peut me trouver avec une recherche ?	Qui peut vous trouver à l'aide de l'adresse électronique que vous avez fournie ?	**Tout le monde**	Modifier
	Qui peut vous trouver à l'aide du numéro de téléphone que vous avez fourni ?	**Tout le monde**	Modifier
	Souhaitez-vous que d'autres moteurs de recherche contiennent un lien vers votre journal ?	**Oui**	Modifier

Figure 2.8 — Paramètres de confidentialité de Facebook

A première vue, le système d'autorisation de Facebook informe clairement l'utilisateur et lui permet de régler à sa convenance sa politique de contrôle d'accès en utilisant sur les différents paramètres présents un système de rôles similaire à RBAC. Pourtant en navigant sur les autres pages d'options, on retrouve d'autres paramètres portant sur les mêmes aspects. Ainsi comment faire la différence entre le paramètre "Qui peut voir mon contenu" de la page "Confidentialité" et le paramètre "Qui peut voir le contenu de mon journal" (le journal est le recueil de photos, publications et expériences qui représente un utilisateur) de la page "Paramètres d'identification et de journal". Il est d'ailleurs possible de restreindre

l'accès aux amis pour le premier et de laisser l'accès libre dans le deuxième mais dans ce cas, que se passera-t-il ? Qui aura accès aux informations ?

Ainsi, le système de Facebook semble facile à utiliser à première vue mais manque de clarté. Disperser les différentes options ou proposer trop d'options à l'utilisateur va amener à des erreurs et à compliquer la mise en place de la politique d'autorisation alors que l'utilisation d'un système graphique est faite pour la faciliter et la rendre accessible pour le plus grand nombre. De plus, que ce soit pour CyanogenMod ou pour Facebook, l'utilisateur ne pourra pas faire ce qu'il veut, il est limité par les choix du concepteur et ne pourra utiliser que les paramètres disponibles. De plus, toutes ces interfaces sont limitées à un modèle, le moindre changement du modèle entraînera un changement de l'interface. Ces changements sont problématiques s'ils sont trop importants ou trop fréquents car ils peuvent désorienter l'utilisateur. Donc si l'interface graphique offre une accessibilité accrue, elle ne permet pas l'écriture de politiques d'autorisations complètes. C'est pourquoi nous allons maintenant présenter un autre type de système d'autorisation : les systèmes textuels qui permettent de définir précisément la politique d'autorisation voulue.

2.2.2 A partir d'un éditeur textuel

Alors qu'une interface graphique peut être spécialement aménagée pour aider l'utilisateur et le guider dans la démarche de mise en place d'une politique d'autorisation, un éditeur textuel utilise un langage de programmation pour écrire la politique d'autorisation. Un éditeur textuel est moins accessible qu'un graphique car il impose à l'utilisateur de connaître le langage de programmation et de savoir l'utiliser. Par contre, il autorise plus de flexibilité et permet d'écrire des politiques d'autorisation bien plus puissantes. Nous allons présenter un des langages permettant d'écrire des politiques d'autorisation via un éditeur textuel : XACML.

XACML [4] (*eXtensible Access Control Markup Language*) est un langage standardisé par OASIS permettant d'écrire des politiques de contrôle d'accès basé sur le format XML. Le langage XACML utilise des attributs pour construire les politiques, de ce fait il convient très bien à ABAC. Malgré tout, il est aussi possible de l'utiliser avec un modèle RBAC. Une politique d'autorisation XACML est construite par un ensemble de règles. Chacune de ces règles décrit une situation et est associée à une décision. Les règles permettent de savoir si un sujet peut faire une action sur une ressource.

Il est possible de définir les sujets, les ressources et les actions avec n'importe quels attributs imaginables. Il est possible d'abstraire les sujets en utilisant des rôles comme RBAC ou d'abstraire des actions en utilisant des activités comme OrBAC. Grâce à sa généricité, XACML donne la possibilité à une personne d'écrire toutes les règles dont elle a besoin et ce n'est pas limité aux trois entités que sont les sujets, les ressources et les actions. Il est tout à fait possible de rajouter des conditions sur les aspects temporels ou spatiaux par exemple à travers la notion de catégorie.

Par contre XACML demande une technique qui n'est pas accessible à tout le monde. On voit déjà avec l'exemple de requête et de politiques qu'il n'est pas envisageable de propo-

4. http ://www.oasis-open.org/committees/xacml/

ser au grand public de devoir écrire ce genre de règles. Des travaux ont été effectués pour faciliter l'écriture de règles XACML. Par exemple, Stepien et al. [72] ont réalisé un éditeur destiné aux utilisateurs n'ayant pas de connaissance en XACML (figure 2.9). L'utilisateur peut choisir pour chaque attribut ce qu'il représente, l'opérateur à utiliser et sa valeur. Il est ainsi possible de créer des règles sans avoir à utiliser le format XML, lourd pour des non initiés. Malgré tout, certaines notions propres à XACML comme la structure d'une politique sont toujours utilisées. Ces notions doivent être apprises et comprises pour pouvoir utiliser l'outil. Cela montre qu'il n'est pas simple de faire une interface graphique pour un langage aussi souple que XACML.

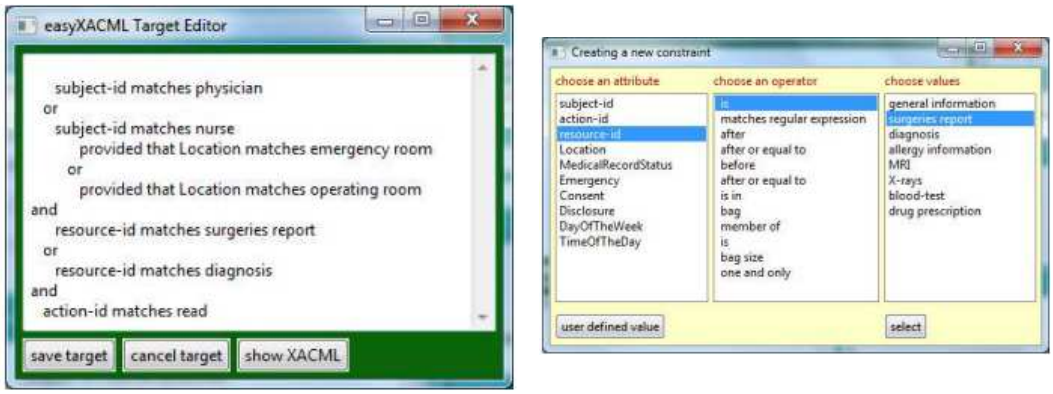


Figure 2.9 — Un éditeur pour faciliter l'écriture de règles XACML

2.3 Conclusion

Ce chapitre a permis de détailler plusieurs modèles de contrôle d'accès. Que ce soit OrBAC, RBAC ou des modèles plus orientés protection de la vie privée comme P-RBAC ou PBAC, ces modèles définissent des catégories et un cadre d'utilisation. Nous avons donc choisi d'utiliser ABAC qui, s'il ne donne pas de cadre, permet d'utiliser n'importe quelles catégories. Cela laisse libre le concepteur du système de s'adapter selon le périphérique sur lequel il va installer le système et selon les informations qui seront disponibles pour effectuer le contrôle d'accès.

Si ces modèles définissent ce qu'il est possible d'exprimer dans une politique d'autorisation, il faut ensuite pouvoir écrire cette politique. Nous avons vu deux approches possibles, l'une qui permet d'éditer les politiques graphiquement et l'autre textuellement. L'édition textuelle est bien plus compliquée que la graphique car elle demande plus de connaissances et de techniques. C'est pour cette raison que nous avons choisi de ne pas utiliser un éditeur textuel, comme cela notre système sera utilisable par tous les utilisateurs. Nous n'avons pas choisi non plus d'utiliser un éditeur graphique pour éviter d'être limité à un seul modèle. Notre approche est détaillée dans le prochain chapitre.

3 Aide à la décision

« Une bonne décision est basée sur la connaissance et pas sur des nombres. »
Platon

LE problème qui est à l'origine de cette thèse est la difficulté pour une personne d'écrire sa politique d'autorisation afin de protéger ses données de vie privée. Si la personne n'est pas capable d'écrire seule sa politique d'autorisation, il lui faut une aide, mais il n'est pas possible de mettre un expert derrière chaque personne. Un système informatique peut aider un utilisateur en apprenant son comportement et en lui proposant des règles afin d'écrire sa politique d'autorisation. Ce chapitre présente les systèmes d'aide à la décision qui peuvent remplir ce rôle, en commençant par une description de ce qu'est l'aide à la décision et les différents systèmes qui existent pour arriver plus particulièrement aux systèmes qui nous intéressent, les systèmes interactifs d'aide à la décision. Le comportement d'un utilisateur va dépendre de la situation, de qui demande l'accès à ses ressources, et aussi d'autres éléments. Par exemple, la décision de divulguer ou non une donnée peut dépendre du moment de la demande, du lieu où l'utilisateur se trouve ou encore de la façon dont cette donnée sera utilisée. Nous avons donc affaire à une décision qui va dépendre de plusieurs critères. Une section est donc destinée à l'aide à la décision multicritère avec des explications sur les méthodes et techniques que l'on utilise. Le but du système est de proposer à l'utilisateur des règles d'autorisation pouvant lui convenir, donc de lui faire des recommandation. Une section du chapitre présente donc les systèmes de recommandation : ce qu'est un profil utilisateur, l'apprentissage automatique, les différentes échelles pour représenter les préférences et enfin les différentes approches de recommandations.

3.1 L'aide à la décision

L'aide à la décision est un ensemble de méthodes et techniques, permettant à une personne confrontée à un problème de l'aider à prendre une décision. L'aide à la décision est utilisée dans des domaines très diversifiés tels que la finance, la gestion ou en médecine [1] [69] [83]. Les méthodes d'aide à la décision permettent de fournir des informations sur le problème (phase d'intelligence), des outils pour l'analyser ainsi que pour trier les informa-

tions pertinentes qui peuvent aider le décideur (phase de conception) dans le but de formuler plusieurs alternatives qui pourront être proposées à ce décideur (phase de choix). Ces étapes se retrouvent dans toutes les prises de décision que Simon a définies comme étant un processus cognitif [68].

Nous pouvons retrouver plusieurs types d'acteurs lors d'une prise de décision. La personne qui va prendre la décision est le décideur. C'est à lui que s'adresse ce processus d'aide à la décision dans lequel il est impliqué pour exprimer ses préférences. Un autre acteur important de ce processus est l'homme d'étude [63]. C'est un expert qui va aider le décideur à comprendre le problème, lui donner des informations essentielles et lui expliquer les conséquences des différentes alternatives lorsque c'est possible. Généralement ces deux acteurs sont deux personnes distinctes. D'autres catégories d'acteurs peuvent intervenir pendant le processus comme le demandeur qui est à l'origine de la requête d'aide à la décision, les intervenants qui vont donner leurs avis en fonction de leurs préférences ou les agis qui regroupent toutes les personnes concernées par les conséquences de la décision.

Dans [63], Bernard Roy nous propose la définition suivante de l'aide à la décision :

Définition. *L'aide à la décision est l'activité de celui qui, prenant appui sur des modèles clairement explicités mais non nécessairement complètement formalisés, aide à obtenir des éléments de réponses aux questions que se pose un intervenant dans un processus de décision, éléments concourant à éclairer la décision et normalement à prescrire, ou simplement à favoriser, un comportement de nature à accroître la cohérence entre l'évolution du processus d'une part, les objectifs et le système de valeurs au service desquels cet intervenant se placé d'autre part.*

L'objectif final de l'aide à la décision n'est donc pas de prendre la décision à la place du décideur mais bien de lui donner les informations importantes pour qu'il puisse les comprendre, lui donner des éléments de solution et les conséquences de chaque alternative afin qu'il puisse prendre la décision la plus appropriée. Il ne s'agit pas de décision optimale mais de décision satisfaisante. En effet Simon explique dans son principe de la rationalité limitée [67] que certaines hypothèses admises à l'époque concernant la rationalité de certaines décisions n'étaient pas valides. Il redéfinit la rationalité classique en rationalité limitée et énonce plusieurs hypothèses :

— L'accès à l'information est limité. Avec les nouvelles technologies et l'accès à internet, on pourrait croire que l'accès à l'information est illimité. Mais s'informer demande beaucoup de ressources que ce soit en terme d'argent ou de temps. Il n'est donc pas possible d'obtenir toutes les informations disponibles pour un problème donné.

— La capacité cognitive d'un individu est elle aussi limitée. Il n'est pas possible pour une personne d'analyser toutes les informations disponibles, de comprendre toutes les situations possibles et d'optimiser son choix. Lorsqu'un décideur arrive à trouver une solution satisfaisante, il doit s'en contenter et ne pas chercher la solution optimale, au sens mathématique, car il arrive que cette solution n'existe pas.

— Le décideur a une vision floue de ses préférences et n'a pas une idée claire de son problème. Les préférences d'un décideur peuvent évoluer pendant le processus de décision et les critères utilisés pour prendre une décision peuvent être contradictoires. Il n'est alors pas possible de trouver une décision optimale. Prenons l'exemple d'une personne voulant acheter une voiture. La voiture doit être très puissante mais aussi la plus écologique possible. Un moteur puissant générera toujours plus de pollution qu'un moteur plus modeste, il est alors impossible de maximiser les deux critères. Le décideur devra faire un compromis entre les deux critères : puissance et écologie.

L'information est une donnée essentielle de l'aide à la décision. Ces dernières années, le développement des outils et du matériel informatique a permis à de nombreux systèmes d'aide à la décision de voir le jour. Il est aujourd'hui possible, en utilisant des systèmes d'aide à la décision, de traiter beaucoup plus d'informations qu'une personne seule. Dans notre cas, nous avons besoin de traiter beaucoup d'informations et nous ne pouvons pas utiliser une équipe d'expert derrière chaque utilisateur pour l'aider. L'utilisation des ordinateurs est donc indispensable. Nous allons présenter un type de système d'aide à la décision en particulier, les Systèmes Interactifs d'Aide à la Décision (SIAD) qui utilise la capacité de traitement des ordinateurs pour aider les utilisateurs.

3.2 Les Systèmes Interactifs d'Aide à la Décision

L'expression Systèmes Interactifs d'Aide à la Décision (SIAD ou *Decision Support System*) a été définie au début des années soixante-dix par Gorry et Scott-Morton [32]. De nombreuses définitions des SIAD ont été proposées depuis, comme par exemple celle de Keen et Scott-Morton [41] :

Définition. *Les systèmes interactifs d'aide à la décision associent les ressources intellectuelles d'une personne avec les capacités d'un ordinateur pour améliorer la qualité des décisions. Ce sont des systèmes informatiques d'assistance pour les décideurs qui doivent gérer des problèmes semi-structurés.*

Les SIAD combinent les modèles mathématiques pour analyser le comportement des décideurs et les outils informatiques pour leur interactivité et leurs techniques de visualisation. Grâce à la puissance de calculs des ordinateurs de nos jours et aux avancées logicielles, un SIAD peut aider un décideur, en recréant un processus de décision, à gérer des problèmes de plus en plus complexes impliquant toujours plus d'information. L'interactivité entre le décideur et le système va permettre à ce dernier d'analyser le comportement du décideur et d'apprendre ses préférences. Cette interactivité va permettre au système de comprendre le point de vue du décideur afin de lui proposer une aide plus précise. L'algorithme général de fonctionnement d'un SIAD est illustré par la figure 3.1 emprunté à [30].

On peut voir l'aide à la décision de deux points de vue : une aide à la décision mono-critère ou une aide à la décision multicritère. L'approche mono-critère comme son nom l'in-

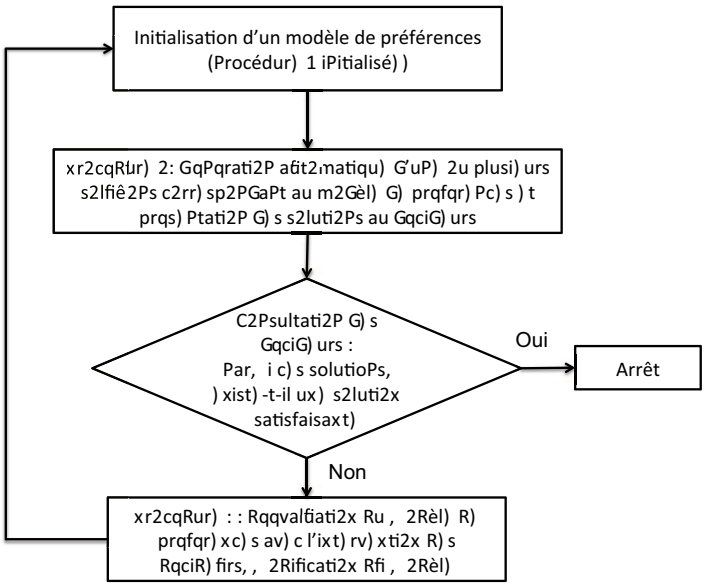


Figure 3.1 — Algorithme général de fonctionnement d'un SIAD [30]

dique ne prend en compte qu'un seul critère dans le processus de prise de décision. Par exemple une entreprise devant choisir un fournisseur pour acheter une pièce dont elle a besoin ne prendra en compte que le prix de la pièce. L'approche multicritère quant à elle utilise plusieurs critères pour prendre une décision. A cause de la nature complexe de notre problème, de nombreux critères peuvent rentrer en compte pour prendre une décision. Par exemple, je suis d'accord pour que ma famille accès à mes agendas à n'importe quelle moment. Par contre, mes amis ne peuvent pas avoir accès à mon agenda professionnel et mes collègues ne peuvent avoir accès qu'à mon agenda professionnel et seulement durant les heures de travail. Dans ce cas là, ma décision dépend de nombreux critères : la donnée, l'heure et la personne qui demande. Nous allons donc maintenant présenter l'aide à la décision multicritère.

3.3 L'aide à la décision multicritère

L'aide à la décision multicritère (*Multi-Criteria Decision Analysis* - MCDA) est un sous domaine de l'aide à la décision comprenant un grand nombre de méthodes permettant de résoudre des problèmes complexes qui ne pourraient pas être résolus avec une approche mono-critère. Selon Zeleny [82] "C'est seulement lorsqu'on est face à plusieurs attributs, objectifs, critères, fonctions, etc., que l'on peut parler d'aide à la décision et de sa théorie. Lorsque les choix deviennent plus complexes et sont décrits par plusieurs objectifs, le problème de combiner ces aspects différents en une seule mesure d'utilité devient plus compliqué et moins réalisable."

Bien que ces méthodes ne soient pas toujours les plus efficaces sur des problèmes simples et bien structurés, elles permettent de modéliser des problèmes plus complexes en intégrant plusieurs paramètres de décision. Le but de ces méthodes reste le même qu'un système d'aide à la décision : aider un décideur et non pas le remplacer. Les solutions obtenues sont

généralement des recommandations qui peuvent être acceptées ou refusées par le décideur.

Deux écoles existent en MCDA et suivent des méthodes différentes. L'école américaine avec des méthodes basées sur les fonctions d'utilité et l'école européenne principalement basée sur le concept de surclassement [63].

3.3.1 L'école américaine

En aide à la décision multicritère, l'école américaine part de l'hypothèse que pour aider au mieux le décideur, il faut maximiser la fonction d'utilité multi-attributs de l'action. Cette fonction d'utilité multi-attributs est construite en agrégeant les valeurs d'utilité de l'action, c'est donc une combinaison de fonction d'utilité mono-attribut. Une fonction d'utilité multi-attributs est une fonction $u : X \mapsto \mathbb{R}$ de la forme :

$$\forall x = (x_1, x_2, x_3), y = (y_1, y_2, y_3) \in X, x \succsim y \Leftrightarrow u(x_1, x_2, x_3) \geq u(y_1, y_2, y_3) \tag{3.1}$$

où $x \succsim y$ signifie que x est préféré ou est indifférent à y.

Une fonction d'utilité multi-attributs va donc permettre d'affecter des scores aux différentes actions et représenter numériquement les préférences du décideur. Le but étant de maximiser ce score, il est donc possible de classer les actions de la moins désirable (score le plus faible) à la plus désirable (score le plus haut). Il y a deux conditions à respecter pour construire une fonction d'utilité multi-attributs :

— les préférences de l'utilisateur doivent être numériquement représentables et si $u(a) = u(b)$ alors cela veut dire que pour l'utilisateur, a et b induisent le même degré de satisfaction.
— la fonction d'utilité multi-attributs doit pouvoir se décomposer en une combinaison simple de fonction d'utilité mono-attribut.

La forme de la fonction d'utilité multi-attributs dépend de cette décomposition. En effet il est plus simple de construire la fonction d'utilité multi-attributs d'une forme dont on sait qu'elle sera réalisable cognitivement. Les formes les plus communes sont :

— la décomposition additive : il existe des fonctions $u_i : X_i \mapsto \mathbb{R}$ telles que $u(x_1, ..., x_n) = \sum_{i-1}^{n} u_i(x_i)$.
— la décomposition multiplicative : il existe des fonctions $u_i : X_i \mapsto \mathbb{R}$ telles que $u(x_1, ..., x_n) = \prod_{i-1}^{n} u_i(x_i)$.
— la décomposition multilinéaire : il existe des fonctions $u_i : X_i \mapsto \mathbb{R}$ et pour tout $j \in J$, ensemble de parties de $1, ..., n$, il existe $\pi_j / in \mathbb{R}$ tels que $u(x_1, ..., x_n) = \sum_{j \in J} \pi_J \prod_{k \in J} u_k(x_k)$.
— la structure décomposable : il existe des fonctions $u_i : X_i \mapsto \mathbb{R}$ et une fonction $F : \mathbb{R}^n \mapsto \mathbb{R}$ telles que $u(x_1, ..., x_n) = F(u_1(x_1), ..., u_n(x_n))$.
— la décomposition additive non transitive : il existe des fonctions $v_i : X_i \times X_i \mapsto \mathbb{R}$ telles que $x \succsim y \Leftrightarrow \sum_{i=1}^{n} v_i(x_i, y_i)$.

Plusieurs méthodes sont issues de la méthode américaine comme par exemple :

— le processus analytique de hiérarchie (*Analytic Hierarchy Process* - AHP) [64] qui utilise une structure hiérarchique et sépare le but, les critères et les alternatives possibles pour chaque critère.
— la théorie de l'utilité multi-attributs (*Multi Attribute Utility Theory* MAUT) [42] qui se base sur les fonctions d'utilité.
— la somme pondérée ordonnée (*Ordered Weighted Averaging* OWA) [38], un opérateur d'agrégation qui permet de modéliser de nombreux types de fonction comme le minimum, le maximum ou la moyenne.

3.3.2 L'école européenne

Alors que l'approche américaine compare les scores des fonctions d'utilité multi-attributs de chaque objet après avoir agrégé les fonctions mono-attribut, l'approche européenne adopte une méthode différente et commence par comparer les critères des objets pour ensuite les agréger. Ces méthodes sont inspirées de la méthode de Condorcet [28] qui dit :

"Méthode de Condorcet.- Un candidat a est préféré à un candidat b si et seulement si le nombre de votants ayant classé a devant b est strictement supérieur au nombre de votants ayant classé b devant a (en cas d'égalité les deux candidats sont jugés indifférents)." [12]

Prenons un exemple ou des personnes doivent voter en classant les trois candidats :
— 19 votent pour $a > b > c$
— 15 votent pour $b > a > c$
— 10 votent pour $c > b > a$

Avec une méthode de vote classique, a remporte le vote avec 19 voix devant b avec 15 voix et c avec 10 voix soit un classement final a>b>c.

En utilisant la méthode de Condorcet donc en comparant paire par paire, on arrive aux résultats suivants :
— 19 votes pour $a > b$ et 25 pour $b > a$
— 34 votes pour $a > c$ et 10 votes pour $c > a$
— 34 votes pour $b > c$ et 10 votes pour $c > b$

Avec cette méthode, on arrive à un résultat différent et un classement final $b > a > c$.

Ces méthodes où a est préféré à b lorsque l'ensemble des critères plaidant en faveur de a est plus important que l'ensemble des critères plaidant en faveur de b sont appelées des méthodes de surclassement [63]. Dans ce cas là, on dit que a surclasse b (aSb). Ces méthodes n'utilisent plus des scores comme celles de l'approche américaine mais avec des préférences. Ces préférences n'ont donc plus à être numériquement représentables pour être comparées. Par contre il se peut que deux objets soient incomparables. Les méthodes les plus connues issues de l'approche européenne sont les différentes versions de la méthode ELECTRE fondée par Bernard Roy [61] [62]

Les deux écoles utilisent des techniques d'agrégation dans leurs méthodes. Nous allons maintenant présenter plusieurs opérateurs d'agrégation utilisés par l'école américaine ou

européenne.

3.4 Opérateurs d'agrégation

Quelle que soit l'approche utilisée, l'utilisation d'une méthode multi-attributs nécessite un opérateur d'agrégation afin d'arriver à l'objectif de l'aide à la décision : proposer une ou plusieurs solutions au décideur afin qu'il puisse prendre sa décision. Nous allons présenter plusieurs opérateurs dans cette section en étudiant les deux types d'opérateurs existants : les opérateurs avec indépendance entre les critères et ensuite les opérateurs prenant en compte les interactions entre les critères.

3.4.1 Opérateurs d'agrégation avec indépendance entre les critères

Pendant très longtemps en aide à la décision multicritère, les critères ont été considérés indépendants les uns des autres. Par exemple, pour connaître la valeur d'un élève, on va calculer sa moyenne. Pour cela, il suffit d'ajouter ses notes, éventuellement de les pondérer par un coefficient et de diviser le nombre obtenu par le nombre de notes. On aura alors agréger les notes en un score : la moyenne de l'élève. Dans ce cas là, il n'y a pas de lien entre les notes donc elles sont indépendantes les unes des autres. Il est compliqué de détecter les liens entre les critères lorsqu'ils sont présents, c'est pourquoi de nombreux opérateurs d'agrégation considèrent les critères comme indépendants. Nous allons présenter les opérateurs les plus connus et utilisés, la somme pondérée et la somme pondérée ordonnée.

3.4.1.1 La somme pondérée

La somme pondérée est un opérateur d'agrégation souvent utilisé pour sa simplicité de mise en place. Elle est définie par :

$$\psi(a_1, ..., a_n) = \sum_{i=1}^{n} w_i a_i \tag{3.2}$$

avec $w_i \in [0, 1]$ des poids des critères tels que

$$\sum_{i=1}^{n} w_i = 1 \tag{3.3}$$

Mais la moyenne pondérée est un opérateur d'agrégation qui reste très limité et qui peut ne pas convenir pour modéliser les préférences d'un décideur. Prenons un exemple pour expliquer cette limite.

Soit trois objets a, b, c chacun décrit par deux critères. Les fonctions d'utilité de ces critères sont u_1 et u_2 et leurs valeurs sont :

$$u_1(a) = 0.3 \quad u_1(b) = 0 \quad u_1(c) = 1$$

$$u_2(a) = 0.3 \quad u_2(b) = 1 \quad u_2(c) = 0$$

Maintenant supposons que le décideur préfère un objet dont tous les critères le satisfassent moyennement plutôt qu'un objet ayant un critère ne le satisfaisant pas du tout. Appliqué à nos trois objets, nous arrivons au classement $a \succ b \sim c$. Établissons maintenant les équations permettant de trouver les poids w_1 et w_2 :

$$b \sim c \Leftrightarrow w_2 = w_1$$

$$a \succ b \Leftrightarrow 0,3(w_1 + w_2) > w_2$$

Étant donné que $w_1 = w_2$, cette dernière équation équivaut à $0.6w_2 > w_2$, ce qui est impossible. La moyenne pondérée ne permet donc pas de modéliser ce type de préférences. Afin de modéliser plus de fonction, un autre opérateur peut être utilisé : la somme pondérée ordonnée.

3.4.1.2 La somme pondérée ordonnée

Déterminer le poids associé à chaque critère est une étape complexe de la construction d'un opérateur d'agrégation. Par exemple, un extrême peut être d'affecter au critère ayant la valeur la plus importante le poids $w_1 = 1$ et les poids $w_2 = 0$ à tous les autres critères. Cela revient à la fonction "Max" en ne considérant que le critère le plus satisfaisant. L'autre extrême est de ne considérer que le critère le moins satisfaisant en lui affectant la valeur w_1 et w_2 à tous les autres. Ce qui revient à la fonction "Min" et donc à donner une importance via un poids à chaque critère.

La somme pondérée ordonnée (*Ordered Weighted Averaging* - OWA) est une classe d'opérateurs d'agrégation introduit par Yager [81] défini par :

$$OWA_w(a_1, ..., a_n) = \sum_{i=1}^{n} w_i a_{(i)} \tag{3.4}$$

avec $W = (w_1, ..., w_n)$ un vecteur de poids, $w_i \in [0, 1]$ tel que $\sum_{i=1}^{n} w_i = n$ et où la notation (.) est une permutation des indices telle que $a_{(1)} \leq ... \leq a_{(n)}$.

Un opérateur d'agrégation appartement à la classe des OWA permet d'affecter un poids à un critère non pas selon sa nature mais selon son niveau de satisfaction. Il est ainsi possible de modéliser un grand nombre de fonction de base comme par exemple :

— la fonction "Max" avec un vecteur $W = (1, 0, ..., 0)$
— la fonction "Min" avec un vecteur $W = (0, ..., 0, 1)$
— la fonction moyenne avec pour n critères un vecteur $W = (\frac{1}{n}, ..., \frac{1}{n})$

Mais il est aussi possible de construire des opérateurs d'agrégation basés sur OWA plus complexe comme par exemple :

— un opérateur ne prenant en compte que les x critères les plus satisfaisants avec un vecteur W où les x premiers poids auront une valeur de $\frac{1}{x}$ et les autres poids 0. Par exemple, une compétition par équipe en golf se joue à six joueurs mais seulement le

score des cinq premiers est pris en compte. Dans ce cas là, le score des cinq premiers joueurs aura un poids de $\frac{1}{5}$ et le dernier aura un poids nul.

— un opérateur où plus les critères seront satisfaisants, plus leur poids sera important. Dans ce cas là, les n éléments du vecteur W respecteront $w1 \geq w_2 \geq ... \geq w_n$. Une fonction de ce style peut être utilisée pour donner plus de poids à un critère ayant une bonne performance. Par exemple, un élève est noté sur trois critères : sa note en math, sa note en français et sa note en sport. Plutôt que de donner un coefficient selon la matière, le coefficient peut être donné selon la note. La meilleure note comptera pour 50% de la moyenne, la 2ème pour 30% et la plus mauvaise pour 10%.

Lorsqu'une personne est confrontée à une prise de décision, plusieurs critères sont impliqués. Il est rare que ces critères soient indépendants les uns des autres. Reprenons l'exemple que nous avions cité plus tôt dans ce chapitre où une personne voulait acheter une voiture puissante mais écologique. Plus une voiture est puissante, plus elle pollue et inversement. Il y a donc une interaction entre ces deux critères et c'est souvent le cas dans des situations de la vie réelle. C'est pourquoi nous allons maintenant présenter des opérateurs d'agrégation avec interactions entre les critères.

3.4.2 Opérateurs d'agrégation avec interactions entre les critères

Construire des opérateurs d'agrégation en considérant les critères indépendants les uns des autres est plus simple car il n'est pas toujours évident de connaître les dépendances ou les liens entre les critères. Mais cela ne permet pas de modéliser toutes les réalités car dans les situations réelles, les critères présentent des interactions entre eux et ne pas considérer ces interactions revient à perdre de l'information qui peut être utile pour arriver à de meilleurs résultats pour l'aide à la décision. Nous allons, dans les sections suivantes, présenter des méthodes d'agrégation prenant en compte ces interactions entre les critères.

3.4.2.1 Mesures floues

Les fonctions comme la somme pondérée ou OWA ne sont pas capables de modéliser les interactions entre les critères. Hors, il est très rare qu'il n'y ait pas de synergie entre les critères. Afin d'obtenir une représentation plus proche de la réalité, ces interactions doivent être prises en compte et pour cela, l'utilisation d'un vecteur poids n'est pas appropriée. Remplacer ce vecteur poids par une fonction non-additive permet en plus de définir un poids pour chaque critère, de définir aussi un poids pour chaque sous-ensemble de critères. L'utilisation de ces fonctions non-additives permet davantage d'exprimer la subjectivité humaine et de modéliser les synergies positives ou négatives entre les critères. Sugeno a utilisé ces fonctions non additives et a proposé de les appeler mesures floues [74].

Définition. *Une mesure floue sur N est une fonction d'ensemble* $\mu : 2^N \rightarrow [0, 1]$ *qui est monotone, c'est à dire* $\mu(S) \leq \mu(T)$ *chaque fois que* $S \subseteq T$*, et vérifie les conditions limites* $\mu(\emptyset) = 0$ *et* $\mu(N) = 1$

$\mu(S)$ est considéré comme étant le poids ou l'importance du sous-ensemble de critère S. Compte tenu la monotonie des fonctions floues, si on ajoute un élément i à l'ensemble

S, le poids de $S + i$ ne pourra être plus faible que le poids de S. Utiliser les mesures floues nécessite de déterminer 2^N poids correspondants aux 2^N ensembles de N. Dans le cas où une mesure floue est additive, c'est à dire si $\mu(S \cup T) = \mu(S) + \mu(T)$, seuls les poids des n critères sont suffisants pour calculer la mesure floue.

Les mesures floues peuvent être utilisées comme opérateurs d'agrégation. En utilisant les poids des critères et des ensembles de critères, il est possible de représenter les interactions entre les critères. Ces fonctions sont appelées des intégrales floues [74]. Il existe plusieurs classes d'intégrales floues, nous allons présenter dans la prochaine section une des plus représentatives, l'intégrale de Choquet.

3.4.2.2 Intégrale de Choquet

L'intégrale de Choquet a été introduite par le mathématicien français Gustave Choquet [20].

Définition. *Soit $\mu \in \mathcal{F}_N$. L'intégrale de Choquet de $x \in \mathbb{R}^n$ par rapport à μ est définie par :*

$$\mathcal{C}_\mu(x) := \sum_{i=1}^{n} x_{(i)}[\mu(A_{(i)}) - \mu(A_{(i+1)})] \tag{3.5}$$

où (.) indique une permutation sur N telle que $x_{(1)} \leq ... \leq x_{(n)}$. D'autre part, $A_{(i)} = (i), ..., (n)$ et $A_{(n+1)} = \emptyset$.

L'utilisation des mesures floues dans l'intégrale de Choquet permet de comprendre les dépendances entre les critères mais aussi l'importance que peut avoir chaque critère. Pour comprendre cette notion d'importance de critères, prenons un exemple avec trois critères avec les valeurs de μ suivantes :

A	c_1	c_2	c_3
$\mu(A)$	0	0.2	0.2
A	c_{12}	c_{13}	c_{23}
$\mu(A)$	0.8	0.8	0.4

En ne considérant que la première ligne de ce tableau, on pourrait penser que le critère c_1 n'est pas utile étant donné que $\mu(c_1) = 0$. Mais on peut voir que lorsque ce critère est inclus dans un groupe de critères, il influe sur la valeur de la mesure floue du groupe. Ainsi $c_{12} > c_1 + c_2$ et $c_{13} > c_1 + c_3$. Alors qu'il n'est pas possible de le déceler avec une fonction additive, dans cet exemple le critère 1 est très important lorsqu'il est lié à d'autres. Shapley [66] a proposé en 1953 la définition d'un indice d'importance. Cet indice d'importance, aussi appelé valeur de Shapley du critère i par rapport à v est défini par :

$$\phi(\mu, i) := \sum_{T \subseteq N \setminus i} \frac{(n - t - 1)!t!}{n!}[\mu(T \cup i) - \mu(T)] \tag{3.6}$$

Cet indice à été proposé par Murofuhi [54] pour être utilisé en aide à la décision multi-critère. Dans le cas où μ est additif, nous avons $\mu(T \cup i) - \mu(T) = \mu(i)$. Dans le cas contraire, cette égalité n'est plus vraie et les critères sont alors dépendants.

Il est alors intéressant de connaître le degré d'interaction entre deux critères. Pour cela, Murofushi et Soneda ont proposé en 1993 un indice d'interaction permettant de donner une valeur moyenne à une interaction entre deux critères [55]. Cet indice d'interaction entre les critères i et j par rapport à μ est défini par :

$$I(\mu, ij) = \sum_{T \subseteq N \setminus ij} \frac{(n-t-2)!t!}{(n-1)!} (\Delta_{ij}\mu)(T) \tag{3.7}$$

avec $(\Delta_{ij}\mu)(T) := \mu(T \cup ij) - \mu(T \cup i) - \mu(T \cup j) + \mu(T)$.

L'indice d'interaction $I(\mu, ij)$ est compris dans l'intervalle $[-1, 1]$ pour tout $i, j \in N$. Si cet indice est positif, alors il y a une synergie entre ces deux critères. A l'inverse si l'indice d'interaction est négatif, on parle alors de critères redondants.

En reprenant l'exemple précédant avec ces deux coefficients, cela donne pour l'indice de Shapley :

$$\phi(1) = \frac{1}{3} * 0 + \frac{1}{6} * 0,6 + \frac{1}{6} * 0,6 + \frac{1}{3} * 0,3 = 0,4 \tag{3.8}$$

$$\phi(1) = \frac{1}{3} * 0,2 + \frac{1}{6} * 0,8 + \frac{1}{6} * 0,2 + \frac{1}{3} * 0,2 = 0,3 \tag{3.9}$$

$$\phi(1) = \frac{1}{3} * 0,2 + \frac{1}{6} * 0,8 + \frac{1}{6} * 0,2 + \frac{1}{3} * 0,2 = 0,3 \tag{3.10}$$

Alors qu'à première vue on aurait pu penser que les critères 2 et 3 étaient les plus importants, il se trouve que c'est le critère 1 le plus important. Les mesures floues sur les groupes de critères montre l'importance du critère 1. Cela se reflète aussi sur l'indice d'interaction. Ainsi les indices $I_{12} = I_{13} = 0.2$ montrent une synergie entre le critère 1 et les deux autres. Au contraire, l'indice $I_{23} = -0.3$ indique que les critères 2 et 3 sont redondants.

Nous avons présenté les systèmes interactifs d'aide à la décision et certaines des techniques qu'ils utilisent. Un SIAD est là pour aider l'utilisateur à prendre des décisions. Dans notre cas, il est là pour aider l'utilisateur à écrire des règles d'autorisation pour autoriser ou non la divulgation de données de vie privée. L'utilisateur n'étant pas capable d'écrire seul ces règles d'autorisation, il faut lui faire des propositions de règles qu'il peut comprendre et accepter si elles lui conviennent. Pour cela, nous avons besoin d'un système qui va apprendre les préférences de l'utilisateur et lui proposer des règles selon ses préférences. Ce type de système que nous allons présenter est appelé un système de recommandation.

3.5 Systèmes de recommandation

Les systèmes de recommandation sont utilisés par les système interactifs d'aide à la décision pour prendre en compte des préférences évolutives. Alors qu'un utilisateur n'a plus aujourd'hui la charge cognitive suffisante pour traiter toute l'information qui se présente à lui, un système de recommandation peut analyser toute cette information. Ce genre de système utilise les préférences de l'utilisateur représenté par un profil pour filtrer et ordonner les informations et en extraire les plus pertinentes qu'il présentera à l'utilisateur.

3.5.1 Profil utilisateur

Dans un système d'aide à la décision, l'utilisateur est défini par l'ensemble de ses préférences et par des données récoltées par le système. Ces données sont stockées dans ce que l'on appelle un profil utilisateur qui peut contenir :

— des données personnelles de l'utilisateur comme son nom, son sexe, son adresse.
— les préférences de l'utilisateur par rapport à un problème de décision.
— un historique des actions de l'utilisateur sur le système ou pour un site internet l'historique des pages visitées.

Ce profil permet de regrouper toutes les informations connues au sujet de l'utilisateur mais peut aussi avoir d'autres fonctions. Un utilisateur peut par exemple exporter son profil contenant toutes ses préférences d'un périphérique à un autre pour éviter que ce nouveau périphérique n'ait à repartir de zéro en réapprenant toutes les préférences de l'utilisateur.

3.5.2 Apprentissage automatique

Connaître les préférences d'un utilisateur ou d'un décideur est essentiel pour proposer des solutions à ses problèmes. Afin de lui faire de bonnes recommandations, un système d'aide à la décision doit continuellement apprendre ses préférences. Pour cela, on peut faire appel aux nombreuses techniques d'apprentissage automatique. Généralement, ces différentes techniques sont regroupées en trois grandes familles que nous allons présenter : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement [22].

3.5.2.1 Apprentissage supervisé

L'apprentissage supervisé est un ensemble de méthodes utilisant un oracle pour aider le système apprenant. L'oracle étiquette les données grâce à une fonction inconnue du système apprenant. Ces étiquettes sont un ensemble de couples entrée-sortie et constituent une base de données d'apprentissage. Le but du système apprenant est alors de se servir de cette base de données pour se rapprocher le plus possible de la fonction de l'oracle. Il est donc possible d'identifier deux phases distinctes :

— la phase d'entraînement où le système apprenant utilise la base de données d'apprentissage de l'oracle et essaie de déterminer une fonction permettant de prédire les étiquettes données par l'oracle.
— la phase de classification ou le système apprenant étiquette des données vierges en fonction de ce qu'il a appris lors de la première étape.

Ces deux phases peuvent être exécutées de façon consécutive, la phase d'entraînement puis la phase de classification. Dans ce cas, on parle d'apprentissage supervisé hors ligne. Mais les deux phases peuvent aussi être exécutées de manière itérative où le système apprenant reviendra à la phase d'entraînement après avoir commencé celle de classification. Cela

permet d'affiner l'apprentissage et d'améliorer la base de données d'apprentissage avec le temps. Dans ce cas là, on parle d'apprentissage supervisé en ligne.

L'apprentissage supervisé permet de résoudre deux types de problèmes. Les problèmes de régression où la valeur de sortie que le système apprenant cherche à estimer est une valeur dans un ensemble continu de réels et les problèmes de classification où dans ce cas, l'ensemble de valeurs de sortie est fini. Il existe un grand nombre de méthodes utilisant l'apprentissage supervisé comme la méthodes des k plus proches voisins, les réseaux de neurones ou les arbres de décision[22].

L'utilisation d'un oracle et d'une phase d'entraînement est incompatible avec notre problème. Nous devons obligatoirement prendre l'avis de l'utilisateur pour apprendre au mieux ses préférences. De plus le comportement de cet utilisateur n'est pas fixe donc étiqueter les entrées-sorties pourrait donner un résultat aléatoire. Le couple (demande d'accès à une ressource, décision) pour une ressource ne sera pas forcément identique à chaque fois selon la situation et apprendre tous les couples possibles selon toutes les situations envisageables serait une perte de temps. Or pour que notre système ait une utilité, nous devons minimiser l'effort de l'utilisateur. Nous allons maintenant présenter l'apprentissage non supervisé dont les techniques n'utilisent pas d'oracle pendant l'apprentissage.

3.5.2.2 Apprentissage non supervisé

Contrairement aux méthodes d'apprentissage supervisé, celles d'apprentissage non supervisé n'utilisent pas d'oracle pour les assister lors de l'apprentissage. Le but du système apprenant est alors de découvrir des catégories et de trouver les règles servant à catégoriser les données. L'apprentissage non supervisé ne comporte qu'une seule tâche : le regroupement de données similaires dans des groupes homogènes. La difficulté consiste à reconnaître la structure d'une donnée et la placer dans le bon groupe sans avoir dans ce cas d'étiquette sur la donnée. Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé n'a pas besoin de connaissance préalable sur les sorties. Les méthodes les plus connues utilisant l'apprentissage non supervisé sont l'algorithme des k-moyennes et l'analyse en composantes.

Le problème de l'apprentissage non supervisé est qu'il fonctionne généralement en mode hors ligne. L'apprentissage est fixe et ne peut pas s'adapter au fil du temps. C'est évidemment contraire à ce que nous voulons faire car l'utilisateur de notre système doit pouvoir changer de comportement et le système doit pouvoir s'adapter à ce changement. L'apprentissage par renforcement permet cet adaptation dans le temps.

3.5.2.3 Apprentissage par renforcement

L'apprentissage par renforcement regroupe des méthodes qui se servent des interactions entre un agent et son environnement pour apprendre le comportement à adopter selon la situation. A chaque action effectuée, une récompense qui peut être négative ou positive va être donnée à l'agent. Grâce à ces récompenses, l'agent va apprendre quelles actions sont les plus bénéfiques et donc quel est le comportement le plus satisfaisant. C'est un apprentissage en ligne où le système va affiner ses connaissances au cours du temps et des interactions.

Barto [8] propose la définition suivante de l'apprentissage par renforcement : *L'apprentissage par renforcement est une approche informatique de l'apprentissage dans laquelle un agent essaie de maximiser le montant total de la récompense qu'il reçoit en interagissant avec un environnement complexe et incertain.*

Ce type d'apprentissage correspond bien à ce que nous voulons faire. Dans notre cas, le système va interagir avec l'utilisateur pour connaître l'action la plus appropriée lorsqu'une entité demande l'accès à une ressource. La décision de l'utilisateur va fournir un retour au système qui va pouvoir apprendre les préférences de l'utilisateur. Le système pourra ensuite utiliser ces préférences pour trouver le comportement qui convient à l'utilisateur et donc la ou les règles d'autorisation qui lui conviennent le mieux.

3.5.3 Échelle de représentation des préférences

La modélisation des préférences est un aspect important de l'aide à la décision multicritère. Une mauvaise modélisation peut entraîner des erreurs de compréhension du comportement de l'utilisateur et amener à lui proposer des solutions qui ne seront pas pertinentes.

Dans notre cas, nous voulons savoir si une demande d'accès à une ressource doit être divulguée ou non. Pour cela nous faisons une analyse multicritère sur cette demande d'accès en utilisant les différentes informations que le système reçoit comme critères (qui demande, sur quelle ressource, quand, etc.). Pour cela nous devons agréger les utilités de chaque critère pour obtenir une seule valeur d'utilité : le score associé à la demande. Il est donc nécessaire de représenter les valeurs des critères sous forme numérique. Mais selon le système et les critères sur lesquels on travaille, plusieurs échelles de représentation sont envisageables.

Des psychologues ont montré que *l'affect* jouait un rôle important dans la prise de décision [70]. L'affect peut être défini comme la qualité spécifique de "bon" ou de "mauvais" ressentie consciemment ou non et qui permet de dégager le côté positif ou négatif d'un stimulus. Pour représenter cette dualité, l'utilisation d'une échelle bipolaire parait évidente. Une échelle bipolaire utilise un axe unique avec deux pôles pour représenter les extrêmes négatif et positif et un point neutre ni positif ni négatif. La figure 3.2 est un exemple d'échelle bipolaire. Le phénomène physique du chaud et du froid est un exemple parfait de concept bipolaire car le froid est l'absence de chaud. Ainsi plus il fait froid, moins le chaud est présent et inversement.

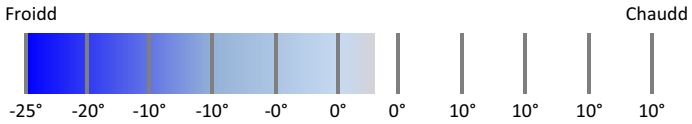


Figure 3.2 — Exemple d'échelle bipolaire

Mais cette échelle bipolaire ne convient pas pour représenter toutes les préférences. Certains événements peuvent nous toucher positivement sans pour autant nous affecter négativement. Prenons le cas d'une personne ayant la phobie d'un animal. Elle a donc un fort ressenti négatif pour l'animal et sûrement très peu voire aucun ressenti positif. Habituer la personne à être en contact avec l'animal peut l'amener à avoir moins peur. Contrairement au

chaud et froid, la baisse des sentiments négatifs n'entraîne pas nécessairement une hausse des sentiments positifs et l'utilisation d'une échelle bipolaire n'est alors pas possible. Afin de pouvoir obtenir une modélisation prenant en compte les effets négatifs et positifs non bipolaires, des psychologues ont proposé l'utilisation de deux échelles unipolaires séparées [14]. Ces échelles (figure 3.3) permettent de différencier les aspects positifs et les aspects négatifs d'un critère.

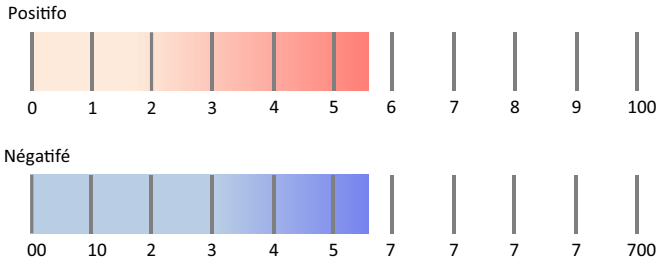


Figure 3.3 — Exemple d'échelles unipolaires séparées

3.5.4 Les différents types de systèmes de recommandation

Il existe plusieurs façons de faire des recommandations à l'utilisateur. Nous allons présenter dans les sections suivantes les trois méthodes utilisées en aide à la décision, la recommandation sur le contenu qui elle se base sur les caractéristiques des objets, la recommandation collaborative qui se base sur l'ensemble des utilisateurs et enfin la recommandation hybride qui comme son nom l'indique mixe les deux premières méthodes.

3.5.4.1 Recommandation sur le contenu

Cette méthode basée sur le contenu n'utilise que les caractéristiques des objets pour faire des recommandations [2]. Toutes les informations disponibles sur un objet peuvent être utilisées pour décrire cet objet. Par exemple, un livre peut être décrit par plusieurs caractéristiques : le titre, l'auteur, le genre, l'année de parution, etc. Pour faire ses recommandations, le système compare les objets pour trouver ceux qui se rapprochent des préférences de l'utilisateur.

Ce type de système de recommandation est très intéressant lorsque l'on se trouve sur un système où les objets sont très détaillés. Un grand avantage se retrouve lorsqu'un nouvel objet est ajouté dans le système, si ses caractéristiques sont proches des préférences de certains utilisateurs, il pourra être recommandé très rapidement. De plus, en utilisant cette méthode, les recommandations faites à l'utilisateur seront toujours très proches de ses préférences, les recommandations seront donc toujours pertinentes, tant qu'il ne change pas son comportement. Par exemple, Martin et al. [49] ont utilisé un système de recommandation basé sur le contenu pour apporter une aide au choix d'un prestataire pour un utilisateur en fonction de son profil.

Mais cette méthode a aussi des inconvénients. Pour faire des recommandations par rapport aux préférences de l'utilisateur, ces dernières doivent être connues du système. Ainsi

pendant le temps d'apprentissage des préférences de l'utilisateur, le système ne sera pas capable de faire des recommandations ou bien celles ci seront peu pertinentes. Aussi lorsqu'un utilisateur a un comportement similaire pendant une longue période et qu'il change brutalement de préférences, le système va accuser une certaine latence avant de comprendre le changement de comportement de l'utilisateur.

Figure 3.4 — Eureka : un système de recommandation basé sur le contenu

La figure 3.4 illustre un exemple de système de recommandation basé sur le contenu. Eureka est un système disponible sur les chaînes de télévision CanalSat [1]. Il analyse les programmes regardés par l'utilisateur pour savoir quel type de programme il apprécie. Ensuite il propose à l'utilisateur une liste de programmes pour les trois prochains jours basée sur ce qu'il a appris.

3.5.4.2 Recommandation collaborative

La recommandation collaborative, aussi appelée collaboration sociale ou filtrage collaboratif utilise les préférences de tous les utilisateurs disponibles pour faire des propositions [2]. L'idée de base de cette méthode est que si un utilisateur a des goûts similaires à d'autres utilisateurs, alors il devrait apprécier les objets choisis par ces autres utilisateurs. Le système peut donc lui recommander ces objets s'il ne les a pas encore choisis. La recommandation collaborative peut être implémentée en utilisant par exemple l'algorithme des K plus proches voisins.

Contrairement à la recommandation basée sur le contenu, le système n'a pas besoin d'avoir beaucoup d'informations pour proposer des objets à l'utilisateur. Il va très rapidement chercher les utilisateurs ayant un profil similaire et trouver des objets qui peuvent l'intéresser. De plus la recommandation collaborative est très utile sur des bases d'objets qui ne sont pas décrits par leurs caractéristiques ou lorsqu'il est difficile de décrire ces objets comme par exemple des opinions.

Mais la recommandation collaborative comporte aussi des inconvénients. Dans le cas d'un système avec peu d'utilisateurs ou si l'utilisateur a des préférences atypiques, il se peut qu'il n'y ait pas d'utilisateur ayant un comportement proche, dans ce cas là les recommandations pourraient ne pas être pertinentes. De la même façon, lorsqu'un nouvel objet arrive dans le système, étant donné qu'il n'aura été choisi par personne, il ne pourra pas être recommandé aux autres utilisateurs. A chaque fois qu'un de ces utilisateurs fait un choix, il faut mettre à jour le processus de recommandation car de nouvelles informations sont disponibles. Cette mise à jour demande beaucoup de ressources si le système a beaucoup d'utilisateurs et il y aura une latence entre les actions des utilisateurs et les nouvelles propositions.

De très grands systèmes utilisent la recommandation collaborative. Par exemple Twit-

1. http ://www.canalsat.fr/pid1358-guide-tv-presentation-eureka.html

ter [2] suggère à ses utilisateurs une liste de personnes à suivre. Cette liste est composée de personnes populaires suivis par des personnes que l'utilisateur suit (figure 3.5). Twitter utilise aussi le filtrage collaboratif pour suggérer en temps réel les thèmes populaires sur la plateforme. Chaque utilisateur peut insérer dans ses tweets des hashtags pour déclarer le thème de son tweet. Twitter rassemble ensuite les thèmes les plus populaires selon un critère de localisation (ville, pays ou mondial) et les présente à l'utilisateur (figure 3.5).

Figure 3.5 — Twitter : suggestions de personnes et de tendances

3.5.4.3 Recommandation hybride

La recommandation hybride est une combinaison de recommandation basée sur le contenu et de recommandation collaborative. Le but étant d'éliminer les inconvénients des deux approches. Il existe plusieurs méthodes pour combiner les deux approches :

— implémenter un système de recommandation à base de contenu et un système de recommandation collaborative et combiner les recommandations faites par les deux systèmes.
— prendre certaines des fonctionnalités d'un système de recommandation à base de contenu et les intégrer dans un système de recommandation collaborative.
— prendre certaines fonctionnalités d'un système de recommandation collaborative et les intégrer dans un système de recommandation à base de contenu.
— Construire un modèle combinant à la base les fonctionnalités des deux méthodes.

Le système de recommandation hybride le plus connu est celui utilisé par Amazon [3] [47]. Pour faire ses recommandations, l'algorithme d'Amazon commence par créer une liste de recommandations à partir des objets similaires au objets achetés ou notés par l'utilisateur, ce qui correspond à l'approche recommandation à base de contenu. Ensuite pour trouver les éléments de cette liste les plus pertinents, l'algorithme d'Amazon crée une table en associant tous les objets que les autres clients ont tendance à acheter en même temps. Ainsi Amazon est capable de présenter à l'utilisateur des objets qui pourrait l'intéresser dès qu'un client visite la page d'un objet ou qu'il ajoute un objet à son panier virtuel (figure 3.6).

2. https ://twitter.com/
3. http ://www.amazon.fr/

Figure 3.6 — Recommandation du site web Amazon

3.6 Conclusion

Ce chapitre a présenté des éléments issus de l'aide à la décision et différentes techniques que ce domaine regroupe. Notre but est d'apprendre comment se comporte un utilisateur quant il s'agit de divulguer ou non ses données de vie privée puis de l'aider à écrire sa politique d'autorisation en lui proposant des règles en fonction des préférences que le système a apprises. Le système que nous voulons mettre en place correspond donc totalement à un système de recommandation.

Bien que l'approche hybride aurait pu être utilisée, cela aurait nécessité de centraliser les préférences des utilisateurs. Or comme nous l'avons mis en avant dans le chapitre sur la protection de la vie privée, cette centralisation peut poser des problèmes liés à la confidentialité des informations. Nous avons donc choisi de rester sur une approche basée sur le contenu où les données de l'utilisateur resteront en local.

4 L'architecture détaillée de Kapuer

« "Kapuer" est une interjection prononcée par un bébé de 6 mois et demi à la vue d'une boîte de mélange à purée »

François Perusse

4.1 Introduction

Kapuer est une combinaison entre un système d'aide à la décision et un système de contrôle d'accès. Ce chapitre présente l'architecture de Kapuer et le détail des différentes étapes nécessaires à l'apprentissage des préférences et comment le système arrive à proposer des règles de sécurité à l'utilisateur. Les différents opérateurs d'agrégation sont décrits, y compris Kagop, l'opérateur d'agrégation que nous avons développé pour Kapuer.

4.2 Modélisation des préférences

4.2.1 Les critères

Un critère représente l'élément de base constituant une requête d'accès à une ressource protégée. Le critère peut correspondre au nom de l'utilisateur, son age, le nom de la ressource, le nom de l'action, etc. Ainsi, *"Jacqueline veut lire le calendrier"* comporte trois critères *Jacqueline*, *lire* et *calendrier*. L'ensemble des critères du système est noté CR. Un critère est composé d'un identifiant et de deux valeurs correspondant aux préférences de l'utilisateur. Nous utilisons deux valeurs de préférences car si les préférences avaient été modélisées une seule valeur par critère, il aurait été compliqué d'interpréter un critère ayant une valeur faible. Cette faible valeur voudrait-elle dire que pour l'utilisateur, ce critère n'est pas un critère favorable à la divulgation ou bien que le système n'a pas encore eu le temps d'apprendre

les préférences de l'utilisateur concernant ce critère. Il n'est pas possible de répondre à cette question. C'est pour cela que nous utilisons des échelles unipolaires séparées et deux valeurs pour chaque critère :

— La première, $g^t : CR \rightarrow [0, \infty[$ représente la préférence de l'utilisateur pour la divulgation d'un critère à l'instant t.
— La deuxième $f^t : CR \rightarrow [0, \infty[$ représente la préférence de l'utilisateur pour la non-divulgation d'un critère à l'instant t.

Ces deux valeurs ne sont pas indépendantes. Nous avons choisi d'utiliser une méthode de calcul de score incrémentale uniquement, donc, lors de leur mise à jour, $g^t(x)$ et $f^t(x)$ ne peuvent qu'augmenter. L'apprentissage des préférences se faisant en continu, une valeur élevée de $g^t(x)$ ne veut pas forcément dire que lorsque le critère x est présent, l'utilisateur est fortement enclin à divulguer ses données. Cela peut aussi vouloir dire que ce critère est souvent apparu parmi les requêtes et qu'il a été mis à jour de multiples fois. Pour identifier la signification d'un critère, il faut calculer soit :

— le score $s^t_D(x)$ correspondant au poids du critère x à l'instant t en faveur de la divulgation. Ce score est issu de la différence entre la valeur de divulgation et la valeur de non-divulgation :

$$s^t_D(x) = f^t(x) - g^t(x) \tag{4.1}$$

— le score $s^t_{nD}(x)$ correspondant au poids du critère x à l'instant t contre la divulgation. Ce score est issu de la différence entre la valeur de non-divulgation et la valeur de divulgation :

$$s^t_{nD}(x) = g^t(x) - f^t(x) \tag{4.2}$$

Calculés ainsi, les scores $s^t_D(x)$ et $s^t_{nD}(x)$ établissent la position du critère x dans les préférences de divulgation ou non de l'utilisateur. Un faible score reflète une absence claire de raisons justifiant la préférence pour l'une des deux actions. Au contraire, un score élevé correspond à l'existence de raisons claires infirmant une préférence stricte en faveur d'une des deux actions.

4.2.2 Les classes de critères

Les modèles de politique de contrôle d'accès proposent des éléments importants à prendre en compte dans les politiques tels que :

— la visibilité : Qui veut avoir accès à la ressource (un ami, un collègue, un inconnu, etc.)
— l'aspect temporel : Le moment de la protection (différents jours de la semaine, différentes heures de la journée, etc.)
— l'aspect spatial : Le lieu où se trouve l'utilisateur (chez lui, au travail, etc.)
— la rétention : Comment la ressource est stockée (combien de temps, qui y aura accès, etc.)

— l'intention : Pourquoi la ressource est stockée (à but philanthropique, pour être revendue, etc.)

Pour exprimer ces éléments, nous introduisons la notion de classe. Chaque critère fait partie d'une classe de critères par la relation $ACC \subseteq CR * C$ ou l'ensemble des classes de critères est noté C. Le système étant générique, les classes de critères ne sont pas fixées et n'importe quelle classe de critères peut être créée. Pour simplifier notre notation par la suite, nous définissons la fonction *class* qui renvoie l'ensemble des critères appartenant à une même classe :

$$class : C \rightarrow \mathcal{P}^{CR}$$

$$x \mapsto \{y \in CR | (y, C) \in ACC\} \tag{4.3}$$

4.2.3 Les méta-critères

Nous définissons la notion de *méta-critère* pour représenter les abstractions des modèles de politique de contrôle d'accès comme le rôle de RBAC [65], les vues/activités de OrBAC [4], les hiérarchies d'intentions de PRBAC [13], etc. Le préfixe méta, dans le vocabulaire scientifique, permet entre autre de désigner un niveau d'abstraction supérieur. Ici, le niveau 0 correspond à un critère, les niveaux strictement supérieurs à 0 correspondent à des méta-critères. Un méta-critère est un critère ayant un niveau d'abstraction supérieur à un ou plusieurs critères appartenant à la même classe. L'ensemble des méta-critères du système est noté MCR, cet ensemble est contenu dans CR mais les deux ensembles ne peuvent pas être égaux car nous considérons que le système doit contenir au moins un critère au niveau d'abstraction 0 et donc n'appartenant pas à MCR. Chaque critère est associé à un méta-critère. Un méta-critère permet de regrouper plusieurs critères partageant une caractéristique commune. Par exemple, considérons les critères "Jacqueline" et "Bernard". Ces deux critères ont une caractéristique commune : être des parents. On peut ainsi définir le méta-critère "Parent" regroupant les critères "Jacqueline" et "Bernard". Les valeurs$f^t(x)$ et $g^t(x)$ du méta-critère x lui sont propres. De même, le méta-critère "Famille" est un niveau d'abstraction supérieur à "Parent".

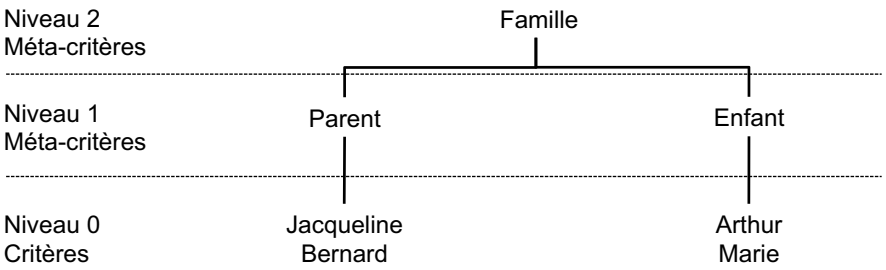


Figure 4.1 — Hiérarchie de critères

Un méta-critère étant aussi un critère, il est possible pour chaque classe de critères cl de créer une hiérarchie $H_{cl} \subseteq CR * MCR$ entre ces critères tel que $\forall (c_1, c_2) \in H_{cl}, class(c_1) =$

$class(c_2)$ (cf. Figure 4.1). Il existe deux cas où un critère ne possède pas dans sa description de méta-critère :

— Le critère est en haut de la hiérarchie (exemple le critère "Famille" dans la Figure 4.1),
— Le critère est indépendant et ne peut donc pas être associé à un méta-critère.

4.2.4 Les groupes de critères

Afin de pouvoir analyser les relations inter-critères, nous définissons le *groupe* de critères. Un groupe de critères est une association de n critères. Un groupe de critère a ses propres valeurs, indépendantes des valeurs des critères qui le composent. Les critères ou méta-critères composant un groupe de critères doivent appartenir à des classes de critères différentes. Ainsi prenons par exemple les critères "Parent" et "Calendrier" appartenant à deux classes différents, nous pouvons créer le groupe de critères {Parent, Calendrier}. L'ensemble des groupes de critères G est défini par :

L'ensemble G est compris dans l'ensemble des parties de CR :

$$G \subseteq \mathcal{P}(CR)$$

Un groupe de critère est composé de deux critères minimum.

$$\forall g \in G, |g| \geq 2$$

Deux critères d'un même groupe ne peuvent appartenir à la même classe de critère.

$$\forall g \in G, \forall (c_1, c_2) \in g \times g, c_1 \neq c_2 \Rightarrow class(c_1) \neq class(c_2)$$

4.2.5 Exemple de formalisation

La figure 4.2 illustre par l'exemple les différentes notions autour du critère.

En reprenant les exemples de la figure 4.2, nous obtenons :

L'ensemble des méta-critères MCR et l'ensemble des critères CR :

$$MCR = \{Fils, Parent, Famile, Donnée, Ressource, Toulouse, France\}.$$
$$CR = MCR \cup \{Max, Pierre, Adresse\ mail, Campus\}$$

La relation de hiérarchie entre un critère et un méta-critère H :

$$H = \{(Max, Fils), (Pierre, Parent), (Adresse\ mail, Donnée), (Campus, Toulouse), (Parent, Famille), (Donnée, Ressource), (Toulouse, France)\}.$$

L'ensemble des classes de critères C :

$$C = \{Qui, Quoi, Où\}.$$

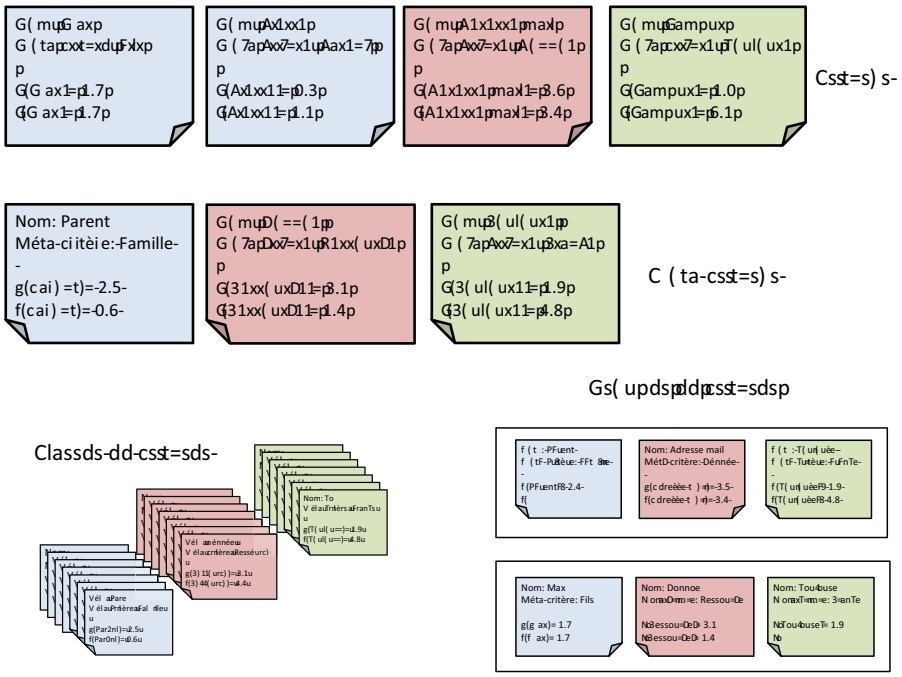


Figure 4.2 — Illustration des notions autour du critère

La relation entre un critère et une classe de critère ACC :

$$ACC = \{(Max, Qui), (Pierre, Qui), (Adresse\ mail, Quoi), (Campus, Où), (Fils, Qui), (Parent, Qui), (Donnée, Quoi), (Toulouse, Où), (Famille, Qui), (Ressource, Quoi), (France, Où)\}.$$

L'ensemble des groupes de critères G :

$$G = \{(Parent, Adresse\ \ mail, Toulouse), (Max, Donnée, Toulouse)\}.$$

A l'instant t les valeurs de divulgation g et de non divulgation f de chaque critère :

$$g^t(Max) = 1.7 \text{ et } f^t(Max) = 1.7$$
$$g^t(Pierre) = 2.3 \text{ et } f^t(Pierre) = 1.1$$
$$g^t(Adresse\ mail) = 3.5 \text{ et } f^t(Adresse\ mail) = 3.4$$
$$g^t(Campus) = 1.0 \text{ et } f^t(Campus) = 5.1$$
$$g^t(Parent) = 2. \text{ et } f^t(Parent) = 0.6$$
$$g^t(Donne) = 3.1 \text{ et } f^t(Donne) = 1.4$$
$$g^t(Toulouse) = 1.9 \text{ et } f^t(Toulouse) = 4.8$$

4.3 Le fonctionnement du système

La figure 4.3 représente l'architecture de Kapuer et met en évidence l'articulation entre le système d'aide à la décision et le système de contrôle d'accès. Cette architecture reprend les principaux composants définis par XACML [1] :

1. http ://www.oasis- open.org/committees/xacml/

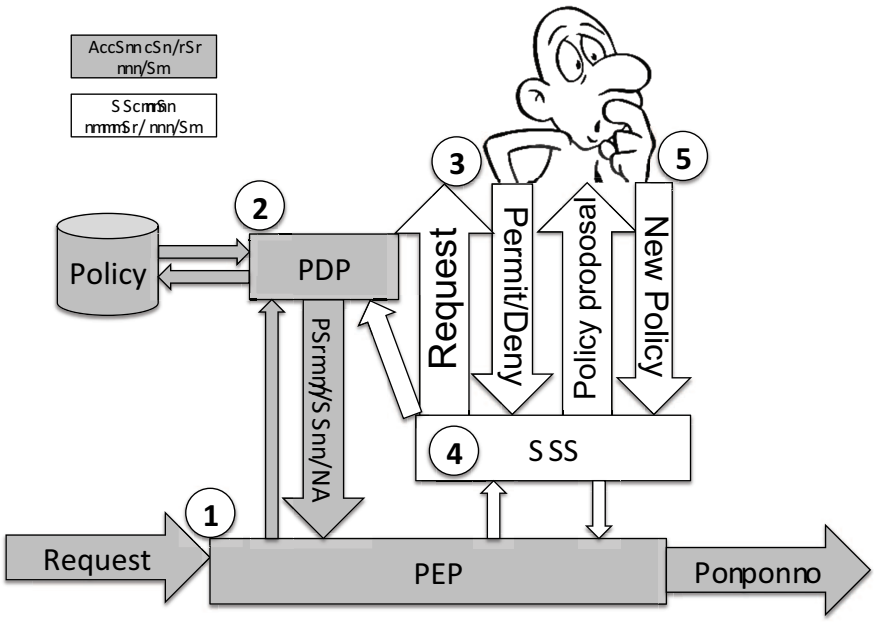


Figure 4.3 — Architecture globale de Kapuer

— le PDP (*Policy Decision Point*) où les politiques sont évaluées et comparées aux requêtes entrantes
— le PEP (*Policy Enforcement Point*) qui reçoit les requêtes, les traduit en XACML et les envoie au PDP. Il attend ensuite la réponse du PDP pour soit envoyer la requête au système d'aide à la décision soit envoyer la réponse du PDP à l'entité ayant fait la requête.
— une base de politiques, simple base de données où sont stockées les politiques.

Le dernier composant de cet architecture, qui ne fait pas partie de XACML, est le DSS (*Decision Support System* - système d'aide à la décision). Les sections suivantes du chapitre décrivent les différentes étapes du système en suivant l'ordre de la numérotation de la figure.

4.3.1 L'initialisation du système

Avant le début de l'exécution de Kapuer, le système doit être initialisé. Cette phase consiste en un échange avec l'utilisateur. Quelques questions permettent au système de percevoir une ébauche de son comportement en terme de protection de ses données de vie privée. L'initialisation du système est une phase très importante pour son fonctionnement. Nous verrons dans ce chapitre que les valeurs des critères sont utilisées pour plusieurs calculs. Sans l'initialisation, toutes ces valeurs seraient égales à 0 et l'apprentissage des préférences de l'utilisateur serait plus long. Le but principal de cette phase est de commencer à apprendre comment l'utilisateur souhaite protéger ses données de vie privée pour permettre ensuite au système, lors de son exécution, de converger plus rapidement et de pouvoir proposer des règles à l'utilisateur en limitant au maximum le nombre d'interactions.

Plus le système a d'informations à la fin de l'initialisation, plus rapidement il peut faire des propositions à l'utilisateur. Mais, il faut faire attention que cette phase ne devienne pas

pénible pour l'utilisateur. Il faut trouver le bon nombre de questions et arriver à en extraire suffisamment d'informations pour que l'initialisation soit utile. Si le nombre de questions est trop important, l'utilisateur n'utilisera pas le système. Au contraire, si le nombre de questions est trop faible, le système n'en tirera pas assez d'informations et l'initialisation n'aura pas été utile. Nous avons effectué une enquête auprès de différents publics (informaticiens et étudiants en deuxième année de master droit et informatique). Les questions que nous avons posées sont disponibles en annexe (voir annexe C). L'enquête a été complétée 17 fois entièrement et 9 fois partiellement. L'analyse des résultats nous a montré que ce questionnaire était trop long, trop compliqué et difficilement exploitable. De plus amples réflexions doivent être menées pour arriver à une phase d'initialisation pouvant être utile pour la suite de l'utilisation de Kapuer.

4.3.2 L'arrivée d'une requête et passage par le PEP

Une demande de partage d'une ou plusieurs ressources de la part d'un particulier ou d'une application est appelée une requête. Les ressources peuvent soit directement concerner l'utilisateur indépendamment du périphérique utilisé (son nom, son adresse-mail, etc.), soit utiliser des services du périphérique pour récolter les informations (le module GPS donne les coordonnées de l'utilisateur, l'agenda son emploi du temps, etc.).

Lorsqu'une requête arrive, elle est interceptée par le PEP (Policy Enforcment Point). Le PEP est un composant faisant partie de l'architecture XACML que nous utilisons pour le contrôle d'accès. Son rôle est de traduire la requête dans un langage à base d'attributs utilisé dans l'approche ABAC (Attribute Based Access Control), et d'appliquer la décision qui est prise concernant la requête. La requête est donc traduite dans un langage à base d'attributs. Le système est générique et il est possible d'utiliser et de transformer n'importe quelle information disponible dans la requête sous forme d'attributs. Chaque requête sera traduite selon les informations qu'elles contiennent. Le résultat est un fichier XML regroupant les attributs sous les balises sujet, ressource, action et environnement.

```
<Request>
   <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>fr.irit.siera.testing</AttributeValue>
      </Attribute>
   </Subject>
   <Resource>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>android.permission.READ_SMS</AttributeValue>
      </Attribute>
   </Resource>
   <Action>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>access</AttributeValue>
      </Attribute>
   </Action>
   <Environment>
      <Attribute AttributeId="When-Day" DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>2</AttributeValue>
      </Attribute>
      <Attribute AttributeId="When-Hour" DataType="http://www.w3.org/2001/XMLSchema#string">
         <AttributeValue>10</AttributeValue>
      </Attribute>
   </Environment>
</Request>
```

Figure 4.4 — Exemple de requête

La figure 4.4 montre un exemple de requête traduite par le PEP. Cette requête est découpée en quatre parties :

— le sujet, l'entité qui demande l'information. Ici c'est une application, l'attribut prend le nom de l'application comme valeur "fr.irit.siera.testing".
— la ressource, l'information demandée. Ici la valeur de l'attribut correspond à la permission requise pour avoir accès à l'information. "android.permission.READ_SMS" est la permission requise pour avoir accès en lecture aux SMS de l'utilisateur.
— l'action qui correspond au type d'action que le demandeur veut effectuer. Ici le demandeur veut avoir un accès à la ressource d'où la valeur de l'attribut "access".
— l'environnement qui correspond à toutes les autres informations de contexte disponibles au moment de la requête. Ici les seules informations disponibles sont le jour et l'heure de la demande, avec les valeurs d'attributs "2" et "10" qui correspondent au deuxième jour de la semaine et à la dixième heure (mardi entre 10h et 11h).

Cette requête est ensuite transmise au PDP (Policy Decision Point) qui va continuer le processus de contrôle d'accès.

4.3.3 L'analyse de la requête et la correspondance avec la base de politiques

La deuxième phase du processus de contrôle d'accès intervient lorsque le PDP reçoit une requête à analyser. Le rôle du PDP est de vérifier s'il existe une correspondance entre les politiques existantes dans la base de politiques et de donner la décision résultante. Une politique est construite sur le même modèle qu'une requête, elle est basée sur des attributs. Le PDP analyse pour chaque attribut de chaque politique la présence de cet attribut et la compare avec la valeur de la requête.

Dans le cas où les attributs et valeurs des attributs de la requête correspondent à ceux d'une politique, la valeur de la règle de cette politique est envoyée en retour au PEP. La règle peut alors prendre deux valeurs, "PERMIT" si le partage de l'information ou des données est accepté, "DENY" s'il est refusé.

Dans le cas où les attributs et valeurs des attributs de la requête ne correspondent à aucune des politiques présentes dans la base de politiques, la décision "NOT APPLICABLE" est renvoyée au PEP. Le PDP n'a pas pu prendre de décision basée sur les politiques existantes, le PEP doit donc utiliser un autre moyen pour arriver à une décision.

La Figure 4.5 montre un exemple de politique présente dans une base de politiques d'un PDP comportant les parties suivantes :

— un identifiant unique permettant d'identifier la politique
— une description de la politique
— la règle de la politique et la décision qui l'accompagne. Ici la décision "PERMIT" permet au sujet de la politique d'effectuer l'action sur la ressource.
— le ou les sujets destinés à la politique. Ici il n'y a qu'un seul sujet, l'application "fr.irit.siera.testing"
— la ressource protégée par la politique. Ici la ressource est déterminée par la permission Android "android.permission.READ_CONTACTS" qui est utilisée pour avoir un accès à la lecture de la liste de contacts de l'utilisateur.
— l'action ciblée par la politique sur la ressource. Ici "access" pour que le sujet puisse

avoir accès à la liste de contacts de l'utilisateur.

— la condition qui permet de rajouter des attributs concernant le contexte de la situation. La politique est évaluée positivement seulement si la requête répond à tous les éléments de la condition. Ici deux attributs sont ajoutés correspondant au jour et à l'heure "When-Day" et "When-Hour".

```
<Policy PolicyId="fr.irit.siera.testing*android.permission.READ_CONTACTS*2.13" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable">
  <Description>This policy PERMIT access to android.permission.READ_CONTACTS for fr.irit.siera.testing on 2.13</Description>
  <Rule RuleId="FinalRule" Effect="Permit">
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">fr.irit.siera.testing</AttributeValue>
            <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">android.permission.READ_CONTACTS</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">access</AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
        <EnvironmentAttributeDesignator AttributeId="When-Day" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">2</AttributeValue>
        </Apply>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
        <EnvironmentAttributeDesignator AttributeId="When-Hour" DataType="http://www.w3.org/2001/XMLSchema#string"/>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">13</AttributeValue>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>
```

Figure 4.5 — Exemple de politique XACML V2

4.3.4 Les interactions avec l'utilisateur

Le PDP peut envoyer trois décisions différentes au PEP : "PERMIT", "DENY" ou "NOT APPLICABLE". La troisième correspond au cas où aucune politique d'autorisation n'est applicable à la requête en cours. C'est alors au concepteur du système d'implémenter un comportement qui permet de répondre à l'émetteur de la requête. La majorité des concepteurs implémente la décision "NOT APPLICABLE" comme la décision "DENY" pour éviter la divulgation d'informations non gérée par une politique d'autorisation.

Dans notre cas, la décision "NOT APPLICABLE" indique qu'il n'y a pas de règles et donc que les connaissances actuelles des préférences de l'utilisateur ne sont pas suffisantes pour proposer à l'utilisateur d'écrire une nouvelle politique d'autorisation. Les préférences de l'utilisateur sont une représentation, par un ensemble de critères, de la politique de confidentialité préférée par l'utilisateur. Le système fait un apprentissage continu de ces préférences afin d'être au plus proche de la politique de confidentialité voulue par l'utilisateur et de s'adapter en cas de changement de comportement de l'utilisateur. Cet apprentissage est effectué par un composant présent dans le système, le système d'aide à la décision. Donc, dans le cas d'une décision "NOT APPLICABLE", le système doit parfaire l'apprentissage des préférences de l'utilisateur et pour cela, il informe l'utilisateur de la requête en cours et

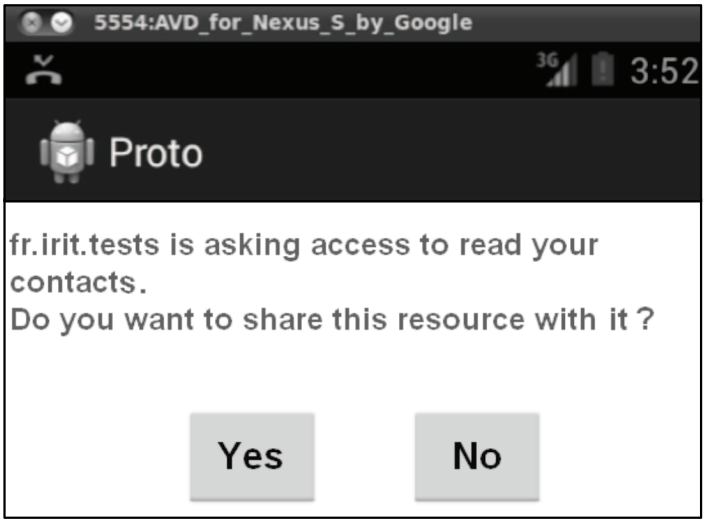


Figure 4.6 — Interaction avec l'utilisateur

lui demande de prendre une décision par rapport à la divulgation ou non de la donnée de vie privée concernée. La Figure 4.6 est une copie d'écran d'un exemple d'interaction avec l'utilisateur. Le système informe l'utilisateur qu'une entité (l'application "fr.irit.tests") veut accéder à une ressource (sa liste de contacts) et lui demande s'il accepte de partager cette ressource. Deux actions sont donc présentées à l'utilisateur, la divulgation D et la non divulgation nD. Une fois que l'utilisateur a répondu, le couple (requête, action) est envoyé au DSS qui l'analyse et utilise l'information pour faire évoluer les préférences de l'utilisateur.

4.3.5 L'utilisation du système d'aide à la décision

Lorsque le DSS reçoit une requête et l'action prise par l'utilisateur, le système effectue une analyse de ce couple. Pour cela nous utilisons une analyse multicritère. Chaque requête peut être décomposée en critères, ces critères sont ensuite agrégés grâce à un opérateur d'agrégation. Les préférences de l'utilisateur sont utilisées pendant cette étape pour pondérer les critères. Le résultat de l'agrégation fournira un score S_R de la requête R, évaluant le degré de connaissance des préférences de l'utilisateur face à cette requête.

4.3.5.1 La décomposition de la requête

Dès qu'une requête est reçue par le DSS après l'interaction avec l'utilisateur, elle est décomposée en critères afin de récupérer un critère pour chaque classe de critères. Ensuite le système d'aide à la décision crée des combinaisons de critères à partir des critères de la requête. Afin d'augmenter la vitesse d'apprentissage du système, des règles plus abstraites que celles correspondant aux requêtes sont proposées. Pour cela, toutes les combinaisons possibles entre les critères et leurs méta-critères hiérarchiques sont calculées, toujours en ne gardant qu'un critère de chaque classe (Exemple Figure 4.7).

L'étape suivante consiste à calculer le score S_R de chacune de ces combinaisons.

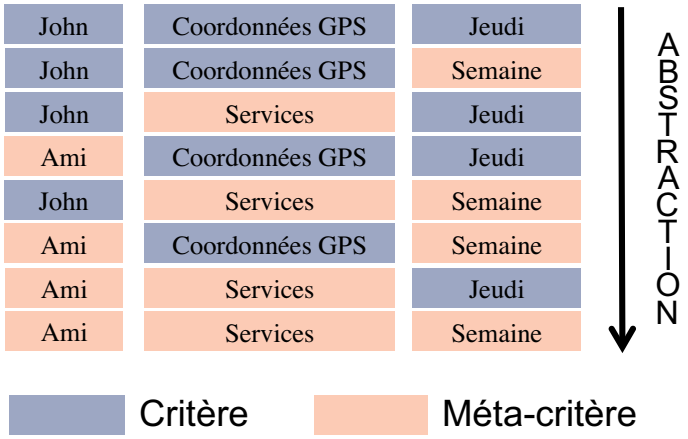


Figure 4.7 — Exemple de combinaisons de critères

4.3.5.2 Le calcul des scores de requête

Une fois que toutes les combinaisons sont créées, le système d'aide à la décision calcule, pour chacune d'entre elles, son score. Ce score est ensuite utilisé pour faire la mise à jour des critères et des méta-critères impliqués dans la requête.

Le calcul du score se fait en utilisant un opérateur d'agrégation sur les critères de la requête. Pour ce calcul, le système utilise pour chaque critère et méta-critère le score $s_D^t(x)$ si l'utilisateur a accepté ou $s_{nD}^t(x)$ s'il a refusé.

Trois opérateurs d'agrégation différents sont testés. Les résultats de ces trois approches sont comparés.

4.3.5.3 La moyenne pondérée

La moyenne pondérée est un opérateur d'agrégation défini par :

$$\psi(a_1, \ldots, a_n) = \sum_{i=1}^{n} w_i(a_i) \tag{4.4}$$

où les $w_i \in [0, 1]$ sont les poids des critères i, tels que :

$$\sum_{i=1}^{n} w_i = 1 \tag{4.5}$$

La moyenne pondérée est un opérateur facile à mettre en œuvre. Chaque critère est associé à un poids. Il suffit de calculer la somme des valeurs des critères et de les pondérer par leur poids. Pour trouver le poids p_x du critère x, nous effectuons la somme des scores de tous les critères de la classe du critère x que nous divisons ensuite par la somme des scores de tous les critères., soit :

— si la requête a été acceptée :

$$p_x = \frac{\sum_{i=1}^{m} s_D^t(i)}{\sum_{j=1}^{n} s_D^t(j)} \tag{4.6}$$

— si la requête à été refusée :

$$p_x = \frac{\sum_{i=1}^{m} s_{nD}^t(i)}{\sum_{j=1}^{n} s_{nD}^t(j)} \tag{4.7}$$

avec $i \in class(x)$ et $j \in CR$, $m = |class(x)|$ et $n = |CR|$.

Plus la somme des scores des critères de la classe C_x est grande, plus les préférences de l'utilisateur concernant cette classe de critères sont précises. Inversement, si cette somme est faible, cela implique que les préférences de l'utilisateur sont encore floues. Les différents poids des critères correspondent à l'importance de chaque critère. Le calcul s'effectue de la même manière pour les méta-critères. Lorsque tous les poids ont été identifiés, le système peut alors calculer les scores S_R de chaque combinaison. La combinaison obtenant le meilleur score est celle qui est retenue pour la suite du processus. La moyenne pondérée est un opérateur d'agrégation très utilisé mais qui a aussi des limites. Il ne prend pas en compte les dépendances entre critères. C'est pour cela que nous avons testé le système avec d'autres opérateurs dont l'intégrale de Choquet qui en plus de prendre en compte l'importance des critères, prend en compte les interactions entre critères afin d'appréhender les préférences de l'utilisateur de manière plus précise.

4.3.5.4 L'intégrale de Choquet

L'intégrale de Choquet [34] est un opérateur d'agrégation permettant de prendre en compte l'importance des classes de critères et les interactions entre deux critères. Là où la moyenne pondérée utilise un poids sur chaque critère pour agréger leurs scores et donner un score global à un objet, l'intégrale de Choquet utilise une capacité (ou mesure floue) pour calculer des poids pour tous les groupes de critères. Par exemple, un objet est décrit selon deux critères A et B. Quand la moyenne pondérée a seulement besoin d'identifier les poids a du critère A et b du critère B, l'intégrale de Choquet en plus de ces deux poids a aussi besoin d'identifier le poids ab du groupe de critère AB.

Définition 1. *Soit $CR = 1, \ldots, n$ un ensemble de critères. Une capacité sur CR est une fonction μ $2^{CR} \rightarrow [0,1]$ vérifiant $\mu(\emptyset) = 0, \mu(CR) = 1$ et $\mu(A) \leq \mu(B)$ si $A \subseteq B$.*

Plusieurs méthodes existent pour identifier la capacité nécessaire pour ensuite calculer l'intégrale de Choquet. Nous avons utilisé la technique du minimum de variance [43]. Cette technique est un problème d'optimisation où les préférences de l'utilisateur sur l'objet sont utilisées afin de contraindre le problème. Nous avons utilisé deux types de contraintes différentes :

— un pré-ordre partiel $\succcurlyeq_{\mathscr{A}}$ sur l'ensemble des combinaisons $\mathscr{A}$ (classement des combinaisons obtenues en fonction de la requête).
— un pré-ordre partiel $\succcurlyeq_{CR}$ sur l'ensemble des critères CR (classement de l'importance des classes de critères obtenu en fonction des préférences utilisateur).

Ces contraintes doivent être acquises grâce aux préférences de l'utilisateur, sans quoi le calcul ne peut pas aboutir et la capacité ne pourra pas être identifiée. Cela signifie que les

préférences doivent être initialisées avant l'utilisation du système pour assurer l'exécution. De la même façon, il se peut que les contraintes soient insuffisantes pour déterminer les capacités et ne permettent pas de trouver une solution au problème. Ce cas n'est pas encore traité, il constitue l'une des perspectives de travail.

Dans le cadre de l'agrégation par l'intégrale de Choquet, les contraintes utilisées peuvent s'écrire sous la forme suivante :

— pour le pré-ordre partiel $\succcurlyeq_{\mathscr{A}}$ sur $\mathscr{A}$:

$$a \succcurlyeq_{\mathscr{A}} b \Leftrightarrow C_\mu(a) - C_\mu(b) \geq \delta_C \tag{4.8}$$

$$a \sim_{\mathscr{A}} b \Leftrightarrow -\delta_C \leq C_\mu(a) - C_\mu(b) \leq \delta_C \tag{4.9}$$

Avec $C_\mu(x)$ l'intégrale de Choquet de la combinaison x et δ_C un seuil de préférence. Ainsi, si la différence entre les valeurs des intégrales de Choquet de deux combinaisons est supérieure au seuil δ_C, on peut dire qu'une combinaison est préférée ($\succcurlyeq_{\mathscr{A}}$) à l'autre. Si la valeur absolue de cette différence est inférieure à δ_C, alors il n'y a pas de préférence mais une indifférence ($\sim_{\mathscr{A}}$) entre les deux combinaisons.

— un pré-ordre partiel $\succcurlyeq_{CR}$ sur l'ensemble CR des critères :

$$i \succcurlyeq_{CR} j \Leftrightarrow \Phi_\mu(i) - \Phi_\mu(j) \geq \delta_\Phi \tag{4.10}$$

$$i \sim_{CR} j \Leftrightarrow -\delta_\Phi \leq \Phi_\mu(i) - \Phi_\mu(j) \leq \delta_\Phi \tag{4.11}$$

Avec $\Phi_\mu(x)$ l'importance du critère x et δ_Φ un seuil de préférence. Ainsi, si la différence d'importance entre deux critères est supérieure au seuil δ_Φ, on peut dire qu'un critère est préféré ($\succcurlyeq_{CR}$) à un autre. Si la valeur absolue de cette différence est inférieure à δ_Φ, il n'y a pas de préférence mais une indifférence ($\sim_{CR}$) entre les deux critères.

Lorsque les contraintes sont définies, le problème d'optimisation du minimum de variance devient le suivant :

$$\begin{aligned} & min \bar{V}(v) \\ & sujet\ à \begin{cases} v(S \cup i) - v(S) \geq 0, \forall i \in CR, \forall S \subseteq CR \setminus i, \\ v(CR) = 1, \\ C_v(a) - C_v(b) \geq \delta_C, \\ -\delta_C \leq C_v(c) - C_v(d) \leq \delta_C, \\ \Phi_v(i) - \Phi_v(j) \geq \delta_\Phi, \\ -\delta_\Phi \leq \Phi_v(k) - \Phi_v(l) \leq \delta_\Phi \end{cases} \end{aligned} \tag{4.12}$$

La résolution de ce problème entraîne l'identification de la capacité et permet le calcul de l'intégrale de Choquet pour chacune des combinaisons.

Définition 2. *Soit μ une capacité sur CR, et $f : CR \rightarrow R$ une fonction représentant les scores d'un objet sur n critères. L'intégrale de Choquet de f par rapport à μ est donnée par :*

$$C_\mu(f) := \sum_{i=1}^{n} [f(\sigma(i)) - f(\sigma - 1))]\mu(A_i) \tag{4.13}$$

Avec $A_i = \sigma(i), \ldots, \sigma(n), f(\sigma(0)) = 0$ et σ une permutation sur CR telle que $f(\sigma(1)) \leq f(\sigma(2)) \leq ... \leq f(\sigma(n))$.

Tout comme pour la moyenne pondérée, lorsque le score de toutes les combinaisons a été calculé, le plus élevé est conservé pour la suite du processus. Contrairement à la moyenne pondérée, l'intégrale de Choquet permet de prendre en compte les interactions entre les critères, ce qui permet d'affiner la précision des résultats et de connaître les dépendances existantes entre plusieurs critères. Cependant, l'intégrale de Choquet est complexe à mettre en œuvre. Nous avons utilisé le plug-in Kappalab [Grabisch et I] disponible pour le logiciel R afin de faciliter son implémentation. L'intégrale de Choquet pose certains problèmes. De nombreuses contraintes doivent être exprimées et ces contraintes ne sont pas toujours faciles à formuler. Nous avons donc développé notre propre opérateur, Kagop, prenant lui aussi en compte les interactions entre les critères.

4.3.5.5 Kagop : Kapuer AGgregation OPerator

Kapuer a pour but d'aider un utilisateur à prendre des décisions complexes basées sur plusieurs critères. Ces critères peuvent être liés les uns aux autres. Ainsi, un utilisateur peut accepter de dévoiler son emploi du temps professionnel avec ses collègues pendant les heures de travail seulement mais peut aussi refuser toute demande de localisation pendant ces mêmes heures de travail. Ici, le critère correspondant à l'heure de la demande est donc fortement corrélé à celui correspondant à la ressource demandée. Dans un cas, l'association des critères "heure de travail" et "calendrier professionnel" implique l'accès à la ressource. Dans le deuxième cas, l'association du même critère "heure de travail" avec cette fois-ci le critère "localisation" implique le refus d'accès à la ressource. Si on devait, dans cet exemple, étiqueter chaque critère soit "Permit" pour un critère favorable à la divulgation soit "Deny" pour un critère défavorable à la divulgation, cela donnerait "Permit" pour le calendrier, "Deny" pour la localisation et il ne serait pas possible d'étiqueter le critère heure de travail car il serait à la fois favorable et défavorable. Il est donc important de tenir compte des interactions entre les critères.

Nous proposons un nouvel opérateur d'agrégation permettant de prendre en compte ces interactions. Pour cela, le score S_R d'une combinaison n'est plus calculé en utilisant seulement les critères et méta-critères mais aussi les groupes de critères de la requête. Par exemple, la requête "John, un ami, demande l'accès à mon calendrier jeudi matin" est décomposé en trois critères, trois méta-critères, douze groupes de critères composés de deux

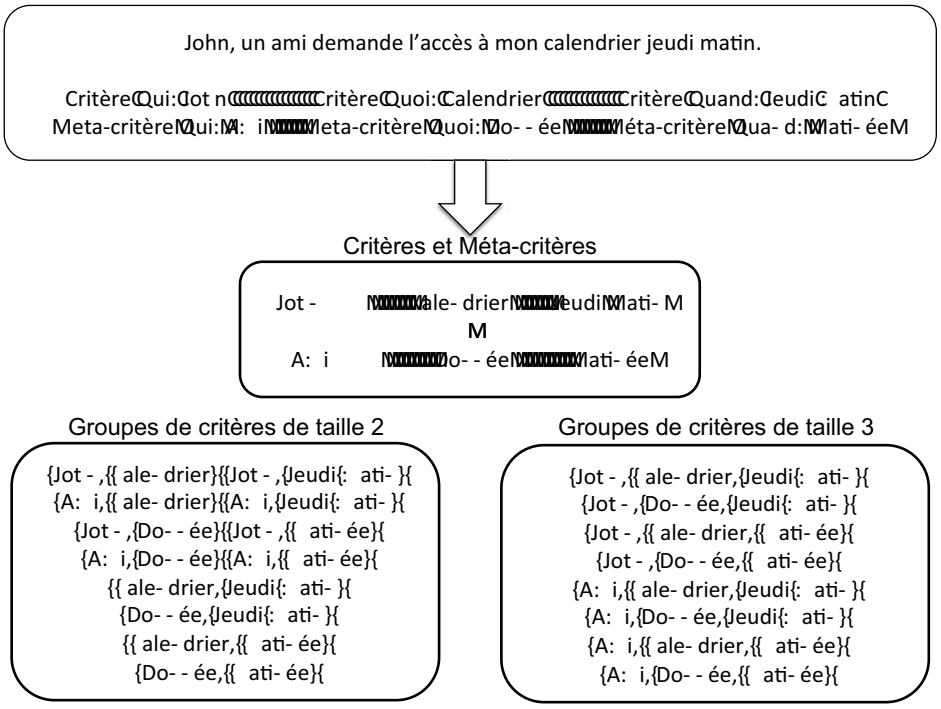


Figure 4.8 — Exemple de décomposition en critères, méta-critères et groupes de critères

critères et huit groupes de critères composés de trois critères comme présenté dans la Figure 4.8. Ces groupes de critères sont obtenus de la façon suivante :

Soit C_R un ensemble de critères correspondant à une combinaison de la requête R. Soit une fonction $GM : CR \rightarrow 2^{CR}$ tel que :

$$x \mapsto \{y \in CR | (x, y) \in H^*\} \tag{4.14}$$

L'ensemble des critères de la combinaison et de leurs méta-critères est alors représenté par $\cup_{c \in C_R} GM(c)$.

Soit $EG \in \mathscr{P}(\cup_{c \in C_R} GM(c))$, alors $GG : C_R \rightarrow 2^G$ tel que :

$$|EG| \geq 2 \tag{4.15}$$

$$\forall c_1, c_2 \in EG, class(c_1) \neq class(c_2) \tag{4.16}$$

GG est la fonction qui pour une combinaison C_R renvoie tous les groupes constituables à partir de cette combinaison. Lorsque tous les critères, méta-critères et groupes de critères de chaque combinaison ont été identifiés, les scores S_R des combinaisons peuvent être calculés de la façon suivante :

— si la requête a été acceptée :

$$S_R = \frac{\sum_{i=1}^{n} s_D^t(i) + \sum_{j=1}^{m} GG(C_R)(j)}{m} \tag{4.17}$$

— si la requête a été refusée :

$$S_R = \frac{\sum_{i=1}^{n} s_{nD}^t(i) + \sum_{j=1}^{m} GG(C_R)(j)}{m} \tag{4.18}$$

avec n le nombre de critères de la combinaison, m le nombre de groupes de critères et $GG(C_R)(j)$ le j-ème groupe de critères de l'ensemble $GG(C_R)$.

Cette méthode de calcul affecte un coefficient aux groupes de critères pour leur donner plus d'importance par rapport aux critères. Elle donne aussi une importance plus grande aux groupes de critères composés d'un plus grand nombre de critères. Le groupe avec le plus gros coefficient correspond à celui composé de tous les critères de la combinaison, il est plus important que les autres car il identifie entièrement le contexte de la requête.

Quelle que soit la méthode d'agrégation utilisée, une fois que le score de chaque combinaison est calculée, le système classe les combinaisons selon leur note et ne garde que celle ayant le score le plus élevé.

4.3.5.6 La mise à jour des critères et méta-critères

Une fois les scores de chaque combinaison calculés, le système les classe par ordre croissant et ne garde que la combinaison ayant obtenu le plus haut score S_R. Cette combinaison est considérée comme celle représentant le mieux le comportement attendu par l'utilisateur. C'est donc à partir de ce score qu'est calculée la valeur de la mise à jour des critères et des méta-critères de la requête.

Le but de cette mise à jour est d'obtenir une meilleure représentation des préférences de l'utilisateur. Pour cela, une seule valeur de chaque critère et méta-critère est mise à jour : $f^t(x)$ dans le cas où l'utilisateur a accepté la requête et $g^t(x)$ dans le cas où il aurait refusé. Par exemple, si l'utilisateur se comporte toujours de la même façon et accepte toutes les requêtes, les valeurs $f^t(x)$ de chaque critère augmentent et $s_D^t(x)$ qui est la différence entre $f^t(x)$ et $g^t(x)$ augmente aussi. Plus $s_D^t(x)$ est important, plus la perception des préférences de l'utilisateur est importante pour le critère x. Donc la mise à jour a bien atteint son but.

Le calcul de la mise à jour est une partie très importante de la phase d'apprentissage du système. La vitesse d'apprentissage est déterminée à ce moment-là. Si la valeur des mises à jour est faible, le système met du temps à évoluer et importune l'utilisateur plus longtemps. Par contre dans ce cas, les préférences sont plus fines et précises, et lorsque le système propose une règle à l'utilisateur, celle-ci est appropriée. Au contraire, si la valeur des mises à jour est grande, le système évolue plus rapidement et les interactions avec l'utilisateur sont moins fréquentes pour arriver aux mêmes valeurs. Mais dans ce cas, les préférences sont moins précises et certaines propositions faites à l'utilisateur ne sont pas pertinentes. Il faut donc trouver la bonne vitesse d'apprentissage pour maximiser la précision des préférences tout en minimisant les interactions avec l'utilisateur.

Nous faisons une différence entre la mise à jour d'un critère et d'un méta-critère. Leur calcul est le même, seule peut changer la vitesse d'apprentissage. Une vitesse plus rapide pour la mise à jour des méta-critères permet de proposer des règles plus agrégées à l'utilisateur. Mais, il peut arriver certains cas où les méta-critères ne sont pas pertinents pour l'utilisateur qui aura besoin d'une précision au niveau du critère. Dans ce cas, une vitesse d'apprentissage plus élevée pour les critères est plus appropriée. La valeur de mise à jour M_c des critères se calcule de la façon suivante :

$$M_c = log(S_R) \tag{4.19}$$

Et la valeur de mise à jour M_{mc} des méta-critères de la façon suivante :

$$M_{mc} = \frac{log(S_R)}{n_c * n_l} \tag{4.20}$$

avec n_c le nombre de critères qui composent le méta-critère et n_l le nombre de niveaux entre les critères et le méta-critère.

Il ne reste plus qu'à appliquer ces valeurs aux critères et méta-critères :

$$f^{t+1}(x) = f^t(x) + M_c \qquad ou\ exclusif \qquad g^{t+1}(x) = g^t(x) + M_c \tag{4.21}$$

$$f^{t+1}(x) = f^t(x) + M_{mc} \qquad ou\ exclusif \qquad g^{t=1}(x) = g^t(x) + M_{mc} \tag{4.22}$$

La mise à jour concerne tous les critères de la requête initiale ainsi que leurs méta-critères. Une fois que la mise à jour est terminée, le système vérifie si une proposition peut être faite à l'utilisateur.

4.3.6 Les propositions à l'utilisateur

Faire des propositions de règles à l'utilisateur est l'un des buts de Kapuer. Le système ne doit pas proposer n'importe quelle règle. Il doit s'assurer de ne proposer que des règles pouvant convenir à l'utilisateur. Pour cela, il se base sur les préférences de l'utilisateur et sur le score de la requête après avoir effectué la mise à jour.

Faire une proposition à l'utilisateur revient à dire que l'action associée à cette proposition est strictement préférée à son contraire. Autrement dit, soit l'utilisateur préfère autoriser la divulgation, soit il préfère la refuser. Si le système n'est pas en mesure de faire une proposition, c'est qu'il n'y a pas de préférence entre les deux actions. Deux situations peuvent entraîner cette non-préférence :

— la première lorsque le système n'a pas une représentation précise du comportement de l'utilisateur, donc quand il manque d'information.
— la deuxième lorsque l'utilisateur n'a pas un comportement fixe et n'agit pas de la même façon à chaque fois. Dans ce cas, le système ne peut pas inférer le comportement de l'utilisateur ni lui proposer une règle.

Ces deux relations forment un système relationnel parfait de préférences (s.r.p.p). Il n'est constitué que des deux relations binaires transitives suivantes :

— l'indifférence $\sim$ ou non-préférence qui correspond à une absence de raisons qui justifieraient une préférence en faveur d'une action ou de l'autre :

$$\sim: a \sim a' \Leftrightarrow aIa' \tag{4.23}$$

I étant une relation symétrique réflexive.

— la préférence stricte $\succ$ qui correspond à l'existence de raisons justifiant la préférence en faveur d'une des deux actions :

$$\succ: a \succ a' \Leftrightarrow aPa' \tag{4.24}$$

P étant une relation asymétrique irréflexive.

Afin de savoir si une proposition est faisable, le système doit recalculer le score $S_R(t+1)$ de la combinaison choisie à l'étape précédente avec les nouvelles valeurs $f^{t+1}(x)$ et $g^{t+1}(x)$ des critères et méta-critères. Ce score est ensuite comparé au paramètre λ, correspondant à la valeur seuil entre la relation d'indifférence et de préférence stricte. Si $S_R(t+1)$ est inférieur à λ, nous nous trouvons dans une situation d'indifférence et aucune proposition ne sera faite à l'utilisateur. Si $S_R(t+1)$ est supérieur à λ, nous nous trouvons dans une situation de préférence stricte et la combinaison peut être proposée à l'utilisateur. λ est un paramètre qui influe sur la vitesse de proposition à l'utilisateur. Plus il est faible, plus le système fait des propositions à l'utilisateur rapidement. Inversement, plus il est élevé, plus le système est lent à faire des propositions à l'utilisateur.

Dans le cas où $S_R(t+1)$ est supérieur à λ, le système propose une nouvelle règle à l'utilisateur. Cela correspond à une nouvelle interaction avec lui. Durant cette interaction, le système propose à l'utilisateur d'insérer dans la base de politiques une nouvelle règle, dont les attributs correspondent aux critères ou méta-critères de la combinaison choisie. La décision de cette règle est aussi communiquée à l'utilisateur. Le choix est donné à l'utilisateur d'accepter cette règle ou de la refuser.

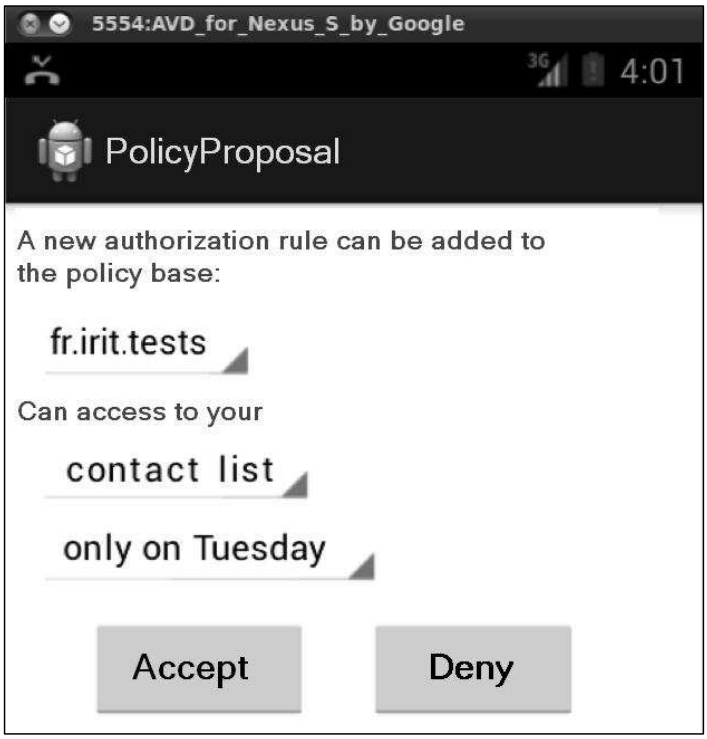


Figure 4.9 — Exemple de proposition de règle

Bien que les attributs présentés à l'utilisateur correspondent à ceux de la combinaison

censée être la plus pertinente pour l'utilisateur, celui-ci peut malgré tout changer chaque attribut. Soit l'attribut représente un critère et l'utilisateur peut choisir un méta-critère pour obtenir une règle plus agrégée. Soit l'attribut représente un méta-critère et l'utilisateur peut choisir le critère de la requête s'il ne veut pas que la règle soit agrégée avec ce méta-critère. La figure 4.9 présente un exemple de proposition de règle à l'utilisateur. Ici la règle présentée à l'utilisateur lui propose d'accepter de partager sa liste de contacts le jeudi avec l'application "fr.irit.tests". L'utilisateur peut changer les attributs pour que la règle soit étendue par exemple à toutes les applications, à ses données (comprenant la liste de contacts, le calendrier, etc.) ou à tous les jours de la semaine. Ainsi, l'utilisateur peut construire une règle lui convenant, sans avoir besoin de la spécifier et de l'écrire.

Si l'utilisateur accepte cette règle, le système d'aide à la décision l'ajoute dans la base de politiques du système de contrôle d'accès avant de communiquer la décision au PEP. S'il refuse, seule la décision est transmise au PEP. Une fois que le PEP reçoit la décision correspondant à la requête, elle est traduite pour être comprise par l'entité ayant fait la demande puis envoyée.

4.4 Conclusion

Nous avons présenté dans ce chapitre Kapuer, un système de recommandation destiné à assister l'utilisateur à protéger ses données de vie privée. Kapuer utilise un système de contrôle d'accès basé sur le modèle ABAC. Ce modèle laisse une grande liberté pour le développeur, lui permettant ainsi de choisir, selon les informations qui seront disponibles pour le système, les catégories ou classes de critères qu'il souhaite utiliser. Cela donne à Kapuer une grande généricité. Pour analyser les décisions de l'utilisateur et apprendre ses préférences, Kapuer utilise une analyse multicritères. Une fois que Kapuer a une vision suffisante des préférences de l'utilisateur, il peut lui proposer une règle d'autorisation. Si l'utilisateur accepte cette règle, elle sera ajoutée à sa politique d'autorisation et ses données de vie privée seront protégées en accord avec ses préférences. Le prochain chapitre présentera les différentes évaluations de Kapuer.

5 Le prototype Android

« Things don't have to change the world to be important. »

Steve Jobs

KAPUER est un système destiné à aider un utilisateur à protéger ses données de vie privée. Il est prévu pour être utilisé sur des systèmes mobiles où l'utilisateur ne peut pas être assisté d'un expert en sécurité pour l'aider à écrire ses politiques de sécurité. Nous avons réalisé un prototype de Kapuer. Ce prototype est basé sur un système Android, système open source donc facilement modifiable et actuellement utilisé par un grand nombre d'utilisateur dans le monde. Le but de ce prototype était de vérifier la faisabilité technique de l'implémentation de Kapuer sur un appareil physique et de pouvoir faire des tests avec des utilisateurs réels pour valider notre approche. Ce chapitre est composé de deux parties. La première présente le système de sécurité d'Android qu'il a fallu modifier pour mettre en place Kapuer et la deuxième détaille l'implémentation du système.

5.1 Le système de sécurité d'Android

Android est un système d'exploitation destiné à un usage mobile. Il est aujourd'hui utilisé par plusieurs centaines de millions d'utilisateurs. Android fournit une système open source comprenant un noyau Linux et un environnement à base d'applications mais aussi des logiciels intermédiaires entre les deux. Au final, Android est organisé en 5 couches (cf figure 5.1) :

— Le noyau Linux servant entre autre à la gestion des périphériques, de la mémoire ou du contrôle d'accès
— Une bibliothèque de logiciels telles que OpenGL ou SQLite
— Dalvik, une machine virtuelle permettant d'exécuter les applications Java.
— Un kit de développement complet pour les applications
— La couche applicative contenant les applications systèmes mais aussi toutes les applications créées par les développeurs

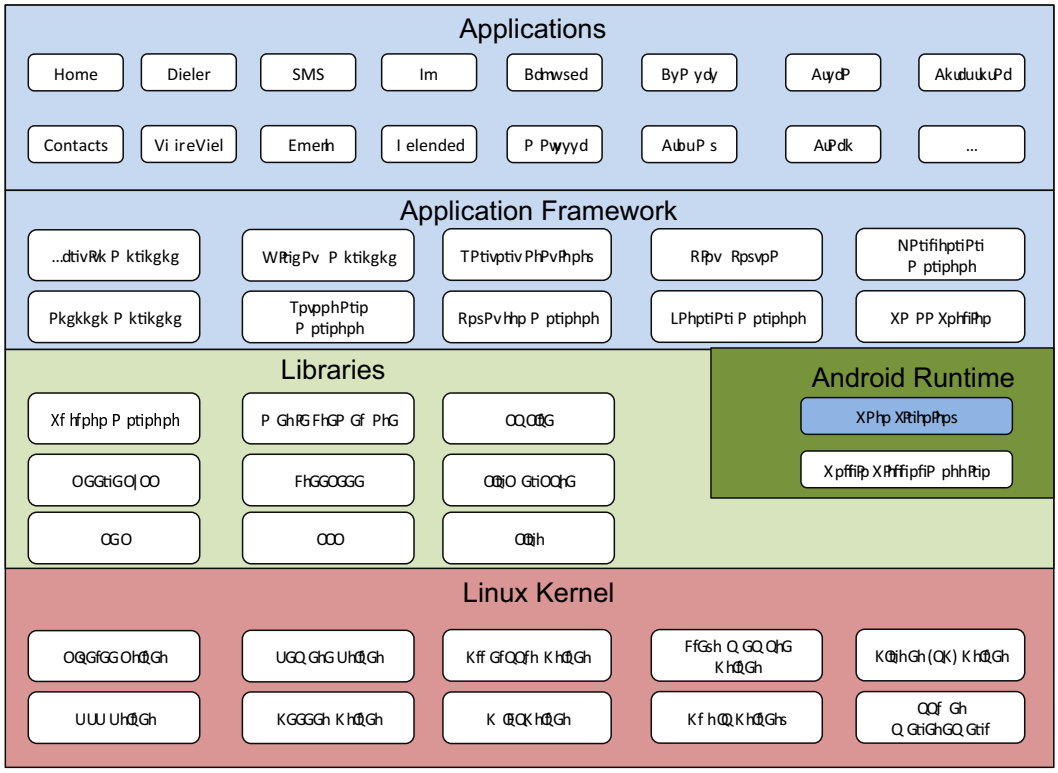


Figure 5.1 — Les couches logiciels d'Android

Grâce à l'utilisation d'un noyau Linux, utilisé depuis plusieurs années par beaucoup d'utilisateurs et en évolution constante, la base du système Android contient des éléments de protection ayant fait leurs preuves. En particulier, un des objectifs prioritaires du noyau est d'isoler les ressources de chaque utilisateur et ainsi éviter qu'un utilisateur puisse lire ou modifier les données d'un autre utilisateur. Android utilise cette isolation en considérant que chaque application est un utilisateur différent donc les applications et leurs données sont isolées les unes des autres.

En plus de la sécurité existante au niveau du noyau, Android sécurise aussi le système au niveau applications. Toutes les applications s'exécutent dans un environnement de type sandbox (bac à sable). Dans cet environnement fermé, le système assigne un identifiant unique à chaque application qui sont ensuite exécutées dans des processus différents. Sans autorisation, les applications ne peuvent pas interagir entre elles. Par défaut, Android ne fournit qu'un accès très restreint aux ressources systèmes. Ces restrictions sont présentes pour éviter qu'une application n'ait accès à des ressources sans que l'utilisateur soit au courant, évitant ainsi des comportements malicieux. Au moment de développer une application, un développeur peut malgré tout demander à avoir accès à ces ressources restreintes. Pour celà, il faut qu'il déclare dans le manifeste de l'application les permissions qu'il souhaite obtenir. Une fois les permissions déclarées, le développeur peut avoir accès aux ressources voulues via les méthodes de l'interface de programmation (API) protégée d'Android.

Ce système de permission permet d'avertir l'utilisateur des fonctionnalités que peuvent avoir les applications. A chaque installation d'une nouvelle application, le système va affi-

Figure 5.2 — Liste des permissions demandées par une application lors de son installation

cher à l'utilisateur la liste des permissions que requiert l'application (cf figure 5.2). L'utilisateur prend à ce moment connaissance des permissions que l'application demande et le choix lui est donné d'accepter de donner ces permissions à l'application et de continuer l'installation ou de refuser de donner ces permissions et dans ce cas, l'installation de l'application ne se poursuit pas. Il n'est pas possible pour l'utilisateur de n'accepter qu'une partie des permissions. Pour que l'application s'installe, il doit accepter et laisser le droit à l'application d'accéder aux ressources correspondantes aux permissions. Une fois l'application installée, l'utilisateur ne peut pas révoquer une ou plusieurs permissions. S'il change d'avis et ne veut plus qu'une application n'ait accès à une certaine permission, son seul choix est de désinstaller l'application. Android ne se contente pas de vérifier les permissions à l'installation d'une application. A chaque fois qu'une application veut accéder à une ressource sensible, elle appelle une méthode appartenant à une API protégée. A l'exécution, l'appel de cette méthode entraîne une vérification (cf figure 5.3) de la part d'Android pour s'assurer que l'application possède bien la permission nécessaire pour avoir accès à cette méthode.

5.2 L'implémentation

Nous avons décidé d'implémenter Kapuer dans Android car ce système d'exploitation est utilisé sur de nombreux périphériques mobiles par un grand nombre d'utilisateurs. De nombreuses applications disponibles sur Android utilisent les données sensibles de l'utili-

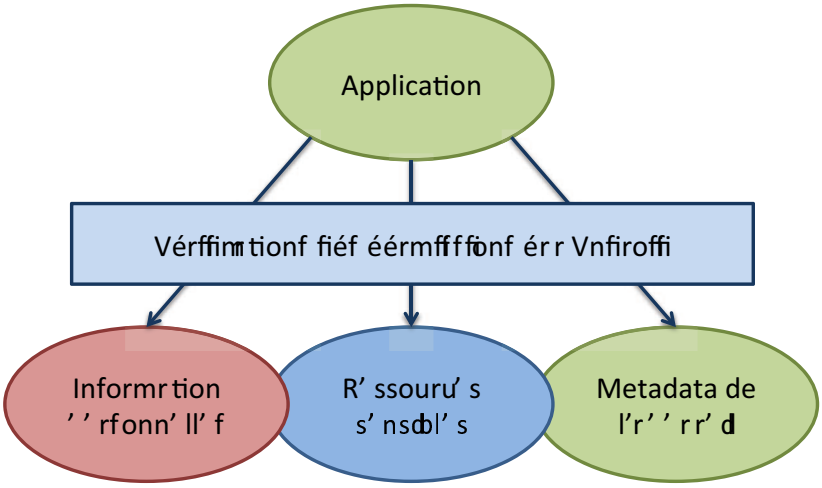


Figure 5.3 — Vérifications des permissions pour l’accès aux données sensibles

sateur qui n’a que peu de moyen d’être informé de ce qu’il advient à ses données et dans quel but elles sont utilisées.

Nous avons vu dans la section précédente qu’une application avait besoin de permissions pour avoir accès aux données sensibles et aux services de l’appareil. L’utilisateur pour se servir d’une application doit avoir obligatoirement accepté de lui donner accès à toutes les permissions qu’elle demande. Après avoir accepté et tant que l’application reste installée sur le périphérique, elle a accès sans restriction à toutes les données et services correspondant aux permissions acceptées. Nous avons donc implémenté Kapuer pour apprendre les préférences de l’utilisateur, savoir quelles permissions il voulait accorder et quelles permissions il voulait révoquer aux applications afin de modifier les permissions de chaque application pour se conformer au souhait de l’utilisateur. Nous avons dû modifier le système de sécurité d’Android pour permettre à Kapuer de gérer l’accès aux ressources non plus via les permissions accordées initialement aux applications mais via les politiques de sécurité qu’il a créées à partir des préférences de l’utilisateur.

L’implémentation de Kapuer dans Android s’est donc déroulée en deux parties : une modification du système d’exploitation pour intercepter les requêtes concernant les ressources sensibles de l’utilisateur et faire tous les traitements sur les critères et une partie du côté application pour gérer les interactions avec l’utilisateur et intégrer XACML pour la création et la gestion des politiques de sécurité.

5.2.1 Les détails de l’implémentation

Dans Android, Kapuer aide l’utilisateur à contrôler l’accès des applications aux ressources sensibles de l’utilisateur. La modification du système d’exploitation a été nécessaire pour la mise en place de Kapuer car toutes les vérifications de permissions se passent à ce niveau. Il n’aurait pas été possible d’avoir une gestion dynamique des permissions en restant au niveau application. La modification du code source d’Android comporte néanmoins

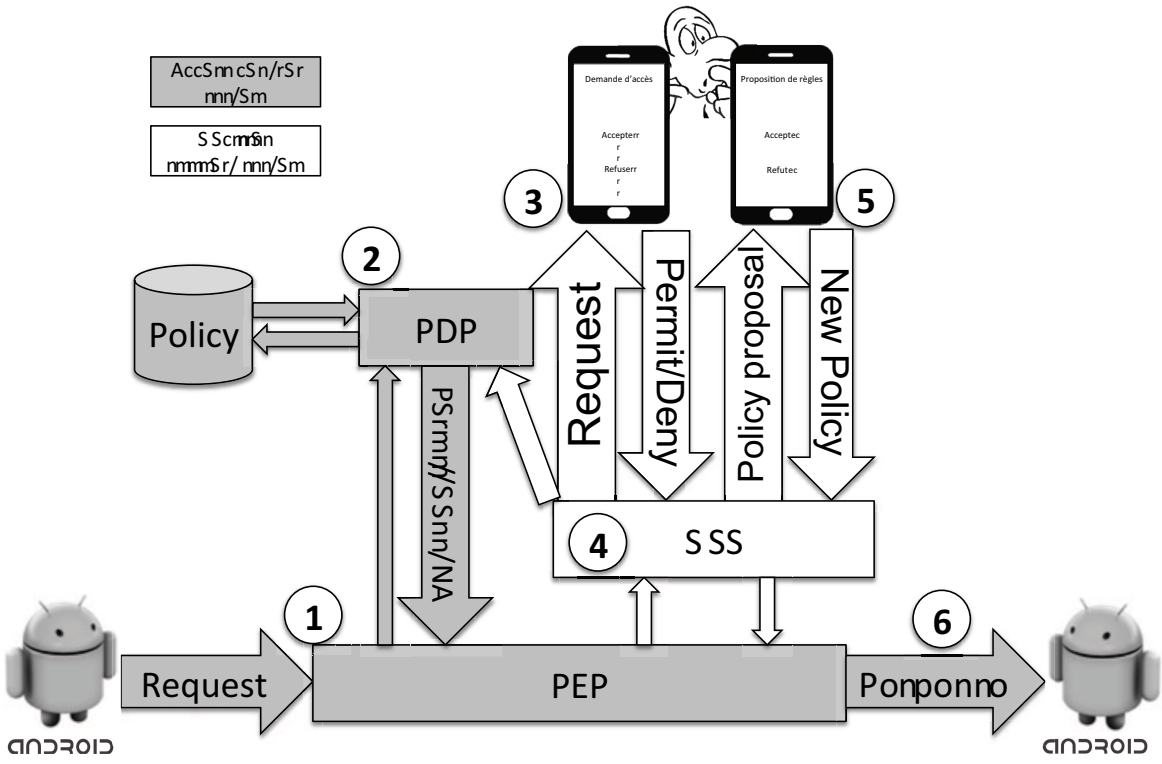


Figure 5.4 — Architecture du prototype

des inconvénients. Tester les modifications nécessite une compilation du système pouvant prendre jusqu'à plusieurs heures pour une compilation complète. Nous avons donc choisi de garder certaines parties de Kapuer au niveau application pour éviter les contraintes de compilation du système.

Une application doit déclarer ses permissions dans son manifeste, cela permet au développeur de l'application de faire des appels sur les méthodes correspondantes à ces permissions. Malgré tout, à l'exécution du code, pour chaque méthode protégée par une permission, Android fait une vérification pour être sûr que l'application a bien la permission d'utiliser cette méthode. Nous avons utilisé cette vérification pour intercepter les demandes d'accès aux ressources sensibles de l'utilisateur. Toutes les permissions ne concernent pas les ressources sensibles de l'utilisateur, certaines concernent seulement des utilisations de l'interface graphique ou l'accès à des informations comme les statistiques liées à la batterie. Nous avons donc défini la liste des permissions qui nous intéressent dans le cadre de Kapuer, celles liées à la vie privée de l'utilisateur.

L'interception d'une permission correspond à une demande d'accès à une ressource. En récupérant toutes les autres informations disponibles, Kapuer dispose d'une requête complète. Le prototype utilise trois informations pour former une requête :

— Le nom de la permission qui permet d'identifier la ressource qui est demandée.
— Le PID du processus qui a fait la demande. Kapuer associe chaque PID à un nom d'application.
— L'heure et le jour de la requête.

La figure 5.4 reprend l'architecture de Kapuer adaptée pour Android. L'étape 1 de cette figure correspond à l'interception de la requête. Cette requête va ensuite être utilisée par le système de contrôle d'accès pour vérifier si une politique de sécurité permettant de la gérer existe. Nous avons utilisé XACML pour la gestion du contrôle d'accès. Bien qu'il soit possible d'intégrer directement XACML au niveau du système d'exploitation, pour faciliter l'utilisation, nous avons préféré gérer XACML comme une application. Dès qu'une requête est reçue par XACML, elle est traduite par le PEP dans le format qu'utilise XACML. Ainsi chaque information de la requête est transformée en attributs stockés dans un fichier XML. La requête est ensuite transférée au PDP pour vérifier si une politique de sécurité pouvant la gérer existe (étape 2 de la figure 5.4). Si le PEP reçoit la réponse "PERMIT" ou "DENY", il envoie cette réponse au système de sécurité (étape 6 de la figure 5.4). Si la réponse qu'il reçoit est "Not Applicable", il envoie la requête au système d'aide à la décision.

Afin de sauvegarder les préférences de l'utilisateur, tous les critères et méta-critères ayant servi au moins une fois sont stockés dans un fichier. Le prototype utilise Kagop comme opérateur d'agrégation. Un autre fichier contient les groupes de critères qui lui sont nécessaires. Ces fichiers sont stockés au niveau du système et sont donc inaccessibles aux applications. Ces fichiers sont lus quand une requête arrive au niveau du système d'aide à la décision et il les modifie pour ajouter un critère jamais rencontré pour l'instant ou pour mettre à jour un critère existant.

Les interactions avec l'utilisateur sont gérées à partir d'une application (cf figure 5.5). Pour interagir avec l'utilisateur, Kapuer fait apparaître au moment voulu une page à l'écran où il décrit la requête et demande à l'utilisateur sa décision. Les informations de la requête sont transmises du système d'exploitation à l'application via un intent. Un intent décrit une opération qu'une application doit effectuer via une action et des données. Pendant son exécution, Kapuer envoit des intents avec deux actions différentes :

— La première avec l'action "android.intent.action.request" pour faire comprendre à l'application qu'il faut interagir avec l'utilisateur dans le cadre d'une requête pour connaître la décision de l'utilisateur (étape 3 de la figure 5.4). Dans ce cas, les données associées à l'intent sont les critères de la requête qui permettent à l'application de la décrire pour l'utilisateur. Lorsque l'utilisateur a pris sa décision, l'application la renvoie au niveau système.
— la deuxième avec l'action "android.intent.action.policy" pour faire comprendre à l'application qu'il faut interagir avec l'utilisateur pour lui proposer une nouvelle règle d'autorisation (étape 5 de la figure 5.4). Les données associées sont aussi des critères mais cette fois-ci ce sont ceux de la nouvelle politique proposée. Après avoir récupéré la décision, l'application l'envoie au niveau système. Dans le cas où l'utilisateur a accepté la proposition, l'application, en utilisant le langage XACML, crée la nouvelle règle et l'ajoute à la base de règles.

Un problème s'est posé lors des premiers tests du prototype. Une application faisant appel à une méthode protégée par une permission a forcément déclaré cette permission dans son manifeste, faute de quoi le développeur de l'application n'aurait pas pu l'utiliser. A l'exécution, si cette méthode est appelée et que la permission est refusée alors Android

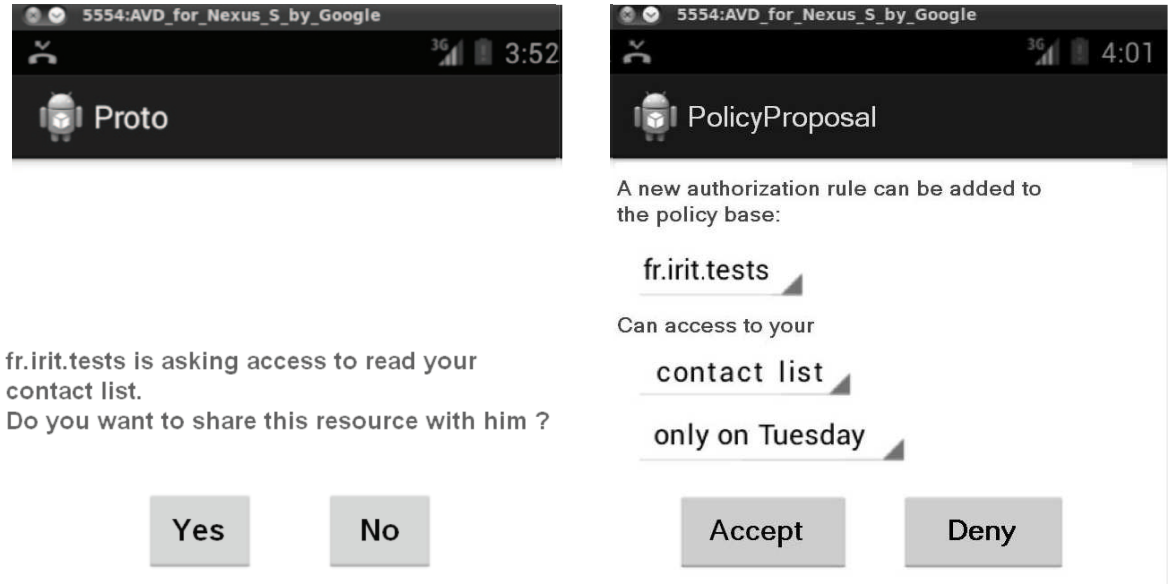


Figure 5.5 — Exemple des 2 types d'interactions possibles avec l'utilisateur

estime qu'une erreur a été commise et lève une exception qui entraîne l'interruption de l'application et l'apparition d'un message d'erreur. Cette exception était levée directement dans le noyau Linux. Kapuer ne peut pas être utilisé si chaque permission révoquée conduit à un arrêt brusque de l'application avec éventuellement la perte des données propres à l'application. Nous avons donc résolu ce problème en supprimant les exceptions dans le noyau. Ainsi les permissions peuvent être révoquées sans perturber les activités de l'utilisateur.

Kapuer travaille en parallèle du système de sécurité d'Android mais ne l'altère pas pour autant. Kapuer se place en amont des différentes vérifications de permissions. Si l'utilisateur décide, via Kapuer, de refuser l'accès à une ressource pour une application, alors Kapuer enverra le refus au système de sécurité qui empêchera l'accès à la ressource. Mais si l'utilisateur décide toujours via Kapuer de laisser l'accès à une ressource, Kapuer ne fera rien et laissera le système de sécurité faire les vérifications de permissions pour être bien sûr que l'application puisse avoir accès à cette ressource. Donc Kapuer ne remplace pas le système de sécurité d'Android mais le complète en permettant de révoquer une permission à l'exécution alors que cette gestion dynamique n'est pas possible avec la version Google d'Android.

5.3 Conclusion

Ce chapitre a présenté une implémentation de Kapuer sur un système Android. Avec ce prototype, un utilisateur peut interagir avec un smartphone ou une tablette dotée de notre version modifiée d'Android afin que Kapuer puisse apprendre ses préférences en terme de protection des données de vie privée pour ensuite lui proposer des politiques de sécurité correspondant à ses préférences. L'ajout de Kapuer ne modifie quasiment pas le système de sécurité mais permet de révoquer dynamiquement les permissions liées aux applications. Grace à cela, l'utilisateur peut installer une application même s'il ne veut pas donner l'accès

à certaines ressources pour cette application. Le prototype montre qu'il est possible d'intégrer Kapuer dans un système mobile et d'utiliser XACML pour la gestion des politiques de sécurité.

Le prototype présente malgré tout des inconvénients. Avoir modifié le code source d'Android nous a forcé à créer une version dérivée d'Android, obligatoire pour se servir de Kapuer. Le test de cette implémentation est donc complexe étant donné qu'elle nécessite de changer toute la partie logiciel d'un smartphone ou d'une tablette Android, y compris le noyau Linux, pour pouvoir utiliser Kapuer. Il n'est donc pas possible pour un possesseur de smartphone ou de tablette d'utiliser son appareil en l'état. De plus un tel prototype n'est pas un outil satisfaisant pour effectuer des tests pour évaluer Kapuer. De nombreux smartphones, autant d'utilisateurs et un temps d'utilisation conséquent seraient nécessaires pour obtenir des résultats. Pour contourner ce problème, nous avons réalisé un simulateur du système pour permettre d'effectuer des tests rapidement. Le simulateur et les résultats qu'il a permis d'obtenir sont présentés dans le chapitre suivant.

6 L'évaluation du système

« Le vrai génie réside dans l'aptitude à évaluer l'incertain, le hasardeux, les informations conflictuelles »

Winston Churchill

L'ÉVALUATION du système est une étape très importante et va nous permettre de valider ou non l'approche développée. Elle va consister en deux phases distinctes. La première consiste à évaluer le moteur d'apprentissage en comparant les trois opérateurs d'agrégation : L'intégrale de Choquet, la moyenne pondérée et l'opérateur que nous proposons, Kagop. La deuxième phase consiste à évaluer Kapuer par rapport aux approches existantes que nous avons décrites dans la section 2.2.

Le prototype de Kapuer développé pour Android n'est pas un outil approprié pour évaluer le système. Il n'est pas possible d'obtenir des résultats rapidement. En plus des moyens matériels et humains que cela aurait demandé, il aurait fallu plusieurs jours pour avoir pour chaque utilisateur des résultats exploitables. Il n'existe pas non plus de plateformes permettant d'évaluer les résultats de notre système. C'est à cause de ces nombreuses contraintes que nous avons décidé de développer notre propre plateforme, un simulateur, pour effectuer les différents tests pour valider notre approche. Cet outil permet de reproduire des situations de façon virtuelle. En modélisant le système et son environnement, il est possible de recréer les conditions d'utilisation et de pouvoir contrôler son évolution. C'est un outil de plus en plus utilisé actuellement car il permet de tester un système informatique dans des conditions se rapprochant d'un test réel mais en limitant les coûts de développement.

6.1 Le simulateur

6.1.1 Les exigences

Le simulateur doit répondre à de nombreuses exigences :

— Au niveau des critères, le simulateur doit être générique pour pouvoir accueillir n'importe quel type de critères et de méta-critères. Ces critères doivent aussi se conformer

aux formalisme de Kapuer afin de pouvoir créer facilement les classes de critères et les groupes de critères.
— Au niveau des requêtes, le simulateur doit aussi bien être capable de créer des requêtes aléatoires à partir de la base de critères que de rejouer un ensemble de requêtes définies venant par exemple d'une simulation déjà exécutée. Cela pour permettre de rejouer des simulations en changeant certains paramètres des algorithmes d'apprentissage et de comparer les résultats. De plus, étant donné que nous voulons comparer plusieurs opérateurs d'agrégations, les mêmes requêtes doivent pouvoir être analysées sur plusieurs opérateurs en même temps.
— Au niveau des résultats à afficher, le simulateur doit donner toutes les informations liées à la simulation. Cela comprend la liste des requêtes avec le score des requêtes pour chacun des opérateurs d'agrégation, la liste des critères ainsi que leurs valeurs au moment actuel pour chaque opérateur et aussi la liste des règles d'autorisation proposées par chaque méthode d'apprentissage.
— Au niveau de la simulation, pour avoir des tests rapides et valides, il n'est pas possible d'avoir un utilisateur pour répondre à chaque interaction. Le simulateur doit pouvoir utiliser des règles de comportement pour savoir comment répondre au système lors d'une interaction.

Enfin pour permettre une évaluation des résultats, le simulateur doit fournir plusieurs métriques qui permettront de comparer les opérateurs d'agrégation. Ces métriques propres à chaque opérateur d'agrégation sont :

— Le nombre de règles d'autorisation créées pendant la simulation.
— Le nombre d'interactions nécessaires pendant la simulation.
— Le pourcentage de complétude, c'est-à-dire le pourcentage de requêtes que le système peut gérer grâce aux règles d'autorisation créées pendant la simulation.
— L'évolution de cette complétude afin de déterminer combien de requêtes sont nécessaires pour arriver à différents seuils de complétude.
— Le taux d'erreurs dans les propositions. Le système peut proposer des règles d'autorisation contraires aux règles de comportement mais aussi des règles d'autorisation trop abstraites et donc partiellement erronées.

Nous allons maintenant présenter le simulateur et comment il se conforme à ces exigences.

6.1.2 Présentation du simulateur

Le simulateur que nous avons développé pour tester Kapuer est un programme Java. Ses buts sont multiples, allant de la génération de requêtes aléatoires sur des critères prédéfinis à la visualisation des valeurs de ces critères en passant par la visualisation des règles de sécurité créées ou de statistiques portant sur la simulation en cours. Il se présente sous la forme d'une fenêtre (cf figure 6.1) décomposée en trois parties :

— La première est composée de boutons et permet de créer des nouvelles requêtes, de

sauvegarder ou de charger des simulations, de lancer une analyse de la simulation et de changer les informations présentes dans la fenêtre centrale.
— La deuxième est la fenêtre centrale. Elle présente les informations importantes et les résultats du simulateur. Il y a trois types d'informations différentes visualisables. Les requêtes générées par le simulateur et pour chaque opérateur d'agrégation, la liste des critères utilisés par cet opérateur ainsi que la liste des règles d'autorisation générées par l'apprentissage pendant la simulation.
— La troisième regroupe certaines métriques d'une simulation pour les différents opérateurs d'agrégation utilisés.

Figure 6.1 — Fenêtre principale du simulateur

Les sections suivantes vont servir à présenter les différentes fonctionnalités du simulateur, à savoir la création des critères, la génération automatique et aléatoire des requêtes, la gestion des opérateurs d'agrégation et la simulation du comportement utilisateur.

6.1.2.1 La création des critères

Pour que le système puisse opérer, il lui faut une base de critères. Cette base de critères doit être créée préalablement à l'exécution du système faute de quoi le système ne pourra rien faire. Selon le test qu'il désire exécuter, l'utilisateur du simulateur va créer les critères dont il a besoin. Pour le bon fonctionnement du système, il faut au moins créer des critères dans deux classes différentes, par contre, le nombre de classes n'est pas limité.

En plus de la création des critères, l'utilisateur du simulateur doit aussi penser à créer les méta-critères de chaque critère. Les différentes hiérarchies de critères doivent donc être définies avant leur création pour permettre de renseigner les associations entre les critères et les méta-critères. L'utilisateur du simulateur peut aussi choisir les valeurs de chaque critère une à une ou directement définir des valeurs de départ pour l'ensemble des critères.

Dans le cas de l'utilisation de plusieurs opérateurs d'agrégation, il faudra créer une instance de la base de critères pour chacun des opérateurs. Étant donné que les valeurs des critères vont évoluer différemment selon chaque opérateur, il n'est pas possible d'opérer avec une seule instance de la base.

Les critères, méta-critères, leurs classes et leurs valeurs peuvent être visualisés dans le simulateur (cf figure 6.2). Par exemple dans la figure, *Compte* est un critère appartenant à la classe n°2, il a comme méta-critère *Données Utilisateur* et comme valeurs g^t et f^t : 2.00. Afin de différencier les critères des méta-critères, un signe "-" a été ajouté devant le numéro de classe des méta-critères. En bas de la figure, on retrouve les groupes de critères comme par exemple le groupe composé des critères {*Toute catégories, Ressource, Action* qui a comme valeurs g^t et f^t : 2.00. Les groupes de critères sont automatiquement créés à partir des critères et des méta-critères. Une fois que les instances de la base de critères sont créées, le simulateur peut générer des requêtes.

Crit1	Crit2	Crit3	Classe	Méta-critère	Score_D	Score_nD
Compte			2	Données Utilisa...	2,00	2,00
Audio			2	Services	2,00	2,00
Calendrier			2	Données Utilisa...	2,00	2,00
Contact			2	Données Utilisa...	2,00	2,00
SMS			2	Données Utilisa...	2,00	2,00
Bluetooth			2	Réseau	2,00	2,00
Paramètre			2	Données Systè...	2,00	2,00
NFC			2	Réseau	2,00	2,00
Log			2	Données Systè...	2,00	2,00
Action			-3	---	2,00	2,00
Accès Externe			-3	Action	2,00	2,00
Accès Local			-3	Action	2,00	2,00
Exécuter			3	Accès Local	2,00	2,00
Lire			3	Accès Local	2,00	2,00
Ecrire			3	Accès Local	2,00	2,00
Envoyer			3	Accès Externe	2,00	2,00
Recevoir			3	Accès Externe	2,00	2,00
Toute Catégories	Ressource	Action			2,00	2,00
Toute Catégories	Ressource	Accès Externe			2,00	2,00
Toute Catégories	Ressource	Accès Local			2,00	2,00
Toute Catégories	Ressource	Exécuter			2,00	2,00
Toute Catégories	Ressource	Lire			2,00	2,00
Toute Catégories	Ressource	Ecrire			2,00	2,00
Toute Catégories	Ressource	Envoyer			2,00	2,00

Figure 6.2 — Visualisation des critères/méta-critères/groupes de critères

6.1.2.2 La génération des requêtes

La génération des requêtes est une fonctionnalité très importante du simulateur. Elle va conditionner les résultats que donnera chaque simulation. La construction d'une requête se fait de manière aléatoire. Pour chaque classe de critères disponible dans la base de critères, le système va en choisir une aléatoirement. La liste de ces critères constituera la requête qui sera analysée par Kapuer. Avec cette méthode, contrairement à un système réel, Kapuer n'aura pas besoin de décomposer la requête en critères, cette dernière étant directement une liste de critères.

Certaines combinaisons de critères peuvent former des requêtes qui ne seraient pas pertinentes et qui fausseraient donc les résultats. Prenons l'exemple d'une simulation pour gérer l'accès aux ressources sensibles pour des applications de smartphones. Le simulateur peut fonctionner avec deux classes de critères, les applications et les ressources. Toutes les applications n'ont pas accès à toutes les ressources, il faut donc que les requêtes générées par le

simulateur ne concernent que des combinaisons (application, ressource) possibles. Pour éviter la génération de requêtes non pertinentes, l'utilisateur du simulateur peut définir avant l'exécution du simulateur l'ensemble des combinaisons non pertinentes. Pour l'instant, il n'est pas possible de définir cette ensemble à partir du simulateur, il faut le faire directement dans le code.

Dans la vue principale, le simulateur affiche les différentes requêtes qui ont été générées (cf figure 6.3). Dans l'exemple de la figure, la base de critères contient trois classes de critères donc chaque requête est composée de trois critères. Le premier correspondant à l'application faisant la demande, le deuxième à la ressource demandée et le dernier à l'action que l'application veut effectuer sur la ressource.

N°	Crit1	Crit2	Crit3
0	Mobile Securi...	Téléphone	Lire
1	Candy Crush...	Téléphone	Lire
2	6Play	3G	Envoyer
3	360 Security...	Contact	Lire
4	6Play	Bluetooth	Exécuter
5	Escape The...	Téléphone	Lire
6	Facebook Me...	Audio	Ecrire
7	Skype	Contact	Lire
8	Papa Pear Sa...	Téléphone	Lire
9	Adobe Reader	Contact	Lire

Figure 6.3 — Exemple de requêtes générées par le simulateur

6.1.2.3 La gestion des opérateurs d'agrégation

L'utilisation d'un simulateur présente de gros avantages : la vitesse d'exécution ou la possibilité de tester facilement les différents composants d'un système. Nous avons ainsi utilisé plusieurs opérateurs d'agrégation afin de comparer leur fonctionnement et les résultats qu'ils obtiennent. Pendant nos différents tests, nous avons opté pour trois opérateurs d'agrégations différents présentés dans le chapitre 4 : la moyenne pondérée, l'intégrale de Choquet et Kagop, l'opérateur que nous avons développé pour Kapuer.

Chaque opérateur d'agrégation a un fonctionnement similaire, il prend en entrée une liste de critères et donne en résultat un score. La liste de critères utilisée en entrée par chaque opérateur correspond aux critères obtenus après décomposition de la requête. Le score obtenu en sortie est le score S_R de la liste de critères prise en entrée. Les opérateurs d'agrégation peuvent donc être vus comme des composants logiciels interchangeables. Pour le simulateur, il n'y a pas de différence entre les opérateurs d'agrégation et il est possible d'en ajouter ou de les remplacer très facilement.

Lorsqu'une nouvelle requête est générée, le simulateur traite cette requête avec tous les opérateurs d'agrégation disponibles. Ce traitement est effectué en série, les uns après les autres. Chaque requête est traitée par tous les opérateurs, ce qui permet de pouvoir comparer leurs résultats sur les mêmes données. Ainsi le simulateur permet de visualiser les scores des requêtes de chaque opérateur (cf figure 6.4).

N°	Crit1	Crit2	Crit3	ScoreChoquet	ScoreKagop	ScoreWMean
0	Lampe Torch...	Paramètre	Lire	0,52	0,29	1,69
1	Doodle Jump	Contact	Lire	1,49	1,76	0,72
2	Clean Master	Wifi	Lire	1,48	1,14	1,73
3	Petite gorge...	Paramètre	Lire	0,25	1,68	1,38
4	Pages Jaunes	Paramètre	Lire	0,95	0,92	1,83
5	Vous avez ca...	Contact	Lire	1,88	1,95	1,34
6	Pages Jaunes	Paramètre	Ecrire	0,45	0,81	0,73
7	Facebook	Téléphone	Exécuter	1,24	0,85	0,42
8	Adobe Reader	3G	Recevoir	0,80	1,59	1,22

Figure 6.4 — Visualisation des scores des différents opérateurs sur le simulateur

6.1.2.4 La simulation de l'utilisateur

Kapuer est un système de recommandation et comme tout système d'aide à la décision, il ne doit pas prendre la décision à la place de l'utilisateur mais l'aider et lui proposer des solutions. Nous aurions pu développer le simulateur pour un fonctionnement identique où l'utilisateur aurait pu prendre les décisions à chaque requête ainsi qu'à chaque proposition de nouvelle règle. Cela aurait impliqué la présence d'une personne à chaque simulation pour prendre toutes les décisions requises. Plusieurs problèmes se posaient alors :

— Pour avoir une simulation complète, Kapuer peut nécessiter plusieurs dizaines, voire plusieurs centaines de requêtes. Cela aurait demandé beaucoup de temps pour accomplir une seule simulation, presque autant que sur un système réel alors que le but du simulateur est justement de gagner du temps par rapport à une implémentation de Kapuer sur une machine physique.

— Pendant la totalité de la simulation, l'utilisateur doit adopter un comportement cohérent dans ses décisions. Pour que Kapuer puisse déduire des règles, l'utilisateur doit avoir un comportement semblable dans des situations qui se reproduisent, faute de quoi on se retrouve dans un cas où l'utilisateur n'a pas de préférences cohérentes, ce qui est plus difficile à apprendre.

Pour éviter ces problèmes, nous avons choisi d'écarter l'utilisateur de la simulation. Mais pour rester dans le cadre d'un système d'aide à la décision et non avoir un système automatisé, nous avons remplacé l'utilisateur par un comportement artificiel. Il est modélisé avant de lancer une simulation par un ensemble de règles de comportement (cf figure 6.5). Ces règles vont donner au système des informations sur la gestion des différentes requêtes. Ainsi lors du traitement d'une requête, à chaque fois que Kapuer doit interagir avec l'utilisateur, il va le faire avec le simulateur. Le simulateur va vérifier s'il existe une règle dont tous les critères correspondent avec les critères de la requête. Si une telle règle existe, le simulateur utilise la décision associée à la règle pour répondre à Kapuer. Dans le cas où aucune règle ne peut répondre à la requête, le simulateur répond à Kapuer par un refus.

De la même façon, lorsque Kapuer doit proposer une règle à l'utilisateur, il la propose à la place au simulateur. Le simulateur vérifie une nouvelle fois avec les règles de comportement et peut donner sa réponse. De ce fait, il est possible à la fin d'une simulation de connaître la complétude des règles créées.

```
/*Règle :
 *R1 *Partage l'apn avec la famille
 *R2 *Partage l'apn le matin avec les collègue
 *R3 *Partage le cal le matin avec collègue
 *R4 *Partage le cal l'aprem avec la famille et les amis
 *R5 *Pargage le gps tout le temps avec la famille
 *R6 *Partage le gps l'aprem avec les amis
 *R7 *Partage le nom avec la famille
 *R8 *Partage les contacts avec les collègue le matin
 *R9 *Partage les contacts avec la famille et les amis l'aprem
 *R10 *Partage l'@mail avec les collègue le matin
 *R11 *Partage l'@mail avec la famille tout le temps
 *R12 *Partage l'@mail avec les amis l'aprem
 *R13 *Ne partage rien avec les inconnus
 * */
```

Figure 6.5 — Exemple de règles de comportement

6.2 Les scénarios de test

Pour évaluer le système, nous sommes partis sur deux scénarios différents. Le premier utilise une base de critères assez petite avec un seul niveau de méta-critère. Il utilise trois classes de critères dont une contextuelle illustrant le moment où la requête est faite. Le deuxième scénario utilise une base de critères plus réaliste qui pourrait être celle que l'on retrouverait en utilisant Kapuer sur un smartphone. Nous allons maintenant décrire ces deux scénarios plus en détails.

6.2.1 Premier scénario

Pour ce scénario , nous avons créé une base de critères composée de trois classes de critères.

Classe de critère n°1 :

Les critères de cette classe correspondent à quelle entité demande un accès. Cette classe comprend les quatre méta-critères et les neuf critères suivants :

- les critères *Jimmy*, *Lee* et *Billy* correspondant à trois membres de la famille de l'utilisateur. Ils ont donc tous les trois le même méta-critère *Famille*.
- les critères *Bob*, *Jay* et *Fred* correspondent à trois amis de l'utilisateur. Ils ont en commun le méta-critère *Ami*.
- les critères *Pierrick* et *Mick* correspondent à deux collègues de l'utilisateur. Ils ont en commun le méta-critère *Collègue*
- le critère *John* est une personne inconnue de l'utilisateur. Son méta-critère est *Inconnu*.

La hiérarchie de critères de cette classe est illustrée par la figure 6.6

Classe de critère n°2 :

La deuxième classe de critères regroupe les critères et les méta-critères utilisés pour les ressources. Ils vont servir à savoir quelle ressource est demandée. Pour cela, nous avons défini les trois méta-critères et les six critères suivants :

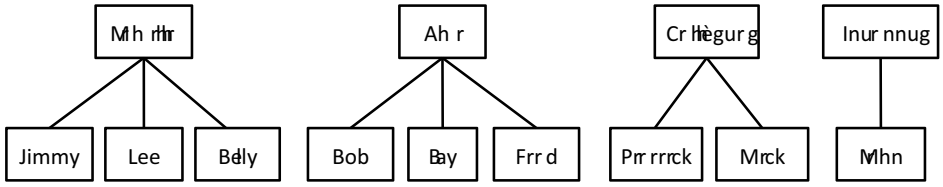


Figure 6.6 — Hiérarchie de critère de la classe n°1 du premier scénario

— les critères *Liste de contacts* et *Calendrier* sont des données appartenant à l'utilisateur mais ne le concernant pas directement. Ces deux critères ont en commun le méta-critère *Donnée*
— les critères *Nom* et *Adresse e-mail* sont des données appartenant à l'utilisateur mais le concernant directement cette fois-ci. Ces deux critères ont en commun le méta-critère *Information*.
— les critères *GPS* et *Appareil photo* sont des services du smartphone de l'utilisateur permettant d'avoir accès à la localisation de l'utilisateur et à capturer des photos ou des vidéos. Ces deux critères ont en commun le méta-critère "Service".

La hiérarchie de critères de cette classe est illustrée par la figure 6.7

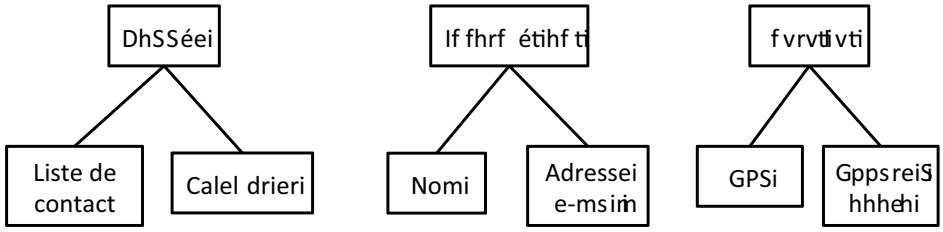


Figure 6.7 — Hiérarchie de critère de la classe n°2 du premier scénario

Classe de critère n°3 :

La troisième et dernière classe de critères concerne l'aspect temporel de la requête. Il est ici question de savoir à quel moment la requête a eu lieu. Pour cela, nous avons défini deux méta-critères et quatorze critères. Les deux méta-critères sont *Matin* et *Après-midi* et les critères correspondent à la demi-journée où la requête a été faite (*Lundi matin* ou *Mercredi après-midi* par exemple). On peut remarquer que ces critères sont déjà des abstractions de l'horaire réel de la requête. Mais au delà de la complexité de créer un nouveau critère à chaque fois qu'un nouvel horaire apparaît, cela n'aurait pas été pertinent au niveau de l'apprentissage. Il est peu évident que deux requêtes soient effectuées exactement à la même heure et cela n'apporterait pas d'information intéressante sur les préférences de l'utilisateur.

La hiérarchie de critères de cette classe est illustrée par la figure 6.8

Une fois cette base de critères mise en place, il ne reste plus qu'à fournir un comportement au simulateur pour lancer les simulations. Avec ce comportement, le système accepte de tout partager, tout le temps, avec la famille ; de tout partager le matin avec les collègues ; de tout partager l'après-midi avec les amis et finalement refuse toutes les requêtes d'un inconnu. Cela donne les règles suivantes :

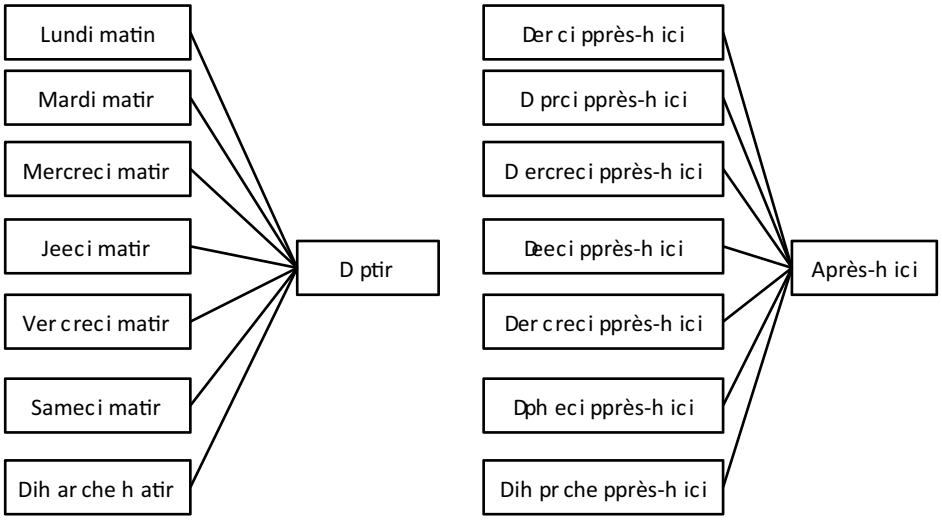


Figure 6.8 — Hiérarchie de critère de la classe n°3 du premier scénario

Liste des règles de comportement :

— Règle n°1 : Partage *Donnée*, *Information* et *Service* avec *Famille* les *Matin* et *Après-midi*.
— Règle n°2 : Partage *Donnée*, *Information* et *Service* avec *Ami* les *Matin* et *Après-midi*.
— Règle n°3 : Partage *Donnée*, *Information* et *Service* avec *Collègue* les *Matin* et *Après-midi*.

Si le système voit une correspondance entre les critères d'une requête et les critères d'une règle, il accepte le partage sinon il refuse. C'est pour cela qu'aucune règle ne concerne les inconnus. Étant donné que toute requête venant d'un inconnu doit être refusé, il n'est pas nécessaire de créer une règle pour décrire cette partie du comportement. L'objectif de Kapuer est de retrouver ces règles de comportement et de les convertir en politique d'autorisation grâce à l'apprentissage.

6.2.2 Deuxième scénario

Ce deuxième scénario utilise une autre base de critères et des hiérarchies plus complètes de méta-critères. Pour créer la nouvelle base de critères, nous sommes allés chercher les entités utilisées par CyanogenMod. Nous sommes donc partis sur une base de critères plus classique, basée sur trois classes de critères représentant les sujets faisant la requête, les ressources demandées et l'action que le sujet veut effectuer sur la ressource.

Classe de critère n°1 :

La première classe de critères, celle qui concerne les sujets, n'utilise plus directement des personnes mais des applications. Sur un smartphone c'est à travers les applications que les données de vie privée vont être divulguées. Donc pour pouvoir exécuter ce test à la fois sur le simulateur Kapuer mais aussi sur le smartphone, nous avons choisi les 50 applications gratuites les plus populaires fonctionnant sous Android. Contrairement au premier test, nous sommes partis sur une base de critères beaucoup plus riche, pour se rapprocher d'un

cas d'utilisation réel car un utilisateur d'Android a en moyenne 32 applications sur son smartphone selon un sondage effectué par Google en 2013[1]. Nous n'avons pris que des applications gratuites car toujours selon le sondage effectué par Google, les applications payantes ne sont qu'en très faible nombre sur les smartphones des utilisateurs et ne représentent en moyenne que 3.5 applications sur les 32. Ces 50 critères dépendent des 8 méta-critères suivants correspondant à chaque fois à la catégorie de l'application :

— le méta-critère *Jeux* est utilisé pour 28 critères soit plus de la moitié des applications. Le profil de l'utilisateur peut donc être considéré comme celui d'un adolescent assez joueur.
— le méta-critère *Social* regroupe les applications de réseaux sociaux. 4 applications sont concernées.
— le méta-critère *Communication* regroupe les application destinées à communiquer textuellement, vocalement ou visuellement. 4 applications sont concernées.
— le méta-critère *Gadget* concerne 4 applications.
— le méta-critère *Divertissement* concerne 3 applications.
— le méta-critère *Outils* concerne 3 applications.
— le méta-critère *Musique&Audio* concerne 2 applications.
— le méta-critère *Voyages&Infos* concerne 2 applications.

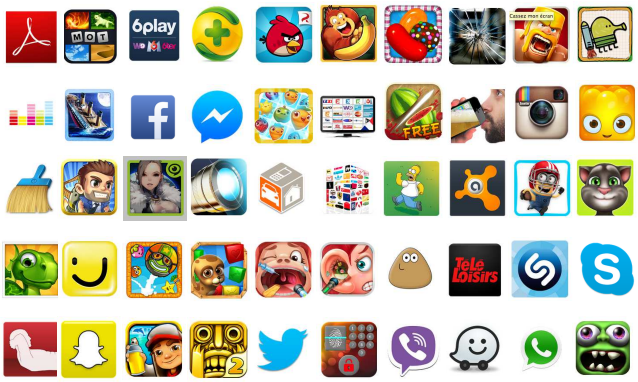


Figure 6.9 — Les 50 applications du scénario 2

Contrairement au premier test, cette fois plusieurs niveaux de méta-critères sont disponibles. Ainsi cette première classe de critères a un deuxième niveau de méta-critères composé du seul méta-critère *Toutes catégories*. Les 8 méta-critères du premier niveau dépendent de ce méta-critère. La hiérarchie de critères de cette classe est illustrée par la figure 6.10

Classe de critère n°2 :

La deuxième classe de critères concerne les ressources. Pour savoir quelles ressources utiliser pour ce test, nous sommes allés inspecter les permissions des 50 applications pour savoir à quelles ressources elles pouvaient avoir accès. Cette inspection nous a permis de trouver un ensemble de 15 ressources (figure 6.11).

La hiérarchisation des méta-critères de cette classe est plus complexe que les précé-

1. http ://think.withgoogle.com/mobileplanet/fr/

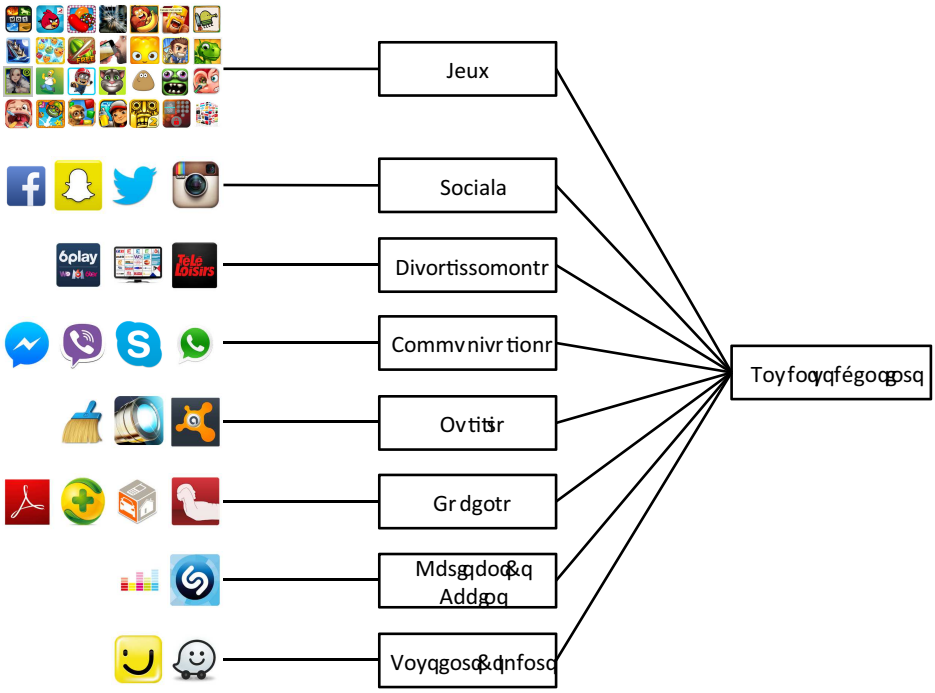


Figure 6.10 — Hiérarchie de critère de la classe n°1 du second scénario

Figure 6.11 — Les 15 ressources de la deuxième classe de critères

dentes. Les 15 ressources sont regroupées de la façon suivante :

— le méta-critère *Service* regroupe les critères *GPS*, *APN*, *Téléphone* et *Audio*.
— le méta-critère *Internet* regroupe les critères *3G* et *Wifi*. Le méta-critère *Réseau* regroupe ensuite le méta-critère *Internet* et les critères *NFC* et *Bluetooth*.
— le méta-critère *Données Utilisateur* regroupe les critères *Calendrier*, *Contacts*, *Compte* et *SMS*.
— le méta-critère *Données Système* regroupe les critères *Log*, *Paramètre* et *Fichier*.

D'autres niveaux de méta-critères sont ensuite présents. Ainsi le méta-critère *Hardware* regroupe les méta-critères *Service* et *Réseau* et le méta-critère *Données* regroupe les méta-critères *Données Utilisateur* et *Données Système*. Enfin le méta-critère *Ressources* est en haut de la hiérarchie et regroupe les méta-critères *Hardware* et *Données*. Cette hiérarchie est représentée par la figure 6.12 et permet d'avoir plus de granularités qu'auparavant et donc une abstraction plus élevée.

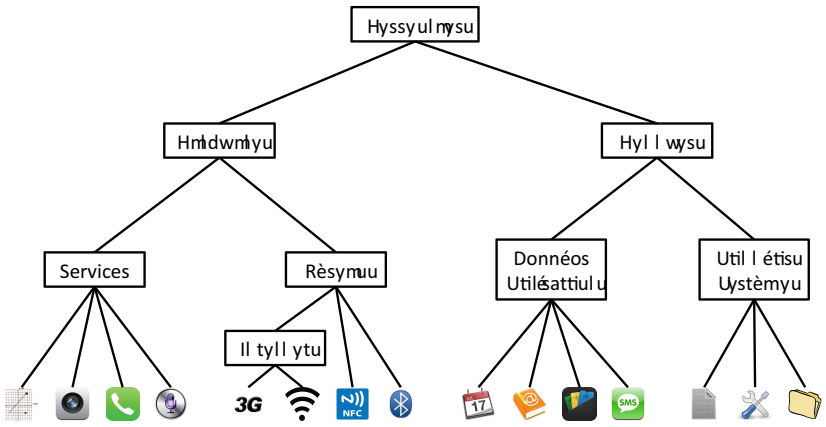


Figure 6.12 — Les 15 ressources de la deuxième classe de critères

Classe de critère n°3 :

Finalement la troisième classe de critères concerne les actions qui peuvent être effectuées par les applications sur les ressources. Pour trouver ces actions, comme pour les ressources, nous sommes allés inspecter les permissions des applications pour savoir comment elles pouvaient agir sur les ressources. Nous avons donc trouvé cinq critères et trois méta-critères pour cette classe :

— le méta-critère *Accès local* regroupe les critères *Lire*, *Ecrire* et *Exécuter*.
— le méta-critère *Accès externe* regroupe les critères *Envoyer* et *Recevoir*.
— le méta-critère *Action* regroupe les méta-critères *Accès local* et *Accès externe*.

La hiérarchie de critères de cette classe est illustrée par la figure 6.13

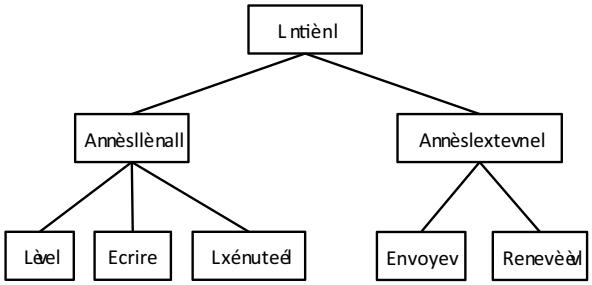


Figure 6.13 — Hiérarchie de critère de la classe n°3 du second scénario

Ces trois classes de critères forment la base de critères de ce test. Nous avons donc au total 70 critères et 20 méta-critères. Nous avons aussi défini un nouvel ensemble de règles de comportement. Nous avons créé huit règles, chacune est centrée sur une catégorie d'applications et décrit à quelles ressources ces applications peuvent avoir accès. Les règles utilisent tout les niveaux d'abstraction au niveau des ressources en allant du critère au plus abstrait des méta-critères. Les huit règles utilisées sont les suivantes :

Liste des règles de comportement :

— **Règle n°1 :** J'autorise les applications de la catégorie Jeux à avoir un accès à Internet.
— **Règle n°2 :** J'autorise les applications de la catégorie Social a avoir accès au réseau et aux données systèmes.
— **Règle n°3 :** J'autorise les applications de la catégorie Communication à avoir accès au réseau et aux services.
— **Règle n°4 :** J'autorise les applications de la catégorie Gadget à avoir accès à Internet.
— **Règle n°5 :** J'autorise les applications de la catégorie Musique&Audio à avoir accès au réseau et à l'audio.
— **Règle n°6 :** J'autorise les applications de la catégorie Outils à avoir accès à tout ce qu'elles demandent.
— **Règle n°7 :** J'autorise les applications de la catégorie Voyages&Infos à avoir accès au réseau et au GPS.
— **Règle n°8 :** J'autorise les applications de la catégorie Divertissement à avoir accès à Internet.

Pour être encore plus proche de la réalité, les requêtes de ce test ne sont plus totalement aléatoires. Nous avons regardé pour chaque catégorie d'applications quelles requêtes pouvaient être réalisées et nous nous sommes restreints à générer seulement ces requêtes. Par exemple, aucun jeu ne peut effectuer d'action sur l'appareil photo donc il n'y aura pas de requête du type *"Le jeu X a demandé à effectuer l'action Y sur l'appareil photo"*. Ainsi, nous nous rapprochons des conditions réelles et nous donnons un intérêt à la classe de critères concernant les actions. Bien que les règles de comportement autorisent un accès pour tout type d'actions, si une catégorie d'applications ne peut avoir qu'un accès en lecture sur une ressource, la règle d'autorisation créé par Kapuer doit correspondre et ne porter que sur la lecture et non sur toutes les actions.

6.3 Évaluation

Maintenant que les scénarios ont été présentés, nous allons pouvoir passer à l'évaluation du système. Cette évaluation s'effectue en deux temps. Premièrement une comparaison des opérateurs d'agrégation et deuxièmement une comparaison entre Kapuer et les approches existantes : Privacy Guard de CyanogenMod et XACML.

6.3.1 Évaluation des opérateurs d'agrégation

Cette partie de l'évaluation est elle aussi découpée en deux phases. Pour la première phase de test, tous les opérateurs d'agrégation utiliseront la même formule de mise à jour des critères et des méta-critères. Une deuxième phase de test sera effectuée en modifiant cette formule pour améliorer les performances du ou des opérateurs se comportant moins bien que les autres avec la formule de base.

6.3.1.1 Première phase

Pour effectuer cette première phase, nous avons lancé 10 simulations de 5000 requêtes chacune sur chaque scénario. L'objectif de la simulation en lançant autant de requêtes est d'essayer de s'approcher au plus près des 100% de complétude afin de pouvoir analyser au mieux l'apprentissage. Étant donné que les critères dans les requêtes sont choisis aléatoirement, il faut lancer un très grand nombre de requêtes pour que le maximum de combinaison puisse sortir et que l'apprentissage soit complet. A chaque fois qu'une nouvelle requête arrive, son traitement se fait sur les trois opérateurs d'agrégation, l'un après l'autre. Les simulations ont été effectuées sur une machine tournant sous Mac OS X dotée d'un processeur Intel Core i7 2.8GHz et de 4Go de Ram DDR3. Comme indiqué plus tôt, dans ce test, tout les opérateurs d'agrégation fonctionne avec la même formule de mise à jour des critères (cf formule 6.1).

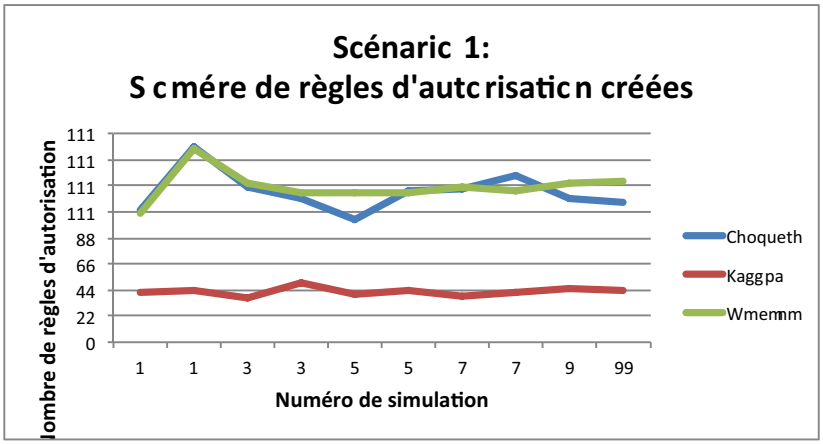


Figure 6.14 — Comparaison du nombre de règles d'autorisation créées sur le scénario 1

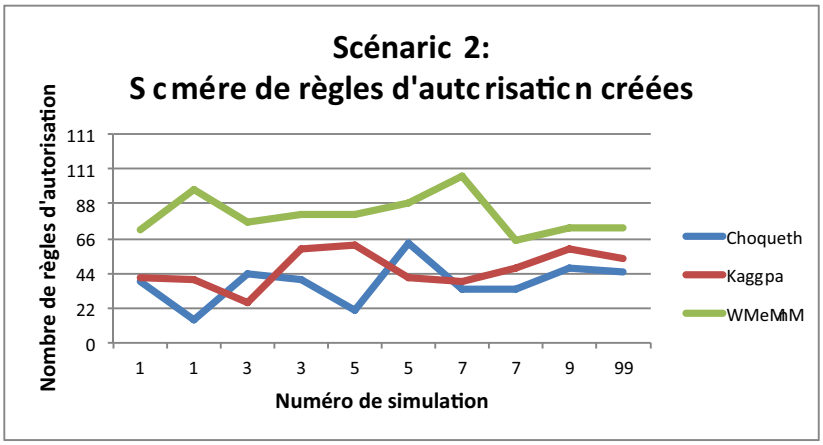


Figure 6.15 — Comparaison du nombre de règles d'autorisation créées sur le scénario 2

Métrique n°1 : Le nombre de règles d'autorisation

Les graphiques des figures 6.14 et 6.15 présentent les résultats de la première métrique : le nombre de règles d'autorisation créées par chaque méthode d'apprentissage pour chaque simulation. Plusieurs enseignements peuvent être tirés de ces graphiques. Sur les deux scé-

narios, Kagop est l'opérateur nécessitant en moyenne le moins de règles d'autorisations. Sur le scénario 1, les résultats sont très stables et tournent légèrement en dessous de 40 règles de moyenne. Pour le scénario 2, les résultats varient un peu plus avec une moyenne de 42 règles par simulation. La moyenne pondérée a quant à elle les plus mauvais résultats sur les deux scénarios. Sur le scénario 1, cet opérateur crée en moyenne trois fois plus de règles que Kagop et quasiment deux fois plus sur le scénario 2. Ce résultat était malgré tout prévisible car de la façon dont sont construites les règles de comportement, il y a interactions entre les critères. La moyenne pondérée ne prenant pas en considération ces interactions, elle est fortement pénalisée pour apprendre correctement les préférences de l'utilisateur. Le troisième opérateur, l'intégrale de Choquet, a des résultats très différent. Alors que sur le scénario 2, il est l'opérateur proposant le moins de règles d'autorisation à l'utilisateur, il est quasiment au même niveau que la moyenne pondérée sur le scénario 1. En ne prenant en compte que cette métrique, Choquet semble donc mieux se comporter sur une grosse base de critères avec des hiérarchies plus importantes.

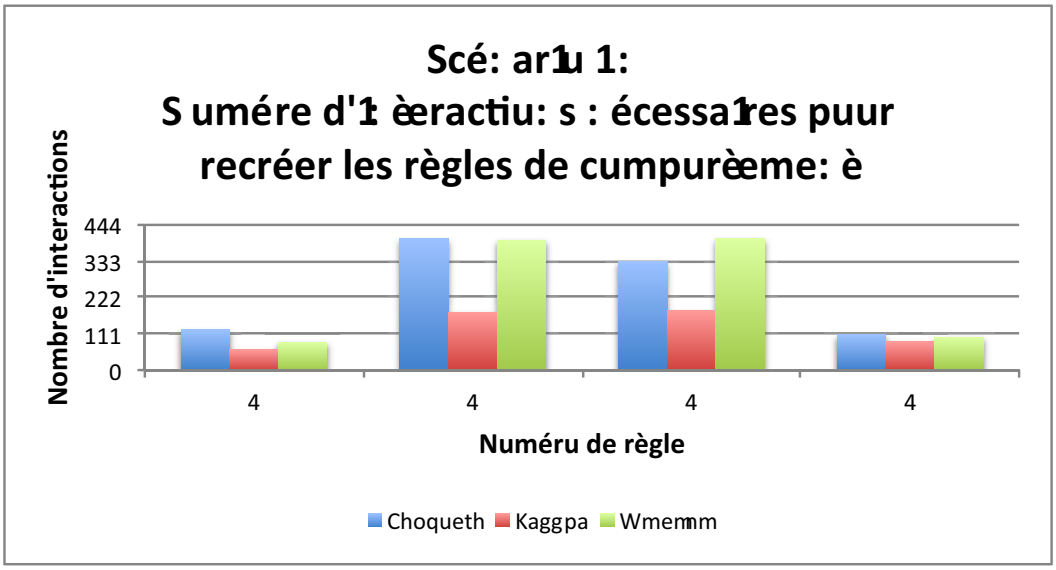


Figure 6.16 — Comparaison du nombre d'interactions nécessaires pour recréer chaque règle de comportement du scénario 1

Métrique n°2 : Le nombre d'interactions

Le nombre de règles d'autorisation créées ne permet pas à lui seul d'évaluer l'apprentissage des différentes méthodes. Cette métrique doit être mise en relation avec d'autres comme le nombre d'interactions nécessaires pour recréer chaque règle de comportement. Pour rappel, le scénario 1 comporte quatre règles de comportement et le scénario 2 en comporte huit. Les résultats de cette deuxième métrique sont illustrés sur les graphiques des figures 6.16 et 6.17.

Les résultats de cette métrique sont cohérent avec ceux de la première. Pour les deux scénarios, Kagop est toujours l'opérateur d'agrégation nécessitant le moins d'interactions pour recréer les règles de comportement. C'est donc lui qui impose le moins d'effort à l'utilisateur pour recréer sa politique d'autorisation. De la même façon, sur l'ensemble des deux scénarios, la moyenne pondérée est l'opérateur nécessitant le plus d'interactions pour re-

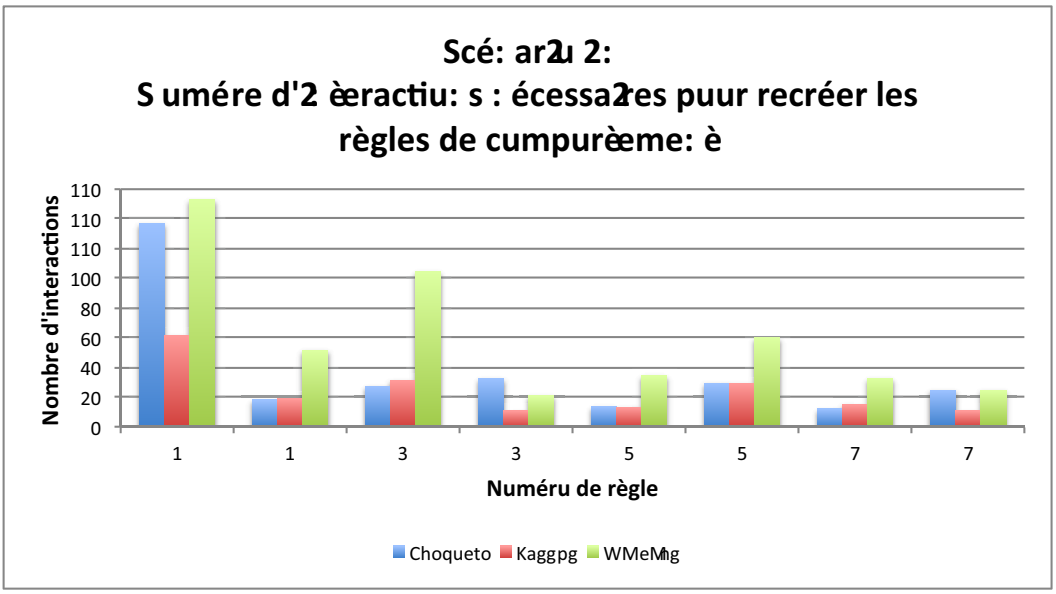


Figure 6.17 — Comparaison du nombre d'interactions nécessaires pour recréer chaque règle de comportement du scénario 2

créer l'ensemble des règles de comportement. Encore une fois, la non prise en compte des interactions entre les critères prive l'apprentissage de beaucoup d'informations. Hors ces informations manquantes sont nécessaires pour avoir une connaissance suffisante des préférences de l'utilisateur. Pour compenser ce manque d'informations, la moyenne pondérée est obligée d'interagir plus souvent avec l'utilisateur pour arriver au même niveau de connaissances des préférences.

Le comportement de l'intégrale de Choquet sur cette métrique est plus erratique. Si on l'analyse sur le scénario 1, il présente des performances similaires aux deux autres opérateurs sur les règles n°1 et n°4. Ces règles sont les plus simples et ne portent que sur une seule classe de critères. La règle n°1 autorise l'accès à toutes les ressources tout le temps pour la famille donc seul le critère *Famille* est important. De même pour la règle n°4 qui refuse tout accès à toutes les ressources pour les inconnus où seul le critère *Inconnu* est important. Pour les règles n°2 et n°3 plus complexes car portant sur les trois classes de critères, l'intégrale de Choquet obtient des résultats similaires à la moyenne pondérée et très loin de Kagop. Pour le scénario 2, bien que les règles de comportement soient plus complexes et portent à chaque fois sur plusieurs classes de critères, Choquet obtient de bon résultats, proche de Kagop, sur toutes les règles hormis la règle n°1. Une fois de plus, Choquet semble mieux se comporter sur une grosse base de critères avec des hiérarchies importantes.

Par contre il est important de remarquer que bien que le scénario 2 utilise une base de critères bien plus grande que le scénario 1 et que les règles de comportement sont plus nombreuses et plus complexes, le nombre d'interactions moyennes pour recréer une règle est en moyenne bien plus faible que pour le scénario 1. Il faut en moyenne 184 interactions pour recréer une règle du scénario 1 contre seulement 40 pour recréer une règle du scénario 2. Bien qu'il y ait deux fois plus de règles dans le scénario 2, le nombre d'interactions total est bien plus faible en faveur du scénario le plus complexe. L'utilisation d'une hiérarchie sur plusieurs niveaux est donc très importante pour réduire l'effort nécessaire à aider l'utilisateur

à écrire sa politique d'autorisation.

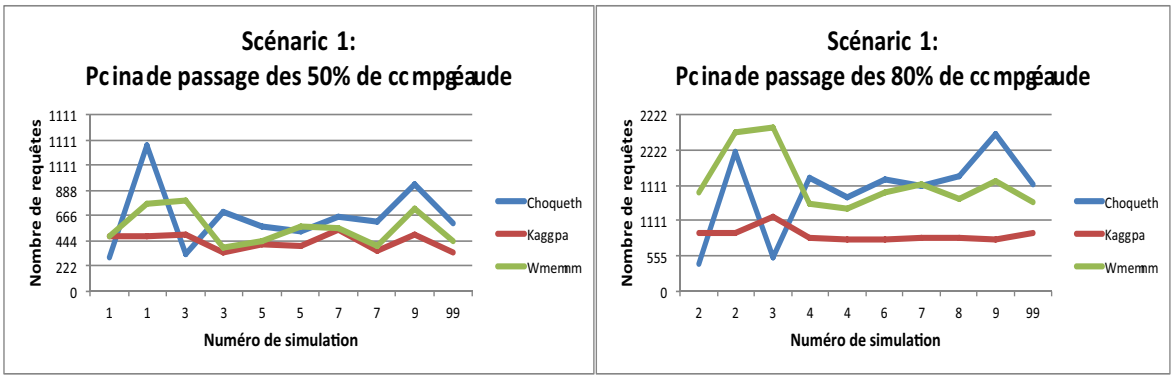


Figure 6.18 — Comparaison des points de passage à 50% et 80% de complétude pour le scénario 1

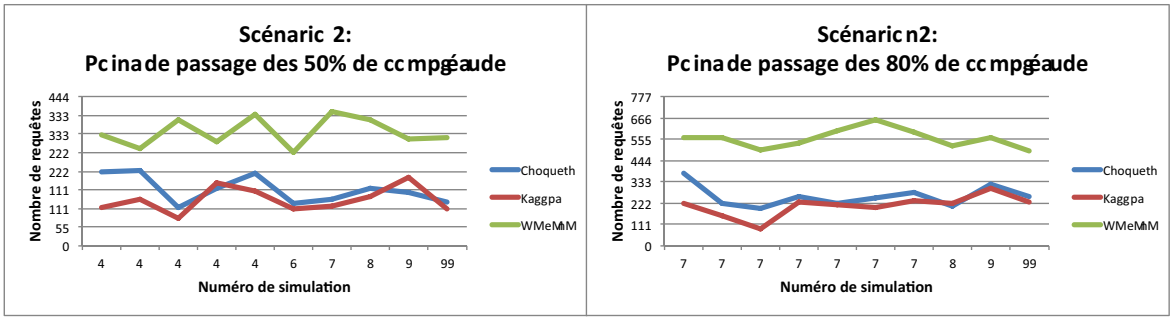


Figure 6.19 — Comparaison des points de passage à 50% et 80% de complétude pour le scénario 2

Métrique n°3 : La complétude

Si le nombre d'interactions total ou pour recréer chaque règle de comportement donne des informations sur la fin des simulations et de l'effort à fournir pour l'utilisateur, il est aussi important de s'intéresser à la rapidité de cet apprentissage et donc de regarder la progression du niveau de complétude. Les graphiques relatifs à cette nouvelle métrique sont disponibles en annexe (Annexe A pour le scénario 1 et Annexe B pour le scénario 2). Les graphiques des figures 6.18 et 6.19 les résument en comparant les nombres de requêtes nécessaire pour arriver à 50% et à 80% de complétude.

En regardant de plus près les graphiques en annexe, on peut voir que pour le scénario 1, quelque soit la méthode employée, les premières règles d'autorisation apparaissent à peu près en même temps. La seule exception est pour Choquet pendant la simulation n°7 où le départ s'effectue légèrement plus tard. Il faut remarquer sur ces graphiques que la différence d'ordonnée entre deux points consécutifs correspond au niveau de complétude amené par la règle d'autorisation correspondant au deuxième point. Plus cette différence est importante, plus la règle est abstraite. On peut voir sur les graphiques du scénario 1 que pour la moyenne pondérée et Choquet, ces écarts sont très faibles et on constate des agglutinements de points correspondant à des règles d'autorisation de bas niveau proposées à des périodes rapprochées. Les points pour Kagop sont un peu plus espacés autant en abscisse qu'en ordonnée, cela veut dire que Kagop propose des règles un peu moins souvent et que

celles ci sont plus souvent abstraites que pour les deux autres opérateurs. Si on regarde la forme de chaque ensemble de points, la progression pour Kagop et la moyenne pondérée est similaire quel que soit la simulation. Pour Choquet, les résultats sont plus différents et il peut se retrouver l'opérateur le plus rapide dans la simulation n°1 et n°3 mais le moins rapide dans les autres.

Pour le scénario 2, les résultats sont totalement différents. Avec des hiérarchies plus complètes et une forte possibilité d'abstraction des règles d'autorisation, on constate des écarts de niveau de complétude entre deux points du même opérateur bien plus important. Si on traçait les courbes de chaque nuage de points, les pentes des courbes du scénario 2 serait plus forte que celle du scénario 1. L'utilisation de hiérarchies importantes permet donc un apprentissage plus rapide. En comparant les opérateurs entre eux sur ce deuxième scénario, on remarque que généralement, la moyenne pondérée en plus d'avoir une pente plus faible que les autres, commence à proposer des règles d'autorisation plus tardivement. En compensant le manque d'informations dû aux interactions entre critères qu'elle ne prend pas en compte par un plus grand nombre d'interactions avec l'utilisateur, elle prend du retard par rapport aux autres opérateurs. La progression de la complétude est assez similaire entre l'intégrale de Choquet et Kagop. On remarque bien sur les graphiques de la figure 6.19 que ces deux opérateurs ont des courbes proches quand la moyenne pondérée est bien au delà. Les écarts sont d'ailleurs bien plus significatifs que sur les graphiques de la figure 6.18.

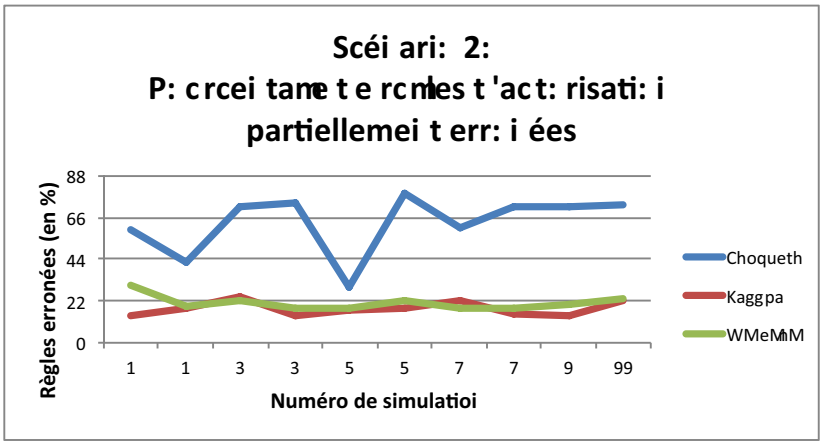


Figure 6.20 — Pourcentage de règles trop abstraites ou erronées pour le scénario 2

Métrique n°4 : Le pourcentage d'erreurs

La dernière métrique à évaluer pour ce test est le pourcentage de règles proposées erronées. Afin de vérifier que l'apprentissage est correct, il faut vérifier que les règles proposées conviennent à l'utilisateur. Pour ce test, cela revient à vérifier que les règles d'autorisation correspondent bien aux règles de comportement de chaque scénario. Si une règle d'autorisation est contraire aux règles de comportement, alors l'apprentissage n'est pas correct et Kapuer ne sera pas utile à l'utilisateur. Mais il y a un autre cas où les règles peuvent être erronées. Une règle d'autorisation peut correspondre en partie à une règle de comportement mais la trop grande abstraction de cette règle fait qu'elle ne correspond pas totalement. Dans ce cas là l'apprentissage, en étant trop rapide sur certains méta-critères, n'apprend pas correctement les préférences de l'utilisateur. Ces fautes sont moins graves pour le système

car lorsque le système sera utilisé par un vrai utilisateur, il est prévu qu'il puisse changer l'abstraction des critères d'une règle qui lui est proposée. Ainsi en cas d'abstraction trop élevée sur un critère, il pourra la baisser et obtenir une règle d'autorisation qui lui convient parfaitement.

Un seul graphique, celui de la figure 6.20 illustre cette métrique. Étant donné que le scénario 1 ne permet pas beaucoup d'abstractions, aucune règle proposée n'a été partiellement erronée dû à une trop grande abstraction. Les seules règles trop abstraites proposées dans ce test l'ont donc été sur le scénario 2. Les résultats montrent que Kagop et la moyenne pondérée ont un nombre de règles proposées erronées semblables, avec 16,4% pour Kagop et 19.0% pour la moyenne pondérée. Par contre, l'intégrale de Choquet admet un nombre de règles proposées erronées très supérieur avec une moyenne de 57,3%. Donc Choquet apprend vite mais mal. En regardant les règles d'autorisation proposées par chaque opérateur, on constate que les règles erronées sont toutes composées d'un ou plusieurs méta-critères du plus haut niveau soit *Toutes catégories*, *Ressources* ou *Action*. Nous avons lancé ce test avec la même formule de mise à jour des critères et des méta-critères. Si cette formule donne des résultats plutôt correct pour la moyenne pondérée et Kagop, elle n'est pas optimisée pour Choquet qui fait augmenter les valeurs des méta-critères trop rapidement et ainsi propose des règles d'autorisation trop abstraites à l'utilisateur.

6.3.1.2 Amélioration du paramétrage de l'intégrale de Choquet

Comme on vient de le voir, l'intégrale de Choquet n'obtient pas de bons résultats en utilisant la même formule pour mettre à jour les méta-critères que les autres opérateurs d'agrégation. Les valeurs des méta-critères augmentent trop rapidement et cela cause une majorité de propositions trop abstraites dans le scénario 2. Il est donc nécessaire de modifier cette formule pour faire en sorte que les valeurs des méta-critères augmentent moins rapidement et obtenir ainsi des préférences plus proche du comportement de l'utilisateur. La formule utilisée pour la première phase était :

$$M_{mc} = \frac{log(S_R)}{n_c * n_l} \tag{6.1}$$

Nous avons modifié cette formule pour avoir des valeurs de mise à jour plus faible. La nouvelle formule de mise à jour des méta-critères pour l'intégrale de Choquet est :

$$M_{mc} = \frac{log(\frac{S_R}{2})}{n_c * n_l} \tag{6.2}$$

Une deuxième phase de test à été lancé pour comparer les résultats avec cette nouvelle formule.

6.3.1.3 Deuxième phase

Pour cette deuxième phase de tests, nous avons repris les mêmes scénarios et les mêmes bases de critères pour chacune des dix simulations. Ainsi nous avons relancé les mêmes

simulations sur la moyenne pondérée, Kagop et l'intégrale de Choquet utilisant tout les trois la même formule 6.1 de mise à jour des méta-critères. Nous rajoutons l'intégrale de Choquet utilisant la nouvelle formule 6.2. Toutes les métriques ne seront pas développées dans cette section mais seulement celles permettant de tirer de nouvelles conclusions.

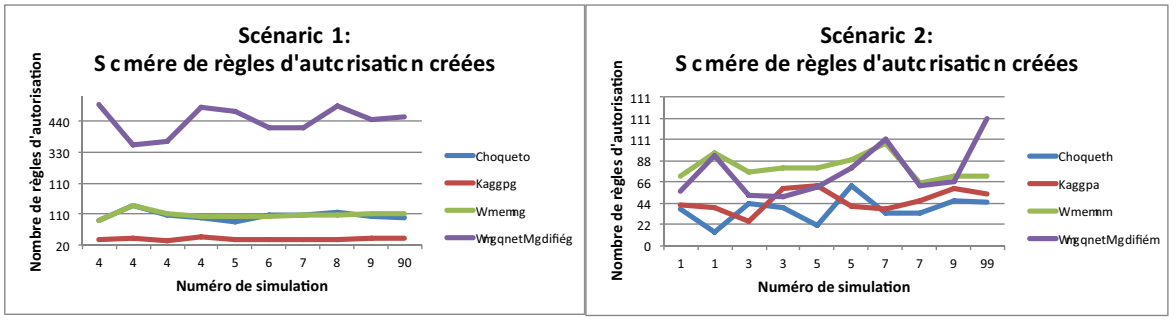


Figure 6.21 — Nombres de règles d'autorisation créées

Les graphiques de la figure 6.21 reprend ceux des figures 6.14 et 6.15 en y rajoutant la courbe de Choquet avec la formule modifiée. On peut voir que dans les deux scénarios, la modification sur Choquet amène un nombre supérieur de règles créées. Étant donné que les valeurs des méta-critères augmentent moins rapidement, les règles proposées sont de bas niveau et en grand nombre pour couvrir tous les cas possibles. Alors qu'originalement Choquet avait des résultats similaires à la moyenne pondérée pour le scénario 1 et à Kagop pour le scénario 2, avec la modification, le nombre a plus que triplé pour le scénario 1 et pour le scénario 2, les résultats sont devenus comparables à ceux de la moyenne pondérée.

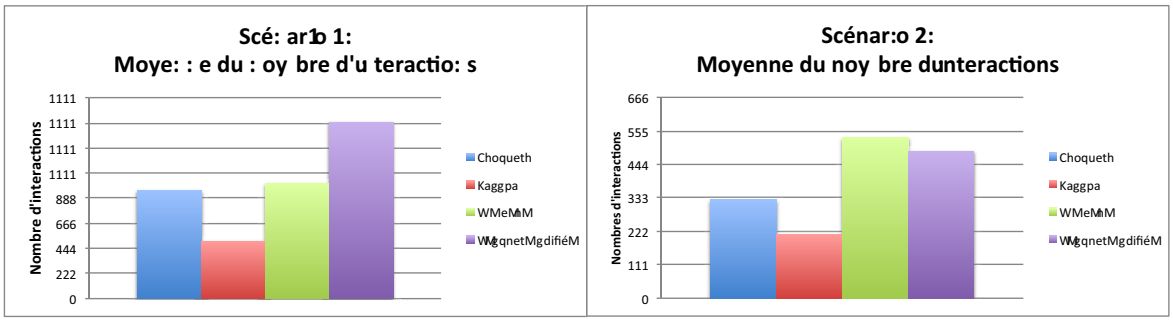


Figure 6.22 — Moyenne du nombre d'interactions

Un apprentissage de plus bas niveau implique plus de règles d'autorisation créées mais implique aussi un plus grand nombre d'interactions, ce que confirme les graphiques de la figure 6.22. Si pour le scénario 2, Choquet modifié interagit toujours moins avec l'utilisateur que la moyenne pondérée, ce n'est plus le cas sur le scénario 1. Les performances sont bien moins bonnes et l'avantage de prendre en compte les interactions entre critère est perdu par le manque d'abstraction par rapport aux autres opérateurs.

Le grand reproche qui peut être fait à Choquet sur la première phase de ce test est le grand nombre de règles d'autorisation erronées. L'utilisation de la nouvelle formule permet de résoudre en partie ce problème. Comme on peut le voir sur le graphique de la figure 6.23, dans la majorité des cas, le pourcentage de règles erronées retrouve une valeur identique

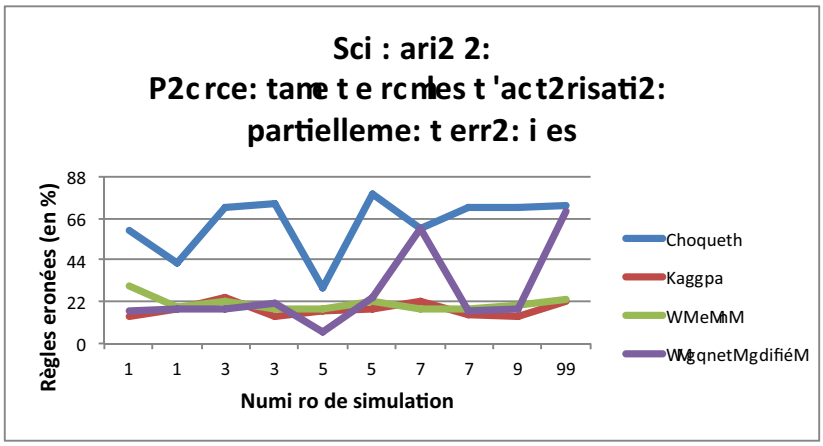


Figure 6.23 — Pourcentage de règles trop abstraites pour le scénario 2 en ajoutant la modification de Choquet

aux deux autres opérateurs. Malgré tout, deux simulations montrent encore un fort taux d'erreurs. En analysant certaines simulations en détail, on retrouve les raisons de ce fort taux d'erreurs. Par exemple si on regarde le critère *Ressource*. La Seule règle valable qui pourrait utiliser ce critère est "Les applications de type *Outils* peuvent avoir accès à toutes les *Ressource*" correspondant à la règle de comportement n°4. L'utilisation du critère *Ressource* dans un autre cas donne une règle erronée. Dans la simulation n°1, ce critère est proposé à tort dans une seule règle d'autorisation alors que dans la simulation n°7, ce critère est proposé dans 24 règles et 51 fois dans la simulation n°10. Dans ces deux simulations, l'apprentissage était toujours trop rapide sur les critères proposant le plus d'abstractions.

6.3.1.4 Conclusion du test

Pendant ces deux phases de test, nous avons testé trois méthodes d'apprentissage, chacune basée sur un opérateur d'agrégation différent. Chacune de ces méthodes a été testé sur deux scénarios, le premier utilisant une base de critères réduite et de faible hiérarchie de critères limitée à un seul niveau d'abstraction et le deuxième utilisant une base de critères plus large et une hiérarchie de critères plus complète sur plusieurs niveaux.

Les résultats confirment que dans ce problème d'aide à la décision multicritère, il y a interaction entre les critères. De ce fait, la méthode utilisant la moyenne pondérée n'obtient pas de bon résultats. Pour arriver à recréer la politique d'autorisation, elle a besoin de plus d'interactions et de plus de règles. Par contre les règles proposées sont pertinentes et l'apprentissage ne fait pas d'erreurs. Les deux autres opérateurs d'agrégation partaient avec un avantage vu qu'ils prennent en compte les interactions entre les critères. Pour autant, l'intégrale de Choquet n'obtient pas de résultats convaincants. Lors de la première phase de tests, les résultats sur le scénario 1 s'approchent de ceux de la moyenne pondérée. Les résultats étaient meilleurs sur le scénario 2 mais le fort taux de règles erronées ne peut pas donner satisfaction à l'utilisateur. Pour corriger ce dernier point, nous avons lancé une deuxième phase de test en modifiant la formule de mise à jour des méta-critères. Avec cette modification, le taux d'erreurs s'est amélioré. Mais si les résultats sur les autres métriques étaient toujours meilleurs sur le scénario 2, ils se sont totalement dégradés sur le scénario 1. On

peut donc en déduire que pour Choquet, la formule de mise à jour des méta-critères est dépendante de la base de critères, des hiérarchies et du comportement de l'utilisateur. Chaque instance du système devrait donc être paramétrée pour arriver à des résultats convenables. Hors il est impensable de demander à l'utilisateur de devoir paramétrer le système avant de l'utiliser. Kapuer est destiné à être utilisé par tout type d'utilisateurs et doit pouvoir fournir des résultats satisfaisant quel que soit le profil de l'utilisateur. Finalement, Kagop, l'opérateur d'agrégation que nous proposons, est l'opérateur qui obtient les meilleurs résultats. De plus ils sont stables sur toutes les simulations. Cette méthode est utilisable aussi bien sur une base de critères réduites qu'une base large et les règles d'autorisation propose plus d'abstractions que les autres méthodes permettant ainsi de recréer la politique d'autorisation de l'utilisateur avec moins d'interactions et moins de règles d'autorisations. L'effort de l'utilisateur est ainsi minimisé.

Maintenant que nous avons évalué les différents opérateurs d'agrégation, nous allons pouvoir passer à l'évaluation de Kapuer vis à vis des approches existantes. Étant donné qu'il présente les meilleurs résultats, nous utiliserons Kagop comme méthode d'apprentissage pour cette évaluation.

6.3.2 Evaluation de Kapuer

Bien que nous ayons évalué Kapuer et les différents opérateurs d'agrégation sur plusieurs métriques, nous ne pouvons toujours pas dire si Kapuer est une alternative intéressante pour l'utilisateur. Nous connaissons l'effort moyen demandé pour qu'un utilisateur puisse écrire sa politique d'autorisation et nous connaissons aussi le taux de propositions qui satisferont l'utilisateur. Il nous reste à comparer l'effort demandé par Kapuer par rapport aux autres solutions déjà existantes. Dans la première partie de ce manuscrit (voir section 2.2), nous avons étudié deux autres solutions permettant à l'utilisateur de définir sa politique d'autorisation :

— CyanogenMod qui utilise une interface graphique, facile à utiliser et accessible à tous.
— XACML, un langage de contrôle d'accès permettant de définir une politique d'autorisation précise mais nécessitant des connaissances techniques.

Il est incontestable que XACML ne peut pas être utilisé directement par le grand public. Sans apprendre le langage, il est impossible de pouvoir écrire sa politique d'autorisation. Donc bien que cela reste une solution, elle n'est pas envisageable dans notre cas. La seule solution restante reste un système comme CyanogenMod.

Nous avons donc utilisé la distribution alternative d'Android pour voir quel était l'effort nécessaire pour écrire notre politique d'autorisation composée des huit règles définies précédemment dans ce chapitre. Nous avons compté le nombre de pressions du doigt nécessaires pour recréer chaque règle de comportement. Lorsque Kapuer est implémenté sur un smartphone, les interactions avec l'utilisateur sont aussi effectuées avec une pression du doigt. Il nous est donc aisé de comparer le nombre de pressions et donc l'effort nécessaire pour chaque méthode. Par exemple, pour recréer la règle n°1 avec CyanogenMod, l'utilisateur doit d'abord choisir l'application, soit une pression et ensuite choisir si la permission

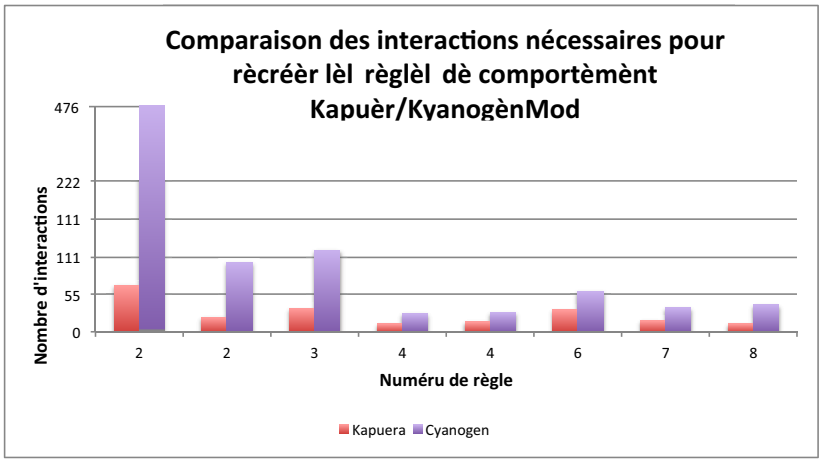


Figure 6.24 — Comparaison du nombre d'interactions nécessaire pour recréer les règles de comportement

doit être accepté ou refusé donc une pression de nouveau par permission. La première règle accorde l'accès à *Internet* pour tous les *Jeux*. Cela correspond à 28 applications, 6 permissions à accepter et 10 permissions à refuser, soit 16 pressions par applications et un total de 476 pressions pour accéder aux 28 applications et choisir les 16 permissions. Le graphique de la figure 6.24 présente les résultats sur les huit règles en comparant CyanogenMod à Kapuer en utilisant Kagop qui est la méthode nécessitant le moins d'interactions avec l'utilisateur. Nous pouvons voir que Kapuer demande moins d'effort à l'utilisateur pour toutes les règles. Au total cela représente 190 pressions pour Kapuer contre 848 pour CyanogenMod. Les autres opérateurs d'agrégation, même en ayant besoin de plus d'interactions avec l'utilisateur sont quand même plus intéressant pour l'utilisateur en terme d'effort que CyanogenMod (295 pour Choquet et 480 pour le moyenne pondérée).

6.4 Conclusion

L'utilisation d'un simulateur nous a permis d'évaluer plusieurs aspects de Kapuer. Alors que ce n'était pas possible avec le prototype, nous avons pu tester et comparer les trois opérateurs d'agrégation sur des milliers de requêtes (100 000 requêtes, 50 000 par scénario). Les résultats des simulations ont permis de démontrer que l'utilisation d'un opérateur d'agrégation prenant en compte les possibles interactions entre critères est indispensable pour avoir de meilleures résultats. Ainsi la moyenne pondérée propose des règles d'autorisation valables mais manque d'abstraction et de vitesse. L'intégrale de Choquet, bien que prenant en compte les interactions entre les critères, nécessite un paramétrage précis dépendant du profil de l'utilisateur et de la base de critères utilisée. Mais il est impossible de laisser ce paramétrage à l'utilisateur donc cette méthode n'est pas non plus la plus appropriée. L'opérateur d'agrégation que nous proposons, Kagop, obtient les meilleurs résultats, quel que soit la base de critères utilisée. De plus, ces résultats restent stable tout au long des simulations et Kagop n'affiche qu'un faible taux de règles d'autorisation erronées.

Ces simulations ont permis de démontrer que Kapuer pourrait être très utile pour ai-

der un utilisateur a écrire sa politique d'autorisation. L'utilisation des différentes notions autour des critères permet un apprentissage précis des préférences de l'utilisateur et celle d'XACML qui reste totalement transparente pour l'utilisateur permet d'écrire les différentes règles d'autorisation avec un niveau d'abstraction qui peut différer pour réduire l'effort de l'utilisateur. En effet, l'utilisation seule d'XACML n'est pas possible pour le grand public et les résultats ont prouvé qu'un système classique comme CyanogenMod demande un effort important à l'utilisateur alors que cet effort est réduit si l'utilisateur utilise Kapuer comme assistant pour écrire sa politique d'autorisation.

Conclusion

« You have to start with the truth. The truth is the only way that we can get anywhere. Because any decision-making that is based upon lies or ignorance can't lead to a good conclusion. »

Julian Assange

L'OBJECTIF de cette thèse était de trouver un moyen d'aider un utilisateur à protéger efficacement ses données de vie privée. Avec l'explosion de l'Internet des Objets, la quantité d'informations disponible sur le réseau a été décuplée. Mais les échanges de données ont eux aussi fortement augmenté. Il est nécessaire pour tout un chacun de pouvoir protéger sa vie privée en contrôlant qui peut avoir accès à ses données. Le contrôle d'accès est un sujet largement étudié où de nombreux modèles existent et permettent d'exprimer, chacun avec son cadre et ses catégories, une politique d'autorisation. Alors qu'il est déjà compliqué de choisir le modèle adéquat, il n'est pas possible pour le grand public, qui n'a pas les connaissances nécessaires dans les langages de contrôle d'accès, d'écrire sa propre politique d'autorisation. Il existe bien des systèmes basés sur des interfaces graphiques permettant à un utilisateur de gérer sa politique d'autorisation sans avoir besoin de connaissances spécifiques. Mais ces méthodes manquent de flexibilité, le moindre changement dans le modèle implique un changement de l'interface graphique qui peut dérouter l'utilisateur. De plus ces interfaces graphiques ne sont pas efficaces pour gérer un grand nombre d'autorisations. La gestion de la politique d'autorisation devient alors longue et fastidieuse.

C'est pourquoi nous avons utilisé les systèmes d'aide à la décision, et plus particulièrement un système de recommandation pour assister l'utilisateur dans l'écriture de sa politique d'autorisation. Un système de recommandation permet un apprentissage adaptatif des préférences de l'utilisateur afin de lui proposer des éléments de solution. Dans notre cas, le système apprend ce que veut faire l'utilisateur avec ses données et lui propose des règles sur mesure pour compléter sa politique d'autorisation.

La contribution principale de cette thèse est Kapuer : un framework générique permettant de développer des systèmes de recommandation pour protéger les données de vie privée des utilisateurs. Couplé à un système de contrôle d'accès basé sur les attributs, Kapuer offre la liberté aux développeurs de créer des classes de critères selon les informations disponibles. Il est ainsi possible de s'adapter à chaque type de périphériques et aussi bien porter

Kapuer sur un smartphone, une tablette, un navigateur web ou des lunettes connectées. Le développeur pourra utiliser les catégories qui lui conviennent pour créer des abstractions sur chaque classe de critères. Pour cela il doit créer des méta-critères et ensuite construire des hiérarchies de critères. La phase la plus importante de Kapuer est l'apprentissage des préférences. Cet apprentissage est effectué au moyen d'interactions entre le système et l'utilisateur. Nous avons utilisé une analyse multicritère sur les demandes d'accès et les décisions prises par les utilisateurs lors des différentes interactions pour apprendre les préférences de l'utilisateur. Pour cela, nous avons étudié plusieurs opérateurs d'agrégation et nous en avons proposé un nouveau : Kagop (*Kapuer AGgregation OPerator*). Cet opérateur est une autre contribution de cette thèse. Il utilise les combinaisons de critères et méta-critères de classes différentes, appelées groupes de critères, pour découvrir les interactions entre les critères. Bien que la création et l'utilisation de ces groupes de critères peuvent provoquer une explosion combinatoire, la comparaison avec les autres opérateurs a montré que Kagop obtenait de meilleurs résultats et permettait de créer des politiques d'autorisation plus proches de ce que veut l'utilisateur tout en minimisant le nombre d'interactions.

Les deux autres contributions de cette thèse sont le simulateur et le prototype Android. Le simulateur nous a permis de faire de nombreux tests qui n'auraient pas été possibles autrement faute de matériel et de temps. De plus il nous a permis de comparer plusieurs opérateurs d'agrégation et d'analyser les apprentissages sous diverses métriques aisément. Le prototype quant à lui montre la faisabilité technique de porter Kapuer dans un environnement mobile. Il implémente Kapuer et utilise XACML pour la gestion du contrôle d'accès.

Perspectives

Mais il reste encore beaucoup de travail à effectuer pour améliorer Kapuer. Les travaux s'ouvrent à de nombreuses perspectives, à court et moyen terme mais aussi à long terme.

A court et moyen terme

Le simulateur est un outil intéressant mais qui pourrait être largement amélioré. Pour le moment, il permet de lancer des simulations et de visualiser les résultats mais tous les paramètres de la simulation doivent être réglés au préalable directement dans le code source. Il peut être beaucoup plus automatisé et ainsi ne pas nécessiter de devoir modifier le code pour faire certains changements. Le simulateur pourrait ainsi être complété avec des fonctionnalités telles que :

— l'ajout, la modification ou la suppression de critères de la bases de critères.
— la sauvegarde de simulations afin de pouvoir la recharger et la rejouer.
— la visualisation et la modification des différentes hiérarchies de critères.
— la création des règles de comportement.

Des modifications peuvent aussi être faites sur le fonctionnement des opérateurs d'agrégation. Implémenter ces opérateurs comme des composants logiciels OSGi permettrait au

simulateur de devenir une vraie plateforme d'apprentissage. Il serait alors possible de l'utiliser pour analyser, évaluer et comparer d'autres méthodes d'apprentissage.

Kapuer a été évalué avec le simulateur mais pas encore en situation réelle. Améliorer le prototype Android en prenant en compte tous les détails de Kapuer permettra de faire des tests avec de vrais utilisateurs. Actuellement le prototype utilise XACML V2. Passer à la V3 permettra de pouvoir utiliser la notion de catégorie introduite dans cette version qui est équivalente à notre notion de classe et qui nous permettra d'utiliser plus facilement les différents types d'information.

Nous avons aussi commencé à travailler sur l'initialisation en lançant des enquêtes auprès de divers publics (voir annexe C). Cette phase d'initialisation pourrait permettre de connaître les bases des préférences de l'utilisateur et avoir une idée de son profil général. Ainsi si cette phase révèle que l'utilisateur est plutôt du style à accepter la majorité des requêtes ou au contraire à les refuser, le système pourrait adapter sa vitesse d'apprentissage pour proposer plus rapidement des règles d'autorisation et avoir aussi un apprentissage plus centré sur les méta-critères pour abstraire au maximum les règles proposées. Au contraire si l'initialisation révèle que l'utilisateur semble pointilleux sur sa politique d'autorisation, le système adapterait sa vitesse pour apprendre plus finement ses préférences pour proposer des règles moins abstraites mais plus pertinentes.

L'adaptation de la vitesse d'apprentissage et du niveau d'abstraction proposé est aussi une piste de réflexion. Par exemple, nous avons vu dans l'évaluation que de nombreuses propositions étaient erronées car trop abstraites. Le système pourrait s'adapter après plusieurs refus de l'utilisateur et baisser la valeur de mise à jour des méta-critères de plus haut niveau pour éviter de continuer à proposer trop de règles trop abstraites. Et au contraire, si l'utilisateur accepte toutes les règles proposées, augmenter la valeur de mise à jour des méta-critères pour proposer moins de règles. Cela permettrait d'avoir une vitesse d'apprentissage adaptée à chacun.

A long terme

Les méta-critères sont extrêmement utiles à Kapuer car ils apportent de l'abstraction sur les règles et permettent donc de créer des règles de plus haut niveau. Mais il peut arriver que certains méta-critères ne correspondent pas à une abstraction voulue par l'utilisateur et deviennent inutiles. Un axe de recherche serait de voir si le système ne peut pas créer de nouveaux méta-critères adaptés au comportement de l'utilisateur. Par exemple si un méta-critère référence 10 critères, le système pourrait créer un nouveau niveau de méta-critères avec 2 nouveaux méta-critères référençant chacun 5 critères s'il voit que cela correspond mieux au comportement de l'utilisateur. Ainsi, les hiérarchies de critères s'adapteraient en fonction de l'utilisateur.

Un autre axe de recherche pourrait toujours concerner les méta-critères. Actuellement, au moment de l'implémentation de Kapuer, le développeur doit créer les hiérarchies de critères pour chaque classe. Le système pourrait se passer de ces hiérarchies et approfondir l'apprentissage pour déduire les méta-critères à utiliser et créer pendant l'exécution les hiérarchies les plus adaptées.

Actuellement, Kagop obtient de bons résultats mais peut engendrer des problèmes de mémoire. L'utilisation des groupes de critères formés à partir de toutes les combinaisons possibles de critères peut créer une explosion combinatoire avec une très grosse base de critères. Travailler sur Kagop permettrait de voir si tous les groupes de critères sont vraiment nécessaires et s'il n'est pas possible d'optimiser leur utilisation.

Première partie

Annexes

A Graphique supplémentaires du scénario 1

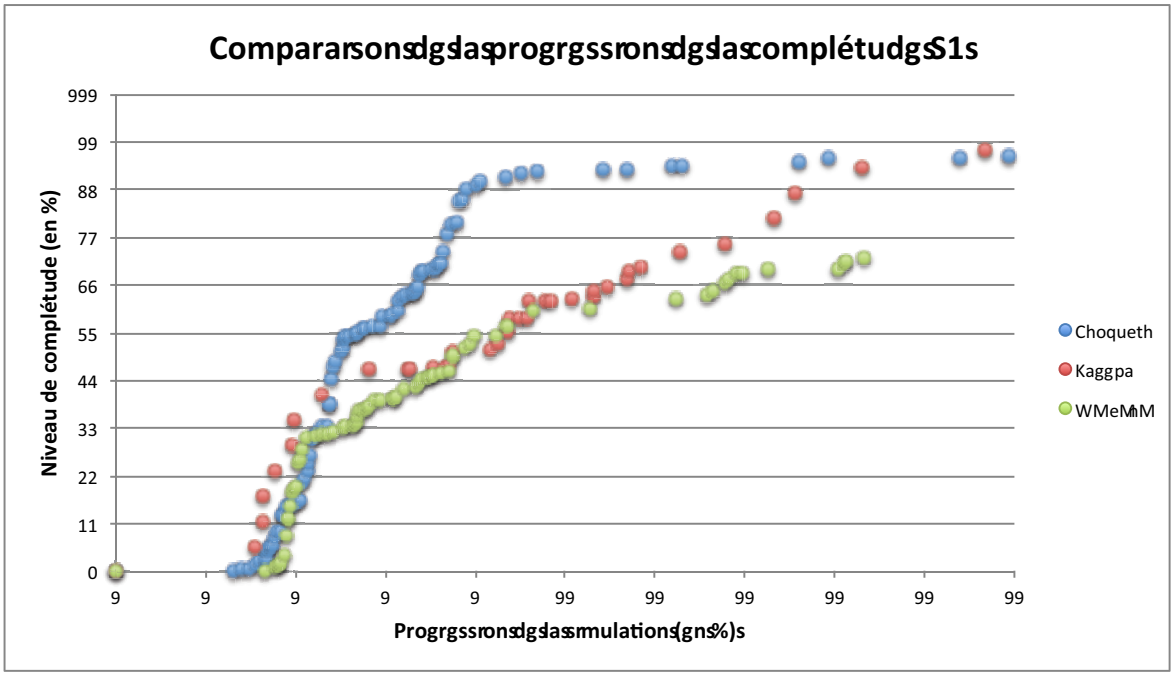


Figure A.1 — Comparaison de la progression de la complétude : simulation 1

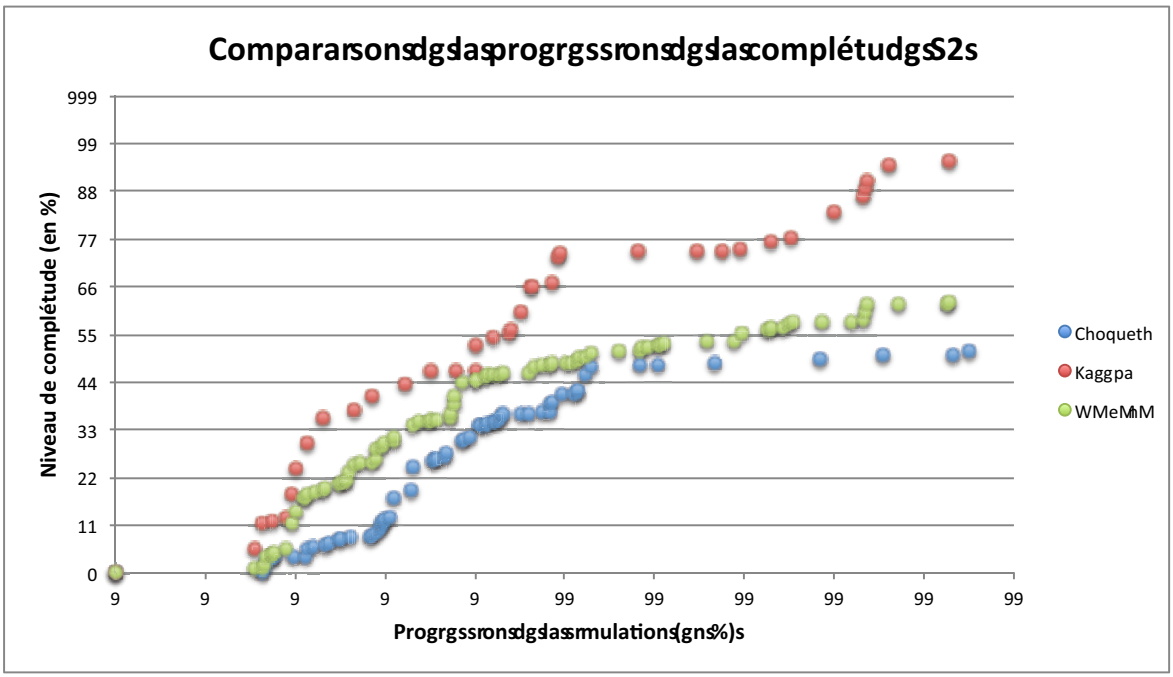


Figure A.2 — Comparaison de la progression de la complétude : simulation 2

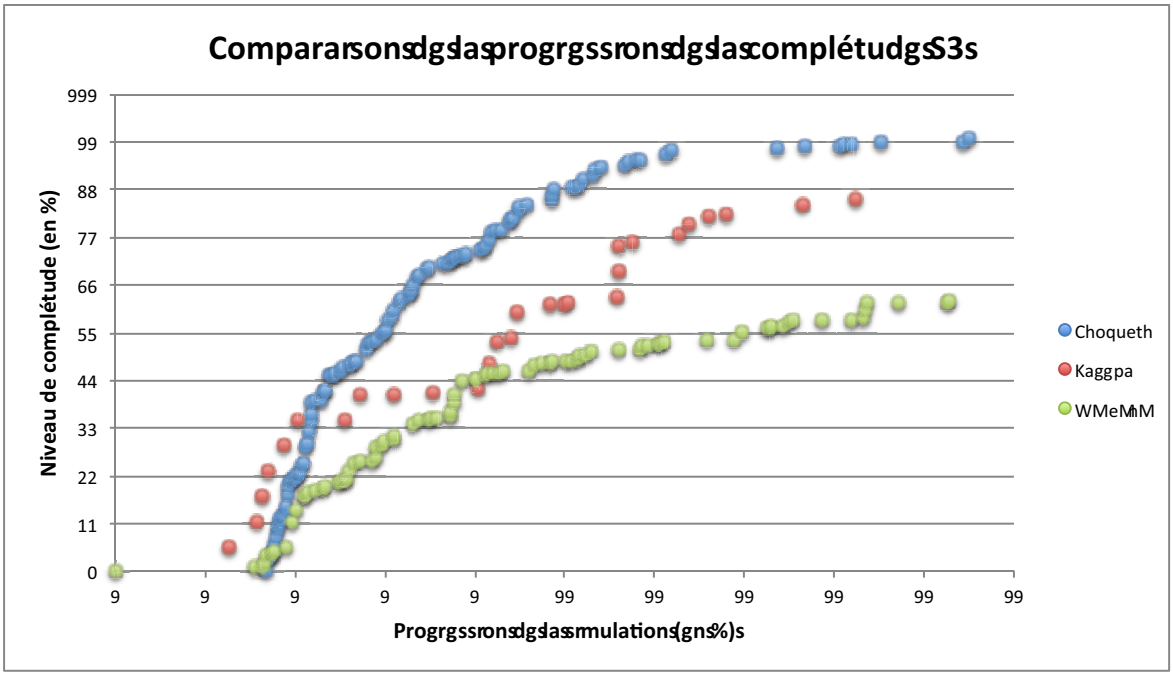


Figure A.3 — Comparaison de la progression de la complétude : simulation 3

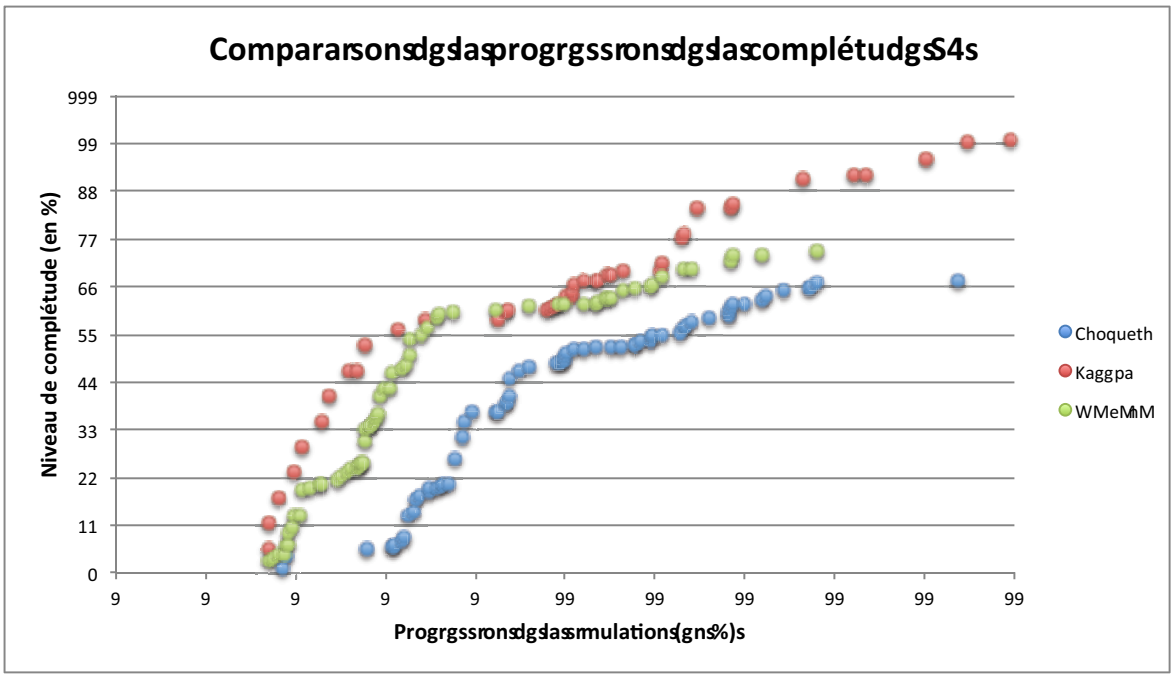


Figure A.4 — Comparaison de la progression de la complétude : simulation 4

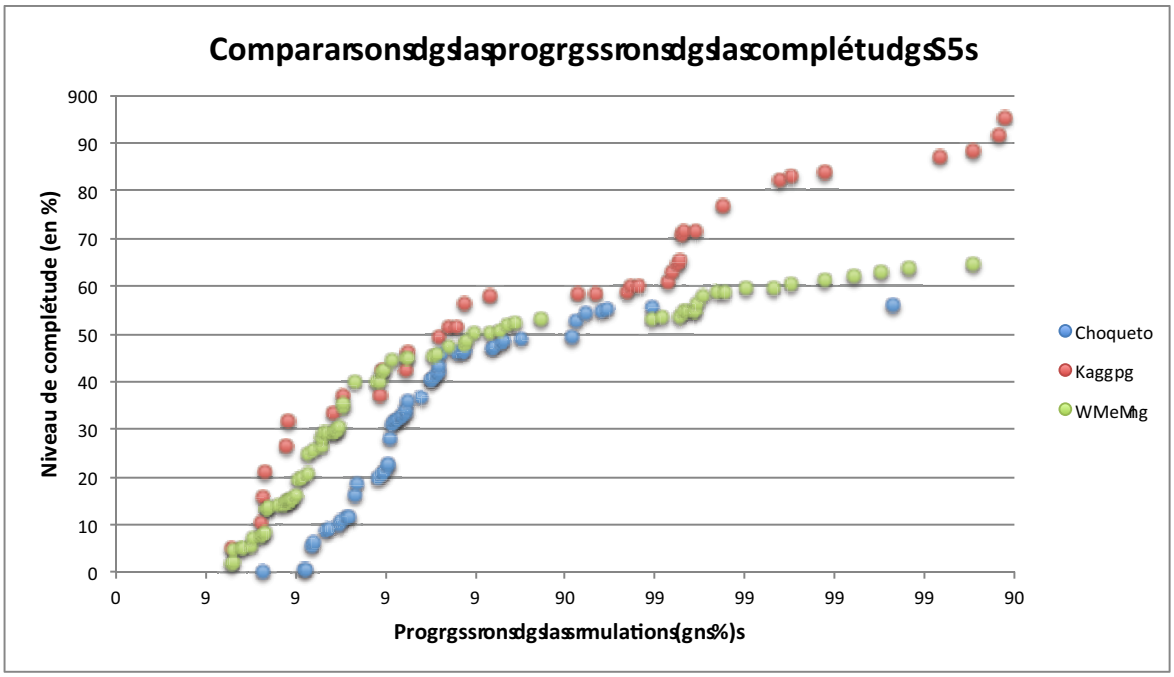


Figure A.5 — Comparaison de la progression de la complétude : simulation 5

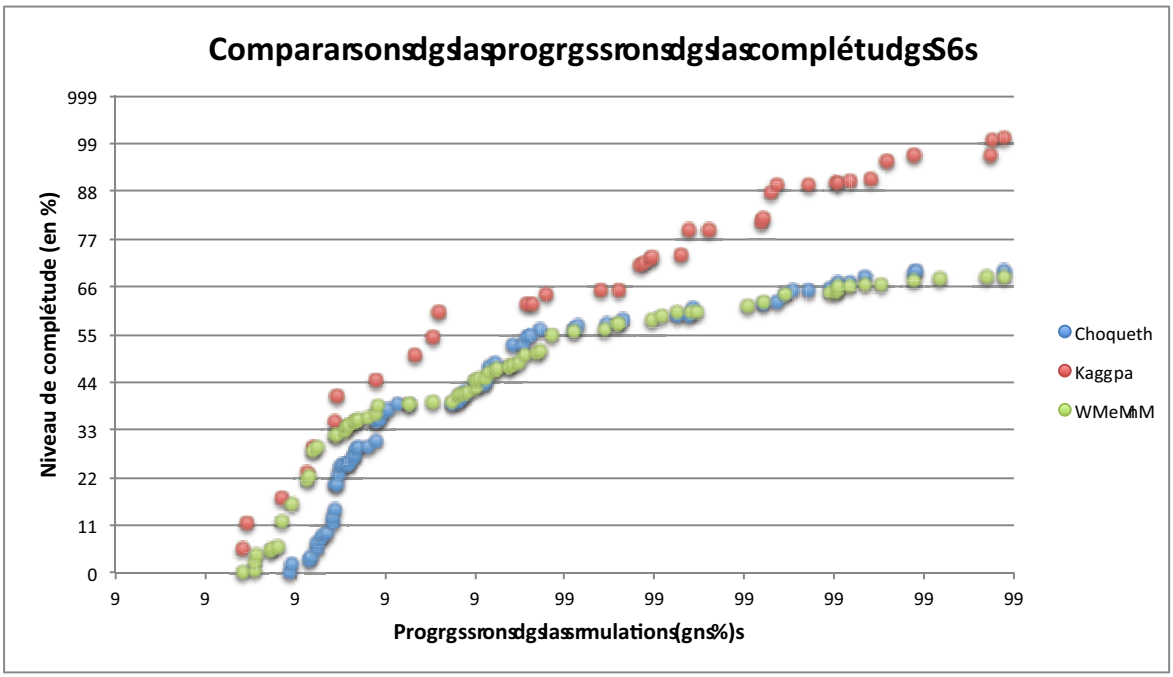


Figure A.6 — Comparaison de la progression de la complétude : simulation 6

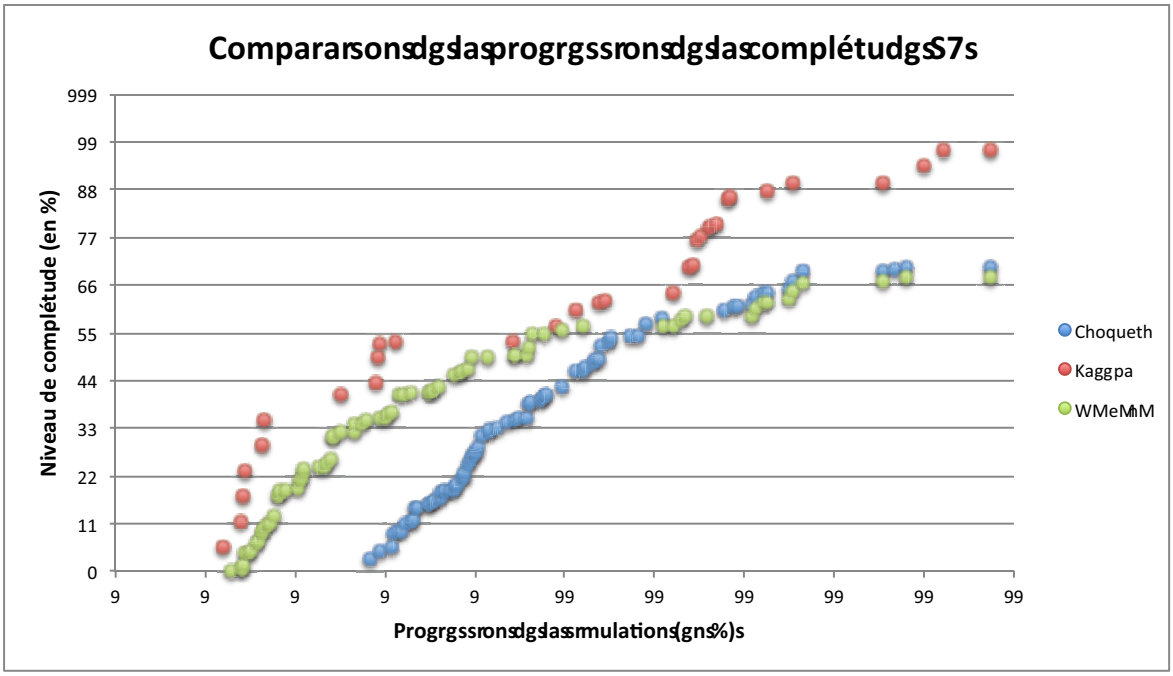


Figure A.7 — Comparaison de la progression de la complétude : simulation 7

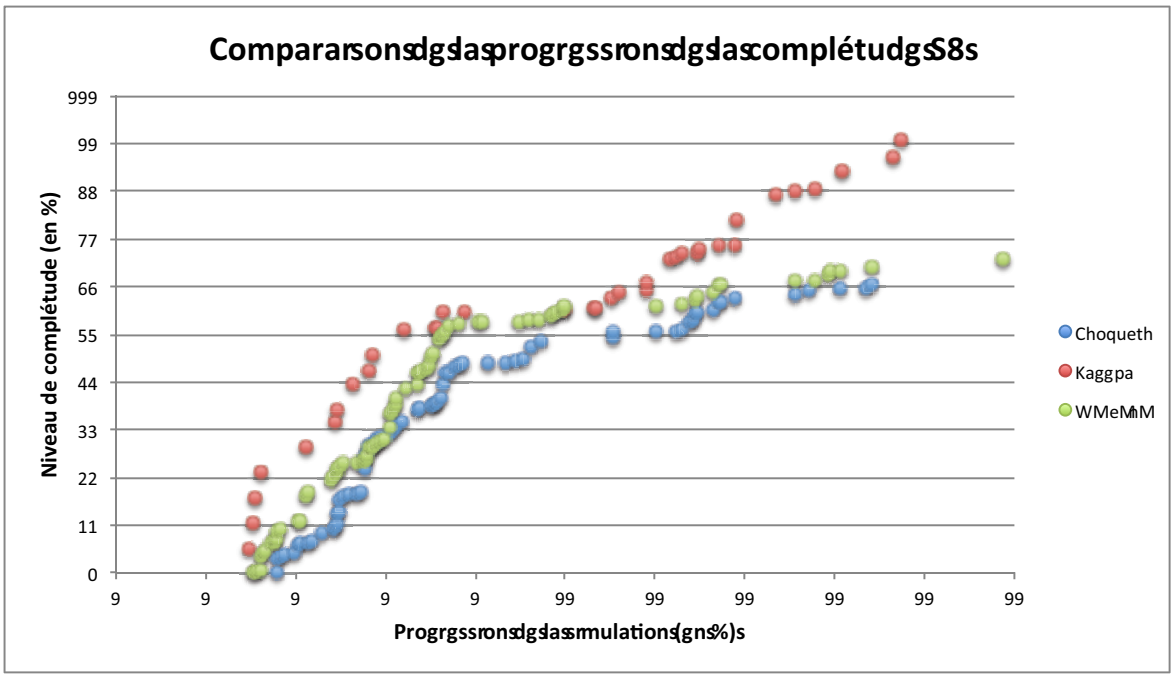


Figure A.8 — Comparaison de la progression de la complétude : simulation 8

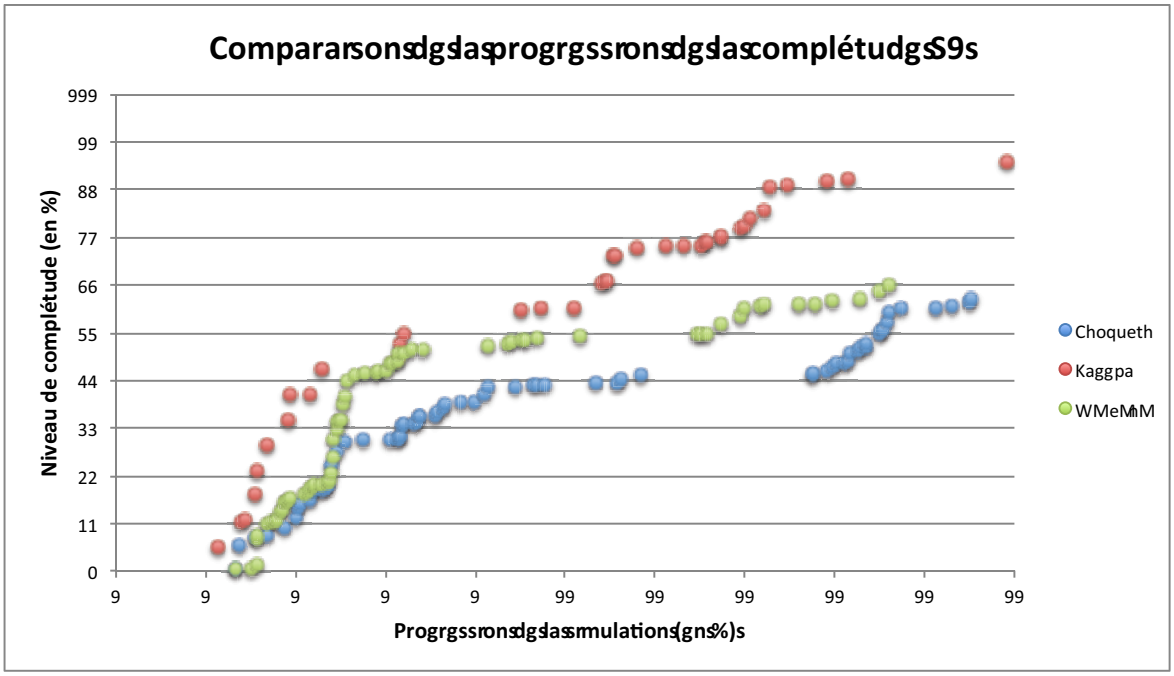


Figure A.9 — Comparaison de la progression de la complétude : simulation 9

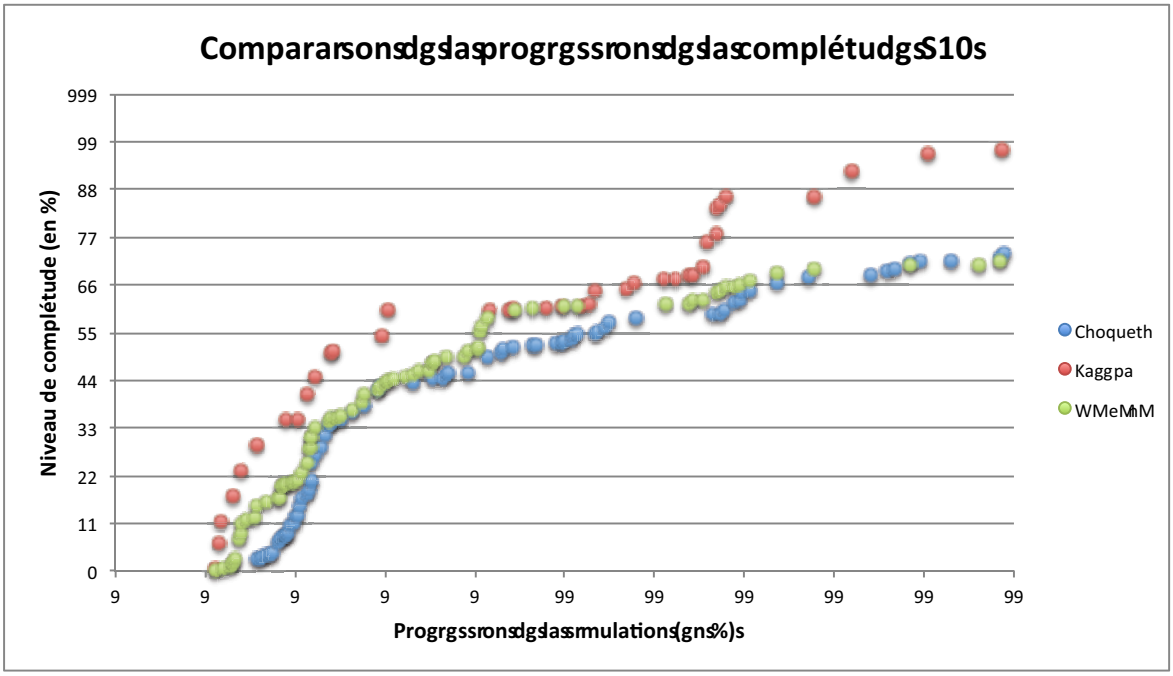


Figure A.10 — Comparaison de la progression de la complétude : simulation 10

B Graphique supplémentaires du scénario 2

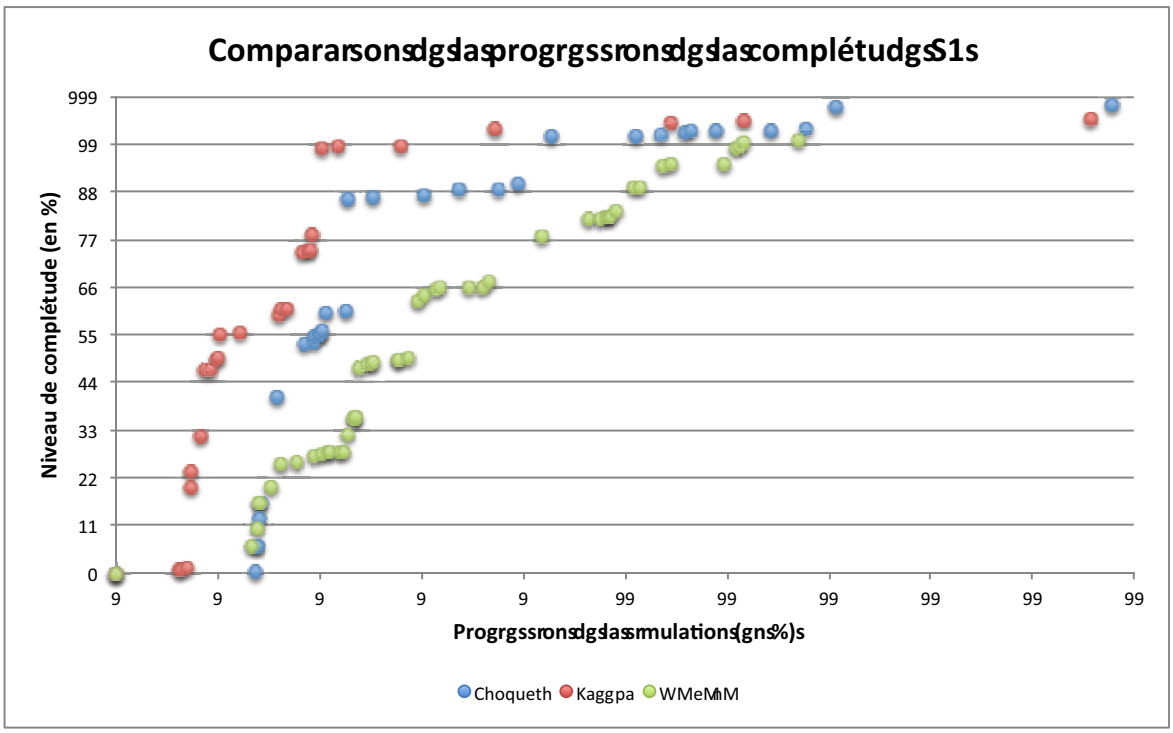


Figure B.1 — Comparaison de la progression de la complétude : simulation 1

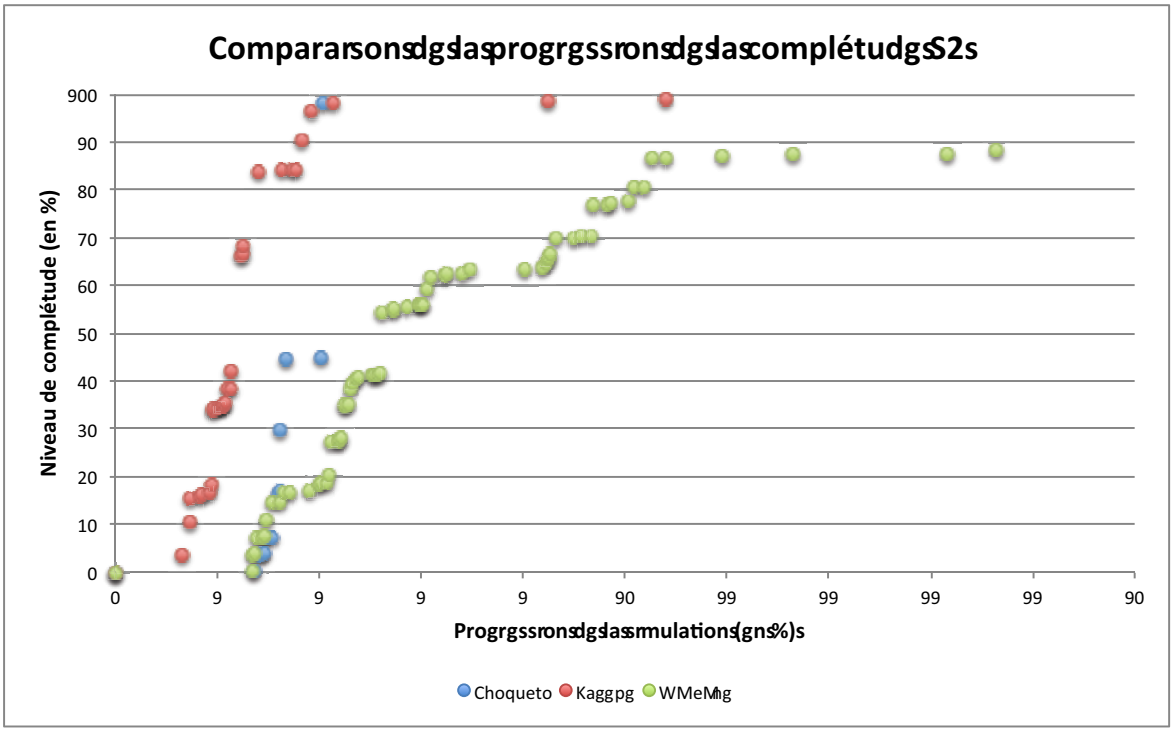


Figure B.2 — Comparaison de la progression de la complétude : simulation 2

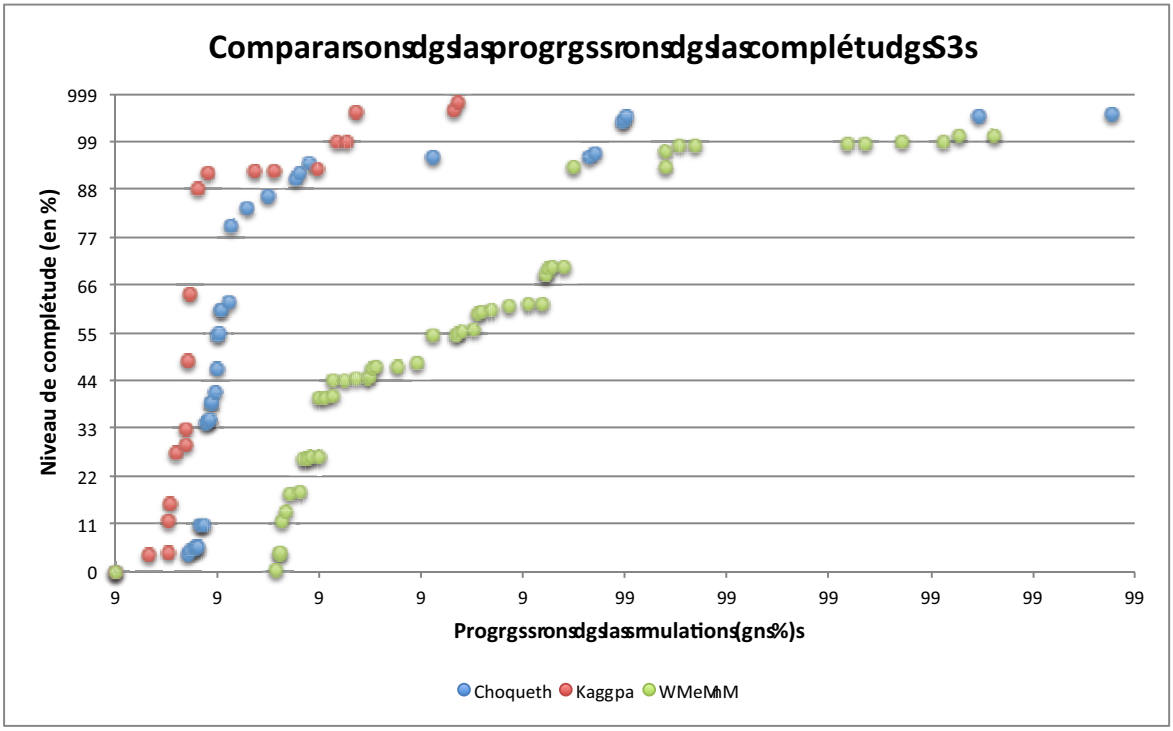


Figure B.3 — Comparaison de la progression de la complétude : simulation 3

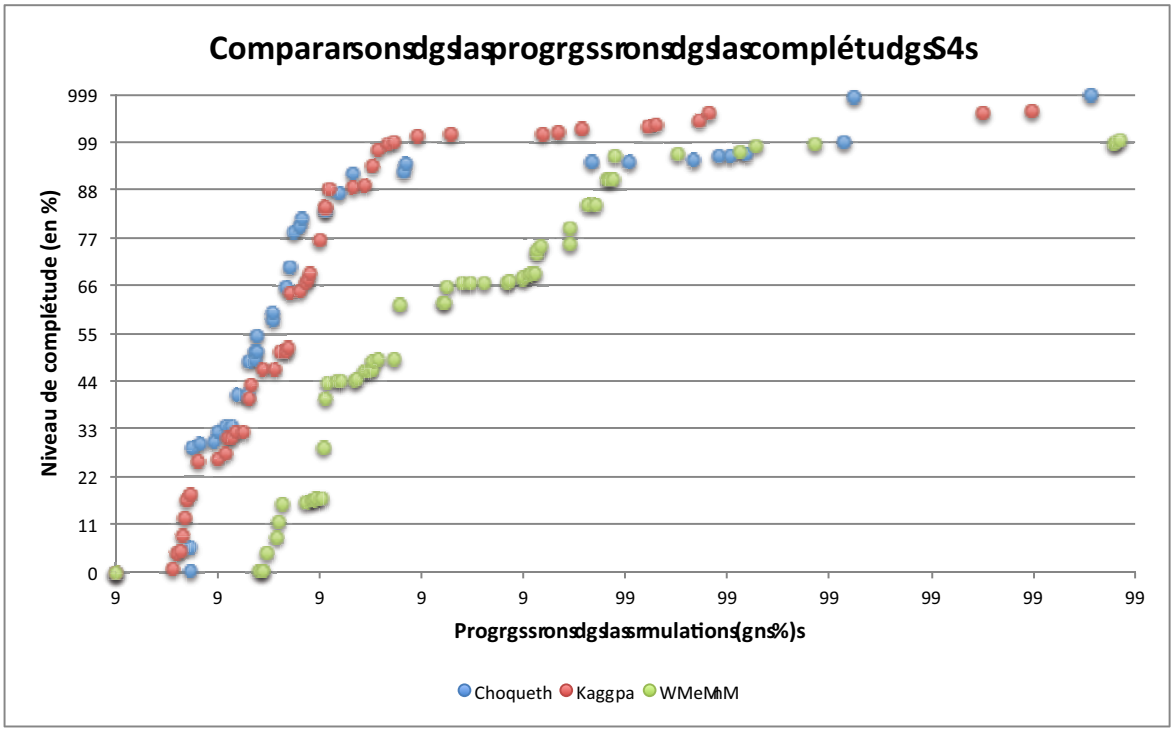


Figure B.4 — Comparaison de la progression de la complétude : simulation 4

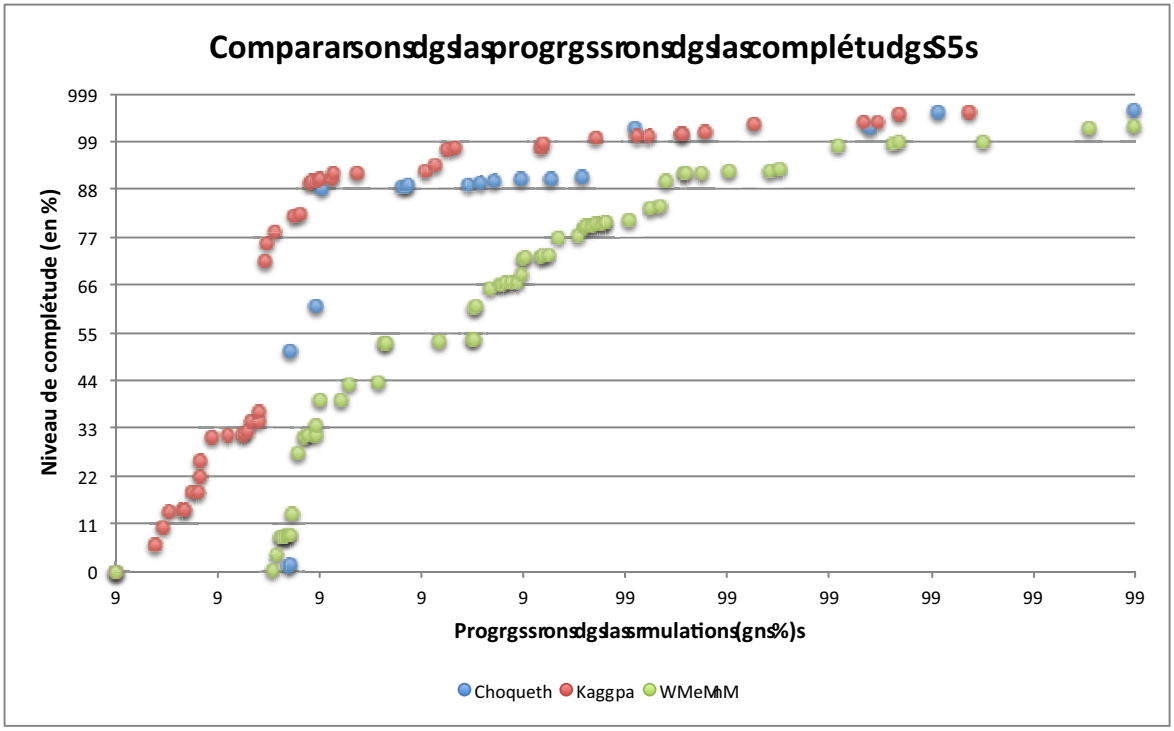


Figure B.5 — Comparaison de la progression de la complétude : simulation 5

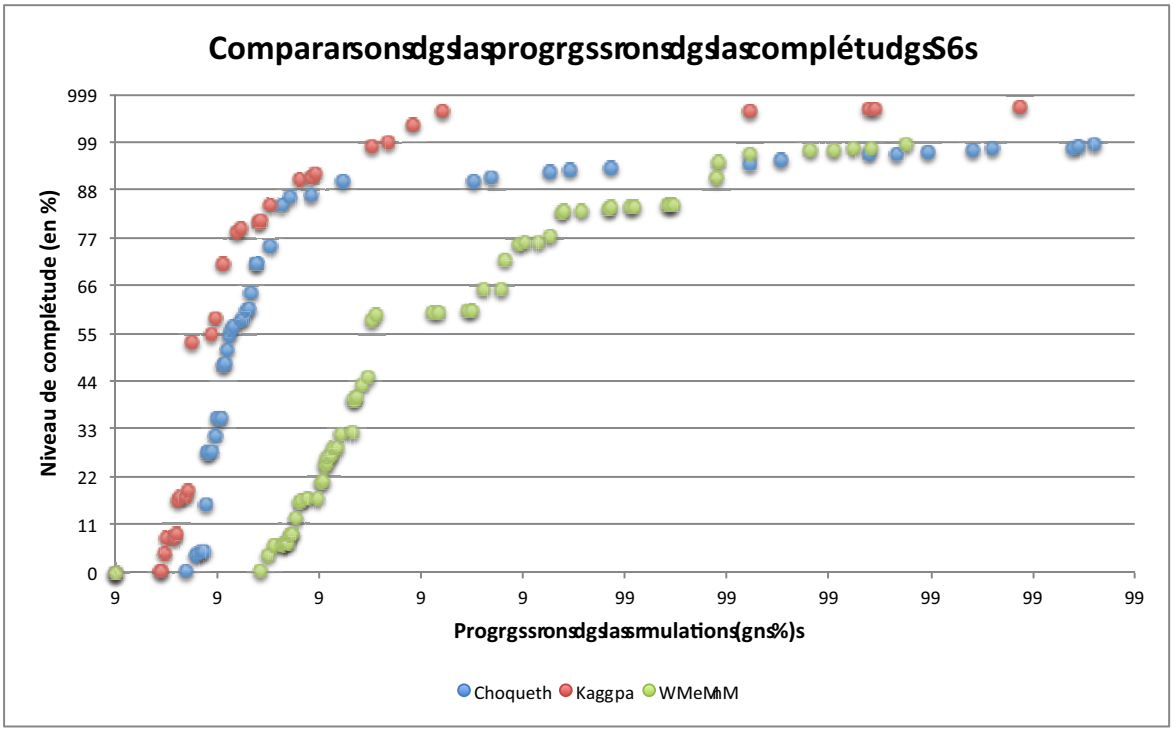


Figure B.6 — Comparaison de la progression de la complétude : simulation 6

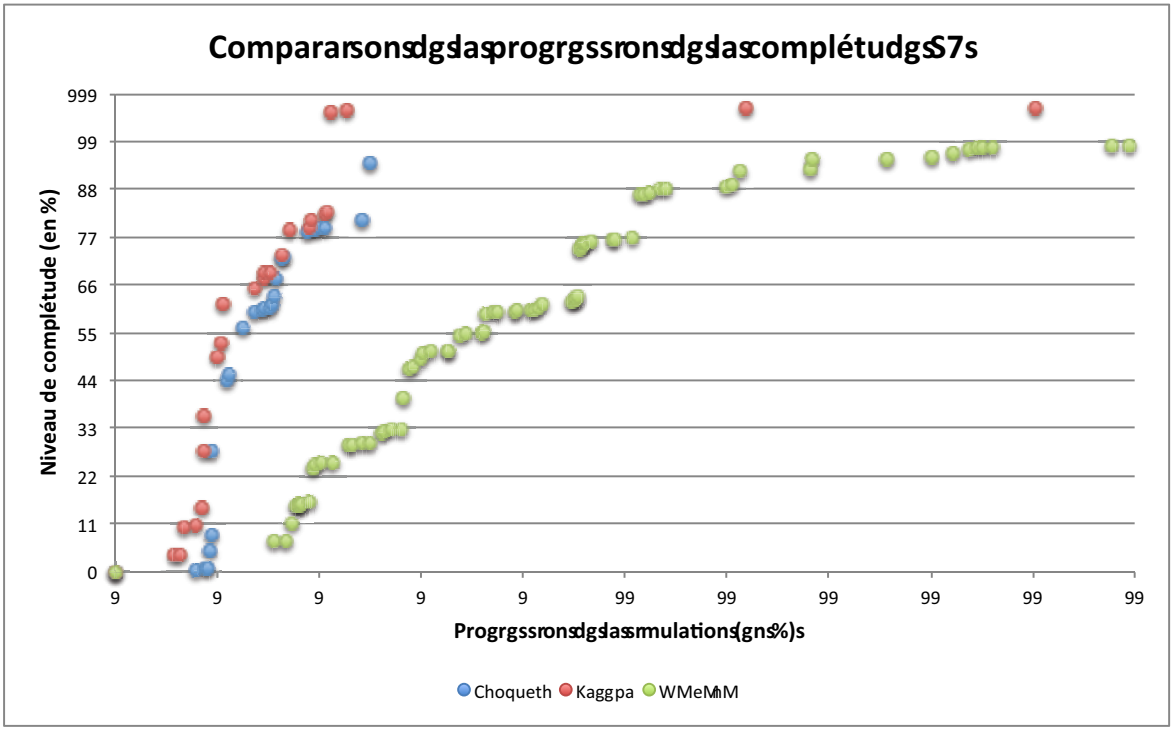


Figure B.7 — Comparaison de la progression de la complétude : simulation 7

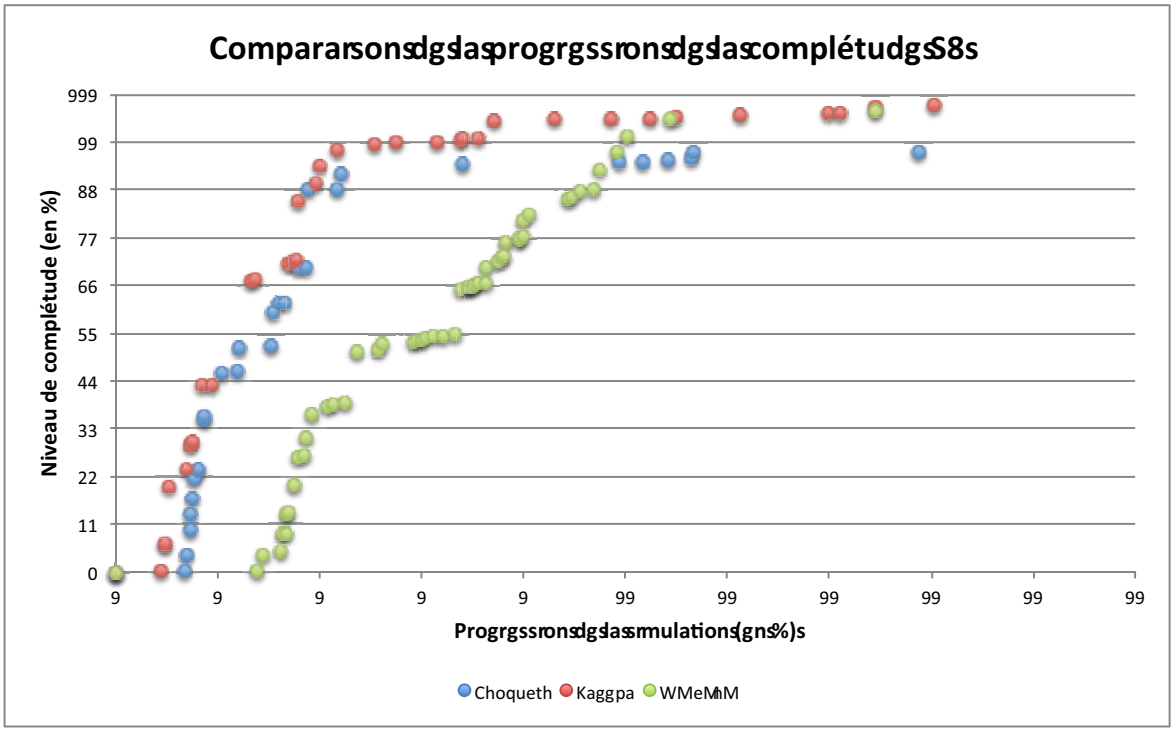


Figure B.8 — Comparaison de la progression de la complétude : simulation 8

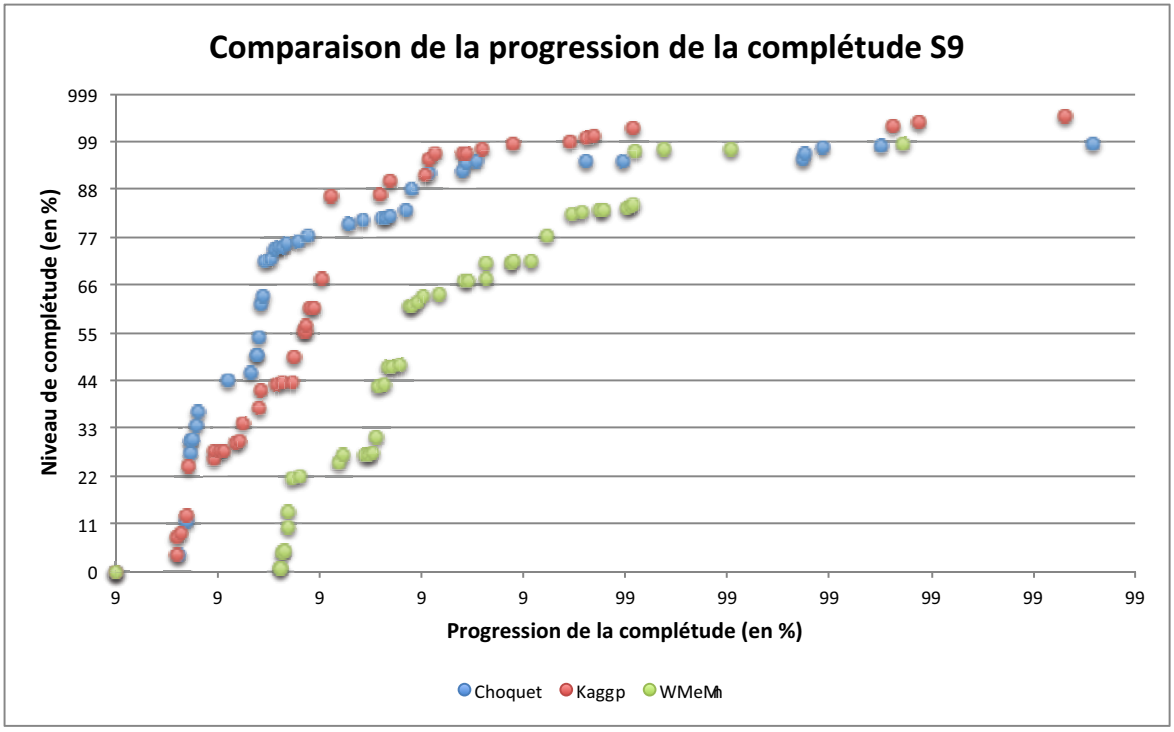


Figure B.9 — Comparaison de la progression de la complétude : simulation 9

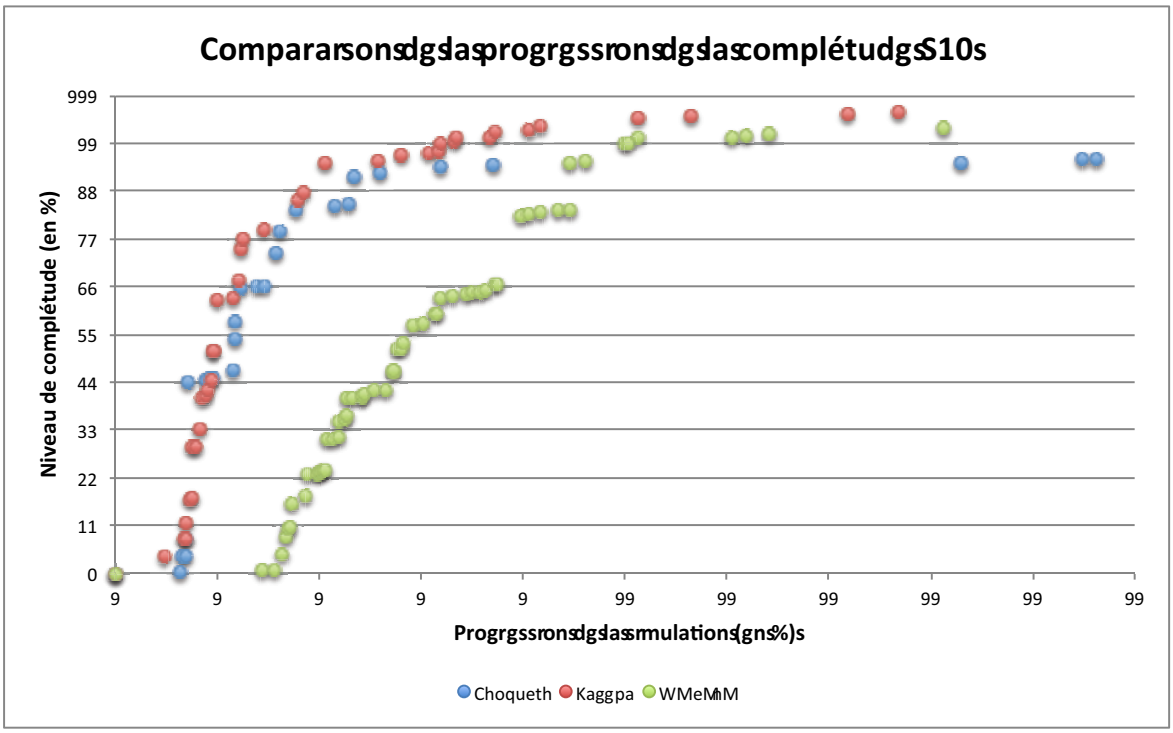


Figure B.10 — Comparaison de la progression de la complétude : simulation 10

C Enquête pour l'initialisation

Smartphone & Vie Privée

Un questionnaire sur l'utilisation de votre smartphone et votre façon d'agir quand on vous demande des informations vous concernant.

Il y a 42 questions dans ce questionnaire

Présentation

1 [1]Quel age avez vous ? *

Veuillez écrire votre réponse ici :

2 [2]A quel groupe de personnes appartenez vous ? *

Veuillez sélectionner une seule des propositions suivantes :

- ○ Informaticien
- ○ Etudiant initié aux problèmes de vie privée
- ○ Autre

Quizz

3 [G2_Q0001]

En général, quand on vous demande de partager des informations concernant votre identité (nom/prénom, adresse, n° de téléphone, ...), que faites vous ?

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ J'évite autant que possible.
- ○ Je ne renseigne que le strict nécessaire, ne remplissant pas les informations facultatives.
- ○ Je donne toutes les informations qu'on me demande.

4 [G2_Q0002]

Une application me notifie qu'un ami me demande mon adresse mail...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ C'est quelqu'un que je connais, je n'ai pas de problème à lui donner.
- ○ Je refuse.

5 [G2_Q0003]

Lors de l'installation d'une application, vous devez créer un compte et un paragraphe vous informe que vos informations pourraient être utilisées à des fins statistiques...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ J'ai confiance en l'application, l'utilisation de mes informations pour des statistiques ne me dérange pas, je continue l'installation.
- ○ Même si le but d'avoir ces informations n'avait pas été renseigné, j'aurai quand même créé le compte et donné mes informations.
- ○ Si la création du compte est obligatoire, j'arrête l'installation et je me passerais de l'application.

6 [G2_Q0004]

En lisant les conditions d'utilisation d'une application, vous apprenez que toutes vos informations seront stockées pendant une courte période sur des serveurs appartenant à l'entreprise qui développe l'application...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ Savoir cela va changer ma façon d'utiliser l'application.
- ○ Je vais continuer d'utiliser l'application comme avant.

7 [G2_Q0006]

Une de vos applications désire connaître votre adresse pour savoir quand vous êtes chez vous et en déduire de vos déplacements où se trouve votre lieu de travail pour lui permettre de ne pas vous importuner lorsque vous travaillez...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ Cela m'intéresse, je ne veux pas être dérangé pendant que je travaille.
- ○ Je ne veux pas qu'on connaisse mon lieu de domicile, mon lieu de travail, etc.
- ○ Je n'ai pas un avis tranché sur la question.

8 [G2_Q0007]

Que pensez vous de tout ce qui touche à vos données d'ordre privées ? (photos, mails, liste de contacts, ...)

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ C'est strictement confidentiel, ça ne regarde que moi.
- ○ Je peux diffuser certaines informations, selon la situation.
- ○ Généralement, si on me demande de partager, je le fais.

9 [G2_Q0008]

Un de mes collègues souhaite accéder à mes agendas pour pouvoir programmer une réunion...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ Je n'ai qu'un seul agenda qui contient des évènements privés, je ne souhaite pas le partager.
- ○ J'ai plusieurs agendas, celui concernant le travail est public, je peux donc lui donner accès.
- ○ Je n'ai rien contre lui donner accès à mes agendas pour lui faciliter la gestion de son travail.

10 [G2_Q0009]

Mon réseau social m'informe que toutes les photos que je transférerai dessus seront stockées indéfiniment sur leurs serveurs...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ Je me désinscris, je ne veux pas que des photos de moi puissent ressortir dans le futur.
- ○ Je fais attention aux photos je transfère, essayant d'éviter de mettre des photos où j'apparais.
- ○ Je continue de mettre toutes mes photos de soirées, de voyage, etc.

11 [G2_Q0010]

En arrivant sur un site web, on vous informe que vos actions sur le site seront analysées pour mieux vous conseiller lors de prochaines visites...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ Je trouve utile que le site essaie de mieux cibler mes attentes
- ○ Je suis un peu réticent mais si il ne fait qu'analyser mes actions, je continue à utiliser ce site
- ○ J'essaie de trouver un autre site web proposant les mêmes services mais sans cette option.
- ○ Je n'y prête pas attention.

12 [G2_Q0011]

Si cela doit vous simplifier l'utilisation de votre smartphone, autoriseriez vous une application à déclencher l'utilisation de certains services ? (GPS, appareil photo, musique, Wifi, etc.)

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ Non je dois rester maitre de l'utilisation de mon smartphone.
- ○ Selon le contexte, je peux l'autoriser mais il faut que je donne mon accord au moins la première fois.
- ○ Si ça doit me simplifier la vie, bien sur.

13 [G2_Q0012]

Vous êtes dans un lieu touristique et une application de réseau social vous propose de mettre en statut l'endroit où vous vous trouvez ...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ Vous refusez, vous voulez vous détendre, pas être traqué.
- ○ Vous n'acceptez jamais de donner des informations sur votre localisation.
- ○ Vous acceptez car vous avez confiance en cette application.
- ○ Vous acceptez car seulement vos amis auront accès à l'information.

14 [G2_Q0013]

Une application vous propose de lui permettre de pouvoir activer le Wifi pour accélérer ses traitements lorsque vous êtes à portée d'un réseau connu...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ Non je suis celui qui choisit quand le wifi est actif ou pas.
- ○ Si je dois avoir le résultat que j'attends de l'application plus rapidement, j'accepte.
- ○ Je ne suis pas contre, mais je préfèrerai qu'il me demande.

15 [G2_Q0014]

Vous avez téléchargé une application permettant de partager vos playlists de musique et ainsi voir les playlists d'autres utilisateurs. Une personne que vous ne connaissez pas souhaite accéder à votre playlist...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ La question ne se pose pas, je n'aurai jamais téléchargé une application ayant accès à ma musique.
- ○ Je refuse, je ne souhaite partager ma musique qu'avec des gens que je connais.
- ○ J'accepte, ça me permettra de connaître ce qu'il écoute.

16 [G2_Q0015]

Une application vous annonce que pour des raisons de sécurité, elle nécessite l'accès à votre appareil photo pour faire de la reconnaissance faciale...

*

Veuillez sélectionner une seule des propositions suivantes :

- ○ Je refuse, je ne veux pas qu'il puisse accéder à mon appareil photo à n'importe quel moment.
- ○ Si elle me dit que c'est pour des raisons de sécurité, j'accepte.
- ○ J'accepte dans tous les cas pour avoir accès à l'application.

Slider

17 [SSMS]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir lire vos sms/mms ou en écrire. *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse lire vos sms/mms ou en écrire et que la note 20 correspond à une volonté de lui laisser cette possibilité.

18 [SPHOTO]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser vos photos/vidéos *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser vos photos/vidéos et que la note 20 correspond à une volonté de lui laisser cette possibilité.

19 [SMAIL]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser vos mails *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser vos mails et que la note 20 correspond à une volonté de lui laisser cette possibilité.

20 [SCAL]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser vos calendriers *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser vos calendriers et que la note 20 correspond à une volonté de lui laisser cette possibilité.

21 [SLCON]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser votre liste de contacts *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser votre liste de contacts et que la note 20 correspond à une volonté de lui laisser cette possibilité.

22 [SENAV]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser vos éléments de navigation web (favoris, historique, cookies). *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser vos éléments de navigation web et que la note 20 correspond à une volonté de lui laisser cette possibilité.

23 [SNTEL]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser votre numéro de téléphone *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser votre numéro de téléphone et que la note 20 correspond à une volonté de lui laisser cette possibilité.

24 [SAMAIL]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser votre adresse mail *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser votre adresse mail et que la note 20 correspond à une volonté de lui laisser cette possibilité.

25 [SADR]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser votre adresse physique *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser votre adresse physique et que la note 20 correspond à une volonté de lui laisser cette possibilité.

26 [SNOM]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser votre nom et/ou prénom *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser votre nom et/ou prénom et que la note 20 correspond à une volonté de lui laisser cette possibilité.

27 [SPSEUD]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser votre surnom *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser votre surnom et que la note 20 correspond à une volonté de lui laisser cette possibilité.

28 [SGPS]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser le GPS de votre téléphone *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser le GPS de votre téléphone et que la note 20 correspond à une volonté de lui laisser cette possibilité.

29 [SBTW]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser le Bluethooth ou le Wifi de votre téléphone *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser le Bluethooth ou le Wifi de votre téléphone et que la note 20 correspond à une volonté de lui laisser cette possibilité.

30 [SSIC]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser le lecteur de musique ou de vidéo de votre téléphone *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser le lecteur de musique ou de vidéo de votre téléphone et que la note 20 correspond à une volonté de lui laisser cette possibilité.

31 [SAPN]Sur une échelle allant de 0 à 20, quel est votre comportement face à une application qui veut pouvoir utiliser l'appareil photo de votre téléphone *

Veuillez vérifier le format de votre réponse.

Veuillez écrire votre réponse ici :

On considèrera que la note 0 correspond à une volonté de refuser que l'application puisse utiliser l'appareil photo de votre téléphone et que la note 20 correspond à une volonté de lui laisser cette possibilité.

Comparaison

32 [1]Durant le questionnaire, vous avez dit vouloir éviter autant que possible de partager les données concernant votre identité (nom/prénom, adresse, n° tel, ...). Pourtant pendant la deuxième partie du questionnaire, vos réponses contredisent cette réponse et montre que vous êtes plutôt d'accord pour partager certaines de ces données. Quel réponse est la plus exacte ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0001.NAOK** == "A1") and (**sum**(**SNTEL**, **SAMAIL**, **SADR**, **SNOM**, **SPSEUD**) >= "45"))

Veuillez sélectionner une seule des propositions suivantes :

- ◯ Je confirme ce que j'ai dis dans le questionnaire, je ne veux pas partager les données concernant mon identité
- ◯ Ca va dépendre de la situation, certaines fois oui, d'autres fois non.
- ◯ Les notes m'ont permis de mieux définir mes préférences, en fait je veux bien partager certaines de mes informations d'identité.

33 [2]Durant le questionnaire, vous avez dit être d'accord pour partager toutes les données concernant vos données d'identité (nom/prénom, adresse, n° tél, ...). Pourtant pendant la deuxième partie du questionnaire, vous avez plutôt indiqué vouloir éviter de trop les partager. Que voulez vous vraiment faire ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0001.NAOK** == "A3") and (**sum**(**SNTEL**, **SAMAIL**, **SADR**, **SNOM**, **SPSEUD**) <= "55"))

Veuillez sélectionner une seule des propositions suivantes :

- ◯ Je maintiens ma réponse du questionnaire, je suis d'accord pour partager les données concernant mon identité.
- ◯ Ca va dépendre de la situation, certaines fois oui, d'autres fois non.
- ◯ Les notes m'ont permis de mieux définir mes préférences, en fait je ne veux pas partager certaines de mes informations d'identité.

34 [3]Durant le questionnaire, vous avez dit ne remplir que les informations strictement nécessaires quand ça concerne vos données d'identité (nom/prénom, adresse, n°tel, ...). Pourtant, dans la deuxème partie du sondage, vous avez indiqué être plutôt en faveur du partage de ce type de données. Quel réponse est la plus exacte ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0001.NAOK** == "A2") and (**sum**(**SNTEL**, **SAMAIL**, **SADR**, **SNOM**, **SPSEUD**) >= "65"))

Veuillez sélectionner une seule des propositions suivantes :

- ○ Ca va dépendre de la situation mais je préfère en partager le moins possible.
- ○ Finalement, je suis d'accord pour partager la majeure partie de mes données d'identité.

35 [4]Durant le questionnaire, vous avez dit ne remplir que les informations strictement nécessaires quand ça concerne vos données d'identité (nom/prénom, adresse, n°tel, ...). Pourtant, dans la deuxème partie du sondage, vous avez indiqué plutôt vouloir garder pour vous ce type de données. Quel réponse est la plus exacte ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0001.NAOK** == "A2") and (**sum**(**SNTEL**, **SAMAIL**, **SADR**, **SNOM**, **SPSEUD**) <= "35"))

Veuillez sélectionner une seule des propositions suivantes :

- ○ Ca va dépendre de la situation, je veux bien partager quelques informations mais le moins possible.
- ○ Finalement, je ne veux pas partager les informations concernant mon identité.

36 [5]Lors du questionnaire, vous avez dit que vos données d'ordre privées (photos, mails, liste de contacts, ...) étaient strictement confidentielles. Pourtant dans la 2ème partie du sondage, vous avez indiqué être plutôt enclin à les partager. Qu'en est il vraiment ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0007.NAOK** == "A1") and (**sum**(**SSMS.NAOK**, **SPHOTO.NAOK**, **SMAIL**, **SCAL**, **SLCON**, **SENAV**) >= "55"))

Veuillez sélectionner une seule des propositions suivantes :

- ○ Je m'en tiens à ce que j'ai dis dans le questionnaire, c'est confidentiel.
- ○ Ca va dépendre de la situation, certaines fois oui, d'autres fois non.
- ○ Finalement, je veux bien partager mes données privées.

37 [6]Lors du questionnaire, vous avez dit être d'accord pour partager vos données privées (photos, mails, listes de contact, ...). Pourtant pendant la 2ème partie du sondage, vos réponses indiquent le contraire. Qu'en est il vraiment ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0007.NAOK** == "A3") and (**sum**(**SSMS.NAOK**, **SPHOTO.NAOK**, **SMAIL**, **SCAL**, **SLCON**, **SENAV**) <= "65"))

Veuillez sélectionner une seule des propositions suivantes :

- ◯ Je m'en tiens à ce que j'ai dis dans le questionnaire, je suis d'accord pour partager ce genre de données.
- ◯ Ca va dépendre de la situation, certaines fois oui, d'autres fois non.
- ◯ Non finalement je ne veux pas partager de type de données.

38 [7]Lors du questionnaire, vous avez indiqué être d'accord pour diffuser certaines de vos données privées (photos, mails, liste de contacts) mais pendant la 2ème partie du sondage, vous vous être montré vraiment en faveur du partage de ces données. Qu'en est il vraiment ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0007.NAOK** == "A2") and (**sum**(**SSMS.NAOK**, **SPHOTO.NAOK**, **SMAIL**, **SCAL**, **SLCON**, **SENAV**) >= "75"))

Veuillez sélectionner une seule des propositions suivantes :

- ◯ Je reste sur ma réponse du questionnaire, je suis d'accord pour partager mais selon la situation.
- ◯ Finalement, je suis d'accord pour partager la majeure partie de mes données privées.

39 [8]Lors du questionnaire, vous avez indiqué être d'accord pour diffuser certaines de vos données privées (photos, mails, liste de contacts) selon la situation mais les réponses de la 2ème partie du sondage montre que vous êtes plutôt contre ce partage. Qu'en est il vraiment ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0007.NAOK** == "A3") and (**sum**(**SSMS.NAOK**, **SPHOTO.NAOK**, **SMAIL**, **SCAL**, **SLCON**, **SENAV**) <= "45"))

Veuillez sélectionner une seule des propositions suivantes :

- ◯ Je maintiens ma réponse du questionnaire, ça va dépendre de la situation.
- ◯ Finalement, je ne veux pas partager mes données privées.

40 [9]Lors du questionnaire, vous avez renseigné ne pas vouloir autoriser l'utilisation des services de votre téléphone (GPS, Wifi, appareil photo, ...) par des applications. Pourtant dans la 2ème partie du sondage, vos réponses indiquent le contraire. Qu'en est il vraiment ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0011** == "A1") and (**sum**(**SGPS**, **SBTW**, **SSIC**, **SAPN**) >= "35"))

Veuillez sélectionner une seule des propositions suivantes :

- ○ Je maintiens ma réponse du questionnaire, je ne veux pas qu'une application puisse utiliser les services de mon téléphone.
- ○ Ca va dépendre de la situation, certaines fois oui, d'autres fois non.
- ○ Finalement, je veux bien laisser l'utilisation des services de mon téléphone par des applications.

41 [10]Lors du questionnaire, vous avez renseigné être d'accord pour laisser les applications utiliser les services de votre téléphone (GPS, Wifi, appareil photo, ...). Pourtant lors de la 2ème partie du sondage, vous avez indiqué dans vos réponses plutôt le contraire. Qu'en est il vraiment ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0011** == "A3") and (**sum**(**SGPS**, **SBTW**, **SSIC**, **SAPN**) <= "45"))

Veuillez sélectionner une seule des propositions suivantes :

- ○ Je maintiens ma réponse du questionnaire, je suis d'accord pour autoriser des applications à utiliser les services disponibles sur mon téléphone.
- ○ Non finalement je préfère ne pas autoriser les applications à prendre le contrôle des services de mon téléphone.

42 [11]Lors du questionnaire, vous avez renseigné être d'accord pour autoriser des applications à utiliser les services de votre téléphone (GPS, Wifi, apareil photo, ...) mais selon le contexte. Dans la 2ème partie du sondage, vous avez indiqué être plutôt contre. Qu'en est il vraiment ?

Répondre à cette question seulement si les conditions suivantes sont réunies :
° ((**G2_Q0011** == "A2") and (**sum**(**SGPS**, **SBTW**, **SSIC**, **SAPN**) <= "28"))

Veuillez sélectionner une seule des propositions suivantes :

- ○ Je confirme ce que j'ai dis dans le questionnaire, c'est au cas par cas selon la situation.
- ○ Non finalement je préfère ne pas autoriser les applications à prendre le contrôle des services de mon téléphone.

Bibliographie

[1] ABDOLMOHAMMADI, M. et USOFF, C. (2001). A longitudinal study of applicable decision aids for detailed tasks in a financial audit. *Intelligent Systems in Accounting, Finance Management*, 10(3):139–154.

[2] ADOMAVICIUS, G. et TUZHILIN, A. (2005). Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749.

[3] AGRAWAL, R., KIERNAN, J., SRIKANT, R. et XU, Y. (2002). Hippocratic databases. *In Proceedings of the 28th international conference on Very Large Data Bases*, pages 143–154. VLDB Endowment.

[4] AJAM, N., CUPPENS-BOULAHIA, N. et CUPPENS, F. (2010). Contextual privacy management in extended role based access control model. *In Data Privacy Management and Autonomous Spontaneous Security*, pages 121–135. Springer.

[5] ASHLEY, P., HADA, S., KARJOTH, G., POWERS, C. et SCHUNTER, M. (2003). Enterprise privacy authorization language (epal 1.2). *Submission to W3C*.

[6] ASSEMBLÉE GÉNÉRALE DES NATIONS UNIES (1948). Déclaration universelle des droits de l'homme. *Résolution 217A (III)*, 10.

[7] BARKER, S. (2009). The next 700 access control models or a unifying meta-model ? *In Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 187–196. ACM.

[8] BARTO, A. G. (1998). *Reinforcement learning : An introduction*. MIT press.

[9] BELL, D. E. et LAPADULA, L. J. (1973). Secure computer systems : Mathematical foundations. Rapport technique, DTIC Document.

[10] BIBA, K. J. (1977). Integrity considerations for secure computer systems. Rapport technique, DTIC Document.

[11] BOUTET, A. (2013). *Décentralisation des systèmes de personnalisation*. Thèse de doctorat, Université Rennes 1.

[12] BOUYSSOU, D., DUBOIS, D., PIRLOT, M. et PRADE, H. (2006). Concepts et méthodes pour l'aide à la décision. 3, analyse multicritère.

[13] BYUN, J.-W., BERTINO, E. et LI, N. (2005). Purpose based access control of complex data for privacy protection. *In Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 102–110. ACM.

[14] CACIOPPO, J. T., GARDNER, W. L. et BERNTSON, G. G. (1997). Beyond bipolar conceptualizations and measures : The case of attitudes and evaluative space. *Personality and Social Psychology Review*, 1(1):3–25.

[15] CASTELLUCCIA, C., DRUSCHEL, P., HÜBNER, S. F., PASIC, A., PRENEEL, B. et TSCHOFENIG, H. (2011). Privacy, accountability and trust-challenges and opportunities. *ENISA.[Online]. Available : http ://www. enisa. europa. eu/activities/identity-and-trust/library/deliverables/pat-study/at download/fullReport*.

[16] CAVOUKIAN, A. (1995). Privacy-enhancing technologies : The path to anonymity. *Privacy Commissioner of Ontario, Canada*.

[17] CAVOUKIAN, A., STODDART, J., DIX, A., NEMEC, I., PEEP, V. et SHROFF, M. (2010). Privacy by design resolution. *In Issued at 32nd Intl. Conf. of Data Protection Privacy Commissioners*.

[18] CHAUM, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90.

[19] CHEAITO, M. (2012). *Un cadre de spécification et de déploiement de politiques d'autorisation*. Thèse de doctorat, Université de Toulouse, Université Toulouse III-Paul Sabatier.

[20] CHOQUET, G. (1953). Theory of capacities. *In Annales de l'institut Fourier*, volume 5, page 54.

[21] CONSEIL DE L'EUROPE (1948). Convention européenne des droits de l'homme.

[22] CORNUÉJOLS, A. et MICLET, L. (2011). *Apprentissage artificiel : concepts et algorithmes*. Editions Eyrolles.

[23] CRANOR, L., LANGHEINRICH, M., MARCHIORI, M. et REAGLE, J. (2002). The platform for privacy preferences 1.0 (p3p1.0) specification. W3C Recommendation.

[24] CUPPENS, F. et MIÈGE, A. (2003). Modelling contexts in the or-bac model. *In Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pages 416–425. IEEE.

[25] CUPPENS, F. et MIÈGE, A. (2004). Or-bac. *Organization Based Access Control, Journées Druide, Le Croisic*.

[26] DANEZIS, G. et GÜRSES, S. (2010). A critical review of 10 years of privacy technology. *Proceedings of Surveillance Cultures : A Global Surveillance Society*.

[27] De Capitani di VIMERCATI, S. et SAMARATI, P. (2011). Primelife project : Next generation policies.

[28] de CARITAT, M. J. A. N. *et al.* (1785). *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. L'imprimerie royale.

[29] DINGLEDINE, R., MATHEWSON, N. et SYVERSON, P. (2004). Tor : The second-generation onion router. Rapport technique, DTIC Document.

[30] DONG, C. (1992). *Développement d'outils d'aide à la décision dans des environnements multi-critères, dynamiques et incertains : Application à des problèmes de planification régionale*. Thèse Doct.Ing. : Université Catholique de Louvain, Louvain-la-Neuve, Belgique.

[31] GLOBAL, L. (2012). The value of our digital identity. boston consulting group.

[32] GORRY, G. A. et MORTON, M. S. S. (1971). *A framework for management information systems*, volume 13. Massachusetts Institute of Technology.

[Grabisch et I] GRABISCH, M. et I, U. D. P. Using the kappalab r package for capacity identification in choquet integral based maut.

[34] GRABISCH, M. et ROUBENS, M. (2000). Application of the choquet integral in multicriteria decision making. *Fuzzy measures and integrals*, (40):348–375.

[35] HEDBOM, H. (2009). A survey on transparency tools for enhancing privacy. *In The future of identity in the information society*, pages 67–82. Springer.

[36] INGLESANT, P., SASSE, M. A., CHADWICK, D. et SHI, L. L. (2008). Expressions of expertness : the virtuous circle of natural language for access control policy specification. *In Proceedings of the 4th symposium on Usable privacy and security*, pages 77–88. ACM.

[37] ITU (2005). ITU Internet Reports 2005, "The internet of Things", 7th edition.

[38] JIANG, H. et EASTMAN, J. R. (2000). Application of fuzzy measures in multi-criteria evaluation in gis. *International Journal of Geographical Information Science*, 14(2):173–184.

[39] JORDAN, C. S. (1987). *Guide to Understanding Discretionary Access Control in Trusted Systems*. DIANE Publishing.

[40] KALAM, A. A. E., BAIDA, R., BALBIANI, P., BENFERHAT, S., CUPPENS, F., DESWARTE, Y., MIEGE, A., SAUREL, C. et TROUESSIN, G. (2003). Organization based access control. *In Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*, pages 120–131. IEEE.

[41] KEEN, P. G. et MORTON, M. S. S. (1978). *Decision support systems : an organizational perspective*, volume 197. Addison-Wesley Reading, MA.

[42] KEENEY, R. L. et RAIFFA, H. (1976). Decision with multiple objectives.

[43] KOJADINOVIC, I. (2007). Minimum variance capacity identification. *European Journal of Operational Research*, 177(1):498–514.

[44] LAMPSON, B. (1971). Protection. *In Proc. 5th Princeton Conf. on Information Sciences and Systems*, pages 18–24. Princeton.

[45] LANGHEINRICH, M. (2009). Privacy in ubiquitous computing. pages 96–156.

[46] LEDERER, S., HONG, J. I., DEY, A. K. et LANDAY, J. A. (2004). Personal privacy through understanding and action : five pitfalls for designers. *Personal and Ubiquitous Computing*, 8(6):440–454.

[47] LINDEN, G., SMITH, B. et YORK, J. (2003). Amazon. com recommendations : Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80.

[48] MACHANAVAJJHALA, A., KIFER, D., GEHRKE, J. et VENKITASUBRAMANIAM, M. (2007). l-diversity : Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3.

[49] MARTIN, A., ZARATÉ, P. et CAMILLERI, G. (2012). Gestion et évolution de profils multicritères de décideur par apprentissage pour l'aide à la décision. *In INFORSID*, pages 223–238.

[50] MARX, G. T. (2001). Murky conceptual waters : The public and the private. *Ethics and Information technology*, 3(3):157–169.

[51] MATOS, A. M. M. (2012). Privacy in next generation networks.

[52] MOORE JR, B. (1984). Privacy : Studies in social and cultural history, armonk, ny : Me sharpe.

[53] MORGAN, R. L., CANTOR, S., CARMODY, S., HOEHN, W. et KLINGENSTEIN, K. (2004). Federated security : The shibboleth approach. *Educause Quarterly*, 27(4):12–17.

[54] MUROFUSHI, T. (1992). A technique for reading fuzzy measures (i) : the shapley value with respect to a fuzzy measure. *In 2nd Fuzzy Workshop*, pages 39–48.

[55] MUROFUSHI, T. et SONEDA, S. (1993). Techniques for reading fuzzy measures (iii) : interaction index. *In 9th Fuzzy System Symposium*, pages 693–696. Sapporo,, Japan.

[56] NAEHRIG, M., LAUTER, K. et VAIKUNTANATHAN, V. (2011). Can homomorphic encryption be practical ? *In Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 113–124. ACM.

[57] NGUYEN, D. H. et MYNATT, E. D. (2002). Privacy mirrors : understanding and shaping socio-technical ubiquitous computing systems.

[58] NI, Q., BERTINO, E., LOBO, J., BRODIE, C., KARAT, C.-M., KARAT, J. et TROMBETA, A. (2010). Privacy-aware role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 13(3):24.

[59] PEARSON, S. et MONT, M. C. (2011). Sticky policies : An approach for managing privacy across multiple parties. *Computer*, 44(9):60–68.

[60] PFITZMANN, A. et KÖHNTOPP, M. (2001). Anonymity, unobservability, and pseudonymity—a proposal for terminology. *In Designing privacy enhancing technologies*, pages 1–9. Springer.

[61] ROY, B. (1968). Classement et choix en présence de points de vue multiples. *RAIRO-Operations Research-Recherche Opérationnelle*, 2(V1):57–75.

[62] ROY, B. (1978). Electre iii : Un algorithme de classements fondé sur une représentation floue des préférences en présence de criteres multiples. *Cahiers du CERO*, 20(1):3–24.

[63] ROY, B. (1985). Méthodologie multicritère d" aide à la décision.

[64] SAATY, T. L. (1988). *What is the analytic hierarchy process ?* Springer.

[65] SANDHU, R., FERRAIOLO, D. et KUHN, R. (2000). The nist model for role-based access control : towards a unified standard. *In ACM workshop on Role-based access control*, volume 2000.

[66] SHAPLEY, L. S. (1952). A value for n-person games. Rapport technique, DTIC Document.

[67] SIMON, H. A. (1976). *Administrative behavior*, volume 3. Cambridge Univ Press.

[68] SIMON, H. A. (1977). The new science of management decision.

[69] SIMON, H. A. (2001). Making management decisions : the role of intuition and emotion.

[70] SLOVIC, P., FINUCANE, M. L., PETERS, E. et MACGREGOR, D. G. (2007). The affect heuristic. *European Journal of Operational Research*, 177(3):1333–1352.

[71] SOLOVE, D. J. (2006). A taxonomy of privacy. *University of Pennsylvania Law Review*, pages 477–564.

[72] STEPIEN, B., FELTY, A. et MATWIN, S. (2014). A non-technical xacml target editor for dynamic access control systems.

[73] STEPIEN, B., MATWIN, S. et FELTY, A. (2011). Advantages of a non-technical xacml notation in role-based models. *In Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*, pages 193–200. IEEE.

[74] SUGENO, M. (1974). Theory of fuzzy integrals and its applications.

[75] SWEENEY, L. (2002). k-anonymity : A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570.

[76] WAGEALLA, W., TERZIS, S. et ENGLISH, C. (2003). Trust-based model for privacy control in context aware systems. *In Second Workshop on Security in Ubiquitous Computing at the Fifth Annual Conference on Ubiquitous Computing (UbiComp2003)*.

[77] WANG, Y. et KOBSA, A. (2006). Privacy-enhancing technologies. *Social and Organizational Liabilities in Information Security*, pages 203–227.

[78] WARREN, S. D. et BRANDEIS, L. D. (1890). The right to privacy. *Harvard law review*, pages 193–220.

[79] WÄSTLUND, E., FISCHER HÜBNER, S., GRAF, C., HOCHLEITNER, C., WOLKERSTORFER, P. et ANGULO, J. (2011). Towards usable privacy enhancing technologies : Lessons learned from the primelife project.

[80] WESTIN, A. F. (1968). Privacy and freedom. *Washington and Lee Law Review*, 25(1):166.

[81] YAGER, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *Systems, Man and Cybernetics, IEEE Transactions on*, 18(1):183–190.

[82] ZELENY, M. et COCHRANE, J. L. (1982). *Multiple criteria decision making*, volume 25. McGraw-Hill New York.

[83] ZIELKE, A., SITTER, H., RAMPP, T., BOHRER, T. et ROTHMUND, M. (2001). Clinical decision-making, ultrasonography, and scores for evaluation of suspected acute appendicitis. *World Journal of Surgery*, 25(5):578–584.

Liste des figures