Une démarche de conception et d'implémentation de la protection de la vie privée basée sur le contrôle d'accès appliquée aux compositions de services

Aurélien Faravelon

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministérial : 7 août 2006

Présentée par

Aurélien Faravelon

Thèse dirigée par **Christine Verdier**
et codirigée par **Philippe Saltel**

préparée au sein **Laboratoire d'Informatique de Grenoble (LIG) - Groupe de Recherche Philosophie, Langage & Cognition (PLC)**
et de **Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Une démarche de conception et d'implémentation de la protection de la vie privée basée sur le contrôle d'accès et appliquée aux compositions de services

Thèse soutenue publiquement le **2 décembre 2013**,
devant le jury composé de :

Laurence DUCHIEN
Professeur, University Lille 1, Rapporteur

Stéphane FRENOT
Professeur, Institut National des Sciences Appliquées de Lyon, Rapporteur

Camille ROSENTHAL-SABROUX
Professeur, Université Paris Dauphine, Examinateur, **Présidente**

Claude GAUTIER
Professeur, Ecole Normale Supérieure Lettres et Sciences Humaines de Lyon, Examinateur

Christine VERDIER
Professeur, Université Grenoble 1, Directeur de thèse

Philippe SALTEL
Professeur, Université Grenoble 2, Co-Directeur de thèse

Remerciements

« Ils ne savaient pas que c'était impossible, alors ils l'ont fait » écrivait Mark Twain. Je pourrais, aujourd'hui, en dire de même d'un travail interdisciplinaire qui n'aurait pu être mené à bien sans le soutien et l'implication de mon entourage scientifique et personnel.

Je tiens à remercier le Professeur Laurence Duchien et le Professeur Stéphane Frénot d'avoir accepté d'être rapporteurs de ce mémoire. Vos relectures et vos commentaires m'ont été précieux dans la dernière phase de mon travail. Je tiens aussi à remercier le Professeur Camille Rosenthal-Sabroux et le Professeur Claude Gauthier pour leur participation à mon jury de thèse.

Philippe Saltel, qui suivait déjà mon travail de Master, a accepté d'encadrer la partie philosophie de ce travail. Je tiens à le remercier de l'attention qu'il a porté à mes recherches.

Il y a 4 ans, je rencontrais Christine Verdier. Merci d'avoir accepté de relever le défi que constitue toute direction de thèse et particulièrement d'une thèse qui se voulait à la croisée de deux disciplines. Merci de tes qualités humaines. Ta présence, tes relectures patientes et nombreuses et tes conseils m'ont été d'une grande aide tout au long de mon travail. Merci enfin de m'avoir littéralement accompagné au bout du monde.

Mes frères et, ma mère et mon beau-père ont activement participé à la réussite de mon travail par le soutien indéfectible. Merci de tous ses moments passés ensemble et de votre présence. Merci aussi à mes grands-parents pour les vacances, les repas, les week-end et tous les moments que nous passons ensemble.

Je ne serai rien sans Paul, Eric, Camille, Aurélie, Gaëlle, Aurore, Charlotte, Isabelle, Benjamin, Juan-Pablo, Mario et tous les autres. Eric, qui aurait pu dire lorsque nous nous sommes rencontrés que nous nous retrouverions en thèse ensemble ? Merci de cette amitié qui représente beaucoup pour moi, pour ta disponibilité et toutes les discussions que nous avons pu avoir. Paul, merci pour les cafés et les sorties. Tu m'as aidé à trouver un équilibre au milieu du travail. Camille, mon petit chaton, merci pour ta sagacité et ta passion pour le féminisme. Merci pour nos soirées, merci de m'avoir accepté végétarien, puis carnivore. Aurélie, merci d'être toujours aussi impressionnante, d'être la personne avec qui enseigner et facile, d'être celle avec qui parler est simple. Merci d'être celle avec qui Stockholm est ma destination favorite pour l'hiver. Gaëlle, merci d'avoir été ma partenaire. Merci pour nos discussions, ton soutien, les pieds dans les mains à l'étage du O'Brother et tous les moments passés ensemble. Aurore, merci de m'avoir nourri pendant la dernière période de ma thèse. Merci d'avoir fait en sorte que la collocation soit naturelle et d'être aussi présente. Merci aussi à Clément et Jima qui m'ont supporté avant toi. Merci aussi à Isabelle et Charlotte d'avoir enseigné avec moi et d'être devenues des amies. Merci à Benjamin d'être mon ancrage parisien. Merci de tes relectures éclairantes. Je suis impressionné par tout le chemin que tu as accompli et toutes les réussites que tu as connues depuis notre rencontre. Merci enfin à Juan-Pablo et Mario d'avoir partagé mon bureau. Juan-Pablo, merci d'être devenu un ami, de ta spontanéité et de tous

tes conseils. Mario, merci pour la Guadeloupe, j'espère que nous aurons rapidement l'occasion de publier dans la même conférence de nouveau.

Je tiens à remercier l'équipe ADELE du Laboratoire d'Informatique de Grenoble. Merci à Philippe Lalanda pour ses relecture et ses conseils qui ont notamment contribué à la qualité de mon papier à ICSOC 2012. Merci à Stéphanie Chollet pour notre collaboration qui m'a permis de repartir de ton approche pour l'appliquer au problème de la vie privée. Merci de m'avoir aidé à structurer la partie de mon manuscrit consacré à l'informatique et de m'avoir permis d'utiliser SECURE FOCAS pour mes premiers prototypages.

Je tiens à remercier l'équipe pédagogique du département informatique de l'IUT 2 de Grenoble qui m'accueille depuis trois ans. Merci de votre confiance renouvelée, j'ai beaucoup appris à votre contact. Merci aussi aux étudiants à qui j'ai pu enseigner ou encadrer dans le cadre de stages. Je remercie en particulier Zine Ukili et Loïc Letreneuf qui ont activement participé à la réalisation du démonstrateur de mon travail.

Enfin, je remercie l'équipe SIGMA du Laboratoire d'Informatique de Grenoble qui m'a accueilli et soutenu pendant ma thèse. Je remercie tout particulièrement Agnès Front pour ses relectures nombreuses et les corrections qu'elle a apportées aux modèles sur lesquels j'ai travaillé.

Publications

Les travaux discutés ici ont été présentés dans les conférences internationales suivantes :

- Aurélien Faravelon, Stéphanie Chollet, Christine Verdier, Agnès Front : Enforcing privacy as access control in a pervasive context. *CCNC 2012* : 380-384.
- Aurélien Faravelon, Contexts and Distributive Justice : Privacy in a Networked Society. *Amsterdam Privacy Conference 2012* : à paraître.
- Aurélien Faravelon, Stéphanie Chollet, Christine Verdier, Agnès Front : Configuring Private Data Management as Access Restrictions : From Design to Enforcement. *ICSOC 2012* : 344-358.
- Aurélien Faravelon, Christine Verdier, Agnès Front : Towards a business-centric definition of access control policies. *RCIS 2011* : 1-11.
- Aurélien Faravelon, Christine Verdier : Towards a Framework for Privacy Preserving Medical Data Mining Based on Standard Medical Classifications. *eHealth 2010*. 204-211
- Aurélien Faravelon. « Penser avec Foucault : le cas de la vie privée ». *Enjeux politiques du document numérique, Actes de la conférence Document numérique et société 2010* : 339-362.

Résumé

La vie privée et sa protection sont aujourd'hui largement discutées. Membres de la société civile, juristes ou encore techniciens, nous sommes tous appelés à nous emparer d'une notion que l'on nous pre ?sente à la fois comme menacée, désuète ou appartenant à nos libertés fondamentales. Aujourd'hui, les controverses autour de la protection de la vie privée ont pour origine des usages techniques. L'informatisation des fichiers étatiques et les possibilités accrues de surveillance issues des innovations en informatique et, plus récemment, les « usages sociaux » des outils numériques comme les « réseaux sociaux », provoquent de vives réactions. Pourtant, le recours à cette notion, notamment pour protéger les libertés individuelles, est-il complètement satisfaisant alors que, d'une part, les outils à l'origine de sa mise en question suscitent un large engouement, et que, d'autre part, ses contours sont mal définis ? Nous adoptons, pour répondre à cette question, une position interdisciplinaire.

D'une part, nous enquêtons d'un point de vue philosophique sur la « condition numérique » contemporaine afin d'en saisir les enjeux. Ce faisant, nous établissons que les outils numériques remettent en cause la notion de « frontière ». Nous montrons simultanément que la possibilité d'une existence séparée est nécessaire pour constituer une subjectivité propre. Se pose alors la question de la mise en pratique d'une telle existence. Nous nous éloignons des approches déontologiques et utilitaristes qui guident actuellement la conception et l'évaluation des outils numériques pour leur préférer une approche fondée sur « l'éthique du souci de soi ». Cette approche nous conduit à établir que le code informatique constitue la structure de la condition numérique et qu'il s'agit de prendre en compte, dès la conception d'une application un ensemble de propriétés, comme la protection de la vie privée.

Nous cherchons dans un second temps à aider les concepteurs d'applications à concevoir au mieux et à réaliser des applications qui permettent de protéger la vie privée des utilisateurs et des possesseurs des données. Notre domaine d'application est l'approche orientée services qui est aujourd'hui un largement utilisée. Nous nous concentrons sur son utilisation pour la réalisation d'applications à partir de compositions de services dynamiques et hétérogènes. Nous cherchons à protéger la vie privée à l'aide du contrôle d'accès. Pour ce faire, nous proposons de configurer les propriétés de contrôle d'accès des services au moyen d'une démarche dirigée par les modèles divisée en deux étapes. Au niveau conception, la composition et la politique de contrôle d'accès à un niveau abstrait sont spécifiées par des experts dédiés. Nous estimons que le contrôle d'accès doit être pris en compte dès la conception de l'application afin d'éviter le recours à la programmation manuelle. En rester à un niveau abstrait permet de s'adapter à l'état de la composition et à l'hétérogénéité et au dynamisme des services. Au niveau exécution, notre architecture permet de configurer les services concrets au moyen de *proxies* responsables de l'exécution du contrôle d'accès. Des transformations de modèles vers textes automatisées permettent de passer d'un niveau à l'autre afin de s'abstraire de la programmation manuelle et de garantir la protection des services concrets par les *proxies*. Notre approche a été validée par la réalisation d'un prototype et son utilisation sur un cas d'application.

Abstract

Privacy is hot topic. Lawyers, technicians and plain people are all concerned by this notion. Nowadays, most discussions focus on the effects of digital tools, such as social media or surveillance software. However, privacy is still ill-defined. Moreover, digital tools which endanger privacy are widely used. Should not we leave privacy aside and accept that we are, maybe more than ever, visible ?

In this doctoral thesis, I address this question from a twofold viewpoint. I first inquire into the nature of our digital condition from a philosophical standpoint. I claim that digital artifacts rework the implementation of our frontiers, be them geographical or social. However, I contend that such frontiers are necessary. As I show that code defines the structure and the effects of digital tools, I point out that properties such as privacy management should be addressed right from the conception of software applications.

Helping out designers to address such properties is the second issue I tackle. I focus on Service-Oriented Computing as it is a widely used paradigm. Most specifically, I deal with the composition of heterogenous and dynamic services. I define access control as an efficient mechanism to protect privacy and I propose a twofold generative approach to secure services compositions. The composition and its access control policies are separately defined at an abstract level. An expert is responsible for each of them. As we promote an abstract description of the application, we free the designer from technical complexity. At runtime, we propose an architecture which selects and protects the actual services by hiding them behind proxies which run the access control policy. Automated model transformations permit to generate the application from its specification. We thus bypass manual programming. We have implemented a modeling and execution environment and applied our approach to a use case in order to validate our work.

Sommaire

I Introduction générale **1**

II Concevoir la condition numérique, une enquête philosophique **7**

1 Introduction **9**
I L'informatique, une technique aux usages controversés 9
II La surveillance, angle d'approche privilégié de l'informatique et de ses usages 10
III Qu'est-ce que vivre à l'ère du numérique ? 11

2 Informatique et agent technique **13**
I Définir l'informatique : une technique abstraite 15
II Spécificités de l'informatique comme technique 18
III La technique comme agent 21
A Toute technique est-elle systématique ? 21
B Formuler des fins propres dans un système technique 22
IV Conclusion 24

3 Individu, individuation et technique **25**
I Agent technique et individuation 27
II Se soucier de soi pour se constituer comme sujet 39
III Le souci de soi et l'éthique de la vertu 50
IV Conclusion 59

4 Géographie du virtuel **61**
I Géographie du virtuel : le mode de présence spécifique des techniques numériques 62
A L'espace numérique, un espace modulé 62
B L'espace numérique, un espace surveillé et programmé 67
C Les multiples facettes de l'identité numérique 72
II Exister séparément dans le monde numérique 75
A Le conflit des deux aspects de la subjectivité 75
B Les formes de séparation 77
III Les pratiques de liberté possibles 81
A La contractualisation des formes de séparation 81
B Inscrire la séparation dans la structure du monde numérique . 82
IV Conclusion 85

5 Conclusion **87**

III Une démarche de conception et d'implémentation de la vie privée basée sur le contrôle d'accès et appliquée aux compositions de services **89**

1 Introduction **91**
I Problématique . . . 91
II Objectifs . . . 92
III Structure du document . . . 92

2 État de l'art **95**
I L'approche orientée services et ses implémentations . . . 96
A Concepts et définitions . . . 97
1 Approche orientée service – Service-Oriented Computing (SOC) . . . 98
2 Architecture orientée service – Service-Oriented Architecture (SOA) . . . 100
B Compositions de services . . . 103
1 Compositions par procédé : orchestration et chorégraphie de services . . . 104
2 Composition structurelle . . . 107
3 Un exemple d'autre type de composition : la composition sémantique . . . 107
4 Synthèse sur les compositions . . . 108
C Implémentations de l'approche orientée service . . . 110
1 Caractérisation des technologies à service . . . 110
2 Services Web . . . 110
3 Universal Plug and Play (UPnP) et Device Profile for Web Service (DPWS) . . . 116
4 Les modèles à composants . . . 121
5 Service Component Architecture (SCA) . . . 124
D Synthèse . . . 126
II Sécurité : concepts et buts . . . 127
A Attaques et menaces dans les compositions de services pour les applications pervasives. . . . 128
1 Menaces pour la vie privée dans les compositions de services pour les applications pervasives . . . 129
B Solutions techniques pour la sécurité . . . 132
1 Solutions pour assurer la sécurité . . . 132
C Contrôle d'accès : challenges et solutions . . . 136
1 Modèles et mécanismes d'exécution du contrôle d'accès 136
2 Principes de base du contrôle d'accès . . . 139
3 Le contrôle d'accès basé sur les rôles . . . 146
4 Le contrôle d'accès basé sur les attributs . . . 149

5 Le contrôle d'accès dédié aux processus, les mécanismes de contrôle d'accès spécifiques aux compositions de service. 153
6 Administration du contrôle d'accès 155
7 Techniques de protection spécifiques à la vie privée . 156
D Synthèse sur la sécurité et le contrôle d'accès 159
III Conclusion à l'état de l'art 160

3 Contribution **161**
I Présentation générale de la démarche 163
A Fondements conceptuels : Sécurité dirigée par les modèles . . 164
1 Principes de l'ingénierie dirigée par les modèles . . . 164
2 Problématique et objectifs 170
3 Présentation de la proposition 172
4 Niveau conception 173
5 Niveau Exécution 175
II Conception et Exécution d'une orchestration de services hétérogènes et dynamiques sécurisée par le contrôle d'accès 178
A Niveau conception : Métamodèles et modèles d'une orchestration de services hétérogènes sécurisée à l'aide du contrôle d'accès 179
1 Métamodèle de l'orchestration de services hétérogènes 179
2 Métamodèle de contrôle d'accès 182
3 Composition des métamodèles 187
4 Modèles d'une orchestration de services hétérogènes et dynamiques sécurisée à l'aide du contrôle d'accès 188
B Gestion de l'hétérogénéité et du dynamisme des services à l'exécution et exécution du contrôle d'accès 193
1 Configuration des services concret : génération des *proxies* 193
2 Calcul du contrôle d'accès 194
C Synthèse 198
III Implémentation de la démarche et validation 200
A Composants dédiés à la gestion de la connaissance et à l'exécution du contrôle d'accès 201
1 Sources d'informations 201
2 Point de décision 204
B Génération de *proxies* pour gérer l'hétérogénéité et le dynamisme des services et les sécuriser 206
1 Principe 206
2 Réalisation : utilisation de JET et déploiement du *proxy* sur un serveur Apache Axis 206
C Extension du registre de services 211

D Extension d'un environnement de modélisation et d'exécution d'une orchestration de services . . . 213
1 Présentation de JBPM et des extensions nécessaires 213
2 Niveau conception : création d'une tâche abstraite sécurisée . . . 215
3 Niveau exécution : liaison retardée et exécution du contrôle d'accès . . . 216
4 Synthèse des extensions apportées à JBPM . . . 217
E Cas d'utilisation . . . 219
1 Démarche de conception et d'exécution d'une orchestration de services hétérogènes et dynamiques sécurisés par le contrôle d'accès . . . 219
2 Définition du cas d'utilisation . . . 219
3 Conception du procédé . . . 220
4 Application du contrôle d'accès au cas d'utilisation . 222
5 Utilisation de l'orchestrateur étendu pour exécuter le procédé . . . 223
F Expérimentations et validation . . . 224
1 Coût de l'exécution du contrôle d'accès . . . 224
2 Coût de l'évaluation de la politique de contrôle d'accès 225
G Synthèse . . . 227
IV Perspectives et extensions . . . 228
A Extension de la démarche à d'autres propriétés . . . 229
1 Principe général de l'extension de la démarche . . . 229
B Extension de la démarche au contrôle d'usage . . . 231
1 Définition du contrôle d'usage et des besoins en termes d'extension . . . 231
2 Implémentation . . . 234
C Synthèse . . . 237

4 Conclusion **239**
I Contributions principales . . . 239
II Discussion et Perspectives . . . 241

IV Conclusion générale **243**

Conclusion générale **243**

Bibliographie philosophique **245**

Bibliographie informatique **251**

Chapitre I

Introduction générale

La vie privée et sa protection sont aujourd'hui largement discutées [1]. Membres de la société civile, juristes ou encore techniciens, nous sommes tous appelés à nous emparer d'une notion que l'on nous présente à la fois comme menacée, désuète ou appartenant à nos libertés fondamentales [2]. Pourtant, la vie privée telle que nous la concevons le plus souvent aujourd'hui – un droit à la protection de territoires intimes ou d'accès limité – est récente. Née sous la plume de deux juristes américains, Louis Warren et Samuel Brandeis, la revendication de la protection de la vie privée a pour origine une innovation technique qui bouleverse l'économie de la visibilité et de l'intimité victoriennes. Face à l'invention de la photographie instantanée et la publication des clichés ainsi obtenus dans la presse très en vogue au XIX^{e} siècle, les deux juristes craignent la diffusion d'images de la vie quotidienne et la révélation de scènes intimes à un public auquel on souhaiterait les dissimuler [3]. Alors que la photographie reposait jusqu'ici sur la mise en scène, le choix des costumes et des poses, la photographie instantanée fait peser le risque de ne plus pouvoir adapter sa présentation à son auditoire, de se voir voler des moments intimes et de se voir exposé à des discriminations.

Aujourd'hui encore, les controverses autour de la protection de la vie privée ont pour origine des usages techniques. L'informatisation des fichiers étatiques et les possibilités accrues de surveillance issues des innovations en informatique et, plus récemment, les « usages sociaux » des outils numériques comme les « réseaux sociaux », ont provoqué de violentes réactions. Nous serions aujourd'hui épiés, enregistrés, catégorisés et les cibles permanentes de politiques et de techniques, comme le *marketing* des traces, qui cherchent à nous influencer à partir de notre visibilité. Face à la surveillance et aux nouvelles techniques manipulatoires qui se développent à partir d'elle, la notion de « vie privée » semble offrir le rempart d'un havre d'un paix où, dissimulé aux regards judicatifs, nous pouvons être « nous-mêmes » . Pourtant, le recours à cette notion est-il complètement satisfaisant alors que, d'une part, les outils à l'origine de sa mise en question suscitent un large engouement, et que, d'autre part, ses contours sont mal définis [4] ?

Juristes, sociologues, philosophes, mais aussi informaticiens, chacun est mis à contribution pour saisir une notion polymorphe, qui pour en définir le champ d'ap-

1. Une recherche sur Google avec les mots clés « vie privée » renvoie 64 200 000 résultats. Une recherche avec le terme anglais privacy renvoie, elle, 4 990 000 000 réponses.
2. Le droit à la protection de la vie privée est inscrit dans la Déclaration des droits de l'Homme et du citoyen.
3. Voir [Waren 1890]
4. Voir [Scanlon 1975]

plication, qui pour en saisir les pratiques, qui enfin pour mettre en œuvre sa protection au travers, par exemple de « réglages de vie privée » [5]. L'étude et la mise en œuvre de la vie privée doit dès lors être interdisciplinaire. Fort de ce constat, notre travail entend adopter une double perspective – philosophique et informatique – afin de déterminer les contours et l'intérêt de la vie privée et d'interroger les moyens de sa protection. Notre enquête philosophique vise à saisir le sens de la vie privée et les relations qu'elle entretient avec la technique et notamment l'informatique. Notre recherche en informatique vise, elle, à proposer un moyen efficace de protéger la vie privée.

Concevoir la condition numérique, une enquête philosophique

Du point de vue philosophique, nous situons l'enquête sur la définition de la vie privée et sa justification dans une recherche plus large sur l'environnement technique contemporain. Notre enquête prend la forme d'une évaluation de cet environnement à la fois pour déterminer comment la technique influence nos pratiques et la définition de nos valeurs et en offrir une critique. Nous développons notre enquête selon trois niveaux d'analyse qui ont pour but de montrer que la restriction de la visibilité est nécessaire à la vie humaine et qu'elle doit être prise en compte dès la conception de nos outils techniques.

Le premier niveau d'analyse est d'ordre historique et sociologique. Face aux nombreuses craintes du développement technique dont font état les travaux philosophiques, nous nous demandons notamment si la technique forme aujourd'hui un « système » qui nous empêcherait de formuler des fins proprement humaines. Face à des approches qui condamnent la Technique à un niveau ontologique en postulant qu'elle perturbe le tissu social, nous opposons une perspective historique au travers de deux exemples. Nous remettons l'essor de la vie privée en perspective avec l'essor technique. Nous étudions aussi l'exemple de l'utilisation de la surveillance de soi par soi et par une communauté choisie afin de réaliser des projets propres. Nous concluons ainsi que la technique n'est pas nécessairement aliénante. Si certains usages techniques, notamment dans des buts de surveillance, nous contraignent fortement, d'autres, au contraire, nous permettent de formuler et mettre en œuvre des fins propres. La technique est ainsi une occasion de formuler, mettre à l'épreuve et mettre en pratique un vaste éventail de comportements et de valeurs. Face à l'ambivalence de la technique, il faut néanmoins en élaborer un cadre d'évaluation. Un tel cadre, pour saisir les implications concrètes de la technique, se doit d'être empirique.

La recherche d'un tel cadre fait l'objet de notre deuxième niveau d'analyse. Nous établissons tout d'abord comment la technique, en structurant ce que nous pouvons voir et montrer, constitue un opérateur de définition de nos comportements. Nous montrons que cet opérateur peut être utilisé dans des stratégies de gouvernement, qui nous sont extérieures ou bien dans des pratiques qui nous permettent d'approfondir

5. Nous reprenons cette expression des réseaux sociaux numériques.

la relation que nous avons à nous-mêmes. Ces dernières nous semblent plus souhaitables que les premières, potentiellement aliénantes. Aujourd'hui, les techniques sont habituellement évaluées à partir de notions utilitaristes ou déontologiques. Nous montrons qu'elles ne permettent pas de mener une étude empirique des techniques ni de justifier leur critique. Dès lors, nous proposons un cadre éthique alternatif fondé sur « l'éthique du souci de soi ». Ce cadre nous permet de recenser un ensemble de questions qui guident l'analyse des techniques et qui portent en particulier sur la structure de visibilité qu'elles permettent de mettre en œuvre.

Le troisième niveau de notre analyse est consacré à l'application de notre cadre éthique au contexte technique contemporain. Nous nous concentrons sur les outils numériques afin d'interroger le régime de visibilité qu'ils participent à construire. Nous établissons la multiplication des formes de surveillance avec l'essor d'opérateurs privés et la connexion grandissante entre les utilisateurs. L'espace numérique, qui remet fortement en question la notion de « frontière » est ainsi un espace fortement surveillé. L'identité numérique que nous possédons tous aujourd'hui est dès lors une identité stratégique, soit parce qu'elle est utilisée par les autres utilisateurs ou des opérateurs privés ou publics, soit parce qu'elle est le fruit de nos stratégies de présentation. Cette identité menace sans cesse de nous échapper et de se voir utilisée pour déterminer comme « hors de nous » notre« profil » et les actions qu'il devrait motiver. Dès lors, la limitation de la visibilité, soit sous la forme de la protection de la vie privée, soit sous la forme de la mise en place d'un oubli automatisé nous semble nécessaire pour en reprendre le contrôle. Nous établissons que la vie privée doit aujourd'hui recevoir une définition contextuelle afin de garantir que des données partagées dans un contexte donné ne sont pas communiquées pour être utilisées dans un contexte différent. Dans la mesure où, dans le monde informatique, le code joue le rôle d'une « loi », il détermine ce que nous pouvons voir et montrer et les possibilités de traitement de nos traces, il nous semble que la protection doit passer par une modification du code informatique. Face aux approches déontologiques ou utilitaristes qui placent dans l'existence de contrats d'utilisation la protection des utilisateurs, nous nous prononçons en faveur des approches qui prennent en compte des propriétés, comme la protection de la vie privée, dès la conception des outils informatiques.

***

Notre enquête philosophique fait l'objet de la première partie du présent mémoire. Elle est composée d'une introduction à la problématique philosophique puis d'une réponse en trois moments. Le premier moment est consacré à la définition de l'informatique, le second à la construction de notre cadre d'évaluation des techniques et le troisième à son application au contexte technique contemporain.

Une démarche de conception et d'implémentation de la vie privée basée sur le contrôle d'accès et appliquée aux compositions de services

Alors que nous avons défini notre acception de la vie privée et sa nécessité, il faut nous demander comment permettre de la prendre en compte au mieux dès la conception des outils numériques. Cette interrogation fait l'objet de la partie informatique de notre travail. Nous soutenons qu'il est nécessaire de savoir protéger les données d'une manière générale afin de protéger les données que l'on souhaite considérer comme privées. Dans le cadre de notre travail, nous choisissons comme domaine d'application l'approche orientée services qui est aujourd'hui un paradigme en plein essor. Nous nous concentrons particulièrement sur les applications pervasives, qui reposent sur la collecte d'un grand nombre de données et représentent dès lors une menace particulière pour la protection de la vie privée. Cette protection repose, pour nous, sur la garantie de l'intégrité et de la confidentialité des données, deux propriétés qu'il est aujourd'hui difficile d'assurer dans les compositions de services. Nous nous demandons ainsi comment permettre aux concepteurs d'applications de prendre en compte de manière optimale la protection de la vie privée tout en s'adaptant aux contraintes spécifiques des applications pervasives où les services sont hétérogènes – ils sont implémentés de manière très variée – et dynamiques – ils peuvent être disponibles ou disparaître à l'exécution d'une application.

L'état de l'art de l'approche orientée services montre que si les services web sont aujourd'hui les services les plus utilisés, d'autres types de services sont nécessaires pour construire des applications pervasives. Cependant, ces types de services ne sont pas faits pour être utilisés conjointement, d'où la nécessité de faciliter leur intégration. L'état de l'art montre aussi que les compositions sont exposées à de nombreuses failles de sécurité qui remettent potentiellement en cause l'intégrité et la confidentialité des données. Le contrôle d'accès, qui est un moyen efficace de garantir ces deux propriétés reste, aujourd'hui encore, difficile à mettre en œuvre dans les architectures orientées services, surtout lorsqu'il doit être pris en compte dès la conception.

Afin de répondre à ces manques, nous proposons une démarche dirigée par les modèles qui vise deux buts. Elle permet de prendre en compte la protection des données, et notamment des données privées, dès la conception d'une application pervasive réalisée sous la forme d'une composition de services. Elle permet de générer une application sécurisée. La phase de conception repose sur la réalisation de modèles qui visent à décrire les fonctionnalités de l'application et la politique de contrôle d'accès qui s'y applique. L'utilisation de modèles permet d'élaborer des spécifications abstraites de l'application qui ne prennent pas en compte les détails techniques de son implémentation. Nous dépassons dès lors l'hétérogénéité des services puisque nous ne considérons à ce niveau que leur fonctionnalité et non pas leur réalisation. De plus, nous retardons la sélection des services nécessaires à l'exécution ce qui nous permet de nous adapter à leur dynamisme. Nous proposons un méta-

modèle pour guider la description des fonctionnalités et un métamodèle pour guider l'expression de la politique de contrôle d'accès. Nous établissons des liens entre ces métamodèles afin de pouvoir composer les deux préoccupations. Au niveau exécution, nous proposons une architecture orientée services qui permet d'exécuter une application pervasive réalisée sous la forme d'une orchestration de services sécurisés par le contrôle d'accès. Nous élaborons des composants dédiés à la gestion du contrôle d'accès qui garantissent son application. Grâce à la sélection des services à l'exécution, notre architecture s'adapte aux services effectivement disponibles et à leur dynamisme. Le passage du niveau conception au niveau exécution est réalisé grâce à des transformations de modèle automatisées. Nous faisons ainsi abstraction de la programmation manuelle et nous capitalisons l'expertise nécessaire au développement du code de l'orchestration dans la réalisation des transformations de modèles.

Notre proposition est implémentée sous la forme d'un environnement de modélisation et d'exécution que nous présentons. Notre prototype étend Jboss JBPM, un orchestrateur que nous étendons afin de permettre la représentation d'une politique de contrôle d'accès et la génération du code nécessaire à l'exécution d'une orchestration sécurisée. Nous avons validé notre prototype sur des orchestrations de services, ce qui nous permet d'évaluer le coût de notre approche et de démontrer sa faisabilité.

***

Notre recherche informatique fait l'objet de la seconde partie du présent mémoire. Après avoir introduit la problématique informatique, nous dressons un état de l'art de l'approche à base de services et des failles de sécurité auxquelles elle est exposée. Nous faisons aussi un état des lieux des travaux sur le contrôle d'accès. Notre proposition fait l'objet de la seconde partie. Nous présentons notre démarche, les métamodèles et les modèles que nous proposons, l'implémentation à laquelle nous sommes parvenus et sa validation. Enfin, nous dressons les perspectives de notre travail.

Chapitre II

Concevoir la condition numérique, une enquête philosophique

Chapitre 1

Introduction

I L'informatique, une technique aux usages controversés

L'informatique est un domaine protéiforme. Il en est de même pour la réflexion philosophique en la matière. Philosophies du langage, éthique, esthétique ou encore philosophies morale et politique prennent l'informatique comme objet d'étude. Néanmoins tous ces domaines se heurtent au manque de définition d'un champ scientifique et technique jeune dont la diffusion à grande échelle est récente. Initialement onéreux et utilisés principalement dans les domaines bancaires et militaires, les ordinateurs sont rapidement devenus omniprésents. Ils sont aujourd'hui employés pour communiquer, travailler ou encore se divertir. Pourtant, la diffusion de l'informatique à un large public est récente. Il ne s'est écoulé que quelques décennies entre la construction des premiers ordinateurs au milieu du XX^{e} siècle et la production des téléphones mobiles aujourd'hui. Ces derniers, le développement rapide de réseaux numériques de communication ainsi que l'essor des réseaux sociaux numériques sont probablement les exemples les plus significatifs de la place contemporaine de l'informatique dans notre vie.

Néanmoins, tous ces nouveaux usages provoquent des réactions contrastées. Tel est le cas des réseaux sociaux numériques. Dédiés, dans un premier temps, aux étudiants des universités américaines, ils étaient considérés comme des outils ludiques et efficaces pour garder contact. Aujourd'hui accessibles à tous, ils permettent de jouer, s'occuper et communiquer. Certains réseaux sont destinés à un usage spécifique : les réseaux sociaux professionnels se présentent comme des moyens de développer sa carrière. Néanmoins, tous ces réseaux font l'objet de critiques. Dès lors, ils favoriseraient le développement d'une personnalité superficielle à la recherche de la séduction d'un public toujours plus large. Ce faisant, ils exposeraient leurs utilisateurs à la perte de leur intimité.

Face à ces risques, les utilisateurs semblent posséder peu de moyens de contestation. Les conditions d'utilisation de ces réseaux changent régulièrement, ce qui complique la compréhension de ce que le réseau peut effectivement faire avec les données des utilisateurs. Les outils légaux, eux, s'appliquent difficilement à des fournisseurs de services numériques dont les sièges sont hebérgés par des pays aux législations hétérogènes. Enfin, l'engouement pour certains outils numériques, malgré les révélations sur leurs pratiques en termes de collecte et de traitement des données, ne témoigne-t-il pas d'un désintérêt des utilisateurs pour ces questions ?

Les enquêtes philosophiques sur l'informatique et ses usages manifestent, elles aussi, des attitudes contradictoires. D'une part, l'informatique est célébrée car elle

permet d'enregistrer et de partager facilement et rapidement les connaissances. Ainsi fait-elle naître l'espoir d'une « intelligence collective » pour P. Levy [1]. Elle est aussi conçue comme un nouvel outil politique dans la mesure où les réseaux sociaux numériques, par exemple, ont largement été utilisés dans les mouvements de contestation politique récents. Néanmoins, les études philosophiques soulignent aussi les aspects négatifs des nouvelles possibilités de diffusion des données et de communication [2]. L'augmentation des puissances de calcul et de stockage des données, l'élargissement de leur collecte à toutes les facettes de notre vie et l'essor des usages distribués de l'informatique conduisent souvent les auteurs à conclure à l'existence d'un « système d'information » planétaire au sein duquel nous serions surveillés en permanence.

II La surveillance, angle d'approche privilégié de l'informatique et de ses usages

Big Brother, le « panoptique » ou encore la « société de contrôle » sont les images couramment invoquées dans les réflexions sur notre société où l'ordinateur est devenu omniprésent [3]. Ces images se recoupent dans la mesure où elle font toutes de l'ordinateur une machine qui abolit les frontières géographiques et sociales. Tous les espaces dans lesquels il pénètre sont « mis en réseau » , d'où la dénonciation d'une surveillance généralisée des utilisateurs d'outils informatiques. La crainte de la surveillance existait déjà face à la constitution de grandes bases de données, notamment par les Etats, mais elle est augmentée par l'enregistrement permanent des moindres détails de notre activité ainsi que par le croisement des données issues de ces enregistrements. Alors que la surveillance ne s'applique habituellement qu'aux individus considérés comme dangereux, des scandales fréquents révèlent combien nous sommes épiés, souvent à notre insu.

Pourtant, ces craintes sont-elles justifiées ? Le plus souvent, nous nous exposons volontairement. Ainsi divulguons-nous des informations qui nous concernent pour utiliser un service numérique ou pour en rendre l'usage plus agréable et plus « personnalisé » , c'est-à-dire plus adapté à nos goûts et à notre personnalité. Les critiques de la surveillance ont ainsi pour but de nous alerter sur les dangers potentiels de cette attitude. Néanmoins, les appels à la protection de la vie privée, par exemple, ne sont-ils que les traces de notions qu'il faudrait aujourd'hui laisser de côté ? Les usages contemporains de l'informatique sont en effet régulièrement accusés de remettre en cause l'existence de la sphère privée. Le cas de la vie privée est

1. Voir [Levy 1998].

2. Une recherche sur *Google Scholar*, une base de données d'articles scientifiques avec les termes *death of privacy* livre presque 2 millions de résultats. La plupart des titres des articles contiennent les mots *digital* ou *computer*.

3. Big Brother est un programme informatique qui, dans 1984 de George Orwell, assure le contrôle de l'ensemble de la société par une surveillance permanente des citoyens. Voir [Orwell 1961]. Le « panoptique » est un modèle de prison imaginé par Jérémy Bentham dans lequel les prisonniers sont sans cesse visibles aux gardiens. Voir [Bentham 1977]. Enfin, la « société de contrôle » est l'expression utilisée par Gilles Deleuze pour caractériser l'informatisation de nos activités qui sont ainsi toutes mises en réseau.

intéressant dans la mesure où la notion est valorisée comme un moyen de maintenir son autonomie, par exemple en contrôlant l'accès au domaine personnel que l'on permet à autrui. La perte de la vie privée à cause de l'informatique signifierait ainsi que la technique nous dépossède d'une partie de notre autonomie, manifeste notamment dans notre capacité à contrôler notre visibilité. La difficulté pour un individu de déterminer précisément l'identité de ceux qui possèdent ses données et le contenu des données connues semblent appuyer cette position.

Pourtant, les promoteurs des moteurs de recherche et des réseaux sociaux numériques notamment ont beau jeu de répondre à ces critiques. Tout d'abord, l'adhésion très large des utilisateurs à leurs services ne plaide-t-elle pas pour une acceptation de la visibilité des conduites, aussi importante soit-elle ? Cette première question, d'ordre factuel, est complétée par une question morale. Pourquoi vouloir nous cacher si nous n'avons rien à nous reprocher [4] ? Cette dernière interrogation subvertit le rapport à l'opacité qu'entretient la vie privée. L'opacité, même relative, de la vie privée est conçue comme un moyen de se retirer et de mener tranquillement sa vie en poursuivant des fins propres. Au contraire, dans le cadre de cette question, l'opacité est vue comme un moyen de dissimuler des actes critiquables. Dès lors, la visibilité se mue en un impératif moral : se montrer, c'est attester le fait que l'on n'a rien à cacher.

Au cœur de l'analyse des usages informatiques contemporains se tient souvent une tension sur le rôle de la visibilité. La visibilité doit-elle être utilisée comme un outil de contrôle social ? La limitation de la visibilité, au contraire, est-elle nécessaire afin de préserver des espaces dans lesquels chacun peut poursuivre des fins propres, loin de tout regard potentiellement réprobateur ? Si nous sommes effectivement aujourd'hui de plus en plus visibles, comment parvenir à poursuivre des fins qui nous appartiennent ?

La réponse à l'ensemble de ces questions appelle une enquête plus large, particulièrement sur la définition de l'informatique, face à laquelle nous nous trouvons devant un « vide conceptuel » [5]. Si l'informatique modifie en profondeur notre vie, il faut en saisir les enjeux. Il s'agit en effet de savoir ce qu'est l'informatique et comment se constitue aujourd'hui une « condition numérique » qui influence notre existence toute entière.

III Qu'est-ce que vivre à l'ère du numérique ?

La réponse à cette question appelle à la fois une enquête sur la définition de « l'ère du numérique » et l'identification de ses effets sur notre vie. Nous menons cette enquête à partir de trois angles d'analyse, respectivement fondés sur trois hypothèses principales.

4. Eric Schmidt, PDG de *Google*, affirmait ainsi dans une interview de 2009, que si nous souhaitions dissimuler nos actions, et notamment refuser leur enregistrement par les moteurs de recherche, c'est que ces actions étaient peut être répréhensibles ou critiquables. Voir http ://news.cnet.com/8301-13860_3-10413473-56.html.

5. Voir [Moor 1985].

Le premier angle d'analyse fait l'objet du premier chapitre. Il prend pour objet l'essence de la technique, et plus spécifiquement de l'informatique. L'informatique possède-t-elle, en tant que technique, des traits spécifiques ? Si tel est le cas, permetent-ils d'en condamner l'usage *a priori* ? L'utilisation des outils informatiques en réseau, la construction d'outils d'aide à la décision en grande partie automatisés ou encore les possibilités accrues de surveillance sont les éléments centraux de la critique de l'informatique. La prise en compte exclusive de ces éléments conduit à définir l'informatique comme une technique qui a envahi et « perturbe » le fonctionnement de notre société. Nous opposerons à cette conception un niveau d'analyse historique fondé sur l'étude des usages techniques et de leurs effets. Les critiques, du point de vue de la surveillance, dont font l'objet les outils informatiques sont en effet similaires à celles déjà adressées à d'autres moyens de communication plus anciens. L'analyse historique montre que les technologies de communication constituent toutes un « agent technique » dans la mesure où elles modifient les modalités de notre présence et structurent ainsi ce que nous pouvons rendre visible aux autres et à nous-mêmes.

Le second niveau de notre analyse est présenté dans le second chapitre de ce mémoire. Il s'attache à la portée éthique de la technique et de l'informatique. Nous établirons un modèle de la relation entre la technique et la subjectivité fondé sur la distinction entre des techniques orientées vers le gouvermenent des conduites et des techniques employées afin de développer sa subjectivité. Cette analyse permet de montrer l'importance de la médiation de la technique dans le rapport éthique à soi même lorsque l'on définit l'éthique comme un ensemble de comportements désirables. Afin d'établir un cadre d'étude et d'évaluation des outils numériques, nous nous tournerons vers « l'éthique du souci de soi » dont nous établirons les points saillants.

Le troisième moment de notre analyse est consacré à l'évaluation des usages numériques. Qu'est-ce que « l'ère numérique » et quelles critiques pouvons-nous en faire à partir de notre enquête éthique ? nous demanderons-nous dans le troisième chapitre à partir des outils que nous avons élaborés lors de notre étude du souci de soi. Nous débuterons par l'analyse de quelques usages numériques et leurs relations afin de montrer les qualités spécifiques de l'espace numérique. Ce premier moment de la réflexion permettra de reprendre l'analyse du thème de la surveillance et de sa portée. Nous établirons que la notion de « frontière » est fortement mise à mal à l'ère numérique. La remise en question de cette notion constitue le fondement principal des critiques que nous élaborons dans un second temps. En effet, les frontières – au sens géographique comme au sens social – apparaissent nécessaires au développement d'un agent responsable de ses actions et de sa vie. Enfin, nous étudierons la portée pratique que l'on peut donner à ces critiques. Nous définirons la programmation informatique et la construction de réseaux numériques comme les gestes créateurs du monde numérique. Nous enquêterons sur la manière dont ces gestes peuvent participer à des pratiques de liberté dans le monde numérique.

Chapitre 2

Informatique et agent technique

Matrix est un film de science-fiction dans lequel tous les humains sont reliés à des machines devenues autonomes. Alimentés, nettoyés et maintenus par des machines, les humains vivent dans une réalité onirique, la « matrice » , générée par des ordinateurs connectés à leur esprit. Les programmes exécutés par ces ordinateurs ont acquis une conscience et des buts propres. Ils contrôlent les actions humaines, pourchassent les dissidents et cachent aux hommes le fait qu'ils ne vivent qu'une illusion. L'humanité est ainsi réduite en esclavage : les humains conservés dans un rêve permanent sont utilisés comme source d'énergie des machines et des programmes qui les contrôlent. La situation décrite par *Matrix* fait probablement écho à notre situation. Aujourd'hui connectés en permanence aux réseaux numériques, équipés d'outils informatiques mobiles qui nous suivent, guident nos trajets et nous adressent des recommandations, ne vivons-nous pas dans une forme de « matrice » dans laquelle les outils informatiques seraient devenus un filtre entre la réalité et nous ?

La domination des humains par les techniques est un thème de prédilection de la littérature et du cinéma d'anticipation. *Matrix* n'est qu'un exemple de dystopie qui dénonce les dérives potentielles du développement technique, et particulièrement de l'informatique. La normalisation de la société au moyen de techniques biologiques décrite dans *Le Meilleur des mondes* ou encore la surveillance à laquelle l'ordinateur *Big Brother* la population de *1984*, illustrent, elles aussi, les craintes suscitées par la technicisation de la vie humaine [1]. Parmi ces craintes, le thème de la surveillance occupe une place particulière. Elle semble en effet intimement liée au développement technique et menacer l'autonomie humaine.

Les images élaborées par la science-fiction sont aujourd'hui reprises de manière fréquente dans l'analyse des outils informatiques montrant que les craintes dont elles font état sont toujours d'actualité en matière d'outils numériques. Ils sont accusés de favoriser la surveillance en multipliant les sources d'observation potentielles, de nous exposer à l'usurpation d'identité ou encore à la discrimination. Enfin, l'usage croissant de techniques prédictives à partir des grandes masses de données aujourd'hui disponibles nous soumettent à de nouvelles formes d'incitations et de normalisation. L'accès au courrier électronique, aux relations sociales se font en grande partie au travers d'outils informatiques qui orientent notre interaction et le contenu auquel nous avons accès. Le plus souvent, la sélection et la présentation du contenu sont réalisées de manière automatique, à partir d'algorithmes qui peuvent « évoluer » d'eux-mêmes et de jeux de données de taille telle qu'un esprit humain ne peut les manipuler. Enfin, avec le développement des réseaux de communication,

1. Voir [Huxley 2010] et [Orwell 1961].

les objets et les différents prestataires de services avec lesquels nous interagissons peuvent échanger des informations qui nous concernent, nous soumettant à ce qui semble être un réseau de surveillance d'une taille sans précédent.

Les mouvements de contestation des outils numériques, la multiplication des actions en justice contre la collecte et l'utilisation de données témoignent de la résistance d'une partie de l'opinion publique face au monde numérique. Une telle résistance se lit aussi dans un pan de la pensée philosophique qui prend l'informatique comme objet d'étude. Les outils informatiques, aujourd'hui majoritairement utilisés en réseaux, semblent tout savoir de nous. Ils participent à nos prises de décisions et à nos activités. Ne sommes nous dès lors pas face à un « système technique » composé d'outils informatiques qui, en prenant un poids de plus en plus important dans nos vies, nous départirait d'une partie de notre autonomie ?

Les usages de l'informatique sont cependant aujourd'hui extrêmement divers et la discipline, encore récente, doit être définie. Science, technique, domaine économique, le terme connaît en effet de nombreuses acceptions que nous articulerons tout d'abord. Cela étant fait, nous interrogerons la spécificité de l'informatique par rapport aux autres techniques. L'« abstraction » de l'informatique comme technique de calcul – qui peut être appliquée à un grand nombre de domaines – et la distribution des outils informatiques sont les deux points qui retiendront notre attention. Ils permettent de rendre compte de la diffusion de l'informatique à l'ensemble de la société et de son rôle comme « agent » dans nos vies. Les fins que nous poursuivons sont ainsi formées au contact des techniques, ce qui semble valider les craintes issues des dystopies. Néanmoins, l'existence d'un tel agent est-elle un obstacle à la formation de fins propres ? C'est à cette question que nous consacrerons la dernière partie de notre analyse.

I Définir l'informatique : une technique abstraite

Mot-valise formé en 1957 à partir de la contraction des substantifs « information » et « automatique » , le terme « informatique » est polysémique [2]. D'une part, l'informatique est une science qui a pour objet le calcul et le traitement de l'information. D'autre part, l'informatique est une activité industrielle, technique et économique qui produit et utilise des machines qui effectuent des calculs automatiques. Ces machines sont nommées « ordinateurs » , ou « calculateurs » . La pensée philosophique ne se trouve, par rapport à l'informatique, peut-être plus face à un « vide conceptuel » [3]. Cependant, les acceptions du terme sont dispersées et doivent être articulées pour le comprendre.

L'informatique contemporaine comme technique repose sur un ensemble d'avancées théoriques, au rang desquels la création de nouveaux « modèles de calcul » . La « machine de Turing » , par exemple, pensée dans les années 30, décrit le fonctionnement d'une machine à calculer et de sa mémoire [4]. Elle permet, encore aujourd'hui, de résoudre des problèmes liés à la calculabilité des algorithmes et à leur complexité. La notion « d'algorithme » , n'est bien sûr pas spécifique à l'informatique. Ainsi Euclide avait-il déjà formalisé un algorithme de recherche du « plus grand diviseur commun » de deux entiers naturels vers 300 avant J.C [5]. Cependant, comme le montre l'appel à la notion de « machine » dans le cadre du modèle de Turing, l'informatique se concentre sur les calculs mécanisés. La mécanisation des calculs n'est cependant pas non plus suffisante pour définir l'informatique. Les abaques, par exemple, existent depuis plusieurs centaines d'années. Néanmoins, l'informatique met en jeu de nouveaux outils mécanisés de classement et de traitement des informations. Ainsi les « mécanographes » du début du siècle reposent-ils sur l'utilisation de cartes perforées pour enregistrer et traiter les données. À partir de 1947, l'invention du transistor, qui est le composant électronique de base des ordinateurs permet le développement de l'informatique que nous connaissons aujourd'hui. Les ordinateurs accélèrent et étendent les capacités de calcul des outils déjà existants. Avec le développement de composants électroniques de petite taille et peu chers, le stockage des informations est rendu plus facile. La portée même du calcul est radicalement étendue. Par rapport à d'autres techniques de calcul, l'informatique, comme technique, introduit une rupture quantitative, en permettant d'effectuer de plus en plus de calculs sur de jeux de données, et une rupture qualitative – ces calculs sont de plus en plus rapides.

Enfin, par rapport à d'autres techniques, l'informatique se distingue par son « abstraction » d'un domaine d'application particulier. Les nombres ne sont qu'une manière de représenter n'importe quel type d'information et les algorithmes les

2. Le terme a été forgé par Karl Steinbuch. Voir [Steinbuch 1957].

3. Voir [Moor 1985]. Depuis 1985, de nombreux travaux se sont consacrés à la définition de l'informatique. Voir, par exemple, [Varenne 2009], où l'auteur se concentre sur la définition de l'informatique du point de vue du calcul symbolique.

4. Turing décrit sa machine dans [Steinbuch 1957].

5. Voir [Euclide 1990].

moyens de traiter ces données. L'informatique s'applique ainsi petit à petit à des domaines de plus en plus variés. Tel est le cas du traitement de texte, des opérations bancaires ou encore de la recherche génétique. C'est pour cette raison que l'informatique peut être considérée comme la discipline consacrée aux algorithmes et à leur utilisation dans le cadre de modèles de calculs.

La notion de « modèle » , que l'on retrouve aussi bien en informatique théorique que dans les applications de l'informatique nous semble ici cruciale dans la définition de l'abstraction de l'informatique. Les traitements informatiques s'appliquent à des jeux de données qui constituent des « modèles » de la réalité, c'est-à-dire une représentation issue d'une sélection d'éléments d'intérêts. En tant que représentation, le modèle repose sur un mécanisme d'abstraction. De telles abstractions peuvent, par exemple, décrire le fonctionnement d'un programme. Les modèles sont alors distincts d'une réalisation spécifique. La solution à un problème donné qu'ils présentent est ainsi générique et peut-être réutilisée. Les modèles peuvent aussi décrire les données traitées. Dès lors, le modèle est le paramètre d'entrée d'un programme informatique. Il permet l'application d'un ensemble d'algorithmes, c'est-à-dire de démarches systématiques de réalisation de calcul. Comme toute portion de la réalité peut être modélisée, l'informatique peut s'appliquer à tous les domaines. Dès lors, il revient non seulement à l'informaticien de définir des algorithmes de calcul, mais aussi de modéliser la portion du monde auquel le programme qu'il développe s'applique. Dans cette perspective, l'informatique est ainsi la technique qui consiste à sélectionner les éléments de la réalité à traiter et à programmer les effets du calcul sur ces éléments et la réalité qu'ils décrivent. De la définition d'une science du calcul, il faut ainsi passer à la définition de l'informatique comme une technique aux effets réels et multiples.

L'abstraction de l'informatique et la diversité de ses effets peuvent se lire dans les ordinateurs. À l'origine encombrants et dédiés uniquement au calcul, les ordinateurs sont aujourd'hui des « machines universelles » utilisées dans tous les domaines de notre vie. Certes, plusieurs domaines économiques, par exemple, reposaient déjà sur un ensemble d'outils de stockage et de traitement de l'information, comme les archives ou la machine à calculer. Néanmoins, en plus des ruptures qualitatives et quantitatives liées à « l'informatisation » , l'ordinateur introduit une rupture vis-à-vis des outils qu'il remplace en rassemblant leurs fonctionnalités. Un simple ordinateur de bureau peut ainsi jouer le rôle de machine à calculer, de téléviseur, de téléphone ou encore d'encyclopédie. Alors que chaque outil remplacé par l'ordinateur possédait une fonction propre et n'était pas conçu pour quitter le lieu sur lequel il devait être utilisé, les ordinateurs sont, pour une grande part, de taille réduite. Ils peuvent ainsi être facilement transportés. Les nouveaux téléphones mobiles sont en fait des ordinateurs suffisamment puissants pour effectuer l'ensemble des opérations d'un ordinateur de bureau[6]. Nos déplacements sont ainsi eux-aussi « informati-

6. L''*iPhone*, téléphone commercialisé par la firme *Apple* depuis 2007 est probablement l'archétype du *smartphone* on « téléphone intelligent » : la fonction téléphone est réduite à une icône affichée parmi de nombreuses autres. Chaque icône représente un programme qui peut permettre, à partir du téléphone, de jouer, travailler, se divertir ou encore entretenir des interactions sociales.

sés » . La petite taille des ordinateurs contemporains permet, elle, de les rendre invisibles et d'imaginer des lieux informatisés qui observent ceux qui les parcourent et interagissent avec eux.

Enfin, la plupart des outils informatiques peuvent aujourd'hui communiquer au travers de réseaux numériques dont le plus connu est l'Internet. Les données nécessaires à une fonctionnalité peuvent ainsi être stockées sur un serveur auquel on se connecte à distance. Au travers du réseau, des applications peuvent être construites et des groupes d'internautes peuvent se constituer. Les outils informatiques contemporains, mobiles et connectés en permanence semblent ainsi avoir permis de construire un « système d'information » , à l'échelle planétaire. Ce système est composé d'objets qui peuvent acquérir, traiter et échanger de l'information sur des sujets aussi variés que le contenu de votre frigidaire, les films que vous préférez ou encore votre courrier électronique. Dans une certaine mesure, ce système fonctionne de manière autonome et nous affecte en retour. Les téléphones portables, par exemple, peuvent enregistrer la localisation de leur utilisateurs et lui adresser des recommandation d'achat ou l'alerter de la présence de personnes qu'il connaît en fonction de cette information. Notre utilisation des sites Internet fait l'objet d'un enregistrement constant afin notamment d'identifier nos goûts et de nous faire parvenir des publicités adaptées. L'utilisation de ces sites par des personnes qui nous ressemblent et que nous connaissons est, elle aussi, utilisée afin de nous inciter à consommer comme elles.

De technique de calcul, l'informatique devient ainsi technique d'enregistrement de l'information et d'influence de nos comportement au moyen d'un système d'information. Les réactions face à la construction d'un tel système sont contrastées. Il suffit, pour s'en assurer, de s'intéresser aux débats récurrents entre ceux qui rejettent viscéralement tel ou tel outil informatique – comme les réseaux informatiques – et ceux qui, au contraire les promeuvent de manière inconditionnelle. Ces débats entre les défenseurs de la technique et de l'extension de son usage et leurs opposants se retrouvent, aussi, dans la communauté académique. Un grand nombre de travaux font des outils informatiques des moyens de promotion de la liberté d'expression. Néanmoins, un nombre au moins équivalent d'articles dénonce l'extension potentielle de la surveillance à tout un chacun grâce à l'informatique [7]. Puisque la plupart des objets de notre quotidien sont informatisés ne sommes-nous pas devenus dépendants de l'informatique ? Dans la mesure où ces outils enregistrent notre activité et l'influencent, ne sommes-nous pas devenus les jouets du système d'information que nous avons construit ? Nous nous consacrons désormais à l'examen de cette possibilité.

7. Une recherche sur *Google* avec les termes « privacy death computer » renvoie 620.000 résultats. On peut noter, par exemple, l'*ACM Computer, Freedom and Privacy* Conference qui étudie les liens entre les développements de l'informatique et la liberté.

II Spécificités de l'informatique comme technique

Dès ses premiers jours, l'informatique est la cible de critiques virulentes. En 1977, alors que l'utilisation de l'informatique en réseau n'en est qu'à ses balbutiements, paraît *Le système technicien*, ouvrage que Jacques Ellul consacre entièrement à la dénonciation des effets de l'informatisation de la société [8]. Cet ouvrage a reçu une large réception à l'étranger et reste à la source de nombreuses critiques de l'informatique [9]. L'ouvrage est écrit alors que le réseau *Advanced Research Projects Agency Network* (ARPANET), ancêtre d'Internet n'a qu'une dizaine d'années et qu'il n'est pas accessible au grand public. Quelques années avant sa publication, a éclaté en France l'affaire « Safari » du nom du « Système Automatisé pour les Fichiers Administratifs et le Répertoire des Individus » que souhaitait mettre en place l'Etat français et qui a donné naissance, en réaction, à la Commision Nationale Informatique et Liberté (CNIL).

Ouvrage de critique des développements de l'informatique, le livre est construit autour d'une thèse centrale : la technique est devenue aliénante. Etymologiquement, la technique désigne les savoir-faire spécifiques d'un artiste. Avec la création de machines de plus en plus perfectionnées, le terme de « technique » a ensuite été utilisé pour désigner les procédés de production de machines spécifiques et leur utilisation. Ainsi faut-il parler, dans ce cas, « des techniques » . Néanmoins, Ellul soutient que cette distinction n'a plus de sens. Tous les domaines techniques s'appuient sur des ordinateurs. Il sont ainsi tous connectés au moyen de réseaux numériques. Par conséquent, la « Technique » est devenue un « système » . Les possibilités de communication accrues permettent à la fois de séparer les centres de production et de décision, mais aussi de rendre des espaces distincts interdépendants. Les prises de décision dans une usine, par exemple, ne sont plus liées à l'usine seule. Des informations issues des données issues d'autres usines, de bases de données financières et d'un grand nombre d'autres sources participent à la construction de décisions qui sont prises grâce à la médiation des ordinateurs. Si les réseaux informatiques sont décentralisés, Ellul note qu'ils dispersent le pouvoir de décision et le contrôle que l'on a sur les données. Le système aliène ainsi puisqu'il diminue les possibilités d'action. Il ne s'agit plus de se battre contre un point fixe – le contremaître ou le chef de l'usine – mais contre un réseau diffus.

La « Technique » est ainsi aussi le nom d'une organisation sociale. Elle désigne un ensemble de procédés d'organisation, de contrôle et d'interconnexion qui dépassent les frontières d'un domaine scientifique ou industriel spécifique. Tous les domaines s'appuient sur l'informatique et cette dernière se développe grâce au perfectionnement des autres champs techniques comme l'électronique. Certes, Ellul note que le phénomène technique préexiste à l'informatique. Néanmoins, pour l'auteur, la révolution numérique se distingue des autres révolutions industrielles. Les premières révolutions industrielles reposent principalement sur les changements de

8. Voir [Ellul 2004].

9. Même si Ellul reste peu étudié en France, il faut ainsi noter l'existence de l'*International Jacques Ellul Societ* dédiée à la diffusion des travaux de l'auteur.

source d'énergie. On passe ainsi du charbon à l'électricité puis à l'énergie nucléaire. La révolution numérique, pour Ellul, ne porte pas sur les matières premières mais sur l'ordre social. Dans la mesure où l'informatique fait varier la distance et les liens entre les centres de décisions et les centres où ces décisions s'appliquent, elle permet de mettre en œuvre de nouveaux procédés d'organisation et de décision. Le développement technique nécessite une organisation spécifique. Les « technocrates » , c'est-à-dire les experts techniques et les responsables politiques qui accordent un rôle prédominant à la technique, acquièrent ainsi une place de premier plan dans la société. Enfin, le développement technique engendre des problèmes que seule la technique peut résoudre. Les grands volumes de données, par exemple, ne peuvent être traités qu'au prix d'une optimisation des outils de calculs qui, pour être le plus précis, nécessitent le perfectionnement des outils de collecte des données.

Par conséquent, le « système technicien » diagnostiqué par Ellul s'entretient tout seul et se développe sans fin. En devenant une « interface » obligatoire et généralisée, la Technique est devenue le « milieu » dans lequel nous vivons. Ce milieu oriente nos actions et participe à nos prises de décision. Dans la mesure où nous ne pouvons plus agir sans lui, il constitue un élément primordial dans la définition de notre identité. Il faut prendre ici le terme de « fin » aussi bien dans le sens de stade final que de but : alors que l'utilisation d'un instrument obéit habituellement à un projet et répond à un ensemble de fins, Ellul note que le développement de la Technique, qui appelle un développement technique toujours plus important, ne peut être qu'aveugle. C'est en ce sens que les fins apparentes – la production des biens et leur consommation – sont toujours réintégrées dans des moyens de développer plus de techniques. Alors que l'on peut avoir l'impression d'être plus libre puisqu'on a plus de choix – de consommation, de divertissements, etc. - on ne peut jamais choisir qu'entre des objets techniques dont l'existence même conduit à construire un monde où la prise de décision est toujours plus décentralisée et éloignée de celui auquel elle s'applique et entraîne une production toujours plus importante. Le choix de consommer ou de décider à partir d'objets techniques entraîne ainsi le développement de la Technique. Dans ce détournement des fins humaines, Ellul lit l'essor d'un « agent technique » , c'est-à-dire d'une entité technique qui décide de notre vie et l'influence.

Ellul critique ici une conception naïvement optimiste du progrès ainsi qu'une forme de « fétichisme » de la Technique. Il dénonce les utopies dans lesquelles la Technique apparaît comme la réponse ultime aux problèmes sociaux. Il dénonce aussi la recherche à tout prix des nouvelles techniques et de leur perfectionnement. Ses critiques, bien que formulée à la lueur des premiers jours de l'informatiques ne peuvent que trouver un écho à notre époque où la mise sur le marché d'un nouveau modèle de téléphone mobile à la mode engendre une véritable frénésie. La médiation de plus en plus importante des objets techniques dans nos activités quotidiennes semble même devoir pousser à accentuer la critique d'Ellul. De fait, de nombreux outils informatiques influencent de manière plus ou moins évidente nos actions. Les organismes de crédits, par exemple, ont recours au classement automatisé de leurs clients potentiels afin de déterminer l'octroi d'un prêt. Les millions de recherches effectuées chaque jour sur un moteur de recherche sont, elles, filtrées et adaptées

en fonction des informations glanées à notre propos au fil de notre navigation sur l'Internet.

Aux traits qui distinguaient déjà pour Ellul l'informatique des autres techniques, il faut ainsi probablement rajouter la création d'un monde personnalisé où, en tant que consommateurs nous sommes régulièrement fidélisés et la cible de publicités ciblées. La production du début du siècle obéissait, comme le montre l'exemple du fordisme, à une logique de rationalisation des coûts pour l'entreprise et d'achat de masse [10]. Ainsi la *Ford T* n'était-elle produite que dans un nombre de couleurs limité susceptible de plaire au plus grand nombre. Aujourd'hui, la plupart des produits – voitures, téléphones ou sites Internet par exemple – peuvent être personnalisés par l'utilisateur ou le sont automatiquement. Le monde numérique nous met ainsi face à un monde profondément esthétisé et facilement paramétrable. Cette personnalisation conduit à une intégration plus forte des individus qui sont incités à consommer plus.

Dans la perspective d'Ellul et d'une partie des penseurs contemporains de la technique [11], il faudrait dès lors conclure à l'existence d'une personne abstraite, la « Technique » . Tous se rejoignent pour en dénoncer les effets. Nous soumettant à des fins qui nous échappent, suivant un développement aveugle, ayant développé une structure systématique, la Technique nous dominerait. Si Ellul se défend de céder à toute forme d'anthropomorphisme, sa personnification de la Technique lui accorde pourtant de nombreuses intentions. Ses analyses répondent aux angoisses liées aux premiers temps de l'informatisation et peuvent trouver des échos dans certains usages contemporains de l'informatique. Néanmoins, elles se fondent sur un parti pris idéologique – Ellul n'hésite pas à qualifier la technique de « cancer » [12] - qui fait du caractère systématique de la Technique une pétition de principe plus qu'une thèse fondée sur des faits [13]. C'est pour cette raison que nous prenons nos distances avec la thèse développée d'Ellul et ses successeurs.

10. Voir [Ford 2013] où Ford théorise les avantages économiques de la standardisation pour les entreprises.

11. La position alarmiste d'Ellul est partagée par plusieurs philosophes de la technique qui lui sont contemporains. H. Marcuse, par exemple, décrit, dans l'introduction à l'*Homme unidimensionnel*, la technique comme un « système de domination » . Heidegger conçoit, lui, la technique comme une manière de violenter la nature. Voir [Marcuse 1989] et [Heidegger 1958]. Ellul se rapproche d'ailleurs d'une telle position lorsqu'il affirme que la technique met en danger le « milieu social » . Voir [Ellul 2004][p.30]

12. Voir [Ellul 2004][p.92].

13. Dans son *Histoire des techniques*, Bertrand Gilles adopte ainsi un point de vue antagoniste à celui d'Ellul en soutenant que toutes les techniques « font système » puisqu'elles sont interdépendantes. Voir [Gille 1978].

III La technique comme agent

À la thèse d'Ellul, dont les fondements historiques nous semblent céder la place à une position idéologique, nous proposons de répondre à l'aide des ressources de l'histoire et de la sociologie. Nous interrogeons tout d'abord le statut du « système technique » contemporain : l'informatique est-elle la seule technique à faire système ? Après avoir montré que tel n'était pas le cas, nous interrogeons la possibilité de formuler des fins propres au contact de la technique.

A Toute technique est-elle systématique ?

« L'agent technique » tel que l'analyse Ellul tire son efficacité du caractère systématique de l'organisation technique issue de l'informatisation. Cette systématicité influence l'espace dans lequel nous vivons. Certes, l'ordinateur a pour effet d'unir de manière informationnelle des espaces physiquement distincts et d'accélérer les communications entre ces différents lieux. Néanmoins, la systématisation des usages techniques sous l'influence des techniques de communication n'est pas spécifique à l'informatique. Dès le XIX^{e} siècle, les juristes américains critiquent l'association de la photographie instantanée et de la presse. Combinées l'une à l'autre, ces deux techniques représentent le risque de pouvoir disséminer auprès d'un large public et hors du domicile des éléments privés capturés par la photographie. Dès lors, l'invention de la photographie instantanée et son association à un support de diffusion risque, pour les auteurs, d'interdire toute intimité dans la mesure où elle ouvre la maison à la rue [14]. Plus récemment, mais avant l'utilisation généralisée de l'ordinateur, MacLuhan théorisait ainsi déjà l'existence d'un « village planétaire » construit grâce au téléviseur et de la radio [15]. Ces deux moyens de communication démultiplient la présence en substituant à la présence physique une modalité de présence sonore ou visuelle. Nous pourrions, à propos des technologies de communication, multiplier les observations quant aux modalités de présence qu'elles mettent en jeu et la rapidité de leur transport. Néanmoins, il nous semble que l'on trouve dans cette brève étude une première nuance à apporter à la position d'Ellul : si l'informatique se distingue par un mode de présence et de communication spécifique, elle n'est pas la première à modifier la manière dont nous apparaissons aux autres et pouvons interagir avec eux. La révolution informatique est de ce point de vue plutôt quantitative que qualitative : la communication par ordinateur permet de cumuler les modalités auditives, visuelles, textuelles et l'interaction qui, avant son invention, étaient disjointes [16].

Le second point de nuance porte sur le caractère systématique des techniques lui-même. Dans son *Histoire des télécommunications*, P. Flichy étudie ainsi la corrélation entre le développement des villes américaines au XIX^{e} siècle et le dévelop-

14. Voir [Waren 1890].
15. Voir [McLuhan 1967].
16. Nous sommes ainsi ici conduits à prendre nos distances avec la définition qu'Ellul propose de l'informatique. Nous prennons nos distances avec l'angle de vue uniquement négatif qu'il prend sur ses effets dans le deuxième chapitre de notre travail.

pement des tramways. L'auteur note ainsi que le perfectionnement des tramways, c'est-à-dire l'augmentation de leur vitesse et le développement de nouvelles lignes est crucial dans la création des banlieues résidentielles. Les espaces domestiques peuvent d'autant plus facilement être séparés des centres de décisions politiques et des espaces de production que la communication entre ces différents lieux est facilitée. En réponse, l'architecture victorienne se développe et les maisons sont de plus en plus adaptées à une vie resserrée autour de la sociabilité, de la vie de famille et des loisirs. Les techniques que sont le génie civil, l'architecture et l'urbanisme s'entrecroisent ainsi dans la construction du cadre de vie de l'homme victorien. Il faut, de plus, ajouter à ces techniques les télécommunications puisque l'invention du téléphone entraîne son entrée dans l'espace domestique.

Flichy, note que si le XIX^{e} siècle est « l'âge d'or de la vie privée » [17], c'est probablement parce que l'ère victorienne est l'une des premières époques où sont construites des maisons qui abritent principalement des membres de la même famille dont les activités productives et politiques se déroulent au dehors de la maison. Si on accepte la thèse de Flichy, la vie privée telle que nous l'entendons aujourd'hui, qui est considérée comme fondamentale dans les sociétés occidentales – elle est, par exemple, mentionnée dans la *Déclaration des droits de l'homme* – tire sa définition d'un certains « système technique ». Il faut ainsi accepter l'hypothèse que l'informatique puisse aider à la privatisation de l'espace [18].

B Formuler des fins propres dans un système technique

Le *Quantified Self* – littéralement le « soi mesuré » – est un mouvement qui prône la connaissance de soi sous forme numérique et l'amélioration de ses performances à partir de ces connaissances. Il se développe principalement au travers de sites Internet et de réunions organisées autour de la présentation d'outils de mesure et d'analyse des données [19]. Les participants du mouvement partagent tous un présupposé commun : nous sommes entourés d'objets qui enregistrent des données à notre encontre et qui, dans la plupart des cas sont utilisées par des firmes privées alors que ces données peuvent nous apprendre beaucoup sur nous. Les tenants du mouvement se proposent ainsi de se fixer des projets – comme l'amélioration de la capacité de travail ou l'amélioration de sa condition physique – et de les mener à bien en enregistrant son évolution et les effets des différentes techniques l'on emploie pour mener à bien son projet. Dans certains cas, plusieurs participants peuvent croiser leurs données.

Le *Quantified Self* met en jeu plusieurs éléments décriés par Ellul. Tout d'abord, il nécessite une surveillance de soi de tous les instants. Dans la mesure où la plupart des projets font l'objet d'une annonce publique sur les sites Internet dédiés au

17. L'expression est de Stéphanie Coontz, qui étudie la formation de la notion de vie privée au XIX^{e} siècle. Voir [Coontz 1988].

18. La détermination des effets de l'informatique fait l'objet du troisième chapitre du présent travail

19. Le site officiel du mouvement répertorie les ressources en ligne et hors ligne ainsi que les réunions organisées autour du mouvement : http ://quantifiedself.com/.

Quantified Self, et que les utilisateurs publient leurs résultats, les autres participants au mouvement sont aussi les acteurs de la surveillance. Les techniques mises en jeu « font système » : le *Quantified Self* coordonne ainsi l'électronique – pour récolter les données – l'informatique – pour les analyser et les visualiser – la diététique et les techniques d'entraînement physique par exemple. Enfin, le mouvement pousse à vivre dans un monde dans lequel les objets techniques occupent une place toujours plus importante, par exemple pour enregistrer le comportement des utilisateurs de manière automatique. Cependant, dans le cas du *Quantified Self*, l'individu n'est pas la cible d'une influence extérieure à laquelle il n'aurait pas consenti. Au contraire, il se prend lui-même comme objet de transformation et cherche, au moyen des outils techniques et sous le regard de ses pairs, à évaluer la réussite de son projet.

La pratique de transformation de soi par soi s'appuie ici sur deux éléments. Tout d'abord, le *Quantified Self* est un mouvement « social » dans la mesure où les participants partagent leurs projets et leurs données. Les projets se font ainsi sous l'œil des autres qui jouent le rôle d'adjuvants, de conseillers ou bien de témoins. Ensuite, le mouvement s'appuie sur une intense pratique de détournement, de modification et de création d'outils d'observation et de mesure de soi. De nombreuses applications pour téléphone mobile sont ainsi consacrées au *Quantified Self*. Des outils comme les caméras sont régulièrement détournés afin de s'observer. C'est ici, nous semble-t-il, que le *Quantified Self* constitue un exemple fécond de réponse d'un usage de techniques « mise en réseau » à des fins qui ne sont pas de l'ordre d'une incitation adressée à l'individu par un outillage technique qui le dominerait. En effet, en créant un mouvement et en transformant des outils, il est possible de remettre au premier plan les intentions des individus auxquelles s'appliquent les techniques. En l'espèce, la médiation des techniques et du regard d'autrui dans la construction de soi n'a pas disparu. Cependant, la technique ne joue plus principalement le rôle d'un intermédiaire presque autonome qui prend des décisions et influence les individus. Elle constitue un milieu au travers duquel les participants au *Quantified Self* s'explorent et se transforment.

L'enregistrement et la diffusion des données, qui sont aujourd'hui au centre de nombreuses polémiques, soit parce qu'elles portent atteinte à la vie privée, soit parce qu'elles exposent à un regard vécu comme une intrusion sont ainsi détournées par le *Quantified Self* au profit de celui qui les récolte et les analyse. Elles permettent aux individus de prendre conscience de leurs habitudes et de mesurer leurs réussites et leurs échecs. C'est en cela qu'elles « délivrent [...] aux individus les moyens de décider librement ce qui est bon pour eux » [20].

20. Voir [Kessous 2013][p. 50].

IV Conclusion

Nous avons ouvert cette section en notant la difficulté de définir l'informatique. Nous avons proposé, en analysant l'histoire de son développement, de concevoir l'informatique comme la discipline qui prend pour objet les procédures de calcul automatisées et qui, pour ce faire, élabore des modèles. Nous avons souligné son caractère « abstrait » par rapport à d'autres techniques. Elle s'applique en effet à une grande variété de domaines d'application. Informatisées et connectées, les différentes facettes de notre vie se retrouvent ainsi prises dans un « système informatique » composé de logiciels et d'ordinateurs qui sont, pour la plupart, portables. Ce système est à l'origine de nombreuses craintes qui portent notamment sur la surveillance dont nous faisons aujourd'hui l'objet et ses effets. Observés et catégorisés, n'avons-nous pas perdu le contrôle sur notre vie ? Nos décisions sont en effet prises au travers de la médiation de la technique et nos actions, enregistrées et analysées, donnent lieu à des incitations ciblées et automatisées. Systématique et presque autonome, la technique aurait ainsi acquis une place sans précédent dans l'organisation de notre vie.

Néanmoins, le recours aux ressources de l'histoire et de la sociologie, afin de remettre en perspective l'informatique avec d'autres techniques permet de montrer que, par essence, les techniques influencent la structure du monde dans lequel nous vivons. Techniques de communication ou de transport, urbanisme ou encore architecture, chaque technique fonctionne en réseau avec les autres et structure notre monde vécu. L'informatique n'est ainsi pas devenue, alors que les autres techniques ne l'étaient pas, un facteur déterminant de notre société. Comme le montre l'exemple du tramway, les techniques l'ont toujours été. L'informatique détermine cependant une nouvelle manière de se montrer, de voir les autres et le monde qui associe plusieurs modalités de présence, malgré la distance.

S'il est indéniable que certains usages informatiques semblent aliénants, l'exemple du *Quantifed Self* montre qu'il est possible, grâce à la manière spécifique qu'offre le monde numérique, de se prendre comme objet de transformation et de former des fins propres. La technique est alors utilisée dans le cadre d'un engagement de transformation de soi. Dès lors, alors que l'opposition, que dresse Ellul par exemple, entre le développement technique et le développement humain repose sur une opposition d'ordre ontologique entre la Technique et l'Humain, nous sommes conduits à nous tourner désormais vers l'humain comme un sujet historique dont les valeurs peuvent se déterminer au contact de la technique.

Chapitre 3

Individu, individuation et technique

L'informatique, aujourd'hui principalement utilisée en réseau, propose un mode de rapport aux autres et à soi caractérisé par une « numérisation » du monde – l'informatique est la science et la technique des calculs – et une communication à distance. Outil de surveillance et d'analyse qui peut soit soumettre à l'influence d'entreprises commerciales ou d'Etats, par exemple, ou bien qui permet de s'observer et de se transformer, l'informatique est ambivalente. Les sombres analyses d'un « système technique » conduisent ainsi à se demander à quel moment la technique devient inacceptable et, plus généralement, comment l'évaluer et justifier son usage. Nous proposons, afin de répondre à cette question, d'adopter le point de vue d'un sujet historique et socialisé et de nous demander comment la technique participe à sa constitution.

Notre analyse se déroulera selon trois niveaux. Nous commencerons par observer les rapports entre la subjectivation – la construction du sujet historique – et la technique. Nous identifierons ainsi un ensemble de techniques tournées vers le gouvernement des conduites et un ensemble dont le rôle est la médiation du rapport de soi à soi et son approfondissement. Ce faisant, nous substituerons à la définition de la technique, la distinction foucaldienne entre les « techniques de gouvernement » et les « techniques de soi » qui nous permettra de montrer la nécessité de la technique dans la subjectivation.

Toutes les formes de subjectivation ne sont cependant pas aussi désirables, les unes étant notamment plus contraintes que les autres. Nous nous situerons ainsi dans un second temps à un niveau éthique en nous demandant comme justifier l'utilisation des techniques. Nous définirons, du point de vue d'une individu constitué par la socialisation, l'éthique comme une forme de comportement. Ce faisant, nous prendrons, comme source de notre réflexion le thème du « souci de soi » en nous appuyant sur les analyses qu'en livre Michel Foucault dans ses derniers travaux. Nous montrerons notamment comment ce modèle permet de dépasser les apories des modèles utilitariste et déontologique, très présents dans la philosophie de la technique et particulièrement de l'informatique.

Cependant, l'éthique du souci de soi n'est pas un thème parfaitement formalisé par Foucault. Tel que conçu par l'auteur, il fait de plus l'objet de nombreuses critiques. Afin de montrer sa cohérence, nous proposerons enfin de rapprocher l'éthique du souci de soi de l'éthique de la vertu à partir des thèmes qui leurs sont communs. Nous définirons ainsi l'éthique du souci de soi comme une éthique de la vertu, terme

qu'elle définit comme un rapport critique aux normes. Ce rapport critique nous semble particulièrement adapté au cadre des nouvelles technologies.

I Agent technique et individuation

Les outils de communication contemporains sont régulièrement critiqués car ils nous soumettent à une surveillance potentiellement de plus en plus étendue : tout serait visible et ce à un nombre d'observateurs plus important. Les réseaux sociaux, par exemple, nous mettent face au regard des entreprises qui les conçoivent, de leurs partenaires commerciaux ainsi que des autres utilisateurs qui sont connectés. La visibilité des contenus que nous partageons, ou des traces enregistrées à notre insu, est ainsi de plus en plus difficile à contrôler. Cependant, cette perte de contrôle, qui expose les utilisateurs à des dangers potentiels bien connus a aussi pour résultante de constituer les outils numériques comme de nouveaux outils politiques. Les réseaux sociaux numériques ont ainsi été des outils importants dans ce qu'il est convenu d'appeler le « printemps arabe » , c'est-à-dire l'ensemble des révoltes et des mouvements de contestation d'Afrique du Nord et du Moyen Orient. Ils ont en effet permis d'organiser rapidement des évènements ponctuels en rassemblant un grand nombre de personnes ainsi que de communiquer sur ces évènements au delà des pays dans lesquels ils se déroulaient [1].

Cet exemple liminaire permet de souligner l'ambivalence des outils numériques. D'une part, les réseaux sociaux soulèvent de nombreuses craintes quant à la préservation de l'autonomie de leurs utilisateurs et de leur capacité à contrôler leur visibilité. D'autre part, cette même visibilité modifie la portée des discours et des actions politiques. Elle modifie les rapports entre les dirigeants et les citoyens d'un pays [2]. Dans les deux cas, les techniques sur lesquelles les réseaux sociaux reposent sont identiques et produisent des effets comparables : elles redistribuent les possibilités de rendre visible, de se montrer et de voir. Dans des contextes distincts, cependant, elles reçoivent des évaluations distinctes selon qu'elles encouragent ou entravent la liberté d'action.

Dans le cas de la dénonciation de l'extrême visibilité des utilisateurs de réseaux sociaux numériques, la médiation technique interroge la marge de manœuvre laissée aux utilisateurs des outils numériques. Observés et catégorisés, cibles de publicités personnalisées, de suggestions d'amis et d'incitations à partager toujours plus, ces derniers sont souvent en danger de voir leurs données largement diffusées ou utilisées [3]. Ainsi la navigation sur l'Internet ou les réseaux sociaux numériques met-elle en présence d'incitations à consommer, à se connecter à des utilisateurs ou à visiter telle ou telle page. Issues du « marketing des traces » , c'est-à-dire de l'analyse au-

1. Sur ce point, voir notamment [Angelis 2012], où l'auteur analyse le rôle des sites Internet dans les révoltes récentes au Moyen-Orient.

2. Dans le cas du « printemps arabe » , c'est principalement le rôle des sites Internet qui est en jeu. Dans les démocraties occidentales, certains sites Internet se présentent comme un contre-pouvoir, par exemple en surveillant l'activité des élus. Le site www.nosdeputes.fr se veut ainsi un « observatoire citoyen » de l'activité des parlementaires français.

3. Facebook, réseau social numérique le plus utilisé à l'heure actuelle, donne accès à chaque utilisateur à une page qui lui permet d'observer ce que les autres utilisateurs auxquels il est connecté partagent avec lui. Cette page est surmontée d'un champ de saisie intitulé « Qu'avez vous à l'esprit ? » . Chaque publication des autres utilisateurs peut être commentée ou partagée, ce à quoi le réseau social incite.

tomatisée de nos traces et de l'identification de notre comportement, ces incitations sont personnalisées et s'adaptent à l'évolution de nos habitudes. Ne sommes-nous pas face aux traits que dénonçait Ellul dans la Technique ? Comment poursuivre des fins qui nous sont propres dans un environnement fait pour surveiller, prévoir, influencer et s'adapter à nos actions ?

Le cas de l'utilisation des réseaux sociaux numériques dans les mouvements de contestation politique n'est-il pas un exemple de réponse à cette interrogation ? Les outils informatiques sont ici employés à des fins qui excèdent leur utilisation initialement prévue – la sociabilité dans les campus américains. Par la transformation d'un ensemble de rapports de visibilité – entre les dirigeants et les censeurs d'une part, et les dirigés d'autre part notamment – ils permettent de nouvelles possibilités d'action ainsi que la recherche d'une nouvelle forme de subjectivité politique et son affirmation. Dans ce cas, les réseaux sociaux sont des outils qui permettent de se constituer et de se représenter comme un opposant politique. Cette brève analyse a pour intérêt de montrer la nécessité de prendre en compte le contexte d'utilisation d'une technique et les stratégies auxquelles elle participe pour en évaluer la portée. Elle montre aussi que les deux évaluations de la technique, celle disant qu'elle permet de peser sur les actions des utilisateurs et celle disant qu'elle leur ouvre un nouveau champ d'action peuvent s'interpénétrer. Les contestataires, en effet, tombent sous le coup du gouvernement de leurs conduites par les réseaux sociaux et par les autres utilisateurs en créant un profil sur ces sites. En modifiant le rapport qui les lie à leurs concitoyens et à leurs dirigeants, ils entrent ainsi dans un nouveau rapport avec le fournisseur de services informatiques et avec ses autres utilisateurs.

Notre thèse est ici que, dans un cas, la technique constitue un outil de gouvernement de conduites d'une population d'utilisateurs, d'où sa critique ; dans l'autre cas, elle constitue un outil qui permet de se constituer comme un sujet moins contraint par des rapports de force auxquels elle permet d'échapper. Dans les deux cas, la structuration de la visibilité des conduites par la technique est au centre de son efficacité. L'exemple des outils numériques montre que, comme ils sont impliqués dans plusieurs rapports de visibilité, la technique peut à la fois permettre d'échapper à des rapports de force et qu'elle est une technique de gouvernement. De plus, la portée de la technique peut changer, puisqu'elle entraîne des réajustements dans les rapports de force existants [4]. Nous sommes ainsi face à une distinction mouvante entre, d'une part, des « techniques de soi » , qui permettent d'étendre le spectre des actions que l'on peut faire et des « techniques de gouvernement » qui nous exposent à l'influence de notre environnement. Cette distinction, d'origine foucaldienne, capture ainsi à la

4. L'exemple de *Google* en Chine est à ce titre parlant. L'Internet chinois est aujourd'hui fortement censuré. De nombreux sites occidentaux sont ainsi inaccessibles depuis le pays. *Google* possédait, il y a quelques années encore, un site Chinois qui permettait de contourner la censure. *Google* avait, dans un premier temps cédé aux injonctions du gouvernement chinois de censurer les recherches contenant certains termes, comme « Tienanmen » . En 2010, la firme à décidé de se retirer du marché chinois suite à une attaque du gouvernement chinois sur ses serveurs. Les Internautes ne peuvent ainsi plus effectuer de recherches Internet qu'au travers de sites chinois, fortement censurés. Voir, sur ce point, [Schmidt 2013][pp.81-120], Eric Schmidt, PDG de *Google* y évoque la censure de l'Internet chinois.

fois les deux faces possibles des techniques et l'importance des techniques quant à la subjectivation.

Le recours à la distinction entre deux types de techniques nécessite d'articuler plusieurs périodes du travail de Foucault. Les « techniques de gouvernement » , par exemple, font l'objet des analyses de *Surveiller et punir* – dans lesquelles Foucault étudie des exemples passés de prisons – ou de *Sécurité, territoire et population*, où l'auteur considère les effets de la mise en œuvre de l'économie libérale [5]. Les « techniques de soi » sont, elles, abordées dans la dernière période du travail de Foucault. Elles font notamment l'objet de *L'Histoire de la sexualité*, où l'auteur s'engage sur le terrain de l'éthique [6]. À première vue, cependant, les préoccupations philosophiques de l'auteur portent plutôt sur la vérité et les relations de pouvoir que sur la technique. De plus, si la pérennité de Foucault en matière de pensée de la surveillance ne fait aucun doute, sa place dans la philosophie technique reste relativement peu établie [7]. Foucault, qui ne se réclame pas toujours comme un philosophe, ne se proclame évidemment pas philosophe de la technique [8]. Cependant, il nous semble que la dichotomie entre techniques de soi et techniques de gouvernement permet de répondre à la difficulté de poursuivre des fins propres dans un environnement technique : leur rôle dans les rapports de pouvoir, les techniques orientent nos possibilités d'action. En influençant ce que nous pouvons faire, elles participent à la constitution de la subjectivité. Dès lors, le rapport que le sujet noue aux objets techniques n'est pas uniquement de délégation d'une fonction à un outil dans un but d'optimisation ou d'automatisation. Ce rapport est aussi constitutif. Les techniques sont génératrices d'effets de savoir et de subjectivation inédits. Certaines connaissances, par exemple, sont impossibles à envisager hors de l'emploi de techniques de calcul informatisées [9]. Contre une opposition entre le développement technique et le développement humain, la conception foucaldienne de la technique permet de considérer que les sujets sont des « hybrides » puisque leur corps et leurs attitudes sont constitués par le rapport entre des éléments biologiques et techniques. Cette conception nous semble être un outil conceptuel de premier plan à une époque où l'assemblage entre les corps et les techniques numériques prend la forme d'une symbiose [10].

5. Voir [Foucault 1993] et [Foucault 2004].

6. [Foucault 1984a].

7. Le thème de la surveillance et l'étude du panoptique ont été la source d'analyse des phénomènes de surveillance. Néanmoins, l'éthique du souci de soi, en tant que telle, a donné lieu à peu de développement dans l'analyse des techniques numériques, à l'exception, par exemple, des travaux de Leslie P. Willcocks. Voir [Willcocks 2006].

8. Seules quelques analyses, comme cellse de Jim Gery, se consacrent au rôle de Foucault dans la pensée de la technique. Voir [Gerrie 2007] où Jim Gerrie soutient que Foucault est un philosophe de la technique même si l'auteur ne se réclame pas expressément comme tel.

9. Dans « What is computer ethics ? » , Moor cite ainsi l'exemple du théorème des quatre couleurs. Ce théorème établit qu'il est possible de colorier une carte divisée en régions en utilisant seulement quatre couleurs et en garantissant que deux régions limitrophes ont des conleurs différentes. Conjecturé en 1852 par Francis Guthrie, ce théorème ne fut démontré qu'en 1976 par Kaneth Appel et Wolfgang Haken grâce à l'usage d'ordinateurs pour réaliser des calculs qui ne pouvaient être faits par des humains. Voir [Moor 1985].

10. Deleuze, alors qu'il étudie l'œuvre de Foucault, souligne l'intérêt de la pensée de ce dernier lorsqu'on tente de saisir le monde contemporain et notamment l'essor de l'informatique : « Ne

Allant à l'encontre d'une caractérisation *a priori* du rôle des techniques comme outil de domination ou de libération, Foucault appelle en effet à l'examen de leurs usages spécifiques qui, seule, permet de déterminer la nature du rapport du sujet à une technique. À une étude sur l'ontologie de la Technique et de l'Humain, il oppose ainsi une étude tournée vers des situations particulières. Si les techniques nous mettent toujours face à un rapport de pouvoir, ce rapport n'est cependant pas nécessairement de domination et permet de se donner des buts – comme l'acquisition d'un type de comportement – et les moyens d'y parvenir. En d'autres termes, nous développons des attitudes non « contre » la technique mais à son contact et il en est de même de nos fins.

D'une part, lorsque Foucault étudie la technique – par exemple dans *Surveiller et punir*, où il analyse les différents moyens qui permettent de mettre en œuvre une orthopédie sociale – elle semble principalement être perçue comme outil de coercition. L'analyse du « panoptique » est probablement l'exemple phare de l'analyse de la domination par Foucault [11]. Imaginé par Bentham, le « panoptique » est une prison annulaire qui vise à assurer l'efficacité du dispositif carcéral en organisant les rapports de force entre les prisonniers et leurs gardiens [12]. Les prisonniers sont enfermés dans des cellules individuelles disposées dans un bâtiment en forme d'anneau. Comme ces cellules sont traversées par la lumière, les prisonniers peuvent être observés en permanence. Les gardiens se trouvent, eux dans une tour centrale faite de telle sorte que les prisonniers ne peuvent jamais savoir s'ils sont observés. Le prisonnier du panoptique est ainsi la source d'un ensemble de savoirs puisque ses faits et gestes sont observés. Ces savoirs sont mis en jeu dans des stratégies qui visent à rééduquer les prisonniers.

Applicable aux prisons, mais aussi aux écoles ou aux ateliers, le panoptique est ainsi l'architecture idéale d'une machine optique qui permet une surveillance complète et asymétrique d'un groupe d'individus qui doivent être éduqués et qui peuvent être observés un par un. C'est à ce titre que l'image du panoptique a été reprise dans plusieurs analyses qui portent sur l'utilisation et les effets des outils numériques [13]. Ne sommes-nous pas tous, à la manière des prisonniers du panoptique, rendus visibles et exposés à l'influence des « gardiens » qui prennent la forme d'entités qui enregistrent et traitent nos traces ?

Néanmoins, la thèse de Foucault sur le rapport entre les prisonniers et les gardiens dans le panoptique est surprenante. Les prisonniers ne sont-ils pas, de manière

dit-on pas couramment que les forces de l'homme sont déjà entrées en rapport avec d'autres forces, celles de l'information, qui composent avec elles autre chose que l'homme, des systèmes indivisibles « homme machine » , avec les machines de troisième espèce ? Une union avec le silicium plutôt qu'avec le carbone ? » . Voir [Deleuze 1986][p.95]

11. Voir [Foucault 1993][pp. 228-264].

12. Voir [Bentham 1977].

13. Le « post-scriptum sur les sociétés de contrôle » de Deleuze prend ainsi pour point de départ de l'analyse de l'essor de l'informatique la figure du panoptique afin d'en évaluer l'actualité. Voir [Deleuze 2003]. Dans *Voir et pouvoir, qui nous surveille ?* Jean Gabriel Ganascia reprend lui aussi l'image du panoptique pour caractériser le rapport qui lie les utilisateurs des services numériques entre eux et avec les entreprises qui les mettent sur le marché. Voir [Ganascia 2009].

évidente, aliénés par le rapport que le bâtiment panoptique instaure entre le gardien et eux ? Foucault lui-même souligne que les prisonniers sont « disciplinés » dans la mesure où leurs mouvements et leurs actions sont contraints. Cette discipline semble interdire toute action volontaire de la part du prisonnier. Leur régime alimentaire, leurs exercices physiques, les moindres détails de leur vie, tout est potentiellement observé et donc contrôlé par les gardiens. Pourtant, sous la plume de Foucault, la discipline est paradoxalement créatrice. Par la pratique d'exercices, la discipline s'imprime dans le prisonnier qui, petit à petit, l'intériorise. Dès lors, la discipline participe, par la modification des comportements du prisonnier, à construire sa subjectivité que Foucault comprend comme un phénomène matériel – la discipline prend appui sur le corps des prisonniers – et historique.

Le passage d'une conception coercitive à une conception productive du rapport de force entre les gardiens et les prisonniers vaut aussi pour les techniques qui participent aux rapports de pouvoir. L'observation, la prescription d'exercices et l'évaluation de la réussite des prisonniers sont, en tant que moyens de contrainte, des outils de modification et de création de comportements. Ainsi les prisonniers, qui ne savent jamais s'ils sont observés, intériorisent le regard potentiel des gardiens et se comportent, en permanence, comme s'ils étaient épiés. Poussé à l'extrême, le panoptique pourrait ainsi fonctionner sans gardiens. Leur présence potentielle suffit à garantir l'efficacité du dispositif. Ce mouvement d'intériorisation du regard s'ajoute à l'intériorisation du comportement attendu par les gardiens au travers des exercices et participe au caractère créateur de la discipline carcérale. Les prisonniers sont « sujets » aux deux sens du terme. Ils sont, bien sûr, assujettis à la machine qu'est le panoptique et aux gardiens. En ce sens « la visibilité est un piège » [14]. Néanmoins, ils sont aussi sujets au sens d'agents puisqu'ils reprennent à leur compte les comportements disciplinaires. Le « sujet discipliné » doit ainsi à la fois être appréhendé comme l'objet de l'observation et des expérimentations menées par les gardiens et comme l'agent de ses comportements. C'est un sujet dont la constitution s'opère au travers de l'histoire de ses rapports au gardien et à l'outillage du panoptique.

Dans la mesure où Foucault étend le modèle panoptique à l'ensemble de la société, on lui a parfois reproché de faire du « pouvoir » - qu'il définit comme un ensemble de procédures qui instaurent des rapports de force – une personne abstraite et d'interdire de sortir des rapports de domination [15]. Le rappel du double aspect du sujet discipliné – à la fois objet et sujet – permet de débouter cette critique : même au contact d'un pouvoir coercitif, le sujet reste agent et peut former des fins propres, comme l'évasion [16].

14. Foucault compare les prisons modernes aux cachots sombres et humides et écrit : « La pleine lumière et le regard d'un surveillant captent mieux que l'ombre, qui finalement protégeait. La visibilité est un piège. » Voir [Foucault 1993][p. 234].

15. Dans *Sécurité, territoire et population*, Foucault définit le pouvoir de la manière suivante « le pouvoir, c'est un ensemble de mécanismes et de procédures qui ont pour rôle ou fonction et thème, même s'ils n'y parviennent pas, d'assurer justement le pouvoir » . Voir [Foucault 2004][p. 4].

16. Dans une interview réalisée par Frank Burbage, Judith Butler interprète l'analyse du panoptique par Foucault dans ce sens. Pour elle, le rappel de la constitution historique du prisonnier permet de rappeler qu'il n'est prisonnier que de manière contingente. Voir [Burbage 2004].

Le gardien, comme dépositaire d'un rôle social, ne possède pas un pouvoir substantiel qu'il appliquerait sur les prisonniers. Au contraire, c'est parce que le gardien se trouve dans la tour et le prisonnier dans sa cellule qu'il peut exister, entre eux, une relation de pouvoir. Par conséquent, le pouvoir n'appartient pas au gardien mais émerge à partir de sa position vis-à-vis des prisonniers. Ce rapport de force est médiatisé par les outils et les techniques mises en œuvre pour construire et entretenir le panoptique. Par suite, le gardien, en tant que sujet au sens de rôle social, est aussi une « création » du rapport de pouvoir qu'est la discipline. La relation de pouvoir qui lie le gardien au prisonnier les crée tous les deux comme sujets.

Il peut sembler étouffant de faire des individus, même ceux qui semblent plus libres que les autres, des créations des rapports de pouvoir. La sensation d'étouffement est d'ailleurs augmentée par la possibilité d'adapter le panoptique à d'autres situations qui nécessitent la mise en place d'un rapport de visibilité asymétrique de la part d'un groupe d'observateurs restreint vers un large groupe d'observés qui doivent être saisis individuellement. Foucault étend ainsi ses remarques sur l'assujettissement du gardien et du prisonnier à des rapports de pouvoir à l'ensemble de la société du XVIII^{e} siècle [17]. Les rapports asymétriques de visibilité et d'injonctions peuvent en effet être retrouvés non seulement dans les prisons, mais aussi dans les ateliers, les hôpitaux ou les écoles. Les contre-maîtres, les médecins et les enseignants jouent tous, vis-à-vis des ouvriers, des malades et des élèves, des rôles comparables à celui du gardien vis-à-vis des prisonniers. Ils sont tous les observateurs, les correcteurs et les dirigeants d'un groupe d'individus fortement individualisés et observés. Dans la perspective foucaldienne, la « discipline » n'est pas un trait propre au panoptique mais est caractéristique d'une forme de rapports de pouvoir où un groupe détient, sur un autre groupe, une possibilité d'observation et d'éducation.

L'analyse du rapport entre gardien et prisonniers, qui peut être étendue au rapport entre maître et écoliers ou contremaître et ouvriers montre que le corps social est pénétré par un ensemble de rapports de force. L'analyse par Foucault des rapports de force qu'instaurent l'essor de la « police » et de la raison d'État ou bien encore la médicalisation de la folie montrent que cette conclusion vaut pour toutes les époques : nous sommes, en tant qu'êtres sociaux, pris dans des rapports de force et de visibilité [18]. Le sujet n'est-il pas, dès lors, toujours plus objet qu'agent ? Est-il illusoire d'imaginer sinon se libérer, au moins diminuer les effets des rapports de pouvoir qui sont omniprésents ?

Répondre ici par l'affirmative serait ignorer que Foucault ne nie ni l'existence de rapports de domination ni la possibilité de les modifier ni même l'existence de rapports de force qui ne sont pas aliénants. Les ouvrages de Foucault, comme *Surveiller et punir*, peuvent être conçus comme des actions en faveur d'une transformation des rapports de pouvoir. Ainsi l'œuvre est elle liée à l'engagement de l'auteur au sen du Groupe d'Intervention des Prisons (GIP). L'analyse des procédures carcérales

17. Voir [Foucault 1993][pp. 200-224].

18. L'analyse de l'essor de la police et de la notion de « raison d'état » fait l'objet de *Sécurité, territoire et population*. Celle de la « science de la sexualité » de *L'Histoire de la folie*. Voir [Foucault 2004][p. 261 et sq.] et [Foucault 1976a].

a ainsi pour but à la fois d'en révéler la portée et de permettre d'en dresser une critique préparatrice à leur modification [19]. En outre, l'œuvre de Foucault permet de distinguer ces usages dominateurs et aliénants du pouvoir d'un ensemble de relations qui permettent de se créer comme un individu dont la subjectivité n'est pas toute entière contrainte.

Les thèmes de la visibilité et de la dissymétrie des relations sont toujours centraux dans les derniers travaux de Foucault, qui portent sur l'éthique. Cependant, ces thèmes sont abordés d'une manière radicalement différente de celle de *Surveiller et Punir*. Le disciple stoïcien, par exemple, livre à son maître la totalité des détails de sa vie [20]. Ce dernier lui prescrit en retour des exercices et des conseils et participe, en cela, à la progression de son élève vers la sagesse. L'asymétrie de la relation entre le disciple et le maître ne conduit dès lors pas à l'aliénation du premier au second. Au contraire, c'est à ce prix que le disciple peut se transformer pour incarner le style de vie du sage stoïcien qu'il se donne à lui-même comme but. Dans cette perspective, la relation de pouvoir soutient la transformation de l'élève. Elle permet à l'élève d'accroître la connaissance de lui-même et de ses limites qu'il possède afin d'augmenter son contrôle sur ses comportements. Elle s'adapte aussi à à l'évolution de l'élève puisque sa progression efface peu à peu l'asymétrie des positions entre son maître et lui.

Le panoptique mettait en scène un rapport de force qui augmentait le contrôle d'un sujet sur un autre. L'écriture stoïcienne présente, elle, un rapport de force entre soi et soi-même où le regard d'autrui joue le rôle d'intermédiaire et d'adjuvant. Le prisonnier et le disciple stoïcien sont ainsi tous les deux des sujets construits par l'histoire de l'utilisation d'un ensemble de techniques et de rapports aux autres mais qui se soldent par des effets distincts. D'une part, le corps et le sujet sont objets de connaissance au sein de relations de pouvoir ; le prisonnier est principalement réduit à cette dimension. D'autre part, les techniques et les exercices prennent appui sur le corps afin d'inscrire un mode d'être au sein du sujet et de conduire à « l'entraînement de soi par soi » [21]. C'est dans cette perspective que le disciple stoïcien est « sujet ».

En délaissant la définition d'un sujet transcendental doué, par exemple, d'une liberté originelle, Foucault est ainsi conduit à redéfinir le terme « éthique » non pas comme l'évaluation d'un rapport à une norme *a priori* mais comme une forme de comportement issue de la socialisation. Cette définition prend origine dans la notion de « souci de soi » - le thème du soin que le sujet porte à son âme et à son corps afin de cultiver un style de vie propre – que Foucault lit dans les textes antiques. Foucault établit l'existence du souci de soi dès Platon et note son importance dans

19. Ainsi Foucault affirme-t-il, dans une lettre de 1980 : « Voyez-vous, j'ai entrepris et achevé, après l'expérience du GIP, mon livre sur les prisons. Et ce qui me chagrine, ce n'est pas que vous ayez l'idée bizarre de déduire de mon livre [...] ma vénéneuse influence sur le GIP ; c'est que vous n'ayez pas eu la toute simple idée que ce livre doit beaucoup au GIP et que s'il contenait deux ou trois idées justes, c'est là qu'il les aurait prises. » . Voir [Foucault 1994][p. 97].

20. Voir, par exemple, [Foucault 2001][pp. 1234-1250].

21. Cette expression traduit le Grec *technê tou biou*. Elle désigne la pratique de l'écriture commune à de nombreuses écoles antiques comme les Pythagoriciens ou les Stoïciens. Voir [Foucault 2001][pp. 1236].

des écoles plus tardives, comme le Stoïcisme [22] . Si Foucault note que le souci de soi prend place dans un espace de conventions et de règles, il note que l'éthique du souci de soi constitue moins une éthique tournée vers l'évaluation de la conformité du sujet à une règle qui lui serait extérieure que la tentative de s'approprier une forme de comportement à partir de l'examen de soi. Alors qu'il se soucie de lui-même, le sujet est ainsi la source de sa propre subjectivation puisqu'il se donne pour but un style de subjectivité qu'il souhaite atteindre. Alors que le mode de subjectivité du prisonnier du panoptique était décidé pour lui, on peut ici parler d'une forme plus « intense » de sujet puisqu'elle provient de l'activité même du sujet et lui permet de s'approprier un ensemble de valeurs. Le rapport de pouvoir mis en jeu est ici de soi sur soi. Tout comme l'exemple de la discipline, il repose sur l'usage de techniques appropriées comme intermédiaires de ce rapport.

L'écriture est probablement l'exemple le plus important des techniques de soi sous la plume de Foucault. Elle permet de se représenter à soi et aux autres et ainsi d'évaluer sa progression vers un idéal de comportement. L'écriture permet de plus d'assurer que l'on cherche en permanence à se transformer. Elle permet d'atteindre quelqu'un malgré la distance, et de le rendre présent en rappelant son souvenir [23]. Ces deux propriétés font de la correspondance stoïcienne une « technique de soi » propre à permettre au disciple stoïcien de se transformer – soit en répertoriant un ensemble de comportements, soit en se présentant à l'autre et en requérant son avis. En faisant retour sur soi, le disciple peut évaluer la distance de ces comportements vis-à-vis de l'idéal du sage et identifier les points qu'il lui reste à modifier et ceux, au contraire qu'il doit conserver. Dans le cas des Stoïciens, la technique offre ainsi des occasions de se forger un caractère, c'est-à-dire de construire un style de comportement que le sujet lui-même et les autres peuvent reconnaître. L'écriture, associée aux exercices physiques et au régime est ainsi « éthopoiétique » dans la mesure où elle permet au disciple de transformer les principes de vie stoïciens en un *êthos* qui s'installera en lui et modifiera sa personnalité [24]. La technique de la correspondance et le rapport de visibilité qu'elle met en jeu – le disciple se rend complètement transparent à son maître et à lui-même – ne sont ainsi pas des obstacles mais les adjuvants du développement d'une subjectivité « propre » dans la mesure où elle est choisie par le disciple.

Néanmoins, l'écriture est l'exemple même d'une technique dont la place vis-à-vis du sujet et des rapports qu'il entretient à lui-même est ambivalente. Dans *Surveiller et punir*, l'écriture est ainsi abordée à partir de l'apprentissage des gestes qui permettent d'écrire. L'apprentissage de l'écriture constitue une forme de discipline dans

22. Voir ´L'éthique du souci de soi comme pratique de la liberté (entretien avec H. Becker, R. Fornet-Betancourt, A. Gomez-M¸ller, 20 janvier 1984), Concordia. Revista internacional de filosofia, no 6, juillet décembre 1984. [Foucault 1994]pp. 99-116.

23. « ce que les autres sont à l'ascète dans une communauté, le carnet de notes le sera au solitaire » . L'écriture est, ainsi une technique qui permet de juguler l'absence, soit parce qu'elle permet d'atteindre quelqu'un malgré la distance, soit de le rendre présent en rappelant son souvenir. Voir [Foucault 2001][pp. 1235].

24. Par « éthopoiétique » , Foucault désigne la capacité qu'à l'écriture pour les pensées antiques à transformer l'éthique en *êthos*. Voir [Foucault 2001][pp. 1237].

la mesure où elle repose sur la structuration du corps de l'écolier en lui enseignant un ensemble de postures et de gestes propices à une écriture de qualité. Les techniques de soi stoïciennes s'appuient sur la capacité du disciple à faire retour sur lui même et à se prendre comme un « projet » . Il en est de même pour les techniques de gouvernement appliquées aux écoliers dont l'efficacité repose sur leur capacité de faire retour sur eux-mêmes pour se prendre comme objets de transformation à partir de conscience qu'ils ont du regard du maître et de la punition qu'il peut leur infliger. Le rapport d'observation et de correction du corps et des comportements qui lie le maître et l'écolier s'appuie ainsi sur la réflexivité de l'écolier pour parvenir à son assujettissement.

L'analyse de la correspondance permet de compléter la définition de la technique dans son rapport à la visibilité. La technique est une structure optique. Elle permet de montrer et de se montrer. Elle permet aussi de s'observer et d'observer. Ce faisant, elle est aussi une structure de subjectivation. En nous rendant visible, elle nous permet de nous exposer aux rapports de pouvoir que nous entretenons avec les autres et avec nous-mêmes. La différence entre les techniques qui contraignent les formes de subjectivité des techniques de soi provient de l'hétérogénéité des stratégies auxquelles elles participent et des buts que ces dernières visent. Par conséquent, les rapports entre sujet et techniques ne sont ontologiquement ni de domination ni de libération même s'ils sont toujours stratégiques [25]. Foucault prend en effet le terme de « technique » , à partir de sa racine grecque, dans son ambivalence. Le terme désigne tout à la fois l'appareillage matériel et l'acte de gouverner [26]. La technique participe ainsi à la mise en œuvre de la « gouvernementalité » , c'est-à-dire un type de pouvoir qui prend la population pour objet au travers d'instruments et de procédures spécifiques [27]. À la notion d'un « système technicien »aliénant, nous proposons ainsi de substituer la notion foucaldienne de « dispositif » , c'est-à-dire un ensemble de techniques, de rapports de force et de rapports de connaissances qui participent à la constitution d'un sujet et qui, pour ce faire, sont interdépendantes puisqu'elles fonctionnent en réseau. Alors que le « système technicien » est extérieur à la société et nocif, le dispositif est nécessaire à la subjectivation et regroupe aussi bien des techniques de gouvernement que des techniques de soi [28]. Il est stratégique

25. Foucault n'est bien sûr pas le seul à envisager l'importance de la technique dans la création du sujet. Heidegger, dont Foucault s'inspire, souligne déjà l'importance de la technique dans nos possibilités de « futurisation » . Voir *Questions on technology,* [Heidegger 1958].

26. Foucault se réfère explicitement au terme grec de *tekne* pour désigner le double sens de la technique, à la fois artefact et méthode de gouvernement. C'est à partir de cette définition de la technique, que Foucault peut affirmer, notamment, que l'architecture n'est pas purement technique mais, aussi, politique. Voir « Espace, savoir et pouvoir » : [Foucault 2001][pp. 1089-1105].

27. Foucault définit la gouvernementalité comme « l'ensemble constitué par les institutions, les procédures, analyses et réflexions, les calculs et les tactiques qui permettent d'exercer cette forme bien spécifique, quoi que très complexe, de pouvoir qui a pour cible principale la population, pour forme majeure de savoir l'économie politique, pour instrument techniques essentiels les dispositifs de sécurité » . Voir [Foucault 2004][p. 111].

28. Foucault définit la notion de « dispositif » de la manière suivante : ´ Ce que j'essaie de repérer sous ce nom, c'est, premièrement, un ensemble résolument hétérogène, comportant des discours, des institutions, des aménagements architecturaux, des décisions réglementaires, de lois, des mesures administratives, des énoncés scientifiques, des propositions philosophiques, morales,

et peut toujours être contré.

Qualifier les techniques de « techniques de gouvernement » ou de « techniques de soi » selon les rapports de pouvoir auxquels elles participent permet de se tenir à mi-chemin de deux positions antagonistes. La distinction entre techniques de gouvernement et techniques de soi, qui coexistent dans la gouvernementalité que Foucault définit comme « la rencontre entre les techniques de domination exercées sur les autres et les techniques de soi » montre la neutralité ontologique de la technique qui peut favoriser aussi bien la domination que la création de soi par soi [29]. Il ne s'agit pas de célébrer aveuglément le progrès technique. Les sombres analyses de *Surveiller et punir* interdisent tout angélisme quant au potentiel libérateur de la technique puisqu'elles mettent en scène un ensemble d'usages techniques qui visent le dressage des individus auxquels elles s'appliquent. Il ne s'agit pas non plus d'adopter une position fataliste qui opposerait le développement technique et le développement humain. La technique ne « libère » pas dans la mesure où elle ne nous permet pas de nous abstraire de tout rapport de pouvoir – le disciple stoïcien est ainsi à la fois son propre objet de pouvoir et celui de son maître. Pour autant l'usage technique n'est pas nécessairement aliénant. Au contraire, même si les analyses de *Surveiller et punir* montrent que la technique peut permettre la domination, les techniques antiques donnent l'exemple de techniques permettant le développement de formes de subjectivité moins contraintes.

La distinction entre plusieurs types de techniques selon leur relation à la subjectivité a une place double dans la pensée foucaldienne. Tout d'abord, elle prend place dans l'étude des rapports de pouvoir passés et des rapports de pouvoir dans lesquels nous vivons. L'identification de techniques de soi et de techniques ambivalentes comme l'écriture permet d'envisager le passage d'une forme de subjectivité contrainte vers une forme plus intense en transformant les rapports stratégiques dans lesquels elle est mise en jeu. Se jouer des gardiens en adoptant des comportements volontairement provocateurs – ou au contraire ne rien faire – introduire dans un espace où tout est visible des outils, comme les téléphones portables ou les ordinateurs, qui permettent de communiquer sans être vu et avec des personnes

philanthropiques, bref : du dit, aussi bien que du non-dit, voilà les éléments du dispositif. Le dispositif lui-même, c'est le réseau qu'on peut établir entre ces éléments. Deuxièmement, ce que je voudrais repérer dans le dispositif, c'est justement la nature du lien qui peut exister entre ces éléments hétérogènes. Ainsi, tel discours peut apparaître tantôt comme programme d'une institution, tantôt au contraire comme un élément qui permet de justifier et de masquer une pratique qui, elle, reste muette, ou fonctionner comme réinterprétation seconde de cette pratique, lui donner accès à un champ nouveau de rationalité. Bref, entre ces éléments, discursifs ou non, il y a comme un jeu, des changements de position, des modifications de fonctions, qui peuvent, eux aussi, être très différents. Troisièmement, par dispositif, j'entends une sorte – disons – de formation, qui à un moment historique donné, a eu pour fonction majeure de répondre à une urgence. Le dispositif a donc une fonction stratégique dominante. Cela a pu être, par exemple, la résorption d'une masse de population flottante qu'une société à économie de type essentiellement mercantiliste trouvait encombrante : il y a eu là un impératif stratégique, jouant comme matrice d'un dispositif, qui est devenu peu à peu le dispositif de contrôle-assujetissement de la folie, de la maladie mentale, de la névrose. Voir « Le jeu de Michel Foucault » , [Foucault 2001][p. 299].

29. Voir « Les techniques de soi » : Voir [Foucault 2001][pp. 1604].

extérieures à une situation particulière sont autant de moyens de détourner les effets d'une technique et d'influencer les rapports de forces qu'elle soutient. La comparaison entre techniques de soi et techniques de gouvernement permet ainsi d'identifier les points sur lesquels agir pour rendre des formes de subjectivité plus intenses.

Cependant, la pertinence des analyses foucaldiennes fait l'objet de critiques, notamment de la part des historiens. L'étendue des sources et la méthode foucaldienne d'interprétation sont les deux points auxquels s'attache cet angle de critique[30]. Néanmoins, ce n'est peut-être pas sur le plan historique qu'il faut situer l'analyse foucaldienne des techniques. La notion de « technique de soi » est ainsi aujourd'hui utilisée en anthropologie des techniques afin d'analyser la place des objets techniques – leur histoire, leurs usages et les formes qu'ils prennent – dans les cultures[31]. Foucault ne se livre pas à un travail d'anthropologue puisque son travail ne repose pas sur des recherches de terrain quant à la relation entre les hommes et les techniques. Ces études montrent néanmoins la force opératoire de la notion « technique de soi » qui permet d'étudier comment les gestes de la vie quotidienne et de la vie productive participent à la création des sujets, c'est-à-dire de leurs habitudes et de leur place sociale.

L'utilisation des travaux de Foucault dans le champ de l'anthropologie rappelle que l'auteur souhaite envisager, au travers de la notion de « technique de soi » un rapport à soi et aux autres qui prend appui sur les techniques sans pour autant être un rapport de connaissance ou un rapport de domination. La distinction entre les techniques de gouvernement et les techniques de soi vaut ainsi dans le cadre d'un projet éthico-politique qui vise à analyser les structures de nos rapports afin d'en modifier les effets. Foucault fournit un exemple lorsqu'il analyse la « gouvernementalité libérale » dans *Sécurité, territoire et population*. Cette analyse est polémique puisque l'auteur établit le caractère subversif de l'utilisation de la notion de « liberté » dans le cadre du libéralisme économique. Le « laissez-faire » , qui pour Foucault est caractéristique du libéralisme, est en effet un moyen d'intégrer les individus à un jeu d'observations et de techniques qui participent à leur gouvernement plus que la recherche de leur libération. La « liberté » comme bien, par exemple, tire ainsi sa valeur d'un ensemble de connaissances – issues notamment de la statistique – et de pratiques qui incitent les individus non seulement à agir ou parler mais aussi à se prendre en charge. Dès lors, la valorisation de la liberté prend sens dans un système politique qui accentue les vertus de l'autodiscipline[32].

Le statut de la notion de « liberté » est d'ailleurs problématique chez Foucault qui

30. Ainsi l'enfermement des fous que Foucault décrit dès le XVIIe siècle dans *Folie et déraison* ne se produit, pour Pierre Morel et Claude Quétel qu'au XIXe siècle. Voir [Morel 1996]. Richard Hamilton, lui, s'attaque à l'analyse du panoptique dans *Surveiller et punir*. Il souligne que l'attention que Foucault lui porte est disproportionnée dans la mesure où le panoptique n'a jamais été vraiment construit. Voir [Hamilton 1996].

31. Dans « Les technologies du sujet » , Jean-Pierre Warnier utilise la notion de « technique » de soi afin d'étudier comment les objets techniques participent à la création d'une culture et à la mise en place d'un système de consommation. Son étude est notamment appliquée au cas de plusieurs villages africains. Voir [Warnier 2010].

32. Voir Voir [Foucault 2004][p. 349 et sq.].

n'envisage jamais une libération complète mais le passage d'un rapport de pouvoir à un autre : la « libération » n'est toujours que momentanée puisque nous sommes toujours, pour être sujets, pris dans des relations de pouvoir. Il faut ainsi distinguer deux niveaux d'analyse de la liberté chez Foucault. Tout d'abord, la liberté est de l'ordre du fait. Les rapports de pouvoir sont toujours mobiles et susceptible d'être renversés, d'où une possibilité permanente de résistance [33]. Pour autant, la libération n'est jamais totale puisque nous ne pouvons nous extraire de tous les rapports de pouvoir. Par conséquent, la libération est, chez Foucault, « transformation » des rapports de pouvoir et, par conséquent, des formes de subjectivité que l'on peut adopter. La liberté est ainsi une pratique qu'il faut perpétuellement mettre en œuvre. Elle fait l'objet de l'analyse foucaldienne de l'éthique de soi.

33. Foucault écrit ainsi "là où il y a pouvoir, il y a résistance et que pourtant, ou plutôt par là même, celle-ci n'est jamais en position d'extériorité par rapport au pouvoir." Voir [Foucault 1976a][pp.125—126].

II Se soucier de soi pour se constituer comme sujet

En définissant la technique à l'aide des analyses de Foucault, nous avons délaissé la conception d'une opposition d'ordre ontologique entre l'Humain et la Technique. Nous avons adopté la perspective d'un sujet qui se conduit au travers de l'histoire de ses usages techniques. L'ambition de ce chapitre est de s'interroger sur le rapport à soi que permettent les « techniques de soi » . Il s'agit ici de mettre en question la possibilité de se constituer comme sujet au contact des techniques et ce de manière volontaire. Si l'éthique est la pratique qui vise à acquérir un mode de comportement déterminé et si les techniques, comme techniques de soi et comme techniques de gouvernement participent à la constitution des comportements, il s'agit de se demander comment les techniques permettent de se « soucier de soi » . En d'autre termes, comment les techniques permettent-elle de porter attention tout à la fois à ses biens – comme la santé ou les biens matériels – et de s'approprier un style de vie [34] ?

N'est-il cependant pas surprenant de choisir le thème du souci de soi comme source d'une réflexion éthique dans le domaine des techniques ? Considérer Foucault comme un philosophe de la technique et un éthicien constitue une interprétation de notre part qui prend pour parti d'étudier l'éthique telle qu'elle est abordée chez Foucault non pas uniquement à partir des travaux qui en traitent directement – les derniers textes de Foucault – mais à partir de l'ensemble de son œuvre. Cette démarche s'appuie sur l'interprétation que donne Foucault lui-même de son travail. Alors qu'on lui demande si son intérêt pour l'éthique dans l'*Histoire de la sexualité* n'est pas un changement de problématique, Foucault répond qu'il ne faut pas lire dans son œuvre une rupture avec ses objets d'études précédents que sont les relations de pouvoir ou les formes de savoir [35]. Au contraire, c'est, pour lui, une nouvelle manière d'aborder les rapports entre sujet et vérité, non plus, comme il l'a déjà fait, sous l'angle du sujet parlant ou contraint mais sous l'angle du rapport du sujet à lui-même et à ses conduites [36]. « Pouvoir » , « savoir » et « éthique » sont ainsi trois pôles d'une étude des rapports entre sujet et vérité, chaque pôle entrant en résonance avec les autres. Le pouvoir et le savoir permettent de constituer le sujet comme une « fonction » au travers de l'application d'un ensemble de techniques [37]. De manière concomitante, le sujet ainsi construit peut s'exprimer pour remettre en cause les cadres dans lesquels il vit et utiliser, pour ce faire, un ensemble de techniques comme le montre l'exemple des techniques de soi que nous avons déjà étudiées.

Pouvoir, savoir et technique sont ainsi associés dans la structuration des pos-

34. Voir « Les techniques de soi » , [Foucault 2001][p. 1610].

35. Voir « Une esthétique de l'existence » , [Foucault 2001][p. 1549 et sq.].

36. Les mots et les choses est consacré au « sujet parlant » dans la mesure où l'ouvrage interroge les conditions de construction et de validité de nos discours. Voir [Foucault 1990a]. *Surveiller et Punir* est l'œuvre dans laquelle Foucault développe son analyse du pouvoir la plus connue.

37. Foucault développe notamment son analyse du sujet comme « fonction » dans « Qu'est-ce qu'un auteur ? » où il affirme que l'auteur est une fonction de discours rendue nécessaire par une « morale d'état civil » . Voir [Foucault 2001][pp. 789--821].

sibilités d'expériences du sujet. La technique occupe une place centrale parmi ces trois pôles et leurs relations : elles permettent à la fois de constituer des éléments de savoir et de transmettre et soutenir des rapports de pouvoir. Les effets de savoir issus de la technique se traduisent aussi par des effets de pouvoir : ils justifient l'emploi de certaines techniques et leur perfectionnement. Dans, *Sécurité, Territoire et Population,* Foucault donne ainsi l'exemple de la vaccination comme cas dans lequel pouvoir et savoirs s'influencent mutuellement au travers de la technique[38]. La vaccination participe à des rapports de pouvoir dans la mesure où elle participe de politiques de santé publique. Elle est aussi mise en œuvre dans le cadre d'un ensemble de savoirs, fournis notamment par la statistique, qui montrent que le risque qu'elle représente à l'échelle individuelle est contrebalancé par ses effets bénéfiques à l'échelle de la population.

L'appel à une perspective historique de la construction du sujet et à l'examen de l'intérêt de la technique dans cette construction constitue une rupture avec les travaux en éthique appliqués à l'informatique. Souvent, l'informatique est considérée comme un objet d'étude spécifique par rapport à d'autres techniques. Il est alors tentant de fonder une « éthique appliquée à l'informatique » comme le font Moor au travers de la *computer ethics* ou Brey dans sa proposition de *disclosive computer ethics*[39]. Néanmoins, l'importance de l'informatique dans la définition des valeurs et la structuration des champs d'action provient de son utilisation conjointe avec d'autres technologies – comme l'électronique ou les statistiques – dans des champs de savoir variés. Le terme « informatique » est souvent utilisé en conjonction avec le nom d'autres champs de savoir ou d'autres techniques. On parle ainsi de « bioinformatique » ou « d'informatique appliquée à la gestion » . Alors que la plupart des domaines de notre vie sont informatisés, une éthique de l'informatique devrait ainsi probablement, prendre la forme d'une enquête plus large sur l'existence dans un monde informatisé.

L'informatique, dont les applications sont protéiformes et changent rapidement, constitue néanmoins un défi pour celui qui veut la penser. Le risque est en effet grand de voir devenir rapidement caduques les réflexions valides à un instant donné. Les premières utilisations du web reposaient ainsi sur l'emploi de nombreux pseudonymes. Ces derniers ont soulevé de nombreuses craintes quant à la possibilité de lier, en ligne, des relations construites sur l'engagement et l'honnêteté des participants, d'où l'essor d'une littérature dédiée à la critique du web et à la mise en évidence de

38. Voir [Foucault 2004][p. 11].

39. Dans « What is Computer ethics ? » , Moor entend fonder une éthique spécifique au monde informatique, motivée par l'absence de considération de l'informatique comme un objet éthique par opposition à d'autres techniques, comme les biotechnologies. Voir [Moor 1985]. Dans « disclosive ethics » , Brey entend lui aussi proposer une éthique focalisée sur l'examen de l'informatique à partir des travaux de John Rawls notamment. Voir [Brey 2000].

son rôle dans l'appauvrissement du contenu des relations sociales [40]. Les nouveaux usages « sociaux » du web ont plus ou moins évincé l'usage de pseudonymes. Ils reposent sur l'utilisation et le partage, par les utilisateurs, de leur état civil et non plus de pseudonymes, d'où la remise en cause de cette forme de critique. Il s'agit moins aujourd'hui de s'interroger sur les portée de la dissimulation de l'identité des utilisateurs que sur leur surveillance permanente. Il s'agit dès lors pour nous moins de participer à une réflexion éthique spécifique à l'informatique que de voir comment des architectures et des usages particuliers d'outils informatiques structurent les possibilités d'action et de rapports aux autres et à soi jusqu'à poser problème et d'identifier des directions afin d'y répondre.

Les réponses apportées aujourd'hui aux difficultés liées à la visibilité des données et à leur conservation sont souvent d'ordre contractuel et guidées par des principes déontologiques. Nous sommes, sur l'Internet des sujets de droits dont la responsabilité est régulièrement engagée : la plupart des sites soumettent leur utilisation à l'acceptation de ces conditions et offrent à leurs usagers des outils pour paramétrer leur visibilité, par exemple. Ces « conditions d'utilisation » , qui précisent le traitement des données collectées par un site Internet et leur utilisation sont les outils privilégiés de la protection de la vie privée, par exemple. Elles jouent le rôle d'outils d'information des utilisateurs qui doivent les lire afin de prendre connaissance des données qui sont enregistrées à leur propos et des traitements qui y sont appliqués. Elles constituent aussi des contrats puisque l'utilisateur doit les accepter afin d'utiliser le service pour lequel elles sont établies. Cet appel aux ressources déontologiques fait écho à l'analyse de techniques différentes de l'informatique. Les notions de « dignité » et de « respect » sont ainsi couramment employées afin d'encadrer l'utilisation des biotechnologies, par exemple dans la « Déclaration universelle sur le génome humain » de 1997 de L'UNESCO [41]. Le droit à la vie privée, est, lui, dans la lignée de la « Déclaration des droits de l'homme » considéré comme une « liberté fondamentale » . Sa protection et sa justification reposent, elles aussi, sur les notions de « dignité » et de « consentement » qui sont au fondement de la « Loi Informatique et Libertés » [42] qui permet, en France, de consacrer un « droit de rectification des données personnelles » .

Néanmoins, quelle protection effective ces mécanismes garantissent-ils ? Dans la plupart des cas, les utilisateurs acceptent de manière presque automatique ces conditions d'utilisations. Leur rédaction dans un langage juridique peu accessible et leur présentation, souvent peu lisible ou peu attrayante, en rendent la lecture et la compréhension complexes [43]. De plus, il faut s'entendre sur l'Internet sur les

40. Nous pensons ici, notamment, aux études de Hubert Dreyfus et Albert Borgmann. L'un et l'autre écrivent avant l'avènement du Web 2.0. Pour eux, le web se réduit ainsi à l'utilisation de pseudonymes et permet l'élaboration de nombreuses stratégies de présentation de soi. Ces conclusions sont fortement remises en question par l'essor du web 2.0 qui met l'accent sur les usages sociaux et repose sur l'utilisation de l'identité réelle des utilisateurs. Voir [Dreyfus 2001] et [Borgmann 1987].

41. On peut ainsi lire dans cette déclaration : « Chaque individu a droit au respect de sa dignité et de ses droits, quelles que soient ses caractéristiques génétiques. » . Voir [UNESCO 1997].

42. Voir [CNIL 2011]

43. Ces remarques servent notamment de point de départ à la critique de la contractualisation

éléments qui manifestent le consentement d'un utilisateur. Certains sites n'hésitent pas ainsi à considérer que le paramétrage du navigateur des internautes exprime la manière dont ils acceptent de voir leurs données collectées ou traitées. La plupart du temps, ces réglages sont automatiques et même lorsque l'utilisateur décide de configurer son navigateur pour indiquer aux sites qu'il visite qu'il ne veut pas voir son activité enregistrée, rien n'oblige ces sites à respecter sa position. Le recours à la protection des droits individuels définis de manière formelle ainsi qu'à la notion de consentement est-il dès lors efficace dans un monde où des éléments matériels déterminent fortement le caractère « éclairé » et effectif du consentement ?

A. Pizzorno pose la même question lorsqu'il évalue le suffrage universel dans l'Italie contemporaine [44]. L'auteur définit le suffrage universel comme exemple d'un moyen de participation politique formellement juste dans la mesure où il vise à accorder une égalité civique aux citoyens en garantissant leur liberté d'expression. Le suffrage universel semble ainsi être l'un des moyens de la possession par le plus grand nombre d'un moyen d'expression politique. Néanmoins, alors qu'il observe la mise en œuvre du suffrage universel en Italie, l'auteur remarque que l'efficacité de ce système de vote comme outil en faveur de la liberté d'expression dépend des conditions effectives dans lesquels il prend lieu. Tout d'abord, comme Tocqueville avant lui, Pizzorno remarque que l'éloignement des centres de décision politique conduit à un morcellement de la force de décision des citoyens qui sont dès lors tentés de s'éloigner des décisions politiques [45]. De plus, il souligne que l'éclatement des expressions politiques en une myriade de positions personnelles réduit leur efficacité dans la mesure où elles se perdent dans un « bruit de fond » [46]. Par conséquent, la taille du corps électoral et l'éloignement des citoyens et des centres de décision politique subvertit la valeur attendue de ce qui devait être un outil de participation politique. L'organisation des votes – par exemple sous la forme d'une loi électorale qui détermine la proportionnalité du scrutin – le nombre de candidats entre lesquels choisir, la publicité de leur programme et la clarté de leur rédaction ainsi que leur honnêteté sont dès lors des éléments importants de l'évaluation du suffrage universel.

L'analyse développée par Pizzorno, qui s'inspire de Foucault, pousse ainsi à quitter le plan de la déontologie pour étudier les conséquences pratiques des techniques que nous utilisons sur nos comportements, leur possibilité et leur portée. La critique féministe issue des réflexions de Catherine MacKinnon, par exemple, invite à interroger la valeur effective de la notion de vie privée dans la protection de l'autonomie individuelle. L'analyse de la constitution de la sphère privée comme un espace séparé des activités productives et politiques montre en effet que l'essor du privé repose sur la division entre des rôles sociaux attribués en fonction du genre de ceux qui

de la protection de la vie privée par Emannuel Kessous. Voir [Kessous 2013][pp. 129 et sq.].

44. Voir [Pizzorno 1989][pp. 236--248].

45. Voir [de Tocqueville 1986].

46. « Le bruit de tout ce qui s'exprime librement est tel que l'expression d'une voix singulière s'en trouve complètement étouffée. » [Pizzorno 1989][p. 242]. Pizzorono écrit aussi : « Il suffit d'être conscient de cette circonstance déprimante que, plus le nombre de voix s'accroît, plus l'efficacité communicationnelle des opinions qui pourraient être exprimées tend vers zéro » [Pizzorno 1989][p. 242].

les incarnent. Dès lors, l'élargissement du privé ne bénéficie potentiellement qu'à un groupe restreint d'individus, principalement masculins, à qui il permet un ascendant plus grand sur les membres de sa maisonnée[47]. À la valorisation *a priori* de la sphère privée, il faut ainsi opposer une réflexion sur la manière dont les rapports entre privé et public sont vécus et structurés. Cette analyse permet de préciser dans quelles mesures le privé peut être un outil d'oppression et comment contrebalancer cette dernière[48].

Sur le plan technique, il est tentant de lier l'usage des techniques de communication numérique à la diminution de l'espace privé ou bien l'élargissement de ce dernier et de la créativité ou de la marge de manœuvre de celui qui en bénéficie. Bien sûr, l'usage des techniques de collecte et de traitement des données dans des stratégies de normalisation est bien connu[49]. Néanmoins, les systèmes de messagerie instantanée, par exemple, permettent à des interlocuteurs qui participent à un événement public de nouer une conversation « privée » en restreignant la visibilité de leurs propos.

L'analyse historique et située des techniques et de leurs effets est centrale dans la réflexion foucaldienne. L'enquête sur le souci de soi répond, dans l'économie de la pensée de l'auteur à une forme de gouvernement dans lequel les individus s'engagent volontairement – le gouvernement de la sexualité[50]. Encouragés par l'apparente libération des discours à propos de la sexualité, les individus s'engagent dans des relations de médicalisation de leurs comportements sexuels – par exemple au travers de la psychiatrie. Ces discours permettent de constituer une « science de la sexualité » chargée de déterminer ce qui est normal et pathologique en matière sexuelle et de guider les actes médicaux. En se situant par rapport aux normes qui entourent la sexualité, les individus acceptent ainsi de confier la détermination de leur identité sexuelle, de sa normalité et de sa correction possible au corps médical et à un ensemble de protocoles d'examens et de gestes correctifs.

Le recours au souci de soi constitue pour Foucault une alternative au « dispositif de sexualité » , c'est-à-dire l'ensemble des procédures qui prennent pour cible la connaissance et la conduite des comportements sexuels. Il n'est pas construit autour d'une « science de la sexualité » mais autour d'un « art de l'existence » , c'est-

47. Dans son *Histoire de la communication moderne*, Flichy met ainsi au jour l'association de la vie privée à des rôles sociaux attribués en fonction du genre de ceux qui les incarnent. Les femmes sont ainsi associées à l'espace et aux tâches domestiques quand les hommes participent aux affaires publiques. C'est aussi le point de vue de Catherine MacKinon. Voir [Flichy 1997] et [MacKinnon 1991].

48. C. MacKinnon propose ainsi de différencier la sphère privée selon le genre de celui ou celle à qui elle est accordée. Il s'agit, pour elle, de prendre en compte la spécificité du sujet féminin et de sa relation à l'enfantement afin de lui accorder une plus grande marge de décision, par exemple sur l'avortement ou la contraception. Voir [MacKinnon 1991].

49. On peut aujourd'hui parler de « marketing des traces » puisque la plupart des publicités auxquelles nous sommes exposés nous sont adressées en fonction de l'étude des traces numériques qui nous concernent. Cette analyse prend la forme d'une « normalisation » puisqu'elle établit des normes de partage entre les différents types de consommateurs afin de déterminer les produits qui leur conviennent. Voir [Kessous 2013][pp. 59--75].

50. Cette analyse fait l'objet de *L'histoire de la sexualité*. Voir [Foucault 1976b]

à-dire la recherche d'un ensemble de comportements qui permettent à un sujet de se constituer un style de vie propre, notamment dans le domaine de la sexualité [51]. L'éthique du souci de soi constitue ainsi une alternative à l'éthique déontologique. Foucault oppose en effet un modèle éthique qui évalue les comportements et les institutions à l'aune de leur rapport à un ensemble de règles à l'éthique du souci de soi, qui évalue le mode de vie du sujet en fonction de son originalité et de la possibilité qu'il offre à l'individu d'être reconnu comme singulier. Le souci de soi repose ainsi sur une « esthétique de l'existence » , c'est-à-dire le rapport à sa vie comme une œuvre que l'on cherche à transformer afin d'en assurer la singularité [52].

Face aux approches déontologiques notamment, l'éthique du souci de soi constitue dès lors une provocation de par son approche empirique de l'éthique et de l'évaluation des techniques. En effet, en insistant sur le rôle des techniques comme opérateur de subjectivation aussi bien comme techniques de soi que comme techniques de gouvernement, Foucault introduit une hétéronomie fondatrice au sein de la représentation de l'agent moral. Ce dernier ne possède plus des droits naturels mais un ensemble de comportements acquis. L'agent du souci de soi est ainsi un agent incarné qui ne possède pas de droits *a priori* – il n'y a notamment pas de sujet originel à libérer. L'éthique du souci de soi conduit ainsi à délaisser le sujet de droit pour lui préférer l'étude de la constitution d'un sujet historique et éthique. Ce faisant, alors qu'il note que les droits naturels sont une forme « d'attitude critique » – ils permettent de refuser une forme de gouvernement en lui opposant une limite – Foucault s'éloigne de cette voie [53]. En effet, dans la mesure où l'individu résulte de procédés de subjectivation, ses droits en résultent aussi [54].

À une évaluation et à une conception des outils techniques guidées par des droits *a priori,* l'éthique du souci de soi conduit à préférer une analyse empirique des techniques, de leurs usages et de leur efficacité. C'est ici que l'éthique du souci de soi s'oppose à une partie de la réflexion en éthique appliquée à l'informatique, et notamment aux réseaux sociaux, qui s'inspire de la notion de « sphère publique » conçue

51. Foucault définit les arts de l'existence comme « des pratiques réfléchies et volontaires par lesquelles les hommes se fixent des règles de conduite, mais cherchent à se transformer eux-mêmes, à se modifier dans leur être singulier et à faire de leur vie une œuvre qui porte certaines valeurs esthétiques et répondent à certains critères de style » . Voir [Foucault 1984a][p. 16].

52. Pour Foucault, à la disparition progressive de l'éthique déontologique doit répondre une « esthétique de l'existence » . Voir « Une esthétique de l'existence » , [Foucault 2001][p. 1551].

53. Dans « Qu'est-ce que la critique ? » , Foucault définit « l'attitude critique » , à partir de la critique kantienne, comme la pratique qui consiste à évaluer les normes qui structurent le monde dans lequel nous vivons. Pour Foucault, cette attitude est spécifique de la modernité. La revendication de droits naturels constitue, pour l'auteur, une forme d'attitude critique puisqu'elle oppose au droit positif des normes alternatives. Voir [Foucault 1990b].

54. Par opposition à une approche fondée sur des droits naturels immuables, Foucault propose ainsi de constituer une « nouvelle déclaration des droits de l'homme » qui prendrait en compte les nécessités de la vie contemporaine. Voir « Face aux gouvernements les droits de l'homme » , [Foucault 2001][pp. 1526--1527]. Dans ce texte, Foucault souhaite établir un « droit absolu à se lever et à s'adresser à ceux qui tiennent le pouvoir » . Il fut question, pour lui, de faire réagir un grand nombre de personnes à ce texte afin de tenter de fonder une nouvelle Déclaration des droits de l'homme.

par Habermas [55]. Foucault lui-même refusait en effet la possibilité de constituer une sphère de communication transparente où pourrait se former l'opinion publique. L'asymétrie concrète de nos relations sociales limitent ce que nous pouvons dire et faire et interdit d'atteindre une situation idéale de communication dans laquelle la communication serait transparente et où chacun pourrait faire entendre sa voix. Il s'agit dès lors non pas de supprimer cette asymétrie, mais de constituer les formes d'asymétrie qui mènent au niveau de domination le plus faible possible [56].

L'approche foucaldienne offre aussi un contrepoint aux théories procédurales comme celle que développe Rawls dans sa *Théorie de la justice* [57]. Rawls envisage la justice de manière « distributive » , c'est-à-dire comme la distribution des biens au sein d'une société. Inspiré par le « règne des fins » kantiens, Rawls propose deux principes de distribution des biens sociaux – une égalité formelle et la minimisation des inégalités – à l'aune desquels évaluer la justice des institutions [58]. Ces principes sont mis en œuvre dans une méthode de constitution d'une société juste. Au moment de leur association, les individus sont placés dans une « position originelle » dans laquelle, placés derrière un « voile d'ignorance » , ils ignorent les places qu'ils occuperont effectivement dans la société afin de les empêcher de réaliser des choix intéressés [59]. À ce moment, les individus ne peuvent répartir les places qu'en fonction des principes rationnels fournis par Rawls. Alors qu'ils obtiennent peu à peu un plus grand nombre d'informations sur la société à propos de laquelle ils prennent des décisions, les futurs citoyens continuent d'appliquer ces principes tout en choisissant la forme des futures institutions qui les gouverneront. Cependant, les

55. Voir [Habermas 1991]. La « sphère publique » constitue pour Habermas la possibilité de se réunir et d'instaurer un contre-pouvoir par la communication et la constitution d'une « opinion publique » . Elle est ainsi un espace distinct de la vie politique et doit, pour fonctionner, fonctionner selon des normes rationnelles de communication. La notion de « sphère publique » telle qu'elle est définie par Habermas est, par exemple, reprise par Charles Ess pour évaluer les réseaux sociaux numériques et souligner que s'ils peuvent permettre l'émergence d'une opinion publique puisqu'ils sont un espace « à part » , ils risquent aussi de l'empêcher puisqu'ils sont, en général, le siège de discussions futiles où les points de vue minoritaires sont peu audibles. Voir [Ess 1996].

56. Foucault lui-même prend ses distances avec une approche d'obédience habermassienne : ´ Le problème n'est donc pas de les dissoudre dans l'utopie de la communication parfaitement transparente, mais de se donner les règles de droit, les techniques de gestion et aussi la morale, l'*êthos*, la pratique de soi, qui permettront, dans ces jeux de pouvoir, de jouer avec le minimum possible de domination. . Foucault affirme aussi : Je m'intéresse bien à ce que fait Habermas, je sais qu'il n'est pas du tout d'accord avec ce que je dis -moi je suis un peu plus d'accord avec ce qu'il dit -, mais il y a cependant quelque chose qui me fait toujours problème : c'est lorsqu'il donne aux relations de communication cette place si importante et, surtout, une fonction que je dirais ´utopique. L'idée qu'il pourra y avoir un état de communication qui soit tel que les jeux de vérité pourront y circuler sans obstacles, sans contraintes et sans effets coercitifs me paraît de l'ordre de l'utopie. » [Foucault 2001][p. 1546].

57. Voir [Rawls 1999].

58. Voir [Rawls 1999][p.53] pour la définition du principe d'égalité des individus. Rawsl écrit *First : each person is to have an equal right to the most extensive basic liberty compatible with a similar liberty for others.* Rawls définit le second principe dans [Rawls 1999][p.311] : les inégalités ne doivent pas désavantager les plus faibles et chacun doit se voir offrir l'égalité des chances.

59. La position originelle est une situation dans laquelle chacun ignore la place qu'il occupera dans la société. Les choix des inégalités et des institutions y sont effectués de manière rationnelle. Voir Voir [Rawls 1999][p.11]

individus placés dans la positions originelle peuvent-ils effectivement dire quoi que ce soit de notre situation et participer à sa réforme ? L'essor des techniques de communication montre en effet la problématisation de manière inédite des rapports entre privé et public et la nécessité de repenser les institutions aussi bien que les outils qui y sont associés. Dans la mesure où de tels changements interviennent sans cesse, l'expérience de la position originelle devrait ainsi être renouvelée en permanence. De même, notre perspective historique sur la définition et la revendication de la vie privée montre que nous existons comme sujets politiques à partir de situations concrètes et historiquement déterminées. Dès lors, les individus purement rationnels de la position originelle ne peuvent rien dire de nos situations et des attitudes qu'elles motivent.

L'éthique du souci de soi permet aussi d'élaborer une critique du potentiel normatif d'une approche procédurale. L'analyse des *espistémés* dans *Les mots et les choses* établit que la construction de la vérité passe par le partage entre un type de discours autorisé et un type de discours irrecevable [60]. Les analyses de *L'Histoire de la sexualité* ou de *Sécurité, Territoire et Population*, établissent, elles, que la normalisation des comportements repose sur la définition d'un partage entre les comportements admissibles et ceux qui ne le sont pas afin de les corriger, de les dénoncer ou de les réprimer [61]. Dans cette perspective, une réflexion qui se fonde sur la définition d'un sujet formel et de ses propriétés, comme la rationalité, constitue une forme de normalisation. Dans la mesure où chaque individu est pris dans des rapports de force, ses intérêts et ses dispositions sont forgés dans ces rapports. Si on suit une approche procédurale, afin de parvenir à se mettre d'accord avec les autres individus, alors que chacun pense à partir de prémisses hétérogènes, il est nécessaire à tous d'objectiver leur jugement. Ces modèles imposent ainsi des contraintes fortes sur la capacité des individus à adopter une position rationnelle. Ils identifient en effet une « bonne rationalité » – qui permet l'argumentation et l'accord et un en-dehors de cette rationalité qui n'a pas le droit de cité [62].

Avec la remise en cause d'un ensemble de travaux en éthique et en philosophique politique, l'éthique du souci de soi conduit à mettre en question tout un pan de la recherche en éthique appliquée à l'informatique. Brey, par la *disclosive* ethics, par

60. Les mots et les choses se proposent d'examiner ce qu'il est possible de dire et de penser à une époque en fonction des normes de vérité en vigueur à une époque donnée. Foucault soutient ainsi que les discours de l'époque classique sont tous organisés en fonction de l'idéal du « tableau » : un énoncé doit répondre à cet idéal afin d'être accepté comme valide. Voir [Foucault 1990a][pp. 86—91].

61. Dans [Foucault 2004][p. 45], Foucault affirme ainsi que la discipline fonctionne à partir d'un partage entre ce qui est permis et ce qui ne l'est pas. L'arsenal disciplinaire a pour but de corriger les dérives des comportements et de rééduquer si nécessaire. La technique disciplinaire par excellence est la loi.

62. Jessica Kulynich propose une comparaison entre la conception politique de Foucault et celle d'Habermas. Elle note que si Habermas soumet la communication à un ensemble de principes rationnels qui obligent à déterminer *a priori* ce qui peut être d'intérêt politique, ce n'est pas le cas chez Foucault. En effet, dans la mesure où la résistance est coextensive au pouvoir, tout rapport de pouvoir, même ceux considérés comme « privés » ou « intimes » peuvent donner lieu à des contestations d'ordre politique. Voir [Kulynych 1997].

exemple, entend proposer un cadre éthique adapté aux spécificités de l'informatique à partir de la pensée de Rawls[63]. Pour ce faire, il propose un ensemble de biens moraux – la distribution équitable des biens, la vie privée ainsi que l'autonomie et la démocratie – au développement desquels l'informatique doit concourir. La *disclosive ethics* est ainsi à la fois un cadre d'évaluation des outils informatiques et de guidage de leur conception afin de favoriser le développement de ces biens. Néanmoins, elle tombe sous le coup de la critique de la normativité des approches procédurales.

Il nous semble que c'est la définition et la place du sujet qui fait difficulté dans ces propositions. Dans les cas où elles ambitionnent de partir des « positions réelles » des individus afin de réformer un dispositif et de tenter, par exemple, de conditionner sa mise en œuvre d'une dispositif, à la satisfaction de cette catégorie ou à la réduction du tort qui lui est fait, elles risquent en effet de perpétuer les modes de domination qui s'appliquent à cette catégorie et à d'autres. Elles supposent, de plus, qu'il est possible de catégoriser les individus de manière fixe et *a priori*. Cela conduit à postuler une identité communautaire qui présente des traits caractéristiques auxquels on peut réduire les subjectivités individuelles. En ne s'interrogeant pas sur la définition des catégories, les règles de langages qui permettent de les utiliser et les pratiques auxquelles elles participent, il nous semble qu'on risque au contraire de perpétuer des formes de domination[64]. L'analyse des modes d'action politique du groupe « Lesbiennes *of colour* » (LOC), constitué de lesbiennes « de couleur » – c'est-à-dire qui ne sont pas caucasiennes – montre que ces dernières ne peuvent se constituer comme des sujets politiques grâce au vocabulaire couramment admis[65]. Le vocabulaire actuel les catégorise soit à partir de leur orientation sexuelle, soit à partir de leur origine ethnique et maintient les deux séparées. Alors qu'elles se confrontent à des problèmes qui les différencient spécifiquement à la fois d'autres lesbiennes et d'autres membres de leur ethnies – comment, par exemple, être une femme voilée et lesbienne – il leur est impossible de les exprimer à partir d'une identité largement reconnue. L'accumulation d'identités communautaires mène ainsi à « l'intersectionnalité » , c'est-à-dire à la construction d'identités sociales qui ne sont pas reconnues. Dans la mesure où elles ne possèdent pas immédiatement de termes pour désigner ce qu'elles vivent, il leur est compliqué de modifier les rapports de pouvoir dont elles sont la cible et doivent, pour ce faire, élaborer un discours qui puisse rendre compte de leur expérience. La perpétuation de catégories existantes risque de faire passer ces dernières pour essentielles et, alors qu'il s'agit de penser une nouveau dispositif, expose au risque de perpétuer l'impossibilité d'exister et d'agir pour certaines subjectivités dans des structures d'autorité préexistantes.

À l'impossibilité de s'exprimer et d'agir comme sujets reconnaissables, les membres

63. Voir [Brey 2000]. Brey propose de construire les outils informatiques à l'aide des principes de Rawls.

64. La sélection et la définition des valeurs que la *disclosive ethics* passe ainsi par le repérage des biens qui sont le plus largement reconnues comme « devant être protégées » et Brey cherche à se prévaloir de définitions largement admises sans toutefois les discuter. Voir [Brey 2000].

65. La sociologue Paola Bachetta livre une étude du groupe qui montre la difficulté pour ses membres de se constituer comme des sujets politiques et la nécessité, pour se faire, de construire de nouveaux outils pour appréhender leur identité. Voir [Bacchetta 2009].

de LOC répondent par la création d'une nouvelle catégorie, les « lesbiennes *of colour* »[66]. L'expression se lit à deux niveaux. Tout d'abord, il s'agit de l'association d'une identité ethnique et sexuelle. Ensuite, l'association du français et de l'anglais rappelle le caractère hybride de la culture de celles qui s'en réclame. En se forgeant un nom, les lesbiennes *of coulour* peuvent ainsi exiger leur reconnaissance et mettre en échec les rapports de pouvoir dont elles sont la cible. Elles échappent aussi à un modèle qui conçoit soit des discriminations raciales soit des discriminations liées à l'orientation sexuelle. Elles se constituent comme sujet politiques et peuvent dès lors se faire entendre et agir. Il nous semble qu'au travers de la création d'une nouvelle expression et la participation à des « démonstrations » comme des manifestations, les lesbiennes *of colour* mettent en pratique un rapport esthétique à elles-mêmes dans la mesure où elles se créent comme agents dans un système qui le leur interdit. Dans ce cas spécifique, il ne s'agit pas pour elles de se prévaloir de droits préétablis mais de montrer que leur situation spécifique appelle une reconnaissance tout aussi spécifique au travers de déplacements langagiers et pratiques[67].

Par conséquent, par opposition à une démarche qui poserait un socle de droits ou de valeurs prédéfinis et instaurerait des mécanismes pour garantir leur respect dans toutes les situations, nous sommes ici face à une démarche qui prend des points de vue particuliers et demande si telle ou telle situation est tolérable. C'est ce que Foucault propose pour des phénomènes comme la prison – en encourageant la diffusion de la parole des prisonniers – mais aussi pour les techniques. Cette approche conduit dès lors à mener une série d'études sur des situations spécifiques.

Si cette démarche est, de manière explicite, à rebours des approches éthiques déontologiques, elle constitue aussi, à notre sens, une alternative à l'utilitarisme. L'utilitarisme est en effet très en vogue dans l'analyse des outils numériques qui sont souvent évalués en fonction de leur caractère pratique ou ludique et de leur capacité à provoquer du plaisir chez les utilisateurs. L'utilitarisme motive aussi bien des études en éthique, mais aussi en sociologie ou en sciences de gestion. Il est aussi au fondement de l'argumentation d'industriels, comme Mark Zuckerberg, PDG de Facebook[68]. La notion de « capital social » , c'est-à-dire l'ensemble des ressources qu'un utilisateur peut obtenir au travers de ses interactions sociales est ainsi au fon-

66. [Bacchetta 2009] : « le collage du français et de l'anglais qui apparaît dans les deux termes waspiennes de France et lesbOccidentées, le caractère d'adjectif et du nom du terme waspiennes de France, et du verbe et du nom dans lesbOccidentées, est de produire des catégories non-essentialisées, non-figées. Ces termes suggèrent et offrent une ouverture aux sujets franco-français amnésiques qui négligeraient encore les problèmes de colonialisme et de racisme. »

67. Dans « Michel Foucault, une interview : sexe, pouvoir et la politique de l'identité » , Foucault tient un raisonnement similaire. Ainsi affirme-t-il que les homosexuels ont besoin d'un « mode de vie gay » . La création de ce mode de vie permet de les constituer comme sujets politiques et, ce faisant, de revendiquer des droits. Voir [Foucault 2001][p. 1554—1665]. Dans [Bacchetta 2009], l'auteure rapproche explicitement la démarche de LOC de celle de Foucault.

68. Mark Zuckerberg est ainsi célèbre pour avoir affirmé que nous acceptions désormais de nous exposer plus que jamais, notamment sur les réseaux numériques car les bénéfices issus de cette exposition, comme la valorisation de nos relations sociales et la personnalisation de la navigation, sont plus importants que la préservation d'un secret qui entourerait nos données. Voir http ://www.theguardian.com/technology/2010/jan/11/facebook-privacy.

dement d'un grand nombre d'études sur la définition et l'usage des outils numériques en sociologie [69]. L'évaluation du capital social repose, le plus souvent, sur l'association d'éléments quantitatifs – comme le nombre d'interactions effectives entre des utilisateurs ou le temps qu'ils passent à utiliser un outil – et qualitatifs – comme la perception que les utilisateurs ont d'un outil [70]. Dès lors, l'évaluation à partir de la notion de capital social corrèle le niveau de plaisir et plus généralement les sentiments positifs provoqués par l'utilisation d'un outil, à sa valeur. Dans la mesure où la gouvernementalité numérique repose sur l'excitation d'affects positifs – qu'ils soient liés au sentiment de sécurité ou au gain d'efficacité qui s'appuient sur les outils numériques, une évaluation des outils numériques à partir de la notion de « capital social » peut difficilement être négative. Dans la mesure où cette évaluation s'attache plus aux « marqueurs » du plaisir des individus qu'au récit de leur utilisation des outils et qu'elle laisse ainsi de côté la structure d'action qu'influence un outil, l'analyse des outils numériques à partir de la notion de capital social tombe sous le coup des mêmes reproches que la gouvernementalité numérique. Elle laisse de côté le rapport à soi que permet d'entretenir la technique. Alors que la critique de la démocratie libérale foucaldienne demande si nous vivons dans un monde tolérable malgré notre apparente liberté, le capital social ne répond en effet pas à la question : tous ces marqueurs positifs nous assurent-ils que notre situation est tolérable vis-à-vis des techniques ?

Au contraire, l'éthique du souci de soi encourage l'acquisition d'un style de vie qui augmente l'intensité de la forme de subjectivité que l'on incarne. Elle pousse à évaluer comment la technique structure la relation à soi, soumet à des influences extérieures et permet effectivement la constitution d'un style de vie propre. Dès lors, toutes les relations sociales ne se valent pas comme le laisserait supposer une évaluation purement quantitative du « capital social » . La constitution d'un style de vie passe en effet par la pratique d'un certain nombre de relations sociales propices au développement de soi, comme l'amitié. Dans cette perspective, la vie humaine apparaît comme une entreprise coopérative. La valeur des interactions avec les membres d'une communauté ou avec des outils repose dès lors sur leur intérêt dans la constitution d'une subjectivité propre, que nous nommons, avec Foucault « intensification » du sujet et qui guide sa méthodologie d'analyse des techniques.

69. Bourdieu définit le « capital social » comme ´ l'ensemble des ressources actuelles ou potentielles qui sont liées à la possession d'un réseau durable de relations plus ou moins institutionnalisées d'interconnaissance et d'interreconnaissance . Voir [Bourdieu 1980][p. 2—3].

70. Mark Burke étudie ainsi les effets de l'utilisation de *Facebook* sur le « capital social » de ses utilisateurs afin de permettre aux constructeurs d'applications informatiques de fournir des applications plus attractives. Voir [Burke 2011]

III Le souci de soi et l'éthique de la vertu

Nous avons jusqu'ici insisté sur le caractère courrent de l'éthique du souci de soi face aux approches principales de l'éthique appliquée à l'informatique. De l'aveu même de Foucault, l'éthique du souci de soi est aussi une éthique de « résistance » puisqu'elle a pour but de permettre la transformation des rapports de pouvoir qui s'appliquent sur les sujets [71]. Néanmoins, l'auteur n'exprime pas de position définitive sur les raisons pour lesquelles une telle résistance est nécessaire ou sur les résultats qu'il faudrait en espérer. Cette absence nourrit l'une des critiques récurrentes à l'encontre de Foucault : alors qu'il critique, par exemple, les valeurs libérales, Foucault ne propose pas explicitement de fondement normatif qui prendrait leur place et pourrait justifier sa critique. En d'autres termes, Foucault ne pourrait pas répondre à la question « pourquoi résister ? » alors que toute son œuvre incite à la résistance. Dès lors, suivre Foucault entraînerait l'adoption d'une forme de relativisme [72]. Pourtant, Foucault adopte explicitement des valeurs comme la liberté. Sa reprise de la question kantienne « Qu'est-ce que les Lumières ? » se présente ainsi à la fois comme une relecture de son travail et comme une présentation de son but, la préparation au « travail infini de la liberté » [73]. Faudrait-il dès lors voir en Foucault un penseur libéral qui ne s'assume pas ?

Afin d'éclairer le potentiel normatif de l'éthique du souci de soi, nous proposons désormais de la définir comme une forme spécifique d'éthique de la vertu. Ce faisant, nous répondrons à une seconde critique adressée au travail de Foucault en éthique qui dénonce le caractère esseulé du sujet éthique du souci de soi qui ne semble se définir que dans le rapport qu'il entretient à lui-même [74]. Notre démarche est notamment guidée par la parenté des sources antiques de l'éthique du souci de soi et de l'éthique de la vertu et de la place centrale qu'elles font au sujet conçu comme historique [75].

71. Ainsi Foucault n'hésite-t-il pas à affirmer que l'éthique du souci de soi est une « pratique de la liberté » . Voir « L'éthique du souci de soi comme pratique de la liberté » [Foucault 2001][p. 1529].

72. Telle est le position de Nancy Fraser qui reproche à Foucault de ne pas proposer de valeurs concurrentes à celles du libéralisme voire, dans certains textes à reprendre ces valeurs tout en les critiquant. Voir [Fraser 1985].

73. Voir « Qu'est-ce que les lumières ? » , [Foucault 2001][pp. 1381--1387].

74. La critique de Derrida est probablement la mieux connue sur ce point. Derrida accuse en effet Foucault de « subjectivisme » puisque Foucault réduit le sujet à une fonction du discours. Voir [Derrida 1979].

75. L'éthique de la vertu fait l'objet d'un regain d'intérêt dans les travaux éthiques consacrés à l'informatique. Dans « Social Networking and the Virtues », Shannon Vallor construit ainsi, à partir de l'éthique de la vertu, une approche qu'elle veut alternative à l'évaluation du capital social. À l'étude du nombre d'interactions entre les utilisateurs, elle propose de substituer l'analyse du type d'individus que les outils numériques permettent de construire. Voir [Vallor 2010]. Telle est aussi la voie que prend Maria Bakardjeva dans « Web 2.0 Technologies of the self ». Neil Levy propose d'étudier Foucault à partir de l'éthique de la vertu afin de préciser les enjeux de l'éthique du souci de soi. Nos conclusions diffèrent des siennes dans la mesure où nous soulignons plutôt les différences entre les deux approches – et notamment la définition originale de la vertu comme critique chez Foucault – que leurs ressemblances. Voir [Levy 2004].

L'expression « éthique de la vertu » désigne, principalement dans la pensée anglo-saxonne contemporaine, une éthique qui met moins l'accent sur les notions de codes et de règlements que sur le caractère de l'agent moral [76]. L'évaluation des actes répond ainsi à la question « l'agent possède-t-il les vertus adéquates ? » et non pas à la question « se comporte-t-il en accord avec tel ou tel impératif ? » [77]. Prenant sa source notamment dans la pensée aristotélicienne l'éthique de la vertu participe ainsi de sources antiques, tout comme l'éthique du souci de soi.

Certes, les sources antiques de Foucault et de l'éthique de la vertu sont différentes. Foucault n'évoque ainsi ni l'*Éthique à Nicomaque* ni plus généralement Aristote lorsqu'il définit le souci de soi. Il se concentre plutôt sur l'étude de Platon où d'auteurs postérieurs comme les Stoïciens. Néanmoins, il faut noter que le rapport qu'entretiennent Foucault et les penseurs de l'éthique de la vertu à leurs sources antiques n'est pas un rapport de fidélité à des sources historiques, ou de redite. Foucault prône ainsi « l'actualisation » des analyses du souci de soi quand MacIntyre, par exemple, entend, par le recours à l'éthique de la vertu, répondre à la difficulté contemporaine que constitue l'absence de consensus en éthique [78].

Dès lors, l'analyse que font ces auteurs des textes antiques peut se rejoindre dans des thèmes communs, dont le souci de soi fait partie, tout comme la priorité de l'évaluation des dispositions sur l'évaluation de la correction des comportements vis-à-vis d'une norme incarnée, par exemple, par une loi. Bien sûr les lois font-elles l'objet d'investigation de la part des Antiques comme le montrent la *Politique* ou *Les Lois*. Néanmoins, comme le notent Foucault ou MacIntyre, les morales antiques sont plutôt orientées vers la recherche de la vie bonne ou belle. Cette dernière repose sur un certain nombre de pratiques à partir desquels sont définis des critères d'excellence. Afin d'évaluer la réussite de l'agent, l'analyse porte dès lors autant sur son caractère et le style de vie qu'il met en œuvre à l'aune de ces critères d'excellence que sur l'évaluation de son respect de lois morales. Le sage de *l'Ethique à Nicomaque* est ainsi évalué selon sa possession de la vertu de la prudence et sa capacité à la mettre en œuvre. Le sage stoïcien, tel qu'il est notamment décrit dans le *Manuel* évalue, lui sa réussite en fonction de sa capacité à ne pas se laisser dominer par les évaluations qu'il porte sur les choses. Alors que les traits de caractères de l'agent sont analysées comme des facteurs d'hétéronomies dans le cadre d'une approche déontologique ou sont laissés de côté lorsqu'on évalue le plaisir que peut ressentir l'agent, ils constituent l'objet de l'intérêt commun que Foucault et les penseurs de l'éthique de la vertu portent aux sources antiques.

Cet intérêt répond au diagnostic de l'épuisement du vocabulaire moral contemporain que partagent Foucault, Anscombe ou encore MacIntyre [79]. Ils apportent,

76. Alasdair MacInture, Elizabeth Anscomne ou encore John MacDowell sont les penseurs les plus connus de l'éthique de la vertu.

77. A. MacIntyre définit les vertus de la manière suivante : « Une vertu est une qualité humaine acquise dont la possession et l'exercice tendent à permettre l'accomplissement des biens internes aux pratiques et dont le manque rend impossible cet accomplissement » . Voir [MacIntyre 1997][p. 186].

78. Voir l'introduction à *Après la vertu* [MacIntyre 1997]

79. Nous avons déjà précisé que Foucault soutient que l'éthique déontologique décline à son

par ailleurs des solutions proches à un tel épuisement. La notion de « vertu » - qui désigne les pratiques ou les dispositions d'un individu – est ainsi la concurrente de la notion de « règle » , par exemple, comme principe rationnel et critère d'excellence qui permet d'évaluer les comportements et leur moralité [80]. G.E Anscombe, ou encore Alistair MacIntyre, deux défenseurs contemporains de l'éthique de la vertu opposent ainsi la notion de vertu aux impératifs que l'on trouve chez Kant et les penseurs qui s'en inspirent. Sous leur plume, l'éthique de la vertu constitue ainsi un projet éthico-politique qui vise à concurrencer notamment les formes de contractualisme qui s'inspirent de la déontologie. MacIntyre dénonce ainsi explicitement les théories, comme celle de Rawls, qui prennent comme modèle de l'agent moral des sujets purement rationnels. Il oppose au sujet formel que Rawls situe au fondement de sa démarche, un sujet historique dont le vocabulaire moral est tout entier tiré de sa vie en communauté.

Anscombe propose, elle, de bannir le vocabulaire déontologique du lexique moral [81]. Cette proposition repose sur l'affirmation que l'éthique de la vertu rend compte de manière adéquate de la manière dont nous vivons la morale, ce que ne parviennent à faire ni l'éthique déontologique, ni l'éthique utilitariste. Les paradoxes entraînés, par exemple, par l'interdiction absolue de mentir que Kant établit ne sont pas nécessaires lorsque l'on étudie les conduites et qu'on les justifie à partir de l'éthique de la vertu [82]. Il ne s'agit en effet pas de demander si mentir est une action universellement mauvaise mais si, dans un cas spécifique, mentir permet de mettre en œuvre des traits de caractères qu'il est souhaitable d'obtenir. En ne faisant aucune différence entre les agents moraux, l'éthique déontologique ne permet pas, par exemple, de justifier des conduites habituelles comme le fait de faire passer nos proches avant des étrangers, ou bien la variété des codes moraux en fonction des époques et des espaces culturels. L'usage du terme « vertu » est aussi avéré chez Foucault par opposition à une morale fondée sur un code détaillé et contraignant [83]. Alors qu'il présente le souci de soi comme une alternative à la déontologie, Foucault définit même « l'attitude critique » , c'est-à-dire le comportement qui consiste à remettre en cause les formes de pouvoir, comme une vertu. D'autre part, il consacre *Le gouvernement de soi et des autres* à l'étude de la vertu du courage, la *parrhesia*,

époque. MacIntyre, lui, souligne dans plusieurs de ses ouvrages que les théories morales contemporaines ne peuvent permettre d'atteindre un consensus. Enfin, Anscombe prône l'abandon du vocabulaire déontologique.

80. Plusieurs chapitres d'*Après la vertu* sont consacrés à la définition de la vertu. MacIntyre note en effet que selon les époques le terme connaît plus acception. Le point comment entre ces acceptions est cependant la prédominance d'une approche prudentielle des comportements moraux, par opposition à un comportement gouverné par la recherche de la fidélité à une règle.

81. Voir [Anscombe 1958].

82. Kant établit qu'il est impossible de se prévaloir d'un droit de mentir dans la mesure où ce droit mettrait à mal la possibilité de conclure des promesses. Comment faire confiance à quelqu'un qui ment ? Cependant, l'interdiction absolue de mentir entraîne des interprétations contre-intuitives de cas éthiques. Comment, par exemple, refuser de mentir à un individu qui veut du mal à quelqu'un que l'on a accepté de cacher ? Voir [Kant 1988].

83. Nous pensons bien sûr ici à l'usage du terme vertu dans *L'histoire de la sexualité*, mais Foucault emploie aussi le terme dans d'autres textes comme « Qu'est-ce que la critique ? » .

qu'il prend comme exemple d'attitude critique [84].

La notion de « tradition » fait partie, dans l'éthique de la vertu, des concepts centraux dans l'explication et l'évaluation des comportements. Par « tradition » , il faut entendre le contexte historique qui donne sens au vocabulaire que nous utilisons [85]. Bien que la notion de tradition ne soit pas présente en tant que telle chez Foucault, il nous semble qu'elle partage des traits communs avec les notions « d'*espistémé* » ou de « dispositif » . Ainsi l'analyse des rapports entre les hommes dans l'Antiquité vise-t-elle à montrer qu'elle est différente de celle des homosexuels contemporains dans la mesure où elle ne participe pas d'un cadre médical [86]. Dans la mesure où nos savoirs et nos pratiques structurent nos capacités d'action, les dispositifs dans lesquels nous vivons sont ainsi des dispositifs de valeurs. Tout comme il le fait en épistémologie, Foucault opère ainsi une redéfinition de l'objectivité en matière morale comme ce qui est non pas conforme à un principe universel mais accepté à une époque et dans un lieu donné. L'objectivité est réduite au « dire vrai » ou au « bien faire » . Ce faisant, dans la mesure où nos dispositifs laissent leur marque sur nous, Foucault permet de distinguer la compréhension d'une conception, la reconnaissance de sa rationalité et sa force, de sa valeur pour nous. Les valeurs nous sont en effet imprimées par l'expérience et la répétition. Les partages entre le défendu et le permis ou bien entre le monstrueux et le normal qui sont ainsi au cœur de *L'histoire de la sexualité* notamment permettent ainsi de rendre compte de la constitution de modes de comportements. [87]. La simple acceptation d'une vérité ne peut se substituer à ces phénomènes. La tentative de construction d'une « ontologie critique du présent » permet en effet de mettre au jour nos orientations morales [88]. Enfin, il faut se rappeler que les « arts de l'existence » que Foucault envisage comme des pratiques de liberté qui permettent d'adopter un « style » de comportement donné prennent pour appui des styles de vie – comme celui du sage stoïcien – définis dans la société.

La notion de tradition permet de pointer un manque des conceptions déontologiques qui laissent de côté la tradition morale pour lui préférer des principes universels et définis hors de toute histoire. Elle permet aussi de rendre compte de la diversité des pratiques morales sur les plans historique et culturel tout en évacuant

84. Foucault définit la critique comme vertu dans « Qu'est-ce que la critique ? » . Voir « le gouvernement de soi et des autres » . La *parrhesia* est la « libre parole » , Foucault en fait un devoir et un droit du gouverné qui peut interpeller les gouvernements afin de leur demander des comptes. voir « Une esthétique de l'existence » . [Foucault 2001][p. 1555].

85. Voir sur ce point [MacIntyre 1997].

86. Voir « Des caresses d'homme considérées comme un art » [Foucault 2001][p. 1134—1136].

87. « C'est vouloir la traiter [la sexualité] comme la corrélation d'un domaine de savoir, d'un type de normativité, d'un mode de rapport à soi ; c'est essayer de déchiffrer comment s'est constitué dans les sociétés occidentales modernes, à partir et à propos de certains comportements, une expérience complexe où se lie un champ de connaissance (avec des concepts, des théories, des disciplines diverses), un ensemble de règles (qui distinguent le permis et le défendu, le naturel et le monstrueux, le normal et le pathologique, le décent et ce qui ne l'est pas, etc.), un mode de relation de l'individu à lui-même (par lequel il peut se reconnaître comme sujet sexuel au milieu des autres). » Voir [Foucault 2001][p. 1398].

88. Tel est le but de « Qu'est-ce que les Lumières ? » .

la possibilité d'une subjectivité radicale des jugements moraux. Nos évaluations morales ne sont pas des choix personnels puisqu'elles sont forgées dans le creuset du vocabulaire commun à partir des mots et des règles d'usage que nous acquérons en société. L'éthique de la vertu et l'éthique du souci de soi partagent ainsi la même thèse sur l'histoire de la pensée morale. La concentration sur un code moral est une préoccupation moderne qui s'est faite au détriment de la considération du caractère de l'agent [89]. Cette constatation se traduit, dans les deux courants, par le déplacement de l'intérêt pour le sujet de droit vers l'examen du sujet éthique, au sens où l'éthique est entendue comme *ethos*. Foucault note ainsi que si les codes moraux existent toujours, leur importance peut-être plus ou moins grande vis-à-vis de la considération des traits de caractères souhaitables chez un individu. De plus, il souligne que l'éthique déontologique et l'éthique de la vertu sont mutuellement exclusives. Les éthiques antiques, pour Foucault, sont ainsi caractérisées par la recherche d'un « style de liberté » , recherche qui a disparu des formes contemporaines d'éthiques [90]. La plupart sont en effet fondées sur un rapport de connaissance à l'agent moral – elles tentent d'en déterminer la structure formelle, la nature ou les droits – que sur l'intérêt pour l'attention que ce sujet peut se porter à lui-même [91].

Alors qu'il envisage l'éthique comme le rapport à soi-même qui permet de s'améliorer et de se modifier, il note que plus l'importance donnée au code est grande, moins la place laissée au « souci de soi » est importante [93]. Le recours à la notion de vertu permet ainsi de déplacer la réflexion morale et sur la technique vers un sujet éthique dont le mode d'être n'est pas de l'ordre de la connaissance mais de l'exercice de soi et de la pratique. Il faut s'entraîner à soi non seulement pour acquérir un style de vie propre mais, plus généralement, pour être vertueux. Il faut, par exemple, pour être charitable, apprendre à repérer les situations qui exigent de nous la charité et parvenir à établir le don approprié. En mettant l'accent sur la répétition nécessaire à l'acquisition d'un type de comportement, on souligne l'attitude volontaire d'un individu qui se prend comme objet d'une transformation de lui-même qu'il juge désirable. Cette approche permet ainsi de prendre en compte les intentions du sujet dans la mesure où le souci de soi fait du sujet une entité active dans sa constitution. La rationalité mise en avant ici n'est pas d'ordre instrumentale. Le sujet répond à un désir du bien, c'est en cela que les actions du sujet éthique sont « volontaires » [94].

89. MacIntyre ouvre ainsi *Après la vertu* sur une fiction pour faire comprendre la perte que constitue l'effacement de l'éthique de la vertu au profit d'une éthique fondée sur le code ou les conséquences. Il imagine un monde où les connaissances scientifiques auraient disparus et où les connaissances enseignées et les pratiques scientifiques auraient perdu tou sens puisqu'elles ne seraient plus justifiées par ces principes. Foucault, lui, souligne que l'éthique de la vertu est originairement grecque tandis que l'époque moderne a conçu l'éthique à partir du prisme de la déontologie dans « Une esthétique de l'existence » Voir [Foucault 2001][p. 1540].

90. Voir « Une esthétique de l'existence » , [Foucault 2001][p. 1550].

91. Dans « Les techniques de soi » , Foucault affirme ainsi que la maxime « connais toi toi-même » a pris le pas sur la maxime « prends soin de toi-même » qui était centrale dans les éthiques antiques. [92][p. 1607].

93. Voir [Foucault 2001][p. 1607].

94. Voir [Foucault 1984b] où Foucault affirme que le sujet est « volontaire » et met en œuvre les techniques de soi au terme d'une « pratique réfléchie » et équipée d'un ensemble de valeurs.

Malgré leur proximité, l'éthique du souci de soi et l'éthique de la vertu ne peuvent cependant être réduites l'une à l'autre. Tout d'abord, dans une perspective contemporaine, le recours à l'éthique de la vertu est parfois présenté comme une manière de corriger une fausse route de la philosophie morale. Tel est le cas, par exemple, de MacIntyre [95]. Certes Foucault analyse-t-il la création des éthiques contemporaines comme le fruit du délaissement du souci de soi pour lequel il a une préférence marquée. Toutefois, il refuse de considérer que la philosophie morale s'est dévoyée et qu'il la sort de son erreur par la reprise du thème du « souci de soi » .

Il nous semble que l'éthique de la vertu et l'éthique du souci de soi raisonnent à partir de deux acceptions du terme de vertu. Tout d'abord, la vertu peut être entendue comme l'excellence que l'on démontre dans une conduite [96]. Dans cette perspective, la vertu est définie par des canons sociaux. C'est dans ce sens que MacIntyre l'entend lorsqu'il évoque la définition de la vertu en rapport avec une pratique organisée [97]. Deuxièmement la vertu est aussi définie par un certain rapport de remise en question des normes [98]. Les vertus aristotéliciennes, par exemple, reposent ainsi toutes sur la prudence de l'agent qui doit être capable de déterminer l'occasion propice à son action ainsi que les modalités adéquates. La prudence constitue ainsi une forme de mise en question des normes de comportements dont elle interroge la définition à l'aune d'une situation spécifique [99]. Il s'agit, par exemple, de savoir ce que signifie la pudeur à un moment donné, et donc de déterminer ce que l'on doit partager avec une personne spécifique avec laquelle nous partageons une relation déterminée. La vertu est ainsi à la fois l'expression d'une conformité à un idéal social et la mise à l'épreuve de cet idéal au contact d'un monde en perpétuel changement. Face à ces changements, la vertu doit dès lors sans cesse trouver de nouvelles expressions.

John MacDowell affirme que la question fondamentale à laquelle l'éthique de la vertu souhaite répondre est « comment doit-on vivre ? » [100]. Foucault se distingue dès lors des penseurs de l'éthique de la vertu dans la mesure où il analyse prin-

MacIntyre écrit, lui, : « Nous ne pouvons donc pas décrire le comportement indépendamment des intentions, ni les intentions indépendamment du décors qui les rend intelligibles aux agents et à nous » . Voir [MacIntyre 1997][p. 201].

95. MacIntyre écrit : « Il ne semble exister aucun moyen de parvenir à un accord moral dans notre culture » [MacIntyre 1997][p. 9].

96. Aristote, dans l'*Ethique à Nicomaque*, dresse ainsi une définition de la vertu dans la deuxième partie du livre et une liste des vertus particulières dans la quatrième partie. Voir [Aristote 1992].

97. « Une vertu est une qualité humaine acquise dont la possession et l'exercice tendent à permettre l'accomplissement des biens internes aux pratiques et dont le manque rend impossible cet accomplissement » [MacIntyre 1997][p. 186].

98. La définition de la vertu, à partir de la notion de « critique » fait l'objet de « Qu'est-ce que la critique ? » . Foucault affirme ainsi que la vertu est un certain rapport critique aux normes. Voir [Foucault 1990b].

99. Aristote définit la prudence de la manière suivante : « Si donc les hommes prudents ont pour caractère propre le fait d'avoir bien délibéré, la bonne délibération sera une rectitude en ce qui concerne ce qui est utile à la réalisation d'une fin, utilité dont la véritable conception est la prudence même » . Voir [Aristote 1992][1142 b31 - 33].

100. John MacDowell, dans « Virtue and reason » , affirme que toute réponse à la question « comment doit-on vivre ? » appelle une réponse fondée sur la notion de vertu. Voir [MaxDowell 1979].

cipalement cette question dans des moments où les normes qui permettraient d'y répondre sont en crise. Herculine Barbin, par exemple, ne peut trouver de normes à partir desquelles construire son existence comme hermaphrodite [101]. Certes Foucault analyse-t-il des formes de subjectivations afin de discerner les règles qui permettent de les constituer comme telles. Néanmoins, le terme de « vertu » est utilisé chez Foucault pour caractériser l'attitude critique ou le courage, deux pratiques qui visent à contester un régime de vérité et, ce faisant, des formes de subjectivations. Si MacIntyre, par exemple, accentue la place de la vertu comme excellence, l'éthique du souci de soi se distingue de l'éthique de la vertu par la prépondérance qu'elle accorde à la critique comme vertu, c'est-à-dire à la vertu comme remise en cause des normes. L'éthique du souci de soi vise ainsi la transformation des normes et, par suite du sujet qui les pratique. La pratique vertueuse foucaldienne permet ainsi de créer un sujet par sa destruction puisque Foucault souligne, dans tous ses travaux « ce que nous ne sommes plus » ou « ce qu'il n'est plus possible d'être » [102]. Foucault fait ainsi œuvre de critique en identifiant les formes de subjectivité permises par un dispositif. Cette identification est le point de départ de la contestation de formes spécifiques de gouvernement. C'est en cela que la *parrhesia* constitue une attitude critique. Elle remet explicitement en cause les normes du vrai en vigueur à une époque et les modes de gouvernements qu'elles permettent.

Nous touchons ici à la spécificité de l'approche éthique foucaldienne. Certes l'auteur met-il en exergue le rôle de la socialisation dans la constitution de notre subjectivité. [103] Néanmoins, la vertu par excellence est celle qui, en constituant le sujet comme sujet éthique dans son rapport à lui-même, le désassujettit [104]. Foucault n'affirme ainsi pass, contrairement à MacIntyre, que nous avons des désirs naturels – comme celui de ne pas partager toute notre vie avec les autres – ou que nous obéissons à une « voix intérieure » . Au contraire, il encourage la création d'une

101. Foucault, publie les mémoires d'un hermaphrodite du XIX^{e} siècle nommée Herculine Barbin. Considéré tout d'abord comme une femme puis opéré afin d'avoir un phénotype masculin. Les mémoires de Barbin relatent ses difficultés à avoir une vie amoureuse ou obtenir un emploi. Ils relatent aussi la difficulté de vivre une existence qui n'est ni à la fois celle d'un homme, ni celle d'une femme mais d'une forme de genre intermédiaire et la confrontation avec des institutions juridiques et médicales qui prennent en charge la « rectification » de son identité. Voir [Foucault 1996].

102. Nous empruntons cette position à Judith Butler qui en fait sa thèse principale dans l'interprétation qu'elle livre de la notion de « critique » chez Michel Foucault. Voir [Butler 2002].

103. « Le sujet se constitue à travers des pratiques d'assujettissement, ou, d'une façon plus autonome, à travers des pratiques de libération, de liberté, comme, dans l'Antiquité, à partir, bien entendu, d'un certain nombre de règles, styles, conventions, qu'on retrouve dans le milieu culturel » voir « Une esthétique de l'existence » ,. [Foucault 2001][p. 1552].

104. Alors que Foucault reprend la notion de souci de soi de la pensée antique, et notamment des Stoïciens, il remarque que ce dernier vise principalement la subjectivation à partir du passé, qui seul nous appartient. Le souci de soi foucaldien est ainsi original par rapport au souci de soi stoïcien dans la mesure où il vise l'avenir que les Stoïciens laissent de côté. Voir « L'écriture de soi » [Foucault 2001][p. 1234].

identité sous la forme d'une différenciation de soi par rapport à soi [105]. La pensée de l'intériorité, qui fonde le plus souvent le recours au vocabulaire intentionnel est en effet étrangère à Foucault. Foucault fait de la liberté une pratique et non pas une propriété obtenue à l'issue de la libération d'un sujet intérieur. Ainsi, si la liberté est la « condition ontologique » de la l'éthique, dans la mesure où la liberté est une pratique, cette dernière est la condition ontologique de l'éthique, pas le sujet lui-même [106]. L'opposition entre Foucault et l'éthique de la vertu se situe ainsi dans la définition du sujet. La conception d'un sujet tout entier historique chez Foucault heurte en effet la définition d'un sujet substantiel dans une partie de l'éthique de la vertu contemporaine [107]. Si MacIntyre, par exemple, souligne que les vertus sont issues de la socialisation, il maintient l'existence d'un sujet substantiel, garant de la cohérence de l'identité personnelle [108]. Foucault, au contraire, en réduisant le sujet à une fonction du discours et en liant la subjectivation aux rapports de force dans lesquels elle est prise, abandonne cette conception. Sous l'agent moral ne se tient en effet pas un sujet à libérer ou qu'il faut examiner afin d'en tirer les propriétés à actualiser, mais un individu complètement construit et qui ne possède une intériorité que dans la mesure où il se l'est créée [109]. Pourtant, le sujet éthique foucaldien n'est pas complètement réduit à un effet du dispositif dans lequel il vit. Foucault instaure l'intensification de la subjectivité comme le but de l'éthique de soi. Lorsque la tradition fait défaut, la recherche d'une intensification de la subjectivité constitue ainsi un moyen de construire des raisons « objectives » et fortes pour orienter nos actions dans telle ou telle direction dans la mesure où elle participe à la construction de l'agent. La notion d'intensification permet ainsi de départager les « techniques de

105. Foucault écrit ainsi : « Les rapports que nous devons entretenir avec nous-mêmes ne sont pas des rapports d'identité ; ils doivent être plutôt des rapports de différenciation, de création, d'innocation. » . Voir « Michel Foucault, une interview : sexe, pouvoir et la politique de l'identité » . Voir [Foucault 2001][p. 1556].

106. Voir « L'éthique du souci de soi comme pratique de la liberté » [Foucault 2001][p. 1531].

107. Dans [Foucault 1984b], Foucault affirme ainsi : « J'ai toujours été un peu méfiant à l'égard du thème général de la libération, dans la mesure où, si l'on ne le traite pas avec un certain nombre de précautions et à l'intérieur de certaines limites, il risque de renvoyer à l'idée qu'il existe une nature ou un fond humain qui s'est trouvé, à la suite d'un certain nombre de processus historiques, économiques et sociaux, masqué, aliéné ou emprisonné dans des mécanismes, et par des mécanismes de répression »

108. La conception du sujet chez Foucault fait l'objet de nombreuses critiques chez MacIntyre. Voir notamment sur ce point [MacIntyre 1991] chapitre IX. MaxIntyre prend pour départ le texte de Foucault « Qu'est-ce qu'un auteur ? » et dénonce le caractère anonyme du sujet foucaldien, qui n'est qu'une « fonction » du discours. Il dénonce aussi sa production par les rapports de pouvoir. Pour MacIntyre, Foucault ne peut ainsi saisir l'identité personnelle des sujets et, par conséquent, rendre compte des justifications des actions et des histoires personnelles. Il nous semble que les études du souci de soi permettent de nuancer cette critique. En effet, Foucault y présente un sujet qui construit et met en forme son identité et sa conduite. De plus, le sujet du souci de soi part d'une dispersion initiale de ses actes et de son identité, de sa soumission à des rapports de force pour se constituer une identité propre.

109. Foucault affirme ainsi : « on est toujours à l'intérieur. La marge est un mythe. La parole du dehors est un rêve qu'on ne cesse de reconduire. On place les « fous » dans le dehors de la créativité ou de la monstruosité. Et, pourtant, ils sont pris dans le réseau, ils se forment et fonctionnent dans les dispositifs du pouvoir. » . Voir « L'extension sociale de la norme » , [Foucault 2001][p. 77].

soi » – propres à favoriser la constitution d'un style de vie propre d'un ensemble de techniques qui permettent une domination potentielle. Ainsi le panoptique permet-il de constituer « l'âme des prisonniers » au travers de techniques d'observation et d'orthopédie du comportement. Une telle âme est très contrainte. Lorsqu'elles sont parfaitement appliquées, les sujets ne s'appartiennent plus. Ils sont assujettis sans jamais pouvoir obtenir le statut de sujet dans la mesure où ils n'ont jamais le contrôle de l'usage de ces techniques [110]. Par opposition, le but de l'éthique du souci de soi est de permettre l'appropriation de soi par soi.

Le sujet du souci de soi cherche ainsi, par la pratique d'exercices, à s'approprier sa propre subjectivité afin d'atteindre un style qui, dans le cadre de l'exemple des stoïciens, est issu de la vie sociale. Il est dès lors tentant d'interpréter Foucault comme un auteur qui substitue, à l'importance que certains auteurs accordent à la nature humaine comme fondement de l'attitude vertueuse, la tradition. Cette dernière constituerait le creuset dans lequel se trouveraient les formes qui motiveraient nos actions. De fait, la notion de « dispositif » , qui influence nos actions, pourrait sembler jouer le rôle de la tradition. Cependant, cette interprétation doit être nuancée puisque Foucault situe la nécessité de l'éthique du souci de soi dans la crise des traditions et le diagnostic des limites des dispositifs dans lesquels nous vivons.

Définir l'éthique du souci de soi comme une éthique de la vertu permet néanmoins de réfuter les accusations de subjectivisme formulées à l'encontre du projet éthique de Foucault. Les vertus soutenues par Foucault, comme la liberté ou l'amitié, proviennent en effet de l'insertion dans la vie sociale. L'éthique du souci de soi conduit cependant non seulement à se demander comment incarner telle ou telle vertu mais aussi à poser cette question dans les cas où les normes qui permettraient de guider l'action sont en crise. C'est pour cette raison que l'éthique du souci de soi est en quelque sorte frustrante [111].

Il nous semble que l'importance, dans l'éthique du souci de soi, de la prise en compte de situations radicalement nouvelles, où l'individu ne peut plus agir comme il le faisait avant, ou refuse de la faire, constitue l'un des intérêts de cette éthique dans l'analyse des outils informatiques. Dans la mesure où la communication numérique impose de nouvelles normes de publication et de partage des contenus [112] et modifie nos pratiques, il s'agit en effet de se demander comment, dans ce nouveau cadre,

110. Foucault écrit ainsi que l'âme est « le corrélatif d'une certaine technologie du pouvoir sur le corps [produit] en permanence, autour, à la surface, à l'intérieur du corps par le fonctionnement du pouvoir » Voir [Foucault 1993][p. 38].

111. Foucault, alors qu'on lui demande quelles institutions créer afin de faciliter la vie des homosexuels affirme ainsi : « Les question de savoir quels types d'institutions nous devons créer est une question capitale, mais je ne peux pas y apporter de réponse. Notre tâche, je crois est d'essayer d'élaborer une solution » . Voir « Michel Foucault, une interview : sexe, pouvoir et la politique de l'identité » . Voir [Foucault 2001][p. 1554].

112. Sur ce point, voir [danah boyd 2008] où l'auteure analyse les normes de communication en vigueur sur les réseaux sociaux numériques. Emmanuel Kessous livre une étude comparable dans son analyse du partage des données numériques. L'auteur souligne que la structure des outils numériques conduit aujourd'hui à considérer « l'attention » que l'on reçoit et que l'on donne, comme un bien. Voir [Kessous 2013].

mettre en pratique nos vertus de manière renouvelée [113].

IV Conclusion

Nous avons ouvert cette partie en nous demandant quels étaient les effets de la technique quant à la structuration de notre visibilité, pournous-mêmes et pour les autres. Nous avions déjà souligné l'influence de la technique sur le milieu social. Nous avons adopté, à la fin du premier moment de notre réflexion, une thèse plus radicale encore. En mettant en forme la visibilité de nos conduites et nos moyens d'action, la technique nous est apparue participer à la construction de notre subjectivité. Néanmoins, nous avons distingué deux groupes d'usages techniques. La situation des prisonniers du panoptique, par exemple, n'est pas enviable dans la mesure où ils sont pris dans un rapport de domination. Dans cette perspective, les techniques qui s'appliquent à eux sont des « techniques de gouvernement » . Ce sont des techniques qui prennent pour objet des conduites et participent à leur modification comme « de l'extérieur » du sujet même si, nous l'avons vu, les prisonniers intériorisent le regard des gardiens et les conduites qu'ils attendent d'eux. Nous avons ensuite identifié les « techniques de soi » . Ces techniques prennent aussi le sujet comme un objet de modification. Cependant, dans ce cas, le sujet se prend lui-même comme objet d'expérimentation et de transformation. La technique est toujours ausi un médium entre le sujet et lui-même et peut nécessiter un regard extérieur mais elle est ici mobilisée dans le cadre d'une action volontaire. Nous parlons ici de « groupes d'usages de techniques » car les techniques de gouvernement et les techniques de soi reposent potentiellement sur les mêmes outils. Cependant, le sens qui est donné à ces outils change radicalement d'un groupe à l'autre. C'est le déplacement du sens de ces outils d'un groupe à l'autre qui permet à la visibilité d'être soit un « piège » soit un outil de construction de soi.

Par conséquent, nous avons délaissé une évaluation des techniques du point de vue d'un ensemble de droits formels dont elles devraient permettre le respect. Nous lui avons préféré une analyse qui demande quelle structure d'action une technique nous impose et quels en sont les effets. Nous avons, pour cela, choisi de placer notre démarche àa la suite de l'analyse foucaldienne du souci de soi qui nous semble constituer une forme d'éthique de la vertu, entendue à la fois comme excellence et, principalement, comme capacité à remettre les normes en cause.

Néanmoins, les analyses foucaldiennes doivent être mises à jour. Foucault reconnaît en effet ne pas avoir directement appliqué sa conception du souci de soi à

113. Voir, sur ce point, « Un système fini face à une demande infinei » . Voir aussi « Bio-histoire et bio-politique » , Voir [Foucault 2001][pp. 95-97]. Dans ce texte, Foucault identifie les identités comme des variations au sein d'une population et nomme ces variations « nuage » , terme que l'on retrouve dans son texte sur la sécurité sociale. Il s'agit ainsi d'envisager la notion statistique de population comme le point de départ d'un « polymorphisme » des identités. Cette interprétation est subversive. Alors que *Sécurité, territoire et population* établit que la population permet la normalisation, en opposant la notion de population à celle de race, c'est-à-dire un ensemble de variations, la race perd sa prétention à l'objectivité.

des problématiques contemporaines [114]. Certes avoue-t-il que son travail a toujours pour point de départ une expérience contemporaine – tel est le cas, par exemple du Groupe d'Intervention sur les Prisons ou de son travail comme stagiaire dans un hôpital psychiatrique [115]. Son approche associe une préoccupation contemporaine à un regard critique sur des dispositifs passés susceptibles d'apporter des éléments de réponse stratégique mais qui, pour être efficaces, doivent être transformés. La lecture de Platon et des Stoïciens à la recherche d'une définition du « souci de soi » et des techniques sur lesquels il repose est ainsi opérée par un homme qui interroge le rapport à la sexualité et sa place sociale afin d'en proposer une autre forme [116]. Cependant, Foucault n'applique pas sa démarche à l'essor de l'informatique. Dans la mesure où elles mettent en jeu des rapports de force spécifiques fondés notamment sur une économie particulière de la visibilité, il nous semble que les techniques de communication offrent un terrain propice à la mise à jour des analyses de Foucault sur le souci de soi qui a été relativement peu exploré jusqu'ici.

114. Voir « Une esthétique de l'existence » [Foucault 2001][p. 1549].
115. Voir « Vérité, pouvoir et soi » [Foucault 2001][p. 1596].
116. Foucault lit ainsi la préoccupation pour le souci de soi dans l'*Alcibiade* et dans le *Manuel*. Voir « Les techniques de soi » , [Foucault 2001][pp. 1609 *et sq.*]

Chapitre 4

Géographie du virtuel

Aujourd'hui, la médiation des techniques, dans nos rapports sociaux notamment, est au centre de nombreuses controverses. Les craintes qui entourent la protection de la visibilité, de l'anonymat ou bien la possibilité d'effacer ses données en sont des exemples [1]. De fait, notre visibilité aux autres et donc la « surface de prise » que nous exposons à des techniques de gouvernement semble s'être accrue avec l'essor des outils informatiques et de leurs usages en réseau. Nous nous interrogerons ici sur les modalités de cette visibilité afin d'étudier ses effets sur notre subjectivation. Nous nous concentrerons particulièrement dans cette étude sur les outils de fouille de données et leur utilisation ainsi que sur les usages de l'informatique dans le cadre des réseaux numériques.

Notre étude sera guidée par trois questions que nous tirons de notre analyse de l'éthique du souci de soi : Qu'est-ce qui est rendu visible ? Qui peut accéder à quoi ? Quels sont les effets de cette visibilité quant à nos actions ? Répondre à ces questions nous permettra d'identifier le gouvernement des conduites à partir de la collecte et du traitement des données des individus. Nous montrerons que ce traitement conduit à envisager les individus comme des êtres déterminés par leurs relations et leur environnement. C'est en cela que la subjectivité numérique est « réticulaire » et programmée. L'identité numérique se construit non seulement grâce à l'activité de celui qu'elle concerne, mais aussi par agrégation de données dispersées sur le réseau. Elle constitue le point de départ de techniques de personnalisation et d'incitation souvent issues du marketing.

Nous revenons ainsi à la question que notre détour par la pensée d'Ellul nous avait conduit à poser : un monde dans lequel nous sommes surveillés, qui s'adapte à nous et nous influence, souvent à notre insu, est-il acceptable ? Nous posons cette question dans la perspective foucaldienne des techniques de soi afin de montrer comment, dans un tel monde, des pratiques de liberté sont tout de même possibles.

1. Nous pensons ici notamment aux différents mouvements de contestation dont les réseaux sociaux et plus généralement les fournisseurs d'outils numériques sont la cible. Les plaintes contre *Facebook*, en Europe, par exemple, se sont multipliées. La photographie des rues par *Google* afin de les utiliser dans le cadre de son système de cartographie a, elle aussi, provoqué de nombreuses réactions.

I Géographie du virtuel : le mode de présence spécifique des techniques numériques

A L'espace numérique, un espace modulé

Le terme de « virtuel » est souvent utilisé pour désigner ce qui a lieu sur les réseaux de télécommunication informatique. En l'espèce, le terme ne peut plus désigner ce qui n'existe qu'en puissance [2] et l'essor des véhicules motorisés facilitent encore le déplacement et la communication [4]. L'espace est ainsi en quelque sorte « condensé » puisqu'il faut de moins en moins de temps pour le parcourir. L'essor de l'informatique accentue encore ce mouvement. Par rapport à l'espace physique, l'espace numérique se contracte car le transport, c'est-à-dire le déplacement physique, est remplacé par la recopie des données. En effet les ordinateurs communiquent en s'échangeant des paquets de *bits* qu'ils recopient avant de les transmettre de nouveau. Deux interlocuteurs peuvent dès lors être séparés par une grande distance sans nuire à leur communication. En effet, leur représentation numérique peut être transmise par copies successives en faisant abstraction des distances parcourues. Par conséquent, l'espace entre eux est modulé comme le montre la démultiplication des formes de présence. Il faut désormais distinguer la présence physique de l'interlocuteur de sa présence virtuelle, ou numérique. Certes, le courrier permettait déjà un contact sans présence mais dans le cas des communications électroniques, il n'y a pas de délai entre les interventions des interlocuteurs, d'où une plus grande ressem-

2. Dans *Différence et répétition*, Gilles Deleuze reprend ainsi la tradition scolastique en soulignant que le virtuel n'a pas d'existence tangible, par opposition à ce qui est actuel. Voir [Deleuze 2010, Deleuze 1977]. Un achat dans une boutique en ligne, par exemple, entraîne des effets réels comme le débit d'un compte bancaire ou bien la livraison d'une commande. Dans le cadre spécifique de l'informatique, le terme « virtuel » signifie que les contenus accessibles – photographies, textes ou vidéogrammes par exemple – ou bien les interactions possibles sont numérisés. Cet adjectif insiste sur le fait que ce n'est pas le support sur lequel est stocké le contenu qui compte mais le découpage du contenu. Les contenus sont échantillonnés – c'est-à-dire divisés en unités atomiques – qui sont quantifiées et stockées, par exemple, sous la forme d'une empreinte sur un disque dur ou bien sous la forme d'un signal électrique. Ces éléments atomiques sont nommés « données » . Ces dernières sont par nature intégrables à un calcul et peuvent donner lieu à une multiplicité d'interprétations, c'est-à-dire de représentations en fonction, par exemple, du terminal de l'utilisateur qui souhaite y accéder. Elles peuvent être déplacées, répliquées ou consultées par plusieurs personnes à la fois sans pour autant être altérées ou disparaître. Les outils numériques offrent ainsi la possibilité de voir le monde comme un espace facile à répliquer et à archiver. Il faut ajouter que ce monde est facile à parcourir puisqu'on navigue aisément d'un site Internet, ou d'une application, à une autre.

La promotion d'une conception spécifique de l'espace semble être une constante des techniques de communication. La communication épistolaire, par exemple, permet d'échanger malgré la distance. L'invention du télégraphe au milieu du XIX^{e} siècle [3]Dans les années 1990 l'accès à l'Internet se répand, tout comme l'offre de sites commerciaux etc. Enfin, en 2007 le téléphone mobile iPhone, qui permet de se connecter à l'Internet en situation de mobilité connaît un grand succès. Aujourd'hui, ce type d'appareil est extrêmement répandu.

4. Les juristes Warren et Brandeis expriment ainsi les angoisses du XIX^{e} siècle face au développement des communications. Patrice Flichy, Paul Virilio ou Gilles Deleuze soulignent l'effets de la rapidité des télécommunications sur la perception et la structure de l'espace. Voir [Flichy 1997, Virilio 1977].

blance avec la vie hors ligne. Le téléphone permet, lui, de communiquer au travers de la voix et rend possible l'interactivité des échanges. Avec leur miniaturisation, il est de plus possible de converser avec à peu près n'importe qui et de n'importe où. L'essor des outils numériques permet, contrairement au téléphone ou au courrier, qui ne reposent que sur une modalité de communication et un nombre restreint de temporalités, de varier, à partir d'un même outil, les modalités de communication. Ainsi est-il possible de faire varier facilement le nombre de destinataires auxquels on s'adresse et de s'exprimer au travers de textes, d'images, de sons ou de vidéos ou bien d'associer ces modalités. Il est aussi possible de communiquer en temps réel ou bien de manière asynchrone. Nous avons déjà défini l'ordinateur comme une « machine à tout faire » , ici il apparaît comme une machine à moduler l'espace – nous sommes connectés aux autres en permanence et sans tenir compte de la distance physique qui nous sépare – et les modalités de présence.

Cette modulation s'appuie notamment sur la miniaturisation des outils informatiques et leur utilisation dans le cas des réseaux informatiques dont le mieux connu est *Internet*, l'abréviation de l'expression « réseau de réseaux » qui désigne la connexion de réseaux informatiques hétérogènes et dispersés. Le premier réseau de réseaux a été construit en 1969 par la mise en communication de quatre ordinateurs dans quatre universités américaines[5]. Le nombre d'ordinateurs connectés entre eux n'a depuis cessé d'augmenter. Le rôle de l'Internet s'est aussi élargi au grè de la multiplication des usages dont il est susceptible. Petit à petit, l'Internet est devenu un réseau où il est possible de se donner rendez-vous, de réaliser des achats ou encore de faire des rencontres. L'Internet est ainsi un espace métaphorique – il n'est finalement physiquement présent nul part – constitué de lieux principalement définis par leur fonction[6]. Ces « lieux » sont aussi bien les « sites » Internet que les forums de discussion ou les *tchats.* Du point de vue géographique, un lieu est en effet un endroit que l'on peut remplir de présence. Dès lors, le lieu a une définition et un intérêt social : il est là où on peut se rendre et interagir avec des objets ou des personnes. Un lieu tire sa fonction de deux composantes. Tout d'abord, un système de localisation qui permet de le désigner et de s'y rendre. Ensuite, un « investissement symbolique » , c'est-à-dire une certaine lecture du lieu par une personne en fonction de l'utilisation qu'elle souhaite en faire[7]. Un lieu est situé par rapport à

5. L'Internet, qui est initialement un projet de la *Defense Advanced Research Project Agency* (DARPA) américain, tire partie d'un ensemble de projets et de techniques pré-existants. Bien sûr, l'ordinateur, dont la première version familiale naît en 1977, joue un rôle crucial. De même, le modulateur-démodulateur, créé dans les laboratoires Bell en 1958, est un élément essentiel du réseau. Les efforts de standardisation des communications entre machines aboutissent en 1977 à la création du *Transport and Control Protocol/Internet Protocol (TCP/IP)* qui permet d'unifier petit à petit la diversité des réseaux existants et des techniques de communication qu'ils utilisent. Alors que seulement 1000 ordinateurs sont connectés en 1984, ce sont 10 000 000 machines qui sont en réseau en 1996. Aujourd'hui, plusieurs milliards d'objets communicants sont connectés à l'Internet.

6. Le géographe Baptiste Beaude désigne l'Internet comme un lieu qu'il faut prendre au sérieux dans la mesure où il permet des interactions qui constituent aujourd'hui une partie importante de notre vie. Dans la mesure où l'Internet est un réseau informatique, Beaude qualifie ce nouveau lieu de « réticulaire » . Voir notamment [Beaude 2011].

7. [de Bussher 1998]montre ainsi que les lieux publics peuvent être détournés comme des lieux

d'autres lieux qui sont à une certaine distance de lui, potentiellement couverte par un moyen de transport. Considérer certains espaces numériques au moins comme des lieux fournit par conséquent un ensemble de critères pour évaluer la spécificité de l'espace numérique et notamment la portée de la modulation spatiale qu'il introduit.

Le passage entre les lieux numériques ne se fait pas par rupture mais par une navigation très simple puisque les sites sont reliés par des liens hypertextes. De plus, les sites s'associent souvent pour fonctionner ensemble. Le lieu numérique est ainsi de forme réticulaire. On ne sort ni n'entre jamais vraiment nulle part dans la mesure où être sur un site, c'est déjà être sur une page composée d'une myriade d'autres pages qui renvoient toutes à d'autres pages. Même si les barrières entre les langues subsistent, il faut aussi noter que ce lieu réticulaire est un « monde unique » abstrait des frontières géographiques puisqu'une société peut facilement offrir ses services à l'étranger tout comme les données peuvent aisément être stockées dans un pays pour être utilisées dans un autre [8]. Enfin, le monde numérique fait aujourd'hui partie intégrante de notre monde vécu. Transporter avec soi un *smartphone*, c'est en effet voyager avec un ordinateur portable qui guide nos trajets et nous permet de nous connecter à nos services numériques préférés.

Plutôt que d'opposer à une « vie réelle » une « vie virtuelle » , il faut ainsi se représenter le passage entre la vie « en ligne » et « hors ligne » sous la forme d'une modulation des supports des interactions sans porter atteinte à l'actualité de ces dernières : utilisateurs de téléphones mobiles connectés en permanence à l'Internet ou bien travailleurs sur un lieu équipé d'une connexion au réseau et d'une ligne téléphonique, peu de gens sont aujourd'hui « déconnectés » et séparés les uns des autres. Dans la mesure où la possibilité d'être ensemble ou bien d'agir ne tient plus ni à la localisation ni à la matérialité d'un support particulier, on peut ainsi se retrouver et agir dans ce qu'il faut appeler, de manière métaphorique des lieux numériques.

Avec la disparition des contraintes imposées par la communication physique, la notion de frontière géographique perd de sa consistance. La structure physique du réseau – les liaisons qui unissent les ordinateurs et les différentes réseaux entre eux – et sa structure logique – c'est-à-dire sa représentation du point de vue de la circulation des données – sont distinctes. Si physiquement, deux machines sont distantes de plusieurs kilomètres et ne sont pas reliées directement entre elles, sur le plan logique, le réseau se charge d'établir une connexion entre elles [9]. La distance qui sépare deux informations ou deux utilisateurs est ainsi le nombre de points qu'il faut parcourir pour les relier mais n'a que peu d'intérêt puisque les machines intermédiaires se contentent de copier les informations. De ce point de vue, la connexion entre les machines importe peu tant qu'elle fonctionne : les frontières physiques sont trans-

de sexualité.

8. Voir http ://www.Facebook.com/principles.php. Site Internet consulté le 16 avril 2012.

9. L'internet est gouverné par deux principes, celui de la communication point à point, c'est-à-dire la communication entre les deux points de la liaison. Les éléments du réseau se chargent de trouver un chemin entre les deux points. Le second principe est celui du *dumb network* : le réseau achemine toutes les données sans considération de ce qu'elles représentent ni ordre de priorité.

parentes [10]. Par exemple, lorsque j'utilise un moteur de recherche, le résultat de ma requête me parvient rapidement alors qu'il a nécessité l'utilisation de nombreuses machines réparties sur le globe.

La modulation physique devient de plus en plus prégnante à partir de la fin des années 1990 avec la démocratisation des accès à l'Internet puis, plus récemment, le développement des accès mobiles à l'Internet qui permettent d'être connecté en permanence. Elle prend pour support l'usage le plus répandu de l'Internet, le *web*, créé en 1989 par Tim Berners-Lee, qui invente le premier système hypertexte complétement distribué sur le réseau. Le web est un ensemble de fichiers – nommés « pages » – reliés par des « hyperliens » qui permettent de naviguer entre eux. La mise en place du *web*, initialement destiné aux chercheurs et aux militaires, est guidée par l'idée d'une bibliothèque de connaissances tout à la fois vaste et facile à parcourir. Aujourd'hui, l'un des usages les plus répandus du web est la socialité numérique, qui se déploie sur les réseaux sociaux numériques. Le plus célèbre d'entre eux, le site *Facebook,* est né en 2004. Ces réseaux numériques ont pour but de relier leurs utilisateurs. Ils prennent appui sur la structure du *web* pour interpréter les liens comme des relations sociales : chaque utilisateur possède une page qu'il peut éditer. La connexion entre deux utilisateurs est matérialisée grâce à un lien dont la signification repose à la fois sur les relations préexistantes à la vie numérique, comme l'amitié ou les relations professionnelles, mais aussi sur le sens et la portée spécifiques que le réseau numérique donne à ce lien.

La relation d'amitié, vue par Facebook, est ainsi un exemple de redéfinition d'une catégorie sociale au travers d'un usage numérique. Hors ligne, l'amitié est une relation définie par la loyauté que l'on doit à l'autre et le partage d'éléments intimes [11]. Sa version en ligne, elle, confond des relations variées entre des individus qui, hors ligne peuvent être amis mais aussi membres d'une même famille ou de simples connaissances.

Les techniques de communication participent ainsi à la structuration de notre espace social, ce dont nous pouvons nous assurer en comparant trois situations. Tout d'abord, imaginons des amis qui doivent vivre en permanence sous le regard d'étrangers avec qui ils ne sont pas amis. Prenons ensuite les mêmes amis, dans la même situation mais à qui un téléphone portable est donné. Enfin, imaginons que le téléphone portable permet d'enregistrer tous les échanges des amis.

La loyauté et le partage d'une forme d'intimité qu'exige l'amitié doivent être cultivées pour perdurer. Le regard permanent d'étrangers perturbe les échanges des amis et, à force leur amitié est vouée à disparaître. Dans la seconde situation,

10. L'absence apparente de prise en compte de la géographie sur Internet ne peut cependant faire oublier que l'accès à l'Internet est beaucoup plus répandu et facile dans les pays développés – qui possèdent les infrastructures nécessaires – que dans les pays en voie de développement dont le nombre d'utilisateurs du réseau reste faible.

11. Notre réflexion prend ici pour point de départ l'analyse de le vie privée par James Rachels. L'auteur soutient que la vie privée est nécessaire au maintien de relations privilégiées qui reposent le partage de l'intimité et affirme que les technologies contemporaines remettent en cause cette possibilité. Voir [Rachels 1975]. Rachels traite principalement de l'amitié, mais il évoque aussi l'exemple des relations amoureuses.

l'usage du téléphone portable permet de contourner cette difficulté. En instaurant un espace de communication instantanée au nombre de participants restreint, la téléphonie mobile permet en effet de creuser dans l'espace physique et social qui les entoure un espace de visibilité limitée et qui est en ce sens « privé » .

Dans le troisième cas, celui de l'amitié entretenue au travers de la messagerie instantanée, la relation est protégée car les utilisateurs peuvent cumuler la pratique de plusieurs rôles sociaux – celui d'ami et d'individu dans une foule avec laquelle on ne partage pas de relation d'amitié – grâce à la technologie. Facebook le montre bien, l'informatique et les réseaux permettent l'entrecroisement de plusieurs réseaux sociaux. Les rôles que ces derniers mobilisent se superposent donc. Un rôle social est notamment défini par un ensemble de prémisses comportementales et décisionnelles qu'il faut mettre en œuvre pour être admis dans un groupe [12]. Par exemple, le rôle d'ouvrier me place dans une économie de gestes et de relations hiérarchiques que je dois respecter sous peine d'être renvoyé. De même, mon rôle de père associe un ensemble de comportements attendus à un ensemble de comportements que j'ai intériorisés. Hors ligne, je ne mets pas nécessairement en acte tous les rôles sociaux que je possède à chaque instant. L'acteur qui souhaite endosser un rôle doit, pour le faire correctement, se trouver dans des circonstances ou un lieu particuliers. Je ne suis ainsi effectivement ouvrier qu'à l'usine. Au contraire, dans une situation de travail où l'informatique et l'Internet sont mis en jeu, mon ordinateur peut servir simultanément à traiter un dossier professionnel, communiquer avec mes proches et réaliser un achat personnel. Je suis, virtuellement au moins, travailleur, parent et client d'un magasin à la fois. Chaque rôle mobilise des formes de comportement qui s'appuient sur des prémisses décisionnelles et comportementales qui définissent ce qu'il est convenable de faire. Dans la mesure où les rôles se superposent, les comportements qui les définissent et leurs effets s'entrecroisent [13]. De

La troisième situation, en permettant la coexistence de rôles disjoints permet ainsi d'ouvrir de nouveaux espaces pour nos pratiques sociales. Cette situation inédite, qui est la nôtre, montre la modulation de l'espace social qui découle de la modulation spatiale. La visibilité des activités et des biens attachés à ces rôles, est ainsi remise en question par cette nouvelle distribution. Mes rôles sociaux sont toujours plus ou moins actifs mais ne sont jamais complètement absents, ce qui présente les vies individuelles comme des *continua* et non pas des successions de rôles sociaux convoqués par des situations sociales spécifiques et séparées. Quand je suis au travail, je peux ainsi envoyer un message textuel à mon compagnon et une fois rentré chez moi je pourrai vérifier mes courriels professionnels. L'enregistrement de toutes les conversations par messagerie instantanée, qui était notre troisième possibilité dans l'étude de la communication entre amis est aujourd'hui effectif. Toutes les ac-

12. Voir [Simon 1991]

13. Dans « Life Scheduling to Support Multiple Social Roles » , A. Grimes et A.J. Brush montrent ainsi que les utilisateurs d'un calendrier électronique, accessible à la fois chez eux et au travail sonts condui à noter sur un même document leurs rendez-vous professionnels et personnels. Dès lors, leurs collègues de travail sont potentiellement mis au courant du contenu de leur vie extra-professionnelle et les utilisateurs font état de leur inconfort face à cette situation. Voir [Grimes 2008].

tions numériques se déroulent sous l'œil attentif d'un nouvel acteur, le fournisseur de services numériques – une entreprise comme *Google* ou *Facebook* par exemple ou bien un opérateur de téléphonie mobile. Lorsque deux amis utilisent un système de messagerie instantanée afin de communiquer sans que les personnes qui pourraient les voir ne le sachent, leur conversation n'est pas « privée » au sens où elle ne serait accessible à personne d'autre qu'aux locuteurs. La conversation est enregistrée par le fournisseur de services en permanence.

B L'espace numérique, un espace surveillé et programmé

Alors que je passe dans la rue, les autres passants ne voient qu'une partie de mon chemin. Au contraire, dans la mesure où les réseaux d'observations à la disposition des sociétés d'Internet épousent la forme de ce dernier, l'observation ne cesse jamais. L'outil *Google Dashboard*[14], par exemple, présente toutes les données que *Google* détient sur un utilisateur de ses services. Il montre tout d'abord que grâce à la baisse du prix des mémoires informatiques et au perfectionnement des techniques de stockage, toute donnée peut aujourd'hui être conservée indéfiniment. Ensuite, l'outil rend évident que toutes les données peuvent être facilement recoupées et associées. La visibilité des conduites change ainsi d'échelle à cause de l'enregistrement constant des données et des nouvelles capacités de traitement de ces dernières. Les services offerts par *Google* étant très variés, l'entreprise possède non seulement les courriels de ses utilisateurs, mais aussi une partie non négligeable de la cartographie de leur environnement social, ainsi que de leurs intérêts. Alors que je suis dans la rue, je peux savoir plutôt aisément si je suis observé dans la mesure où je vois si beaucoup de gens m'entourent et je peux voir les dispositifs d'observation comme les caméras. Au contraire, l'interface du moteur de recherche, extrêmement épurée, ne manifeste pas la présence des autres utilisateurs alors que mes actions peuvent être visibles sur le réseau social de la société. En effet, tout utilisateur des services de Google se voit octroyer un profil sur le réseau social *Google +*, sur lequel les activités de l'utilisateur peuvent apparaître. Mon écran dissimule ainsi en quelque sorte la collecte de données par Google ainsi que leur traitement.

Les données collectées ne sont en effet que des données primaires qui, sur l'Internet, constituent les traces que laissent nos interactions avec un système informatique. *Dashboard* montre que Google déduit, à partir de ces données primaires, un ensemble de données dérivées comme l'âge, le sexe, la localisation et les intérêts des utilisateurs[15]. Une telle déduction n'est pas nouvelle. Hume, par exemple, déduit à partir d'écrits antiques qui font état de l'existence de neige à Rome en hiver dans l'Antiquité l'augmentation de la température à Rome au XIX^{e} siècle. Les écrits antiques constituent ainsi pour Hume des données primaires et l'hypothèse d'un réchauffement du climat romain une donnée dérivée[16]. Néanmoins, l'éventail des

14. L'outil est accessible à l'adresse https ://www.Google.com/dashboard/. Site consulté le 19 mars 2012.

15. *Google* précise les données dérivées et les fondements de cette dérivation sur http ://support.Google.com/adsense/bin/answer.py ?hl=en&answer=140378. Site consulté le 26 mars 2012.

16. Voir [Hume 1985]

données stockées par *Google* et les capacités de traitement dont l'entreprise dispose augmentent le nombre de données qui peuvent être dérivées [17].

Dans de nombreux cas, la collecte et le traitement des données primaires sont présentés comme des outils pour construire des applications informatiques plus utiles ou plus sécurisantes [18]. Ce surcroît de confort ou de sécurité est octroyé par l'application de techniques mathématiques. La nature numérique des données en permet de les intégrer facilement à un calcul. Plus largement, le *web* est un objet facilement représentable mathématiquement [19]. Il se présente comme un graphe dont les noeuds sont les différentes pages. Les arcs du graphe sont les liens hypertextes. L'importance des liens est différentielle dans la mesure où elle repose notamment sur le nombre de liens qu'un nœud possède avec les autres. Les algorithmes de recherche de l'information, le plus célèbre étant le *pagerank*, utilisé par *Google*, prennent en compte les relations des pages entre elles [20]. Afin de les parfaire, la signification du contenu des pages et des liens entre elles sont envisagés comme des outils pour améliorer la performance des recherches. Il s'agit d'en découvrir les liens et de les modéliser afin d'en trouver le sens [21].

Dans la mesure où les utilisateurs des réseaux sociaux numériques sont représentés et identifiés par des pages web, une démarche similaire est appliquée aux communautés numériques. Ainsi le parcours des liens entres les différents profils permet-il de constituer un sociogramme qui éclaire les relations entre les utilisateurs [22]. Néanmoins, le sociogramme formé par le parcours des pages web dépasse la représentation des relations sociales. *Facebook,* par exemple, cartographie non seulement l'ensemble des relations sociales d'un individu mais aussi ses centres d'intérêt, les objets communicants qu'il utilise ainsi que ses achats en ligne. Cette nouvelle cartographie est nommée *OpenGraph* [23] et représente les liens entre les personnes physiques, les personnes morales et les objets physiques ou numériques. Grâce à la prolifération des appareils mobiles qui permettent de se localiser et d'indiquer sa position lors de l'interaction avec un réseau numérique, et grâce à la collaboration de

17. Aujourd'hui, les requêtes adressées au moteur de recherche de Google constituent une mine de données primaires dont n'importe quoi peut être dérivé, de l'âge de la personne qui émet la requête à la propagation d'une épidémie.

18. Tel est le cas des systèmes de recommandation sur les sites marchands par exemple, ou bien de requêtes dont les résultats dépendent de l'utilisateur.

19. Voir par exemple Floriana Gargiulo et S. Huet, « Opinion dynamics on a group structured adaptive network » , 2009.

20. Le *pagerank* est un algorithme qui donne à chaque page d'un ensemble de pages reliées par des liens hypertextes un poids qui exprime son importance relativement aux autres pages. L'algorithme est breveté : U.S. Patent 6,285,999 et utilisé exclusivement par *Google*.

21. On nomme « web sémantique » l'entreprise qui consiste à ajouter à chaque donnée disponible sur Internet des éléments de description de son sens. Ces descriptions peuvent ensuite être utilisées par les programmes informatiques afin de prendre en compte le sens des données qu'ils manipulent.

22. Un sociogramme est le diagramme des relations sociales qu'entretient un individu. Les sociogrammes sont élaborés par Jacob Levy Moreno comme outil pour la sociométrie, l'analyse quantitative des réseaux sociaux. Voir [Moreno 1970]

23. La définition de l'OpenGraph par Facebook est disponible à l'adresse (consultée le 6 avril 2012) http ://developers.Facebook.com/docs/opengraph/

Facebook avec de nombreuses autres entreprises [24], l'*OpenGraph* s'étend ainsi à l'espace physique et à toutes les dimensions de la vie d'un utilisateur. Il offre une vision systémique des relations entre les utilisateurs et les objets, qui sont tous pris dans un même graphe. Dans la mesure où la société *Facebook* dispose de grands volumes d'informations sur les utilisateurs, elle connaît l'évolution de leurs relations sociales et de leurs goûts. Ces évolutions sont archivées et comparées avec celles des autres utilisateurs afin d'en tirer des profils d'utilisateurs qui partagent des traits comportementaux et des modèles statistiques qui permettent de prédire les comportements d'un utilisateur.

Le projet *Predestination* de Microsoft, qui n'est pas directement lié à la sociabilité numérique mais qui montre bien l'utilisation des traces, est un exemple de construction et d'utilisation d'un modèle de comportement à l'aide des statistiques. Ce projet montre que l'intérêt du sociogramme repose dans la modélisation des forces qui parcourent les réseaux sociaux et des règles qui les gouvernent. Dans le cas de *Predestination*, la force considérée est le déplacement des conducteurs des véhicules. Le nom du projet est une métaphore. *Predestination* est un appareil GPS [25] qui émet des recommandations en temps réel quant à la destination de l'utilisateur et au meilleur moyen d'y parvenir. Alors que lorsqu'ils utilisent un GPS traditionnel les utilisateurs saisissent eux-mêmes leur destination, *Predestination* suggère en temps réel aux utilisateurs le meilleur chemin pour parvenir à la destination que l'appareil devine. Cette approche repose sur deux présupposés. Tout d'abord, le manque de connaissances des conducteurs, qui ne savent pas nécessairement comment atteindre leur destination de manière optimale [26]. Ensuite, les concepteurs supposent l'existence de règles qui gouvernent la conduite des utilisateurs et qui peuvent être inférées à partir de l'observation des conducteurs. Des profils types de conducteurs peuvent ainsi être constitués afin d'améliorer la pertinence des recommandations qui sont faites à un groupe d'utilisateurs partageant les mêmes caractéristiques.

L'appareil auquel a donné naissance *Predestination* repose sur l'utilisation d'une carte géographique représentée sous la forme d'un réseau de probabilités. Chaque point de la carte est lié aux autres par un chemin pondéré par une probabilité. À chaque étape du parcours, l'appareil possède ainsi plusieurs représentations potentielles du chemin suivi. La progression sur ce chemin entraîne l'abandon d'une partie des destinations possibles jusqu'à la détermination de la destination du conducteur. Les probabilités de la carte proviennent de l'analyse d'un jeu de données variées. Les parcours d'un grand nombre de conducteurs – analysés non seulement en fonction des déplacements géographiques mais aussi en fonction des rythmes des semaines et des années – les calendriers des vacances scolaires et les données géographiques sont croisés afin d'obtenir une carte qui montre que les trajets individuels sont en fait des

24. Les grandes entreprises de l'Internet proposent ainsi des services aux propriétaires de sites web qui permettent d'analyser l'activité qui s'y déroule. *Facebook* propose aussi le système des *likebox* qui permet de rendre visible sur le réseau social les activités réalisées sur d'autres sites.

25. L'acronyme GPS signifie *Global Positionning System* (traduit par Système de Positionnement Global). Le système de GPS permet de localiser tout utilisateur équipé d'un émetteur-récepteur sur n'importe quel point de la planète.

26. Voir [Krumm 2007]

tropismes. Ils sont conformes à des régularités et influencés par des facteurs environnementaux. Cette conclusion justifie dès lors l'automatisation des recommandations émises par l'appareil.

Predestination, comme d'autres outils numériques, repose sur la division des données selon leur caractère primaire ou dérivé. Les données primaires sont issues de l'activité des utilisateurs. Elles ont une valeur potentielle en tant que matériaux à interpréter, ici au moyen de statistiques et d'inférences probabilistes. L'utilisation des données primaires pour produire les données secondaires et l'utilisation des données dans un but projectif montrent que ces dernières sont considérées comme des signes : nos actes et nos paroles expriment ce que nous sommes et constituent des moyens d'accès à l'individu qui en est la source [27]. Dans cette perspective, les données sont « objectives » car elles proviennent de l'utilisateur lui-même et relatent de manière exhaustive ses actions. Au cœur des sociogrammes étendus peuvent se lire l'identité de l'individu et les règles qui le gouvernent [28]. La représentation temporelle des évolutions du graphe permet en effet d'observer la formation et la modification des nœuds et des arcs mais aussi de tenter de prévoir les états suivants du graphe [29].

Nous pourrions multiplier les exemples de techniques similaires à celles sur lesquelles repose *Predestination.* De la suggestion de la musique en fonction de l'humeur à la publicité ciblée, les techniques comme les statistiques sont de plus en plus utilisées de manière prédictive. Après la catégorie de l'espace, c'est donc celle du temps dont la représentation et la perception est modifiée. Tout d'abord, le temps est accéléré dans la mesure où les individus sont sollicités en permanence, par des messages ou des courriels par exemple. Le laps de temps qui est laissé à la réaction est, lui aussi, réduit puisqu'il faut répondre ou publier le plus rapidement possible [30]. Le sentiment de l'urgence est par conséquent généralisé [31]. Ensuite, le temps est rendu homogène. Il est régulé par des cycles comme l'évolution du trajet des automobilistes, ou bien celui du bonheur des populations [32]. Le temps est un mouvement gouverné par un principe de régularité : l'exemple couronné de succès de *Predestination* montre que les statistiques alimentent la croyance en l'existence de telles régularités. Les statistiques ne sont pas, en effet, qu'un calcul. Au sens étymologique du terme, elles désignent le dénombrement des biens de l'État. Les statistiques sont

27. « Il faut que le donné fonctionne comme signe pour qu'il soit seulement donné. » Voir [Levinas 1961][p.65].

28. Voir par exemple [Nath 2006]. Les auteurs utilisent un grand volume de données éparses pour déterminer les causes des comportements criminels.

29. La fouille des réseaux sociaux et la prédiction de leur évolution est ainsi l'objet principal du projet Filtrage, Recherche et Annotations dans des Graphes d'Interaction Sociaux (FRAGRANCE) soutenu par l'Agence Nationale de la Recherche française.

30. Emmanuel Kessous insiste ainsi sur les effets de la structure des outils numériques sur la quête de reconnaissance des utilisateurs. L'importance accordée au nombre de visites des pages dans les classements qui gouvernent leur présentation, par exemple, oblige à publier fréquemment des contenus attractifs. Voir [Kessous 2013][p. 67].

31. [Jauréguiberry 1997]

32. À l'adresse http ://apps.Facebook.com/usa_gnh/ *Facebook* fournit ainsi un « indice de bonheur brut » qui représente la valeur plus ou moins positive des mots postés par les utilisateurs sur le réseau.

ainsi une forme particulière d'interprétation. La présupposition de l'existence des régularités qu'elles visent à découvrir motive la collecte et l'analyse des données [33].

Par conséquent, il faut replacer la collecte et le traitement des données dans une perspective stratégique qui est, le plus souvent, celle du *marketing* des traces, c'est-à-dire d'une analyse des données qui a un but commercial. La plupart des services numériques sont en effet gratuits uniquement parce que les données constituent des biens pour les sociétés qui les détiennent. Soit les données peuvent être vendues, soit elles peuvent être analysées pour saisir le comportement des consommateurs et optimiser les bénéfices [34]. La variété des informations qu'il est possible de tirer de l'analyse des données - du « bonheur brut » à la propagation d'une épidémie du point de vue d'un épidémiologiste [35] - ainsi que la possibilité d'utiliser les données apparemment les moins significatives pour extraire des informations font que toute donnée a une valeur potentielle [36]. Par conséquent, toutes les données peuvent être des « données opérationnelles » – c'est-à-dire qu'elles peuvent être mises en jeu dans des stratégies de détection ou d'influence des conduites – dans la mesure où elles peuvent permettre de trouver des régularités dans l'organisation du monde.

Dans la mesure où les données ont de la valeur comme unités qui peuvent être associées, cette saisie ne peut s'opérer qu'à partir d'un grand ensemble de données. La collecte des données est ainsi justifiée par la valeur des associations. La collecte, qui s'apparente à une forme d'observation, entraîne la redéfinition d'un ensemble de phénomènes de surveillance. Le premier point porte sur l'identité du surveillant, qui est traditionnellement une personne physique ou morale possédant un degré élevé d'autorité hiérarchique. Tel est le cas, par exemple, de la police ou d'un surveillant pénitentiaire [37]. Certes, les États restent des surveillants effectifs dans la mesure où ils accumulent des données sur leurs citoyens. De plus, les réseaux de surveillance traditionnels persistent. Tel est le cas des relations de surveillance entre un médecin et ses patients dans un hôpital, ou entre un contre-maître et ses ouvriers dans une usine et des réseaux de surveillance sur lesquels ces relations s'appuient. Néanmoins, l'essor des réseaux de communication entraîne deux changements. Tout d'abord, la surveillance opérée par les États devient elle-même réticulaire. Elle prend pour support, par exemple, les caméras de surveillance qui sont toutes connectées les unes aux autres ou bien l'observation des communications électroniques. Ensuite, de nouvelles instances de surveillance émergent.

Tout d'abord, les usages du web à des fins de sociabilité numérique et plus généralement la disponibilité des données facilite l'assouvissement de la curiosité

33. Voir Foucault, Michel, *Voir [Foucault 2004][p.280]* Les statistiques, « étymologiquement, c'est la connaissance de l'État, la connaissance des forces et des ressources qui caractérisent un État à un moment donné ».

34. Voir par exemple [van Wel 2004] *Google,*par exemple, tirent la plupart de leurs revenus des activités liées à la vente et au traitement des données comme la vente d'encart de publicités ciblées en fonction des utilisateurs.

35. Voir [Jeremy Ginsberg 2009]

36. Ainsi le croisement de l'historique de mes connexions WIFI avec la carte des points d'accès sans fil permet-il de découvrir où j'habite, où je travaille et, potentiellement, mes attirances sexuelles par exemple. Voir [Cheng 2013].

37. Foucault, *Surveiller et Punir.*

des utilisateurs qui nous surveillent ainsi potentiellement en permanence. Ensuite, il faut noter l'essor des entreprises qui collectent et traitent les données comme instances de surveillance. En France, l'importance de cet essor se lit dans les barrières que veulent lui opposer les textes qui encadrent la collecte et le traitement des données. La Commission Nationale Informatique et Libertés (CNIL) a ainsi pris place sur le fondement de la loi « Informatique etLlibertés » de 1978, suite au projet de Système Automatisés pour les Fichiers Administratifs et le Répertoire des Individus (SAFARI) de 1974. Dans la mesure où SAFARI émanait d'une volonté étatique de croiser les informations détenues sur les citoyens, la mission de la CNIL était prioritairement dédiée à la surveillance et à l'encadrement de la collecte et du traitement des données par les États. En 2004, la loi « informatique et liberté » est modifiée afin de prendre en compte les dangers que représente un nouveau groupe d'acteurs de collecte et de traitement des données, les entreprises [38]. Aujourd'hui, une grande partie des conflits autour de la collecte et du traitement des données met en jeu des entreprises qui manipulent des données comme *Google* ou *Facebook*.

C Les multiples facettes de l'identité numérique

On the Internet, nobody knows you're a dog pouvait-on affirmer avant l'essor du web 2.0 et de ses usages sociaux. Aujourd'hui, la protection de l'anonymat ou bien le contrôle de données sont devenues des thèmes régulièrement abordés dans la réflexion sur les outils numériques. Alors que les premiers usages du web reposaient sur l'usage de pseudonymes ou d'avatars, les nouveaux usages du web reposent en effet sur le partage de l'identité des utilisateurs dont les modalités sont influencées par la structure de l'espace numérique. Afin de délimiter les contours de l'identité numérique et ses enjeux, il faut se situer de deux points de vue. L'identité numérique est produite par un utilisateur, elle est aussi manipulée par les différents acteurs avec lesquels il est en relation.

Du point de vue des utilisateurs, l'identité numérique doit être définie de manière large comme toutes les traces qui le concernent et à la production desquelles il participe. Tout d'abord, il faut noter que, le plus souvent, cette identité est celle d'un sujet de droit. En effet, le partage de données et l'utilisation de services numériques repose sur le consentement des utilisateurs, recueilli par leur acceptation de contrats – les conditions d'utilisation. La constitution d'une identité numérique engendre ainsi la mise en cause de la responsabilité de l'utilisateur. La répétition de la collecte du consentement, demandé à chaque fois qu'un utilisateur peut voir ses données collectées montre que l'identité numérique est une identité dispersée – entre les sites Internet qui concernent un utilisateur, les fichiers dans lesquels il est enregistré ou encore les données des appareils qu'il utilise. Dans une certaine

38. La Loi d'Orientation et de Programmation pour la Sécurité Intérieure (LOSPI), qui porte sur les moyens qui peuvent être utilisés pour préserver la sécurité, porte ainsi la trace de la confrontation entre États et entreprises. Alors que la loi qui a créé la CNIL restreignait les privilèges de l'état Français en matière de collecte et de traitement de l'information, la LOPSI réaffirme le privilège en la matière de l'État par rapport aux entreprises. La LOPSI stipule notamment que toutes les données qui concernent la conduite des affaires sont réservées à l'usage des États.

mesure, l'identité numérique est une forme de représentation de soi qui peut-être qualifiée d'active, voire de stratégique. Ainsi un profil de réseau social présente-t-il les contenus partagés par un utilisateur. Dans cette perspective, l'identité numérique est entièrement constituée pour être observée. La plupart des sites fournissent des indicateurs de popularité d'un utilisateur et de ses messages – en comptant le nombre de visites sur sa page ou bien le nombre de réactions à un message. Dès lors, l'identité numérique est aussi une identité potentiellement faite pour être attrayante et susciter l'adhésion. La persistance des pratiques comme l'utilisation de pseudonymes peut enfin faire de l'identité numérique – conçue ici comme le récit qu'un utilisateur élabore sur lui-même – un récit ludique ou fictionnel.

Néanmoins, et les profils des utilisateurs sur les réseaux sociaux le montrent bien, l'identité numérique est aussi construite par l'ensemble des utilisateurs et des entités avec lesquelles un utilisateur entre en relation. Les utilisateurs qui publient sur mon profil, les sociétés qui observent mes activités – par exemple grâce à des cartes de fidélité – et déterminent mon profil de consommateur, tous participent à la constitution de données qui me concernent. Les autres utilisateurs jouent d'ailleurs un rôle important dans la constitution de mon identité puisqu'ils peuvent m'identifier dans des photographies ou des messages, par exemple, et influencer directement le récit de mes activités.

Par conséquent, l'identité numérique est, du point de vue d'un utilisateur, difficile à contrôler. La facilité d'accès aux sites web et la possibilité d'en recouper le contenu permet tout d'abord facilement d'obtenir une vue globale de mon identité, même si j'ai choisi de relater différents aspects de mon identité sur des sites différents. De plus, la diversité des plateformes rend complexe la mise à jour de toutes les facettes d'une même identité. Dans la mesure où ces plateformes échangent souvent entre elles des données et sont indexées par des moteurs de recherche qui conservent les données qu'ils classent, l'effacement des données est particulièrement complexe.

Par conséquent, les stratégies individuelles d'organisation de l'identité numérique entrent en conflit potentielles avec les stratégies des autres acteurs du réseau. Souvent, l'identité numérique est une identité sous surveillance, par exemple dans un but marketing ou sécuritaire. Les Etats ont ainsi de plus en plus souvent recours aux données issues de l'utilisation des outils numériques. Les opérateurs privés, eux, mettent en place de larges gammes de services. Les utilisateurs effectuent ainsi de nombreuses actions et génèrent de grands volumes de données diverses qui accroissent la connaissance dont les opérateurs numériques disposent à leur sujet. Les sociétés numériques peuvent aussi s'associer entre elles afin d'observér l'activité des utilisateurs sur des plateformes qui ne sont pas les leurs [39]. Alors que l'utilisateur

39. Nous pensons ici à l'outil « Likebox » que Facebook met à disposition des développeurs de site Internet ou bien aux *cookies*. Les Likebox sont des outils qui permettent aux propriétaires d'un site d'utiliser les données connues par Facebook sur un utilisateur (sa liste d'amis, son état civil, etc.) pour son site. En échange, Facebook collecte les interactions de l'utilisateur qui passent par la Likebox. Les *cookies* sont, eux, des fichiers que laissent les sites Internet sur les machines des internautes. Ces fichiers permettent non seulement d'accéder aux informations de l'utilisateur et de la machine qu'il utilise mais ils peuvent aussi être mis à jour au cours de la navigation sur l'Internet.

peut avoir l'impression de posséder une identité fragmentée, du point de vue de la plupart des acteurs numériques, ce n'est pas le cas. Les mécanismes d'inférence dont ces derniers disposent leur permettent d'ajouter aux données qu'ils connaissent déjà des données qu'ils conjecturent.

Le lieu réticulaire se présente comme un système parcouru par un ensemble de forces – envie d'achat, de déplacement, etc. – qui contient tous les objets du monde – personnes, biens, territoires etc. Un tel lieu ne connaît pas d'extérieur. Les forces qui le parcourent ne cessent jamais et leurs modulations relient tous les objets qu'il contient. L'unicité de ces objets, et avec elle la notion d'individualité, se trouvent mises à mal avec l'effacement de la dichotomie entre un intérieur et un extérieur. Sur le *web*, par exemple, elle laisse la place à la personnalisation [40]. La personnalisation consiste à adapter la présentation d'un contenu numérique en fonction de la personnalité de celui qui souhaite y accéder. Elle repose sur la détermination de cette personnalité et constitue ainsi une forme d'individuation analytique et extérieure à l'objet qu'elle individualise. La suggestion d'une destination par *Predestination* ou bien celle d'un achat par un site de commerce en ligne en sont deux exemples. La personnalisation repose sur trois étapes. Tout d'abord, un ensemble de données qui représente le comportement des utilisateurs est analysé afin de tirer des profils, c'est-à-dire des régularités, dans les actions. Ensuite, lorsqu'un utilisateur souhaite accéder à une application, il est analysé afin d'énumérer ses propriétés. Le logiciel peut dès lors déterminer la catégorie à laquelle l'utilisateur appartient et adapter le service auquel l'utilisateur souhaite accéder en fonction de ses caractéristiques et du profil d'utilisateur auquel elles renvoient. La personnalisation est ainsi une forme de catégorisation et de programmation de l'action des utilisateurs. C'est en cela qu'elle constitue une technique de gouvernement.

La constitution des données comme signes confère ainsi un rôle crucial à l'interprète des données dont le travail suit deux phases. La première, de collecte et d'analyse des données permet de tirer des « profils » , c'est-à-dire les caractéristiques de groupes d'utilisateurs. La seconde phase est celle de la reconnaissance du profil d'un utilisateur donné. Le rapport qu'entretient cet interprète au lieu réticulaire et aux créatures qui le peuplent est ainsi un rapport de connaissance [41], plus précisément de compréhension qui justifie la capillarité de la surveillance contemporaine. « Comprendre » , c'est contempler toutes les facettes de l'objet considéré. Au travers de la détermination des profils des utilisateurs, ce rapport de connaissance conduit à la détermination du sens des actions des individus en fonction de leur dépendance avec les autres éléments du système. Pris du point de vue de la connaissance, le lieu réticulaire constitue ainsi une totalité au sein de laquelle l'identité d'un individu est observée en permanence et renvoie à celle des autres utilisateurs [42].

40. Néanmoins, la personnalisation n'est pas spécifique au *web*, elle est au cœur des démarches de fidélisation des clients qui permettent, en contrepartie de récompenses financières, d'observer et de catégoriser leur profil de consommateur. Voir [Kessous 2013][p. 51].

41. Voir [Levinas 1961][p. 54)].

42. Nous empruntons la notion de « totalité » à Emmanuel Levinas qui écrit « La face de l'être qui se montre dans la guerre, se fixe dans le concept de totalité qui domine la philosophie occidentale. Les individus s'y réduisent à des porteurs de forces qui les commandent à leur insu. Les individus

II Exister séparément dans le monde numérique

La surveillance dont nous faisons aujourd'hui l'objet est régulièrement décriée. Cependant, il faut noter qu'elle est le plus souvent invisible. Les outils de capture des données sont en effet peu visibles ou sont présentés comme réalisant d'autres fonctions. Tel est le cas des cartes de fidélité qui sont présentées comme permettant de réaliser de bonnes affaires alors qu'elles permettent de mieux connaître un consommateur. Enfin, la surveillance, dans le mesure où elle est automatisée, peut difficilement être évitée. Néanmoins, cette surveillance est en quelque sorte acceptée et créée par les individus. En effet, plus l'utilisateur noue de relations avec des entreprises qui utilisent des bases de données et plus il utilise des services numériques, plus il diffuse son identité numérique et, ce faisant, s'expose aux stratégies d'autres acteurs. Pour sombre que cette perspective paraisse, la large adhésion des utilisateurs aux outils numériques ne conduit-elle pas à nuancer les risques que représente cette surveillance ?

A Le conflit des deux aspects de la subjectivité

Mener une existence qui échappe au regard d'autrui est ainsi rendu difficile par la structure même des outils numériques. La perte de contrôle de nos données, les incitations à les partager et les récompenses que l'on peut tirer de ce partage constituent probablement des facteurs de l'accroissement de notre visibilité [43]. Cette dernière serait donc un fait causé par l'essor des outils numériques. Cependant, les déclarations des promoteurs des outils numériques ne situent pas uniquement la visibilité au niveau des faits. Au contraire, Eric Schmidt, PDG de *Google*, en fait un impératif d'ordre moral. Souhaiter être moins visible ne signifierait-il pas avoir quelque chose à cacher ? Cette position est complétée par la prétention à une forme de pouvoir bienveillante – la devise de *Google* est *Don't be evil*– et qui prend pour appui la réalité elle-même, décrite par des données collectées automatiquement et générées par les utilisateurs. Une telle position se heurte néanmoins aux exemples de discrimination ou bien de mésaventure survenues à ceux qui s'exposent largement. Ces exemples nous conduisent désormais à nous demander si une telle « morale de la transparence », qui justifie la collecte des données et leur traitement peut jamais coincider avec la réalité de l'existence collective. Les deux points de vue que nous avons adoptés quant à l'identité numérique nous mettent face aux deux aspects du sujet historique que nous avons défini dans le cadre de notre étude du souci de soi. D'une part, le sujet est assujetti. En tant qu'objet du monde, son état est influencé par ce dernier. D'autre part, le sujet est agent et tend vers des buts. Chaque aspect entraîne un rapport à soi et aux autres distinct. Souscrire à une forme de « morale

empruntent à cette totalité leur sens (invisible en dehors de cette totalité). » . Levinas écrit aussi à propos de la totalité qu'elle surgit d'une action qui « instaure un ordre à l'égard duquel personne ne peut prendre distance. Rien n'est dès lors extérieur. » [Levinas 1961][p. IX].

43. Telle est le thèse centrale de l'ouvrage d'Emmanuel Kessous qui voit, dans l'essor des outils numériques, l'essor de l'attention comme un nouveau bien que les individus privés, comme les entreprises commerciales, cherchent à obtenir et conserver. Voir [Kessous 2013].

de transparence » , nous semble-t-il, réduit le sujet à son premier aspect et nous placerait dans une situation proche de celle du prisonnier du panoptique.

Le premier aspect du sujet correspond au point de vue des observateurs sur les consommateurs et les utilisateurs des services numériques. Le conducteur auquel s'adresse *Predestination*, par exemple, est produit par l'environnement auquel il appartient : ses déplacements sont des habitudes qui peuvent être formalisées et dont les facteurs peuvent être identifiés. Il vit ainsi une vie qui se déroule selon un rythme régulier et répétitif. Il en est de même du consommateur conçu sous l'angle de son profil : ses achats sont fortement liés à la catégorie à laquelle il appartient. Dans les deux cas, l'identité est le produit d'une identification, c'est-à-dire d'un rapport de connaissance entre l'individu et un observateur qui détermine, comme de l'extérieur, qui est l'individu. Dans cette perspective, les actions sont exprimées selon des termes quantitatifs. Elles sont aussi la cible d'une activité manipulatoire qui est de l'ordre de la programmation [44]. Par conséquent, ce premier aspect du sujet évacue la notion de « valeur » du point de vue de l'utilisateur. Ce dernier n'agirait pas en fonction de valeurs qu'il souhaiterait incarner au travers de ses actions mais principalement sous l'influence de son environnement.

La considération du sujet comme un agent, sous l'influence du thème du souci de soi permet cependant d'opposer à cette représentation de l'individu un récit intentionnel et non plus programmatique de ses actions. On peut ainsi se déplacer ou consommer afin d'acquérir un style de vie que l'on estime souhaitable. On peut constituer un profil numérique et le mettre en forme afin de présenter aux autres une identité « qui nous ressemble » ou vers laquelle on souhaiterait tendre. Dès lors, l'identité constitue une manière de se situer dans un espace de biens qui peuvent être attirants comme le style de vie du sage l'est pour le disciple stoïcien. Alors que le point de vue de la programmation des existences fait du désir une force que l'on peut quantifier et dont on peut déterminer les composants, le point de vue subjectif conduit à envisager la possibilité de désirs de second ordre qui permettent de désirer travailler sur ses désirs pour se prendre comme objet de modification. Le rapport à soi sur lequel s'appuie cette perspective est ainsi potentiellement éthique. Or, un tel rapport ne peut exister que si je me possède moi-même et que je peux expérimenter mes formes de subjectivité. Ce rapport exige dès lors l'imprévisibilité de mes comportements et une forme d'existence séparée qui comprend la possibilité de ne pas être visible pur tous de manière permanente. L'exemple du disciple stoïcien montre en effet que ce dernier ne devient authentiquement sujet qu'à l'issue d'un long processus de domestication de ses désirs. Cette domestication intervient dans le cadre d'une relation privilégiée avec son maître [45].

44. Nous empruntons la distinction entre la programmation et l'intention, qui désigne la tension vers un bien à un Elizabeth Anscombe. Voir [Anscombe 2000].

45. MacIntryre écrit ainsi : « Pour que ma vie ait un sens, je dois pouvoir m'engager dans des projets à long terme, ce qui exige la prévisibilité ; pour que ma vie ait un sens, je dois être en possession de moi-même, et pas seulement le résultat des projets d'autrui, intentions et désirs d'autrui, ce qui exige l'imprévisibilité » Voir [MacIntyre 1997][p. 102].

B Les formes de séparation

Deux formes d'existence séparée nous semblent aujourd'hui profondément remises en cause par la condition numérique ; tout d'abord, la vie privée qui nécessite de pouvoir se retirer dans un lieu d'accès limité ; ensuite, l'oubli, qui constitue la possibilité de laisser de côté une partie de son passé.

Le paradigme général de la vie privée est celui de la séparation d'espaces en fonction de leur niveau de visibilité [46]. On trouve plusieurs versions de ce contraste. La distinction aristotélicienne entre la maison et la place publique est ainsi bien connue [47]. Au XIX^{e} siècle, les juristes américains Warren et Brandeis, consacrent la vie privée [48]. Les auteurs écrivent l'essai dans lequel ils défendent la nécessité de la vie privée en réaction à l'essor de la presse et de la photographie instantanée, avec la conviction que ces nouveautés techniques nécessitent de penser de nouveaux moyens de protéger les personnes et assurer leur tranquillité [49]. Le « constraste » entre les espaces privés et les espaces publics est aussi un constraste entre des espaces normatifs hétérogènes. Ainsi Aristote associe-t-il l'espace public à la politique alors que l'espace domestique est, pour lui, le lieu de l'économie. Warren et Brandeis reprennent cette distinction et lui ajoutent une opposition entre la puissance étatique

46. Dans son introduction à l'*Histoire de la vie privée*, George Duby écrit : « Nous sommes donc partis de cette évidence que, de tout temps et partout, s'est exprimé dans le vocabulaire le contraste, clairement perçu par le sens commun, qui oppose au public, ouvert à la communauté du peuple et soumis à l'autorité de ses magistrats, le privé. Qu'une aire particulière, nettement délimitée, est assignée à cette part de l'existence que tous les langages disent privée, une zone d'immunité offerte au repli, à la retraite, où chacun peut abandonner les armes et les défenses dont il lui convient d'être muni lorsqu'il se risque dans l'espace public, où l'on se détend, où l'on se met à l'aise, en négligé, délivré de la carapace d'ostentation qui assure, au dehors, la protection. Ce lieu est de familiarité. Il est aussi domestique. C'est aussi celui du secret. Dans le privé se trouve serré ce que l'on possède de plus précieux, ce qui n'appartient qu'à soi, ce qui ne regarde pas autrui, ce qu'il est interdit de divulguer, de montrer, parce que trop différent de ces apparences que l'honneur exige de sauver en public [...]. Naturellement inscrite à l'intérieur de la maison, de la demeure, enfermée sous des serrures, dans des clôtures, la vie privée apparaît donc muréee. « ([Voir [Ariès 1999][p. 10])

47. Cette distinction fait l'objet des *Politiques*, elle est fondamentale pour déterminer ce qui est politique : est politique tout ce qui ne relève pas de conduite de la maison, conduite qu'Aristote nomme « économie » . Voir [Aristote 1999].

48. La distinction entre privé et public se trouvait déjà sous la plume de Jeremy Bentham ou Emmanuel Kant. Chacun des deux auteurs insiste sur le fait que privé et public sont des ordres normatifs hétérogènes. Warren et Brandeis notent aussi que la loi française contemporaine du moment d'écriture de leur essai fait mention de la vie privée. Néanmoins, Warren et Brandeis sont les premiers à penser un « droit à la vie privée » .

49. La similitude entre notre situation et celle de Warren et Brandeis et la nôtre est frappante lorsque l'on compare leurs écrits et leur dénonciation de la presse et de la photographie comme des moyens de déposséder l'espace domestique de son caractère sacré et les écris de l'architecte contemporain Paul Virilio par exemple. Ainsi Virilio affirme-t-il *video-surveillance and its regime of control [. . .] the banalization or popularization of global surveillance, or to put it another way, the democratization of voyeurism on a planetary scale, has overexposed our most private activities. So doing, it has exposed us to a major iconic risk. In the best case, only marketing specialists can gauge the amplitude of this risk ; in the worst, the military, investigators charged with tracking unlawful activities, political police, and automated systems for information collection.*Voir [Virilio 2002][p. 109]

et l'autonomie individuelle. La capacité à être tranquille, accordée par la vie privée, a ainsi pour but d'échapper à l'arbitraire de la puissance publique ou à la curiosité et de pouvoir mener sa vie comme on le souhaite. La vie privée est ainsi un espace de retrait garant d'une forme d'autonomie. Elle permet aussi de cultiver plusieurs types de relation définis, notamment, par le niveau d'intimité sur lequel ils reposent [50]. Dans le domaine numérique, le plus souvent, la vie privée est protégée par le contrôle de l'accès à un ensemble de données que l'on juge « personnelles » . Tel est le cas de toutes les données qui permettent de déterminer directement l'identité d'un individu. Néanmoins, ce mode de protection est rendu peu efficace par la possibilité d'accéder à de très grands volumes de données et d'en inférer des informations variées. À partir de données qui ne semblent pas personnelles il est ainsi possible de reconstituer des données qui le sont, ou bien de deviner des données qui ne sont pas accessibles [51].

Dès lors, la vie privée numérique peut difficilement avoir une portée territoriale puisqu'on ne peut parler d'espace numérique que métaphoriquement. Cependant, même dans le cas de l'utilisation des outils numériques, nous entretenons des relations variées puisqu'elles mobilisent des contenus et des niveaux d'intimité distincts. Dès lors, chaque relation met en jeu un contexte constitué par l'identité de celui avec lequel on partage et le sens donné à la relation [52]. Cette approche conduit à démultiplier les frontières qui délimitent des espaces de contraste. Elle conduit aussi à redéfinir la protection de la vie privée non plus comme le contrôle de l'accès à un ensemble de données mais comme le contrôle des changements de contexte : la vie privée est ainsi envahie si des informations sont transférées d'un contexte à un autre sans le consentement de celui qu'elles concernent [53].

À la difficulté d'être séparé du point de vue de la diffusion des données, l'essor du numérique ajoute la difficulté d'oublier ces dernières. C'est en réaction à cette difficulté que la notion de « droit à l'oubli » connaît aujourd'hui un certain succès. L'expression « droit à l'oubli » ne peut cependant que surprendre alors que nous sauvegardons, archivons et enregistrons volontairement de nombreux détails de notre vie. C'est plutôt la « mémoire » entendue comme faculté d'enregistrer et rappeler le souvenir qui fait l'objet de nos soins. Il suffit de se pencher sur les cas pathologiques d'oubli, qui surviennent suite à des lésions, par exemple, pour voir que l'oubli est, du

50. Tel est le sens de la défense de la vie privée par James Rachels. L'auteur soutient que l'existence privée est nécessaire pour entretenir des relations variées comme l'amitié ou les relations amoureuses. Voir [Rachels 1975].

51. Latanya Sweeny montre ainsi qu'à partir d'informations non personnelles (comme le code postal et la profession), il est possible de reconstituer l'identité de la plupart des citoyens américains. Voir [Sweeney 2002].

52. Cela est la thèse de Thomas Scanlon qui propose de définir les relations à partir du contenu qu'elles mobilisent et de l'engagement entre leurs participants. Voir [Scanlon 1975].

53. Helen Nissembaum propose ainsi de définir une « vie privée contextuelle » . Elle montre l'application de cette définition dans le cas de la publication d'une photo de portrait d'une jeune fille prise dans la rue. La jeune fille a obtenu gain de cause dans la mesure où la Cour a reconnu que le contexte de la rue ne permettait pas à la jeune fille de s'attendre à être photographiée et à voir sa photographie largement diffusée. Voir [Nissenbaum 1998]. H. Nissembaum souligne que nos relations peuvent se croiser mais que la spécificité des outils numériques est de nous imposer des croisements.

point de vue subjectif, un phénomène qui porte atteinte à l'identité définie comme le récit de l'histoire personnelle. L'oubli empêche en effet l'accès à des éléments de cette histoire et aux apprentissages. Sur un plan subjectif, la psychanalyse freudienne est peut-être le seul courant de pensée à donner à l'oubli une connotation positive lorsqu'il prend la forme du refoulement ou du « trou de mémoire » qui visent à protéger l'individu. L'oubli est alors un phénomène proche de la « cicatrisation » : les éléments de la mémoire perdent petit à petit de leur présence et sont oubliés. Au niveau subjectif, l'oubli est ainsi un fait peut contrôlable. Il ne s'agit pas ici de l'aborder comme un phénomène volontaire ou une prérogative.

Cette prérogative se heurte d'ailleurs aux nécessités de la conservation des traces issues de la vie sociale. Le droit à l'oubli ne peut, de manière évidente, prendre la forme d'une effacement des données nécessaires à la mise en œuvre de la justice ou des missions de l'État comme l'impôt. En d'autres termes, le « droit à l'oubli » se heurte aux cas dans lesquels les traces, même individuelles, constituent des « biens publics » . Tel est le cas des traces utilisées à des biens statistiques. L'oubli comme prérogative individuelle est ainsi confronté aux impératifs pratiques de la gestion et du maintien des institutions. Dans le cadre des outils numériques, il est aussi confronté aux intérêts financiers des entreprises pour qui les données constituent des actifs que l'on peut valoriser.

Enfin, il faut ajouter à cet impératif pratique l'impératif moral que constitue, pour les Etats, « l'injonction à se souvenir » [54]et à perpétuer le souvenir de certains évènements particulièrement graves de l'histoire humaine comme la Shoah. Ce devoir a pour fonction, notamment, de rendre honneur aux victimes de ses évènements mais aussi de lutter contre leur répétition. Il interdit dès lors l'effacement des traces qui concernent ces évènements. La notion même de « devoir de mémoire » fait débat chez les historiens eux-mêmes. Ainsi la mémoire individuelle est-elle un phénomène biologique et subjectif. Au contraire, la mémoire présentée par l'historien est objectivée et reconstruite, c'est donc une « histoire » . Dès lors, il faudrait parler, plutôt que d'un « devoir de mémoire » d'un « devoir d'histoire » . Dans la mesure où l'histoire implique la sélection des faits que l'on met en lumière, l'oubli est inhérent à sa mise en œuvre. Le devoir d'histoire est autant un devoir d'oubli qu'un devoir de mémoire. Néanmoins cet oubli n'est pas le fait d'un individu mais une tâche d'ordre scientifique.

Pourtant, les exemples récurrents d'utilisation de données anciennes et parfois inexactes contre ceux qu'elles concernent plaide pour un effacement de ces dernières. Sur le modèle de la « prescription » , cet oubli ne constituerait pas un « caprice » propre à se heurter aux nécessités pratiques et morales de la vie en société. Il constituerait plutôt un « droit de recommencer » consenti par la société à ses membres. Tout comme la distinction, d'origine levinassienne, entre « l'infini » et la « totalité » nous a conduit à postuler la nécessité de la distinction entre privé et public, il nous semble qu'elle pousse à accepter la nécessité de l'oubli qui, en empêchant que nous ne soyons réduits à la somme de nos inscriptions dans le monde,

54. Voir [Ricoeur 2000][p. 106].

nous garantit la possibilité d'un renouveau.

Aujourd'hui, vie privée et oubli prennent la forme de droits garantis par des contrats comme les « conditions d'utilisation » des outils numériques. Nous avons, lors de notre enquête sur l'éthique du souci de soi, pris nos distances avec une approche contractualiste. Nous présentons désormais les difficultés que nous voyons dans sa mise en œuvre et une piste de réflexion afin de saisir la possibilité d'une vie privée et de définir les modalités d'oubli dans le monde numérique.

III Les pratiques de liberté possibles

La culture d'une existence séparée nous semble être une pratique de liberté propre à permettre le développement d'un sujet éthique, c'est-à-dire capable de constituer pour lui-même une subjectivité propre. Nous interrogeons désormais les formes que peuvent prendre ces pratiques dans le monde numérique. Dans la mesure où la structure des outils informatiques est porteuse d'effets spécifiques quant à la possibilité d'être séparé, notre thèse est ici que les pratiques de liberté doivent s'appuyer sur la modification de cette structure.

A La contractualisation des formes de séparation

Aujourd'hui, la protection de la vie privée prend la forme de la garantie d'un « droit » [55]. Le respect et l'exercice de ce droit sont assurés par un ensemble d'outils. D'une part, la récolte du consentement des utilisateurs à voir leurs données récoltées, traitées et conservées. Cette récolte garantit que l'utilisateur a choisi de divulguer des données qui le concerne. Elle assure aussi que l'utilisateur a été informé des données qui seront collectées et de traitements qui pourront s'y appliquer. D'autre part, certains sites offrent des outils qui permettent de configurer la visibilité des données. Tel est le cas de la plupart des réseaux sociaux numériques, qui permettent de créer des groupes d'utilisateurs auxquels on est connecté et de spécifier, pour chaque groupe, le niveau de visibilité d'un ensemble de données. Enfin, des mécanismes institutionnels, permettent d'exercer auprès des possesseurs de données un « droit de retrait » . En France, la « Loi Informatique et Libertés » octroie ainsi un droit de rectification et de retrait des données numériques à ceux qu'elles concernent. Le « droit à l'oubli » n'est pas, lui, encore formalisé de manière aussi explicite [56]. L'effacement des données est d'ailleurs souvent complexe, soit parce que les entités qui les collectent n'offrent pas d'outils pour demander l'effacement, soit parce que la dispersion des données contraint à s'adresser à un grand nombre d'entités.

Lawrence Lessig, juriste à l'initiative des *Creative Commons*, un système de licences pour les œuvre numériques, remarque que le code informatique constitue une loi dans la mesure où il contraint nos capacités d'action [57]. Ce faisant, Lessig propose d'appliquer l'utilisation de licences à la protection de la vie privée et du droit à l'oubli [58]. Chacun pourrait ainsi, lors de la diffusion de ses données, leur adjoindre un document précisant les utilisations qui en sont autorisées [59]. La licence constituerait

55. En France, le droit à la vie privée est ainsi garantit par l'article 9 d *Code Civil*.

56. Le « droit à l'oubli » comme droit autonome fait ainsi l'objet d'un projet de règlement européen actuellement étudié. Voir http ://www.cnil.fr/linstitution/actualite/article/article/construire-ensemble-un-droit-a-loubli-numerique/

57. Voir [Lessig 2000].

58. Lessig propose que les utilisateurs se voient octroyer un droit de propriété sur leurs données et qu'ils expriment les restriction qui s'appliquent à leur usage. Voir [Kessous 2013][p. 128 et sq.].

59. Aujourd'hui, les *Creative Commons*, permettent par exemple de spécifier si une œuvre numérique peut être librement partagée, si elle peut faire l'objet d'une commercialisation et si elle peut être modifiée.

ainsi un contrat qui devrait être honoré par celui qui collecte les données. Dans cette perspective, le droit à la vie privée et à l'oubli constituent des formes de droits de propriété. L'utilisateur pourrait spécifier les conditions d'utilisation de ses données auxquelles il consent, ce qui semblerait renforcer ses capacités de contrôle sur le devenir de ses données.

Cependant, cette approche nous semble complexe à mettre en œuvre. Tout d'abord, elle repose sur la possibilité d'appliquer des normes juridiques à des acteurs qui sont hébergés par des pays aux législations hétérogènes. De plus, les outils numériques sont enchevêtrés – nous utilisons simultanément plusieurs appareils pour accéder à des sites qui échangent des données sans que nous le sachions. Il est ainsi difficile de savoir ce à quoi nous consentons vraiment lorsque nous acceptons de diffuser nos données. Consentir à la collecte des données par un réseau social, par exemple, se solde ainsi souvent par le partage des données avec les nombreux partenaires du site. De plus, la difficulté est accrue par la diversité des définitions de l'expression du consentement. L'acceptation de conditions d'utilisation est le cas le plus explicite de consentement – même si, nous l'avons souligné, leur présentation et leur rédaction influencent beaucoup leur acceptation. Néanmoins, certains outils s'appuient, pour détecter le consentement des utilisateurs sur des éléments moins explicites, comme le paramétrage des navigateurs web. Ainsi, accepter l'utilisation de *cookies*, est-ce accepter de voir ses données collectées et conservées ? Chaque site apporte une réponse spécifique à cette question et chaque navigateur offre des réglages spécifiques.

Enfin, la portée du contrôle sur la diffusion, la collecte et le traitement des données et déterminée par les fonctionnalités des outils de paramétrage de la visibilité des données. Dans la plupart des cas, un ensemble de données sont « publiques » par défaut, ce qui limite d'autant la possibilité de restreindre sa visibilité. Certains outils augmentent ainsi la difficulté pour un utilisateur de paramétrer sa visibilité en le forçant à modifier pour cela des paramètres qui, par défaut, l'exposent largement. Le plus souvent, les utilisateurs n'exercent ainsi pas leur droit à limiter la visibilité de leurs données [60]. Même lorsqu'ils le font, ils doivent régulièrement renouveler cette opération. Les outils changent et forcent les utilisateurs à mettre à jour leurs paramètres. De plus, les outils modifient l'étendue des données visibles par défaut. C'est pour cette raison que la protection de la vie privée ou de l'oubli à partir d'un droit de propriété nous semble difficile à mettre en œuvre. Dans un monde fortement connecté et en perpétuel changement, elle exige une activité permanente des utilisateurs qui est probablement impossible à exercer.

B Inscrire la séparation dans la structure du monde numérique

Plusieurs navigateurs web permettent aujourd'hui d'étendre la liste de leurs fonctionnalités au moyen de logiciels dédiés qui ont pour but d'empêcher la collecte de

60. Seuls 15 à 30% des utilisateurs de *Facebook* utilisent les réglages de vie privée. http ://www.lefigaro.fr/web/2009/12/10/01022-20091210ARTFIG00800-nouveaux-reglages-de-facebook-les-pieges-a-eviter-.php

données ou bien d'en contrer les effets. Les extensions les plus connues permettent ainsi de supprimer l'affichage des publicités ciblées, de diminuer la quantité d'informations accessibles à un site, voire d'y empêcher l'accès si le site, par exemple, diffuse les données à des tiers. Ces extensions tirent parti de la possibilté de modifier le code des programmes informatiques et des sites Internet – ainsi empêcher l'affichage des publicités revient-il à réécrire le code des pages *web*. Elles ont pour intérêt d'automatiser la protection de l'individu et de l'appliquer à tous les sites. Ce n'est plus ce dernier qui doit, pour chaque site, assurer sa protection, mais l'outil qui en devient le garant. Dès lors, l'utilisateur n'a plus qu'à indiquer ses préférences une seule fois. Ces extensions corrigent en quelque sorte la structure des sites web et en contrent les effets. Elles montrent l'efficacité d'une approche qui consiste non plus à obliger l'utilisateur à être sans cesse vigilant et à renouveller sa protection mais à modifier le code informatique. Néanmoins, elles restent des outils appliqués *a posteriori* au monde numérique et ne protègent que ceux qui les utilisent.

Une approche comparable mais qui étendrait ses effets à l'ensemble du monde numérique est ainsi la prise en compte de la vie privée – ou d'autres formes de séparation – dès la conception des outils numériques [61]. Cette prise en compte passe, notamment, par la cartographie de la diffusion des données et son contrôle ainsi que la restriction de leur durée de conservation. La durée de conservation des traces des recherches effectuées par les moteurs de recherches sont un exemple de prise en compte à la conception de l'oubli. Ces traces contiennent le plus souvent l'énoncé de la recherche et l'identité de celui qui l'a effectuée. Elles sont notamment utilisées dans des buts de personnalisation des résultats de recherche, de publicité. Même dans les cas où elles sont « anonymisées », l'identité des personnes qui ont effectué des recherches peut être reconstituée en les croisant avec d'autres données. Il est ainsi possible de découvrir des informations sur les utilisateurs, ce qui confère une connaissance extrêmement précise et importante du comportement des usagers aux moteurs de recherche. Aujourd'hui, ces traces sont conservées de moins en moins longtemps par les moteurs de recherche [62]. Cet oubli automatisé n'est pas issu d'une modification de la législation mais de la compétition entre les différents moteurs de recherche qui ont fait de l'effacement des données un argument publicitaire. L'automatisation de l'oubli permet, dans ce cas, de décharger les utilisateurs de la responsabilité de l'effacement des données et de sa difficulté technique. Elle ne répond pas à toutes les difficultés liées à la structure des échanges numériques. En effet, si les données ont été dupliquées, l'oubli des données par les serveurs des moteurs de recherche ne provoque pas l'effacement des serveurs sur lesquels elles ont été copiées. Cependant, cet exemple montre, une fois de plus, que le code informatique constitue la « loi » du monde numérique dans la mesure où il configure la visibilité, sa durée et *in fine*, les possibilités d'action des utilisateurs.

C'est tout l'intérêt, nous semble-t-il des approches qui ne font pas de la protection de la vie privée, par exemple, la responsabilité des utilisateurs mais celle des

61. Voir [Métayer 2013].

62. Google conserve ainsi les données des utilisateurs 9 mois et Yahoo 90 jours. Voir [Kessous 2013][p. 138].

fournisseurs des outils qui peuvent la violer. Dès lors, les valeurs, comme la vie privée et l'oubli, sont prises en compte dès la conception des outils qui doivent intégrer des mécanismes pour les mettre en œuvre. Avec l'essor de techniques de production des logiciels qui permettent de prendre en compte ces valeurs et leur réalisation et de les modifier tout au long du cycle de vie du logiciel, il est ainsi possible d'envisager la possibilité de revenir sur leur implémentation, de les mettre à jour et d'ajouter la prise en compte de nouvelles valeurs.

IV Conclusion

Alors qu'il analyse l'essor des outils informatiques, Deleuze diagnostique l'essor d'une « société de contrôle » fondée sur une visibilité permanente, qui nous suivrait dans tous les espaces de notre vie[63]. Notre analyse des techniques de gouvernement en vigueur dans le monde numérique nous laisse penser que nous vivons peut-être dans une société comparable à celle que décrit Deleuze. Observés, comparés, classés, enregistrés, nous sommes devenus des cibles marketing dont l'identité est déterminée comme de l'extérieur par la remise en perspective de nos traces avec celles produites par les autres individus et plus généralement notre environnement.

Face à la capillarité du contrôle, Deleuze souligne la nécessité de créer des moyens de mener une existence protégée de l'observation constante et de ses effets afin de conserver une forme d'autonomie. Nous souscrivons à cette thèse. La préservation de moyens de mener une existence protégée d'une observation complète et permanente nous semble nécessaire pour constituer une « vie propre » . Dans la mesure où le contrôle est un effet de la structure des outils numériques, nous proposons de modifier le code qui la constitue soit *a posteriori* en étendant les fonctionnalités des outils ou en changeant le contenu qu'ils affichent, soit dès la conception des outils numériques.

63. Voir [Deleuze 2003].

Chapitre 5

Conclusion

Notre réflexion a pris pour point de départ les débats passionnés qui entourent les outils numériques. Faut-il se jeter sur les dernières nouveautés, « informatiser » toutes les facettes de notre vie, ou, au contraire, craindre le développement de nouvelles applications et les adopter avec méfiance, voire les rejeter ? Nous avons choisi d'adopter, face à ces positions tranchées, un point de vue nuancé. Nous avons établi que la technique n'est pas un « corps étranger » au corps social qui le mettrait en danger. Elle n'est pas plus, en soi, un outil de libération ou de développement. Certes, nous avons mis au jour un « agent technique » . Cependant, cet agent nous est apparu intrinsèquement « neutre » . Les techniques peuvent tout à la fois soutenir des rapports de domination ou intensifier la subjectivité de celui qui les pratiquent. Surtout dans le cas des technologies de transport et de communication, leur agent technique repose sur le conditionnement de notre mode de présence et de notre visibilité qu'elles permettent. Cependant, évaluer ce conditionnement hors d'un contexte spécifique nous est apparu vain. Nous avons ainsi déplacé la perspective de l'analyse vers une enquête sur les usages techniques plus que sur l'essence de la technique.

Nous avons entrepris ce déplacement en nous mettant en quête d'un cadre éthique propre à fournir un modèle de l'agent technique et de son effectivité et à guider le développement et l'évaluation des techniques et de leurs usages. Si de nombreux courants de pensée ont donné lieu à des propositions en éthique appliquée aux techniques – nous pensons ici notamment à la phénoménologie ou aux éthiques déontiques – nous avons choisi de nous inspirer de l'éthique du souci de soi telle que Michel Foucault l'envisage. L'éthique du souci de soi permet de placer la technique au centre de la réflexion éthique en en faisant un opérateur nécessaire dans la construction de la subjectivité. Nous avons aussi montré qu'elle place la réflexion sur la visibilité – comprise comme ce que l'on donne à voir de soi aux autres et comme ce que l'on donne à voir à soi-même – au centre du processus de subjectivation. Ces deux points nous semblent particulièrement importants dans le cadre d'une analyse des outils numériques puisqu'ils sont souvent conçus comme des moyens de surveillance.

Le troisième moment de notre réflexion s'est notamment attaché à analyser l'ampleur de la surveillance à l'ère du numérique et ses effets. Nous avons remis en perspective les débats contemporains sur le droit à l'oubli et à la vie privée avec les outils numériques en nous demandant quelles conditions de vie ils déterminaient. Nous avons souligné la remise en cause des frontières spatiales, géographiques et sociales à leur contact. L'espace est contracté grâce à l'accélération des communications. Les données provenant de relations sociales distinctes sont agrégées afin

de déterminer les profils des utilisateurs. Les prévisions permettent d'imaginer uniformiser le temps et d'influencer les évènements à venir. Nous avons montré ce en quoi cette remise en cause des frontières perturbait les pratiques sociales. Nous en avons aussi souligné les limites. Enfin, dans la perspective de l'éthique du souci de soi, nous avons identifié des directions de transformations possibles. Il nous semble en effet que les derniers travaux en éthique appliqués à l'informatique, qui mettent l'accent sur la transformation des interfaces et l'information des utilisateurs ne vont pas assez loin. Si on accepte que la structure des outils influence leurs usages et leur importance éthique, c'est cette structure qu'il faut modifier, notre enquête philosophique appelle ainsi un ensemble de travaux pratiques qui participeront à cette modification.

Chapitre III

Une démarche de conception et d'implémentation de la vie privée basée sur le contrôle d'accès et appliquée aux compositions de services

Chapitre 1

Introduction

I Problématique

Qui aurait pu prédire, au début des années 2000, que l'approche orientée services serait aujourd'hui aussi répandue ? Alors que les procédés sont des moyens bien connus de représenter le fonctionnement d'une application, les services sont aujourd'hui devenus un outil de choix pour les réaliser. Le développement de langages dédiés à la représentation des procédés sous la forme d'une composition de services et l'existence de plusieurs logiciels largement utilisés pour concevoir et exécuter ces compositions témoignent de la vitalité et de la large adoption de l'approche orintée services.

Cependant, les services sont polymorphes. Le terme de « service » recouvre ainsi une réalité beaucoup plus large que les « services web » qui sont le type de services le plus connu et le plus utilisé. Les services sont en effet des entités informatiques autonomes réparties sur un réseau et qui peuvent être facilement composées. Tout peut, dès lors être service. Des protocoles, comme l'Universal Plug and Play (UPnP), permettent de les utiliser dans le cadre de la domotique. Les infrastructures, les données ou les logiciels peuvent être des services. L'approche orientée services est ainsi bien adaptée à de nouveaux paradigmes comme l'informatique pervasive, qui repose sur l'intégration, au sein d'une application, d'entités réparties et autonomes comme des logiciels, des téléphones ou des capteurs.

Néanmoins, ces nouveaux paradigmes confrontent l'approche orientée services à de nouveaux défis. Alors qu'habituellement la conception d'une application à partir de services repose sur l'utilisation d'un ensemble de services bien connus, l'informatique pervasive repose sur des services hétérogènes, du point de vue de leur technologie et de leur implémentation, et dynamiques. Ils peuvent à tout moment rejoindre ou quitter le réseau au travers duquel l'application est réalisée. La gestion du dynamisme et de l'hétérogénéité des services nécessite des connaissances très importantes sur des services spécifiques et reste aujourd'hui souvent réalisée au moyen de solutions ad hoc. Pour chaque type de services une nouvelle solution d'intégration dans une composition doit donc être développée.

De plus, la plupart du temps ces applications doivent satisfaire une grande diversité d'exigences. Dans le cas des bâtiments intelligents, par exemple, les données manipulées par les services sont hautement sensibles. Leur diffusion menace de manière évidente la vie privée des habitants. La prise en compte des propriétés comme la sécurité reste un champ d'investigation dans le cadre des études sur l'approche orientée services. Les travaux se concentrent en effet souvent sur la prise en compte

de contraintes liées aux fonctionnalités des applications. Concevoir et réaliser de manière optimale des compositions de services qui prennent en compte les spécificités des environnements pervasifs tout en respectant des propriétes de sécurité comme le contrôle d'accès reste ainsi difficile.

II Objectifs

Notre but est d'aider les concepteurs d'applications à concevoir au mieux et mettre en œuvre des orchestrations de services hétérogènes dynamiques et sécurisées par le contrôle d'accès afin de protéger la vie privée des utilisateurs et des possesseurs des données utilisées par l'orchestration. Afin d'y parvenir, nous proposons de configurer les propriétés de contrôle d'accès des services au moyen d'une démarche dirigée par les modèles divisée en deux étapes :

- **Au niveau conception**, nous proposons de spécifier séparément l'orchestration et la politique de contrôle d'accès à un niveau abstrait. Chaque préoccupation est ainsi manipulée par un expert dédié. Nous estimons que le contrôle d'accès doit être pris en compte dès la conception de l'application afin d'éviter le recours à des *patchs* et à la programmation manuelle à l'exécution. Ces deux solutions sont en effet sources d'erreurs, ce qui n'est pas acceptable en matière de sécurité. En rester à un niveau abstrait permet de se libérer des détails de l'implémentation et de s'adapter à l'état de la composition afin de ne pas souffrir de l'hétérogénéité et du dynamisme des services.
- **Au niveau exécution**, nous développons une architecture qui permet de configurer les services concrets au moyen de *proxies* responsables de l'exécution du contrôle d'accès. Seuls ces *proxies* sont directement accessibles lors de l'exécution de l'orchestration. Les services concrets sont ainsi protégés.
- **Le passage d'un niveau à l'autre se fait au moyen de transformations de modèles vers textes automatisées**. Ceci permet de s'abstraire de la programmation manuelle et de garantir la protection des services concrets par les *proxies*.

III Structure du document

Le présent document est organisé en deux parties principales. Après une partie d'état de l'art, nous introduisons notre contribution.

L'état de l'art est organisé en deux chapitres principaux :

- Le chapitre A, intitulé « L'approche orientée services et ses implémentations » présente les concepts fondamentaux de l'approche orientée services et leur implémentation. Nous analysons les limites de ces implémentations dans la perspective des compositions de services hétérogènes et dynamiques et de la prise en compte des propriétés non fonctionnelles.
- Le chapitre B, intitulé « Sécurité : concepts et buts » présente la définition de la sécurité et ses concepts clés. Nous mettons en exergue les problèmes

de sécurité dans les compositions de services et nous montrons en quoi le contrôle d'accès est une solution possible pour résoudre les problèmes liés à la protection de la confidentialité et l'intégrité des données, deux propriétés nécessaires dans la protection de la vie privée. Nous dressons aussi un état des avancées dans le domaine du contrôle d'accès appliqué aux services.

Nous présentons notre contribution en cinq chapitre principaux :

- Le chapitre A, intitulé « Présentation générale de la démarche» introduit la structure globale de notre démarche dirigée par les modèles. Nous montrons ses buts et ses intérêts. Nous présentons aussi les concepts de l'ingénierie dirigée par les modèles qui motivent notre travail.
- Le chapitre B, intitulé « Conception et exécution d'une orchestration de services hétérogènes et dynamiques sécurisée par le contrôle d'accès » présente en détail le niveau conception et le niveau exécution de notre démarche. Au niveau conception, nous présentons les métamodèles que nous avons élaborés et leurs liens. Nous présentons au niveau exécution l'architecture que nous avons conçue afin d'exécuter une orchestration de services hétérogènes et dynamiques sécurisée par le contrôle d'accès.
- Le chapitre C, intitulé « Implémentation de la démarche et validation » décrit les outils que nous avons réalisés pour implémenter notre démarche et son application au cas de la gestion d'urgences médicales dans un bâtiment intelligent. Ce chapitre permet d'établir la faisabilité de notre solution et son efficacité.
- Le chapitre D, intitulé « Perspectives et extensions » s'attache à démontrer que notre démarche peut être appliquée à d'autres propriétés non-fonctionnelles. Nous prenons comme exemple le contrôle d'usage.

Chapitre 2

État de l'art

L'approche orientée services est aujourd'hui largement utilisée pour construire des applications à partir d'éléments informatiques existants et répartis. Néanmoins, l'adaptation de l'approche à services à de nouveaux paradigmes, comme l'informatique pervasive, est mise en question par l'importante collecte de données qu'ils nécessitent et à l'hétérogénéité et au dynamisme des éléments sur lesquels ils reposent.

Nous présentons dans la Section 1 les fondements de l'approche à services. Nous commençons par en donner les concepts de base. Nous présentons ensuite les différents moyens de composition des services en insistant sur la solution qu'ils apportent aux problèmes du dynamisme et de l'hétérogénéité des services. Enfin, nous exposons les implémentations possibles de l'approche orientée services avant de dresser une synthèse générale sur l'approche à services.

La Section 2 est consacrée à la sécurité, envisagée notamment dans les compositions de service. Nous insistons plus particulièrement sur le défi que constitue la protection de la vie privée et la solution que constitue le contrôle d'accès. Nous présentons tout d'abord une définition générale de la sécurité avant de dresser la liste des menaces auxquelles les compositions de services pour les applications pervasives sont exposées. Ceci fait, nous exposons les solutions techniques en matière de sécurité puis nous nous concentrons sur le contrôle d'accès avant de conclure sur les solutions et les challenges en matière de sécurité dans le cadre des applications pervasives réalisées sous la forme de compositions de services.

La Section 3 conclut notre état de l'art. Nous précisons les voies de recherche qui restent ouvertes dans la production d'applications pervasives réalisées sous la forme de compositions de services prenant en compte des propriétés de contrôle d'accès.

I L'approche orientée services et ses implémentations [1]

Les services sont aujourd'hui largement répandus et utilisés pour réaliser des applications à partir de fonctionnalités déjà implémentées et disponibles. Dans le monde du web, par exemple, les services sont employés pour réaliser des applications réparties. Dans le monde de l'informatique pervasive, de nouveaux appareils, comme les smartphones, peuvent héberger des services ou bien interagir avec eux [2]. Enfin, d'autres domaines, comme les réseaux domestiques, sont aussi peuplés de services qui permettent de fournir de nouvelles fonctionnalités.

Néanmoins, ces services sont extrêmement hétérogènes, ce qui complique leur intégration au sein d'applications. Dans le cas des applications pervasives, les services sont le plus souvent dynamiques, ce qui en complique la gestion. Dans cette première section, nous présentons l'approche orientée services et ses différentes implémentations. Nous mettons en évidence le manque d'outils pour concevoir des applications avec des services hétérogènes et dynamiques.

Dans un premier chapitre, nous introduisons les concepts et les définitions de base de l'approche orientée services. Nous définissons notamment l'architecture orientée service. Dans un second chapitre, nous présentons le principe de la composition de services et trois exemples de compositions, la composition par procédé, la composition structurelle et la composition sémantique. Dans le troisième et dernier chapitre, nous présentons les implémentations les plus répandues de l'approche orientée services. Nous décrivons les services Webs, l'*Universal Plug and Play* (UPnP) et le *Device Profile for services Web* (DPWS) et les composants orientés services. Une synthèse clôt ce chapitre pour en rappeler les grandes lignes.

1. Cette partie de l'état de l'art s'inspire des travaux présentés dans [Marin 2008] et [Chollet 2009a] et notamment de la classification et de la caractérisation des services que les auteurs proposent.

2. Il est par exemple possible d'accéder à un ensemble de services à partir d'un iPhone. Voir par exemple l'application *Service Orchestration Client* disponible à l'adresse http ://itunes.apple.com/us/app/service-orchestration-client/id394379182 ?mt=8

A Concepts et définitions

Malgré son utilisation très répandue, la notion de service connaît toujours plusieurs définitions. Ceci est probablement dû au fait que n'importe quoi peut être un service, des infrastructures aux logiciels en passant par les données [Banerjee 2011].

Le *World Wide Web Consortium* (W3C), qui a en charge de promouvoir les standards du web définit un service comme

> *an abstract resource that represents a capability of performing tasks that form a coherent functionality from the point of view of providers entities and requesters entities. To be used, a service must be realized by a concrete provider agent.*

Le W3C met ainsi l'accent sur le niveau d'abstraction des fonctionnalités du service, qui est permise par la description du service qui laisse de côté les détails techniques de son implémentation.

Cependant, le W3C ne s'intéresse qu'aux applications exposées comme des services sur le réseau Internet alors que des appareils, comme les capteurs, ou les infrastructures, peuvent eux aussi être exposés comme services. Il est ainsi nécessaire de disposer d'une définition plus générale des services, comme celle proposée par Mike Papazoglou [Papazoglou 2003] qui affirme que :

> *Services are self-describing, platform-agnostic computational elements that support rapid, low-cost composition of distributed applications. Services perform functions, which can be anything from simple requests to complicated business processes.*

La notion de service n'est pas dépendante d'une plateforme particulière. Un service réalise ainsi un ensemble de fonctions bien définies. Il peut être implémenté dans n'importe quelle technologie. Il est autonome car il n'est pas dépendant d'autres services pour réaliser les fonctionnalités qu'il propose et car il ne sait pas dans quel contexte il va être utilisé. Ces deux propriétés permettent le faible couplage des services.

Ce faible couplage permet l'interopérabilité des services et leur réutilisabilité en définissant les approches à service sur laquelle Sward [Sward 2011] met l'accent lorsqu'il définit l'approche orientée service comme :

> *an approach in which modular, accessible, self-describing, implementation independent, interoperable, and reusable components are published as services which can be remotely invoked and consumed by other applications or combined with other services.*

Sward souligne ainsi qu'une application faite de services est réalisée par la composition de plusieurs services répartis et issus de multiples organisations. La description des services, qui en présente les propriétés et qui est exposée sur un réseau, comme l'Internet ou un réseau domestique, est primordiale pour découvrir et invoquer les services. Cette description constitue l'interface du service et permet à un grand nombre de clients potentiels de rechercher et d'utiliser le service.

Dès lors, les fournisseurs et les clients doivent se mettre d'accord sur les propriétés des services afin de faciliter leur sélection par les clients et de permettre aux fournisseurs de prouver qu'ils proposent des services utilisables. Cet accord est nommé *Service Level Agreeement* (SLA). Il précise l'engagement du fournisseur à réaliser un service qui respecte un ensemble de contraintes. Le SLA permet de guider la sélection des services en exprimant les contraintes qui s'appliquent à un service. Il permet de préciser :

- **un ensemble de contraintes syntaxiques** (les méthodes du service et leur signature) [Beugnard 1999]
- **un ensemble de contraintes comportementales** (les pré-conditions et les post-conditions qui s'appliquent aux méthodes du service) [Beugnard 1999]
- **un ensemble de contraintes quantitatives** (le taux d'erreurs du service, sa disponibilité, le nombre de requêtes concurrentes qu'il peut accepter, son temps de latence, etc.) [Bianco 2008]
- **un ensemble de contraintes qualitatives** (l'interopérabilité du service, son évolutivité et ses propriétés de sécurité) [Bianco 2008]

Le SLA permet ainsi aux fournisseurs de services et à leurs clients de s'entendre sur les propriétés attendues d'un service, qui sont reprises dans son interface.

Dans la mesure où notre travail mobilise toutes les facettes des services, qui sont abordées séparément par chaque définition, nous proposons une définition de travail d'un service que nous formulons de la manière suivante :

> Un service est une entité logicielle qui réalise un ensemble de propriétés fonctionnelles et non fonctionnelles spécifiées par un accord entre le fournisseur et le client du service. Ces propriétés forment l'interface du service, exposée sur un réseau sous la forme d'une description. Les clients peuvent ainsi rechercher, sélectionner et invoquer le service en respectant l'accord conclu entre client et fournisseur. La notion de service est indépendante d'une plateforme. Un service est autonome et peut être facilement composé avec d'autres services.

1 Approche orientée service – Service-Oriented Computing (SOC)

Les services sont les éléments de base de l'approche orientée service ou *Service-Oriented Computing* (SOC). SOC est une architecture qui vise à composer des services existants tout en garantissant leur faible couplage [Papazoglou 2003]. Cette composition est permise par la communication entre trois entités logiques :

- Le **fournisseur de service**, qui implémente les services, les décrit et les met à disposition.
- Le **registre de services**, qui répertorie l'ensemble des services disponibles.
- Le **client**, qui consomme des services.

Les communications entre ces entités se font deux à deux :

- **Le fournisseur de service publie les services qu'il offre dans l'annuaire**, c'est-à-dire qu'il enregistre leur description dans l'annuaire.

- **Le client découvre les services disponibles en parcourant l'annuaire** à la recherche des services qui répondent à ses besoins.
- **Le client invoque les services pertinents** afin de les consommer en se liant au fournisseur de service approprié.

La Figure 2.1 reprend le fonctionnement de SOC en présentant les interactions entre fournisseur, client et annuaire de services.

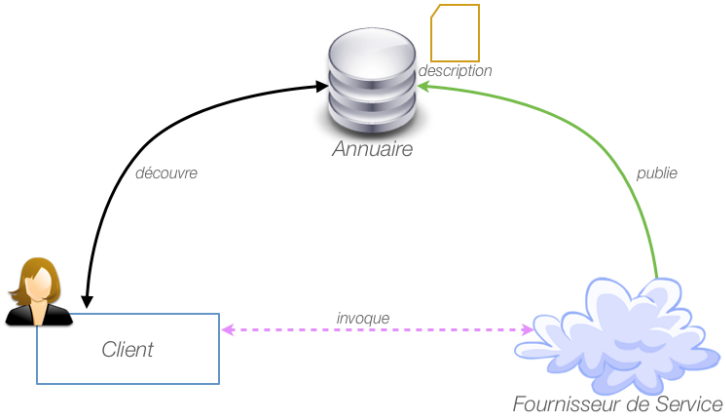

FIGURE 2.1 – Entités nécessaires à SOC

De par son caractère réparti, SOC met l'accent sur la communication entre plusieurs organisations. Les unes jouent le rôle de fournisseur de service et les autres de clients. Au sein de cette communication, la description du service a une place primordiale puisque sa publication permet la recherche et la sélection des services qui correspondent aux besoins des clients.

La description précise les propriétés du service. Elle est abstraite dans la mesure où elle ne fait pas état des détails de l'implémentation des services. La manière dont le service est dès lors réalisé est ainsi masquée au client.Tous les services qui répondent à un SLA (c'est-à-dire à tous les services qui partagent la même interface, et donc la même description) peuvent être utilisés de manière équivalente. Ce niveau d'abstraction permet de masquer l'hétérogénéité des services en ne se préoccupant pas de leur implémentation. S'il est courant de réduire les services aux services Web, qui en sont l'implémentation la plus répandue, ce ne peut être le cas dans une application pervasive, qui met en jeu un grand nombre de technologies. Un service qui disparaît peut être remplacé par un autre, équivalent, afin de permettre à l'application de continuer à s'exécuter. Ceci est important afin de gérer la disparition d'un service défectueux ou bien l'apparition et la disparition de services dans les applications dynamiques, comme les applications pervasives.

La gestion du dynamisme nécessite ainsi de mettre en place un mécanisme de liaison retardée : le client ne doit pas connaître le service qu'il invoque avant l'exécution. Afin de favoriser l'adaptatibilité de la composition, ce n'est qu'à l'exécution que le client choisit, parmi les services disponibles, celui qui lui convient. Le client

doit aussi pouvoir changer de service si le service qu'il utilise venait à disparaître afin de garantir la continuité de l'application.

Le client doit donc pouvoir se lier à un nouveau fournisseur et invoquer un nouveau service si celui qu'il utilise vient à disparaître. Plusieurs solutions sont possibles. Par exemple, de nouvelles primitives de communications entre client, fournisseur et annuaire peuvent être ajoutées [Escoffier 2007]. Le fournisseur peut ainsi indiquer au registre des services qu'il souhaite retirer le service qu'il propose. Le client peut, lui, être notifié de la disparition ou de l'ajout d'un service correspondant à ses besoins. La Figure 2.2 reprend l'architecture proposée par SOC augmentée de ces nouvelles possibilités de communication.

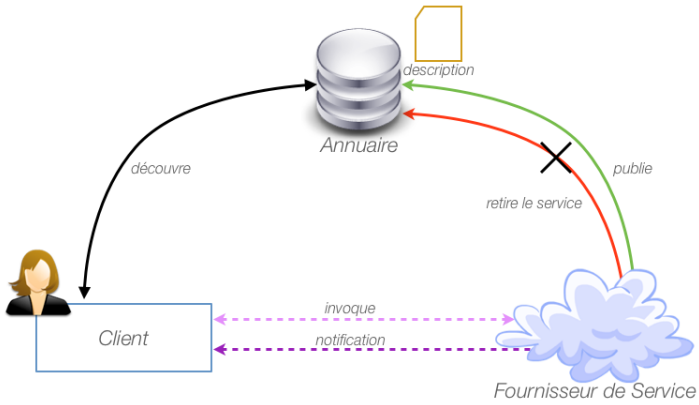

FIGURE 2.2 – Architecture de SOC adaptée au dynamisme

Enfin, la gestion du dynamisme peut être confiée à une plateforme dédiée [Callaway 2008] chargée de fournir les services au client. La plateforme masque alors au client la disparition et l'apparition des services.

2 Architecture orientée service – Service-Oriented Architecture (SOA)

SOC définit un style d'architecture pour une application répartie qui doit être réalisé par une architecture orientée service ou *Service-Oriented Architecture* (SOA). L'OASIS[3] définit une SOA comme :

> *A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.*

Ainsi une SOA vise-t-elle à intégrer des services faiblement couplés et distribués. Une SOA a pour but de permettre aux clients de découvrir et de consommer les

3. *Organization for the Advancement of Structured Standards*, c'est un organisme qui maintient les standards liés aux services Web.

services disponibles publiés par des fournisseurs. Pour ce faire, une SOA doit reposer sur :

- Un mécanisme pour permettre l'interopérabilité entre les services implémentés de manière hétérogène sur des plateformes variées. Le plus souvent, l'interopérabilité est permise par la définition de standards pour la description des services et les protocoles de communication.

- Le partage des services disponibles entre plusieurs partenaires en constituant un annuaire de services.

Une SOA doit ainsi proposer des mécanismes d'intégration des services et de gestion de propriétés non fonctionnelles. La Figure 2.3 reprend les différentes couches nécessaires dans une SOA [4] :

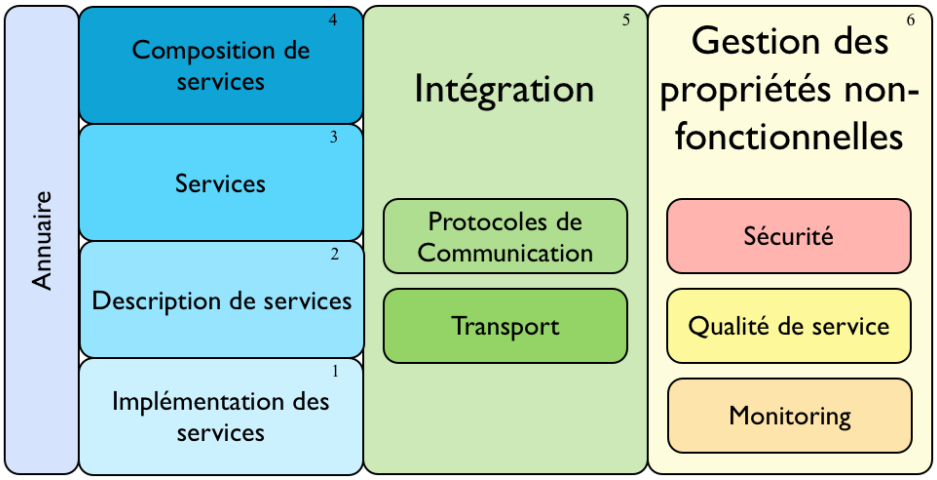

FIGURE 2.3 – Couches nécessaires dans une SOA

Chaque couche participe à la réalisation d'une architecture orientée service :

- **La couche d'implémentation** des services est responsable de la réalisation des fonctionnalités que les services offrent.
- **La couche de description** des services permet de recenser les propriétés fonctionnelles et non fonctionnelles du service. Elle exprime aussi comment le service doit être invoqué.
- **La couche services** contient les services qu'un ensemble de fournisseurs choisit de partager. Les services peuvent être découverts et invoqués. Les descriptions des services sont exportées et publiées dans cette couche.
- Les compositions des services spécifiées dans la couche précédente, sont exposées dans **la couche de compositions de service**. Ces compositions permettent la réalisation de cas d'utilisation spécifiques des services disponibles.
- **La couche d'intégration** permet la communication entre client et fournis-

4. Cette figure est inspirée par le modèle d'architecture proposé par IBM : http ://www.ibm.com/developerworks/webservices/library/ws-soa-design1/.

seurs en opérant des médiations de protocoles si nécessaire ainsi que le routage des requêtes.

- **La couche de gestion des propriétés non fonctionnelles** a pour but d'assurer des propriétés non fonctionnelles comme la sécurité, la qualité de service (qui permet de vérifier que le service offert respecte les propriétés définies dans le SLA) ou encore la surveillance de l'exécution des services.
- **L'annuaire de services** est défini de manière transverse afin de capitaliser et partager les connaissances des différentes couches.

Afin de concevoir et exécuter des applications réalisée à partir de services, il faut composer et administrer ces derniers. Nous nous intéressons désormais à ces deux points. Nous introduisons tout d'abord, la notion de composition des services. Nous présentons les différentes manières de représenter et contrôler les flux d'informations au sein des compositions. Ensuite, nous détaillons les implémentations des services, qui introduisent de nombreux points de variation par rapport au style architectural de l'approche orientée service.

B Compositions de services

Les services sont des unités réutilisables qui doivent être combinées les unes avec les autres afin de réaliser des applications complexes. Nous nommons composition cette combinaison. Une composition consiste à associer un ensemble de services et à contrôler leur ordonnancement ainsi que le flot de messages qu'ils échangent. La composition obtenue peut conduire à un service composite. La composition de service est alors dite récursive car les services composites peuvent eux-même participer à des compositions.

La mise en place d'une composition peut être décomposée en cinq étapes qui visent à passer d'une spécification abstraite de la composition à son exécution :

1. **La définition des fonctionnalités attendues de l'application**. Cette phase permet de modéliser les fonctions attendues de l'application et les interactions entre elles. Des méthodologies existent pour guider cette phase [5].
2. **L'énumération des services.** Selon les fonctionnalités définies à l'étape précédente, il faut identifier les services disponibles qui répondent aux besoins de l'application.
3. **La construction de la composition.** Cette phase permet de sélectionner les services pertinents. Il faut parcourir la liste des services potentiellement utilisables afin de sélectionner les services qui répondent le mieux aux besoins de la composition et qui sont implémentés de manière adaptée. Il est probable que les services sélectionnés à l'étape précédente présentent des différences qui nécessitent, pour les faire communiquer, de mettre en place une médiation qui est réalisée à cette étape. Par exemple, les services peuvent manipuler des types de données différents qu'il faut alors traduire les uns vers les autres.
4. **L'exécution de la composition.** Une fois les services sélectionnés, il faut les invoquer.

La mise en place d'une composition reste fastidieuse dans la mesure où, la plupart du temps, les développeurs sont obligés de réaliser des couches de médiations entre les services, qui n'ont pas été conçus pour fonctionner ensemble. La nécessité du développement manuel rend la mise en place d'une composition longue et complexe. Deux autres facteurs accroissent la complexité du développement d'une composition de services. Tout d'abord, la spécification d'une composition nécessite le plus souvent une expertise métier importante. En outre, le développeur est sans cesse confronté aux difficultés techniques de l'approche à service (description, sélection et invocation des services, etc.). Le développeur doit aussi avoir une bonne expertise technique. Par conséquent, le développeur doit posséder une double expertise qui reste rare.

Il est dès lors nécessaire de développer des outils qui permettent de faciliter aussi bien la spécification de la composition – en permettant de se concentrer sur la logique métier – que son exécution – en facilitant le travail de développement du code de la composition.

5. http ://www.ibm.com/developerworks/rational/library/07/1023_amsden/

Il existe plusieurs manières de réaliser des compositions de services. Le choix entre ces différentes formes se fait notamment à partir du type de coordination des services. Le contrôle des services peut en effet être intrinsèque ou extrinsèque aux services. La liaison des services, c'est-à-dire la manière dont les services interagissent, est un autre point qui caractérise les différentes compositions.

Le moment auquel la liaison intervient est important dans la gestion du dynamisme. Soit la liaison avec un service donné est spécifiée au moment de la conception de la composition, auquel cas la composition est statique, soit la liaison est retardée jusqu'à l'exécution, ce qui permet de sélectionner le service avec lequel s'effectue la liaison parmi les services disponibles à l'exécution. La liaison retardée permet de construire des compositions dynamiques.

Nous présentons maintenant les différents types de compositions en précisant le contrôle des services qu'elles permettent et le type de liaison sur lequel elles reposent.

1 Compositions par procédé : orchestration et chorégraphie de services

Un procédé est une séquence d'actions organisées pour atteindre un but représentée sous la forme d'un graphe orienté. Les noeuds du graphe sont les actions nécessaires pour progresser vers le but attendu. Elles sont réalisées par des services. Les arêtes du graphe correspondent aux messages échangés entre les actions. Les procédés sont spécifiés dans un langage dédié qui est ensuite interprété par un moteur d'exécution. Ce moteur est chargé de gérer les communications avec les services.

On distingue deux types de composition par procédé – l'orchestration et la chorégraphie – en fonction du type de contrôle des services qu'elles proposent. La chorégraphie permet de décrire, d'un point de vue global, la collaboration entre plusieurs acteurs qui participent à un procédé et les messages qu'ils échangent. L'orchestration permet de décrire, pour chaque participant, le procédé qu'il réalise pour jouer son rôle dans la collaboration.

À ce jour, les compositions par procédé ne peuvent être mises en œuvre que pour les services Web, qui sont une implémentation particulière des services que nous présentons dans la suite de ce document. Nous présentons désormais en détail la définition de la chorégraphie et de l'orchestration de services. Pour ce faire, nous nous appuyons sur un procédé simplifié de gestion d'une urgence médicale. Ce procédé met en jeu quatre acteurs – le patient, le samu, une ambulance et un hôpital. Ces quatre acteurs collaborent pour traiter l'urgence médicale.

1.1 Chorégraphie de services

Le W3C[6] définit une chorégraphie de services de la manière suivante :

> *A choreography defines the sequence and conditions under which multiple cooperating independent agents exchange messages in order to perform a task to achieve a goal state*[7].

6. Le *World Wide Web Consortium* est un organisme chargé de promouvoir des normes pour le web. Voir http ://www.w3.org/.

7. http ://www.w3.org/2003/glossary/

Une chorégraphie de services met l'accent sur la collaboration d'un ensemble de services pour atteindre un but donné. Elle se place du point de vue global de tous les participants qui collaborent au sein de la composition. Elle ne présente que les interactions visibles entre les services en ignorant leur fonctionnement interne.

Pour ce faire, elle spécifie les échanges de messages qui doivent intervenir entre les différents services et l'ordre de ces échanges. Une chorégraphie spécifie aussi les dépendances entre les messages, comme, par exemple, leur ordonnancement. En se situant à un haut niveau d'abstraction, la chorégraphie permet de s'abstraire de participants spécifiques en ne précisant pas la manière dont l'activité doit être réalisée. Plusieurs participants concrets peuvent ainsi jouer les rôles définis par la chorégraphie, ce qui garantit la possibilité de la réutiliser. La Figure 2.4 présente la collaboration nécessaire pour prendre en compte une urgence médicale. On remarque que le contrôle est réparti entre les différents participants.

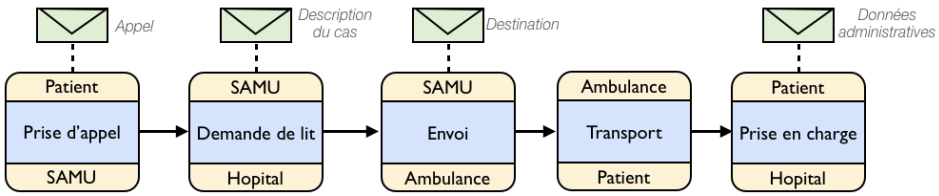

FIGURE 2.4 – Collaboration nécessaire pour prendre en compte une urgence médicale

1.2 Orchestrations de services Le W3C définit une orchestration de services de la manière suivante :

> *An orchestration defines the sequence and conditions in which one service invokes other services in order to realize some useful function. i.e., an orchestration is the pattern of interactions that a service agent must follow in order to achieve its goal*[8].

L'orchestration de services met ainsi l'accent sur l'ordre de l'invocation des services ainsi que les traitements qui interviennent entre ces différents appels.

La Figure 2.5 présente un exemple d'orchestration de services pour le participant « Hôpital » de la chorégraphie que nous avons définie plus haut. On voit sur cette figure que le contrôle est centralisé. L'orchestrateur est chargé d'invoquer les services qui permettent de réaliser les différentes activités qui composent le procédé. Il est aussi chargé de réaliser la médiation entre les services en gérant le contexte d'exécution du procédé. Pour ce faire, il stocke et met à jour les variables globales du procédé. Ainsi, chaque service ignore son contexte d'exécution, ce qui en facilite la réutilisation dans plusieurs orchestrations.

1.3 Synthèse de la composition par procédé La composition par procédé est un moyen intéressant de créer des applications à partir de services car elle per-

8. http ://www.w3.org/2003/glossary/

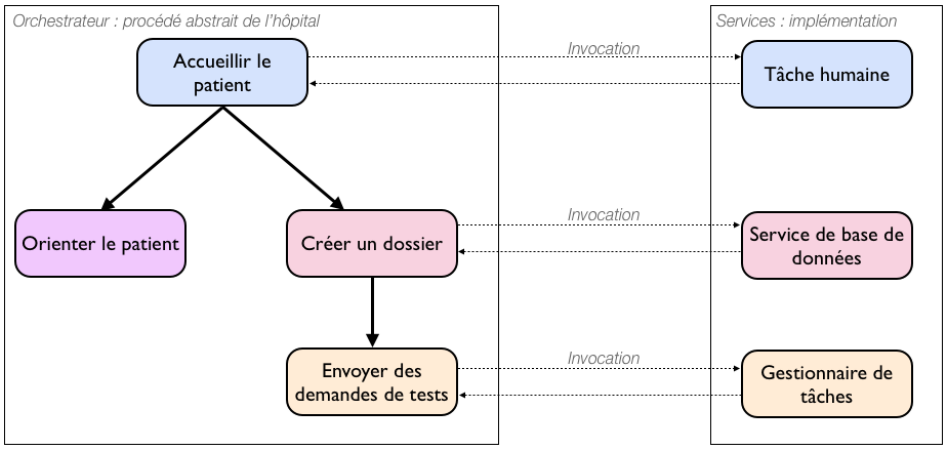

FIGURE 2.5 – Exemple d'orchestration de services

met d'isoler la description du fonctionnement de l'application de sa réalisation. Le procédé est ainsi réutilisable dans la mesure où chaque action qui le compose peut potentiellement être réalisée par plusieurs services qui offrent les mêmes fonctionnalités. Une composition par procédé repose sur le modèle client-serveur : les services sont liés directement, soit entre eux, soit à un orchestrateur.

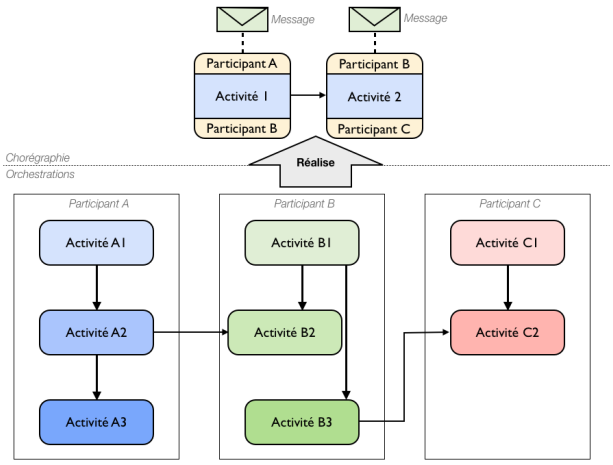

FIGURE 2.6 – Relation entre la chorégraphie et l'orchestration de services

L'orchestration de services est la forme la plus répandue de composition par procédé comme le montre le nombre de moteurs d'orchestration qui existent [9]. L'orchestration et la chorégraphie de services peuvent être utilisées conjointement au sein d'une ou plusieurs organisations. Dans ce cas, l'orchestration permet de décrire les applications disponibles dans une organisation. La chorégraphie permet, elle, de

9. On peut ainsi citer Apache Orchestrator Direction Engine (http ://ode.apache.org/) ou Juju (https ://juju.ubuntu.com/).

spécifier les interactions entre les différents applications. La Figure 2.6 présente la relation entre la chorégraphie et l'orchestration de services.

Enfin, la composition par procédé n'est pas adaptée à toutes les applications. Elle ne permet pas de réaliser des algorithmes complexes et ne permet de spécifier que peu d'interactions entre les services.

2 Composition structurelle

La composition structurelle définit une application comme un assemblage de services. Dans le cadre d'une composition structurelle, le contrôle est interne aux services. Le développeur connaît les services qu'il veut utiliser et leurs interactions. Lors de la livraison de la composition, il fournit ainsi les services et leur logique de coordination décrite, par exemple, sous la forme d'un fichier java. Le principe de la composition structurelle est présenté sur la Figure 2.7.

Dans le cadre de la composition structurelle, le développeur connaît les services composés et peut utiliser des algorithmes complexes. Cependant, la composition structurelle ne permet pas de réutiliser facilement les services puisque le contrôle est précisé à l'intérieur même de ces derniers : chaque service connaît les services dont il dépend.

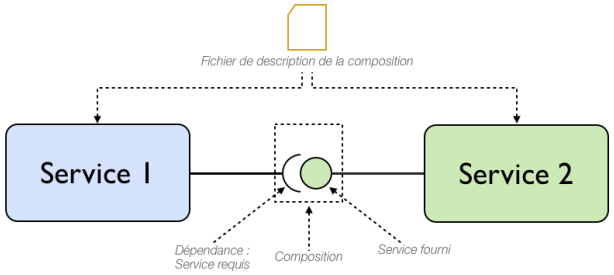

FIGURE 2.7 – Principe de la composition structurelle

Ce type de composition reste peu utilisé. Seule la spécification *Service Component Architecture* (SCA) – que nous présentons plus loin dans ce document – et quelques travaux comme [Marin 2006], permettent de la mettre en œuvre.

3 Un exemple d'autre type de composition : la composition sémantique

La composition sémantique de services est l'opération qui consiste à composer un ensemble de services afin de parvenir à réaliser un but de haut niveau. Par exemple, « gérer les dossiers patients dans un hôpital » est un but qui peut être réalisé grâce à une composition sémantique de services.

La composition sémantique repose sur deux mécanismes. Tout d'abord, l'étiquetage des services de manière à exposer les propriétés qui permettent leur sélection. De nombreux travaux proposent d'étendre des langages de description de services afin d'ajouter, à la description fonctionnelle des services, une description sémantique.

Dans ces travaux, les ontologies sont un élément important pour aider à l'étiquetage des services. Elles permettent de standardiser et de classifier le vocabulaire utilisé pour décrire les services et de représenter les liens entre les différents termes utilisés. Ces liens permettent de raisonner sur les propriétés des services. OWL-S [10] ou le *Web Service Modeling Ontology* [11] sont des ontologies qui permettent d'annoter la description des services Web à l'aide de propriétés sémantiques.

Le second mécanisme nécessaire à la réalisation d'une composition sémantique est la sélection de services pertinents qui permet le passage de la description de la composition sous la forme de buts à sa réalisation. Les ontologies permettent de guider la recherche des services nécessaires à la réalisation d'un but grâce à des mots clés. D'autres approches, comme l'analyse de concept formel (FCA pour Formal Concept Analysis), permettent de classer les services en fonction de leurs propriétés. L'approche FCA est particulièrement efficace dans le cas d'une composition de services hétérogènes et dynamiques [Chollet 2011]. Elle permet de constituer des classes de services équivalents car ils possèdent les mêmes propriétés. Si un service utilisé disparaît, il est possible de se lier à un autre service de la classe auquel il appartient sans pour autant mettre en danger la réalisation de la composition.

L'intérêt de la composition sémantique réside ainsi dans la description abstraite d'une composition sous la forme de buts à atteindre. Cette description est proche de la langue naturelle. La composition est ensuite réalisée à la dernière minute grâce à la sélection et la liaison des services pertinents. La composition sémantique trouve un domaine d'application dans le Web sémantique, qui promeut la description sémantique des services Web mais aussi dans d'autres domaines comme les applications pervasives où l'hétérogénéité et le dynamisme des services obligent une sélection et une liaison fréquentes.

4 Synthèse sur les compositions

Nous avons étudié dans cette section quelques manières de réaliser une application en associant un ensemble de services. Nous avons choisi de présenter les compositions selon le type de contrôle des services sur lequel elles reposent, du type de liaison entre les services qu'elles mettent en jeu et de leur capacité à gérer le dynamisme des services. Cette classification permet de montrer que chaque type de composition répond à un domaine d'application particulier. La composition structurelle, par exemple, est bien adaptée à des domaines de taille restreinte où le développeur contrôle complètement les services. La composition sémantique, elle, peut s'appliquer aux cas où le nombre de services disponibles est très important et lorsque le développeur ne les contrôle pas.

Nous reprenons dans le Tableau 2.1 les caractéristiques des trois types de composition de services que nous avons étudiés, la composition par procédé, la composition structurelle et la composition sémantique.

Il apparaît ainsi que la composition par procédé, qui est la forme de composi-

10. http ://www.w3.org/Submission/OWL-S/
11. http ://www.wsmo.org/

tion la plus répandue, permet la gestion du dynamisme grâce au mécanisme de la liaison retardée. Comme nous l'avons vu, la composition sémantique est, elle aussi, un moyen efficace de gérer l'hétérogénéité et le dynamisme des services. Dans les deux cas, c'est la description de la composition à un haut niveau d'abstraction et la concrétisation des spécifications abstraites à la dernière minute qui permet de s'adapter aux services existants, à leur apparition et à leur disparition. Les deux mécanismes de composition peuvent être associés. La composition par procédé peut ainsi s'appuyer sur la sélection basée sur le FCA pour découvrir et se lier aux services disponibles.

	Composition par procédé	**Composition structurelle**	**Composition sémantique**
Type de contrôle	Extrinsèque	Intrinsèque	Extrinsèque
Type de liaison	Directe : client-serveur	Indirecte : par l'intermédiaire d'un *Broker*	Directe : client-serveur
Gestion du dynamisme	Liaison retardée Ajout de primitives de communication	Non prise en compte	Liaison retardée Ajout de primitives de communication

TABLE 2.1 – Caractérisation de trois types de composition

Le développeur d'une composition se heurte à plusieurs difficultés. Tout d'abord, chaque type de composition met en jeu des langages différents. Il doit donc posséder une connaissance approfondie de chaque type de composition afin d'orienter son choix et des outils qui permettent de les réaliser. Ceci étant, le développeur doit aussi posséder une expertise du métier qui lui permet de spécifier la composition. Enfin, dans le cadre d'une composition de services hétérogènes, le développeur doit connaître en profondeur les technologies de services qu'il est susceptible d'utiliser.

Pour toutes ces raisons, la spécification, et à plus forte raison la réalisation d'une composition de services hétérogènes et dynamiques reste complexe. Si les développeurs disposent parfois d'environnements de développement de compositions, ils sont le plus souvent dédiés aux services Web. De plus, la plupart du temps, ces environnements ne permettent pas de spécifier la totalité d'une composition, c'est-à-dire non seulement ses propriétés fonctionnelles mais aussi ses propriétés non fonctionnelles. Par conséquent, il est nécessaire de proposer un environnement qui simplifie le développement d'une composition de services hétérogènes et dynamiques. Pour ce faire, un tel environnement ne devrait pas nécessiter un temps important d'apprentissage. Il devrait aussi rendre transparente l'utilisation de technologies hétérogènes en prenant en charge leur gestion.

C Implémentations de l'approche orientée service

Nous avons jusqu'ici insisté sur la définition des services, des architectures qui permettent de les composer et des différents styles de composition. Nous nous intéressons désormais aux différentes technologies qui implémentent l'approche à service afin de mettre en évidence leur diversité.

1 Caractérisation des technologies à service

Comme nous le verrons, les technologies à services sont extrêmement hétérogènes, il est ainsi nécessaire de construire un cadre pour les étudier et les comparer. Notre étude des principes et des concepts de l'approche à services permet d'identifier six critères de caractérisation des technologies à services. Ces critères sont :

- **La manière dont les services sont implémentés.**
- **Le type de description de service** proposé par chaque technologie.
- **Les protocoles de communication** mis en jeu par les différentes technologies.
- **Le type de registre de services** utilisé par chaque technologie.
- **La possibilité de composer les services** et le ou les types de composition possibles.
- **La gestion de la liaison du client avec le fournisseur de service** et la possibilité de la gestion du dynamisme.

Nous appliquons ces critères dans les pages qui suivent à quatre technologies à services largement utilisées : les services Web, l'*Universal Plug and Play* et le *Device Profile for Web Service* et les composants orientés services.

2 Services Web

Les services Web sont l'implémentation la plus populaire des services. Nous commençons par en présenter les principes, avant de détailler les technologies qui y sont attachées.

2.1 Principes Selon le W3C, un Service Web est un système logiciel conçu pour permettre l'interaction entre plusieurs machines grâce à un réseau [12]. Les services Web permettent de rendre disponibles sur un réseau des applications informatiques. Ils permettent l'interopérabilité entre des applications indépendantes et implémentées de manière différente. Un service Web possède une interface décrite dans un langage qu'une machine peut traiter. Les autres systèmes interagissent avec lui en fonction de sa description en utilisant des messages au format *Simple Object Access* Protocol (SOAP) [13], habituellement transmis au format XML à l'aide du protocole HTTP [14].

12. http ://www.w3.org/2003/glossary/
13. http ://www.w3.org/TR/soap/
14. http ://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/

Le fournisseur de service peut publier ses services Web dans un annuaire qui permet au client de découvrir et localiser les services qui l'intéressent [15].

La Figure 2.8 présente l'architecture orientée service qui doit être déployée lorsque l'on utilise des services Web. Cette architecture est conforme à l'architecture orientée service que nous avons déjà décrite. Au sein de cette architecture, le service web est une boîte noire pour le client qui n'en connaît pas l'implémentation. De plus, l'architecture dissimule toute la complexité de la gestion des services au client.

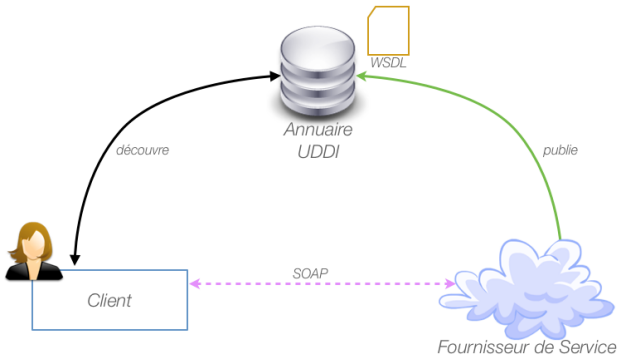

FIGURE 2.8 – SOA pour les services web

Nous détaillons désormais les différentes technologies nécessaires à la mise en place de cette architecture.

2.2 Web-Service Description Language : langage de description des services Web Les services Web sont aujourd'hui très populaires car ils permettent d'utiliser des fonctionnalités déjà existantes sans se soucier de leur implémentation. Seule l'interface fonctionnelle du service présente ainsi un intérêt pour son client dans la mesure où ce dernier est principalement à la recherche d'une fonctionnalité et pas d'une implémentation.

Le *Web Service Description Language* (WSDL) [16] est un langage XML de description des services Web. Un fichier WSDL décrit les fonctionnalités d'un service, les paramètres et les types de retour de ces fonctionnalités. Sur ce point, il est l'équivalent d'une signature de méthode dans un langage de programmation. Un fichier WSDL décrit aussi comment appeler le service, c'est-à-dire l'adresse à laquelle il est localisé et le protocole qui doit être utilisé pour le contacter. Un fichier WSDL peut être traité par une machine et lu par un client afin de découvrir les fonctionnalités qu'offrent un service. Enfin, le fichier WSDL ne donne pas d'indications sur la manière dont le service est implémenté.

15. Le format *Universal Description and Discovery Integration* (UDDI) est UDDI est une spécification d'annuaire de services basé sur XML. Il n'est pas dépendant d'une plateforme particulière. Voir https ://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm Cependant, UDDI est peu utilisé, c'est pour cette raison que nous ne le détaillons pas. Chaque fournisseur de services met en place son propre annuaire.

16. http ://www.w3.org/TR/wsdl

Un fichier WSDL ne décrit qu'un ensemble limité de propriétés fonctionnelles d'un service, même si des travaux comme ceux de Carminatti *et al.* [Carminati 2006] tentent d'étendre WSDL afin de prendre en compte les propriétés comme la sécurité. Les spécifications *WS-Security*[17] et *WS-Transaction*[18] proposent elles aussi d'étendre l'ensemble des propriétés décrites par WSDL, respectivement avec des propriétés de sécurité et des informations sur les transactions.

Un fichier WSDL est structuré en deux parties :

- **La description abstraite**, qui définit les types de données abstraits manipulés par le service web et les paramètres et les types de retour de chaque fonctionnalité du service web. La description abstraite regroupe la description des *opérations*, c'est-à-dire des fonctionnalités offertes par le service web, des *messages*, qui décrivent les informations nécessaires pour exécuter une *opération* et la description des *port types* qui associent les *opérations* et les *messages* qui leur sont nécessaires.
- **La description concrète**, qui décrit le service comme une collection de ports. Un port définit l'adresse à laquelle le service web peut-être contacté. Un port est habituellement représenté sous la forme d'une URL. La description concrète décrit aussi le *binding*, c'est-à-dire le protocole à utiliser pour invoquer le service web.

Le code suivant présente un exemple de fichier WSDL.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap
       /">
3    <wsdl:types>
4      <s:schema elementFormDefault="qualified" targetNamespace="http
           ://www.webservicex.net">
5        <s:element name="VerifyAddress">
6          <s:complexType>
7            <s:sequence>
8              <s:element minOccurs="0" maxOccurs="1" name="City"
                   type="s:string"/>
9              <s:element minOccurs="0" maxOccurs="1" name="State"
                   type="s:string"/>
10             <s:element minOccurs="0" maxOccurs="1" name="Zip" type
                   ="s:string"/>
11           </s:sequence>
12         </s:complexType>
13       </s:element>
14       <s:element name="VerifyAddressResponse">
15         <s:complexType>
16           <s:sequence>
17             <s:element minOccurs="0" maxOccurs="1" name="
                   VerifyAddressResult" type="tns:ArrayOfAddress"/>
18           </s:sequence>
19         </s:complexType>
20       </s:element>
21       <s:complexType name="ArrayOfAddress">
22         <s:sequence>
23           <s:element minOccurs="0" maxOccurs="unbounded" name="
                 Address" nillable="true" type="tns:Address"/>
24         </s:sequence>
25       </s:complexType>
26       <s:complexType name="Address">
27         <s:sequence>
28           <s:element minOccurs="1" maxOccurs="1" name="Rank" type
                 ="s:int"/>
30 |_____________________
```

17. https ://www.oasis-open.org/committees/tc_home.php ?wg_abbrev=wss
18. http ://www.ibm.com/developerworks/library/specification/ws-tx/

```
29            <s:element minOccurs="1" maxOccurs="1" name="Quality"
                  type="s:double"/>
31            <s:element minOccurs="0" maxOccurs="1" name="City" type
                  ="s:string"/>
32            <s:element minOccurs="0" maxOccurs="1" name="State" type
                  ="s:string"/>
33            <s:element minOccurs="0" maxOccurs="1" name="
                  ZipRangeFrom" type="s:string"/>
34            <s:element minOccurs="0" maxOccurs="1" name="ZipRangeTo"
                   type="s:string"/>
35          </s:sequence>
36        </s:complexType>
37        <s:element name="ArrayOfAddress" nillable="true" type="tns:
              ArrayOfAddress"/>
38      </s:schema>
39    </wsdl:types>
40    <wsdl:message name="VerifyAddressSoapIn">
41      <wsdl:part name="parameters" element="tns:VerifyAddress"/>
42    </wsdl:message>
43    <wsdl:message name="VerifyAddressSoapOut">
44      <wsdl:part name="parameters" element="tns:
            VerifyAddressResponse"/>
45    </wsdl:message>
46    <wsdl:portType name="USAddressVerificationSoap">
47      <wsdl:operation name="VerifyAddress">
48        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/
              wsdl/">Verify US Address</wsdl:documentation>
49        <wsdl:input message="tns:VerifyAddressSoapIn"/>
50        <wsdl:output message="tns:VerifyAddressSoapOut"/>
51      </wsdl:operation>
52    </wsdl:portType>
53    <wsdl:service name="USAddressVerification">
54      <wsdl:port name="USAddressVerificationSoap" binding="tns:
            USAddressVerificationSoap">
55        <soap:address location="http://www.webservicex.net/
              usaddressverification.asmx"/>
56      </wsdl:port>
57    </wsdl:service>
58  </wsdl:definitions>
```

Exemple de code 2.1 – Exemple de fichier

Les fichiers WSDL permettent de générer automatiquement le code nécessaire côté client pour appeler le service. Ils sont le plus souvent publiés dans un annuaire afin de permettre la découverte des services. Cet annuaire peut être au format UDDI.

2.3 Simple Object Access Protocol (SOAP) : Communication entre les services Web Le *Simple Object Access Protocol* (SOAP) est un protocole basé sur XML. Il permet d'effectuer des appels distants aux fonctionnalités d'un service web sans tenir compte de la plateforme sur laquelle le service web s'exécute. SOAP est une recommandation du W3C. Depuis sa version 1.2 SOAP n'est plus un acronyme car la notion d'objet est devenue obsolète.

Le transport des messages SOAP se fait le plus souvent au moyen du protocole HTTP, ce qui garantit la comptabilité de SOAP avec les technologies antérieures. SOAP peut aussi être utilisé avec d'autres protocoles comme *Simple Network Management Protocol* (SMNP) [19]. Les messages SOAP sont composés de deux parties :

- **Une enveloppe**, qui contient des informations sur le message pour permettre son transport et son traitement.

19. SMNP est protocole standard pour gérer des appareils au travers du réseau Internet. Voir http ://www.snmplink.org/.

- **Le corps du message**, qui contient la méthode à invoquer et, si nécessaires, les paramètres qui y sont associés.

Le code suivant est un exemple de message SOAP :

```
POST /nbapi/event HTTP/1.1
Content-Length: 309
SOAPAction: ""
Content-Type: text/xml;charset=utf-8
Host: ems-sv258:1774
Connection: Keep-Alive


<?xml version="1.0" ?>
<soapenv:Envelope
        xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:ns1="http://cisco.com/mwtm">
    <soapenv:Body>
        <ans:getNote xmlns:ans="http://cisco.com/mwtm">
            <eventID>1000</eventID>
        </ans:getNote>
    </soapenv:Body>
</soapenv:Envelope>
```

Exemple de code 2.2 – Exemple de message SOAP

SOAP est un langage flexible qui peut être étendu au niveau de l'enveloppe des messages. C'est aussi un langage simple qui ignore les propriétés non fonctionnelles comme la sécurité et les transactions. Néanmoins, SOAP a de piètres performances car le langage XML est très verbeux.

2.4 WS-BPEL Dans la mesure où chaque service Web offre une fonctionnalité particulière, il est probable qu'une fonctionnalité complexe, comme la gestion des lits dans un hôpital, mobilise plusieurs services Web. Il faut donc pouvoir représenter leur composition, qui s'effectue sous la forme d'un procédé.

Le *Web Service-Business Process Execution Language* (WS-BPEL) est une spécification de l'OASIS qui permet de décrire les interactions des services Web au sein d'une composition sous la forme d'une orchestration. Sa syntaxe repose sur XML. Depuis 2007, WS-BPEL en est à sa version 2.0.

Les spécifications des orchestrations à l'aide de WS-BPEL sont de deux types :

- **Les spécifications abstraites** ne sont pas destinées à être exécutées. Elles décrivent les échanges de messages entre les services mais ne précisent pas le comportement des participants.
- **Les spécifications concrètes** sont destinées à être exécutées. Les participants sont identifiés, tout comme les messages qui sont échangés. Il faut aussi préciser comment les erreurs et les exceptions sont gérées.

Dans les deux cas, un procédé est composé d'activités. Les activités peuvent être simples ou complexes. Les activités simples sont notamment de type *invoke* pour invoquer un service ou *receive* pour attendre le retour d'un service par exemple. Les activités complexes sont faites par composition d'autres activités.

WS-BPEL permet aussi d'utiliser les structures de contrôle habituelles des langage de programmation afin de contrôler le procédé. Il n'y a pas à proprement parler d'échanges de données entre les activités. Il faut passer par des affectations

de variables qui sont transmises d'une activité à l'autre. La plupart des outils qui implémentent WS-BPEL permettent de définir les variables et de leur affecter des valeurs. La Figure 2.9 présente un exemple de procédé WS-BPEL dans l'outil *Java Business Process Management* (JPBM) [20].

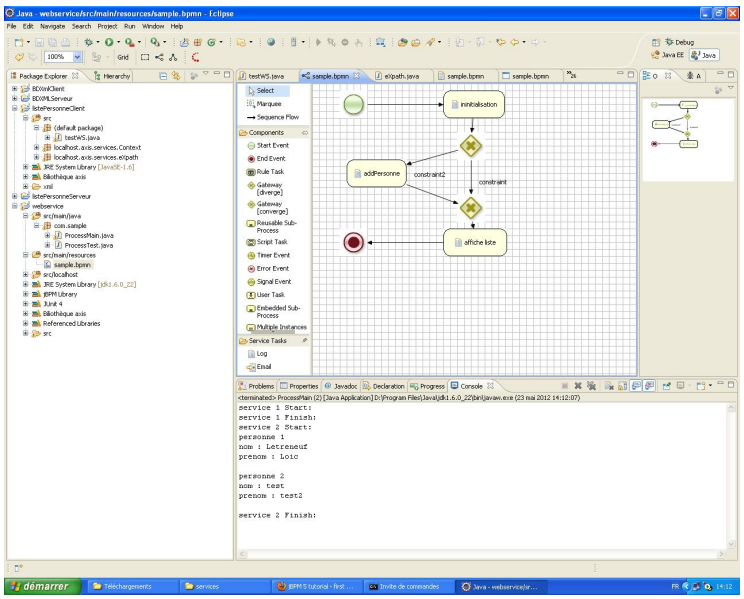

FIGURE 2.9 – Exemple de procédé WS-BPEL

Les procédés BPEL doivent être exécutés grâce à des moteurs dédiés comme *Oracle BPEL Process Management* [21] ou Apache ODE [22]. Ces moteurs sont centralisés et exécutent les procédés comme des orchestrations. Ils sont responsables de la gestion des variables du procédé et de la communication entre les services concrets.

Aujourd'hui, WS-BPEL reste limité car il ne prend pas en compte les propriétés comme la qualité de service par exemple, d'où de nombreuses tentatives de l'étendre.

2.5 Web-Service Choreography Description Language (WS-CDL) Aujourd'hui, peu de langages de spécification des chorégraphies existent et peu de moteurs d'exécution permettent de les interpréter. WS-CDL, qui n'est encore qu'une proposition de langage, est ainsi le langage le plus répandu même s'il reste très peu utilisé [23]. WS-CDL permet de décrire des chorégraphies de services Web. C'est une proposition faite par le W3C. Depuis 2004, WS-CDL en est à sa version 1.

WS-CDL a pour but de permettre la description de la collaboration entre plusieurs organisations. Un fichier WS-CDL divise la spécification d'une chorégraphie en deux parties :

20. http ://www.jboss.org/jbpm
21. http ://www.oracle.com/technology/products/ias/bpel/index.html
22. http ://ode.apache.org/
23. http ://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/

- **La partie abstraite** exprime les rôles et les services nécessaires pour la chorégraphie.
- **La partie concrète** spécifie un procédé, c'est-à-dire l'ensemble des interactions entre les services dans la chorégraphie.

WS-CDL Eclipse est ainsi un des rares moteurs qui existe pour exécuter les chorégraphies de services Web [24].

2.6 Synthèse sur les services Web Les services Web sont aujourd'hui la technologie de services la plus populaire. Ils sont décrits au moyen de fichiers WSDL, publiés et découverts dans des annuaires UDDI. L'interaction entre les services et avec l'annuaire de service se fait au moyen de messages au format SOAP échangés le plus souvent à l'aide du protocole HTTP.

Toutes ces technologies reposent sur XML et fonctionnent bien ensemble. Cependant, elles ignorent toutes les propriétés non fonctionnelles comme la sécurité, d'où la nécessité de les étendre afin de mieux prendre en compte les besoins des clients. Aujourd'hui, de nombreuses propositions d'extension existent et aucun standard ne fait office de loi en la matière.

Le Tableau 2.2 récapitule les éléments que les services Web mettent en jeu.

But	Technologie
Description	WSDL
Publication – Découverte	Annuaire UDDI
Communication entre les services	SOAP avec HTTP
Composition des services Web	WS-BPEL - WS-CDL

TABLE 2.2 – Technologies utilisées pour les services web

À ce jour, la technologie des services Web ne permet pas de mettre en œuvre le dynamisme des architectures orientées services dans la mesure où les compositions de service identifie les services à invoquer. Il faut donc envisager la possibilité d'introduire un plus grand niveau de généricité dans la spécification des compositions de services Web.

3 Universal Plug and Play (UPnP) et Device Profile for Web Service (DPWS)

En étudiant les services Web, nous nous sommes principalement intéressés à l'utilisation des logiciels comme services. Dans la mesure où une application pervasive met en jeu non seulement des logiciels mais aussi un grand nombre de dispositifs (capteurs, téléphone mobile, etc.), il est nécessaire de pouvoir exposer ces appareils eux-mêmes comme services. Il est aussi nécessaire de gérer le dynamisme propre aux appareils qui rejoignent et quittent le réseau de manière aléatoire. Souvent, la gestion du dynamisme est une gestion d'évènements.

24. http ://sourceforge.net/projects/wscdl-eclipse/

Deux technologies orientées service dédiés aux appareils permettent de mettre en œuvre une telle gestion. Il s'agit de l'Universal Plug and Play (UPnP) et du Device Profile for Web Service (DPWS).

3.1 *Universal Plug and Play* (UPnP) UPnP est une initiative industrielle gérée par l'UPnP forum [25] qui spécifie un ensemble de protocoles réseaux. UPnP vise à permettre la communication d'appareils dans un réseau domestique. À ce jour, UPnP en est à sa version 1.1.

UPnP fonctionne sur tous les appareils supportant l'*Internet Protocol* (IP) et repose sur des protocoles de communication standardisés (TCP/IP [26], HTTP, SOAP...). Il est indépendant d'un système d'exploitation ou d'un langage de programmation. UPnP permet ainsi de connecter des appareils de constructeurs variés. Le nom UPnP provient du *plug and play* [27], lorsque les appareils joignent le réseau, ils signalent leur présence, et peuvent être immédiatement utilisés.

UPnP définit une architecture qui met en jeu des points de contrôle et des interactions entre les différents appareils. Tous les appareils possèdent une adresse IP : chaque appareil implémente un client DHCP et recherche un serveur DHCP lorsqu'il se connecte au réseau afin d'obtenir une adresse IP. S'il n'y a pas de DHCP, l'appareil se donne une adresse à lui-même. Tous les services offerts par un appareil possèdent une description qui en précise les fonctionnalités ainsi que toutes les variables nécessaires pour modéliser l'état du service à l'exécution. C'est le point de contrôle qui maintient la représentation de l'état du service.

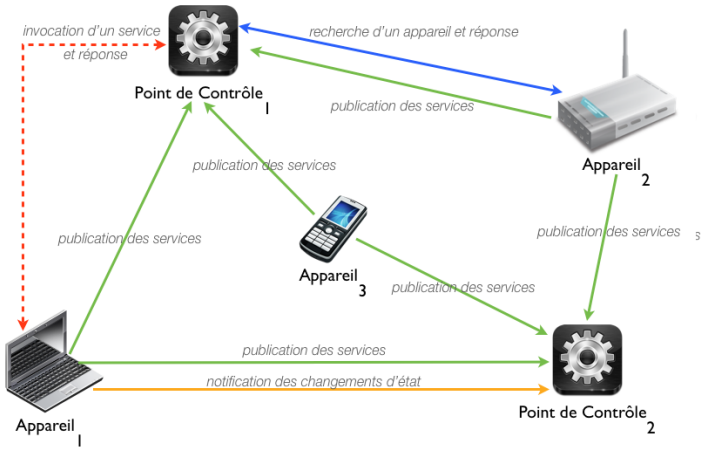

FIGURE 2.10 – Architecture UPnP

La Figure 2.10 présente le schéma de l'architecture UPnP. On peut noter que

25. http ://upnp.org/

26. Le sigle TCP/IP désigne la suite des protocoles utilisés pour la communication sur l'Internet. HTTP, par exemple, appartient à cette suite.

27. Le *plug and play* consiste à pouvoir brancher un appareil à un ordinateur et à pouvoir l'utiliser immédiatement sans devoir redémarrer la machine.

quatre interactions sont possibles entre les appareils et les points de contrôle :

- **Le *Simple Service Discovery Protocol* (SSDP)** [28] **permet aux services de se découvrir les uns les autres** : lorsque l'appareil se connecte au réseau, il communique la liste de ses fonctionnalités aux points de contrôle.
- **Les points de contrôle peuvent rechercher les appareils** qui les intéressent sur le réseau.
- **Les points de contrôle peuvent aussi envoyer des ordres aux appareils** afin d'invoquer les services qu'ils fournissent et récupérer les résultats des actions effectuées. Les ordres sont envoyés au format SOAP. Le point de contrôle peut aussi, si nécessaire, mettre à jour la représentation de l'état du service invoqué.
- **Les services peuvent notifier les points de contrôle du changement de leur état**.

UPnP est efficace pour construire un environnement domestique pour l'exécution d'une application dynamique et hétérogène. Il permet de gérer le dynamisme des services et l'hétérogénéité de leurs constructeurs. Néanmoins, UPnP ne repose pas sur une architecture orientée service standard. Ceci complique ainsi la composition de services UPnP avec d'autres types de service et nécessite de créer des architectures dédiées [Bottaro 2007].

Enfin, UPnP ne peut pas fonctionner hors d'un réseau domestique car il ne prévoit pas la gestion de la sécurité. Par exemple, UPnP ne prévoit pas de mécanisme d'authentification des utilisateurs [29].

3.2 Device Profile for Web Sservice (DPWS) Le but de DPWS est similaire à celui d'UPnP. Cependant, DPWS respecte complètement la spécification des services Web. Il permet ainsi de publier, découvrir et invoquer les appareils de la même manière que les services Web. DPWS en est à sa version 1.1. C'est une spécification proposée à l'origine par Microsoft et aujourd'hui maintenue par l'OASIS [30].

Afin de prendre en compte les caractéristiques spécifiques des appareils et la gestion du dynamisme, DPWS ajoute plusieurs types de services à l'architecture orientée services proposée pour les services Web. Tout d'abord, DPWS divise les services en *hosting service* (les appareils, responsable de la notification de leur présence) et les *hosted services* (des services fonctionnels hébergés par les *hosting service*). Les appareils exécutent ces deux services.

Nous avons vu que la gestion du dynamisme nécessitait l'ajout de primitives de communication au schéma de SOA. DPWS rajoute ces primitives au travers de trois types de services que l'appareil utilise pour signifier sa présence et son retrait :

- ***Discovery service*** **:** l'appareil utilise ce service pour signaler sa présence et

28. http ://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf

29. Des extensions non officielles de UPnP, comme le UPnP – User Profile (UPnP-UP, voir http ://www.upnp-up.org/) existent néanmoins pour prendre en compte certaines propriétés comme l'authentification.

30. http ://docs.oasis-open.org/ws-dd/ns/dpws/2009/01

découvrir d'autres services. Il s'appuie sur la spécification *WS-Discovery* des services Web [31].

- ***Metadata exchange service*** **:** permet d'accéder aux métadonnées du *hosting service* et des *hosted services* qu'il héberge. Il s'appuie sur la spécification *WS-Metadata Exchange* des services Web [32].
- ***Publish subscribe eventing service*** : permet de s'abonner aux évènements de l'appareil. Il s'appuie sur la spécification *WS-Eventing* des services Web [33].

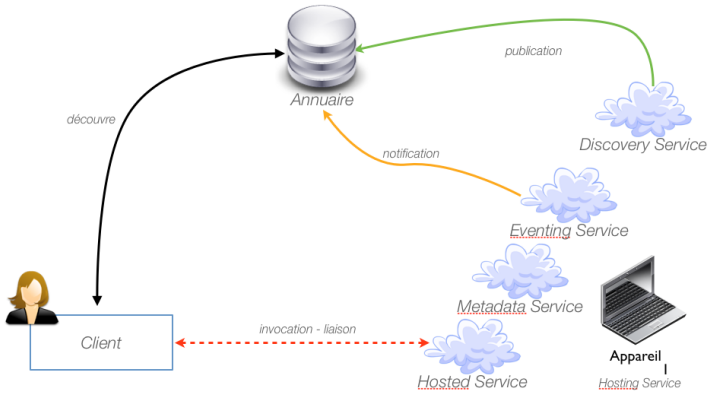

FIGURE 2.11 – Architecture DPWS

Pour les constructeurs, l'exposition de leurs appareils comme des services DPWS passe par la réalisation de code qui permet de faire le pont entre le code, souvent propriétaire, des appareils et le code DPWS. Le services DPWS sont décrits dans un fichier WSDL étendu. La figure 2.11 présente un exemple d'architecture utilisant des services DPWS.

DPWS est efficace pour construire des architectures orientées services qui doivent utiliser des appareils. En étant compatible avec les services Web, DPWS permet d'intégrer de manière transparente des appareils à des SOA déjà construites. Enfin, DPWS peut profiter des spécifications propres au services Web pour la gestion de propriétés non fonctionnelles, comme la sécurité en s'appuyant sur la spécification *WS-Security*.

3.3 Synthèse de UPnP et DPWS UPnP et DPWS sont aujourd'hui en passe de devenir des standards pour les applications à services dans le cadre domotique. Nous reprenons les caractéristiques de chaque technologie dans le Tableau 2.3.

31. http ://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf
32. http ://www.ibm.com/developerworks/library/specification/ws-mex/
33. http ://www.w3.org/Submission/WS-Eventing/

	UPnP	DPWS
Spécification	• Description des services spécifiques à UPnP • Pas de propriété non fonctionnelle	• WSDL étendu • Propriétés non fonctionnelles : sécurité
Implémentation	• Nombreux langages et constructeurs • Création d'un *proxy* du côté client	• Nombreux langages et constructeurs • Création d'un *proxy* du côté client
Découverte	• Découverte dynamique des services	• Découverte dynamique des services
Compositions de service	• Client/serveur	• Client/serveur
Communication	• SOAP • Notification et abonnement aux évènements des services	• SOAP • Notification et abonnement aux évènements des services
Liaison	• Directe	• Directe

TABLE 2.3 – Comparaison entre DPWS et UPnP

Aujourd'hui, UPnP et DPWS coexistent. Les deux technologies poursuivent le même but, exposer des appareils comme des services. DPWS s'inspire ainsi fortement de UPnP et veut même en être la nouvelle version. Il faut cependant noter que chaque technologie apporte une réponse à la gestion du dynamisme. Ces deux technologies restent pour le moment peu implémentées ou cantonnées à de petites applications. L'absence de gestion de propriétés non fonctionnelles par UPnP, par exemple, interdit de manipuler des données ou des appareils sensibles.

4 Les modèles à composants

4.1 Principes L'approche orientée composants, notée CBSE pour *Component-Based Engineering*, définit la réalisation d'une application comme l'assemblage de composants. Elle est antérieure à l'approche à services mais peut, dans certains cas, lui être associée [Szyperski 2002]. Elle repose sur deux principes :

- **La séparation des préoccupations** : chaque composant contient un ensemble de données ou offre un ensemble de fonctionnalités spécifiques et indépendantes. Les composants sont ainsi modulaires et faiblement couplés.
- **La réutilisation des composants** : les composants doivent pouvoir participer à plusieurs assemblages, c'est-à-dire à plusieurs applications afin de permettre leur développement rapide à partir de fonctionnalités déjà existantes.

Les composants sont ainsi des entités conçues principalement pour être réutilisées. Par conséquent, leur implémentation importe peu, le client d'un composant n'est intéressé que par les fonctionnalités qu'il offre. Nous présentons ici brièvement les composants, qui peuvent être associés aux services et apportent des réponses aux problèmes de l'hétérogénéité et du dynamisme.

4.2 Définition d'un composant La notion de composant reste aujourd'hui l'objets de discussion. Dans [Szyperski 2003], Clemens Szyperski soutient qu'un composant possède les cinq propriétés suivantes :

- **Un composant est une unité de déploiement** : c'est une brique logicielle exécutable par une machine.
- **Un composant encapsule du code et des données** qui permettent de réaliser ses fonctionnalités.
- **Les fonctionnalités d'un composants sont exposées au travers d'interfaces**. Le code et les données encapsulées sont ainsi invisibles de l'extérieur du composant.
- **Un composant est fait pour participer à une composition** : il dépend d'un ensemble d'autres composants et peut lui-même être nécessaire à un ensemble de composants.
- **Un composant ne possède pas d'états**, il est ainsi nécessaire de maintenir un contexte d'exécution par instance de composants.

Un composant est ainsi une brique logicielle qui encapsule des données et des fonctionnalités : le composant cache les données qu'il contient et l'implémentation de ses fonctionnalités. Pour un client, la communication avec un composant se fait ainsi au moyen d'une interface qui répertorie les fonctionnalités qu'offre le composant. Un composant implémente une interface qui est décrite au moyen d'un langage de description d'interface (IDL pour *Interface Description Language*). Un IDL est indépendant d'un langage donné ce qui permet au client d'ignorer l'implémentation du service [34].

34. Le *Corba Component Model* (CCM) est un exemple de spécification qui décrit l'architecture d'une application faite de composants

De plus, une interface peut être implémentée par plusieurs composants. Par conséquent, des composants qui implémentent la même interface peuvent être substitués les uns aux autres puisqu'ils offrent les mêmes fonctionnalités.

Les composants communiquent entre eux au moyen de connecteurs qui permettent de spécifier les fonctionnalités offertes et les fonctionnalités requises par chaque composant. Les dépendances entre les composants peuvent être logiques – elles portent alors sur les fonctionnalités dont les composants ont besoin pour s'exécuter – ou physiques – elles portent sur le code nécessaire à un composant pour s'exécuter.

Un composant peut être instancié plusieurs fois. Afin de favoriser la réutilisabilité des composants, les instances ne maintiennent pas d'informations sur leur état. Chaque composition doit ainsi maintenir un contexte d'exécution pour chaque instance de composants qu'elle utilise dans la mesure où les instances de composants forment la logique métier d'une application.

Chaque approche à composants fournit un modèle de composants – la définition de ce qu'est un composant dans l'approche – ainsi qu'un langage de composition nommé ADL pour *Architecture Description Language*. Chaque approche fournit aussi une plateforme d'exécution chargée de maintenir le contexte des instances de composants et de procéder à leur composition. Chaque approche permet ainsi de gérer le cycle de vie complet d'une application et notamment le déploiement et l'instanciation des composants, ce que ne traite pas l'approche à services. La composition des composants ainsi que la gestion de leur cycle de vie sont réalisées par une machine d'exécution.

Au sein d'une approche à composants, la gestion du cycle de vie d'un composant ainsi que de ses propriétés non-fonctionnelles est souvent confiée à un conteneur. Les développeurs des composants séparent en effet souvent les préoccupations fonctionnelles et non fonctionnelles. Les conteneurs encapsulent ainsi le code fonctionnel et sont chargés d'instancier et détruire les composants ainsi que de mettre en œuvre des propriétés non fonctionnelles comme la gestion de la sécurité ou des transactions.

Chaque approche à composant propose ainsi une implémentation spécifique des conteneurs. Tel est le cas de la plateforme .Net de Microsoft [35] ou bien des *Enterprise Java* Beans, aujourd'hui maintenus par Oracle [36]. La plupart du temps, les conteneurs se concentrent sur un ensemble restreint de propriétés non-fonctionnelles même si certaines approches, comme Fractal [37], permettent l'utilisateur de définir les propriétés non-fonctionnelles qui intéressent. Enfin, il existe des bibliothèques de code non-fonctionnel dans la mesure où certaines propriétés non-fonctionnelles sont récurrentes.

La Figure 2.12 reprend le principe de l'utilisation des conteneurs et de l'assemblage de composants.

35. http ://www.microsoft.com/net
36. http ://www.oracle.com/technetwork/java/javaee/ejb/index.html
37. http ://fractal.ow2.org/

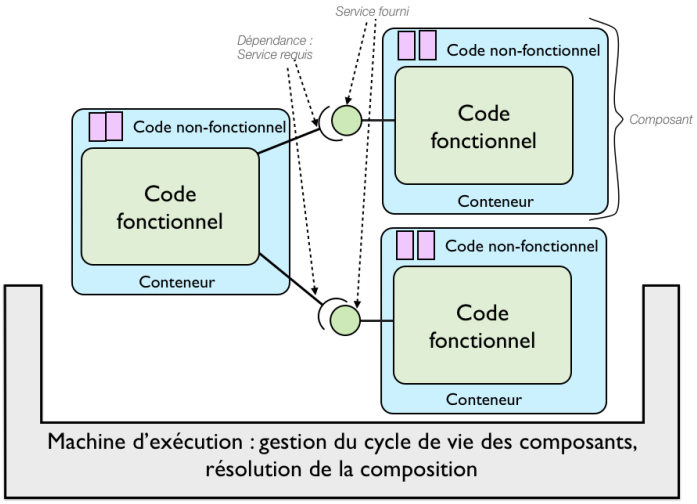

FIGURE 2.12 – Utilisation des conteneurs et de l'assemblage de composants

4.3 Composants orientés services Le développement d'applications dynamiques et sensibles au contexte a entraîné la création de modèles à composants orientés services tout d'abord définis dans [Cervantes 2004] et récemment utilisés dans [Gama 2009]et [Maurel 2011] par exemple. En associant les avantages de l'approche à composants et de l'approche à services, les composants orientés services visent ainsi à faciliter le développement des applications à composants et à proposer des mécanismes de gestion du dynamisme. Dans [Cervantes 2004], Cervantes note six propriétés des composants orientés services :

- **Les services sont des fonctionnalités qui peuvent être consommées.** Ils sont ainsi un ensemble d'opérations réutilisables.
- **Les services sont décrits par des interfaces qui précisent leurs propriétés utiles pour la composition.** Ces propriétés sont notamment les propriétés fonctionnelles du service et certaines propriétés non fonctionnelles comme une description sémantique. Les interfaces doivent indiquer les dépendances du services vis-à-vis d'autres services. Les interfaces constituent des contrats.
- **Les composants implémentent des contrats et respectent ainsi les contraintes définies par une ou plusieurs interfaces.** Les composants communiquent entre eux au moyen des services qu'ils offrent.
- **Les descriptions (ou contrats) permettent la substituabilité des composants.** Les composants qui implémentent les mêmes interfaces peuvent être substitués les uns aux autres.
- **Les principes de SOC sont utilisés pour résoudre les dépendances entre composants.** Les services offerts par un composant sont publiés dans un registre et peuvent ainsi être découverts et invoqués. Le registre est un élément primordial de la gestion du dynamisme car il maintient la représentation

des services réellement disponibles et permet ainsi de les invoquer.

- **Une composition est décrite sous la forme d'une spécification de services.** Les spécifications sont utilisées afin de choisir les composants nécessaires. La liaison avec les services n'a pas besoin d'être explicitée, elle est inférée des dépendances entre les services.

L'approche à composants orientés services présente ainsi l'avantage de prendre en compte le dynamisme des applications et d'associer les avantages de l'approche à composants – la gestion du cycle de vie des composants et la composition structurelle – et de l'approche à services – la substituabilité des services ainsi que la liaison retardée. OSGi, l'acronyme de *Open Service Gateway Initiative Framework* est l'une des formes les plus courantes de composant orienté service. Il s'agit d'une spécification de plateforme de services pour JAVA maintenue par l'OSGi Alliance [38].

5 Service Component Architecture (SCA)

SCA est une technologie issue d'initiatives industrielles pour la réalisation d'applications à partir de composants sans avoir à tenir compte de leur implémentation. SCA met en pratique les principes qui guident une SOA. Elle est est spécifiée et implémentée pour un grand nombre de technologies hétérogènes (EJB, BPEL, services Web, etc.). Web Sphere ou Tuscany [39] sont des exemples d'implémentation de SCA.

Dans SCA, les données sont exprimées sous la forme de *Service Data Objects* (SDO), une spécification qui permet de décrire des structures de données indépendante d'une plateforme spécifique donnée [40]. SDO est ainsi un langage pivot qui permet de s'abstraire de l'hétérogénéité des technologies auxquelles s'applique SCA.

SCA repose sur des composants décrits par un fichier XML. Ils sont déployés dans un système SCA qui ne contient que des composants qui appartiennent à une seule organisation. Les composants peuvent être regroupés dans des modules dédiés à un ensemble de fonctionnalités particulières. Ils sont de deux types :

- Les **composants** contiennent les fonctionnalités offertes par un module. Les fonctionnalités sont exposées sous la forme de services. Un composant est une instance d'une **implémentation**. Un composant donne une valeur aux **propriétés** définies dans l'implémentation. L'implémentation peut, par exemple, être rédigée en Java ou en BEPL. On peut accéder à distance aux services offerts par les composants. Les services peuvent être offerts à des composants hors du module au travers de **points d'entrée**. Les composants peuvent aussi réutiliser des fonctionnalités offertes par d'autres composants grâce à des **références**.
- Les **composites** sont un assemblage de composants. Ils peuvent être assemblés de manière récursive : un composite peut contenir des composites. Un

38. Voir http ://www.osgi.org/Main/HomePage. D'autres composants orientés services existent comme injected Plain Old Java Object (iPOJO), une implémentation de OSGi R4 ou Java Management Extension(JMX), spécification maintenue par Oracle. Voir http ://www.oracle.com/technetwork/java/javase/tech/best-practices-jsp-136021.html.

39. Voir http ://tuscany.apache.org/ et http ://www-01.ibm.com/software/websphere/

40. http ://www.oasis-opencsa.org/sdo

composite encapsule les composants qu'il contient. Il peut exposer les services qu'il offre, ses propriétés et ses références. Ces informations proviennent des composants qu'il contient. Les composites sont décrits grâce au *Service Component Definition Language* (SCDL) [41].

SCA permet la déclaration de propriétés non fonctionnelles comme la qualité de service ou la sécurité. Chaque composant est ainsi associé à un ensemble de propriétés non fonctionnelles. Les applications sont faites d'un assemblage de composants, ce sont donc elles-mêmes des composites. Néanmoins, SCA ne permet pas d'administrer les applications faites d'un assemblage de composants. Seinturier *et al.* [Seinturier 2012] proposent un cadriciel pour SCA qui répond à ce manque.

SCA est ainsi une spécification qui permet de mettre en œuvre la composition structurelle. Les compositions sont décrites dans un langage spécifique qui sépare les services et les liaisons. SCA permet d'ignorer le langage d'implémentation des composants. Cependant, dans la mesure où les liens entre les composants sont définis dès la conception de la composition, SCA ne permet pas d'intégrer des composants dynamiques et de modifier les compositions à l'exécution.

41. http ://tinyurl.com/nf27quh

D Synthèse

Cette première section nous a permis de définir l'approche à services, de définir l'architecture qui permet de l'implémenter et de faire un état des lieux des technologies qui permettent aujourd'hui de la mettre en œuvre.

De cette étude, on voit que l'approche à services est un moyen efficace et largement utilisé de réaliser des applications à partir de services déjà existants, qui peuvent être réutilisés. La gestion du dynamisme, permise notamment par le mécanisme de la liaison retardée, fait de l'approche à services une approche prometteuse pour la réalisation d'application pervasives, qui sont souvent sensibles au contexte.

Néanmoins, notre analyse conduit à constater l'hétérogénéité des technologies à services et des capacités qu'elles offrent. Ainsi, les services Web, la forme de services la plus utilisée, ne s'adresse qu'à l'intégration d'applications. Dans le cas des technologies qui permettent l'intégration d'appareils et d'applications, des outils sont manquants. Jini, par exemple, n'offre pas de langage de composition. Enfin, toutes ces technologies ne peuvent pas être facilement intégrées, soit parce qu'elles reposent sur un langage particulier, soit parce qu'elles proposent des architectures qui ne sont pas compatibles. Par conséquent, l'hétérogénéité des services reste un frein à l'intégration de services hétérogènes tant sur le plan technologique que sur le plan de leur implémentation.

La prise en compte des propriétés non-fonctionnelles, comme la sécurité ou les transactions, est un autre frein à l'utilisation de l'approche à services. En effet, toutes les technologies ne permettent pas de prendre en compte des propriétés non-fonctionnelles et certaines ne permettent d'en prendre en compte qu'un nombre limité.

Ces deux freins s'appliquent tout particulièrement dans le domaine pervasif, ou les services sont susceptibles d'être très nombreux et hétérogènes. Par conséquent, afin de développer des applications pervasives, il est nécessaire d'intégrer de manière transparente une grande variété de technologies à services. De plus, il faut pouvoir prendre en compte des propriétés non fonctionnelles, et notamment la sécurité, et la protection de la vie privée qui conditionnent l'acceptation des applications par leurs utilisateurs. C'est à la prise en compte de cette propriété que nous nous intéressons désormais en étudiant les failles de sécurité auxquelles sont exposées les applications orientées services et les solutions qui y ont déjà été apportées.

II Sécurité : concepts et buts

La sécurité d'un système informatique consiste à se défendre contre un adversaire qui cherche à utiliser les ressources et les données du système de manière illégitime. Les utilisations illégitimes peuvent conduire soit à l'observation ou à la modification des données, soit au déni de services en empêchant l'utilisation du système et de ses ressources [Jajodia 2007]. La sécurité informatique doit ainsi permettre de protéger trois propriétés :

- **La confidentialité des données**, qui garantit que seules les parties autorisées connaissent les données.
- **L'intégrité des données**, qui garantit que les données ne sont modifiées que par des utilisateurs ou agents logiciels autorisés.
- **La disponibilité**, qui garantit l'accès au système et à ces ressources.

La protection de ces trois propriétés est l'objet d'une politique de sécurité. Le choix de cette dernière est dépendant de l'application considérée et doit donc être pris en compte dès la conception de l'application afin de repérer les menaces potentielles qui s'appliquent à l'application et d'identifier les mécanismes de sécurité à mettre en œuvre.

De nombreuses méthodes permettent d'identifier les phases de conception des applications ainsi que celles de la conception de la politique de sécurité ainsi que les liens entre la conception de l'application et de la politique de sécurité [42]. On peut repérer quatre étapes principales dans la conception d'une politique de sécurité.

Le développement d'une application sécurisée commence par l'analyse des besoins auxquels l'application répond et sa conception. Cette première phase permet de repérer les menaces de sécurité auxquelles l'application est exposée. Elle permet aussi de maîtriser tous les risques connus. Une fois les menaces repérées, la politique de sécurité peut-être conçue. Les menaces doivent être modélisées afin de trouver les réponses appropriées.

Une fois conçue, l'application peut être implémentée, tout comme la politique de sécurité. Cette phase consiste à choisir les outils et le langage de sécurité utilisé pour implémenter la sécurité.

Après l'implémentation, la sécurité et l'application peuvent être testées. À cette étape, il est possible de simuler le comportement de la politique de sécurité et de s'assurer qu'elle respecte certaines bonnes propriétés.

Si les tests sont validés, l'application et sa politique de sécurité peuvent être déployées et mises à disposition d'une communauté d'utilisateurs. Suite à cette phase, il est possible de procéder à un audit régulier des erreurs qui se trouvent dans l'application et la politique de sécurité afin de les corriger.

Dans cette question, nous repérons les menaces auxquelles les compositions de services sont exposées. Nous mettons l'accent sur les menaces qui mettent en danger la vie privée. Pour chacune d'elles, nous présentons les protections possibles.

42. Le *Security Development Lifecycle* (SDL) de Microsoft, par exemple, découpe le développement de la sécurité en phases qui se déroulent simultanément au développement d'une application. Voir http ://www.microsoft.com/security/sdl/default.aspx.

A Attaques et menaces dans les compositions de services pour les applications pervasives.

Les SOA sont des applications réparties. Elles reposent sur la composition de services déployés sur des plateformes hétérogènes accessibles grâce à un réseau. Dans une SOA, la communication et la réalisation des fonctionnalités des services reposent sur des couches illustrées sur la Figure 2.13.

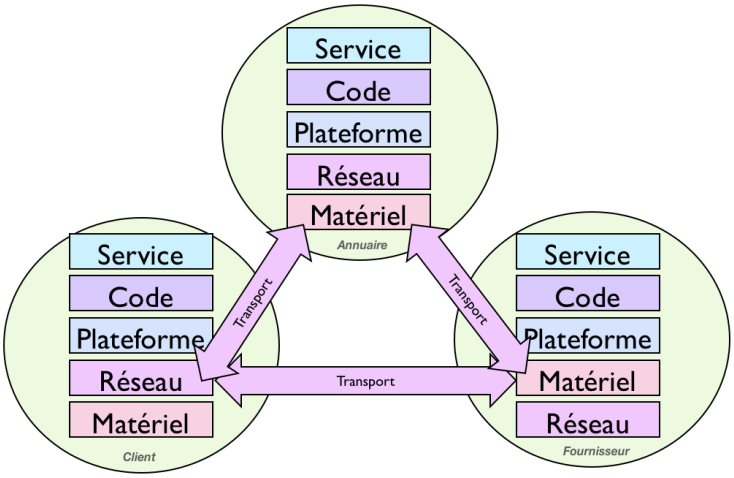

FIGURE 2.13 – Couches d'une SOA

Les failles de sécurité dans les compositions de services pour les applications pervasives découlent de deux sources. Tout d'abord, la structure répartie de l'application qui oblige le transport d'information au travers d'un réseau entre annuaire de service, client et fournisseur. Ensuite, la sensibilité de l'application au contexte.

Chaque couche est susceptible de connaître des failles de sécurité et hérite des propriétés de sécurité des couches précédentes. Au niveau matériel, les appareils peuvent ainsi être construits pour protéger la confidentialité des données [Hély 2007]. Au niveau du réseau, la confidentialité et l'ingrité peuvent être mises en danger. Des protocoles sécurisés comme HTTPS, qui combinent les protocoles HTTP et TSL/SSL permettent de transporter les données de manière sécurisée sur l'Internet [Pedersen 2011]. Il faut noter que les Web-Services, qui sont les services les plus répandus, utilisent le protocole SOAP pour communiquer. SOAP est fondé sur HTTP et non pas HTTPS, ce qui rend très vulnérable les architectures qui reposent sur les services Web.

Des mécanismes dédiés à chaque langage de programmation permettent de protéger l'accès au code. Java 7 offre ainsi un *SecurityManager* pour contrôler l'accès aux classes. Au niveau service, il est possible d'assurer la sécurité du transfert de l'information entre les services ou bien de contrôler l'accès au service. Comme le montre la Figure 2.14, notre approche se situe au niveau service. Nous cherchons à assurer la confidentialité et l'intégrité des données et des ressources afin de protéger la vie privée au niveau des services à partir d'une politique de contrôle d'accès

définie au niveau de la composition.

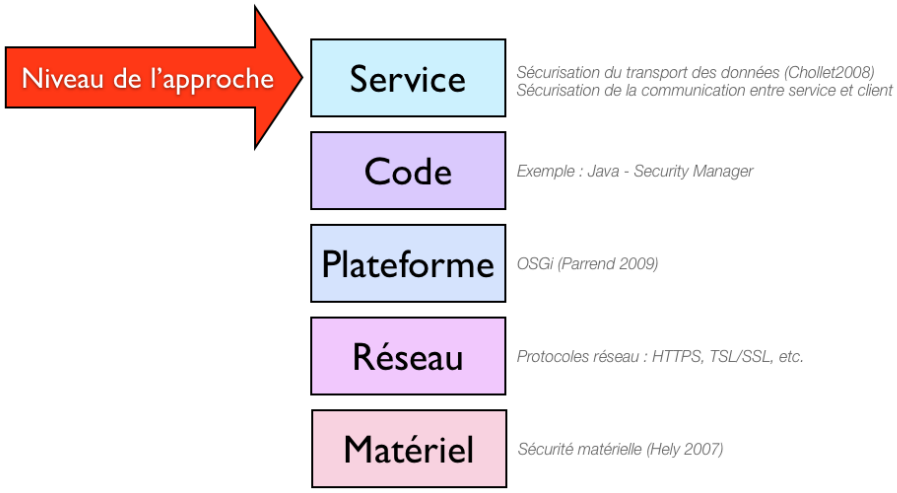

FIGURE 2.14 – Situation de notre approche dans les couches SOA

Notre objet d'étude est la protection de la vie privée. Nous postulons que la politique de vie privée est un type particulier de politique de sécurité. Nous définissons une immixtion dans la vie privée comme tout accès frauduleux à des données afin de les divulguer ou de les modifier et d'atteindre à la réputation ou à l'honneur de quelqu'un [43]. Nous ne nous restreignons pas aux données à caractère personnel, c'est-à-dire qui permettent d'identifier un utilisateur, puisque toutes les données peuvent être croisées [Sweeney 2002]. Nous définissons ainsi la vie privée comme un problème de confidentialité et d'intégrité des données. Afin de savoir protéger la vie privée, il faut savoir protéger l'intégrité et la confidentialité des données d'une manière générale. L'étude de la littérature sur la sécurité permet de repérer cinq failles de sécurité principales qui peuvent potentiellement permettre une immixtion dans la vie privée dans les compositions de services pour les applications pervasives que nous présentons désormais.

1 Menaces pour la vie privée dans les compositions de services pour les applications pervasives

Nous présentons ici les menaces principales auxquelles les compositions de services sont exposées et qui peuvent mettre la vie privée en danger.

1.1 Écoute et espionnage du réseau Dans la mesure où le client et le service qu'il souhaite atteindre communiquent au moyen d'un réseau – Internet ou Intranet – un pirate peut se mettre à l'écoute du réseau afin de récupérer les informations qui y transitent. Cette attaque remet en cause la confidentialité des données. Afin

43. Nous nous inspirons ici de la formulation juridique du droit à la vie privée qui le lie au droit à l'honneur et à la réputation ainsi qu'à la protection de la confidentialité d'un ensemble de données.

de la protéger, il faut s'assurer que les données ne peuvent être comprises que par le client et l'instance de service. Chiffrer les données est un moyen d'y parvenir.

1.2 Tromperie La tromperie consiste à prétendre être un client autorisé à accéder à un service alors que ce n'est pas le cas. Cette attaque permet de s'emparer des données ou des fonctionnalités auxquelles un client légitime a accès. Cette attaque peut permettre d'accéder aux données d'une utilisateur et de les modifier. Afin de l'empêcher, il faut mettre en place des moyens d'authentification des clients qui permettent de prouver leur identité[Frénot 2012].

1.3 Détournement d'informations Le détournement d'informations est une forme d'écoute qui consiste à intercepter la requête adressée à un service par un client. Une fois la requête interceptée, l'espion peut la réécrire. Le détournement permet de divulguer des informations confidentielles et de corrompre la composition afin de causer un déni de service. Si le service visé par l'attaque est un service Web, l'espion peut envoyer du code XML corrompu au client et bloquer le parser XML nécessaire au fonctionnement du service. Dans le cas d'autres types de service, un virus peut être injecté sous la forme de code malveillant. Pour éviter cette attaque, il faut mettre en place des mécanismes qui garantissent l'intégrité des données qui permettent d'assurer qu'elles n'ont pas été modifiées de manière frauduleuses.

1.4 Cheval de Troie Un cheval de Troie est un programme malveillant qui utilise les droits dont il bénéficie sur un environnement afin d'exécuter des actions à l'insu d'un utilisateur [Desmedt 2011]. Un service qui offre un ensemble de fonctionnalités nécessaires à une composition peut être corrompu en lui faisant exécuter des fonctions cachées [Guruge 2004]. Elles peuvent permettre de corrompre la composition et de mener à un déni de service. Si le service corrompu est un service Web, il peut envoyer du code XML corrompu au client et bloquer le parser nécessaire à l'exécution du service. Dans le cas d'autres types de service, un virus peut être injecté sous la forme de code malveillant. La protection contre les chevaux de Trois nécessite de contrôler les flux d'information afin d'en assurer la confidentialité et l'intégrité.

1.5 Attaques spécifiques aux applications pervasives Dans le cadre d'une application pervasive, le contexte, constitué notamment de l'état dans lequel se trouvent un client et un service, peut être utilisé pour conditionner l'accès à un service. Il est possible de déduire des informations sensibles de la décision d'accès au service [Hengartner 2006] de deux manières. Dans le premier cas, si le client connaît les contraintes d'accès, il peut déduire de l'accès au service la valeur des propriétés du service. Dans le second cas, si le service connaît les contraintes d'accès, il peut déduire de l'accès au service la valeur des propriétés du client. Cette attaque remet en cause la confidentialité de données qui ne sont pas directement utilisées par la composition mais qui sont nécessaires pour en protéger l'accès. Afin de l'éviter,

plusieurs remèdes sont possibles. La politique de contrôle d'accès peut être tenue secrète. Cependant, en observant les accès au service, elle peut être inférée. La mise en place d'une politique de contrôle d'accès à ces informations est un autre moyen de les protéger.

Attaque	**Menace**	**Conséquences**	**Protection**
Espionnage	Divulgation d'informations	Perte de la confidentialité des données	Confidentialité des données
Cheval de Troie	Divulgation d'informations, corruption de la composition	Perte de la confidentialité et de l'intégrité des données et des ressources	Confidentialité et intégrité des données
Tromperie	Usurpation d'identité	Perte de la confidentialité des données, atteinte à la représentation des utilisateurs	Authentification des utilisateurs et confidentialité des données
Détournement d'informations	Virus	Corruption de la composition	Intégrité des informations
Spécifiques aux applications pervasives	Divulgation et inférence d'informations	Perte de la confidentialité des données	Confidentialité des données

TABLE 2.4 – Synthèse des menaces

1.6 Synthèse des menaces et des protections possibles Le Tableau 2.4 reprend la liste des attaques que nous avons présentées, des menaces qu'elles représentent, de leurs conséquences potentielles ainsi que des protections possibles contre chacune d'entre elles. Les protections que nous avons identifiées doivent être mises en œuvre dès la conception de la composition. Elles font l'objet de la section suivante.

Il faut noter que toutes ces attaques ne sont pas spécifiques aux compositions de services ou aux applications pervasives. Elles sont souvent liées à l'utilisation d'un réseau. Néanmoins, la nouveauté et la difficulté de notre travail résident dans la recherche de solutions à ces attaques pour les compositions de services.

B Solutions techniques pour la sécurité

Concept de sécurité	But de la protection	Solution
Confidentialité	Preuve de l'identité	Nom d'utilisateur, Mot de passe et Certificat
	Protection de l'information	Chiffrement
	Autorisation	Contrôle d'accès
Intégrité	Interdiction de la dégradation des informations	
		Signature, utilisation d'un filigrane

TABLE 2.5 – Solutions contre les menaces et relations avec les concepts de sécurité

Le Tableau 2.5 présente les menaces que nous avons identifiées et les associe aux concepts de sécurité que nous avons isolés. Plusieurs solutions techniques existent pour se prémunir contre les menaces que nous avons détaillées. Nous reprenons une partie de ces solutions techniques dans le Tableau reftab :concMen, qui associe l'intégrité et la confidentialité aux solutions qui permettent d'assurer chacune de ces propriétés.

La confidentialité des données nécessite de mettre en place des moyens pour contrôler l'accès aux données. La protection par nom d'utilisateur et mot de passe ou bien par certificat permet de prouver qu'un utilisateur est bien qui il est. La vérification de cette preuve s'appelle authentification. Afin de restreindre les moyens d'action de l'utilisateur dans l'application, le contrôle d'accès est un moyen efficace. Contrôle d'accès et authentification sont souvent complémentaires. Afin de réduire la lisibilité des données, elles peuvent être chiffrées. L'intégrité des données repose sur la protection de l'information soit au moyen du contrôle d'accès – on peut alors vérifier qu'une modification des données est autorisée – soit en signant l'information pour s'assurer qu'elle n'a pas été modifiée ou détruite. Les deux solutions sont là aussi complémentaires. Il est nécessaire d'observer le fonctionnement de la politique de sécurité lorsque l'application est déployée. La solution la plus simple est d'enregistrer les actions commises dans le système. Enfin, la protection contre le déni de service ou la corruption de l'application est le plus souvent dépendante de l'application considérée et de son implémentation.

1 *Solutions pour assurer la sécurité*

Nous présentons ici quatre des solutions que nous avons identifiées pour assurer la sécurité d'un système. Nous présenterons plus en détail le contrôle d'accès, qui fait l'objet de nos recherches, dans la sous-section suivante.

1.1 Chiffrement Le chiffrement permet de masquer un message. Un message en clair est donné en entrée d'un algorithme qui produit un texte chiffré incompréhensible lorsqu'on ne possède pas la clé de déchiffrement.

Il existe deux familles d'algorithmes de chiffrage :

- Le chiffrement symétrique si la même clé est utilisée pour chiffrer et déchiffrer. Les algorithmes DES [44], Triple DES [45] et AES [46] permettent le chiffrement symétrique.
- Le chiffrement asymétrique ou cryptographie à clé publique met en jeu une clé publique pour le chiffrement et une clé privée associée pour le déchiffrement. La clé privée ne peut pas être déduite à partir de la clé publique. L'algorithme RSA est le plus connu pour ce type de chiffrement.

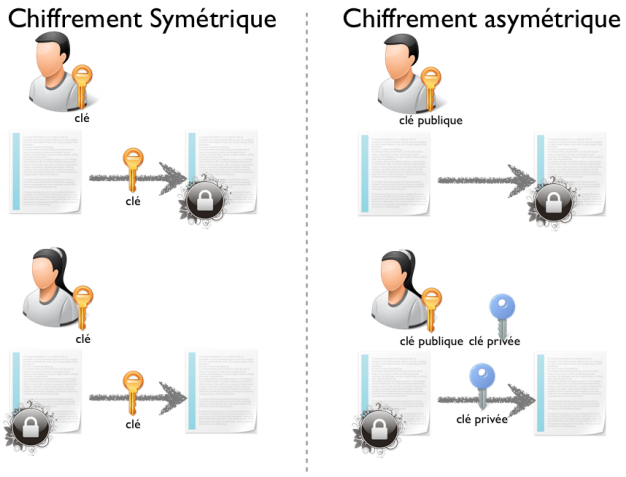

FIGURE 2.15 – Types de chiffrement

La Figure 2.15 illustre les principes du chiffrement symétrique et du chiffrement asymétrique. Le choix entre l'une et l'autre méthode repose sur leurs avantages respectifs. Le chiffrement asymétrique n'oblige pas à communiquer la clé de manière sécurisée mais reste bien plus long que le chiffrement symétrique.

1.2 Nom d'utilisateur et mot de passe L'utilisation d'un nom d'utilisateur et d'un mot de passe permet d'identifier les utilisateurs et de prouver leur identité lorsqu'ils se connectent à une application. Pour être efficace, le mot de passe ne doit être connu que par son propriétaire. Afin d'être sûr, il doit être construit selon des règles comme le mélange de plusieurs casses de caractère, l'utilisation de nombres

44. Data Encryption Standard. Cet algorithme utilise une clé de 56 bits. Il est recommandé de ne plus l'utiliser car les messages peuvent être déchiffrés par un pirate en un temps raisonnable.

45. Algorithme de chiffrement symétrique qui enchaîne trois applications successives de DES sur un même bloc de 64 bits avec 2 ou 3 clés différentes.

46. Advanced Encryption Standard. C'est le standard de chiffrement des administrations américaines. Il est appliqué à des blocs de 126 bits avec des clés de 126, 192 ou 256 bits.

et de lettres. Le mot de passe ne doit pas non plus être un mot du dictionnaire. La Figure 2.16 illustre le principe de l'authentification par nom d'utilisateur et mot de passe.

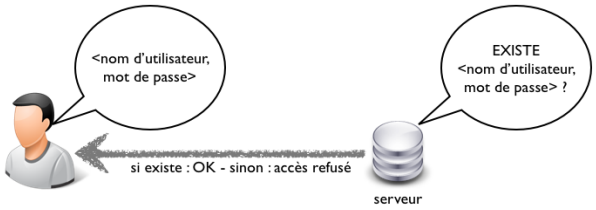

FIGURE 2.16 – Utilisation d'un nom d'utilisateur et d'un mot de passe

Les paires formées du nom de l'utilisateur et de son mot de passe sont stockées sur un serveur. Chaque paire est unique. Lorsqu'un utilisateur se connecte, il fournit une paire composé de son nom d'utilisateur et de son mot de passe. La paire est comparée à celles qui sont stockées sur le serveur. S'il existe une paire identique, alors l'utilisateur est autorisé à accéder à l'application.

La communication du nom d'utilisateur et du mot de passe au travers d'un réseau peut permettre à un pirate qui observe le réseau de les récupérer. Il est donc préférable de chiffrer le nom d'utilisateur et le mot de passe lors de leur communication. Il est aussi possible de limiter leur validité à un intervalle de temps.

L'identification grâce à des propriétés biométriques comme les empreintes digitales ou rétiniennes est analogue à l'utilisation d'un nom d'utilisateur et d'un mot de passe. Elle garantit l'identification de l'utilisateur car elle repose sur une propriété unique.

1.3 Signature numérique La signature numérique garantit l'intégrité d'un document électronique et en identifie l'auteur à l'aide de l'empreinte du document que l'auteur génère. Son fonctionnement est illustré sur la Figure 2.17. La signature numérique repose sur le chiffrement asymétrique. L'empreinte obtenue dépend du document, de la fonction de hachage utilisée et de la clé privée de l'auteur.

Lorsqu'un utilisateur souhaite signer un document, il en génère une empreinte à l'aide d'une fonction de hachage[47]. Il chiffre ensuite cette empreinte à l'aide de sa clé privée. La clé publique de l'auteur permettra de déchiffrer l'empreinte.

Lorsque l'auteur communique le document, il communique aussi l'empreinte qu'il a généré. Le destinataire du document peut ainsi à son tour générer une empreinte à partie de la même fonction de hachage que celle utilisée par l'auteur. À l'aide de la clé publique de ce dernier, il déchiffre l'empreinte qui lui a été communi-

47. Les fonctions les plus connues sont *Message Digest 5* (MD5), fonction de hachage cryptographique utilisée pour obtenir les empreintes des documents. Son utilisation n'est pas recommandée dans les architectures à clés publiques car elle permet de nombreuses attaques et *Secure Hash Algorithm* (SHA)

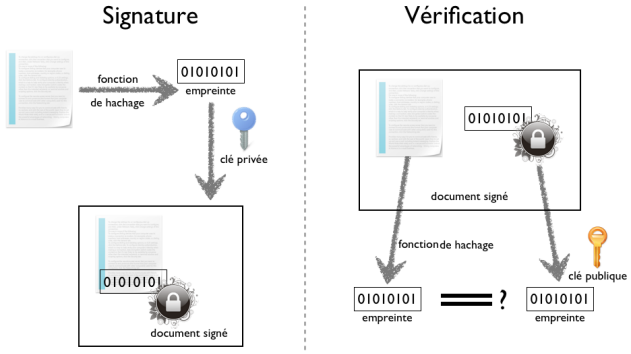

FIGURE 2.17 – Utilisation d'une signature électronique

quée et obtient une empreinte en clair. Soit les deux empreintes correspondent et le document n'a pas été modifié, soit elles sont différentes et le document a été altéré ou la clé privée utilisée pour signer n'est pas associée à la clé publique

La signature numérique est un moyen sûr de s'assurer de la modification d'un document ou de l'usurpation de l'identité de l'auteur lorsque le couple clé publique/clé privé n'est pas valide. En effet, les clés sont difficiles à reproduire car elles dépendent de chaque document.

1.4 Certificat électronique Le certificat électronique ou certificat de clé publique identifie une personne physique ou morale. Il est l'analogue numérique d'une carte d'identité et est signé par un tiers de confiance pour prouver sa validité. X.509 est le standard de certificat le plus répandu [48]. Un certificat X.509 contient notamment :

- Une version
- Le nom de l'autorité de certification
- Une durée de validité : une date avant laquelle il n'est pas valide et une date après laquelle il n'est plus valide
- Le nom du détenteur du certificat
- L'algorithme de clé publique
- La clé publique

Un certificat est un document public. Seul son détenteur détient la clé privée à laquelle la clé publique du certificat est associée. Le certificat prouve le lien entre son détenteur et la clé publique qu'il contient. L'autorité de certification signe le certificat pour prouver son authenticité et son intégrité. Elle détient ainsi une paire de clés privée et publique.

48. http ://www.itu.int/net/home/index- fr.aspx

C Contrôle d'accès : challenges et solutions

1 Modèles et mécanismes d'exécution du contrôle d'accès

Le contrôle d'accès est l'interception des requêtes d'accès aux ressources ou aux données d'un système et l'examen de la requête afin de déterminer si elle doit être autorisée ou interdite.[Samarati 2000]Une fois la décision prise, elle doit être mise en œuvre par un mécanisme qui implémente la politique de sécurité.

Le contrôle d'accès protège trois propriétés :

- la confidentialité des données, c'est-à-dire l'accès uniquement à des personnes autorisées
- l'intégrité des données contre des modifications qui ne sont pas autorisées
- l'accès aux données et aux ressources aux utilisateurs qui en ont légitimement le droit

La définition et l'exécution du contrôle d'accès se découpent en trois niveaux :

- **Les modèles de contrôles d'accès** fournissent un vocabulaire et une grammaire pour exprimer les règles de contrôle d'accès. La plupart permettent des vérifications formelles afin de prouver les propriétés de contrôle d'accès.
- **Les politiques de contrôles d'accès** sont des ensembles de règles écrites à partir d'un modèle donné pour un système donné.
- **L'exécution d'une politique de sécurité** est effectuée par des mécanismes dédiés.

Définir une politique de contrôle d'accès se heurte à deux difficultés. Tout d'abord, les politiques sont souvent complexes, notamment quand l'application considérée est grande, que le nombre d'utilisateurs est important ou qu'ils changent souvent. Ensuite, le nombre de modèles de contrôle d'accès est très important, il faut donc parvenir à choisir le modèle le plus adapté afin de capturer la politique considérée.

Dans cette section, nous étudions les modèles de contrôle d'accès existants afin d'en saisir la structure. Pour chaque modèle, nous détaillons les mécanismes d'exécution existants ainsi que ces limites intrinsèques et celles liées à son utilisation dans une composition de services pour une application pervasive.

Dans un premier temps, nous présentons les entités de base du contrôle d'accès communes à tous les modèles. Nous introduisons ensuite les différents modèles de contrôle d'accès par groupes selon leur approche de ces entités et de leurs relations.

1.1 *Les entités du base du contrôle d'accès*

Le contrôle d'accès repose sur quatre entités fondamentales :

- **Les objets** sont les entités passives du système. Ils constituent les ressources disponibles dans une application.
- **Les sujets** sont les entités actives du système qui demandent accès aux objets.
- Les actions sont les modes d'accès des sujets aux objets, par exemple l'écriture d'un fichier ou sa création.
- **Les contraintes** permettent de spécifier les critères qui gouvernent l'accès à un objet.

Les sujets et les objets peuvent être regroupés en catégories afin de faciliter la gestion des droits au sein de l'application.

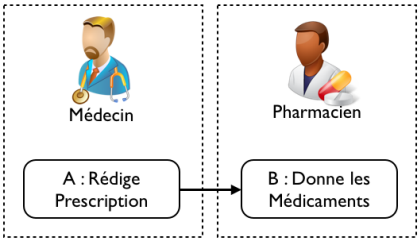

FIGURE 2.18 – Extrait d'un processus métier médical

La Figure 2.18 présente un extrait de processus métier auquel on veut associer un ensemble de contraintes de contrôle d'accès. À cet extrait on pourrait notamment associer une contrainte composée des sous-contraintes suivantes :

1. Afin d'effectuer l'action B, le sujet doit être pharmacien.
2. Un pharmacien ne peut effectuer la tâche B que pendant ses heures de service.
3. La tâche B doit être effectuée par un sujet qui n'a pas effectué la tâche A.

Ces sous-contraintes sont toute d'un type différent. La contrainte 1 est statique et porte sur la position du sujet dans l'organisation. La contrainte 2 est dynamique et mobilise des informations sur l'environnement et le sujet. La contrainte 3 est appelée séparation des privilèges. Une séparation des privilèges énonce qu'un sujet ne doit pas avoir suffisamment de privilèges pour utiliser l'application de manière malveillante. Il permet d'éviter les conflits d'intérêt. On peut envisager la séparation de privilège de manière statique (certains droits ne peuvent pas être attribués au même sujet) ou bien de manière dynamique (un même sujet ne peut pas activer ensemble un groupe de droits).

Chaque contrainte mobilise des informations différentes selon son type. Dans le cas de politiques d'accès complexes, et c'est toujours le cas des politiques de contrôle d'accès pervasif, il faut ainsi repérer les différents types de contraintes mis en jeu afin d'aider les architectes à mettre en place les sources d'information pertinentes.

La figure 2.19 présente une classification des différentes contraintes de contrôle d'accès. Nous nous inspirons de la classification proposée par Kohler [Kohler 2007], que nous étendons avec des contraintes qui portent sur toutes les entités de base du contrôle d'accès. Nous ajoutons aussi une colonne précisant les informations nécessaires à l'évaluation de chaque type de contrainte. Ce Tableau peut ainsi devenir un guide pour l'architecte d'une application protégée par le contrôle d'accès car il permet de repérer immédiatement les sources d'information qui doivent exister.

Nous séparons les contraintes en contraintes statiques des contraintes dynamiques. Ce premier découpage prend pour critère le fait que les informations sur lesquelles porte la contrainte changent. Les contraintes statiques portent sur des éléments qui n'évoluent pas dans le temps – comme les rôles d'un utilisateur – et ne

nécessitent pas de prendre en compte des informations sur les activités précédentes du sujet par exemple.

Les contraintes dynamiques portent sur des informations qui changent souvent, comme le temps ou la localisation d'un sujet. Elles peuvent aussi nécessiter de connaître les activités précédentes d'un sujet. Par exemple, une contrainte de cardinalité qui précise combien de fois un sujet peut effectuer une action nécessite d'enregistrer les activités du sujet.

Type de contraintes	Sous-Type de contraintes	Contraintes	Exemple d'informations nécessaires
Contraintes statiques	Contraintes structurelles	Rôles des sujets	Annuaire des sujets
		Rôles des objets	Annuaire des sujets
	Authentification	Identification	Certificats
Contraintes dynamiques	Contraintes sur les activités	Séparation des privilèges Liaison des privilèges	Journal des actions des utilisateurs
		Cardinalité	
		État d'achèvement	Journal des activités
	Contraintes sur l'environnement	Attributs de l'environnement Attributs des sujets Attributs des objets	Sources d'information contextuelles

FIGURE 2.19 – Classification des différentes contraintes de contrôle d'accès.

Nous divisons les contraintes statiques en deux sous-groupes. Les contraintes structurelles portent sur la catégorisation des sujets et des objets. Les catégories obtenues sont souvent appelées « rôles ». Les contraintes les plus courantes portent sur l'appartenance à une catégorie. Par exemple, pour donner des médicaments, un sujet doit appartenir à la catégorie Pharmacien. Lorsque l'on souhaite implémenter de telles contraintes, il faut maintenir une représentation des différentes catégories et des entités qui y appartiennent. Un annuaire des utilisateurs d'une application, par exemple, peut jouer ce rôle.

L'authentification n'est pas obligatoire dans tous les cas du contrôle d'accès. Par exemple, s'il faut simplement compter le nombre de personnes dans une pièce afin d'en restreindre le nombre, il n'est pas nécessaire d'enregistrer leur identité. L'authentification porte sur la possibilité de prouver qu'un sujet est bien qui il prétend être. Elle nécessite de mettre en place un système d'authentification comme un login et un mot de passe et de stocker les informations correspondantes.

Les différents modèles se distinguent de part les contraintes qu'ils permettent d'exprimer. Cette expressivité dépend des entités du modèle et de leur mode de catégorisation. Dans cette perspective, notre classification peut aussi être un moyen de choisir le modèle de contrôle d'accès pertinent en fonction des contraintes que

l'on souhaite exprimer. Nous étudions maintenant les différents modèles existants afin d'en situer les avantages et les inconvénients, notamment dans la perspective du contrôle d'accès pervasif.

2 *Principes de base du contrôle d'accès*

Les entités de base du contrôle d'accès sont habituellement utilisées selon un ensemble de principes et notamment le principe du Moindre privilège (*Least priviledge*). Ce principe énonce qu'un sujet ne doit posséder que les privilèges minimaux requis pour effectuer les actions qu'il doit faire. Ce principe évite les erreurs dues à des inadvertances. Il permet aussi de limiter le danger que représentent des utilisateurs malveillants.

Nous présentons ici l'utilisation de ces entités dans le cadre des principaux modèles de contrôle d'accès.

2.1 ***Le contrôle d'accès discrétionnaire*** Le contrôle d'accès discrétionnaire (DAC pour Discretionary *Access Control*) repose sur l'identité de l'utilisateur et sur les règles qui explicitent ce que les utilisateurs peuvent ou ne peuvent pas faire. Il est dit « discrétionnaire »car un utilisateur qui possède une permission peut la déléguer à n'importe quel autre. Le contrôle d'accès nécessite un mécanisme d'authentification fort afin d'être sûr.

	Fichier 1	**Fichier 2**	**Programme 1**
Bob	own read write	read	
Alice	read write		execute
Eve		own read write	execute write

Table 2.6 – Exemple de matrice de contrôle d'accès

DAC est historiquement lié au développement des matrices de contrôle d'accès formalisées par Harison, Ruzzo et Ullman [Harrison 1976] (nous les notons HRU pour *Harison, Ruzzo and Ullman model*). Un exemple de matrice est présenté dans le Tableau 2.6.

Une matrice de contrôle d'accès offre une représentation abstraite de la protection d'un système. Elle repose sur l'identification des objets à protéger et des sujets qui exécutent des actions sur ces objets. Lles objets peuvent être des fichiers ou des programmes. Les sujets sont les utilisateurs du système. Chaque cellule de la matrice précise les actions qu'un sujet peut effectuer sur un objet. Un état du système

protégé est défini par les privilèges d'un sujet donné. Il est représenté par le triplet (S, O, M) où S est l'ensemble des sujets, O l'ensemble des objets, M la matrice de contrôle d'accès. Dans M, les lignes correspondent aux sujets et les colonnes aux objets. M[s,o] désigne la cellule de coordonnées [s,o] et donc les droits du sujet s sur l'objet o. Les sujets peuvent être considérés comme des objets. Dans ce cas, S est inclus dans O. Par exemple, Alice peut lire et écrire dans le Fichier 1 et exécuter le Programme 1.

Harison *et al.* identifient six primitives qui permettent d'ajouter ou supprimer un objet ou un sujet de la matrice et de donner ou de retirer un privilège aux sujets. Ces primitives peuvent être composées sous la forme de commandes qui permettent d'administrer le contrôle d'accès. Chaque commande est composée d'une condition et d'un corps et est de la forme

---

Commande c(x1,...,xn)
si r1 in A[xs1, xo1] and ...rn in A[xsm, xom] **alors**
op1 ... opn

Ici, n>0 et m>0. r1...rn sont des actions et op1...opn des primitives définies par HRU. Si m=0, alors la commande n'a pas de partie conditionnelle. Par exemple, la commande suivante crée un programme et donne au sujet qui le crée le privilège *own* sur le programme. Elle autorise aussi l'utilisateur à exécuter le programme.

Commande CREATE(createur, programme)
create object programme
enter Own into M[createur, programme] and enter Execute into M[createur, programme]
End

Les commandes ne sont pas paramétrées. Il faut donc créer une commande à chaque fois que l'on souhaite permettre ou révoquer le droit de faire une action. La commande suivante permet de donner le droit d'effectuer une action aux propriétaires d'un objet et doit être écrite pour toutes les actions.

Commande GRANT(proprietaire, objet)
si Own in M[proprietaire, objet] **alors**
enter action into M[proprietaire, objet]

Les exceptions peuvent être exprimées sous la forme d'autorisations ou d'interdictions. Elles sont habituellement mutuellement exclusives, d'où deux approches du contrôle d'accès :

- Hypothèse du monde fermé : par défaut, les utilisateurs n'ont pas le droit d'accéder aux objets. La politique de sécurité ne permet l'accès à un objet pour un sujet que s'il existe une autorisation positive qui le permet.
- Hypothèse du monde ouvert : par défaut, les utilisateurs peuvent accéder aux

objets. La politique de sécurité refuse l'accès à un objet uniquement s'il existe une autorisation négative qui en refuse l'accès.

L'hypothèse du monde ouvert ne peut être utilisée lorsqu'un très haut niveau de sécurité est nécessaire. Souvent les autorisations positives et négatives sont utilisées conjointement, d'où deux problèmes :

- Incomplétude de la politique de contrôle d'accès : que faire si pour un accès aucune autorisation n'est spécifiée ?
- Inconsistance de la politique de contrôle d'accès : que faire si une permission négative et une permission positive sont octroyées à un même objet ?

L'incomplétude peut être traitée en faisant prévaloir l'hypothèse du monde fermé ou bien celle du monde ouvert. Résoudre l'inconsistance de la politique de contrôle d'accès nécessite de mettre en œuvre des mécanismes de résolution de conflits. Il n'existe pas de mécanisme unique pour parvenir à gérer ces conflits [Jajodia 2004].

HRU ne définit pas de politique d'administration du contrôle d'accès mais la formulation des commandes permet de tenir compte de cette administration. Par exemple, un marqueur peut être défini pour exprimer le fait qu'un privilège peut être transféré à d'autres sujets par son possesseur. On peut aussi définir des marqueurs pour préciser que lorsqu'un sujet transfère ses privilèges, il les perd.

Enfin, d'autres extensions des matrices de contrôle d'accès portent sur l'ajout de contraintes sur les droits d'accès. Les sujets, les objets et les actions peuvent être classés de manière hiérarchique. Ceci permet de définir des préconditions à l'accès à un objet [Shen 1992]. Tel est le cas, par exemple, dans UNIX. Pour accéder aux fichiers dans un répertoire, un utilisateur doit posséder l'accès *execute* sur ce répertoire.

La flexibilité des matrices de contrôle d'accès nécessite de vérifier une nouvelle propriété, nommée *safety* qui s'applique à la propagation des privilèges dans le système [Boudol 2009]. Le problème est de savoir si un sujet s donné peut acquérir un privilège sur un objet o. Ce problème permet d'éviter la fuite des privilèges au sein du système. D'une manière générale, le problème de *safety* est indécidable. Néanmoins, Kleiner et Newcomb [Kleiner 2007] montrent que le problème de *safety* est décidable lorsque l'on choisit des fragments de logique appropriés. Sandhu [Sandhu 1992] propose une extension des matrices dans laquelle les objets et les sujets reçoivent un type qui ne peut jamais changer. Sous certaines conditions, le problème de *safety* devient alors décidable.

Même dans les cas où la propriété de *safety* est vérifiée, le contrôle d'accès discrétionnaire reste cependant vulnérable à des attaques comme celles qui utilisent un cheval de Troie. Le principe de cette attaque est illustré sur la Figure 2.20. Tom, un cadre, crée un fichier *secret* qui contient des information sensibles. Mallory, un subordonné de Tom, souhaite accéder à *secret*. Pour ce faire, il crée un fichier *espion* et autorise Tom à écrire dedans sans que Tom ne connaisse ni *espion* ni les droits qu'il possède sur ce fichier. Enfin, Mallory modifie l'application que Tom utilise pour accéder à ses fichiers. Quand Tom lit un fichier, l'application le recopie dans *espion*.

Le contrôle d'accès discrétionnaire ne peut pas empêcher le cheval de Troie d'agir pour deux raisons :

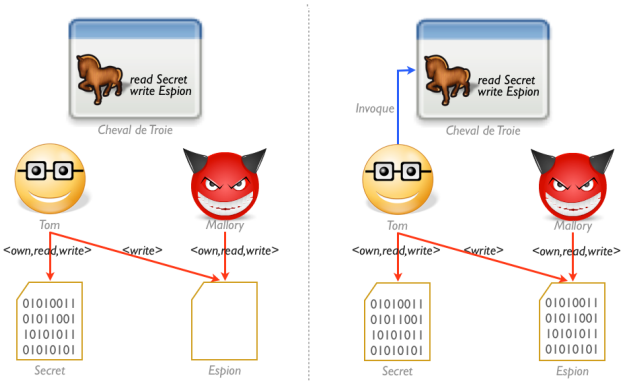

FIGURE 2.20 – Vulnérabilité du cheval de Troie au contrôle d'accès discrétionnaire

- Il s'applique aux utilisateurs et ne fait pas de différence entre les utilisateurs et les processus qu'ils exécutent, qui possèdent tous les privilèges de l'utilisateur.
- Le flux de l'information n'est pas contrôlé. Dans notre cas, il est par exemple évident que le fichier *Secret* contient des informations sensibles et que Mallory, qui n'est qu'un subordonné, ne devrait pas posséder les accréditations nécessaires pour les lire.

Comme le montre le Tableau 2.6, une matrice est pour la plupart vide, elle ne peut donc pas être stockée sous la forme d'un Tableau à deux dimensions afin de ne pas gâcher de l'espace mémoire. On peut noter trois formes principales d'implémentation des matrices de contrôle d'accès :

- La matrice peut être stockée sous la forme d'une table d'autorisation. Les cellules qui ne sont pas vides de la matrice sont stockées sous la forme de tuples de la forme <Objet, Sujet, Action>. Chaque tuple correspond ainsi à une autorisation. Les bases de données hippocratiques [Agrawal 2002], utilisent une table d'autorisation afin de stocker les privilèges des utilisateurs sur les données.
- La matrice peut aussi être stockée par colonne. On obtient alors une liste de contrôle d'accès (notée ACL pour *Access Control Lis*). Chaque objet est associé à une liste d'utilisateurs et d'actions qu'ils peuvent effectuer. Les ACL sont utilisées dans UNIX. Elles permettent de voir facilement qui a accès à un objet. Cependant, retrouver les privilèges d'un utilisateur est complexe. Enfin les ACL ne sont pas nécessairement adaptées au contrôle d'accès discrétionnaire. Dans UNIX, par exemple, chaque objet a un propriétaire qui peut seul administrer les droits sur l'objet. Par conséquent, les utilisateurs ne peuvent pas administrer l'ACL à leur discrétion, les privilèges d'administration du contrôle d'accès sont restreints.
- La matrice peut être stockée par lignes. Chaque utilisateur est associé à une liste qui indique pour chaque objet ses droits d'accès. Il est facile de retrouver les privilèges d'un utilisateur mais coûteux de retrouver tous les accès permis à un objet. Cette forme de stockage est particulièrement adaptée aux systèmes

répartis. Il suffit que l'utilisateur s'authentifie à un point d'accès du système pour que lui soit donnée la liste de ses capacités. Lorsqu'il demande l'accès à un objet, le point d'accès à l'objet peut ensuite vérifier que la liste des capacités provient d'une autorité de confiance, que les privilèges de l'utilisateur sont valides et que la liste des capacités lui appartient effectivement. Néanmoins, les listes de capacités peuvent être contrefaites, il faut donc en protéger l'authenticité.

2.2 ***Le contrôle d'accès obligatoire*** Le contrôle d'accès obligatoire (MAC pour *Mandatory Access Control*) repose sur une politique définie par une autorité centrale qui édicte les règles de contrôle d'accès [Samarati 2000]. Les utilisateurs ne peuvent pas éditer eux-même les règles, c'est pour cela qu'il est obligatoire.

MAC repose sur la différenciation des utilisateurs et des sujets. Les utilisateurs sont les personnes qui se connectent au système. Les sujets sont les processus informatiques qui agissent au nom des utilisateurs dans le système. Cette distinction permet de contrôler les fuites de données dues à l'exécution des processus.

L'exemple le plus courant d'utilisation de MAC est la mise en place de niveau d'accès à l'information à partir de la classification des sujets et des objets. Une classe d'accès est donnée à chaque objet et à chaque sujet. Les classes d'accès sont ordonnées partiellement par une relation de dominance qui est réflexive, transitive et, antisymétrique. Une classe d'accès c1 domine une classe d'accès c2, ce que l'on note c1 >= c2 si et seulement si le niveau d'accès de c1 est plus grand ou égal à celui de c2 et que les catégories de sujet de c1 contiennent celles de c2. La relation de dominance possède un plus petit majorant et un plus grand minorant.

Le plus souvent, une classe d'accès comprend deux composants :

- Un niveau d'accès, un élément d'un ensemble ordonné. Par exemple, on peut définir les niveaux d'accès *Top Secret (TS)* , *Secret (S)* et *Confidentiel (C)* tel que $TS>S>C$.
- Une catégorie de sujet, un élément d'un ensemble non ordonné qui représente, par exemple une compétence ou une fonction. Par exemple, *Civil* et *Armée* peuvent être des catégories de sujets.

Les classes de contrôle d'accès ordonnées selon la relation de dominance forment ainsi un treillis [Denning 1976]. La Figure 2.21 montre le treillis obtenu pour les niveaux de sécurité T et S, avec T>S et l'ensemble de catégories {Armée, Civil}.

Le treillis peut avoir deux interprétations, selon que la politique de contrôle d'accès est définie pour protéger la confidentialité des données ou bien leur intégrité.

Une politique de contrôle d'accès obligatoire qui protège la confidentialité des données s'applique au flux de données directs et indirects afin d'éviter la fuite de données. Les niveaux de sécurité associés aux données expriment le degré de confidentialité des données. À chaque sujet est associée une autorisation qui exprime le degré d'accès à l'information auquel un sujet à droit. Les catégories définissent les fonctions des utilisateurs. Le contrôle d'accès obligatoire est ainsi un bon moyen de mettre en œuvre le principe du privilège minimal. Chaque catégorie de sujet peut se

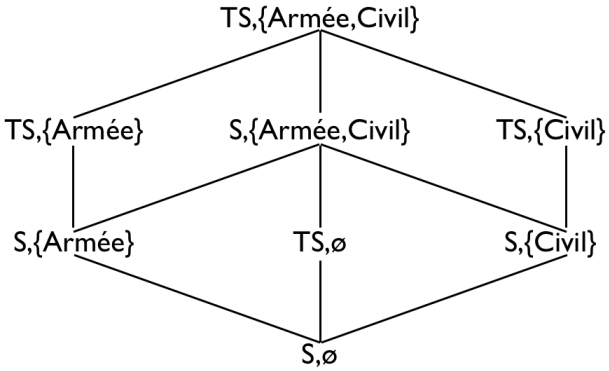

FIGURE 2.21 – Exemples de niveaux de sécurité

voir doter du niveau d'autorisation nécessaire et suffisant pour accéder aux informations nécessaires pour réaliser sa tâche. Le modèle de Bell-Lapadula est un exemple de contrôle d'accès obligatoire visant à protéger la confidentialité des données. Il connaît plusieurs versions dans [Bell 1973], [Bell 1973],[Bell 1973] et [Bell 1973].

Biba [Biba 1977] propose de reprendre l'attribution de niveaux d'accès aux utilisateurs et aux objets afin de protéger l'intégrité de ces derniers. Le contrôle d'accès est alors mis en œuvre selon les principes de *no-read-down*(un sujet ne peut lire un objet que si la classe d'accès de l'objet domine celle du sujet) et *no-write-up*(un sujet ne peut écrire dans un objet que si la classe d'accès du sujet domine celle du sujet). Ces principes interdisent de transférer de l'information contenue dans des objets dont le niveau d'accès est bas dans des objets dont le niveau d'accès est élevé.

MAC est implémenté dans les systèmes d'exploitation comme FreeBSD [49] ou bien dans le système d'exploitation *Windows*, à partir de la version *Vista* afin de protéger l'intégrité des données. Plusieurs travaux ont tenté d'implémenter MAC dans les bases de données [Samarati 2000, Jajodia 1991]. Néanmoins, il est compliqué d'attribuer un niveau d'accès à un niveau très fin comme celui d'une champ. De plus, une même donnée peut être instanciée plusieurs fois avec des niveaux d'accès différents ce qui complique l'administration du contrôle d'accès. MAC est ainsi peu appliqué dans les bases de données [Siponen 2002].

MAC ne permet pas de protéger parfaitement la confidentialité des données. En effet, MAC ne s'applique qu'aux canaux de partage de l'information prévus à la conception du système. MAC ne prend pas en compte les canaux de l'information dissimulés. Par exemple, si un processus qui a un niveau d'accès bas peut voir les résultats de l'exécution des processus ayant un haut niveau d'accès, il existe un canal dissimulé entre eux. Un tel canal peut par exemple se créer lorsqu'un processus avec un niveau d'accès bas demande l'accès à une ressource – comme le CPU ou la mémoire – qui est déjà utilisée par un processus ayant un haut niveau d'accès. En ne donnant pas l'accès à la ressource au processus ayant un niveau d'accès bas, le

49. http ://www.freebsd.org/

système peut fournir de l'information au processus ayant un niveau d'accès bas.

À la conception du système, il faut ainsi gérer soigneusement le blocage des ressources et les accès concurrents [Atluri 1999]. Néanmoins, la découverte des canaux d'information dissimulés est souvent effectuée une fois que le système est implémenté en observant les flux d'informations. Cependant, cette solution n'est pas satisfaisante car elle ne fait apparaître les failles de sécurité qu'à la fin du cycle de développement. Certains modèles tentent d'éviter la formation de canaux d'information dissimulés en imposant des contraintes sur les entrées et les sorties du système [Goguen 1984]. La plupart du temps, ces modèles permettent de mettre en œuvre le principe de non-interférence : les entrées avec un haut niveau de sécurité ne doivent pas interférer avec les sorties ayant un niveau de sécurité bas.

Enfin, MAC a connu récemment de nombreuses extensions. Sirer *et al.* proposent de mettre en œuvre MAC pour les Web Services [Sirer 2002]. Les auteurs proposent une architecture dédiée à la construction d'un site web à partir de Web Services qui applique MAC.

[Ray 2006] et Jafarian et al. [Jafarian 2008] proposent d'adapter MAC au contrôle d'accès contextuel. Pour ce faire, les niveaux d'accès attribués aux objets et aux sujets sont sensibles au contexte. Dans le cas de [Ray 2006], les niveaux d'accès ne sont sensibles qu'aux changements de localisations des utilisateurs. [Jafarian 2008] constitue une extension de [Ray 2006] n'importe quelle contrainte Les auteurs montrent que cette approche est, par exemple, adaptée aux cas où le niveau d'accès nécessaire pour accéder à une information décroît avec le temps. Par extension, cette approche est aussi adaptée au cadre pervasif.

Enfin, DAC et MAC peuvent être associés afin de protéger DAC des chevaux de Troie [Mao 2009]. Dans ce cas, pour être autorisé, il faut qu'il existe une autorisation nécessaire. Il faut que la politique de contrôle d'accès obligatoire soit respectée. DAC ne peut ainsi que restreindre les accès octroyés par MAC lorsque DAC et MAC sont utilisés ensemble.

La plupart des associations de MAC et DAC portent sur l'ajout de mécanismes de contrôle du flux d'information dans DAC [Bell 1973, Denning 1976]. McCollum et al. [McCollum 1990] proposent de mettre en œuvre un contrôle d'accès obligatoire dynamique. Chaque objet est associé à une ACL qui se propage, lorsqu'un sujet y accède à tous les objets dans lesquels le sujet pourrait transférer le contenu de l'objet auquel il a accédé. Le modèle proposé supporte aussi la représentation d'exceptions.

La politique de la muraille de Chine est un autre exemple d'utilisation commune d'éléments issus de MAC et de DAC [Brewer 1989]. Cette politique cherche à éviter la circulation d'informations qui entraînerait des conflits d'intérêts. Les utilisateurs ne devraient ainsi pas posséder des informations qui concernent plusieurs organisations concurrentes. Elle met en place une séparation des privilèges dynamiques pour les utilisateurs en adaptant leurs droits d'accès à l'historique de leurs actions.

3 *Le contrôle d'accès basé sur les rôles*

Dans le cadre de DAC et de MAC, la gestion des politiques de contrôle d'accès, surtout quand elles sont de grande taille, est complexe. Le contrôle d'accès par rôle (RBAC pour *Role-Based Access Control*) vise à simplifier cette gestion. Les sujets et les objets du contrôle d'accès changent souvent. Il faut ainsi ajouter un ensemble de nouvelles règles de contrôle d'accès pour tout nouvel utilisateur ou modifier les règles existantes à l'arrivée d'une nouvelle ressource. La Figure 2.22 présente l'essence du modèle RBAC tel qu'il est standardisé.

Le modèle RBAC repose sur les entités suivantes :

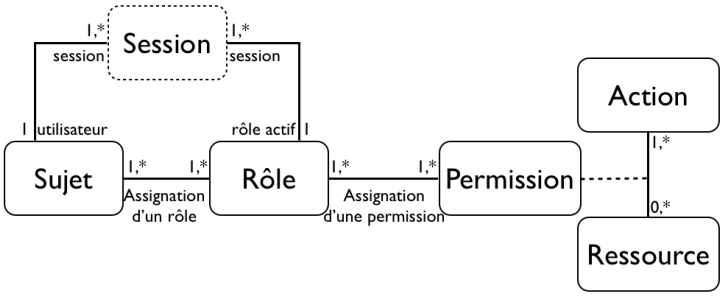

FIGURE 2.22 – Modèle RBAC

- **Les ressources** correspondent aux objets que nous avons déjà évoqués.
- **Les permissions** sont la modalité de la relation entre ressource et action. Elles représentent des droits.
- **Les rôles** sont à la fois des ensembles de sujets et des ensembles de droits. Ils regroupent tous les sujets qui ont les mêmes droits. Les droits sont directement attribués aux rôles et non plus aux sujets. Le plus souvent, les rôles correspondent à des catégories organisationnelles comme des fonctions au sein d'une entreprise. Néanmoins, il n'existe pas de règle de constitution des rôles. Il permettent de gérer plus facilement les droits car ils sont plus stables que les sujets. Un rôle diffère d'un groupe dans la mesure où il peut être activé ou désactivé alors que l'appartenance à un groupe est permanente.
- L'entité **session** figure en pointillé car elle est optionnelle. Il est important de noter qu'un sujet peut posséder plusieurs rôles et qu'un rôle peut être attribué à plusieurs sujets. Les rôles sont assignés aux sujets. Lorsqu'un sujet veut accéder au système, il doit activer l'un des rôles qu'il possède, il ouvre alors une session et ne possède à un moment donné que les droits qui correspondent aux rôles qu'il a activés. Un même rôle peut être activé par plusieurs utilisateurs à un moment donné.

Le modèle RBAC permet de scinder la gestion des autorisations en deux étapes. Tout d'abord, les rôles sont attribués aux utilisateurs. Ensuite, les permissions sont accordées aux rôles. Lorsqu'un nouvel utilisateur est ajouté à l'application, il suffit de lui attribuer les rôles nécessaires. Lorsque de nouvelles ressources apparaissent,

les autorisations ne doivent être données qu'aux rôles. RBAC peut être configuré afin d'implémenter MAC et DAC [Phillips 2003, Sandhu 1998].

RBAC permet aussi d'organiser les rôles selon des hiérarchies. Une hiérarchie est une relation d'ordre partiel gouvernée par les principes de généralisation et de spécialisation. La hiérarchie est transitive, anti-symétrique et réflexive. La hiérarchisation des rôles présente l'intérêt de s'adapter naturellement aux hiérarchies existantes des entreprises. Néanmoins, le sens des hiérarchies est laissé à la discrétion du concepteur de la politique de sécurité. Elle prend souvent la forme d'un arbre ou d'un arbre inversé. La hiérarchisation peut permettre de propager les permissions d'un rôle père à tous ses rôles fils ou bien d'automatiser l'activation de tous les rôles pères d'un rôle lorsqu'il est activé. Néanmoins, la propagation des permissions peut contrevenir au principe du moindre privilège en conduisant à l'activation d'un très grand nombre de permissions. Enfin, RBAC permet aussi l'application de contraintes. Elles peuvent porter sur l'assignation des rôles aux sujets ou bien l'activation des rôles par les sujets.

Dans le cadre de RBAC, la séparation des privilèges est appliquée au niveau des rôles et non pas des sujets. Lorsqu'elle est statique, la séparation des privilèges porte sur l'assignation des rôles aux utilisateurs : un utilisateur ne peut pas activer tous les rôles d'un ensemble. Lorsqu'elle est dynamique, la séparation des privilèges porte sur l'activation des rôles : un utilisateur ne peut pas activer tous les rôles qu'il possède.

Lorsque des hiérarchies de rôles existent, les séparations de privilèges sont héritées. Dans le cas de la séparation des privilèges statiques, la séparation de privilèges ne peut pas s'appliquer à des rôles liés par une relation hiérarchique car une instance d'un rôle fils est une instance d'un rôle père. Dans le cas de la séparation des privilèges dynamiques, la contrainte peut s'appliquer à des rôles liés par une hiérarchie.

Néanmoins, RBAC laisse ouverts de nombreux problèmes d'administration. Tout d'abord, RBAC ne fournit pas de guide pour l'ingénierie des rôles alors que cette dernière est cruciale pour une gestion efficace des permissions, surtout lorsqu'il existe de nombreux rôles. Ensuite, l'authentification des utilisateurs ne fait pas partie des problèmes traités par RBAC. Cette question est cruciale pour s'assurer qu'un utilisateur se voit pas octroyer des privilèges auxquels il n'a pas le droit. Le modèle ne précise pas le nombre de rôle maximal qu'un utilisateur peut se voir assigner. Le modèle ne précise pas non plus la nature des permissions. Il ne précise pas s'il existe des permissions négatives ni la granularité des permissions. Il ne précise pas de mécanismes pour révoquer l'attribution des rôles et plus généralement de modèles d'administration. Il ne précise pas non plus de mécanismes que les utilisateurs doivent utiliser pour activer leurs rôles ou déléguer les permissions. Ces questions ont ainsi des réponses dépendantes des différentes implémentations de RBAC.

RBAC est aujourd'hui largement implémenté. RBAC est utilisé dans les systèmes d'exploitation Windows Server 2003 et Solaris [50]. Le cadriciel .Net Framework

50. Voir http://msdn.microsoft.com/en-us/library/ms952386.aspx pour Windows Server 2003 et http://docs.oracle.com/cd/E19963-01/html/821-1456/rbac-1.html

4 intègre RBAC pour la programmation orientée objet. RBAC est aussi adaptable au modèle CORBA. Oracle DBMS 7, par exemple, permet de mettre en œuvre RBAC dans des bases de données.

Enfin, RBAC peut être utilisé dans le monde des services grâce à l'eXtensible Access Control Markup Language (XACML) de l'Advancing Open Standard for Information Society (OASIS) [OASIS 2010, OASIS 2003].

3.1 Extensions de RBAC Comme de nombreuses autres applications, les applications pervasives manipulent des données sensibles. Les maisons intelligentes, par exemple, enregistrent des informations sur l'intimité de leurs habitants. Dans la mesure où elles reposent sur un environnement réparti qui tire partie d'un réseau informatique, les applications pervasives sont connectées en permanence à d'autres applications et peuvent être utilisées par des acteurs qui sont eux aussi répartis. Si on reprend l'exemple d'une maison intelligente adaptée au maintien à domicile des personnes âgées, l'application peut, par exemple, mettre en jeu un centre d'appel ou un médecin référent. Selon l'activité qu'ils réalisent, les habitants peuvent souhaiter être plus ou moins visibles à ces acteurs. Cependant, si les habitants sont en danger, le système doit pouvoir générer et communiquer une alerte aux acteurs pertinents.

Par conséquent, les applications pervasives sont sensibles au contexte. Toute donnée peut être utilisée à des fins de filtrage. RBAC n'est pas parfaitement adapté à ce domaine d'application. Au niveau modèle, les rôles de RBAC sont définis pour les sujets uniquement. Il n'est ainsi pas explicitement prévu de prendre en compte des rôles dépendants du contexte ou bien des rôles d'objet. Au niveau exécution, il faut pouvoir collecter et traiter les données contextuelles pertinentes.

Covington *et al.* s'attaquent à ces deux problèmes [Covington 2001]. Au niveau modèle, les auteurs proposent de généraliser la notion de rôle afin de pouvoir prendre en compte les rôles d'objets et les rôles contextuels, qu'ils nomment « rôles environnementaux ». Au niveau exécution, les auteurs proposent un *middleware* chargé de collecter les données contextuelles et de transformer les données brutes en informations de haut niveau utilisables pour mettre en œuvre le contrôle d'accès.

Les permissions sont attribuées à des groupes de rôles composés de rôles de sujets et de rôles environnementaux. Par exemple, afin de préciser qu'un réparateur ne peut accéder à la maison que le lundi entre 12 :00 et 14 :00, il suffit d'attribuer la permission d'entrer dans la maison au groupe formé par les rôles *réparateur* et *lundi midi*.

La plupart des extensions de RBAC portent sur l'ajout de contraintes ou la modification du point de vue à partir duquel le rôle est créé. Thion *et al.,* Kulkarni *et al.* et Kumar reprennent ainsi les rôles contextuels afin de contraindre l'activation de permissions selon le contexte courant de l'application [Kumar 2002, Thion 2005, Kulkarni 2008]. Barker *et al.* proposent, eux, de modéliser non pas le rôle d'un utilisateur, qui est statique dans la mesure où il repose souvent sur une position hiérarchique mais le statut de l'utilisateur citeBarker2008. La notion de statut étend

pour Solaris.

celle de rôle car elle reprend la définition d'une catégorie statique et l'étend avec des éléments dynamiques comme l'historique de l'utilisateur. Les permissions sont attribuées au statut et suivent ainsi l'évolution de l'utilisateur.

Le modèle OrBAC, pour *Organisation-Based Access Control* [Cuppens 2008] peut-être vu comme une extension de RBAC. OrBAC introduit notamment une entité *Contexte* pour modéliser les contraintes contextuelles qui s'appliquent à la politique de contrôle d'accès. OrBAC entend ainsi faciliter la modélisation et l'implémentation du contrôle d'accès et le passage de l'une à l'autre de ces phases.

OrBAC introduit deux niveaux de représentation d'une politique de sécurité, le niveau abstrait et le niveau concret, reliés par des transformations automatiques. OrBAC repose sur un langage logique dont l'expressivité est très importante, cependant, il n'est pas complètement déterminable. OrBAC est implémenté sous la forme d'un outil d'administration dédié ainsi que sous la forme d'un profil XACML.

3.2 *Conclusion à l'étude de RBAC* Le modèle RBAC a comme avantage d'être aujourd'hui très répandu et standardisé. De plus, de nombreux travaux montrent qu'il peut être étendu afin de prendre en compte de nouvelles contraintes comme celles issues des applications pervasives. Cependant, pour prendre en compte chacune de ces contraintes, il faut créer un nouveau rôle qui est, par définition, une entité stable et qui ne change pas. Le nombre de rôles à créer et à gérer est ainsi susceptible d'être très élevé dans une application et donc complexe à maintenir. Ainsi, RBAC simplifie la gestion des permissions dans le cas d'applications où le nombre de rôles est restreint et donc facile à gérer. L'introduction de rôles contextuels fait exploser le nombre de rôles à modéliser et complexifie donc la tâche de l'administrateur du contrôle d'accès.

4 Le contrôle d'accès basé sur les attributs

Le contrôle d'accès basé sur les attributs (ABAC pour *Attribute-Based Access Control*) est proche de MAC car il repose lui aussi sur l'identification d'un ensemble de propriétés de sécurité. Néanmoins, il repose sur une architecture de mise en œuvre du contrôle d'accès qui ne repose pas nécessairement sur une autorité centrale. De plus, les attributs sur lesquels portent le contrôle d'accès peuvent appartenir aux sujets ou à l'environnement de l'application.

4.1 *Modèle ABAC* La Figure 2.23 présente les entités principales du modèle ABAC et leurs relations. ABAC, ne possède pas un modèle qui fait consensus. Nous reprenons ici les entités que l'on trouve le plus couramment dans la littérature et leurs relations [OASIS 2010, Yuan 2005, Li 2007].

ABAC reprend les entités de base du contrôle d'accès et les étend avec trois concepts :

- **Un attribut** est une propriété sur laquelle porte le contrôle d'accès. Chaque entité se voit attribuer un ensemble d'attributs qui peuvent prendre une valeur choisie dans un ensemble de valeurs.

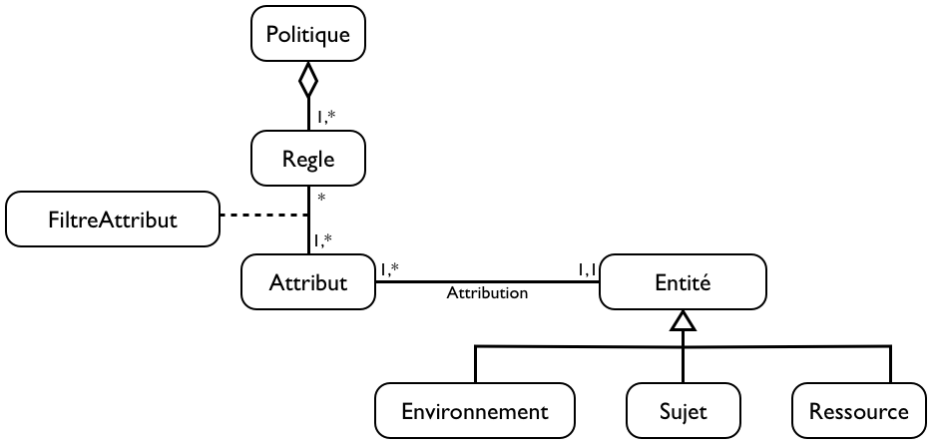

FIGURE 2.23 – Le contrôle d'accès basé sur les attributs

- **Une règle** restreint les valeurs autorisées d'un ensemble d'attributs. Elle est vérifiée si et uniquement si la valeur des attributs auxquels elle s'applique appartient à l'ensemble des valeurs autorisées de l'attribut.
- **Une politique** est composée d'un ensemble de règles.

ABAC peut facilement s'adapter à d'autres modèles de contrôle d'accès. En considérant les rôles des sujets comme des attributs, ABAC peut ainsi remplacer RBAC ou bien OrBAC. Néanmoins, face à RBAC qui est le modèle le plus implémenté aujourd'hui, ABAC présente deux avantages :

- En se concentrant sur le filtrage des attributs, ABAC ne rend pas nécessaire la définition et la gestion de rôles statiques.
- En mettant l'accent sur les attributs, ABAC permet de définir des règles avec une granularité très fine. Dans le cas de politiques d'accès complexe, ABAC passe mieux à l'échelle que RBAC, qui nécessite de créer un grand nombre de rôles. Dans RBAC, en effet, l'augmentation des rôles et des permissions en fonction du nombre d'attributs est ainsi exponentielle. [Yuan 2005].
- Même si de nombreuses extensions de RBAC l'appliquent au contrôle d'accès contextuel, RBAC n'est pas spécifiquement adapté à ce cas alors qu'ABAC permet de prendre en compte les attributs d'un environnement.

Nous présentons désormais l'architecture logique nécessaire pour exécuter ABAC puis son implémentation.

4.2 Architecture logique d'exécution de ABAC ABAC est exécuté dans une architecture qui repose sur quatre entités logiques :

- **Une autorité** doit définir et gérer les politiques de sécurité.
- **Des sources dédiées permettent de rendre tous les attributs accessibles**. Les sources peuvent stocker elles-mêmes les attributs et leurs valeurs. Les attributs peuvent aussi être stockés par d'autres entités – par exemple, un annuaire LDAP.

- **Le point de mise en œuvre du contrôle d'accès (PEP, pour *Policy Enforcement Point*)** reçoit les requêtes des sujets. Il demande l'autorisation de les effectuer. Plusieurs PEP peuvent exister afin de les répartir sur le réseau dans le cas d'une application répartie. Afin de garantir l'exécution du contrôle d'accès, le PEP doit intercepter toutes les requêtes et ne doit jamais pouvoir être contourné.
- **Le point de décision (PDP, pour *Policy Decision Point*)** évalue les requêtes en fonction de la politique de contrôle d'accès et des attributs des sujets, des ressources et de l'environnement.

Cette architecture est présentée sur la Figure 2.24.

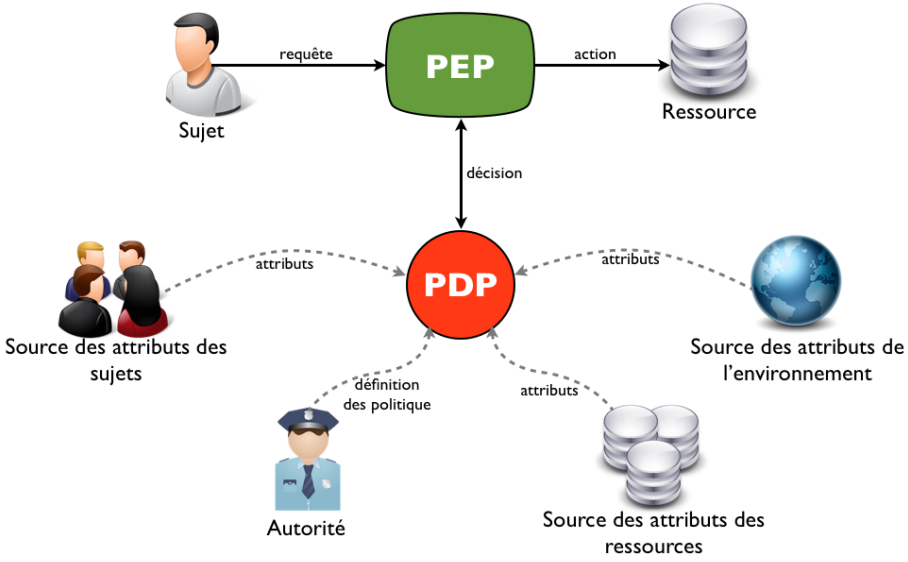

FIGURE 2.24 – Architecture d'exécution d'ABAC

4.3 Implémentation de ABAC *L"Advancing Open Standards for the Information Society* (OASIS) propose, afin d'implémenter ABAC, l'*eXtensible Markup Language* (XACML). XACML est un schéma générique d'autorisation qui reprend les entités logiques que nous avons identifiées plus haut. XACML définit aussi une grammaire XML qui permet de représenter des politiques de sécurité ainsi qu'une grammaire pour exprimer les requêtes de contrôle d'accès et les décisions rendues. XACML peut être facilement étendu au moyen de profils pour s'adapter à d'autres modèles. Il existe ainsi un profil pour RBAC et un profil pour OrBAC notamment.

Le code ci-desous donne un exemple de règles de contrôle d'accès XACML. Cette règle précise que pour tous les sujets et toutes les ressources, l'action *login* ne peut être effectuée qu'entre neuf et dix-sept heure.

```
1
2  <Rule id="LoginRule" Effect="Permit">
3    <!-- Only use this rule id the action is login -->
4    <Target>
5      <Subjects>
6        <AnySubject/>
7      </Subjects>
8      <Ressources>
9        <AnyRessources/>
10     </Ressources>
11     <Actions>
12       <ActionMatch MatchId="urn :oasis :names :tc :xacml :1.0 :
             func- tion :string-equal">
13         <AttributeValue DataType="http ://www.w3.org/2001/
               XMLSchema#string">login</AttributeValue DataType>
14         <ActionAttributeDesignator DataType="http ://www.w3.org
               /2001/XMLSchema#string"AttributeId="ServerAction"/>
15       </ActionMatch>
16     </Actions>
17   </Target>
18   < !-- Only allow logins from 9am to 5pm -->
19     <Condition FunctionId="urn :oasis :names :tc :xacml :1.0 :
         function :and">
20       <Apply>
21         <FunctionId="urn :oasis :names :tc :xacml :1.0 :function :
             time-greater-than-or-equal"/>
22         <FunctionId="urn :oasis :names :tc :xacml :1.0 :function :
             time-one-and-only">
23         <EnvironmentAttributeSelector DataType="http ://www.w3.org
             /2001/XMLSchema#time"
24         <AttributeId="urn :oasis :names :tc :xacml :1.0 :
             environment :current-time"/>
25         <AttributeValue DataType="http ://www.w3.org/2001/
             XMLSchema#time">09 :00 :00
26         </AttributeValue>
27       </Apply>
28       <Apply>
29         <FunctionId="urn :oasis :names :tc :xacml :1.0 :function :
             time-less-than-or-equal"
30         <FunctionId="urn :oasis :names :tc :xacml 1.0 :function :
             time-one-and-only">
31         <EnvironmentAttributeSelector DataType="http ://www.w3.org
             /2001/XMLSchema#time" />
32         <AttributeId="urn :oasis :names :tc :xacml :1.0 :en-
             vironment :current-time"/>
33         <AttributeValue DataType="http ://www.w3.org/2001/
             XMLSchema#time">17 :00 :00/>
34       </Apply>
35     </Condition>
36 </Rule>
```

Exemple de code 2.3 – Exemple de code XACML

L'architecture définie par XACML est implémentée sous de nombreuses formes. Dans le domaine des Services, *XACMLight*[51] offre une implémentation d'un PDP et d'une autorité de création et de gestion des politiques pour le *framework* Apache Axis 2[52]. XACMLight ne s'applique ainsi qu'aux Web Services.

4.4 Conclusions sur ABAC ABAC est particulièrement bien adapté aux applications où un nombre important de données doit être pris en compte afin de mettre en œuvre le contrôle d'accès. Il pourrait ainsi remplacer, dans ces domaines au moins, d'autres modèles comme RBAC. Cependant, dans certaines applications comme les processus métiers, la notion de rôle reste pertinente, des solutions hy-

51. http ://xacmllight.sourceforge.net/
52. http ://axis.apache.org/axis2/java/core/

brides qui associent RBAC et ABAC sont peut-être à même de mettre en commun les avantages respectifs des deux approches [Kuhn 2010]. Ainsi ABAC permet d'éviter l'explosion du nombres de rôles dans une application complexe ou pervasive. Néanmoins, la notion de rôle offre un moyen intuitif de structurer les politiques de sécurité qui permet de faciliter leur représentation, ce qui fait défaut à une politique de contrôle d'accès composée d'une liste de contraintes sur des données.

Enfin, pour être efficace, ABAC doit s'appuyer sur des mécanismes pour sécuriser la collecte et le traitement des données nécessaires aux décisions du contrôle d'accès. De tels mécanismes restent à inventer, tout comme une méthode qui permet de modéliser les données à prendre en compte malgré leur hétérogénéité.

5 Le contrôle d'accès dédié aux processus, les mécanismes de contrôle d'accès spécifiques aux compositions de service.

5.1 ***Le contrôle d'accès basé sur les tâches*** Les modèles que nous avons étudiés jusqu'ici s'appliquent principalement à des actions élémentaires. Néanmoins, d'autres points de vue sur le contrôle d'accès sont possibles. Par exemple, le contrôle d'accès basé sur les tâches (TBAC pour *Task-Based Access Control*) [Thomas 1998] déplace la perspective en introduisant la notion de tâche, c'est-à-dire d'action composite sur laquelle porte le contrôle. L'attribution des permissions se fait au fur et à mesure du déroulement des tâches. Ce type de modèle de contrôle d'accès est particulièrement adapté aux processus. Le contrôle d'accès ainsi mis en place est dynamique dans la mesure où il suit l'exécution de l'application.

5.2 ***Mécanismes de contrôle d'accès dédié aux compositions de services*** Si certains auteurs ont développé des langages de contrôle d'accès spécifiques pour les services [Sirer 2002, Srivatsa 2007], les modèles et les concepts traditionnels du contrôle d'accès peuvent s'appliquer [Chae 2012]. Le modèle ABAC connaît ainsi un grand succès dans les compositions de service [Shen 2006, Yeh 2011].

Néanmoins, les compositions de services présentent deux propriétés qui remettent en cause la mise en œuvre du contrôle d'accès. Tout d'abord les services sont hétérogènes. Il existe plusieurs types de service et chaque type de service possède de multiples implémentations. L'hétérogénéité des services nécessite de maintenir plusieurs versions du code de contrôle d'accès, chacune devant être adaptée à un service particulier. Le dynamisme des services, qui sont ajoutés à la composition au dernier moment, oblige à concevoir une architecture de contrôle d'accès flexible.

Ces deux propriétés peuvent être prises en compte en tirant partie de la sélection des services à partir de leur description. De nombreux auteurs proposent ainsi d'ajouter la description des propriétés de sécurité à la description des services eux-même [Carminati 2006]. L'exécution de la politique de sécurité est ainsi partagée en deux temps. La politique de sécurité est conçue en même temps que l'application. Lors de l'exécution de la composition de services, les services avec les propriétés de sécurité qui correspondent à la politique sont sélectionnés. Cependant, cette approche ne permet pas de s'assurer qu'il existe au moins un service capable d'exécuter

la politique de sécurité. De plus, elle oblige les fournisseurs de services à développer des services qui possèdent des propriétés non-fonctionnelles, ce qui peut nuire à leur réutilisation.

Le problème posé par l'hétérogénéité et le dynamisme des services a ainsi été résolu pour des domaines d'application spécifiques des services [Hung 2007] ou bien pour des types de service particulier. Il manque cependant une approche suffisamment générique pour prendre en compte tous les types de service.

6 *Administration du contrôle d'accès*

Nous avons jusqu'ici étudié les différents modèles de contrôle d'accès existants. L'administration d'une politique de contrôle d'accès, c'est-à-dire sa création et sa maintenance sont cependant des tâches cruciales. Dans une application centralisée, un seul utilisateur, comme un administrateur de base de données ou bien un super utilisateur dans le cas d'un système d'opération peut concevoir la politique de contrôle d'accès.

Dans une application répartie, il peut exister plusieurs personnes qui peuvent édicter des règles de contrôle d'accès. Dans le cas d'une SOA, par exemple, les fournisseurs des différents services peuvent édicter des politiques de contrôle d'accès à leurs services. Le concepteur d'une composition de services peut, lui aussi, créer une politique de contrôle d'accès.

La politique d'administration du contrôle d'accès précise ainsi qui peut créer des catégories de sujets ou d'objets, leur assigner des sujets ou des objets et octroyer ou révoquer des droits. XACML, par exemple, spécifie une politique d'administration du contrôle d'accès distincte de la politique de contrôle d'accès elle-même.

La délégation est un élément important que le contrôle d'accès droit prendre en compte. Dans le cas d'une politique de contrôle d'accès décentralisée, plusieurs modes d'administration sont possibles [Samarati 2000] :

- **Administration hiérarchique :** Un administrateur central peut permettre à d'autres administrateurs de modifier le contrôle d'accès.
- **Administration coopérative :** La politique de contrôle d'accès peut être modifiée par plusieurs administrateurs à la fois.
- **Administration basée sur la propriété** : Chaque objet possède un propriétaire. Les propriétaires peuvent accorder et révoquer les droits d'accès à leurs objets.
- **Administration décentralisée** : Les propriétaires de chaque objet peuvent gérer les droits d'accès à leurs objets, déléguer et permettre leur délégation par d'autres utilisateurs.

Dans le dernier cas, la délégation permet une plus grande souplesse du contrôle d'accès même si elle en complique la gestion et rend difficile pour un utilisateur de se représenter qui a accès à ses objets.

La politique d'administration du contrôle d'accès est responsable du maintien de l'intégrité de la politique de contrôle d'accès. Il faut par exemple trancher les cas où un utilisateur se voit retirer des privilèges qui influencent d'autres privilèges. Dans les cas où des droits ont été délégués, il faut établir ce qu'il advient des privilèges obtenus par plusieurs délégations successives lorsque la première délégation est révoquée. SQL, par exemple, permet dans sa version 2011 de révoquer les délégations de manière récursive ou en cascade. SQL permet aussi de ne pas révoquer les permissions en cascade. Ceci est particulièrement utile lorsqu'on redéfinit les rôles ou les catégories de sujets d'une politique de contrôle d'accès et que l'on ne souhaite pas que ces modifications affectent toute la politique de contrôle d'accès.

7 *Techniques de protection spécifiques à la vie privée*

Nous avons jusqu'ici traité du contrôle d'accès de manière générale. Néanmoins, la gestion de la vie privée impose des contraintes spécifiques qui peuvent être prises en charge par le contrôle d'accès – comme la mise en œuvre d'autorisation d'accès aux informations – d'autres qui lui sont orthogonales (comme la préservation de l'anonymat). Nous présentons désormais deux exemples de techniques dédiées à la protection de la vie privée, l'une liée au contrôle d'accès, l'autre qui ne l'est pas.

7.1 ***Techniques issues du contrôle d'accès*** L'*Enterprise Privacy Authorization Language* (EPAL)[53] est une proposition de recommandation du W3C qui étend ABAC pour l'adapter à la gestion de la vie privée. Depuis 2003, EPAL en est à sa version 1.3 et n'est pas utilisé. EPAL permet aux entreprises de décrire leur politique en matière de gestion des données. Le modèle de EPAL est présenté sur la Figure 2.25.

EPAL ajoute des entités comme la catégorie d'utilisateur ou le but poursuivi par ce dernier aux entités du contrôle d'accès. EPAL est un langage de spécification qui permet notamment d'exprimer des obligations imposées, par exemple, par la loi. Il est par exemple possible d'exprimer le fait qu'une donnée doit être détruite au bout de trente jours.

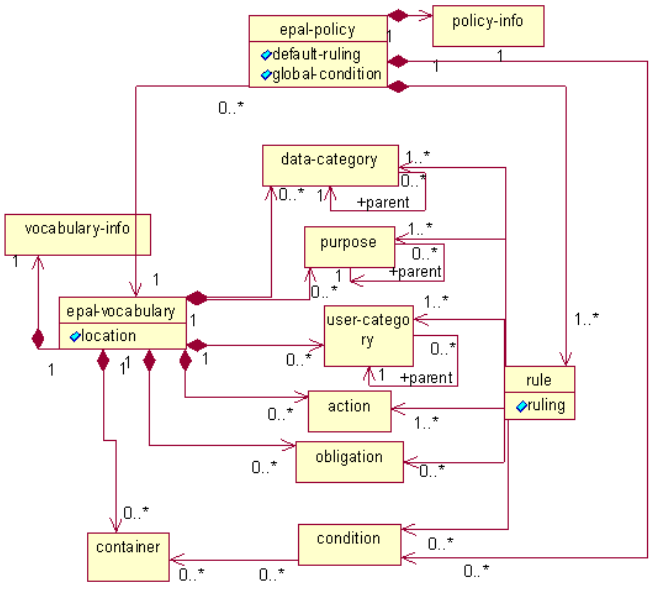

FIGURE 2.25 – Enterprise Privacy Authorization Language

EPAL offre ainsi un vocabulaire spécifique à la gestion de la vie privée. Cependant, il constitue un sous-ensemble du modèle ABAC dans la mesure où toutes les entités qui composent les règles d'EPAL peuvent être vues comme des attributs.

53. http ://www.w3.org/Submission/2003/SUBM-EPAL-20031110/

De plus, EPAL n'offre pas de fonctionnalités pour exécuter les entités qu'il ajoute, comme les obligations. Par conséquent, il est possible d'utiliser les langages fondés sur ABAC, comme XACML, pour exprimer les politiques d'EPAL.

7.2 ***Manipulation des données*** La protection de la vie privée repose non seulement sur la gestion des accès aux données mais aussi sur le contrôle de leur diffusion, qui peut viser notamment à préserver la propriété d'un ensemble de données ou bien l'anonymat des personnes décrites par ces données.

La propriété et l'intégrité des données peut être préservée en introduisant un filigrane dans les données. Ce filigrane est altéré lorsque les données sont modifiées. Son caractère unique et idéalement inimitable permet de prouver la propriété des données. Les filigranes sont notamment utilisés pour protéger les bases de données [Gross-Amblard 2011] et les images [Perwej 2012].

L'anonymat est probablement la propriété la plus intuitivement raccrochée à la protection de la vie privée. La préservation de l'anonymat est rendue complexe par la dissémination des données sur l'Internet, par exemple,t qui permet, par croisement et inférences de reconstituer l'identité des individus décrits par un jeu de données même si elles ont été anonymisées [Minami 2010]. Latanya Sweeney propose une solution à ce problème nommée « k-anonymat »[Sweeney 2002] dont le principe est illustré sur la Figure 2.26.

Sweeney montre que le code postal et la race d'une personne peuvent permettre de l'identifier. L'auteur propose dès lors de noyer un tuple dans un ensemble de taille k de tuples qui lui sont identiques. Il est dès lors impossible d'identifier le possesseur des données. La Figure 2.26 présente une base de données sans k-anonymat (PT), et deux exemples de la même base avec un facteur d'anonymat k de valeur 3 (GT1 et GT2). On observe que la mise en œuvre du k-anonymat repose ici sur la substitution des valeurs des attributs Race et ZIP afin de garantir qu'il existe au moins trois tuples identiques. La substitution peut s'appuyer sur des arbres de généralisation, la classe de valeurs Person regroupe ainsi les classes Asian, Black et White.

Race	ZIP
Asian	2138
Asian	2139
Asian	2141
Asian	2142
Black	2138
Black	2139
Black	2141
Black	2142
White	2138
White	2139
White	2141
White	2142

PT

Race	ZIP
Person	2138
Person	2139
Person	2141
Person	2142
Person	2138
Person	2139
Person	2141
Person	2142
Person	2138
Person	2139
Person	2141
Person	2142

GT1

Race	ZIP
Asian	2130
Asian	2130
Asian	2140
Asian	2140
Black	2130
Black	2130
Black	2140
Black	2140
White	2130
White	2130
White	2140
White	2140

GT2

FIGURE 2.26 – Exemple de Base de données sans (PT) et avec (GT1 et GT2) k-anonymat

Dans le cas d'une application répartie, où les sources de données sont hétérogènes, Pareshi, par exemple, propose d'utiliser la généralisation des données pour protéger la vie privée dans les compositions de service pour les applications pervasives [Pareschi 2008].

D Synthèse sur la sécurité et le contrôle d'accès

Cette partie nous a permis de repérer les menaces auxquelles les applications informatiques, et plus particulièrement les applications pervasives, sont exposées. Protéger la vie privée nécessite de savoir protéger la confidentialité et l'intégrité des données. Le contrôle d'accès nous est apparu être une technique efficace pour y parvenir. Afin de déterminer les forces et les faiblesses des différentes formes de contrôle d'accès, nous les classés avons en fonction des contraintes qu'elles permettent d'exprimer. La Figure 2.27 présente notre catégorisation des modèles.

Type de contraintes	Sous-Type de contraintes	Contraintes	Modèles de contrôle d'accès
Contraintes statiques	Contraintes structurelles	Rôles des sujets	RBAC - ABAC (pas natif) - OrBAC
		Rôles des objets	RBAC étendu - ABAC
	Authentification	Identification	
Contraintes dynamiques	Contraintes sur les activités	Séparation des privilèges Liaison des privilèges	RBAC-ABAC
		Cardinalité	ABAC
		État d'achèvement	ABAC
	Contraintes sur l'environnement	Attributs de l'environnement Attributs des sujets Attributs des objets	RBAC étendu - ABAC - OrBAC

FIGURE 2.27 – Catégorisation des modèles en fonction des contraintes

Dans le cas des applications pervasives, les modèles doivent permettre de représenter des politiques d'accès complexes. Les modèles dérivés de RBAC ne sont pas nécessairement adaptés au cadre pervasif dans la mesure où le nombre de rôles explose avec le nombre d'attributs à prendre en compte. ABAC est un modèle très souple qui permet de représenter des contraintes complexes et nombreuses. Il est ainsi bien adapté au cadre pervasif. Néanmoins, les politiques de ABAC sont complexes à lire.

Prendre en compte le contrôle d'accès dès la conception la conception d'une application est nécessaire. Le modèle choisi influence l'architecture de l'application en imposant le maintien de la représentation d'un certain nombre d'informations.

Néanmoins, il existe une grande distance entre les concepts de haut niveau manipulés par le contrôle d'accès qui peuvent être aisément compris par des personnes ne possédant pas de capacités techniques, et le niveau technique qui implémente le contrôle d'accès. L'implémentation du contrôle d'accès reste ainsi une problématique de recherche.

III Conclusion à l'état de l'art

À l'issue de notre état de l'art, il apparaît que l'approche orientée services est une manière prometteuse de réaliser des applications pervasives. Les efforts de prise en compte du dynamisme, le caractère réparti des applications à services et la variété des types de services sont en effet trois propriétés qui permettent d'envisager la construction d'applications dynamiques à partir de services hétérogènes. Néanmoins, les applications pervasives imposent aux approches à services des challenges qui restent, à ce jour des perspectives de recherche.

Tout d'abord, l'intégration de services hétérogènes et dynamiques au sein d'une même application reste perfectible. Certaines solutions, comme la composition par procédés, ne sont en effet possibles que pour certains types de services, le plus souvent les services web. Il faut donc les étendre à tous les services.

Ensuite, l'informatique pervasive nécessite la récolte d'un grand nombre de données comme, par exemple, la localisation des utilisateurs, et plus généralement les données liées à leur contexte d'utilisation. De plus, dans la mesure où l'informatique pervasive vise à être intégrée à notre environnement, un utilisateur peut accéder à une application à partir de plusieurs contextes d'utilisation, certains pouvant être légitimes alors que d'autres ne le sont pas, ce qui peut causer des fuites de données. Il faut par conséquent limiter la diffusion des données et des accès en fonction du contexte.

Par conséquent, l'approche orientée services reste, aujourd'hui, encore, exposée à de nombreuses failles de sécurité qui, pour être traitées de manière efficaces, doivent être prises en compte dès la conception d'une application. La protection des données en général est en effet un préalable à la protection de préoccupations particulières, comme la vie privée. Aider les concepteurs à prendre en compte de manière optimale la protection des données dans le contexte de compositions de services pour l'informatique pervasive reste ainsi un problème ouvert.

Chapitre 3

Contribution

Les procédés sont des applications collaboratives qui nécessitent le partage de données, souvent sensibles. Ils sont de plus en plus souvent exécutés dans des environnements pervasifs et dynamiques. L'état de l'art nous a permis de répertorier les différents modèles de contrôle d'accès et de montrer qu'ils restent difficiles à mettre en œuvre dans les orchestrations de services. Il est complexe de définir une politique de contrôle d'accès à l'échelle d'une composition car le client d'un service ne contrôle pas ce dernier. Dans le cas du contrôle d'accès pervasif, la collecte de données peut mettre en danger la vie privée des utilisateurs en entraînant la récolte d'un grand nombre d'informations.

Par conséquent, il manque un moyen de configurer les services afin de garantir l'exécution d'une politique de contrôle d'accès. Ce manque est particulièrement criant dans les orchestrations qui utilisent des services hétérogènes et dynamiques. Dans la mesure où on ne sait jamais quel service sera appelé, il est impossible de supposer qu'ils seront capable d'exécuter une politique de contrôle d'accès. Nous soutenons que le contrôle d'accès est un moyen efficace de protéger la vie privée des possesseurs des données qui sont produites, transformées et consommées dans les orchestrations de services hétérogènes et dynamiques. Nous entreprenons ici de combler ce manque. Notre but est de permettre aux concepteurs d'applications de prendre en compte au mieux les politiques de contrôle d'accès dans la réalisation de compositions de services pour les applications pervasives.

Nous proposons pour ce faire une démarche dirigée par les modèles qui se divise en deux temps. La phase de conception permet de spécifier, à un niveau abstrait, l'orchestration et la politique de contrôle d'accès qui lui est associée. La phase d'exécution est chargée d'exécuter la composition de services sous la forme d'une orchestration. La phase d'exécution permet de sélectionner et de configurer les services concrets nécessaires à l'exécution de l'orchestration sécurisée par le contrôle d'accès.

Les apports principaux de notre proposition sont :

- **un métamodèle d'orchestration de services hétérogènes et dynamiques** et un métamodèle de contrôle d'accès pervasif ;
- **Un mécanisme de liaison de ces métamodèles** afin de permettre à chaque expert de spécifier la vue de l'orchestration dont il est responsable ;
- **Une architecture d'exécution d'une orchestration** chargée de gérer le dynamisme et l'hétérogénéité des services, de configurer leurs propriétés de contrôle d'accès et de protéger la vie privée des utilisateurs.

Nous présentons dans cette partie notre contribution. Dans la Section 1, nous introduisons les principes généraux de notre travail. Nous présentons tout d'abord les

fondements conceptuels de notre approche avant d'établir notre problématique et nos objectifs. Nous donnons ensuite une vision globale de notre travail. Dans la Section 2, nous entrons en détail dans notre proposition. Nous commençons par expliquer la structure de nos métamodèles, leurs liens et les transformations de modèles que nous avons établies. Nous présentons ensuite l'architecture orientée services que nous proposons pour exécuter des compositions de services pour les applications pervasives respectueuses de la vie privée. La troisième Section est consacrée à l'implémentation de notre approche et à sa validation. Enfin, nous donnons dans la quatrième Section les perspectives que nous envisageons à notre travail.

I Présentation générale de la démarche

Nous présentons dans ce chapitre une démarche dirigée par les modèles qui permet de sécuriser une orchestration de services hétérogènes et sécurisés à l'aide du contrôle d'accès. Nous commençons par introduire les fondements conceptuels qui dirigent notre travail puis nous formulons notre problématique et nos objectifs. Enfin, notre démarche fait l'objet de la dernière étape de ce chapitre.

A Fondements conceptuels : Sécurité dirigée par les modèles [1]

La littérature sur le contrôle d'accès s'accorde pour le considérer comme nécessaire dans une approche de protection de la vie privée dans la mesure où il restreint le visbilité des données sensibles. Cependant, l'état de l'art montre qu'il subsiste deux difficultés dans la réalisation d'applications sécurisées par le contrôle d'accès. Les failles de contrôle d'accès ne peuvent être recherchées uniquement lorsque l'application est terminée. Ceci reviendrait à laisser ces failles ouvertes. Il est donc nécessaire d'établir des modèles de contrôle d'accès qui permettent de spécifier certaines bonnes propriétés dès la conception de l'application et et de tester leur respect. La seconde difficulté tient à la difficulté de l'implémentation du contrôle d'accès et à la nécessité de pouvoir faire évoluer la politique de contrôle d'accès au cours du cycle de vie de l'application. Cette difficulté est encore accentuée dans les compositions de services hétérogènes et dynamiques. Il est nécessaire de maintenir plusieurs versions de la politique de contrôle d'accès pour chaque service cible en fonction de son implémentation. L'exécution de la politique doit s'adapter aux services disponibles et prendre en compte leur arrivée et leur départ dans la composition.

Afin de prendre en compte le contrôle d'accès dès la conception d'une composition d'une application, des liens doivent exister entre les langages de modélisation des fonctionnalités de l'application et les modèles de contrôle d'accès. C'est dans ce but qu'a été proposée la sécurité dirigée par les modèles, méthode de développement fondée sur l'ingénierie dirigée par les modèles. Nous présentons tout d'abord les notions fondamentales de l'ingénierie dirigée par les modèles avant d'introduire la sécurité dirigée par les modèles.

1 Principes de l'ingénierie dirigée par les modèles

L'Ingénierie Dirigée par les Modèles (IDM) est une méthode de développement informatique qui prend sa source dans le génie logiciel assisté par ordinateur (noté CASE pour *Computer-Aided Software Engineering*) qui vise à faciliter le développement des logiciels, notamment à partir d'outils graphiques [Fuggetta 1993]. L'IDM connaît aujourd'hui encore plusieurs définitions. Cependant, Atkinson remarque que l'IDM est caractérisée de la manière suivante :

> *Instead of requiring developers to use a programming language spelling out how a system is implemented, it allows them to use models to specifying what system functionality is required and what architecture is to be used. [Atkinson 2003]*

Le but de l'IDM est de faciliter le travail des développeurs en les libérant des considération techniques sur l'implémentation des applications et en leur permettant de se concentrer sur ce que fait l'application plutôt que comment elle le fait. En ce sens, l'IDM se rapproche de la programmation déclarative qui décrit ce qu'est une

1. Nos fondements conceptuels ont été formulé à partir de la sécurité dirigée par les modèles et des travaux de Stéphanie Chollet sur la prise en compte de la sécurité dans les composition de services hétérogènes et dynamiques. Voir [Chollet 2009a]

application, par opposition à la programmation impérative qui décrit comment une application réalise ses fonctionnalités.

Pour ce faire, l'IDM repose sur des artefacts nommés modèles. Les modèles visent à être réutilisables dans la mesure où ils sont le plus souvent indépendants d'une plateforme spécifique. Les modèles permettent aussi de s'abstraire de la complexité des applications. La Figure 3.1 présente la structure de l'IDM et sa comparaison avec des moyens non informatiques de représentation du monde.

L'OMG propose une implémentation de l'IDM sous la forme de l'architecture dirigée par les modèles (notée MDA [2] pour *Model-Driven Engineering*) qui met l'accent sur le développement de logiciels à partir de modèles. Cependant, d'autres approches proposent d'utiliser les modèles à l'exécution des applications pour en maintenir une représentation à jour [Denker 2010].

La « sécurité dirigée par les modèles » [Jensen 2011], par exemple, est une utilisation particulière de l'IDM pour le développement d'applications sécurisées. Avant d'en présenter l'état actuel, nous introduisons les notions de modèle et de métamodèle, qui constituent les fondements de l'IDM.

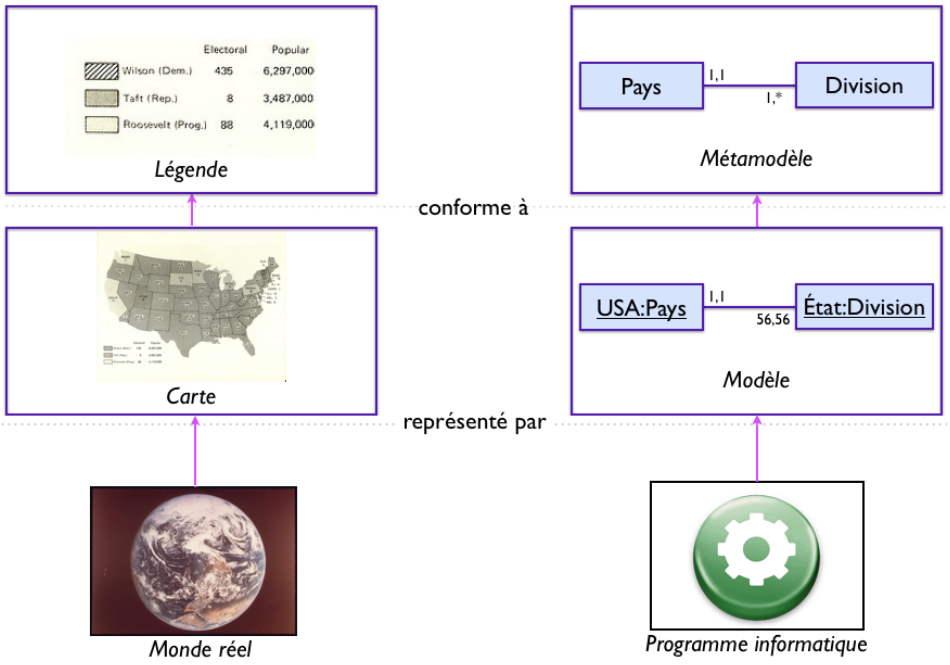

FIGURE 3.1 – Définition des notions de modèles et de métamodèles

1.1 Définition de la notion de modèle La notion de modèle ne possède pas de définition consensuelle. Cependant, la définition synthétique de Jean Bezivin est au fondement de la plupart des propositions en la matière. L'auteur affirme que

A model is a representation of a system. [Bézivin 2005]

2. http ://www.omg.org/mda/

Par analogie, et comme illustré sur la Figure 3.1, un modèle est à un programme informatique ce qu'une carte est à un territoire. Un modèle présente un point de vue spécifique sur un système. Par conséquent, un modèle est conçu pour un destinataire particulier, et plusieurs modèles d'un même système peuvent coexister afin d'en présenter les multiples aspects. Cette présentation est souvent réalisée sous la forme d'un graphe mais cette représentation graphique est optionnelle.

Les modèles, en temps que spécifications, sont le support de la réalisation d'un logiciel qui peut être, partiellement au moins, automatisée. Des transformations de modèle en texte permettent ainsi de générer du code à partir des modèles [Jouault 2008]. Les modèles sources de la génération de code, qui permettent de décrire un système qui n'existe pas sont dits prescriptifs, dans la mesure où ils spécifient ce que doit être un système [Bézivin 2005]. Les modèles, dans la mesure où ils ne sont pas nécessairement liés à l'implémentation d'un système permettent de réaliser les spécifications qu'ils présentent de plusieurs manières et donc d'en favoriser l'interopérabilité. Enfin, les modèles permettent aussi la mise en œuvre de simulation d'une application et plus généralement l'utilisation du *model-checking*, c'est-à-dire la vérification dès la phase de spécification d'un système de la satisfaction d'un ensemble de propriétés. Schmeling et al. [Schmeling 2012] proposent ainsi de vérifier dès la conception si plusieurs propriétés non fonctionnelles d'une application n'entrent pas en conflit.

Dans le domaine de la vie privée, l'utilisation de l'IDM pour concevoir des applications peut être un moyen de mettre en œuvre le respect de la « vie privée par conception » (autrement appelée *privacy by design* [Spiekermann 2012]). La génération d'une application à partir de modèles peut en effet potentiellement garantir qu'une implémentation correspond à un ensemble de contraintes sur la gestion de la vie privée. Néanmoins, l'application de l'IDM à la « vie privée par conception » reste à ce jour une perspective de recherche.

L'utilisation de l'IDM dans le cadre de l'approche orintée services est particulièrement efficace [Orriëns 2003]. En effet, dans la mesure où les services sont faits pour être réutilisés puisqu'ils ne gèrent pas leur contexte d'exécution et permettent de se concentrer sur les fonctionnalités qu'ils offrent, une composition de services peut être facilement décrite de manière déclarative plutôt qu'impérative. De plus, les applications composées services sont complexes car elles mettent le plus souvent en jeu un grand nombre de services et mobilisent des plateformes hétérogènes pour gérer leur distribution. Il est ainsi nécessaire de s'abstraire de ces contraintes techniques pour se concentrer sur ce que fait l'application. Lorsqu'une application met en jeu des services dynamiques, il est impossible de prévoir à la conception comment l'application va s'exécuter. Par conséquent, modéliser l'application et retarder la sélection des services pertinents parmi les services disponibles à l'exécution permet de construire des applications adaptables.

Enfin, de nouvelles utilisations de l'IDM comme l'auto-explication des applications, c'est-à-dire leur capacité à décrire leur fonctionnement émergent aujourd'hui [García Frey 2010]. Dans cette perspective, les modèles ne spécifient plus uniquement ce qu'un système doit être. Il en décrivent l'état à un moment donné

[Bézivin 2005].

1.2 Notion de métamodèle La définition de la notion de modèle nous a permis de préciser le sens à donner à la relation de représentation entre le modèle et les systèmes qu'ils représentent. Comme l'illustre la Figure 3.1, les cartes géographiques sont conformes à leur légende qui précise les éléments utilisables sur la carte et leurs relations. Par analogie, un modèle obéit à un ensemble de règles qui sont rassemblées sous la forme d'un métamodèle que Jean Bezivin définit de la manière suivante :

> *A metamodel is a formal specification of an abstraction, usually consensual and normative. [Bézivin 2005]*

En d'autres termes, un métamodèle fournit un vocabulaire et une grammaire nécessaires à l'élaboration de modèles. XACML, par exemple, peut être considéré comme un métamodèle qui permet de construire des politiques de contrôle d'accès qui constituent dès lors des modèles. Les diagrammes d'UML obéissent tous au métamodèle d'UML qui précise les entités qui peuvent être utilisées, leurs relations et les contraintes qui s'appliquent à elles. Par conséquent, contrairement aux ontologies qui permettent de représenter des connaissances en général, les métamodèles portent sur des domaines spécifiques liés à une application. Le contrôle d'accès, la structure ou le comportement d'une application sont trois exemples de ces domaines.

Par suite, l'utilisation principale des métamodèles est la séparation des préoccupations, c'est-à-dire, l'analyse des différents points de vue possibles sur le système et leur circonscription [Vallecillo 2010]. Chaque métamodèle décrit les concepts d'un domaine particulier et leurs relations. La séparation des préoccupations est nécessaire dans le cadre d'un travail collaboratif afin de permettre à plusieurs acteurs, avec des champs d'expertises différents, de représenter le système à l'aide de leur vocabulaire habituel.

1.3 Composition de métamodèles Afin d'obtenir une vision complète d'un système représenté à partir de plusieurs points de vue, et surtout dans le cadre d'une approche générative à base de modèles, il est nécessaire d'établir des liens entre les différents métamodèles utilisés [Vallecillo 2010]. Ces liens sont établis entre les concepts et les relations que mettent en jeu les métamodèles utilisés. Le cas le plus simple est le repérage de l'identité de concepts ou de relations dans plusieurs métamodèles.

Le repérage de liens entre plusieurs métamodèles repose sur l'existence d'un métamodèle commun auquel les différents métamodèles à unir sont conformes. Dans le cadre de *Model Driven Architecture* (MDA) [3], l'OMG propose ainsi le *Meta-Object Facility* (MOF), auquel sont conformes tous les modèles nécessaires à MDA. La plateforme Eclipse permet, elle, de décrire les métamodèles au format *ecore* [4]. L'existence de ces métamodèles permet le développement d'un grand nombre de métamodèles de domaines pour constituer des langages de modélisation spécifiques à un

3. Voir http ://www.omg.org/mda/
4. Voir http ://www.eclipse.org/modeling/emft/ ?project=ecoretools

domaine (notés DSML pour *Domain Specific Modeling Language*) tout en garantissant qu'ils pourront être associés pour construire une vision globale d'un système. En effet, ils se fondent tous sur les mêmes entités de bases utilisées par leurs métamodèles.

Les liens identifiés au niveau des métamodèles sont propagés à l'instanciation des modèles. Dans le cadre d'une approche générative, ces liens peuvent être utilisés pour imbriquer le code généré à partir de chaque point de vue sur le système. La programmation orientée aspect propose ainsi de considérer chaque point de vue sur un système comme un « aspect ». Le code des aspects, qui peut être généré à partir d'un ensemble de modèles, est tissé à l'exécution par un tisseur d'aspects comme AspectJ pour le langage Java [5].

La Figure 3.2 reprend les éléments nécessaires à la mise en place d'une approche générative et leurs liens.

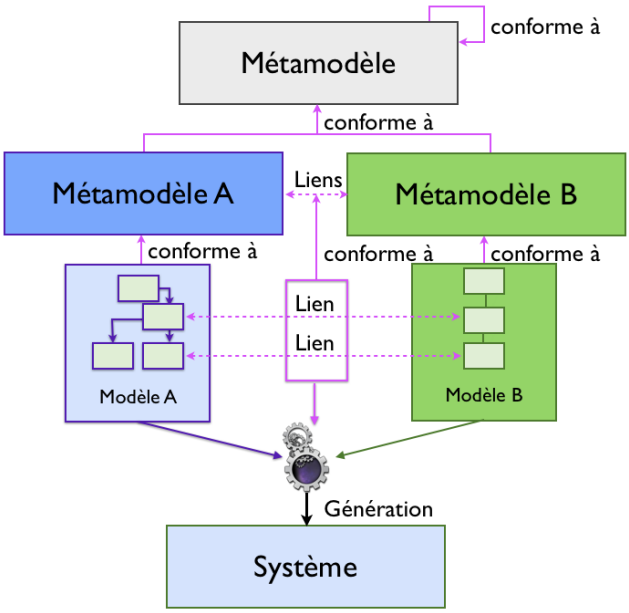

FIGURE 3.2 – Structure d'une approche générative

1.4 Sécurité dirigée par les modèles Inspirés par l'IDM, Basin et al. [Basin 2006], proposent d'étendre UML avec des annotations de contrôle d'accès. Wolter [Wolter 2007b, Wolter 2007a] et Rodriguez [Rodríguez 2007] font de même avec les notations dédiées aux processus (BPMN, etc.).

Les liens entre modèles fonctionnels et modèles de contrôle d'accès reposent sur l'association des grammaires des langages fonctionnels et non fonctionnels. Il permettent de concevoir une application sécurisée à un haut niveau d'abstraction afin de d'abstraire des difficultés techniques liées à l'implémentation. Une fois les

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

modèles conçus, ils peuvent servir de guides pour l'implémentation manuelle ou bien être intégrés à une démarche dirigée par les modèles. Basin et al. [Basin 2006] proposent ainsi, à partir de modèles UML étendus par des contraintes de contrôle d'accès, de générer les squelettes de composants utilisables dans le cadre d'une architecture CORBA réalisés sous la forme d'*Enterprise Java Beans*. Les squelettes obtenus implémentent le contrôle d'accès.

Wolter [Wolter 2007b, Wolter 2007a] et Rodriguez [Rodríguez 2007] appliquent une démarche similaire aux processus métiers. Les auteurs représentent les processus à l'aide de BPMN, qu'ils annotent de contraintes de contrôle d'accès. Les modèles BPMN étendus obtenus sont utilisés par les auteurs pour générer du code XACML. Cependant, les auteurs ne s'intéressent pas à l'exécution du contrôle d'accès ou bien à la génération de l'application elle-même. Leur démarche doit ainsi être complétée afin d'aider à la réalisation d'applications exécutables.

La Figure 3.3 reprend les notions principales de la sécurité dirigée par les modèles.

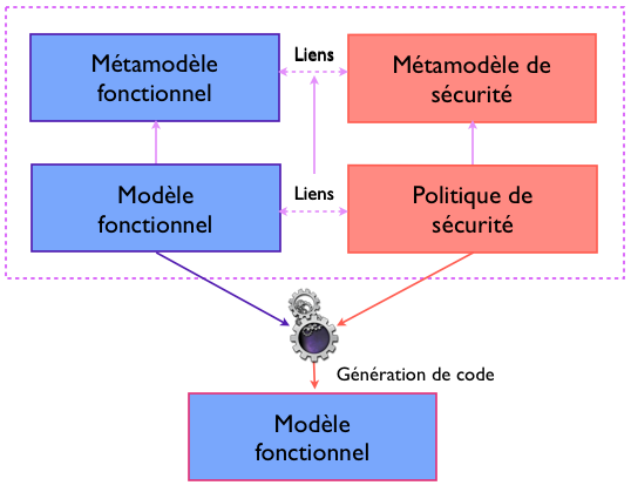

FIGURE 3.3 – Sécurité dirigée par les modèles

Cette approche permet potentiellement de prendre en compte le contrôle d'accès dès la conception d'une application. Dans [Chollet 2009a] [Chollet 2008], S. Chollet propose une démarche dirigée par les modèles de réalisation d'une orchestration de services hétérogènes et dynamiques sécurisée grâce à un métamodèle d'orchestration abstraite associé à un métamodèle de sécurité. L'auteure propose, à partir de génération automatique de code, de construire une orchestration sécurisée qui prend en compte l'hétérogénéité et le dynamisme des services. Notre travail constitue une application de la sécurité dirigée par les modèles, et particulièrement de la version qu'en donne [Chollet 2009a] à la protection de la vie privée à l'aide du contrôle d'accès.

2 Problématique et objectifs

Dans le cadre de notre travail, nous nous concentrons sur la protection de la vie privée. Nous définissons cette protection comme un problème de maintien de la confidentialité et de l'intégrité des données. Lorsqu'il s'agit d'assurer la protection de la vie privée dans les applications pervasives réalisées sous la forme de compositions de services, l'état de l'art que nous avons dressé nous permet de parvenir à un ensemble de constats, que nous plaçons à la base de notre travail et présentons ci-après.

- **La construction d'une application pervasive est techniquement complexe**. Les entités qui la composent sont réparties, dynamiques et hétérogènes. L'hétérogénéité s'entend aussi bien au niveau des technologies qui permettent d'implémenter les applications pervasives que des éléments concrets implémentés dans ces technologies. Ces éléments sont repartis dans l'espace et connectés par un réseau qu'ils peuvent quitter et rejoindre à tout moment. L'état de ces éléments change fréquemment.
- **La protection de la vie privée dans une application pervasive est difficile**. La récolte de données inhérentes aux applications pervasives rend non seulement nécessaire la protection des données utilisées par les fonctionnalités de l'application mais aussi la protection des données des utilisateurs récoltées afin de capturer leur contexte. Dans la mesure où l'accès à l'application peut se faire de n'importe où et n'importe quand, la modélisation et le traitement du contexte sont des éléments centraux de la protection de la vie privée.
- **Le contrôle d'accès est une technique efficace pour protéger la vie privée.** Le contrôle d'accès permet d'assurer la protection de la confidentialité et de l'intégrité d'un ensemble de données. La protection de la vie privée repose sur la protection de l'intégrité et de la confidentialité des données ainsi que sur la protection d'un ensemble de données spécifiques.
- **Le contrôle d'accès doit être pris en compte à l'échelle d'une application**. Les droits d'accès sont dépendants non seulement des caractéristiques des acteurs mais de l'état plus général de l'application. Cela est particulièrement vrai lorsque plusieurs activités réalisées par une application sont enchaînées et que le résultat d'une activité influence l'exécution des activités suivantes. Cela est aussi particulièrement vrai lorsqu'un même utilisateur peut faire appel à un service à partir de plusieurs contextes d'utilisation qui influent sur ses droits d'accès.
- **La mise en œuvre du contrôle d'accès dans une application pervasive est complexe**. Le code d'exécution du contrôle d'accès doit être adapté à chaque entité concrète et doit permettre la dynamicité des droits. Les acteurs qui interagissent avec l'application, mais aussi les entités concrètes qui réalisent les fonctionnalités offertes par l'application, ne sont connus qu'à l'éxécution. La gestion des droits des utilisateurs doit donc être conditionnée au contexte de l'application.

- **L'architecture à services permet la constitution d'applications pervasives mais laisse ouvert le problème de l'application du contrôle d'accès**. Des entités aussi diverses que des capteurs, des logiciels, des infrastructures ou encore des données peuvent être exposées comme des services. Les architectures à services sont ainsi bien adaptées au domaine pervasif. La liaison retardée des services permet de choisir à l'exécution les services nécessaires. Pourtant, l'hétérogénéité subsiste au sein même d'une architecture à services car plusieurs standards – UPnP, DPWS, etc. - de description des services coexistent. Enfin, les architectures orientées services ne fournissent pas de mécanismes dédiés au contrôle d'accès.

Notre objectif est de permettre aux concepteurs d'applications de prendre en compte la protection de la vie privée par la mise en œuvre du contrôle d'accès dans les applications pervasives réalisées sous la forme d'un procédé implémenté comme une orchestration de services. Afin de réduire le cadre de notre travail, nous faisons deux hypothèses :

- Nous nous situons dans un cadre dans lequel les services disponibles sont fiables : ils respectent l'accord de service sans chercher à effectuer d'actions cachées. Nous ne traitons pas de la confiance accordée au service.
- Nous cherchons à contrôler la diffusion des données mais nous ne traitons pas le problème de l'inférence de données protégées à partir des données auxquelles les utilisateurs ont accès.

Nous estimons que le contrôle d'accès doit être pris en compte dès la conception de l'application dans des termes qui s'abstraient des difficultés de l'implémentation. Nous pensons aussi que ce code doit être généré automatiquement, modifié et maintenu simplement.

Pour ce faire, nous proposons une démarche dirigée par les modèles qui permet de configurer la politique de contrôle d'accès au niveau de la composition de services. Sa mise en œuvre n'intervient qu'à l'exécution. Cette démarche est divisée en deux phases :

- **Première phase : conception**. Le concepteur de l'application définit les fonctionnalités nécessaires à l'application et le procédé qu'elles permettent de réaliser. Un expert en sécurité annote ensuite ces fonctionnalités avec des règles de contrôles d'accès. Cette phase ignore la complexité de l'implémentation des activités réalisées par des services et de la gestion du contexte.
- **Seconde phase : exécution**. Tous les services sont encapsulés dans un *proxy* qui garantit l'exécution du contrôle d'accès. Nous proposons d'inclure dans le *proxy* un service dédié à l'interrogation des sources contextuelles et un service consacré à l'évaluation de la politique de contrôle d'accès. Le *proxy* est aussi chargé de notifier les utilisateurs des tâches qu'ils doivent accomplir et de gérer l'interaction entre les utilisateurs et l'application.
- **Le passage de la première à la seconde phase est réalisé par génération automatique de code**. Il permet d'obtenir le code fonctionnel adapté à la plate-forme cible de manière transparente pour le concepteur.

Cette démarche est implémentée sous la forme d'un environnement de développement et d'exécution qui fournit notamment un éditeur de modèles, un outil de génération automatique de code ainsi qu'un moteur d'exécution du contrôle d'accès pervasif.

3 Présentation de la proposition

Dans cette partie, nous introduisons notre proposition, une démarche dirigée par les modèles pour la réalisation d'applications pervasives respectueuses de la vie privée, ainsi que les principes qui la motivent. Après avoir présenté la structure globale de notre démarche dirigée par les modèles, nous exposons les principes qui en guident les étapes de conception et d'exécution.

3.1 Approche globale Notre objectif est de faciliter la production d'applications pervasives respectueuses de la vie privée réalisées sous la forme de compositions de services. Pour ce faire, nous proposons une démarche dirigée par les modèles présentée sur la Figure 3.4.

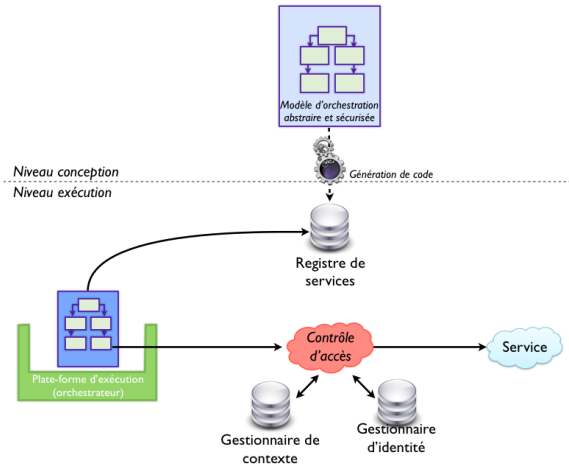

FIGURE 3.4 – Approche globale

Notre approche repose sur deux phases :

1. **La phase de conception** permet de capturer les propriétés fonctionnelles de la composition, sous la forme d'un procédé. Elle permet aussi de capturer, de manière séparée, la politique de contrôle d'accès qui s'applique à ce procédé.
2. **La phase d'exécution** permet l'invocation des services nécessaires à la réalisation de l'application. L'invocation est contrainte par la mise en œuvre du contrôle d'accès qui protège l'accès aux données dans l'application.

Nous avons implémenté ces deux phases sous la forme d'un environnement de modélisation et d'un ensemble d'outils pour l'exécution d'une composition de services dans laquelle s'applique une politique de contrôle d'accès.

L'environnement de modélisation permet de représenter un procédé, les échanges de données qui s'y déroulent et la politique de contrôle d'accès qui lui est associé. Ce procédé est défini de manière abstraite de manière à s'abstraire de l'hétérogénéité des services potentiellement disponibles et de leur dynamisme. Afin de s'adapter aux services effectivement disponibles, nous ne choisissons les services à invoquer qu'à l'exécution. La politique de contrôle d'accès est présentée dans une vue séparée de celle du procédé et est elle-même définie de manière abstraite.

Au niveau exécution, nous avons choisi de réaliser le procédé sous la forme d'une orchestration de services. Nous proposons un ensemble d'outils pour invoquer les services effectivement disponibles et nécessaires à l'exécution des activités du processus à partir de leurs spécifications abstraites. Nous nous appuyons notamment sur un registre de services qui permet de maintenir une représentation des services disponibles et de leurs propriétés. Sur le plan du contrôle d'accès, nous proposons d'encapsuler les services car on ne peut pas garantir qu'ils offrent les bonnes propriétés de contrôle d'accès. Nous maintenons une représentation de l'état de la composition et du contexte des utilisateurs de manière à pouvoir vérifier les contraintes de contrôle d'accès contextuelles. Enfin, nous ajoutons à l'orchestration de services un gestionnaire d'identités chargé de recenser les privilèges détenus par les utilisateurs. Le niveau exécution est obtenu grâce à la génération du code qui permet d'encapsuler les services. Cette phase repose sur des *templates* spécifiques à chaque type de services supportés. Le code obtenu est complété avec des informations spécifiques au service considéré.

4 Niveau conception

4.1 Abstraction La composition de services a pour but de construire une application à partir de services existants ou non. La Figure 3.5 présente les mécanismes disponibles dans le cadre de l'approche orintée services pour s'abstraire de la disponibilité des services.

FIGURE 3.5 – Mécanismes d'abstraction dans les SOA

On remarque une grande hétérogénéité de l'implémentation des services : une même fonctionnalité peut être offerte par plusieurs services qui peuvent être implémentés de manière très différente. Les services peuvent être classés selon les technologies à service qu'ils implémentent. Ils peuvent aussi être classés en fonction des fonctionnalités qu'ils offrent et de leurs autres propriétés. Ces propriétés sont représentées sous la forme d'un service abstrait qui prend la forme d'une description de service.

Ce mécanisme d'abstraction n'est cependant pas suffisant. Nous avons en effet vu que chaque technologie à service repose sur un type de description de service spécifique ainsi que sur une architecture particulière. Il est par conséquent nécessaire de s'abstraire des technologies qu'implémentent les différents services.

Pour ce faire, nous proposons de représenter une composition de services de manière abstraite sous la forme d'un modèle. Cette technique d'abstraction est couramment utilisée pour représenter un point de vue particulier d'une application. Par exemple, les diagrammes UML permettent de représenter une application du point de vue de sa structure ou de ses comportements de manière plus compréhensible que le code de l'application. BPMN permet, lui de représenter un processus sans se soucier de son implémentation.

Dans notre cas, il est nécessaire de définir les services abstraits, leur communication et leurs propriétés de contrôle d'accès. Pour ce faire, nous spécifions un métamodèle qui sépare la description des services abstraits en deux parties :

- **L'interface fonctionnelle** permet de décrire les fonctionnalités offertes par le service, c'est-à-dire ses méthodes. La signature des méthodes permet de décrire les messages que doit recevoir le service et qu'il peut envoyer.
- **Les propriétés de contrôle d'accès** sont exprimées sous la forme de règles qui permettent, pour chaque service, de décrire les restrictions qui gouvernent son accès. Ces règles sont contextuelles.

Cette approche a pour avantage de faire abstraction des détails techniques des technologies que les services implémentent et permet ainsi de se concentrer sur la définition de la composition. Dès lors, la composition peut-être facilement réutilisée et sa réalisation adaptée aux services disponibles à l'exécution de l'application. À l'exécution, il ainsi nécessaire de générer le code qui permet d'appeler les services disponibles et sélectionnés.

4.2 Séparation des préoccupations et composition de métamodèles La conception d'applications pervasives sécurisées nécessite la collaboration de plusieurs experts. Les experts métiers, par exemple, sont chargés de concevoir la logique métier de l'application. Cependant, il est probable qu'ils ne maîtrisent pas la conception d'une politique de sécurité, qui doit dès lors être conçue par des experts spécialisés. Enfin, pour des raisons légales, il est nécessaire d'offrir la possibilité aux sujets des données de configurer la gestion de leurs données. Cependant, ils ne doivent pas pouvoir modifier les fonctionnalités de l'application ou perturber son fonctionnement pour d'autres utilisateurs.

Par conséquent, il faut offrir à chaque expert et aux utilisateurs une vue sur l'application qui corresponde à leurs préoccupations. Chaque vue doit être indépendante afin de ne pas corrompre les autres. Pour ce faire, nous définissons un métamodèle pour chaque préoccupation, c'est-à-dire un métamodèle pour la composition de services et un métamodèle de contrôle d'accès. Des liens sont définis entre ces métamodèles afin de permettre de les associer et d'obtenir une vue globale sur l'application. Notre travail vise ainsi à séparer les préoccupations que sont les fonctionnalités de l'application et la politique de vie privée capturée sous la forme de règles de contrôle d'accès. À l'exécution, les liens entre les métamodèles sont instanciés afin de réaliser la synthèse des deux préoccupations.

4.3 Synthèse du niveau conception La Figure 3.6 présente une vue globale du niveau conception de notre proposition pour la conception d'applications pervasives sécurisées par le contrôle d'accès. Notre travail est dirigé par les principes d'abstraction – afin de gérer le dynamisme et l'hétérogénéité des services – et de séparation des préoccupations – afin d'offrir à chaque concepteur une vue sur l'application qui lui est adaptée.

Nous associons deux métamodèles – un métamodèle pour la composition de services et un métamodèle pour le contrôle d'accès – en définissant des liens entre eux. Nous proposons aussi des transformations de modèle vers texte pour automatiser la génération de l'application à partir des modèles construits par les concepteurs.

Nous présentons désormais les grandes lignes de notre proposition pour l'exécution d'une composition de services hétérogènes et dynamiques sécurisée par le contrôle d'accès.

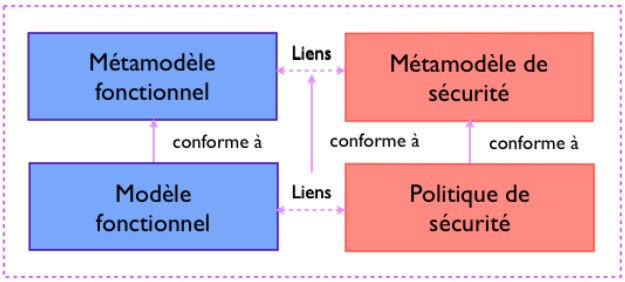

FIGURE 3.6 – Synthèse du niveau conception

5 Niveau Exécution

À l'issue de la phase de conception, nous possédons un ensemble de modèles qui représentent l'application de manière abstraite. Il faut rendre ces spécifications concrètes tout en se soumettant à deux contraintes. Tout d'abord, les spécifications doivent être adaptées aux services existants, aussi hétérogènes soient-ils. Ensuite,

l'architecture orientée services dans laquelle s'exécute l'application doit pouvoir gérer le dynamisme des services et l'exécution du contrôle d'accès.

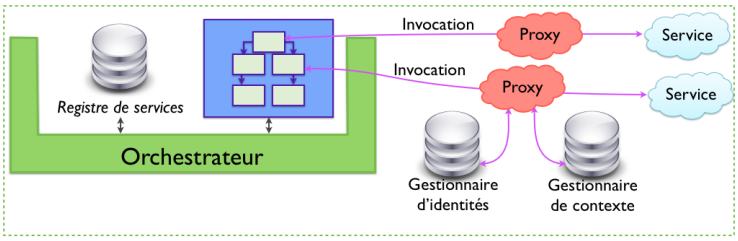

FIGURE 3.7 – Niveau exécution

Nous présentons désormais les principes qui guident notre proposition d'architecture conforme à ces deux contraintes. La Figure 3.7 en présente une vue synoptique.

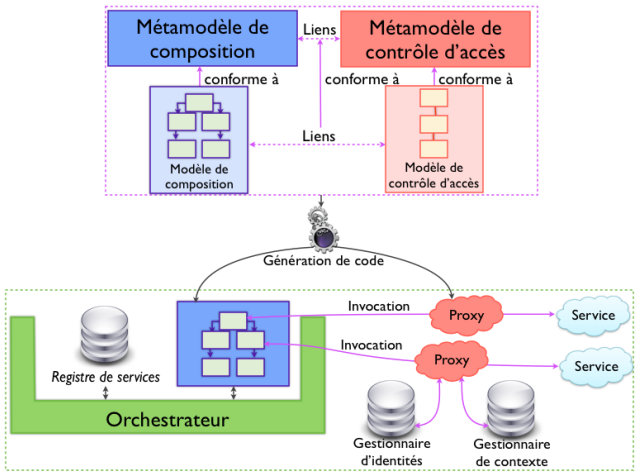

FIGURE 3.8 – Architecture de notre proposition

5.1 Génération de code et encapsulation des services À l'exécution, les services disponibles sont hétérogènes et dynamiques. Il est ainsi impossible de se fier à leur capacité à exécuter la politique de contrôle d'accès définie au niveau conception. Afin de configurer les services, nous proposons de les protéger à l'aide d'un *proxy* chargé d'intercepter les invocations et de jouer le rôle de point d'application du contrôle d'accès. Pour gérer l'hétérogénéité des services, les *proxy* sont générés à l'exécution de l'application pour chaque service. Afin de garantir la sécurisation de l'application, les *proxy* sont générés automatiquement lors de l'enregistrement des services dans le registre de service. Cette génération automatique des *proxy* est réalisée à l'aide de transformation de modèles vers texte. Le registre ne conserve

que l'adresse des *proxy* et non pas des services disponibles. Toutes les invocations transitent ainsi par les *proxy* et sont donc sécurisés.

5.2 Gestion des connaissances et de la vie privée Afin d'exécuter l'application et sa politique de vie privée, il est nécessaire de maintenir une représentation de l'ensemble des services disponibles et de l'ensemble des informations contextuelles à propos de l'application, de son environnement et des utilisateurs Pour y parvenir, nous proposons des registres dédiés. Le registre des services permet de mémoriser les services et leurs fonctionnalités. Le gestionnaire des identités permet de mémoriser les utilisateurs et leurs privilèges ainsi que les sources d'informations sur leur contexte. Le gestionnaire de contexte regroupe, lui, les informations sur la composition et l'environnement dans lequel elle s'exécute.

La Figure 3.8 reprend l'architecture que nous proposons pour exécuter des applications pervasives respectueuses de la vie privée réalisées sous la forme de compositions de services.

II Conception et Exécution d'une orchestration de services hétérogènes et dynamiques sécurisée par le contrôle d'accès

Nous avons présenté de manière globale notre contribution dans la section précédente. L'objectif de cette section est de présenter en détail le niveau conception et le niveau exécution de notre approche ainsi que la transition qui permet de passer de l'un à l'autre. Nous montrons que notre proposition permet de dépasser l'hétérogénéité et le dynamisme des services en configurant les propriétés de contrôle d'accès des services. Elle permet aussi de minimiser la collecte des données nécessaires au contrôle d'accès pervasif.

Le premier chapitre est consacré aux deux métamodèles qui guident la conception d'une application réalisée sous la forme d'une orchestration de services hétérogènes et dynamiques sécurisée par le contrôle d'accès. Nous présentons un métamodèle pour l'orchestration et un métamodèle pour le contrôle d'accès. Le chapitre présente aussi les liens que nous établissons entre ces métamodèles afin d'obtenir une vue complète sur l'application.

Le second chapitre présente les modèles d'orchestration hétérogène et dynamique sécurisée par le contrôle d'accès que nous obtenons à l'issue de la phase de conception. Il présente aussi la composition de ces modèles.

Enfin, le troisième chapitre est consacré à l'architecture d'exécution que nous proposons. Il présente les extensions que nous apportons à l'architecture habituelle d'exécution des orchestrations de services.

A Niveau conception : Métamodèles et modèles d'une orchestration de services hétérogènes sécurisée à l'aide du contrôle d'accès

Nous introduisons dans cette section les métamodèles que nous utilisons pour représenter une orchestration de services hétérogènes et la politique de contrôle d'accès qui lui est associée.

1 Métamodèle de l'orchestration de services hétérogènes

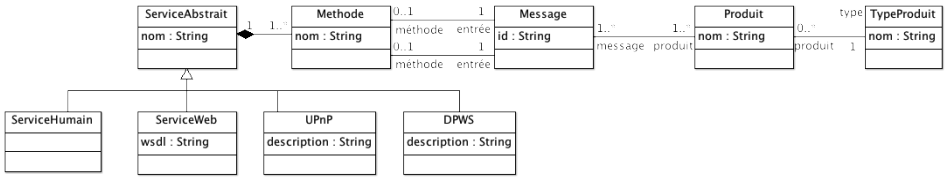

FIGURE 3.9 – Métamodèle de l'orchestration de services hétérogènes

À part quelques langages comme l'*Abstract Process Engine Language* (APEL) proposé par [Dami 1998], les langages de haut niveau pour la définition des procédés ne s'appliquent qu'à un ensemble restreints de services. Les orchestrations définies avec WS-BEPL, par exemple, ne peuvent utiliser que des services web. La plupart des langages de définition de procédés ne permettent pas de prendre en compte l'hétérogénéité des technologies et des implémentations des services.

De plus, ces langages imposent souvent au concepteur d'un procédé de connaître les services qu'il souhaite invoquer. Ainsi, dans WS-BEPL est-il nécessaire de préciser l'adresse des services web utilisés par la composition. Il est ainsi impossible d'utiliser la liaison retardée pour choisir, à l'exécution, les services à invoquer. Ces langages ne permettent donc pas de prendre en compte le dynamisme des services à l'exécution d'un procédé.

Afin d'y parvenir, nous proposons, au moment de la conception, de nous en tenir à la spécification d'une orchestration abstraite de tout détail technique. Nous appelons « service abstrait » l'interface fonctionnelle des services nécessaires à l'exécution de l'orchestration. Elle ne contient pas de détails techniques, ce qui permet de retarder la sélection d'un service spécifique à l'exécution. Par opposition un service concret est un service particulier implémenté dans une technologie donnée. La Figure 3.9 présente les concepts nécessaires à la description d'un service abstrait. Ils sont définis de la manière suivante :

- **Un service abstrait** représente un type de service qui peut être appelé de manière à effectuer une activité. Un service abstrait constitue la spécification d'un service, c'est-à-dire son interface fonctionnelle. Cette spécification est exprimée à un haut niveau d'abstraction et ignore le plus de détails liés à l'implémentation possible. Un service abstrait est défini :
  – par son nom ;

– de manière optionnelle, par des informations spécifiques à la technologie du services. Ces informations sont contenues dans les extraits de fichiers WSDL ou UPnP, par exemple, qui décrivent les services concrets. Ces informations sont indépendantes de l'implémentation du service contrairement, par exemple, aux fichiers WSDL complets. En fournissant ces informations, on sous-entend que la technologie du service est choisie lors de la conception de la composition, ce qui est fréquemment le cas. Ces informations sont ainsi nécessaires car elles permettent de générer un code d'appel du service plus efficace.

Chaque service abstrait est une interface fonctionnelle qui est composée de méthodes. Une méthode est une fonctionnalité offerte par un service. Chaque méthode est définie par sa signature à partir des concepts suivants :

- **Un Produit** est un objet abstrait que les services s'échangent : les services peuvent les produire, les transformer ou les consommer. Les produits constituent les entrées et les sorties des méthodes des services. Ils sont échangés au travers des messages. La sortie de la méthode d'un service peut ainsi être l'entrée de la méthode d'un autre service.
- **Un Message** permet de définir les produits qui transitent par les méthodes. Un message d'entrée contient les paramètre nécessaires à l'exécution de la méthode. Si la méthode a un type de retour, elle renvoie un message de sortie.
- **Un Type de produit** permet de spécifier, sous la forme d'une chaîne de caractère, le type des produits échangés par les services.

Un service abstrait est composé de méthodes : il offre un ensemble de fonctionnalités. Les méthodes acceptent des messages d'entrée et produisent des messages de sorties. Tous les messages contiennent des produits qui ont un type.

Nous voyons les concepts que nous venons de présenter comme des extensions des concepts que l'on trouve déjà dans les modèles de procédés. Ces extensions permettent de prendre en compte l'hétérogénéité et le dynamisme des services. Notre approche est indépendante d'un modèle de procédé spécifique. Le Tableau 3.1 présente les classes de trois métamodèles de procédé APEL, BPMN et BEPL qui peuvent être étendues par l'utilisation de notre métamodèle de service abstrait et la réalisation de cette extension [6].

6. Le Tableau est établi à partir de la spécification de BPMN 2.0 accessible à l'adresse http ://www.omg.org/spec/BPMN/2.0/ et de la spécification de BEPL 2.0 accessible à l'adresse http ://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html#_Toc164738480

Concepts	Extension d'APEL	Extension de BPMN	Extension de BEPL
Service Abstrait	Identité entre la classe *Service Asbtrait* de APEL et notre classe Service Abstrait.	Extension de la classe *Service*, qui représente une activité réalisée par un service. Notre classe *Service Abstrait* devient une sous-classe de la classe *Service*.	Extension de la classe *extensionActivity* de BEPL, qui permet d'ajouter à un procédé BEPL des activités que BEPL ne prend pas en charge. Notre classe Service Abstrait est une sous-classe de *extensionActivity*.
Produit	Identité entre la classe *Produit* de APEL et notre classe *Produit*.	Extension de la classe *Data* de BPMN, qui représente les données manipulées par les activités d'un procédé. La classe *Produit* devient une sous-classe de la classe *Data*.	
Message			Extension de la classe *Variable* de BEPL, qui décrit les variables échangées par les services.

TABLE 3.1 – Classes d'APEL, BPEL et BPMN pouvant être étendues par notre métamodèle

Lorsqu'un concepteur conçoit une orchestration en utilisant des services abstraits, il spécifie uniquement l'interface fonctionnelle des services et la technologie qu'il souhaite utiliser dans la mesure où il la connaît le plus souvent. Pour ce faire, il utilise les concepts que nous fournissons pour définir un service abstrait et laisse de côté des détails techniques de services. La recherche du service concret nécessaire à la réalisation de cette fonctionnalité n'est réalisée qu'à l'exécution, ce qui permet de mettre en œuvre la liaison retardée. Si un service utilisé vient à disparaître, l'orchestration peut se poursuivre en recherchant un service qui offre la fonctionnalité nécessaire.

Nous ne proposons pas de langage de représentation graphique des services abstraits. Nous nous appuyons sur le langage graphique que propose chaque langage de spécification de procédé afin de permettre au concepteur de manipuler une syntaxe graphique qu'il connaît déjà et de faciliter l'appropriation et l'utilisation de nos concepts.

2 Métamodèle de contrôle d'accès

Nous nous intéressons aux applications pervasives réalisées sous la forme d'orchestrations de services. Dans une telle application, le contrôle d'accès doit lui-même être pervasif, c'est-à-dire tenir compte du contexte dans lequel l'application s'exécute. Sa mise en œuvre risque donc d'entraîner la collecte d'un grand nombre de données, notamment sur les utilisateurs. Il faut ainsi restreindre le volume des données collectées sur les utilisateurs afin de ne pas mettre en danger leur vie privée.

Afin de répondre à ces deux difficultés, nous proposons un métamodèle qui reprend les concepts du contrôle d'accès basé sur les attributs. Nous les étendons afin de prendre en compte les contraintes du contrôle d'accès contextuel appliqué aux orchestrations de services et de cartographier la visibilité des données des utilisateurs. La structure de notre métamodèle est présentée sur la Figure 3.10.

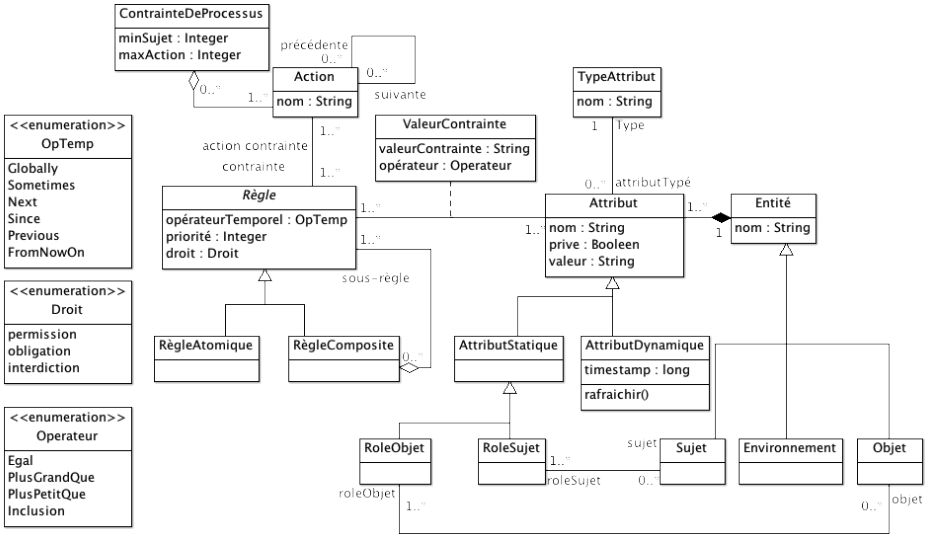

FIGURE 3.10 – Métamodèle de contrôle d'accès

Notre métamodèle fournit à un expert en sécurité les primitives nécessaires à l'expression des droits d'accès dans un procédé. Il repose sur les concepts suivants :

- Le contrôle d'accès s'applique à un ensemble **d'entités** :
  - Un **sujet** désigne un utilisateur humain ou bien un agent logiciel qui agit dans l'application.
  - Un **objet** est une ressource qui peut être utilisée par un sujet dans un procédé.
  - L'**environnement** de l'application est l'ensemble des éléments qui influencent son exécution comme l'heure ou un ensemble de variables globales à l'application.
  - Toutes les entités possèdent des **attributs**. Afin de gérer au mieux leur collecte et leur persistance à l'exécution, il est important de savoir s'ils sont dynamiques ou statiques. Ils sont dynamiques lorsque leur valeur change

souvent et statique sinon. Afin de cartographier la diffusion des données des sujets notamment, les attributs peuvent être privés ou publics. Ils sont publics si leur valeur est directement accessible au sein de l'orchestration, privés sinon. Deux types d'attributs statiques sont particulièrement utiles pour mettre en œuvre le contrôle d'accès :
  - Les **rôles des sujets** permettent de classer les sujets. Ces rôles désignent, par exemple, les fonctions qu'un sujet peut occuper dans une organisation.
  - Les **rôles des objets** permettent de classer les objets d'une organisation. La notion de rôle n'implique ici pas de critère de classement particulier.
- Une **action** est une étape du procédé. C'est un mode d'accès à un objet. Les actions sont enchaînées selon un ordre chronologique.
- Une **contrainte de processus** porte sur un groupe d'actions. Parmi ces contraintes, nous nous attachons notamment à la restriction du nombre de sujets qui peuvent exécuter un groupe d'activités. La séparation des fonctions – *separation of duties* – permet d'obliger la présence de plusieurs sujets pour effectuer plusieurs actions. La fusion des fonctions – *binding of duties* – permet de forcer un sujet à effectuer un groupe d'actions. Les contraintes de processus sont définies par un nombre minimal de sujets qui peuvent exécuter les actions du groupe et le nombre maximal d'actions que chaque sujet peut exécuter dans le groupe.
- Une **règle** permet de contraindre l'accès à une action par un sujet. Une règle associe un droit d'accès à un ensemble de contraintes sur la valeur des attributs des entités de l'application. Ces contraintes définissent les valeurs que doivent prendre les attributs au moyen d'opérateurs comme l'égalité, la différence ou l'inclusion d'une valeur dans un ensemble. Une règle peut être atomique ou composite, c'est-à-dire composée de sous-règles. Ceci permet de réutiliser les règles déjà définies.

Nous ne proposons pas de syntaxe graphique pour notre métamodèle de contrôle d'accès. Dans la mesure où les politiques de contrôle d'accès sont souvent complexes, nous pensons que l'ajout d'éléments graphiques risque de surcharger le modèle de l'orchestration sécurisée et de rendre sa compréhension difficile.

Nous présentons désormais les contraintes que nous imposons lors de l'instanciation de notre métamodèle et la sémantique formelle que nous lui associons. Les contraintes et la sémantique formelle nous permettent de garantir la cohérence des politiques de sécurité rédigées à partir du métamodèle et d'automatiser leur vérification.

2.1 Sémantique formelle du métamodèle de contrôle d'accès Concevoir une politique de sécurité consiste à décrire les règles de contrôle d'accès à l'aide du vocabulaire et de la grammaire que fournit notre métamodèle. Néanmoins, la conception et la modification des règles peuvent entraîner des conflits : lorsqu'une entité possède deux rôles, ils peuvent avoir des droits contradictoires. Afin de gérer ces conflits nous imposons les restrictions suivantes :

- Pour chaque classe du métamodèle, l'attribut nom joue le rôle d'identifiant. Ainsi, à l'instanciation, l'attribut nom de chaque classe prend-il une valeur unique.
- Chaque règle est associée à un niveau de priorité. Lorsque deux règles sont en conflit – un sujet possède deux rôles avec des droits d'accès contradictoires, par exemple – la règle avec le plus haut niveau de priorité est exécutée.
- Lorsque deux règles sont en conflit et que l'une prescrit une autorisation alors que l'autre prescrit une interdiction, l'interdiction est exécutée.

Notre métamodèle permet de représenter l'ordre d'activation des droits des sujets sur les objets d'une application. Afin d'être efficace, une politique de contrôle d'accès doit posséder un ensemble de propriétés comme sa cohérence et la satisfaction d'exigence de sécurité. La cohérence signifie, dans notre cas, que deux règles composées des mêmes attributs ne donnent pas accès à des droits contradictoires. La satisfaction d'exigence de sécurité signifie que l'on doit pouvoir prouver que quelque chose n'est pas permis par la politique de contrôle d'accès. Par exemple, qu'un rôle n'obtient jamais un certain droit d'accès. Nous définissons une sémantique formelle pour notre métamodèle. Ces propriétés peuvent être prouvées lorsque la politique de contrôle d'accès est exprimée de manière formelle. Dans la mesure où les procédés sont un enchaînement chronologique d'actions, nous choisissons d'exprimer la politique de contrôle d'accès sous la forme d'un modèle de logique temporelle.

Du point de vue du contrôle d'accès, une action est un ensemble de contraintes sur les valeurs des attributs des sujets, des objets et des environnements. Ces contraintes sont des relations mathématiques comme l'égalité entre deux valeurs, leur différence ou l'inclusion d'une valeur dans un ensemble. Soit A une action, C une contrainte quelconque sur la valeur d'un attribut et D le droit associé à la satisfaction de la règle. Afin d'accéder à la valeur d'un attribut, nous autorisons la notation pointée : la valeur d'un attribut peut-être obtenue en faisant précéder son nom d'un point et du nom de la source dont il est extrait. La grammaire d'une action Ac est écrite, en forme de Backus-Naur :

- Ac := C(Sujet.attribut)$^+$, C(Environnement.attribut)$^+$,C(Objet.Attribut)$^+$, D

Un procédé est l'enchaînement chronologiquement ordonné d'un ensemble d'actions. Soit P un procédé et Ac une action quelconque. P s'écrit, en forme de Backus-Naur :

- P := Ac$^+$

Enfin, nous exprimons les contraintes sur les flux sous la forme de règles booléennes qui spécifient le nombre maximal d'actions appartenant à un même sous-processus – une suite d'actions dans le processus – qu'un utilisateur peut effectuer. Ces règles sont exprimées sous la forme suivante, avec P, un sous-processus, MinAc le nombre maximal d'actions qu'un utilisateur peut effectuer dans un groupe :

- P $\rightarrow$ MinS $\wedge$MaxAc

Cette condition signifie que lorsque le sous-processus P est exécuté, au moins MinS utilisateurs peuvent exécuter au plus MaxAc actions.

À l'exécution d'un procédé, les actions sont effectuées dans un ordre chronologique. Chaque action peut être suivie par un nombre indéterminé d'actions, ce qui engendre des possibilités d'alternatives dans le procédé. Il est ainsi possible de représenter, par exemple, un scenario nominal et des cas d'erreurs. Un procédé peut ainsi être représenté sous la forme d'un modèle de logique temporelle arborescente (notée CTL pour *Computational Tree Logic*) [Emerson90]. CTL s'appuie sur une représentation arborescente du temps dans laquelle chaque moment peut mener à un nombre indéterminé de moments suivants. Cette représentation est ainsi bien adaptée à la structure arborescente d'un procédé.

Il est nécessaire, lorsque l'on contraint les actions par des règles de contrôle d'accès de spécifier si les règles s'appliquent à une seule branche de l'arbre CTL ou à toutes les branches. On utilise pour cela deux quantificateurs. Le quantificateur *All* – noté A – s'applique à une clause logique qui s'applique à toutes les branches. Le quantification *Exists* – noté E – spécifie qu'il existe au moins une branche de l'arbre dans laquelle une clause est vérifiée. La signification des opérateurs est représentée sur la Figure 3.11.

CTL repose sur quatre opérateurs temporels :

- *Globally* φ –noté G φ signifie que la formule φ doit être satisfaite pour tous les moments qui suivent le moment courant.
- *Sometimes* φ – noté $\Diamond\ \varphi$ signifie que la formule φ doit être satisfaite à un des moments qui suivent le moment courant.
- φ *Until*ψ– noté $\varphi\ U\psi$ signifie que la formule φ doit être satisfaite jusqu'à ce que la formule ψ soit satisfaite.
- *Next* φ – noté X φ signifie que la formule φ doit être satisfaite au moment qui suit le moment courant.

Nous ajoutons trois opérateurs issus de la logique temporelle arborescente du passé (PCTL pour *Past CTL*[9] :

- *Since* φ – noté S φ signifie que la formule φ doit être satisfaite depuis un moment donné.
  Previous φ – noté X-1 φ signifie que la formule φ doit être satisfaite au moment qui précède le moment courant.
- *From Now On* φ – noté N φ signifie que la formule φ doit être à tous les moments qui suivent un moment donné.

Nous identifions trois modalités des droits : la permission, l'obligation et l'interdiction. Chaque modalité peut être représentée à l'aide des opérateurs de la logique déontique :

- La permission est notée Pφ, l'interdiction $\neg$Pφ.
- L'obligation est notée Oφ

La syntaxe du langage logique d'expression des politiques de contrôle d'accès que nous proposons est donnée par la grammaire suivante où φ et ψ sont deux formules logiques, dans notre cas deux politiques de contrôle d'accès :

$\varphi, \psi := \neg\varphi|\mathrm{p}|\varphi \wedge \psi|(\mathrm{A}|\mathrm{E})\varphi|(\mathrm{A}|\mathrm{E})\Diamond\varphi|(\mathrm{A}|\mathrm{E})\ \varphi|(\mathrm{A}|\mathrm{E})\varphi\mathrm{u}\psi|\mathrm{S}\varphi|\mathrm{X}{-}1\varphi|\mathrm{N}\varphi|\ \mathrm{P}\varphi|\mathrm{O}\varphi$

Une politique de contrôle d'accès est constituée de l'ensemble des règles qui s'appliquent à un procédé. Leur enchaînement peut être représenté sous la forme

d'une structure de Kripke [Afrati 2003], un graphes orienté acyclique. La Figure 3.11 illustre une représentation graphique d'une structure de Kripke dans le cas de quatre politiques de sécurité composées à partir des opérateurs de PCTL. Les moments représentés en vert satisfont la politique φ. Le moments en violet, lui, satisfait la politique ψ.

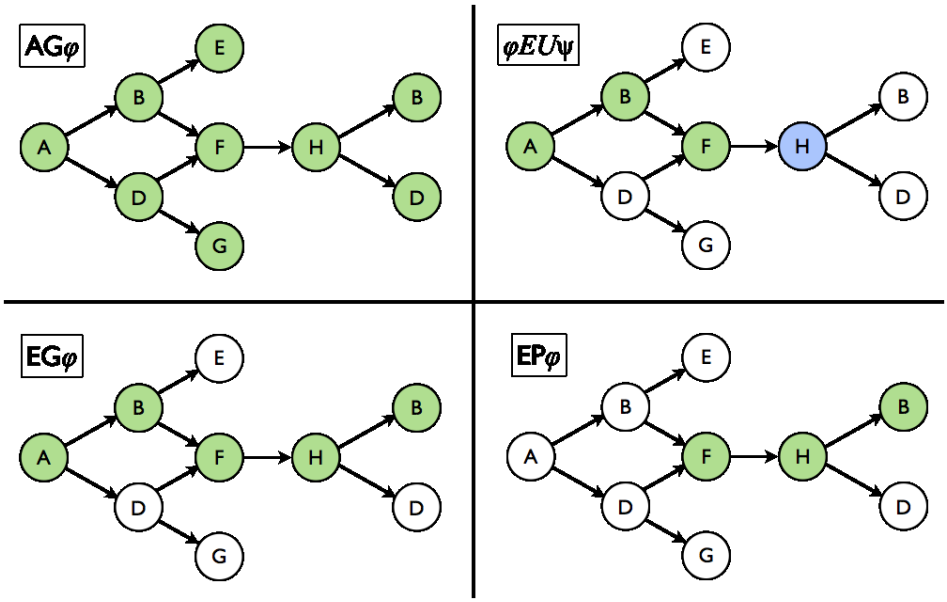

FIGURE 3.11 – Exemple de structures de Kripke et d'expression de contraintes CTL

Les modèles de politique de contrôle d'accès ont deux usages :

- À la conception, ils permettent de prouver la cohérence et la satisfaction d'exigences de sécurité de la politique de contrôle d'accès.
- À l'exécution, ils permettent de déterminer les droits d'accès d'un utilisateur.

Ces deux opérations sont réalisées sous la forme d'une vérification automatique de la satisfaction du modèle de la politique de contrôle d'accès soit par un ensemble de clauses que l'on ne veut jamais voir permises, soit par la description de l'état du procédé, de ses utilisateurs et de ses ressources.

La vérification du respect de la politique de contrôle d'accès est effectuée grâce à la structure de Kripke. Nous définissons une relation de satisfaction entre les politiques φ et ψ et un flux d'actions C. Nous notons cette relation de satisfaction $|=$. Cette relation précise à quelles conditions C satisfait φ et/ou ψ. Nous définissons la relation de satisfaction $|=$ par induction de la manière suivante :

$$
\begin{aligned}
(C,i) \models &\quad p \text{ iff } p \in a_i \\
(C,i) \models &\quad \neg\phi \text{ iff } (C,i) \not\models \phi \\
(C,i) \models &\quad \phi \wedge \psi \text{ iff } (C,i) \models \phi \text{ and } (C,i) \models \psi \\
(C,i) \models &\quad \mathbf{E} \bigcirc \phi \text{ iff } (C,i+1) \models \phi \\
(C,i) \models &\quad \mathbf{E}\phi u\psi \text{ iff there exists } k \geq 0 \text{ s.t. } (C,i+k) \models \phi \\
&\qquad \text{and } (C,i+j) \models \psi \text{ for all } k > i \geq 0 \\
(C,i) \models &\quad \mathbf{A}\phi u\psi \text{ iff for all } a_n \text{ there exists } k \geq 0 \text{ s.t. } (C,i+k) \models \phi \\
&\qquad \text{and } (C,i+j) \models \psi \text{ for all } k > i \geq 0 \\
(C,i) \models &\quad X^{-1}\phi \text{ iff } n > 0 \text{ and } (C,i-1) \models \phi \\
(C,i) \models &\quad \phi S \psi \text{ iff there exists } k \geq n \text{ s.t. } (C,k) \models \phi \\
&\qquad \text{and } (C,i) \models \psi \text{ for all } k < i \leq 0 \; (C,i) \models N\phi \text{ iff } a_n \models \phi
\end{aligned}
$$

Cette relation guide l'évaluation d'une politique de sécurité à l'exécution d'une orchestration de services. Une composition peut être exécutée, c'est-à-dire qu'un ensemble d'actions peut être réalisé par des sujets sur un ensemble de ressources dans un contexte donné si et uniquement si C $|=$ φ, ψ.

Nous avons présenté dans ce paragraphe et le suivant notre métamodèle de contrôle d'accès basé sur le contrôle d'accès basé sur les attributs. Notre métamodèle offre le vocabulaire nécessaire à l'expression du contrôle d'accès. La sémantique formelle permet de prouver les propriétés d'une politique et d'en vérifier la satisfaction à l'exécution.

3 Composition des métamodèles

L'utilisation de deux métamodèles permet d'assurer la séparation des préoccupations liées à l'orchestration de services et des préoccupations liées au contrôle d'accès. Chacune est d'intérêt pour un expert spécifique et chaque préoccupation est amenée à évoluer séparément. La politique de sécurité peut ainsi changer lorsqu'une organisation change – par exemple parce que de nouveaux rôles sont créés – sans que les procédés que l'organisation réalise ne changent. Afin d'exécuter une orchestration de services hétérogènes et dynamiques sécurisée à l'aide du contrôle d'accès, il faut néanmoins avoir une vue complète de sa spécification. Pour ce faire, nous composons maintenant nos deux métamodèles.

La composition des métamodèles nécessite de déterminer la nature des associations entre les classes de chaque métamodèle et leurs cardinalités. Dans notre métamodèle de contrôle d'accès, nous définissons une Action comme une étape dans un procédé qui manipule des Objets. Dans le métamodèle de service abstrait, nous définissons un ServiceAsbtrait comme un ensemble de méthodes qui agissent sur des Produits. Par conséquent, une Action est exécutée par un ServiceAbstrait. Cette exécution est contrainte par une règle de contrôle d'accès.

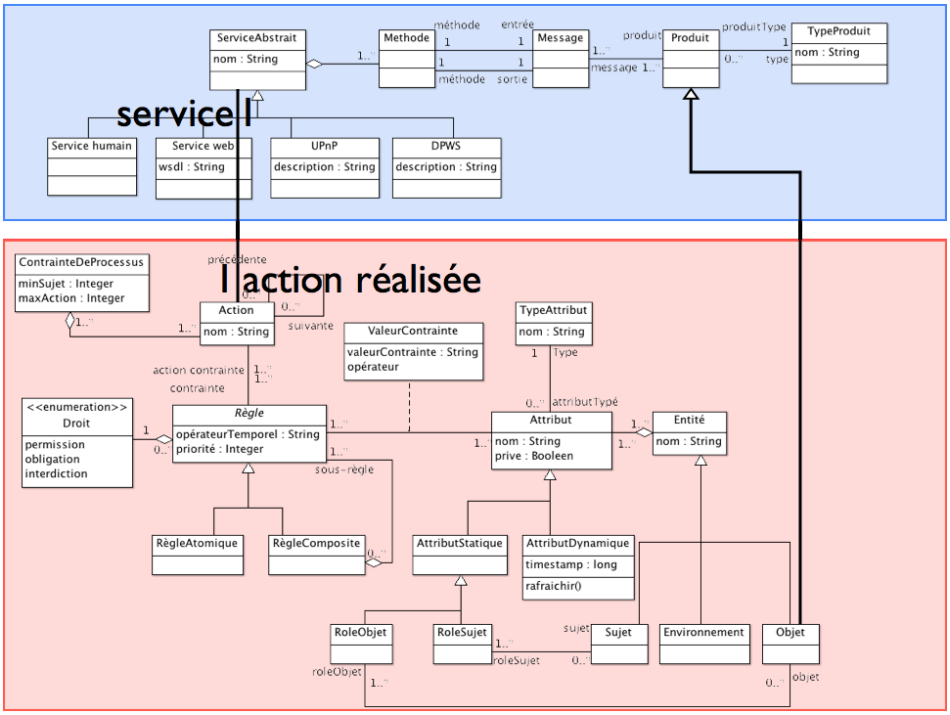

FIGURE 3.12 – Composition des métamodèles

Un Objet est une entité consommée, produite ou modifiée par une Action. Un Objet est donc à une Action ce qu'un Produit est à un Service Abstrait. Un Objet du métamodèle de contrôle d'accès est ainsi un type spécifique de Produit du métamodèle de l'orchestration de services Par conséquent, nous définissons un Objet comme un type de Produit et nous relions ces deux classes par une association d'héritage.

Le résultat de la composition de nos métamodèles est présenté sur la Figure 3.42. Chaque préoccupation est spécifiée par un expert spécifique. Ces spécifications sont conformes au métamodèle concerné. Les préoccupations sont ensuite composées en fonction des associations que nous avons définies entre les classes des deux métamodèles. Chaque service, défini lors de la conception de l'orchestration de services, exécute une action définie lors de la spécification de la politique de contrôle d'accès. L'appel du service est ainsi contraint par les règles d'accès qui s'appliquent à l'action qu'il exécute.

4 Modèles d'une orchestration de services hétérogènes et dynamiques sécurisée à l'aide du contrôle d'accès

Les métamodèles que nous proposons permettent de spécifier une application pervasive à partir de deux types de modèles qui représentent chacun une vue sur l'application :

- **Le modèle de l'orchestration** a pour but de représenter les services abstraits nécessaires à l'exécution de l'application, leur enchaînement et les objets qu'ils manipulent. Il est conforme au métamodèle de l'orchestration que nous avons défini. Le modèle de l'orchestration sert de point de départ à la génération du code d'exécution du procédé et d'appel des services. Le concepteur peut, si nécessaire, préciser la technologie de chaque service abstrait.
- **Le modèle de contrôle d'accès** précise les règles de contrôle d'accès qui s'appliquent aux actions d'un procédé et leurs conditions. Il est conforme à notre métamodèle de contrôle d'accès.

Chaque modèle est conçu par un expert spécifique dans un environnement de modélisation commun. L'expert fonctionnel réalise le modèle de l'orchestration tandis que l'expert en sécurité conçoit le modèle de contrôle d'accès en annotant chaque activité à l'aide de règles de contrôle d'accès. La politique de contrôle d'accès globale est générée à partir des règles qui s'appliquent à chaque activité. On peut alors vérifier automatiquement si elle satisfait certaines propriétés.

L'environnement de modélisation permet à chaque expert de spécifier la vue dont il responsable sous la forme d'un modèle. Les liens entre les modèles sont générés automatiquement à partir des noms des activités. Les actions de la politique de contrôle d'accès ont le même nom que les activités de la composition qu'elles protègent. Dans l'environnement de modélisation que nous proposons, les modèles et leurs liens sont générés et mis à jour dès qu'un des experts modifie la préoccupation dont il est responsable. La Figure 3.13 représente les étapes de la conception et le rôle des métamodèles dans ces étapes.

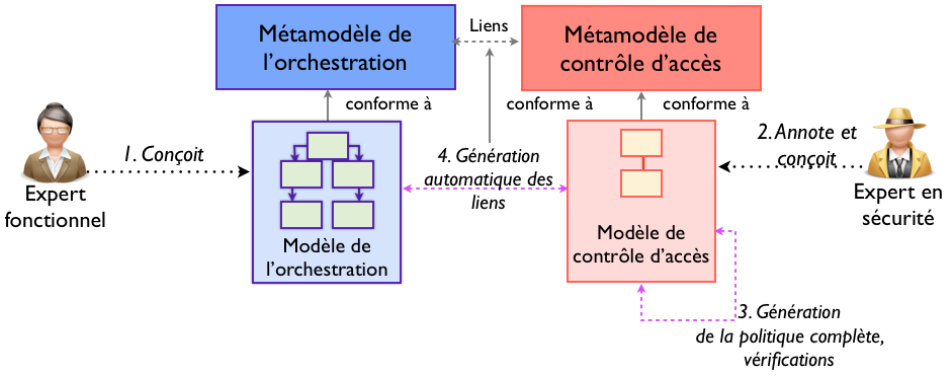

FIGURE 3.13 – Étapes de la conception d'une orchestration de services sécurisée

Afin d'illustrer notre démarche, nous l'appliquons maintenant à une partie d'une application de fouille de données dans un bâtiment intelligent. Nous détaillons les modèles obtenus et leurs liens.

4.1 Modèle de l'orchestration Nous nous intéressons ici à l'extrait d'un procédé de fouille de données dans un bâtiment intelligent. La fouille des données a pour but de repérer des évènements, comme la chute d'un habitant. Dans un bâtiment

intelligent, les services sont hétérogènes – il faut des services web et des appareils exposés comme des services UPnP ou DPWS. Ils sont aussi dynamiques : les capteurs peuvent facilement disparaître, l'utilisateur peut aussi désactiver certains capteurs qui lui semblent trop invasifs.

La fouille de données repose sur trois phases. Tout d'abord, la préparation des données, c'est-à-dire leur récolte. Vient ensuite la recherche de motifs. Enfin, les motifs trouvés peuvent être consultés par des opérateurs chargés de s'assurer que les habitants sont en sécurité. La Figure 3.14 présente une partie du modèle de l'orchestration des services nécessaires à la réalisation de ce procédé. Ce modèle est conforme à notre métamodèle d'orchestration abstraite.

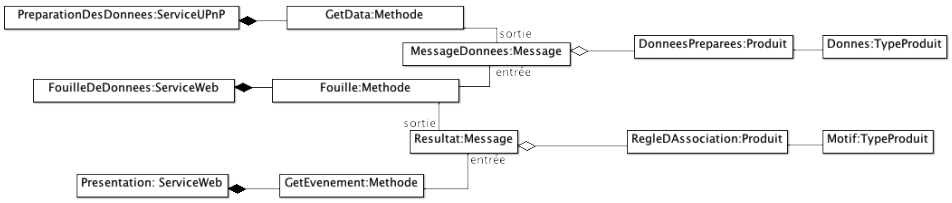

FIGURE 3.14 – Exemple de modèle d'orchestration

Chaque étape du procédé est représentée sous la forme d'un service abstrait. Par conséquent, nous laissons de côté les détails liés à l'implémentation des services et à leur disponibilité effective. Les services concrets ne seront sélectionnés qu'à l'exécution de l'application. Chaque service abstrait possède une méthode qui doit être invoquée afin d'exécuter l'activité nécessaire au bon déroulement du procédé. Les services communiquent au moyen de messages qui leur permettent de s'échanger des données.

Comme nous l'avons précisé, nous utilisons, pour modéliser les orchestrations, la syntaxe graphique des langages de modélisation de procédé qui existent déjà. La Figure 3.15 présente la représentation du procédé de fouille de données selon le formalisme graphique d'APEL.

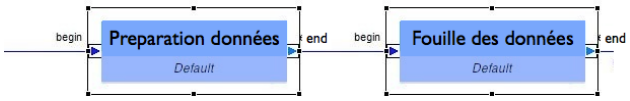

FIGURE 3.15 – Représentation graphique en APEL

Le lien entre les deux activités représente la communication entre elles. Les produits ne sont pas représentés graphiquement.

4.2 Modèle de la politique de sécurité Nous cherchons à contrôler l'accès aux évènements de la maison afin de protéger la vie privée des habitants. Les opérateurs peuvent accéder aux évènements aux conditions suivantes :

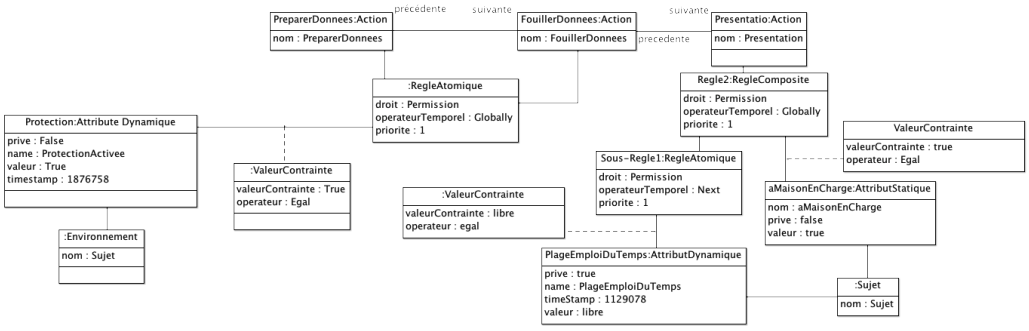

FIGURE 3.16 – Exemple de modèle de contrôle d'accès

- L'opérateur ne peut accéder aux motifs que pendant ses horaires de travail. Cette information est obtenue en interrogeant son emploi du temps qui est un attribut privé.
- L'opérateur ne peut accéder qu'aux évènements des maisons qu'il a en charge.

Dans la maison, les données ne peuvent être collectées et fouillées que si la protection est activée. Nous considérons que l'activation de la protection est stockée sous la forme d'une propriété de l'environnement de l'application.

Dans la mesure où l'emploi du temps d'un individu peut contenir des informations qu'il ne souhaite pas divulguer, nous définissons l'emploi du temps comme un attribut privé. « Privé » a ici le même sens que dans un langage orienté objet comme JAVA : on ne peut pas accéder directement à la valeur de l'attribut. Dans la mesure où l'emploi du temps est susceptible d'être fréquemment mis à jour, c'est un attribut dynamique.

À partir de notre sémantique formelle, la politique de sécurité s'écrit de la manière suivante :

- Soit P le processus de fouille des données et PreparerDonnees, FouillerDonnees et Presentation les actions qui le composent. Alors P = PreparerDonnees, FouillerDonnees, Presentation.
- L'activation de la protection est nécessaire pour exécuter l'orchestration. Elle s'écrit de la manière suivante : PreparerDonnees = FouillerDonnees = G (sujet.hasAttribute(protection, true))
- La règle 2, s'écrit Presentation = sujet.hasRole(operateur) ∧ sujet.hasAttribute(auTravail, true) ∧ sujet.hasAttribute (aMaisonEncharge, true)

En composant les différentes règles, on obtient la politique de contrôle d'accès suivante :

G (sujet.hasAttribute(protection, true)) ∧ sujet.hasRole(operateur) ∧ sujet.hasAttribute(auTravail, true) ∧ sujet.hasAttribute(aMaisonEncharge, true)

La Figure 3.16 donne une représentation graphique de notre politique de sécurité. Cette représentation est conforme au métamodèle que nous avons présenté dans la section précédente.

4.3 Composition des modèles de l'orchestration et de contrôle d'accès

À l'issue de la phase de conception, chaque expert a spécifié la vue de l'application dont il est responsable à l'aide de nos métamodèles. Afin d'obtenir une vue globale de l'application, il est nécessaire de composer les modèles de l'orchestration et de contrôle d'accès. Nous avons fourni des associations entre les métamodèles afin de composer ces vues. Cette composition permet d'associer à chaque action un service qui sera chargé de la réaliser. L'accès au service sera, à l'exécution, contraint par les règles qui s'appliquent à l'action qu'il réalise.

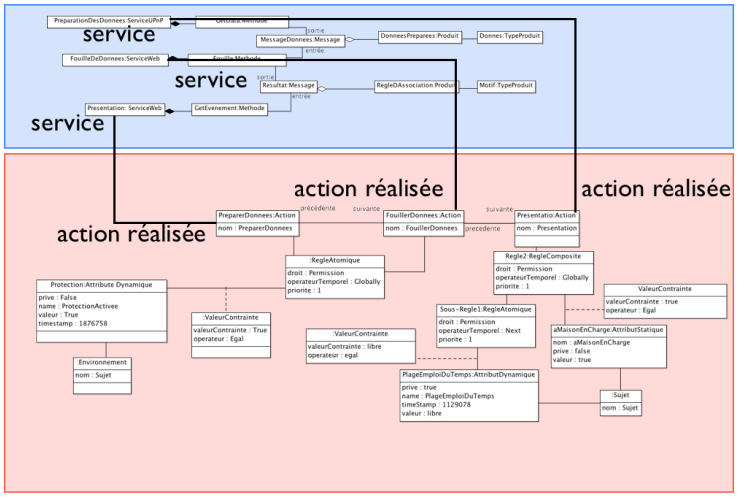

FIGURE 3.17 – Composition du modèle de processus et du modèle de contrôle d'accès

En conformité avec les liens que nous avons établis entre nos métamodèles, les instances des classes Action et Service Abstrait sont automatiquement reliées par un lien. L'action Préparation est réalisée par le service abstrait Préparation-DesDonnées. Il en est de même pour l'action Fouille, réalisée par le service abstrait de FouilleDeDonnées. Les objets qui sont manipulés par les actions sont reliés aux produits utilisés par les services. Ici, les objets Données spécialisent les produits Données. Le résultat de la composition des modèles de l'orchestration et de contrôle d'accès est présenté sur la Figure 3.17.

À l'issue de la phase de conception, nous disposons ainsi d'une vue globale sur une orchestration de services par le contrôle d'accès. Cette vue permet de s'abstraire de l'hétérogénéité des services nécessaires à l'orchestration. Elle précise aussi le niveau de visibilité des données des entités sur lesquels le contrôle d'accès s'applique. Nous présentons désormais l'architecture qui nous permet d'exécuter cette composition. Cette architecture a pour but de configurer les services concrets de manière à ce qu'ils puissent exécuter le contrôle d'accès et de vérifier que la politique de contrôle d'accès est satisfaite tout au long de l'orchestration.

B Gestion de l'hétérogénéité et du dynamisme des services à l'exécution et exécution du contrôle d'accès

Afin d'exécuter les spécifications que nous obtenons à l'issue de la phase de conception, il faut les adapter aux services concrets disponibles et configurer ces derniers. Pour ce faire, nous présentons dans cette section une architecture fondée sur des transformations de modèles qui permet d'exécuter une orchestration de services hétérogènes et dynamiques sécurisée à l'aide du contrôle d'accès.

1 Configuration des services concret : génération des *proxies*

Le principe général de la phase de configuration des services concrets est de générer, pour chacun d'entre eux, un *proxy* réalisé sous la forme d'un service web. Chaque *proxy* protège un service concret en jouant le rôle de point de mise en œuvre de la politique de sécurité. Il intercepte les requêtes adressées au service qu'il protège et contacte le point de décision du contrôle d'accès afin de vérifier que l'invocation du service est autorisée. Le *proxy* n'invoque le service qu'il protège que si tel est le cas.

La génération du *proxy* est réalisée par des transformations de modèles vers texte qui s'appuient sur deux types d'informations :

- **La description du service concret** permet de connaître les propriétés d'un service concret.
- **Des *templates*** décrivent la structure du service web qui implémente le *proxy* et notamment le code chargé d'invoquer le service qu'il protège et de déclencher l'exécution du contrôle d'accès. Ils permettent ainsi, à partir des transformations de modèles, de tisser le code fonctionnel et le code non-fonctionnel dédié à l'exécution du contrôle d'accès. Ils sont adaptés à chaque technologie de service cible mais ils sont génériques car ils ne contiennent pas d'informations sur un service particulier. Les transformations de modèle vers texte sont chargées, à partir des *templates*, de générer le code des *proxies* spécifiques à chaque service concret. Elles utilisent la description des services concrets pour alimenter les templates. L'utilisation des *templates* permet de résoudre l'hétérogénéité des services en s'adaptant aux propriétés de chaque service concret.

Afin de garantir le respect de la politique de contrôle d'accès, il faut configurer les services concrets non sécurisés de manière à assurer qu'il est impossible de les invoquer sans passer par le *proxy* qui les protège. Pour y parvenir, nous étendons le rôle du registre de service. Habituellement, le registre de service a pour fonctionnalités principales l'inscription et la désinscription des services qui peuvent être utilisés dans une composition. Nous modifions l'inscription des services. À chaque fois qu'un nouveau service est ajouté au registre, ce dernier génère un *proxy* chargé de protéger l'accès au service en exécutant la politique de contrôle d'accès. La génération d'un *proxy* à l'inscription du service à partir d'un *template* permet de prendre en compte le dynamisme des services en les sécurisant dès qu'ils rejoignent la composition.

Le procédé de génération du *proxy* est illustré sur la Figure 3.18. Il suit trois

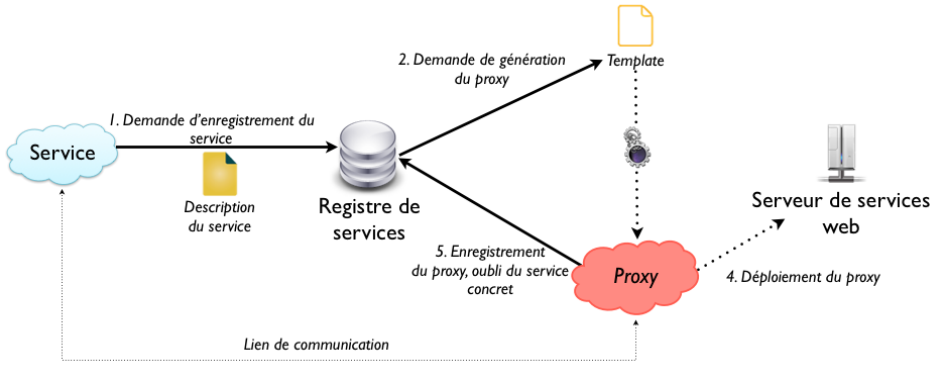

FIGURE 3.18 – Génération du *proxy*

étapes :

1. Un fournisseur de services concrets non sécurisés contacte le registre de services de l'application afin d'y enregistrer un service. Il fournit la description du service qu'il souhaite inscrire dans le registre.
2. Le registre des services recherche ensuite le *template* du *proxy* adapté aux activités que le service peut réaliser et à la technologie du service. Le registre des services alimente le *template du proxy* à l'aide d'informations issues de la description du service concret, comme l'adresse à laquelle il peut être invoqué. Cette étape repose sur les transformations de modèles vers texte que nous avons évoquées. Ces transformations permettent de produire le code d'un service web qui joue le rôle de *proxy* vers le service concret.
3. Le registre des services déploie le *proxy* comme un service web et enregistre uniquement l'adresse du *proxy* sécurisé. Seul ce *proxy* peut contacter le service concret qu'il protège. Il est impossible d'invoquer directement les services concrets qui ne sont pas sécurisés car seul le *proxy* peut être découvert par les clients qui souhaitent l'invoquer lors de l'exécution de l'orchestration.

La phase de génération de code nous permet de résoudre le problème posé par l'hétérogénéité et le dynamisme des services. Nous présentons désormais de manière détaillée le calcul du contrôle d'accès lorsque le *proxy* est invoqué.

2 Calcul du contrôle d'accès

Dans une application pervasive, la politique de contrôle d'accès est susceptible de changer souvent et de nécessiter de collecter de grands volumes de données, souvent dynamiques, notamment sur les utilisateurs. Par conséquent, l'exécution du contrôle d'accès se heurte à trois problèmes majeurs :

- Il faut pouvoir représenter et mettre à jour les connaissances sur l'environnement, les utilisateurs et les ressources de l'orchestration.

- Il faut pouvoir mettre à jour et maintenir la politique de contrôle d'accès sans influencer le déroulement de l'application ou devoir la reconstruire complètement.
- Il faut pouvoir gérer efficacement la collecte des données nécessaires à l'exécution du contrôle d'accès, c'est-à-dire en limitant le nombre de requêtes nécessaires et en ne mettant pas en péril la vie privée des utilisateurs.

Afin de répondre à ces difficultés, notre architecture d'exécution du contrôle d'accès est guidée par deux choix :

- Nous déléguons une partie de l'exécution du contrôle d'accès aux dispositifs des utilisateurs. Ce mécanisme, nommé « contrôle d'accès basé sur les clients », limite la diffusion des données.
- Nous ajoutons à l'orchestration des sources de données qui stockent et maintiennent les données sur l'environnement, les ressources et les utilisateurs de l'orchestration.

2.1 Gestion des données des utilisateurs, des ressources et de l'environnement Le registre des services permet de représenter l'état des services disponibles à un moment donné pour l'orchestration. Afin d'exécuter le contrôle d'accès contextuel, il faut aussi enregistrer et mettre à jour les données qui concernent l'environnement, les ressources et les utilisateurs de l'orchestration. Dans ce but, nous ajoutons à l'architecture d'exécution de l'orchestration deux sources d'informations :

- Le **gestionnaire d'identité** est chargé de sauvegarder et de mettre à jour les informations sur les utilisateurs. Ces informations concernent notamment leur identité et leurs rôles. Lorsque l'attribut d'un utilisateur est privé, le gestionnaire d'identité stocke l'adresse du service à contacter pour obtenir la valeur de cet attribut.
- Le **gestionnaire de contexte** est chargé d'enregistrer et de mettre à jour les informations sur l'environnement dans lequel l'orchestration s'exécute ainsi que sur les ressources de l'application. Tout comme le gestionnaire d'identité, il peut stocker directement ses données ou bien l'adresse des services qui permettent de les obtenir.

Lorsque les attributs stockés par l'un des deux gestionnaires sont dynamiques, les gestionnaires stockent l'heure d'enregistrement de la valeur de l'attribut. Cette valeur n'est ainsi rafraîchie lorsque cela est nécessaire.

2.2 Délégation de l'exécution du contrôle d'accès aux dispositifs des utilisateurs Le contrôle d'accès pervasif repose sur l'utilisation de données sensibles appartenant, notamment, aux utilisateurs. La divulgation de ces données peut être excessive. Supposons par exemple que, pour accéder à un service, un utilisateur doive être disponible. Pour obtenir cette information, on peut interroger son emploi du temps. S'il n'est pas disponible, l'interroger directement permet d'obtenir des informations qui ne sont pas directement pertinentes pour le contrôle d'accès. On peut, par exemple, apprendre que l'utilisateur a un rendez-vous médical. Nous proposons

ainsi de déléguer une partie de l'exécution du contrôle d'accès aux clients. Pour chaque attribut privé du modèle de contrôle d'accès, nous déléguons la vérification de sa contrainte au dispositif de l'utilisateur. Cependant, nous ne contrôlons pas ses clients et il est donc risqué de leur faire confiance. Il faut ainsi s'assurer qu'ils ne peuvent pas mettre en péril l'exécution de la politique d'accès. Pour ce faire, nous proposons de déléguer l'évaluation des contraintes sur les attributs à des services de contrainte, qui sont fiables. La vérification des contraintes s'effectue alors en trois étapes :

- Une requête est adressée à un service de contrainte. Cette requête contient un ou plusieurs attributs – les attributs considérés comme privés dans le modèle de contrôle d'accès – et les valeurs qu'ils doivent prendre.
- Le service de contrainte évalue la requête. Il vérifie si l'attribut a la valeur autorisée. Ce service est fiable. Il possède, par exemple, un certificat qui permet de l'authentifier. Ce service peut être construit spécifiquement ou peut être hébergé sur les dispositifs d'un utilisateur. Ce choix est réaliste, des mécanismes permettent de se fier aux applications déployées sur les outils des utilisateurs. Par exemple, les systèmes d'exploitation de nombreux dispositifs permettent de certifier l'authenticité d'une information.
- Enfin, le service de contrainte transmet le résultat de l'évaluation de la requête.

Ce principe est illustré sur la Figure 3.19. Le contrôle d'accès basé sur les clients permet de certifier qu'une contrainte sur un attribut est satisfaite. Si la contrainte n'est pas satisfaite, la valeur de l'attribut n'est pas divulguée.

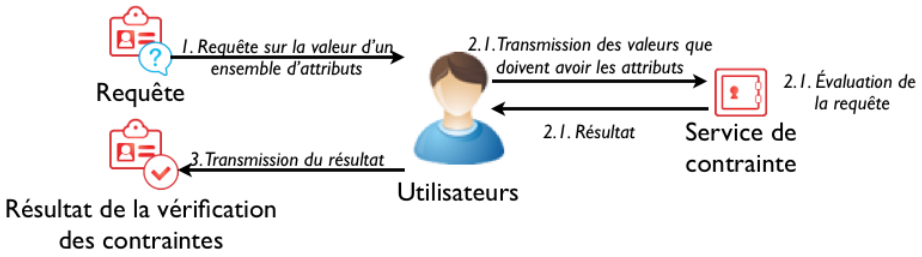

FIGURE 3.19 – Contrôle d'accès basé sur les clients

2.3 Exécution du contrôle d'accès Le contrôle d'accès est exécuté à chaque fois que l'on atteint, lors de l'exécution de l'orchestration, une action associée à une ou plusieurs règles de contrôle d'accès. L'orchestrateur invoque alors le *proxy* sécurisé qui correspond à cette activité. Le *proxy* joue le rôle de point de mise en œuvre du contrôle d'accès. Si le sujet qui veut réaliser l'activité possède les droits requis pour effectuer une activité, le *proxy* invoque le service concret auquel il est associé. Si tel n'est pas le cas, il rejette l'invocation et répond par un message d'erreur.

Afin de savoir si un sujet possède les droits nécessaires à la réalisation d'une activité, le *proxy* contacte le point de décision. Ce dernier est chargé d'évaluer la

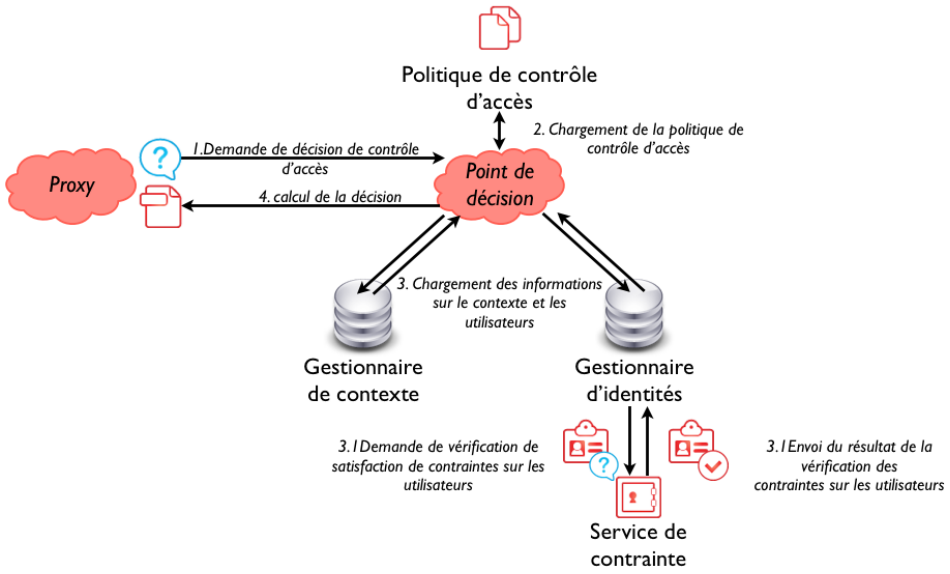

FIGURE 3.20 – Exécution du contrôle d'accès

politique de sécurité en fonction de l'état courant de l'orchestration et de son environnement. Pour ce faire, il contacte les gestionnaires d'identité et de contexte afin de récupérer les informations qui lui sont nécessaires. Il évalue ensuite les règles qui s'appliquent à l'activité considérée en fonction de ces données. Il transmet ensuite la décision de contrôle d'accès au *proxy*. La politique de contrôle d'accès peut changer sans qu'il y ait besoin de modifier toute l'application. Seul le point de décision a accès à la base de la politique de contrôle d'accès. Cette base peut être mise à jour sans influencer le reste de l'orchestration. La Figure 3.20 reprend le fonctionnement général de l'exécution du contrôle d'accès.

C Synthèse

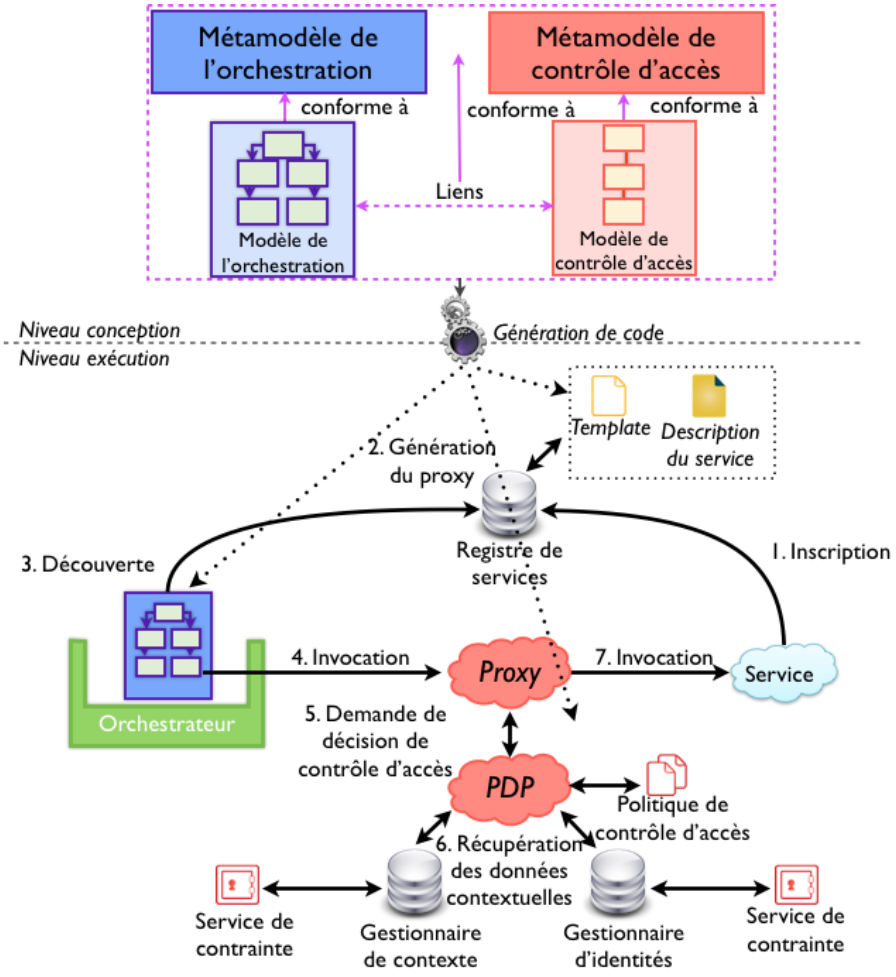

FIGURE 3.21 – Structure de notre proposition

La structure de notre proposition est illustrée sur la Figure 3.21. Notre but était de sécuriser une orchestration de services hétérogènes et dynamiques. Pour ce faire, nous avons proposé une démarche dirigée par les modèles en deux étapes :

- **Le niveau conception** permet de spécifier l'orchestration du point de vue de l'enchaînement des services et du point de vue du contrôle d'accès. Cette phase repose sur des métamodèles qui permettent de s'abstraire des détails techniques et de prendre en compte l'hétérogénéité des services. Ces métamodèles sont composés afin d'obtenir une vue globale de l'application.
- **Le niveau exécution** permet de s'adapter aux services disponibles afin de prendre en compte leur dynamisme. Il permet d'exécuter le contrôle d'accès. En s'appuyant sur le contrôle d'accès basé sur les clients, la diffusion des informations privées est limitée.

Nous avons automatisé la majeure partie des étapes de la conception et de l'exécution de de notre proposition. Le Tableau 3.2 présente ces étapes et leurs responsables.

	Tâche humaine	**Tâche automatisée**
Conception de l'orchestration	Expert fonctionnel	
Conception de la politique de sécurité	Expert en sécurité	
Génération des liens entre les modèles		Environnement de modélisation
Inscription d'un service	Fournisseur de services	
Génération d'un *proxy*		Extension du registre de services
Sélection d'un service		Registre de services

TABLE 3.2 – Réparition des tâches humaines et automatisées

Nous présentons désormais les réalisations auxquelles notre proposition ont donné lieu et leurs évaluations.

III Implémentation de la démarche et validation

Nous avons introduit dans le chapitre précédent notre démarche dirigée par les modèles de conception et d'exécution d'une composition de services hétérogènes et dynamiques sécurisée par le contrôle d'accès. Nous implémentons notre démarche sous la forme d'un système de conception et d'exécution du contrôle d'accès qui permet de protéger une préoccupation spécifique, la vie privée. .

Nous présentons tout d'abord les gestionnaires d'informations liées au contrôle d'accès et leur fonctionnement. Nous expliquons le fonctionnement de notre point de décision et proposons un mécanisme de calcul du contrôle d'accès.

Nous proposons ensuite un ensemble de transformations de modèle vers texte qui permet de gérer le dynamisme et l'hétérogénéité des services disponibles pour la composition et de les sécuriser. Cet outil permet de configurer les propriétés de contrôle d'accès des services concrets.

Nous montrons ensuite comment nous intégrons cet outil à une architecture orientée services. Nous détaillons les extensions que nous apportons au registre de services qui permettent de sécuriser des services hétérogènes et dynamiques.

Nous présentons ensuite notre outil de modélisation qui repose sur l'extension d'un outil de modélisation de processus auquel nous ajoutons des primitives pour exprimer le contrôle d'accès. Cet outil permet de générer les modèles de l'orchestration et de la politique de contrôle d'accès qui lui est associée

Enfin, nous appliquons notre démarche au cas d'un habitat intelligent afin de démontrer la faisabilité et l'intérêt de notre approche. Nous présentons, dans la dernière partie de ce chapitre, les résultats obtenus et la validation de notre approche.

A Composants dédiés à la gestion de la connaissance et à l'exécution du contrôle d'accès

Afin de vérifier le respect de la politique de contrôle d'accès au cours de l'exécution de la composition, il est nécessaire de posséder un point de décision mais aussi des composants dédiés à la gestion des informations nécessaires à l'évaluation du contrôle d'accès. Ces informations portent sur l'environnement dans lequel l'orchestration s'exécute ainsi que les utilisateurs qui participent à son exécution. Nous présentons ici le point de décision et les sources d'informations que nous avons implantées sous la forme de services web.

1 Sources d'informations

Notre architecture repose sur deux sources d'informations, le gestionnaire d'identité et le gestionnaire de contexte, qui s'appuient chacune sur une base de données. Ces sources stockent et maintiennent les informations nécessaires pour exécuter le contrôle d'accès. Pour ce faire, elles ont deux rôles :

- Elles permettent au concepteur d'une application de définir le schéma des bases de données qu'elles utilisent. Pour ce faire, le concepteur modélise les entités d'intérêts pour le contrôle d'accès (les utilisateurs, l'environnement et les ressources) et leurs attributs. Elles permettent aussi de préciser si ces derniers sont privés ou dynamiques. Elles permettent aussi d'instancier les entités en ajoutant, par exemple, un nouvel utilisateur et la valeur de ses attributs.
- Elles permettent, à l'exécution, d'accéder à la valeur des attributs qu'elles stockent.

Ces deux sources sont réalisées sous la forme de services web qui permettent de créer et mettre à jour des bases de données XML. Ces bases contiennent la description des attributs des utilisateurs et des ressources de l'application. Parmi les informations que ces données contiennent, certaines sont susceptibles de ne pas changer – tel est le cas de l'état civil d'un utilisateur. Certaines, comme la disponibilité d'une ressource, sont susceptibles de changer fréquemment et de devoir être souvent mises à jour. Il n'est donc pas efficace de les stocker directement dans les sources d'information. Enfin, la révélation de certaines informations, comme la localisation d'un utilisateur, peuvent empiéter sur sa vie privée. Ainsi, si on veut contraindre l'accès d'un utilisateur à une activité en fonction de sa présence sur son lieu de travail, il est nécessaire de savoir s'il s'y trouve mais, si tel n'est pas le cas, pas de savoir où il est. Au travers d'une interface, le concepteur de l'application peut définir le schéma du fichier XML qui contient les informations stockées par les sources d'informations. Ce schéma décrit les entités manipulées par la source d'information, leurs attributs et leurs propriétés. Il stocke, pour chaque attribut les informations suivantes :

- L'entité à laquelle appartient l'attribut.
- Le type de l'attribut.
- Si l'attribut est privé, le schéma précise qu'il faut stocker l'adresse du service

de contrainte qui permet de vérifier que la valeur de l'attribut satisfait un ensemble de contraintes.

- Si l'attribut est dynamique :
  - Si l'attribut est public, le schéma précise qu'il faut stocker la date et l'heure à laquelle la valeur de l'attribut a été enregistrée et la durée de validité de cette valeur.
  - Si l'attribut est privé, le schéma précise qu'il faut stocker l'identifiant de la contrainte testée sur l'attribut, le résultat de l'évaluation et la date à laquelle la contrainte a été évaluée.

Le code ci-dessous est présente un extrait du schéma d'un gestionnaire d'identité qui stocke des utilisateurs, leurs rôles et leur localisation. La localisation est un attribut privé et dynamique alors que le rôle est public et statique :

```
<xs:element name="contextManager">
  <xs:complexType>
    <xs:sequence>
    <xs:element name="user"/>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="localisation">
            <xsd:appinfo>
              <foo:private value="true"/>
              <foo:dynamic value="true"/>
            </xsd:appinfo>
            <xs:complexType>
              <xs:sequence>
                <xs:element name="serviceContrainte"/>
                <xs:element name="requete"/>
                <xs:element name="resultat"/>
                <xs:element name="timesStamp"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="role">
            <xsd:appinfo>
              <foo:private value="true"/>
              <foo:dynamic value="true"/>
            </xsd:appinfo>
          </xs:element>
          </xs:sequence>
          <xs:attribute name="name"/>
        </xs:complexType>
      </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

Exemple de code 3.1 – Exemple de schéma du gestionnaire d'identités

Nos gestionnaires d'informations sont légers afin de pouvoir être remplacés par des solutions propriétaires qu'un concepteur d'application a l'habitude d'utiliser. Le gestionnaire d'identité, par exemple, peut être remplacé par des gestionnaires d'identité disponibles dans le commerce, sous la forme notamment d'implémentations d'annuaires *Lightweight Directory Access Protocol* (LDAP).

1.1 Réalisation des services de contraintes Les services de contraintes ne divulguent pas directement la valeur d'un attribut mais indiquent si une contrainte sur cet attribut est satisfaite. Dans le cas de la disponibilité d'une ressource, la ressource elle-même peut jouer le rôle de service de contrainte. Lorsque l'on souhaite

accéder à un attribut privé, le gestionnaire d'information adéquat interroge le service de contrainte. Le service de contrainte doit être authentifié afin d'assurer la fiabilité des données obtenues. Lorsqu'il est interrogé, il n'envoie pas directement la valeur de la donnée mais répond à la question de contrôle d'accès : il indique seulement si les contraintes sur les valeurs des données de l'utilisateur sont satisfaites. Le principe de l'utilisation d'un service de contrainte est illustré sur la Figure 3.22.

Nous avons développé les services de contraintes comme des applications web réalisées en Javascript et HTML 5. Nous choisissons ces technologies car elles sont utilisables par la plupart des appareils disponibles comme les ordinateurs, les *smartphones* et les tablettes et permettent d'utiliser les capteurs de ces appareils comme le GPS. Il est ainsi possible d'utiliser les appareils des utilisateurs afin de récolter des informations, comme leur localisation.

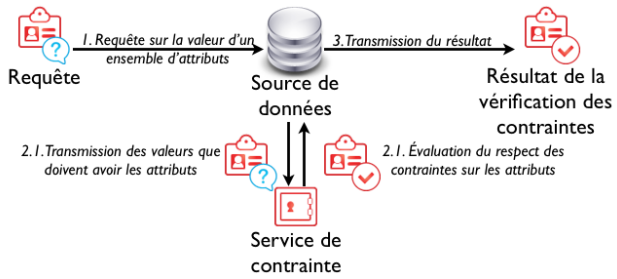

FIGURE 3.22 – Service de contrainte

À chaque fois qu'une source de données souhaite vérifier si la valeur d'un attribut satisfait une contrainte, elle adresse au service de contrainte une requête contenant cette contrainte. Cette requête est construite à partir de la contrainte contenue dans la politique de contrôle d'accès. Le service de contrainte évalue cette contrainte et répond à la source de données. La communication entre la source de données et le service de contrainte repose sur HTTPS, un protocole de communication sécurisé qui permet d'authentifier un serveur et un client qui communiquent au travers d'un réseau informatique et qui permet de chiffrer cette communication. L'utilisateur doit posséder un certificat installé sur son appareil afin de pouvoir utiliser le service de contrainte.

Le code suivant présente une requête sur la localisation de Bob adressée au service de contrainte :

```
<requete>
  <attributeName>localisation</attributeName>
  <contrainte>
    <type>egalite</type>
    <valeur>atHome</valeur>
  </contrainte>
</requete>
```

Exemple de code 3.2 – Requête sur la localisation d'un utilisateur

Cette requête est évaluée en utilisant notamment les informations tirées de l'API Geolocation HTML 5 qui permet d'obtenir la localisation d'un utilisateur. Le

service de contrainte vérifie si la contrainte est satisfaite et répond sous la forme d'un flux XML. Le code ci-dessous présente la réponse obtenue à la requête sur la localisation de Bob.

```
<reponseRequete>
  <attributeName>localisation</attributeName>
  <contrainte>
    <satisfaite>true</satisfaite>
    <timestamp>154789</timestamp>
  </contrainte>
</reponseRequete>
```

Exemple de code 3.3 – Réponse à la requête sur la localisation d'un utilisateur

2 Point de décision

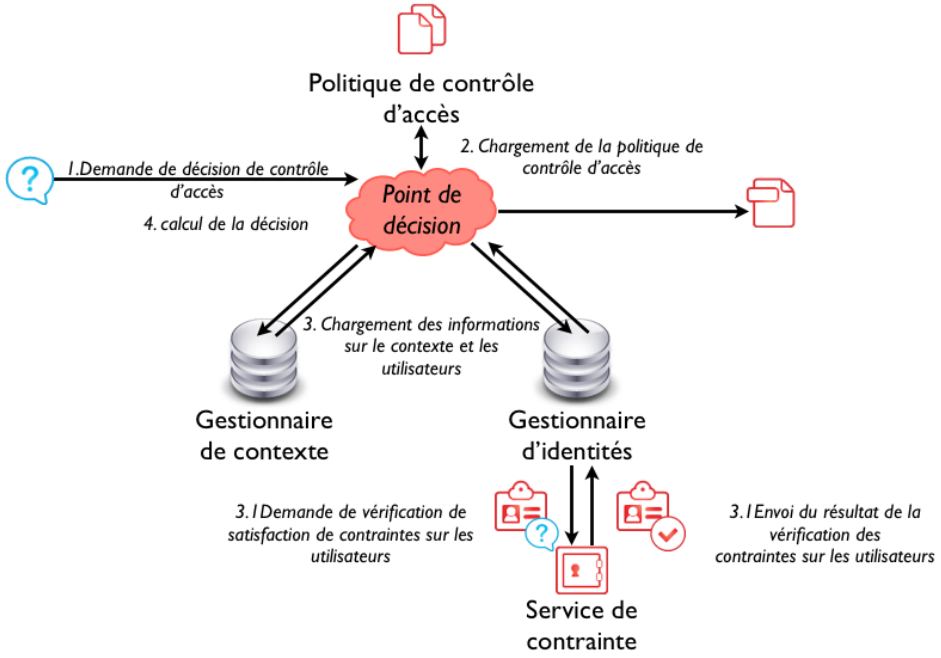

FIGURE 3.23 – Fonctionnement du point de décision et des sources de données

Le point de décision est chargé de vérifier que la politique de contrôle d'accès est satisfaite à un moment donné en fonction d'un utilisateur souhaitant accéder à une activité et des valeurs des propriétés de l'utilisateur et de l'environnement d'exécution de l'orchestration. Pour ce faire, le point de décision interroge le gestionnaire de contexte et le gestionnaire d'identité qui peuvent, eux, interroger des services de contraintes. À partir de leurs réponses, le point de décision évalue la satisfaction de la politique de contrôle d'accès. La Figure 3.23 reprend le fonctionnement général du point de décision et des sources de données.

Nous avons implémenté le point de décision sous la forme d'un service web. L'évaluation de la politique de contrôle d'accès repose sur le moteur *Drools*, un

système de gestion de règles et d'inférences[7]. *Drools* repose sur la logique du premier ordre pour définir des contraintes sur des objets JAVA afin d'en manipuler les attributs. Dans notre, cas, nous utilisons une classe Java pour représenter les utilisateurs et une classe Java pour représenter le contexte. Nous exprimons les règles de contrôle d'accès comme des contraintes sur la valeur des objets qui représentent les utilisateurs et le contexte.

Drools permet le raisonnement temporel à l'aide d'opérateurs temporels qui portent sur des évènements. Le Tableau 3.3 présente la correspondance entre ces opérateurs et ceux de notre métamodèle. Dans le Tableau 3.3, D désigne une date données et E et F des évènements.

Métamodèle	Drools	Définition
Globally E	E Coincides D	E doit se produire dès D
Sometimes E	Finished By D	E doit se terminer avant D
E Next F	F After E	F se produire après E
E Since F	F Starts E	E produire avant F
E Previous F	E Before F	E doit se produire avant F
E From now on D	E Coincides D	E doit être vérifié à partir de D

TABLE 3.3 – Correspondance entre Drools et les opérateurs de notre métamodèle de contrôle d'accès

Soit la règle suivante, qui s'applique à une activité : un utilisateur ne peut effectuer l'activité que s'il est médecin et s'il débute l'activité au moins 30 minutes avant la fin de son service. Le code ci-dessous présente l'écriture de cette règle au format Drools :

```
1 rule "isEqualTo"
2   when
3     a : Action( name == "isEqualTo")
4     u : User(hasAttribute("role","meteorologue"), hasAttribute("
          localisation","hopital")) ;
5   then
6   u.setAutorise(true);
7 end
```

Exemple de code 3.4 – Exemple de règle Drools

Afin d'exécuter cette règle, il faut fournir au moteur d'exécution de Drools l'objet qui représente l'utilisateur dont on souhaite évaluer les droits. La variable $user pointe sur cet objet uniquement s'il possède le rôle de médecin. On peut alors vérifier que l'accès est autorisé 30 minutes avant que l'utilisateur ne finisse son service. Si tel est le cas on accorde le droit d'accès à l'activité à l'utilisateur et on mémorise la date et l'heure auxquelles ce droit est accordé.

7. http ://www.jboss.org/drools/

B Génération de *proxies* pour gérer l'hétérogénéité et le dynamisme des services et les sécuriser

1 Principe

À l'issue de la phase de conception de la composition, nous disposons des modèles de l'orchestration ainsi que de la politique de contrôle d'accès. Le modèle de l'orchestration est abstrait car il ne contient pas d'informations sur les services à appeler pour exécuter la composition. À l'exécution, il faut ainsi adapter l'orchestration aux services concrets disponibles et sécuriser ces derniers afin de garantir l'exécution de la politique d'accès.

Pour y parvenir, nous nous appuyons sur des transformations de modèle vers texte pour générer du code. Ces transformations ont deux buts :

- Elles produisent le code fonctionnel nécessaire pour invoquer les services concrets. Cette étape repose sur les informations spécifiques à chaque service qui sont contenues, par exemple, dans leur description. Elle permet d'invoquer un service particulier en fonction de son adresse et de sa technologie notamment.
- Elles insèrent dans le code fonctionnel un appel à la vérification de la politique de contrôle d'accès ainsi que le code nécessaire à la mise en œuvre de la décision de contrôle d'accès.

La génération de code produit un *proxy* sécurisé chargé d'appeler le point de décision de contrôle d'accès et de jouer le rôle de point de mise en œuvre du contrôle d'accès en invoquant le service concret qu'il protège ou en rejetant l'invocation. Ce *proxy* est lui-même un service web. Le procédé de génération de code produit aussi le code nécessaire à son déploiement et à son invocation.

2 Réalisation : utilisation de JET et déploiement du *proxy* sur un serveur Apache Axis

Afin de mettre en œuvre notre procédé de génération de code, nous nous appuyons sur des templates *Java Emitter Templates* (JET) [8]. Un template JET est un fichier qui contient du code informatique sous la forme d'un « texte à trous » : il décrit la structure générale du code que l'on souhaite obtenir ainsi que les endroits du code où des informations doivent être fournies afin de générer du code exécutable. Le code obtenu peut être exprimé dans n'importe quel langage informatique.

Nous décrivons avec un template JET la structure d'un *proxy* chargé d'invoquer un service concret, d'appeler le point de décision de contrôle d'accès et de la mettre en œuvre en invoquant le service concret qu'il protège ou en refusant cette invocation. Dans la mesure où chaque technologie de service fournit des mécanismes spécifiques pour l'invocation, nous proposons un template par technologie.

Peu importe la technologie du service concret que l'on souhaite protéger, nos templates produisent un *proxy* réalisé sous la forme de code Java. Tous les *proxy*s peuvent être intégrés de manière transparente à l'orchestration de services en étant

8. Notre utilisation des templates JET pour générer des proxies par les services est inspirée par [Chollet 2009a] et [Chollet 2009b]

exposés comme des services web. Nous choisissons de les déployer à l'aide du framework Apache Axis [9] – un serveur SOAP qui permet d'exposer du code Java comme un service web. Il nous faut ainsi générer non seulement le code java du *proxy*, mais aussi les fichiers pour le déployer comme un service web. Nous proposons un template JET pour le code du *proxy* et les quatre fichiers nécessaires à son déploiement sous la forme d'un service web sur Axis. La Figure 3.24 présente le déroulement de la génération de code à l'aide de templates JET et leur enchaînement.

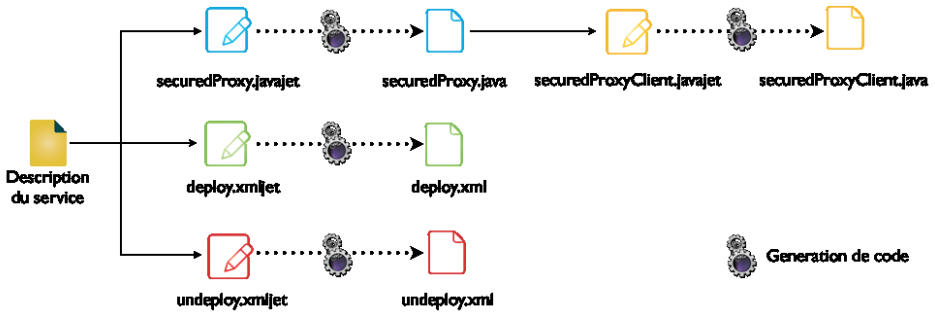

FIGURE 3.24 – Générations de code à l'aide de templates JET

Le template principal est le template de génération du code Java du *proxy* sécurisé nommé *securedProxy.javajet*. Il produit une classe Java qui est le code du *proxy*. Le code ci-dessous présente une partie du code de ce template consacrée à la protection d'une méthode d'un service.

```
1  public <%= methode.getReturnType().getName()%> secured<%=methode
       .getName()%>(java.lang.String userName, java.lang.String
       sessionID, <%for (Class c: parametresMethode) {numParam
       ++;%><%=c.getName()%> <%=nomParam+numParam%><%if (numParam<(
       parametresMethode.length)){%><%=", "%><%}%><%}%>) {
2    <%=serviceInterfaceName%> portToService = new <%=
       serviceLocator%>();
3    //Creation du decision point depuis le endpoint
4    String right="DENIED";
5    DecisionPointService dp = new DecisionPointServiceLocator();
6    try {
7      DecisionPoint portToDecisionPoint = dp.getDecisionPoint();
8      try {//Test : l'utilisateur a-t-il le droit d'acceder a l'
         activite ?
9        right = portToDecisionPoint.makeDecision(userName,
           activity, sessionID);
10     } catch (RemoteException e) {
11       // TODO Auto-generated catch block
12       e.printStackTrace();
13     }
14     if (!right.equalsIgnoreCase("DENIED")) { //Si l'utilisateur
         a les droits necessaires, on appelle methode
15       //Recuperation du Stub qui implemente le Service
           Definition Type
16       <%=serviceName%> port = portToService.<%=
           getServicePortMethod%>();
17       try {
18         <%if (!methode.getReturnType().getName().equals("void"))
             {%>return<%}%> port.<%=methode.getName()%>(<%
             numParam=0; for (Class c: parametresMethode) {
             numParam++;%><%=nomParam+numParam%><%if (numParam<(
             parametresMethode.length)){%><%=", "%><%}%><%}%>);
20 ------------------------------------------------------------
```

9. http ://axis.apache.org/axis/

```
19          } catch (RemoteException e) {
21            // TODO Auto-generated catch block
22            e.printStackTrace();
23          }
24        }
25      } catch (ServiceException e) {
26        e.printStackTrace();
27      }
28      <%if (!methode.getReturnType().getName().equals("void")){%>
            return null;<%}%>
29    }
30  <%
31    }
32  %>
33  }
```

Exemple de code 3.5 – securedProxy.javajet

Le template est construit en deux parties :

- Le début du template permet de récupérer les informations liées au service à protéger : son nom – l'attribut *serviceName* – son adresse – l'attribut *serviceLocator* – et un pointeur vers le port du service pour invoquer ses méthodes – l'attribut *getServicePortMethod*.
- Le reste du template est dédié à la génération du code du *proxy*. Pour chaque méthode du service à protéger, le template génère une méthode pour le *proxy*. Ces méthodes sont chargées de sécuriser l'appel aux méthodes du service concret que le *proxy* protège. Pour ce faire, la méthode appelle le point de décision du contrôle d'accès. Dans le template, le point de décision est identifié par la variable *portToDecisionPoint*. L'appel est réalisé au moyen de la méthode *makeDecision* dont les arguments sont le nom de l'utilisateur, l'activité courante à laquelle il demande l'accès ainsi que l'identifiant de la session d'exécution de l'orchestration. Si l'utilisateur a le droit d'exécuter la méthode, alors le *proxy* invoque la méthode correspondante du service concret en lui transmettant les paramètres nécessaires. Sinon, le *proxy* renvoie un message d'erreur.

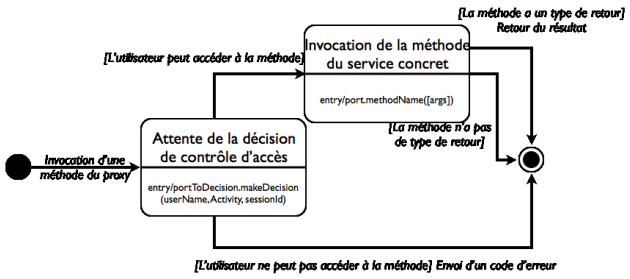

FIGURE 3.25 – Fonctionnement du proxy

Le diagramme d'états-transtions présenté sur la Figure 3.25 illustre le fonctionnement du *proxy* lors de l'invocation de l'une de ses méthodes.

Le code suivant présente un exemple de code Java généré à l'aide de ce template :

```
1  public java.lang.String securedmin(java.lang.String Wsdl, java.
       lang.String sessionID, java.lang.String activity) {
2    long timestampAvantExecution= System.currentTimeMillis();
3    //port to the non secured service
4    OpeService_pkg.OpeServiceService portToService = new
       OpeService_pkg.OpeServiceServiceLocator();
5    //Creation du decision point depuis le endpoint
6    PolicyDecisionPointImplService dp = new
       PolicyDecisionPointImplServiceLocator();
7    try {
8      PolicyDecisionPointImpl portToDecisionPoint = dp.getPDP();
9      String[] lstUsers=null;
10     try {//Test : l'utilisateur a-t-il le droit d'acceder a l'
         activite ?
11       lstUsers = portToDecisionPoint.lesUtilisateursAutorises(
           activity);
12     }
13     } catch (RemoteException e) { e.printStackTrace();}
14     if (lstUsers!=null) { //Si l'utilisateur a les droits
         necessaires, on appelle methode
15       NotificationServiceServiceLocator session = new
           NotificationServiceServiceLocator();
16         session.setMaintainSession(true);
17         NotificationServiceService service = session;
18         try {
19           NotificationService notification = service.
               getNotificationService();
20           String id = notification.sendNotification(activity,
               lstUsers,"min",Wsdl);
21           while(!(notification.getEtatNotification(id).equals("
               Terminer"))){
22           try {
23             Thread.sleep(1000);
24           } catch (InterruptedException e) {e.printStackTrace()
               ;}
25           }
26           String donnee = notification.getInfoNotification(id);
27           String[] param = donnee.split(" :: ");
28         OpeService port = portToService.getOpeService(); //
             Recuperation du Stub qui implemente le Service
             Definition Type
29         long timestampAvantAppel= System.currentTimeMillis();
30         java.lang.String resultat=port.min(param[0]);
31         return resultat;
32
33
34       } catch (RemoteException e) {e.printStackTrace();}
35     }
36   } catch (ServiceException e) {
37     e.printStackTrace();
38   }
39   return null;
40 }
```

Exemple de code 3.6 – securedProxy.javajet

Apache Axis a besoin d'un fichier pour déployer le code Java comme un service web. Ce fichier décrit le nom du service, la classe Java principale du service ainsi que la visibilité des méthodes. Nous fournissons un template qui génère le fichier de déploiement *deploy.xml* à partir du nom du *proxy* et du nom de sa classe Java. Le code ci-dessous présente le template de génération du fichier de déploiement nommé *deploy.xmljet* :

```
1 <%@ jet package="templates" imports="java.util.*" class="
    DeployTemplate" %>
2 <% HashMap<String,String> lesParametres = (HashMap<String,String>)
     argument; %>
3 <deployment xmlns="http://xml.apache.org/axis/wsdd/" xmlns:java="
    http://xml.apache.org/axis/wsdd/providers/java">
4 <!-- Nom du service, invoqueé en RPC -->
```

```
<service name="<%=lesParametres.get("serviceName")%>" style="java:RPC">
  <!-- Classe compilée associée au service -->
  <parameter name="className" value="<%=lesParametres.get("className")%>">
  <!-- toutes les méthodes du service sont atteignables -->
  <parameter name="allowedMethods" value="*"/>
</service>
</deployment>
```

Exemple de code 3.7 – deploy.xmljet

Le code ci-dessous présente un exemple de fichier de déploiement généré à l'aide du template.

```
<deployment xmlns="http://xml.apache.org/axis/wsdd/" xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
<service name="SecuredWidgetPrice" style="java:RPC">
  <!-- Classe compilee associee au service -->
  <parameter name="className" value="fr.SecuredWidgetPrice.SecuredWidgetPrice"/>
  <!-- toutes les methodes du service sont atteignables -->
  <parameter name="allowedMethods" value="*"/>
</service>
</deployment>
```

Exemple de code 3.8 – Exemple de fichier de déploiement

Apache Axis utilise un fichier XML pour désinstaller un service. Ce fichier décrit le nom du service à désinstaller et permet de retirer un service du serveur Apache Axis. Nous fournissons un template qui génère le fichier de désinstallation du *proxy undeploy.xml* à partir du nom de ce dernier. Le code suivant présente le template de génération du fichier de désinstallation nommé *undeploy.xmljet* :

```
<%@ jet package="templates" class="UndeployTemplate" %>
 <undeployment xmlns="http://xml.apache.org/axis/wsdd/">
  <!-- Nom du service -->
  <service name="<%=argument%>"/>
</undeployment>
```

Exemple de code 3.9 – undeploy.xmljet

Le code ci-dessous présente un exemple de fichier de désinstallation généré à l'aide du template :

```
<undeployment xmlns="http://xml.apache.org/axis/wsdd/">
  <service name="OpeService"/>
</undeployment>
```

Exemple de code 3.10 – Exemple de fichier de désinstallation

Enfin, nous fournissons un template qui permet de générer le code d'automatisation de la génération du *proxy*, de ses fichiers de déploiement et de désinstallation et de son client. Ce template produit un ensemble de tâches Ant décrites au format XML [10].

10. http ://ant.apache.org/ Ant est un outil qui permet d'automatiser la construction de logiciels en spécifiant un ensemble de tâches – comme la compilation d'un fichier ou sa copie – et leur enchaînement.

C Extension du registre de services

Le registre de services permet de stocker les informations sur les services disponibles au cours de l'exécution de l'orchestration. Comme nous l'avons précisé dans la présentation globale de notre approche, nous étendons le rôle du registre de services afin de générer les *proxies* sécurisés lors de l'inscription d'un service concret. La Figure 3.26 présente le fonctionnement du registre de service lors de l'enregistrement d'un service.

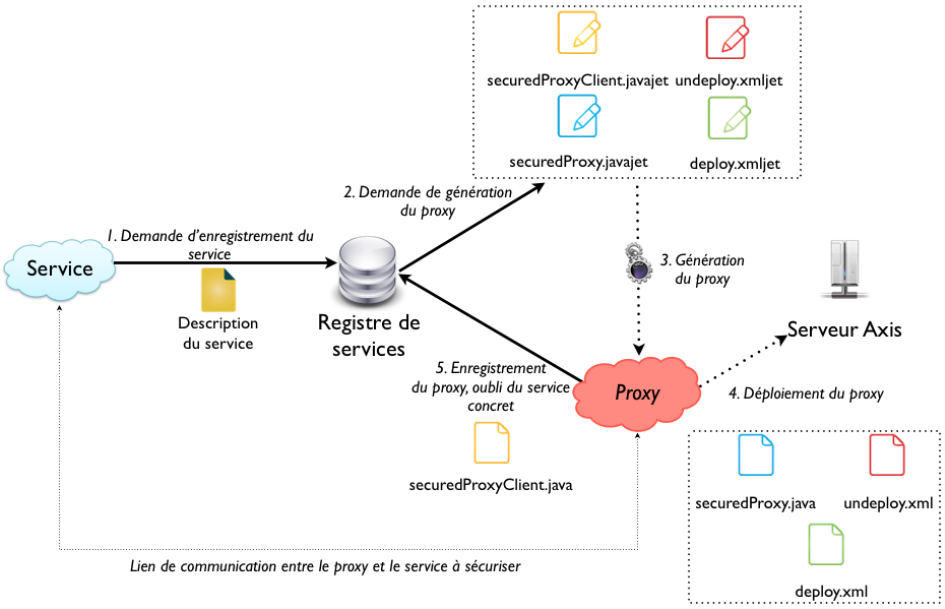

FIGURE 3.26 – Enregistrement d'un service

Le registre de service offre une méthode publique qui permet à un fournisseur de service d'inscrire ses services en fournissant leur description et la description des fonctionnalités qu'ils peuvent réaliser.

À l'inscription d'un service le registre génère un *proxy* qui exécute la politique de contrôle d'accès. La génération et le déploiement du *proxy* sécurisé se fait en quatre étapes :

- Le registre de services alimente les templates nécessaires à la génération du *proxy* sécurisé avec la description du service non sécurisé. Les templates sont choisis en fonction de la technologie du service.
- Le code du *proxy* et les fichiers nécessaires à la gestion de son cycle de vie comme service web sont ensuite générés.
- Le *proxy* est ensuite déployé sur un serveur Axis.
- Enfin, le registre de service ne mémorise que l'adresse du *proxy* sécurisé, c'est-à-dire qu'il déploie le client qui permet d'accéder au *proxy*. Seul le *proxy* connaît l'adresse du service concret qu'il protège. Il est ainsi impossible d'accéder directement au service concret et d'empêcher l'évaluation de la politique

de contrôle d'accès.

Notre extension du registre de services permet ainsi de sécuriser des services hétérogènes et dynamiques qui peuvent faire partie d'une orchestration. À l'exécution de l'orchestration, le registre de service permet aussi de mettre en œuvre la liaison retardée : il permet de sélectionner les services concrets qui peuvent réaliser une fonctionnalité. Chaque service est identifié par un nom abstrait et décrit par son url. Pour chaque méthode du service, on donne un nom abstrait et son nom réel ainsi que celui de ses paramètres. Le code suivant donne un extrait du contenu du registre de service.

```
1 <serviceReg>
2   <service nomAbstrait="Meteo" url="http://lyo.67?wsdl"
3   urlNonSecured="http://bhk.kiu">
4     <methode nomAbstrait="donneeMeteo" nomReel="securedGetWeather
          ">
5       <parametre nom="ville"/>
6       <parametre nom="pays" />
7     </methode>
8 ...
```

Exemple de code 3.11 – Extrait du contenu du registre de service

D Extension d'un environnement de modélisation et d'exécution d'une orchestration de services

De nombreux outils de modélisation et d'exécution des procédés existent dans le commerce [11]. Cependant, ils ne permettent pas de réaliser une orchestration de services hétérogènes et dynamiques sécurisée par le contrôle d'accès. Ces orchestrateurs nécessitent en effet de connaître les services à invoquer dès la conception de l'orchestration. De plus, ils ne permettent pas de spécifier une politique de contrôle d'accès à l'échelle de l'orchestration.

Afin d'implémenter et de valider notre approche, nous étendons un orchestrateur existant, Jboss JBPM. Nous ajoutons à Jboss JBPM des primitives de modélisation des services abstraits et du contrôle d'accès. Nous intégrons à l'architecture d'exécution de l'orchestration le registre et les outils d'exécution du contrôle d'accès que nous avons présentés dans le chapitre précédent.

1 Présentation de JBPM et des extensions nécessaires

Jboss JBPM est un outil de conception et d'exécution des procédés. Il permet de représenter un procédé sous la forme d'une suite de tâches et d'un ensemble de variables globales. Les tâches du procédé sont réalisées soit par des utilisateurs humains soit par des éléments logiciels, comme des services. Elles communiquent entre elles par lecture et affectation des variables globales.

JBPM utilise le métamodèle et la syntaxe graphique de BPMN 2.0 pour représenter les procédés. La Figure 3.27 donne un exemple de procédé spécifié avec JBPM.

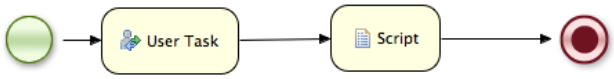

FIGURE 3.27 – Exemple de procédé spécifié avec JBPM

JBPM a pour but de permettre une conception collaborative des orchestrations de services. Il permet de rassembler deux types d'experts, les experts métiers, qui n'ont pas nécessairement de connaissances techniques, et les développeurs qui produisent le code d'appel des services. Un concepteur qui ne possède pas de compétences techniques peut spécifier la suite de tâches nécessaires pour exécuter le procédé et les données qu'il manipule. Pour chaque tâche, JBPM permet de spécifier comment elle est réalisée. Une tâche peut, par exemple, être exécutée sous la

11. On peut notamment citer BonitaSoft (www.bonitasoft.com), FlowMind (http ://www.flowmind.org/) , Business First (http ://www.w4global.com/) ou Jboss JBPM (www.jboss.org/jbpm/). Nos premiers prototypages se sont réalisés à partir de l'outil SECURE FOACS de S. Chollet qui propose de générer du code à partir des modèles d'une orchestration abstraite et de ses propriétés de sécurité pour invoquer des services duynamiques et sécurisés. Pour une description complète de SECURE FOCAS et de l'utilisation des templates dans l'outil qui a guidé notre réflexion, voir [Chollet 2009a]

forme d'un service web ou d'une tâche humaine. JBPM offre des connecteurs qui permettent de préciser les données d'entrée et de sortie de chaque tâche. Un concepteur ayant des connaissances techniques peut ainsi produire le code nécessaire à la communication entre l'orchestration JBPM, les éléments logiciels nécessaires à la réalisation du processus et les utilisateurs.

JBPM n'est pas adapté aux orchestrations de services hétérogènes et dynamiques, notamment parce qu'il s'appuie sur BPMN. Seuls les services web peuvent être utilisés par JBPM. Les services concrets utilisés par l'orchestration à son exécution doivent être connus dès la conception du procédé. Les concepteurs doivent ainsi, pour chaque service, programmer en Java le code qui permet de l'invoquer. Ce code est souvent complexe car il nécessite de connaître précisément la description du service et de savoir comment le manipuler. Dans le cas des tâches manuelles, par exemple, les développeurs doivent intégrer à l'orchestrateur un outil capable d'interagir avec les utilisateurs du procédé. Le recours à la programmation manuelle pour exécuter les tâches de l'orchestration est ainsi coûteux en temps et entraîne toujours le risque d'introduire des erreurs dans le code d'exécution de l'orchestration. Enfin, JBPM ne permet pas de définir ou d'exécuter une politique de contrôle d'accès.

La conception et l'exécution d'orchestrations de services hétérogènes et dynamiques sécurisée par le contrôle d'accès nécessite non seulement la collaboration entre les concepteurs de l'orchestration mais aussi la collaboration entre les experts fonctionnels et les experts en sécurité. Nous identifions, pour y parvenir, deux niveaux auxquels JBPM doit être étendu :

- Au niveau conception, il faut pouvoir se passer de programmation manuelle et s'abstraire de l'hétérogénéité et du dynamisme des services existants. Il faut aussi pouvoir définir une politique de contrôle d'accès. La définition de l'orchestration et celle du contrôle d'accès doivent pouvoir se faire séparément afin de permettre à chaque expert de manipuler un langage avec lequel il est familier.
- Au niveau exécution, il faut permettre la liaison retardée : les services concrets à exécuter doivent pouvoir être choisis à l'exécution seulement. Il faut aussi vérifier, tout au long de l'exécution de l'orchestration, le respect de la politique de contrôle d'accès et permettre aux utilisateurs d'interagir avec l'orchestration de manière sécurisée.

Nous étendons JBPM à ces deux niveaux. Au niveau conception, nous ajoutons aux primitives qui permettent de décrire des tâches dans JBPM une primitive pour décrire des tâches abstraites et sécurisées. Ces tâches permettent de définir un service abstrait et les règles de contrôle d'accès qui s'y appliquent. Afin d'intégrer les tâches sécurisées à JBPM, nous étendons le métamodèle de BPMN avec nos métamodèles. Au niveau exécution, nous intégrons notre registre de services, les sources d'informations et le point de décision que nous avons développés à l'architecture d'exécution d'une orchestration. Le reste de cette partie présente nos extensions et leur application pour la conception et l'exécution d'un procédé.

2 Niveau conception : création d'une tâche abstraite sécurisée

JBPM offre un mécanisme standardisé pour étendre le nombre de types de tâches que l'on peut modéliser : il est possible de définir des tâches personnalisées et leurs attributs. La définition des tâches personnalisées repose sur la création de trois fichiers :

- Un fichier texte de définition de la tâche personnalisée permet d'en préciser le nom et les attributs et les éléments graphiques associés.
- Une classe Java nommée *handler* permet de préciser les opérations que l'orchestrateur doit effectuer lorsqu'il parvient à une tâche personnalisée. spécification des éléments qui permettent de concevoir et de représenter la tâche.
- Un fichier qui permet d'indiquer à JBPM les chemins d'accès aux définitions des tâches personnalisées et à leurs *handlers*.

À partir de ces trois fichiers, JBPM gère l'affichage des tâches, leur exécution et l'interaction avec les utilisateurs.

Nous définissons une tâche abstraite sécurisée comme une tâche personnalisée dans JBPM. Chaque tâche sécurisée représente un service abstrait dont l'exécution est contrainte par des règles de contrôle d'accès. Les tâches abstraites sécurisées sont définies par :

- Leur nom, unique dans l'orchestration.
- Leur technologie, par exemple « service web », « UPnP », « DPWS » ou « Tâche humaine »
- Les règles d'accès qui s'y appliquent.
- La méthode du service abstrait qu'elle appelle.

Ces informations sont contenues dans le fichier de définition de la tâche sécurisée dont le code est le suivant :

```
1  import org.drools.process.core.datatype.impl.type.*;
2  [
3    // the Security work item
4    [
5       "name" : "Security Task",
6       "parameters" : [
7       "Service" : new StringDataType(),
8       "Methode" : new StringDataType(),
9       "RegleAc" : new StringDataType(),
10      "Techno" : new StringDataType(),
11     ],
12     "displayName" : "SecurityTask",
13     "icon" : "icons/cadena.png"
14   ]
15 ]
```

Exemple de code 3.12 – Définition d'une tâche sécurisée

La Figure 3.28 présente la représentation graphique d'une tâche abstraite sécurisée de récupération des données météorologiques. La Figure présente aussi la boîte de dialogue qui lui est associée. Cette boîte de dialogue permet à un concepteur de spécifier une tâche sécurisée et la représentation graphique de cette dernière. Nous choisissons d'utiliser une boîte de dialogue plutôt que de représenter tous les éléments sur le modèle de l'orchestration afin de ne pas le surcharger. La boîte de dialogue permet la collaboration des concepteurs de l'orchestration sécurisée :

l'expert fonctionnel peut spécifier les tâches à utiliser et les données nécessaires au processus comme il le fait habituellement dans JBPM. L'expert en sécurité peut, lui, éditer les règles de contrôle d'accès au moyen d'une boîte de dialogue dédiée.

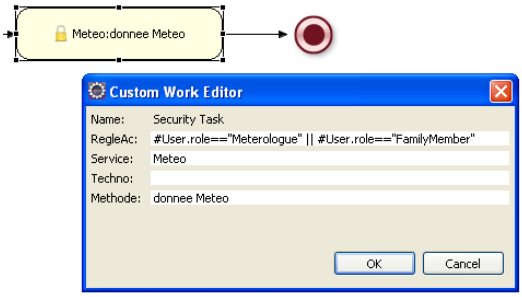

FIGURE 3.28 – Tâche abstraite et Édition des propriétés de contrôle d'accès

La spécification des règles de contrôle d'accès se fait au moyen du langage de contrôle d'accès que nous avons défini. L'accès au service de météo, par exemple, est contraint par la valeur des rôles de l'utilisateur.

3 Niveau exécution : liaison retardée et exécution du contrôle d'accès

Nous étendons l'orchestrateur afin de remplacer la liaison de la composition avec un ensemble de services spécifiques dès sa conception par la liaison retardée. Les services nécessaires ne sont sélectionnés qu'à l'exécution parmi les service concrets effectivement disponibles. Lorsqu'un service est invoqué, la satisfaction de la politique de contrôle d'accès est vérifiée.

L'intégration du registre de services que nous avons développé à l'architecture d'exécution de l'orchestration permet de mettre en œuvre la liaison retardée. Lorsque l'orchestrateur parvient à une tâche abstraite sécurisée, il contacte le registre de services en lui fournissant le nom de la fonctionnalité qu'il souhaite exécuter. Le registre de services sélectionne alors le service concret adéquat et l'invoque.

L'intégration de notre registre de services à JBPM permet aussi la mise en œuvre du contrôle d'accès. À l'inscription d'un service, le registre de service génère un *proxy* sécurisé chargé d'exécuter le contrôle d'accès. Le registre de services ne peut contacter que les *proxies* et non pas directement les services concrets. Toutes les requêtes transitent ainsi par les *proxies* et le contrôle d'accès est donc obligatoirement exécuté. Lorsqu'un *proxy* est invoqué, il demande une décision de contrôle d'accès au point de décision. Le point de décision se charge ensuite de contacter les sources d'information nécessaires et d'évaluer la politique de contrôle d'accès.

Si la tâche à exécuter nécessite l'intervention d'un utilisateur, par exemple afin de fournir des informations, le *proxy* demande au point de décision de rechercher les utilisateurs autorisés à réaliser la tâche. Nous avons développé un service de notification qui permet d'informer les utilisateurs des tâches qu'ils peuvent effectuer. Chaque utilisateur possède une pile de tâches que le service de notification met à

jour. Pour chaque tâche dans sa pile, l'utilisateur peut choisir de l'effectuer ou de l'ignorer – sauf s'il est obligé de l'effectuer. Lorsqu'un utilisateur traite une tâche, elle devient inaccessible pour les autres utilisateurs. L'interaction des utilisateurs avec les tâches qu'ils peuvent exécuter se fait au moyen de formulaires qui sont générés pour chaque tâche à partir de la description du service concret choisi pour l'exécuter. Lorsqu'une tâche est traitée, elle disparaît de la pile des utilisateurs qui peuvent la réaliser. La Figure 3.29 présente un exemple de formulaire au travers duquel un utilisateur peut interagir avec la tâche météo en fournissant les paramètres qui permettent de rechercher des données météorologiques.

FIGURE 3.29 – Interface d'utilisation d'une tâche sécurisée

4 Synthèse des extensions apportées à JBPM

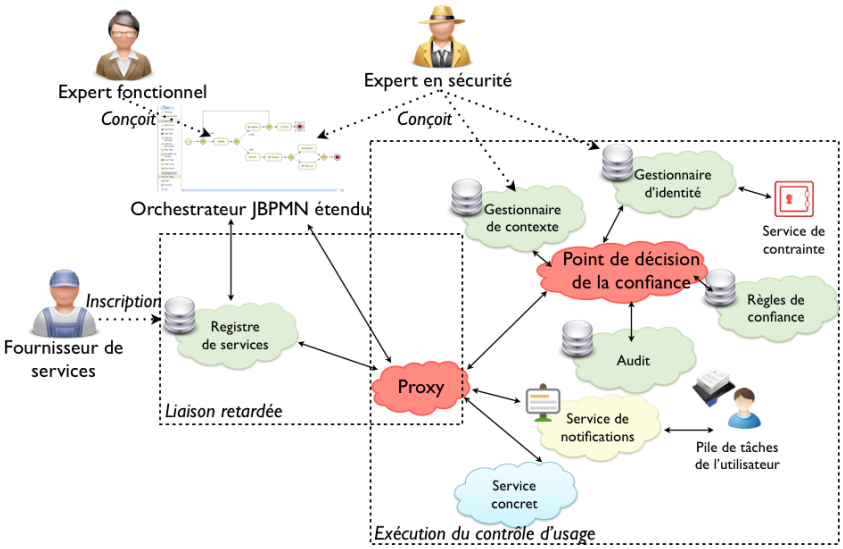

FIGURE 3.30 – Synthèse des extensions apportées à JBPM

Les extensions que nous avons apportées à JBPM visent à permettre la concep-

tion et l'exécution d'une orchestration de services hétérogènes et dynamiques sécurisée par le contrôle d'accès. Nous avons étendu l'environnement de modélisation de JBPM afin de permettre de spécifier des services abstraits sécurisés. Cette extension permet aux experts en sécurité et aux experts fonctionnels de collaborer. Les experts fonctionnels définissent les tâches nécessaires à l'exécution et les données qu'elles s'échangent. Les experts en sécurité définissent la politique de contrôle d'accès et les données nécessaires au contrôle d'accès.

Nous avons étendu l'architecture orientée service d'exécution d'une orchestration proposée par JBPM. Nous avons ajouté un registre de services qui permet la liaison retardée et la sécurisation des services concrets. Nous avons aussi intégré à JBPM des composés dédiés à l'exécution du contrôle d'accès et à la gestion des participants à l'orchestration. Ils sont prévenus des tâches qu'ils peuvent effectuer et peuvent interagir avec l'orchestration au moyen de formulaires générés dynamiquement. La Figure 3.30 reprend la conception et l'exécution d'une orchestration de services hétérogènes et dynamiques à l'aide des extensions que nous avons apportées à JBPM. Elle présente aussi les différents acteurs qui participent à la conception et à l'exécution de l'orchestration.

E Cas d'utilisation

Dans ce chapitre, nous utilisons la version de JBPM que nous avons étendue afin de concevoir et d'exécuter une orchestration de services hétérogènes et dynamiques sécurisée par le contrôle d'accès. Nous présentons tout d'abord la démarche générale d'utilisation de l'orchestration puis le cas d'utilisation sur lequel nous les appliquons et les résultats que nous obtenons.

1 Démarche de conception et d'exécution d'une orchestration de services hétérogènes et dynamiques sécurisés par le contrôle d'accès

Notre proposition divise la production d'une orchestration de services en une phase de conception et une phase d'exécution :

- La phase d'exécution permet de spécifier l'orchestration et sa politique de contrôle d'accès dans l'orchestrateur de JBPM étendu. Elle permet aussi de cartographier les informations nécessaires au contrôle d'accès et leur niveau de visibilité.
- La phase d'exécution repose sur l'utilisation du registre de services pour invoquer les services nécessaires et vérifier la satisfaction de la politique de contrôle d'accès.

À tout moment, des fournisseurs de service peuvent inscrire ou désinscrire des services. L'orchestration doit ainsi pouvoir s'exécuter en s'adaptant aux services concrets effectivement disponibles.

2 Définition du cas d'utilisation

Nous appliquons notre démarche à l'exemple d'un procédé simplifié de détection et de gestion des alertes médicales dans un bâtiment intelligent de santé. Dans un bâtiment intelligent de santé, par exemple une maison ou un appartement, se trouvent des capteurs et des outils de stockage et de gestion des données médicales. Les capteurs permettent de collecter des données sur l'environnement du bâtiment ou ses habitants. Les bases de données permettent de maintenir, par exemple, des informations sur le dossier médical du patient.

Le procédé est représenté sur la Figure 3.31. Il démarre par l'analyse de l'environnement, par exemple en enregistrant la température ou la concentration de certains allergènes. En parallèle, le procédé analyse les données médicales des habitants qu'il faut surveiller. Par exemple, le procédé peut vérifier qu'aucun habitant n'est tombé et enregistrer ses constantes physiologiques.

Les informations obtenues à l'issue de ces activités d'analyse sont ensuite utilisées afin de détecter des alertes. Une alerte représente une situation à risque pour un habitant. Une alerte est définie par des contraintes sur la valeur d'un ensemble de propriétés de l'environnement et des habitants.

Lorsqu'une alerte est détectée, selon son niveau de gravité, elle est soit transmise à la personne pour qu'elle en prenne connaissance soit transmise à un médecin régulateur qui décide alors de la nécessité d'envoyer une équipe médicale.

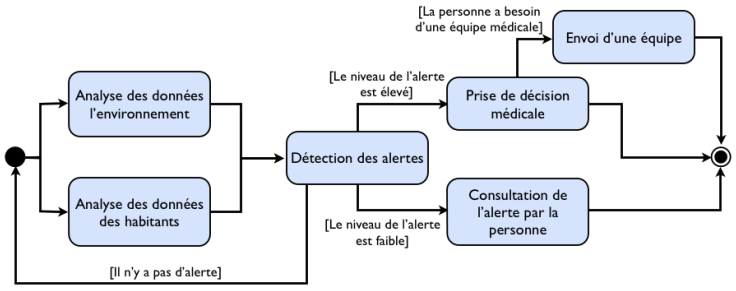

FIGURE 3.31 – Exemple d'un procédé de récolte et d'analyse de données

Nous choisissons ce procédé car il possède trois propriétés qui nous intéressent. Il repose sur des services hétérogènes. Les activités d'analyses sont réalisées par des capteurs, l'activité de détection des alertes repose sur une base de données et les autres activités sont réalisées par des être humains. Certains de ces services sont dynamiques. Les capteurs, par exemple, sont susceptibles de quitter ou de rejoindre la composition. Enfin, le procédé met en jeu des informations sensibles sur la santé des patients. Les activités de prise de décision médicale et de consultation des alertes doivent donc être sécurisées afin de garantir que seuls les utilisateurs autorisés peuvent accéder aux informations médicales.

Nous présentons dans les sections qui suivent l'utilisation de notre démarche pour concevoir et exécuter ce procédé en mettant en œuvre une politique de contrôle d'accès.

3 Conception du procédé

La première étape de notre démarche est la conception du procédé. La phase de conception permet de définir les activités du procédé et les produits qu'elles consomment. Dans le cas du procédé de détection et de gestion des alertes, nous identifions six activités :

- L'analyse de l'environnement. Cette activité est réalisée par des services UPnP, DPWS ou des services web qui permettent de capturer les informations sur l'environnement du bâtiment. Ces services sont dynamiques.
- L'analyse des habitants. Cette activité est réalisée par des services UPnP ou DPWS qui permettent de capturer les informations sur les constantes physiologiques des habitants. Ces services sont dynamiques.
- La détection des alertes repose sur un service web qui accède à la base des alertes potentielles qui concernent le patient et les évalue.
- La prise de décision médicale. Cette activité est réalisée par une tâche humaine.
- La consultation de l'alerte médicale. Cette activité est réalisée par une tâche humaine.
- L'envoi d'une équipe médicale. Cette activité est réalisée par une tâche humaine.

La conception du procédé est réalisée à l'aide de l'environnement graphique fournipar notre orchestrateur. Le résultat est présenté sur la Figure 3.32.

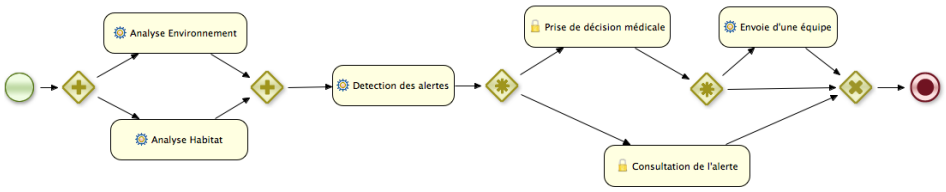

FIGURE 3.32 – Représentation graphique du procédé

L'orchestrateur permet de définir les services abstraits et leurs règles d'accès. Ils sont identifiés par une icône en forme de cadenas. Pour tous les services abstraits, le concepteur peut spécifier une technologie cible au travers d'un onglet présenté sur la Figure 3.33. Le modèle graphique de l'orchestration reste ainsi visuellement très simple, ce qui en facilite la compréhension.

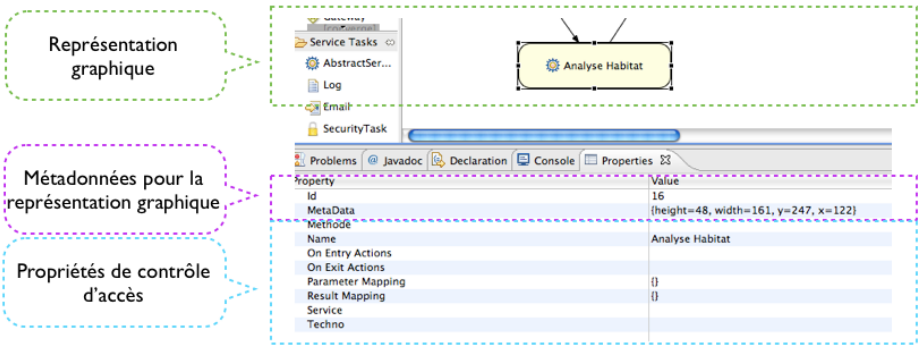

FIGURE 3.33 – Représentation d'une tâche abstraite sécurisée

L'orchestrateur permet aussi de définir les produits qui sont utilisés par le procédé et à quel moment du procédé ils sont utilisés. Les activités communiquent entre elles au moyen d'affectation de variables. Afin de détecter et de gérer les alertes, nous identifions quatre produits :

- **La description d'un habitant.** Une table de hachage où les clés représentent les noms des propriété. Chaque clé est associée à la valeur des propriétés d'un habitant. Cette description est obtenue en sortie du service d'analyse des habitants et est utilisée comme entrée par le service de détection des alertes.
- **La description de l'environnement d'un habitant,** une table de hachage. Les clés représentent le nom d'une propriété de l'environnement à laquelle est associée sa valeur. Cette description est obtenue en sortie du service d'analyse de l'environnement et est utilisée comme entrée par le service de détection des alertes.

- **Une liste d'alertes détectées.** Chaque alerte est définie par son nom, sa description et son niveau de gravité. La liste des alertes est obtenue en sortie du service de détection des alertes et est donnée en entrée des services de prise de décision médicale et de consultation d'une alerte.
- **La décision d'envoi d'une équipe médicale,** une variable booléenne. Sa valeur est modifiée par le service de prise de décision médicale.

Les produits sont définis dans des onglets qui permettent d'éditer les propriétés du procédé et des activités et de ne pas surcharger le modèle graphique du procédé. La Figure 3.34 présente l'onglet d'édition des propriétés du procédé.

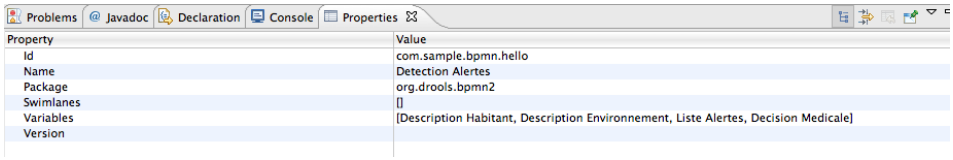

FIGURE 3.34 – onglet d'édition des propriétés du procédé

4 Application du contrôle d'accès au cas d'utilisation

Les activités de prise de décision médicale et de consultation des alertes doivent être sécurisées par des règles de contrôle d'accès. Nous retenons les règles suivantes :

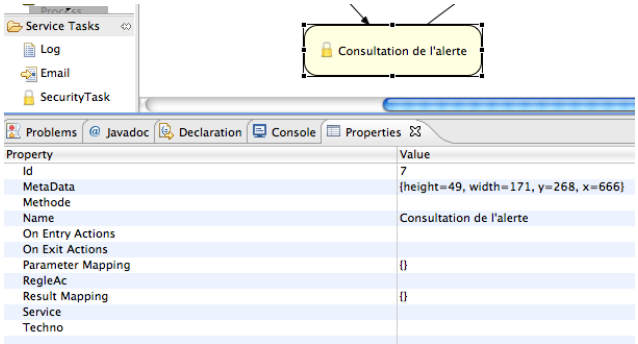

FIGURE 3.35 – Edition des règles de contrôle d'accès

- Règle 1 : Afin de prendre une décision médicale, un utilisateur doit être médecin dans la spécialité concernée par l'alerte. Il doit être disponible, c'est-à-dire ne pas être en rendez-vous ou déjà en train de prendre une décision. Il doit aussi être proche du domicile du patient sur lequel l'alerte porte.
- Règle 2 : Afin de consulter une alerte, un utilisateur doit être patient et l'alerte doit le concerner.

Ces règles peuvent être définies dans l'orchestrateur : les tâches de prise de décisions médicales et de consultation des alertes sont des tâches sécurisées. Elles sont indiquées sur le procédé avec une icône spécifique comme le montre la Figure 3.35. Les règles qui s'appliquent à chaque tâche peuvent être spécifiées au travers d'une boîte de dialogue ou d'un onglet dédié. La Figure 3.35 présente l'onglet qui permet d'éditer les règles qui s'appliquent à l'activité de consultation des alertes.

La vie privée des médecins peut être mise en danger par le contrôle d'accès pervasif. La collecte de la disponibilité des médecins et de leur localisation peut livrer des informations sensibles. Nous identifions ainsi deux attributs privés :

- La localisation de l'utilisateur.
- La disponibilité de l'utilisateur.

Ces attributs sont définis comme privés lors de la définition du schéma de la base de données du gestionnaire d'identité.

Obstacle	Cause	Prise en compte
Hétérogénéité des services	Types et implémentations de services variées	Abstraction des détails techniques à la conception et génération de code spécifique à une implémentation à l'exécution.
Dynamisme des services	Arrivée et départ de services à l'exécution	Abstraction à la conception et sélection des services à invoquer à l'exécution.
Protection de la vie privée	Possibilité de fuite de données	Annotation d'un procédé avec des règles de contrôle d'accès et proxification des services à l'exécution

TABLE 3.4 – Prise en compte de l'hétérogénéité, du dynamisme des services et de la vie privée

5 Utilisation de l'orchestrateur étendu pour exécuter le procédé

L'exécution d'un procédé dans JBPM se fait au moyen de deux fichiers :

- Un modèle graphique permet de représenter les activités du procédé et les variables globales du procédé.
- Une classe Java permet d'exécuter le procédé et de faire appel aux ressources logiciels nécessaires.

Nous conservons ce mode d'exécution. Le modèle graphique est augmenté des services abstraits sécurisés. La classe Java permet de référencer le *handler* des services abstraits sécurisés et de lancer l'exécution du procédé.

Le Tableau 3.4 présente de manière synoptique comment l'utilisation de notre orchestrateur étendu permet de prendre en compte le dynamisme et l'hétérogénéité des services ainsi que la vie privée dans une orchestration de services.

F Expérimentations et validation

Étendre une architecture d'exécution d'une orchestration afin de mettre en œuvre le contrôle d'accès augmente le temps d'exécution d'une orchestration. Notre hypothèse est que ce surcoût de temps d'exécution est raisonnable – il n'augmente pas de manière exponentielle – et que notre solution peut ainsi être mise en œuvre dans les cas où un concepteur souhaite protéger la vie privée dans l'application qu'il réalise.

1 Coût de l'exécution du contrôle d'accès

Le coût de l'exécution du contrôle d'accès provient de la génération des *proxies* sécurisés, de la vérification de la politique de sécurité et de la gestion des notifications pour les utilisateurs.

La génération du *proxy* est l'opération la plus coûteuse de notre proposition. Cependant, elle prend à peu près toujours le même temps – environ 7 secondes. De plus, la génération du *proxy* n'intervient qu'une fois et à l'inscription du service. Par conséquent, elle n'a pas d'influence sur l'exécution de l'orchestration qui repose, elle, sur l'invocation des services.

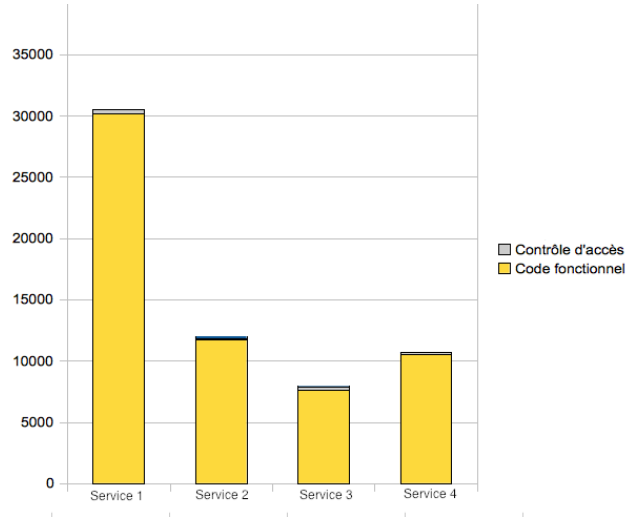

FIGURE 3.36 – Coût de l'exécution du contrôle d'accès par rapport au coût d'invocation des services

La Figure 3.36 représente le coût de l'exécution du contrôle d'accès pour quatre services utilisés lors de l'exécution de notre cas d'utilisation. La Figure représente aussi le temps d'exécution du code fonctionnel, c'est-à-dire le temps nécessaire pour l'invocation des fonctionnalités des services. Service 1 et Service 2 sont chargés de l'analyse des données, Service 3 et Service 4 sont chargés de leur présentation.

Le temps d'exécution du contrôle d'accès comprend la demande d'une décision auprès du point de décision et sa mise en œuvre. Dans le cas de la politique de

contrôle d'accès de notre cas d'utilisation, ce temps d'exécution est très faible – entre 100 et 400 millisecondes. Il ne représente ainsi que moins de 2% de l'invocation d'un service sécurisé à l'aide de notre approche. Notre proposition est ainsi pertinente dès que l'appel du code fonctionnel prend au moins 400 millisecondes. Dans le cas contraire, l'exécution du contrôle d'accès serait plus longue que l'exécution du code fonctionnel.

Dans le monde des services, surtout dans le cas des services web utilisés dans le cas de sites Internet, le délai de réponse d'un service doit être faible afin de ne pas perturber la navigation de l'utilisateur. Nous étudions désormais comment le coût de l'évaluation d'une politique de contrôle d'accès évolue avec sa complexité.

2 Coût de l'évaluation de la politique de contrôle d'accès

Dans la mesure où nous prenons en compte de nombreux éléments lorsque nous modélisons la politique de contrôle d'accès, cette dernière est susceptible d'être de très grande taille. La complexité théorique de la vérification d'une politique de contrôle d'accès est$O(|C| * |\varphi|)$ où $|C|$ est la taille de la composition déjà effectuée à un instant t et $|\varphi|$ le nombre de littéraux dans la politique de contrôle d'accès. On peut donc supposer une progression linéaire du temps nécessaire à l'évaluation de la politique de contrôle d'accès : plus la politique est grande – ou plus on progresse dans la composition – et plus il faudra du temps pour exécuter la politique de contrôle d'accès.

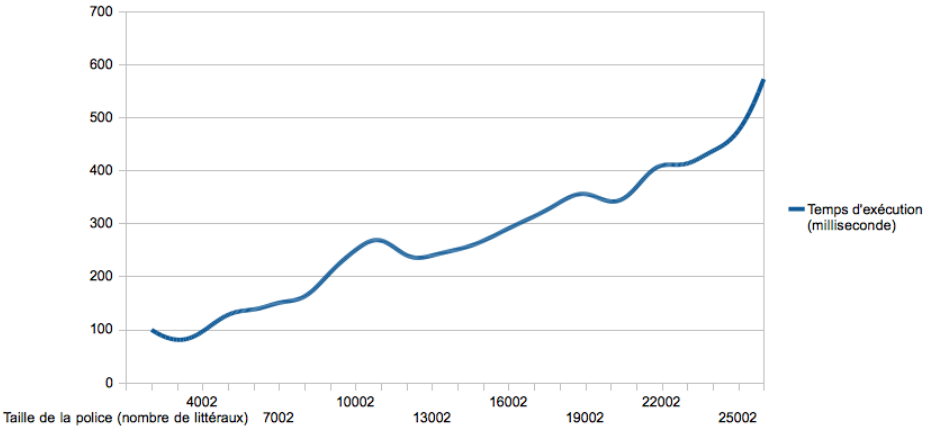

FIGURE 3.37 – Coût de l'évaluation du contrôle d'accès

Afin de vérifier cette supposition et d'établir que le temps nécessaire à la vérification de la politique de contrôle d'accès est faible, nous avons testé notre point de décision sur des politiques de tailles variées. Nous avons construit des politiques comprenant de 1000 à 26000 littéraux. Nous avons, pour chaque politique, calculé le temps nécessaire pour la vérifier. Ce temps représente le temps nécessaire pour obtenir les informations utiles pour évaluer les règles de la politique ainsi que le

temps nécessaire pour vérifier les règles à l'aune de ces informations. Ce temps d'exécution est ainsi indépendant du nombre de services à exécuter et de leur temps d'exécution.

La Figure 3.37 montre que dans le cadre de notre implémentation du point de décision et des sources d'information, l'exécution de politiques de contrôle d'accès prend moins d'une seconde même lorsque la politique comprend plus de 22000 littéraux.

Cette expérimentation permet de déterminer le coût de notre proposition en terme de temps d'exécution. La protection de la vie privée représente un surcoût de temps. Déterminer la relation entre le coût de l'évaluation de la politique de contrôle d'accès et sa taille permet ainsi de guider la conception d'une politique de contrôle d'accès dans les cas où d'autres propriétés – comme le temps de réaction de l'application – doivent être satisfaites.

G Synthèse

Niveau de l'approche	Technologies utilisées pour le prototypage	Apports
Conception	Java, JBPM, JET	Langage d'expression de contraintes de contrôle d'accès. Extensions de JBPM. Liens entre le modèle de procédé et les contraintes de vie privée.
Exécution	JAVA, BPMN, Drools	Génération automatique de code Proxification : Intégration du contrôle d'accès dans une SOA Prise en compte de l'hétérogénéité et du dynamisme des services

Table 3.5 – Apports de notre réalisation

Nous avons présenté dans ce chapitre l'implémentation de notre approche. Nous avons étendu l'orchestrateur Jboss JBPM afin de permettre la conception d'une orchestration et de la politique de contrôle d'accès qui lui est associée. Le Tableau G reprend les technologies que nous avons utilisées à des fins de protypage et les apports de notre implémentation.

L'orchestration est ensuite exécutée dans une architecture orientée service étendue. Cette architecture repose sur un registre de services qui permet de générer, pour chaque service à sécuriser, un *proxy* qui joue le rôle de point de mise en œuvre du contrôle d'accès. La capture des données nécessaires à l'exécution du contrôle d'accès et le calcul des décisions de contrôle d'accès sont réalisés par des services web dédiés.

Nous avons appliqué notre approche à un exemple de la détection et de la gestion des alertes médicales. Nous avons montré que notre approche permettait de modéliser l'orchestration et le contrôle d'accès. Nous avons aussi montré qu'elle n'augmentait que très peu le temps d'exécution de la composition. Enfin, nous avons montré que, même lorsque la politique de contrôle d'accès était de grande taille, notre proposition restait efficace. Nous présentons désormais les perspectives de notre travail.

IV Perspectives et extensions

Nous présentons dans cette section les perspectives de notre travail. Nous introduisons le principe général de l'extension de notre démarche. Nous montrons que le découpage de la production d'orchestrations de services hétérogènes et dynamiques en une phase de conception et une phase d'exécution peut s'appliquer à d'autres propriétés non-fonctionnelles.

A Extension de la démarche à d'autres propriétés

1 Principe général de l'extension de la démarche

Notre démarche dirigée par les modèles repose sur deux phases représentée sur la Figure 3.38.

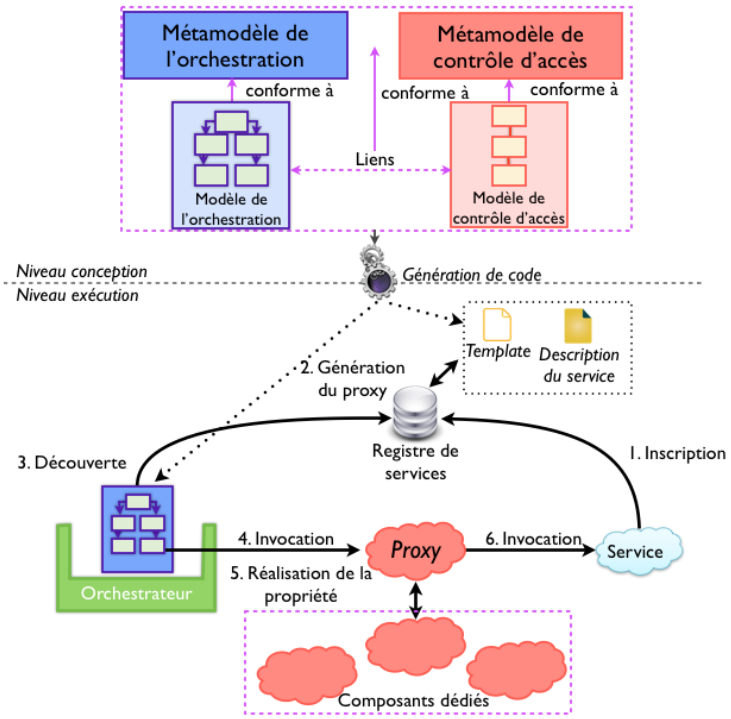

FIGURE 3.38 – Principe général de notre démarche

- Une orchestration de services et ses propriétés non fonctionnelles sont spécifiées à un niveau abstrait afin de laisser de côté les détails liés à l'implémentation.
- À l'exécution, la sélection des services permet de s'abstraire du dynamisme des services. La génération de code adapté aux technologies des services disponibles permet, elle, de dépasser leur hétérogénéité.

Le niveau conception repose sur la composition d'un métamodèle fonctionnel – le métamodèle de l'orchestration – avec un ou plusieurs métamodèles non-fonctionnels. Chaque métamodèle est préservé lors de la conception afin de conserver le vocabulaire propre à chaque vue.

Le niveau exécution repose sur le développement de transformations de modèles qui permettent de générer le code nécessaire à l'exécution de l'orchestration étendue de propriété non fonctionnelle. Il faut aussi développer les composants nécessaires à l'exécution de ce code.

1.1 Extension du niveau conception Notre niveau conception est construit autour des principes d'abstraction et de séparation des préoccupations. Nous laissons de côté les détails techniques. Chaque propriété fonctionnelle et non-fonctionnelle est décrite à l'aide d'un métamodèle autonome. Les métamodèles sont ensuite reliés. Le premier point d'extension de notre démarche est ainsi l'ajout d'une nouvelle propriété non fonctionnelle au niveau conception. Il faut distinguer deux cas :

- Soit, comme nous l'avons fait pour le contrôle d'accès, on n'ajoute qu'une propriété non-fonctionnelle à l'orchestration et, dans ce cas, seuls les liens entre les métamodèles de l'orchestration et de la propriété non-fonctionnelle doivent être déterminés.
- Soit on souhaite ajouter plusieurs propriétés non fonctionnelles à une orchestration et, dans ce cas, il faut alors non seulement déterminer les liens entre les métamodèles des propriétés et celui de l'orchestration, mais aussi entre les métamodèles des propriétés non fonctionnelles. Ces dernières peuvent ainsi être incompatibles ou nécessiter d'être exécutées dans un ordre spécifique.

Sur le plan de l'outillage, l'extension du niveau conception conduit à la modification de l'orchestrateur afin de permettre la représentation des nouvelles propriétés non-fonctionnelles.

1.2 Extension du niveau exécution Deux éléments doivent être modifiés au niveau exécution afin de prendre en compte l'ajout de nouvelles propriétés :

- Un ensemble de composants dédiés à l'exécution de la propriété doivent être développés. Dans le cas de l'ajout de la propriété d'audit à la composition, par exemple, il faut développer un outil de surveillance et d'enregistrement de l'exécution de la composition.
- Les transformations de modèle vers textes qui permettent de générer le code des *proxies* doivent être modifiés afin de permettre le tissage du code fonctionnel et du code non-fonctionnel. Les *proxies* doivent ainsi être capables d'appeler les composants ajoutés à l'orchestration et de leur fournir les informations pertinentes.

Dans le cadre de notre approche, l'extension du niveau exécution est réalisée en modifiant les templates de *proxies* et en ajoutant les composants nécessaires à l'exécution d'une propriété non fonctionnelle.

B Extension de la démarche au contrôle d'usage

1 Définition du contrôle d'usage et des besoins en termes d'extension

Le contrôle d'usage désigne la surveillance, dans une application, de l'utilisation des données et des ressources afin de vérifier qu'elles sont bien employées de manière autorisée. La gestion des droits numériques (notée DRM pour *Digital Right Management*) est un exemple de contrôle d'usage : l'utilisation d'un support numérique est restreinte, par exemple à une zone géographique ou selon un nombre d'activations.

Dans le cas des applications collaboratives et des procédés, le contrôle d'usage est souvent mis en œuvre en enregistrant l'activité des utilisateurs. À chaque utilisateur est donné un indice de confiance. Cet indice est mis à jour lorsque l'utilisateur réalise des actions. En fonction de la valeur de cet indice, l'utilisateur se voit accorder des privilèges différents. Lorsque l'utilisateur atteint un niveau de confiance trop bas, par exemple, il peut se voir refuser l'accès à l'application. Le contrôle d'usage est ainsi utilisé afin de partager des données sensibles uniquement avec des tiers de confiance.

La Figure 3.39 présente le métamodèle du contrôle d'usage que nous proposons de mettre en œuvre.

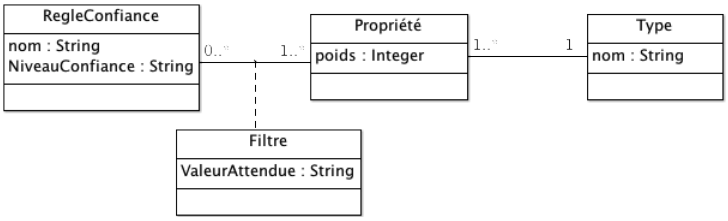

FIGURE 3.39 – Métamodèle du contrôle d'usage

Le métamodèle repose sur l'entité « RègleConfiance ». Une règle de confiance précise le niveau de confiance nécessaire pour que la règle soit satisfaite. Ce niveau est calculé à partir de la valeur d'un ensemble de propriétés surveillées : lorsqu'une action est effectuée, on s'attend à ce qu'une propriété ait une certaine valeur. Une propriété peut être toute information d'une application, comme l'heure à laquelle un utilisateur effectue une action ou son résultat. Chaque propriété a un poids dans la détermination de la confiance : plus la propriété est importante, plus son poids est élevé. Le poids d'une propriété est compris entre 0 et 1. La somme des poids des propriétés qui participent à une règle vaut 1.

La mise en œuvre du contrôle d'usage repose sur l'audit des propriétés surveillées. La Figure 3.40 présente le métamodèle de l'audit.

Au sein d'une application, toutes les propriétés peuvent ainsi être enregistrées dans un fichier.

Afin de montrer que notre démarche s'applique à d'autres propriétés non-fonctionnelles, nous identifions deux besoins d'extension pour ajouter le contrôle d'usage à une orchestration de services hétérogènes et dynamiques :

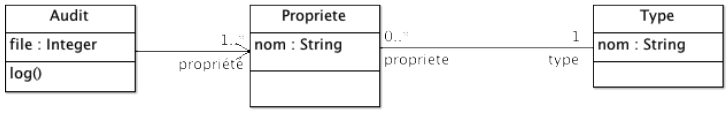

FIGURE 3.40 – Métamodèle de l'audit

- Au niveau conception, le métamodèle du contrôle d'usage et le métamodèle d'audit doivent être liés au métamodèle de l'orchestration. Ces liens permettent de modéliser les niveaux de confiance nécessaires pour effectuer les différentes activités de l'orchestration. Les métamodèles doivent aussi être liés entre eux.
- Au niveau exécution, le contrôle d'usage repose sur la mise en œuvre de la propriété d'audit qui permet d'enregistrer les actions des utilisateurs. Elle repose aussi sur le développement d'un service qui permet de mettre à jour les niveaux de confiance des utilisateurs.

Nous présentons désormais ces deux niveaux d'extension.

1.1 Niveau conception : composition d'un métamodèle de contrôle d'usage et d'un métamodèle d'orchestration Nous divisons l'extension du niveau conception en deux étapes. Tout d'abord, nous identifions les liens entre le métamodèle d'audit et celui du contrôle d'usage. Ensuite, nous établissons les liens entre ces métamodèles et le métamodèle de l'orchestration.

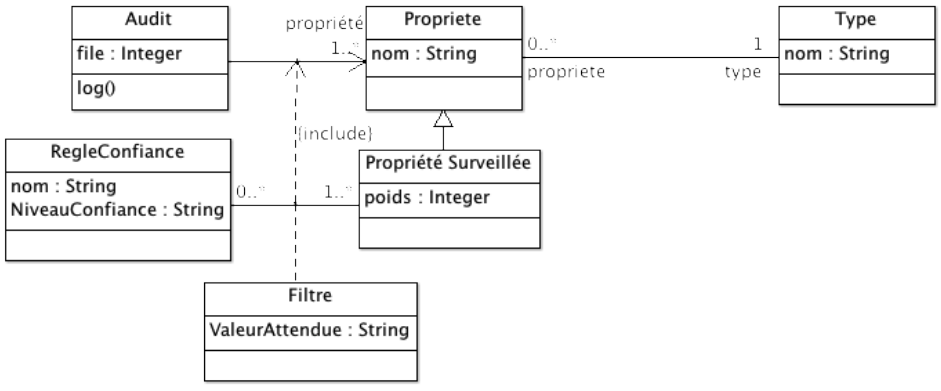

FIGURE 3.41 – Composition des métamodèles de confiance et d'audit

Afin d'exécuter le contrôle d'usage, les propriétés sur lesquelles il porte doivent être enregistrées. Dans la mesure où les métamodèles de contrôle d'usage et d'audit reposent tous les deux sur l'entité « Propriété » mais que les propriétés manipulées par le métamodèle de contrôle d'usage possèdent un attribut « poids » qui leur est propre, nous établissons que la classe « Propriété » du métamodèle de contrôle d'usage – que nous renommons « Propriété surveillée » - est une sous-classe de la

classe « Propriété » du métamodèle d'audit. Nous ajoutons, de plus, que lorsqu'une propriété est surveillée – elle est utilisée dans une règle de confiance – elle doit être enregistrée. Cette contrainte est exprimée à l'aide d'un lien d'inclusion. La composition des métamodèles ainsi obtenue est présentée sur la Figure 3.41.

L'audit et les règles de confiance portent sur les activités de la composition, nous relions ainsi les classes « RegleConfiance » et « Audit » avec la classe « Activité » du métamodèle d'orchestration.

Le résultat de la composition des métamodèles est présenté sur la Figure 3.42.

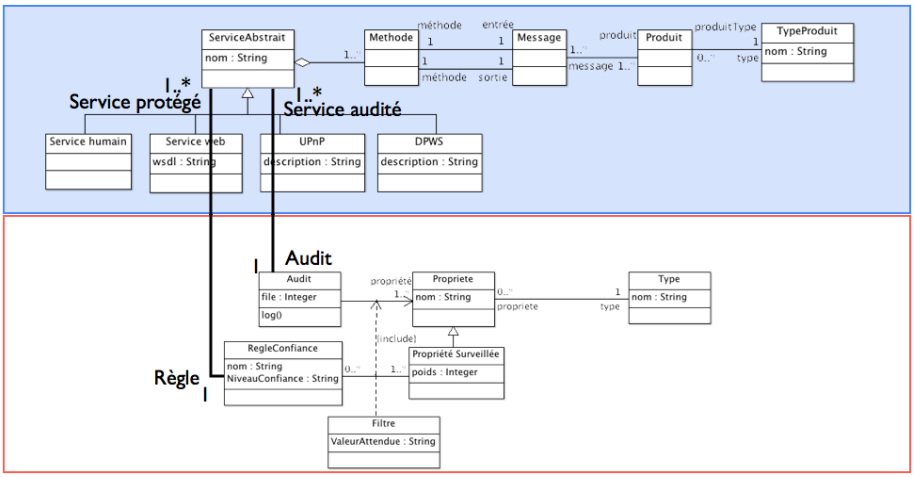

FIGURE 3.42 – Composition des métamodèles

1.2 Niveau exécution : modification des templates de génération des *proxies* À l'exécution, lorsqu'on parvient à un service abstrait protégé par une règle de contrôle d'usage, les valeurs des propriétés sur lesquelles cette règle porte doivent être enregistrées. Les utilisateurs avec le niveau de confiance nécessaire doivent être sélectionnés.

Afin de garantir que le contrôle d'usage sera toujours exécuté et de s'abstraire de l'hétérogénéité des services, nous conservons notre mécanisme de génération de *proxies* à l'aide de templates. Nous modifions ainsi les *proxies* afin de tisser le code fonctionnel et le code nécessaire à l'exécution du contrôle d'usage. Le code suivant donne un extrait du proxy :

```
1  public java.lang.String securedmin(java.lang.String Wsdl, java.
       lang.String sessionID, java.lang.String activity) {
2    long timestampAvantExecution= System.currentTimeMillis();
3    //port to the non secured service
4    OpeService_pkg.OpeServiceService portToService = new
         OpeService_pkg.OpeServiceServiceLocator();
5    //Creation du decision point depuis le endpoint
6    TrustDecisionPointImplService dp = new
         TrustDecisionPointImplServiceLocator();
7    try {
8      TrustDecisionPointImpl portToDecisionPoint = dp.getPDP();
9      String[] lstUsers=null;
```

```
10        try {//Test : l'utilisateur a-t-il le droit d'acceder a l'
              activite ?
11          lstUsers = portToDecisionPoint.lesUtilisateursDeConfiance(
                activity);
12          }
13        } catch (RemoteException e) { e.printStackTrace();}
14        if (lstUsers!=null) { //Si l'utilisateur a les droits
              necessaires, on appelle methode
15          NotificationServiceServiceLocator session = new
                NotificationServiceServiceLocator();
16            session.setMaintainSession(true);
17            NotificationServiceService service = session;
18            try {
19              NotificationService notification = service.
                    getNotificationService();
20              String id = notification.sendNotification(activity,
                    lstUsers,"min",Wsdl);
21              while(!(notification.getEtatNotification(id).equals("
                    Terminer"))){
22              try {
23                Thread.sleep(1000);
24              } catch (InterruptedException e) {e.printStackTrace()
                    ;}
25              }
26              String donnee = notification.getInfoNotification(id);
27              String[] param = donnee.split(" :: ");
28            OpeService port = portToService.getOpeService(); //
                  Recuperation du Stub qui implemente le Service
                  Definition Type
29            long timestampAvantAppel= System.currentTimeMillis();
30            java.lang.String resultat=port.min(param[0]);
31            return resultat;
32
33
34          } catch (RemoteException e) {e.printStackTrace();}
35        }
36      } catch (ServiceException e) {
37        e.printStackTrace();
38      }
39      return null;
40    }
```

Exemple de code 3.13 – Exemple de message SOAP

Avant d'appeler la méthode d'un service concret, le *proxy* demande au point de décision quels sont les utilisateurs qui possèdent un niveau de confiance nécessaire pour réaliser l'activité. Une fois la méthode appelée et exécutée, le *proxy* demande l'enregistrement des propriétés nécessaires. Le niveau de confiance de l'utilisateur est alors mis à jour.

Nous avons présenté jusqu'ici les modifications à apporter au niveau conception, nous montrons désormais les changements nécessaires en termes d'implémentation afin d'exécuter le contrôle d'usage.

2 Implémentation

2.1 Extension de l'orchestrateur Afin de permettre la spécification des règles de confiance, nous étendons de nouveau Jboss JBPM. À chaque service abstrait est associé une boîte de dialogue et un onglet de propriété qui permet de préciser le niveau de confiance nécessaire pour exécuter le service ainsi que les propriétés à partir desquelles ce niveau est calculé.

La Figure 3.43 présente la boite de dialogue et l'onglet de propriété du service abstrait TrustedTask protégée par le contrôle d'usage.

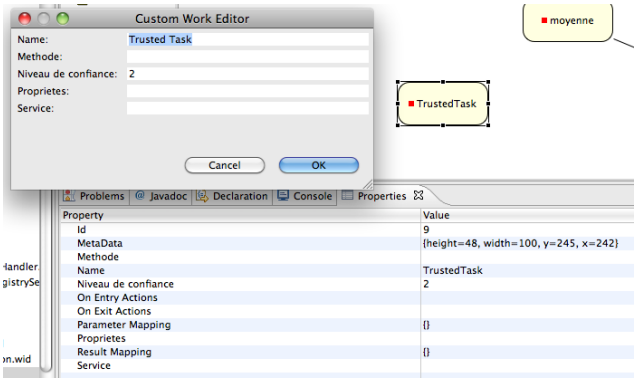

FIGURE 3.43 – Boite de dialogue et l'onglet de propriété du service abstrait TrustedTask

L'utilisation de l'orchestrateur reste identique à son utilisation dans le cas du contrôle d'accès.

2.2 Développement d'un composant dédié au maintien de la confiance

Afin de sélectionner les utilisateurs en fonction de leur niveau de confiance, il est nécessaire de mettre cette dernière à jour. La construction de modèles de confiance est un sujet de recherche en soi qui ne concerne pas directement notre approche. Nous nous inspirons du modèle proposé par [Schmidt 2007] qui repose sur les probabilités bayesiennes. À chaque fois que l'orchestrateur interagit avec un utilisateur, il met à jour le niveau de confiance de ce dernier en fonction des propriétés qui l'intéressent. Il peut, par exemple, évaluer le temps de réponse de l'utilisateur ou la pertinence de ses réponses. Le modèle repose sur les principes suivants :

- Pour chaque utilisateur, l'orchestrateur maintient un réseau de probabilités de voir les propriétés qui l'intéressent adopter les valeurs qu'ils souhaitent. Chaque réseau a pour un racine un noeud T, qui exprime la satisfaction d'une interaction.
- On note T = 1, le cas où une propriété d'intérêt adopte la valeur attendue. On note T = 0 le cas contraire. La probabilité de T=1, notée p(T=1) vaut m/n où m est le nombre d'interactions satisfaisantes et n le nombre total d'interactions. La probabilité de T=0, notée p(T=0) est donnée par 1-p(T=1). À chaque interaction, p(T=1) est ainsi mise à jour.
- À tout moment, on peut calculer la probabilité d'avoir une interaction satisfaisante à l'aide de la formule suivante, tirée des probabilités bayesiennes : p(e) exprime la probabilité du fait e, p(h) exprime la probabilité de l'hypothèse h. p(h|e) exprime la probabilité de l'hypthèse h quand e est vérifié et p(e|h) la probabilité de e quand l'hypothèse h est vérifiée.
  Si on souhaite savoir quelle est la probabilité qu'un utilisateur réponde à temps, on applique la formule de la manière suivante :

$$\Pr(h|e) = \frac{\Pr(e|h).\Pr(h)}{\Pr(e)}$$

Dès lors, on obtient : p(« réponse à temps » | T=1) = p(« réponse à temps » , T=1) / p(T=1). Dans la mesure où plusieurs propriétés peuvent être utilisées pour déterminer le niveau de confiance d'un utilisateur, son niveau de confiance global est donné par la somme pondérée des niveaux de confiance pour chaque propriété.

Afin de mettre en œuvre ce modèle de confiance, il est nécessaire d'enregistrer les interactions passées et les niveaux de confiance. Nous avons ainsi développé un service web dédié à l'exécution de la propriété d'audit. Ce composant permet de mettre à jour p(T=1) pour toutes les propriétés surveillées d'un agent. Nous avons aussi développé un service web qui joue le rôle de point de décision pour la confiance. Le point de décision calcule la probabilité d'une interaction satisfaisante avec un utilisateur. Enfin, nous avons développé une base des règles de confiance qui décrit les propriétés surveillées, leurs valeurs attendues et les pondérations de chaque propriété dans la détermination de la confiance. L'architecture globale nécessaire est présentée sur la Figure 3.44. Cette architecture s'appuie sur les composants que nous avons développés pour la spécification et l'exécution du contrôle d'accès. Comme dans ce cas, l'expert métier conçoit la vue fonctionnelle de l'orchestration quand un expert en sécurité conçoit les règles de confiance.

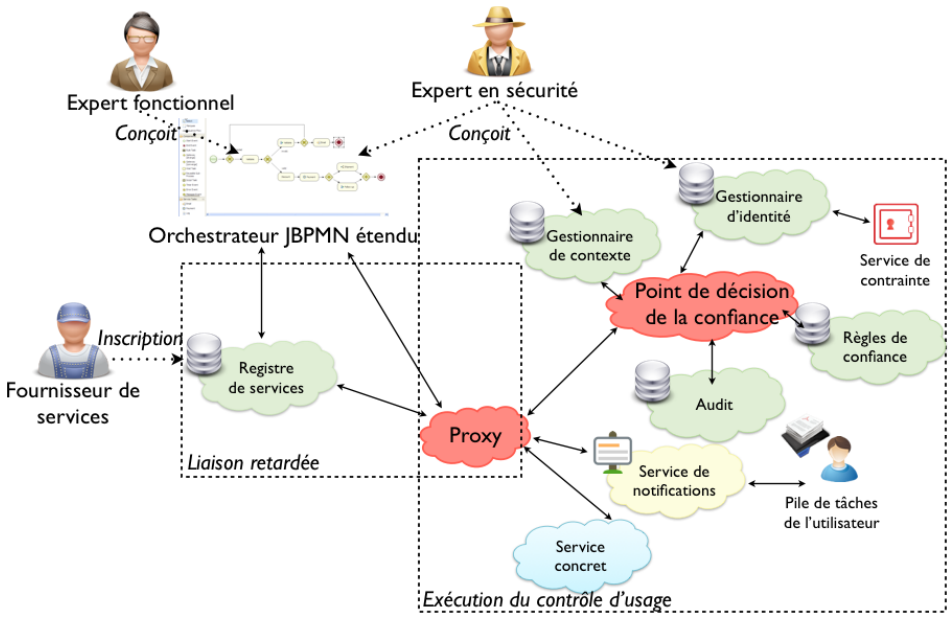

FIGURE 3.44 – Architecture pour le contrôle d'usage

C Synthèse

Nous avons présenté dans cette section les perspectives de notre travail. Les extensions les plus évidentes de notre travail sont l'ajout de nouvelles propriétés non-fonctionnelles aux orchestrations de services dynamiques et hétérogènes. Ces ajouts nous conduisent à modifier le niveau conception et le niveau exécution de notre démarche ainsi que les outils que nous avons développé.

Afin de prouver l'extensibilité de notre démarche, nous avons choisi d'ajouter le contrôle d'usage à une orchestration de services hétérogènes et dynamiques. Nous avons montré qu'en conservant le même métamodèle d'orchestration ainsi que le même procédé de génération de code, la structure de démarche s'adaptait bien à cette nouvelle propriété non-fonctionnelle. Ceci nous incline à penser que notre démarche peut s'appliquer dans les cas où il faut configurer des services à l'aide d'une propriété définie par un métamodèle propre.

Le travail de composition des métamodèles et de développement de transformations de modèles et des composants nécessaires à l'exécution est coûteux. Cependant, alors qu'habituellement les développeurs sont obligés de concevoir, pour chaque service nécessaire à l'orchestration le code qui permet de les invoquer, nous capitalisons le travail de développement dans des *templates*, ce qui permet de limiter la nécessité de la programmation manuelle.

Chapitre 4

Conclusion

I Contributions principales

Notre réflexion a pris pour point de départ l'adoption grandissante de l'approche orientée services pour réaliser des applications de plus en plus complexes. Nous avons noté, alors que nous débutions ce document que les services étaient probablement une moyen efficace de répondre aux besoins fonctionnels de nouvelles applications, comme les applications pervasives. Cependant, l'approche à services ne permet de concevoir et d'exécuter que la partie fonctionnelle de ces applications. Dans la mesure où la plupart de ces applications manipulent des données sensibles, il nous est apparu qu'il était nécessaire d'ajouter aux compositions de services des propriétés non fonctionnelles, comme le contrôle d'accès. Nous avons ainsi choisi de proposer de configurer les services afin de leur permettre d'exécuter le contrôle d'accès lorsqu'ils participent à une composition.

Cependant, la configuration se heurte à deux difficultés. Les services, surtout dans une application pervasive, sont hétérogènes car ils implémentent des technologies différentes. De plus, ils sont dynamiques. Il est ainsi impossible, à la conception d'une application, de savoir quels services seront disponibles, et donc évoqués, à l'exécution.

Afin de permettre la configuration des services tout en prenant en compte ces deux difficultés, nous avons proposé une démarche dirigée par les modèles qui permet de séparer la production des orchestrations de services hétérogènes et dynamiques sécurisées par le contrôle d'accès en deux phases :

- **Au niveau conception**, l'orchestration et la politique de contrôle d'accès qui lui est associée sont spécifiées à un niveau abstrait. Chaque préoccupation est spécifiée séparément afin de pouvoir être maintenue indépendamment et de respecter l'expertise des concepteurs qui en sont responsables. En spécifiant l'orchestration à un niveau abstrait, nous résolvons les problèmes posés par l'hétérogénéité et le dynamisme des services : il n'est pas nécessaire de savoir quels services seront disponibles à l'exécution de l'orchestration. Au niveau conception nos contributions sont :
  - Un métamodèle d'orchestration et un métamodèle de contrôle d'accès qui reprennent les vocabulaires acceptés dans chaque domaine.
  - Les liens entre ces métamodèles afin de réaliser la fusion des préoccupations.
- **Au niveau exécution**, les spécifications de l'orchestration et de sa politique de contrôle d'accès sont utilisées afin de configurer les services disponibles. Tous les services concrets sont abrités derrière un *proxy* chargé de mettre en

œuvre la politique de contrôle d'accès. Au niveau exécution, nos conceptions sont :

- Un ensemble de transformation de modèles qui permettent de tisser le code fonctionnel et le code de contrôle d'accès. Ces transformations permettent de générer le code des *proxies* chargés de protéger les services concrets.
- Une architecture d'exécution d'une orchestration de services hétérogènes et dynamiques qui permet d'exécuter le contrôle d'accès.

Chaque niveau de notre proposition a donné lieu au développement d'un ensemble d'outils. Nous avons montré sur le cas de la gestion des urgences médicales l'efficacité de notre proposition et l'utilisation de nos outils.

La Figure 4.1 présente une vue globale de nos contributions.

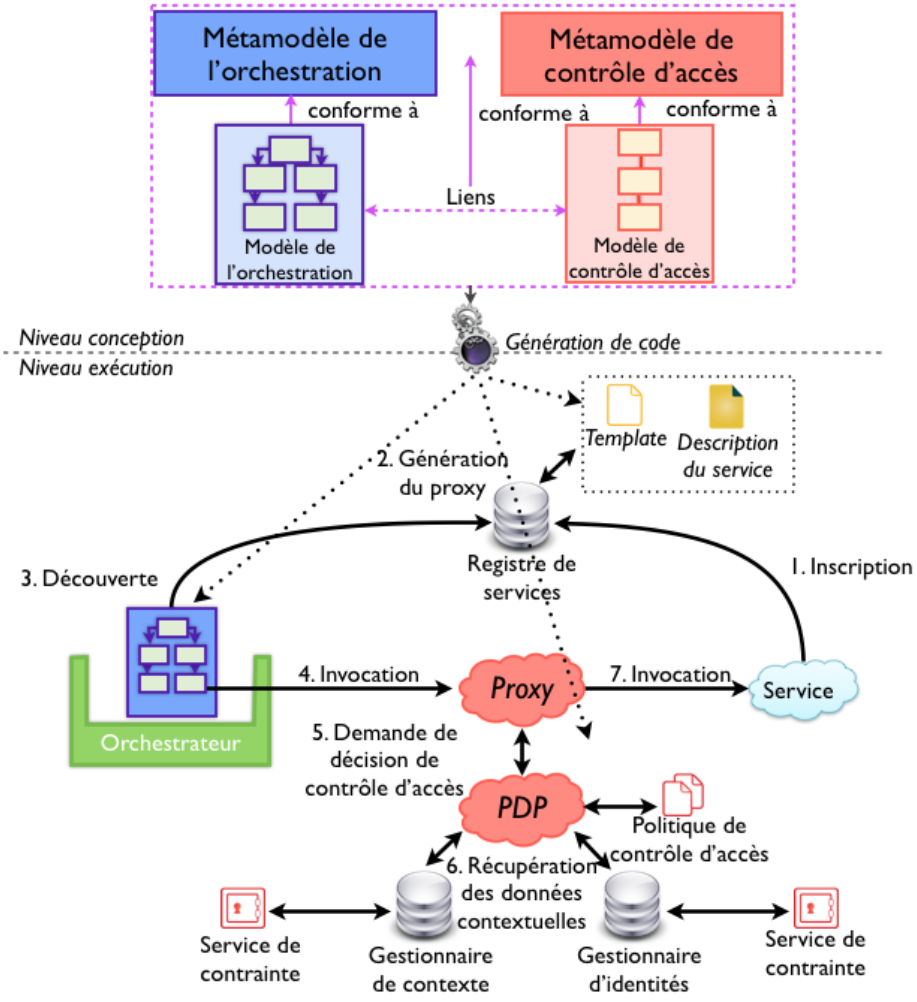

FIGURE 4.1 – Vue globale de nos contributions

II Discussion et Perspectives

Notre proposition permet de simplifier le développement d'orchestrations de services hétérogènes et dynamiques en automatisant la génération de code et en évitant ainsi le recours à la programmation manuelle. De plus, l'utilisation de templates permet de factoriser le code nécessaire et de le maintenir plus facilement. Lorsque le code de contrôle d'accès doit être modifié, seuls les templates doivent être modifiés. Ces modifications sont ensuite propagées grâce aux transformations de modèle.

Cependant, le choix de l'ingénierie dirigée par les modèles déplace les points de difficultés. La conception de métamodèles, la détermination de leurs liens et l'écriture des templates sont en effet des étapes complexes. Nous pensons néanmoins qu'il y a plus d'avantages à adopter cette démarche – le temps de développement est concentré à la conception de l'orchestration et sur quelques composants nécessaires à son exécution – plutôt que ne pas séparer les préoccupations non fonctionnelles et fonctionnelles, au risque de recourir à un développement répétitif et coûteux.

Notre démarche, comme nous l'avons montré au travers de l'exemple du contrôle d'usage, peut être étendue à d'autres propriétés non-fonctionnelles. Il faut alors identifier de nouveau les liens entre le métamodèle non-fonctionnel et le métamodèle d'une orchestration et développer les services nécessaires. L'exemple du contrôle d'usage est d'autant plus intéressant que sa mise en œuvre nécessite d'ajouter une seconde propriété non fonctionnelle à une orchestration, la propriété d'audit. Cet exemple permet de remettre en perspective notre travail avec les travaux sur la modularité et la composition des métamodèles. Il nous semble que de nombreuses investigations restent à mener sur l'ajout d'un groupe de propriétés non-fonctionnelles à une orchestration de services et sur la gestion des conflits entre elles. Notre travail constitue un premier pas dans cette direction.

Chapitre IV

Conclusion générale

Notre travail a pris pour point de départ les controverses qui entourent l'utilisation des outils numériques. Parmi les critiques qui les condamnent sans appel et celles qui vouent aux objets techniques une forme de fétichisme, les discussions qui entourent la surveillance, sa mise en œuvre et ses effets aujourd'hui, ont particulièrement retenu notre attention. Nous serions aujourd'hui observés en permanence par un grand nombre d'observateurs. Des valeurs fondamentales dans les sociétés occidentales, comme la protection de la vie privée ou l'autonomie, seraient dès lors remises en cause. Pourtant, ne sommes-nous pas, lorsque nous nous exposons sur des sites Internet par exemple, la source de cette observation ? Nous avons cherché, dans ce travail, à évaluer l'influence de la technique, et notamment l'informatique, sur nos pratiques et à proposer des moyens de protection de la vie privée dès la conception des outils informatiques..

Pour ce faire, nous avons adopté un point de vue double, celui de la philosophie et l'informatique. Notre enquête philosophique a pris la forme d'une mise en question des rapports entre technique et subjectivité. Elle nous a conduit à adopter un cadre empirique d'étude des techniques. Ce cadre nous a permis de déterminer les propriétés de notre environnement technique. Nous avons montré que nous vivions dans un espace modulable et largement surveillé. Néanmoins, si nous adhérons largement aux outils qui nous permettent de construire cet espace, nous avons montré que la limitation de la visibilité, soit sous la forme de la protection de la vie privée, soit sous la forme d'un droit à l'oubli, était nécessaire. Dans la mesure où les outils que nous utilisons influence nos possibilités de nous montrer, de voir et d'être vus, nous soutenons qu'il s'agit, pour changer le régime de visibilité dans lequel nous sommes pris, de modifier leur structure qui, dans le cas des outils informatiques, dépend de leur code.

Nous avons entrepris la modification du code dans la partie informatique de notre travail. Pour ce faire, nous avons proposé de prendre en compte dès la conception des outils des propriétés comme la protection de la vie privée afin d'en garantir le respect. Notre proposition prend la forme d'une démarche dirigée par les modèles qui divise la production des applications respectueuses de la vie privée en deux étapes. La première, la conception, permet de capturer les fonctionnalités de l'application et la politique de vie privée. La seconde, l'exécution, permet de mettre effectivement en œuvre l'application. Le passage d'un niveau à l'autre se fait de manière automatique afin de garantir la mise en œuvre de la politique de vie privée. Nous avons, au moyen d'un prototype, démontré la faisabilité de notre approche.

Au-delà des apports respectifs à chaque discipline que notre double approche

nous a permis de réaliser, l'interdépendance de notre approche philosophique et de notre étude technique nous amène à penser que le réel de la philosophie se situe, en partie au moins, dans l'étude et l'évaluation de situations concrètes. De manière corrélative, notre analyse de l'imbrication des techniques et des régimes de visibilité nous conduit à penser que le geste technique possède une dimension technique et politique. Ce sont ces dimensions qui font de l'informatique un élément central de notre condition contemporaine.

Bibliographie philosophique

[Angelis 2012] Engelico De Angelis. *The web and political actors in post-revolutionary Egypt and Tunisia† : some considerations*. Rapport technique, Cordoba Foundations, 2012.

[Anscombe 1958] Elizabeth Anscombe. *Modern Moral Philosophy*. Philosophy, vol. 33, pages 1–19, 1958.

[Anscombe 2000] G. E. M. Anscombe. *Intention*. Harvard University Press, 2000.

[Ariès 1999] Philippe Ariès et Georges Duby. *Histoire de la vie privée, tome 1 : De L'Empire romain à l'an mil*. Seuil, 1999.

[Aristote 1992] Aristote. *Ethique à Nicomaque*. Presses Pocket, 1992.

[Aristote 1999] Aristote. *Les politiques*. Flammarion, 1 1999.

[Bacchetta 2009] Paola Bacchetta. *Co-Formations : des spatialités de rÈsistance décoloniales chez les lesbiennes « of color »en France*, Novembre 2009.

[Beaude 2011] Baptiste Beaude. *De l'importance des lieux réticulaires*, 2011.

[Bentham 1977] Jeremy Bentham. *Le Panoptique*. P. Belfond, 1977.

[Borgmann 1987] Albert Borgmann. *Technology and the Character of Contemporary Life : A Philosophical Inquiry*. University Of Chicago Press, 1987.

[Bourdieu 1980] Pierre Bourdieu. *Le capital social. Notes provisoires*. Actes de la recherche en sciences sociales, vol. 31, pages 2–3, 1980.

[Brey 2000] Philip Brey. *Disclosive Computer Ethics : The Exposure and Evaluation of Embedded Normativity in Computer Technology*. Computers and Society, vol. 30, pages 10–16, 2000.

[Burbage 2004] Franck Burbage. *Foucault dans la psychanalyse Questions à Judith Butler*. Cahiers philosophiques, vol. 99, pages 110–122., 2004.

[Burke 2011] Moira Burke, Robert Kraut et Cameron Marlow. *Social capital on facebook : differentiating uses and users*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, pages 571–580, New York, NY, USA, 2011. ACM.

[Butler 2002] Judith Butler. The political : Readings in continental philosophy, chapitre What is critique ? An essay on Foucault's virtue., pages 212–229. Basil Blackwell, 2002.

[Cheng 2013] Ningning Cheng, Xinlei Wang, Prasant Mohapatra et Aruna Seneviratne. *Characterizing Privacy Leakage of Public WiFi Networks for Users on Travel*. In IEEE International Conference on Computer Communications (INFOCOM), April 2013.

[CNIL 2011] CNIL. *Loi 78-17 du 6 janvier 1978 modifiée*, 2011.

[Coontz 1988] Stephanie Coontz. *The social Origins of Private Life. A History of American Families (1600-1900)*. Verso, New York, 1988.

[danah boyd 2008] danah boyd. Structures of participation in digital culture, chapitre None of this is Real, pages 132–157. Social Science Research Council, New York, 2008.

[de Bussher 1998] Pierre Olivier de Bussher, Rommel Mendès-Leite et Bruno Proth. *Lieux de rencontre et backrooms*. Actes de la recherche en sciences sociales, vol. 128, pages 24–28, 1998.

[de Tocqueville 1986] Alexis de Tocqueville. *De la démocratie en Amérique, tome 2*. Gallimard, 1986.

[Deleuze 1977] Gilles Deleuze et Claire Parnet. *Dialogues*. Columbia Univ Press, 1977.

[Deleuze 1986] Gilles Deleuze. *Foucault*. Editions de Minuit, 1986.

[Deleuze 2003] Gilles Deleuze. *Pourparlers 1972-1990*. Editions de minuit, 2003.

[Deleuze 2010] Gilles Deleuze. *Différence et répétition*. PUF, 2010.

[Derrida 1979] Jacques Derrida. *L'écriture et la différence*. Editions du Seuil, 1979.

[Dreyfus 2001] Hubert Dreyfus. *On the Internet*. Routledge, 2001.

[Ellul 2004] Jacques Ellul. *le système technicien*. Cherche midi, 2004.

[Ess 1996] Charles Ess. Philosophical perspectives on computer-mediated communication, chapitre The Political Computer : Democracy, CMC and Habermas, pages 197–230. SUNY Press, 1996.

[Euclide 1990] Euclide. *Les elements*. PUF, 1990.

[Flichy 1997] Patrice Flichy. *Une histoire de la communication moderne*. La Découverte, 1997.

[Ford 2013] Henry Ford. *My Life and Work*. CreateSpace Independent Publishing Platform, 2013.

[Foucault 1976a] Michel Foucault. *Histoire De La Folie à L'âge Classique*. Gallimard, 1976.

[Foucault 1976b] Michel Foucault. *Histoire de la sexualité. 1 - La volonté de savoir*. Paris : Gallimard, 1976.

[Foucault 1984a] M. Foucault. *Histoire de la sexualité 3 Le souci de soi*. Gallimard, 1984.

[Foucault 1984b] Michel Foucault. *L'éthique du souci de soi comme pratique de la liberté (entretien avec H. Becker, R. Fornet-Betancourt, A. Gomez-Muller)*. Concordia. Revista internacional de filosofia, vol. 6, pages 99–116, 1984.

[Foucault 1990a] Michel Foucault. *Les Mots et les choses*. Gallimard, 1 1990.

[Foucault 1990b] Michel Foucault. *Qu'est-ce que la critique ?* Bulletin de la Société française de philosophie, vol. 2, pages 35–63, 1990.

[Foucault 1993] Michel Foucault. *Surveiller et punir*. Gallimard, 1993.

[Foucault 1994] Michel Foucault. *Dits et Ècrits 1954-1988, tome 4 : 1980-1988*. Gallimard, 10 1994.

[Foucault 1996] Michel Foucault. *Herculine Barbin.* Messageries du Livre, 1996.

[Foucault 2001] Michel Foucault. Dits et ecrits, tome 2 : 1976 - 1988, chapitre Qu'est-ce que les lumières ?, pages 1498–1507. Gallimard, 2001.

[Foucault 2004] Michel Foucault. *Sécurité, territoire, population : Cours au Collège de France (1977-1978).* Seuil, Paris, 10 2004.

[Fraser 1985] Nancy Fraser. *Michel Foucault : A "Young Conservative" ?* Ethics, vol. 8596, pages 165–184, 1985.

[Ganascia 2009] Jean-Gabriel Ganascia. *Voir et pouvoir : qui nous surveille ?* Editions le Pommier, 2009.

[Gerrie 2007] Jim Gerrie. *Was Foucault a Philosopher of Technology ?* Techné : Research in Philosophy and Technology, vol. 11, page 1, 2007.

[Gille 1978] Bertrand Gille. *Histoire des Techniques.* Gallimard, 1978.

[Grimes 2008] Andrea Grimes et A. J. Bernheim Brush. *Life scheduling to support multiple social roles.* In CHI, pages 821–824, 2008.

[Habermas 1991] Jürgen Habermas. *The Structural Transformation of the Public Sphere : An Inquiry into a Category of Bourgeois Society (Studies in Contemporary German Social Thought).* The MIT Press, 1991.

[Hamilton 1996] Richard F. Hamilton. *The Social Misconstruction of Reality : Validity and Verification in the Scholarly Community.* Yale University Press, 1996.

[Heidegger 1958] Martin Heidegger. *Essais et conférences.* Gallimard, 1958.

[Hume 1985] David Hume. *Essays : Moral, Political, and Literary.* Liberty Fund, 1985.

[Huxley 2010] Aldous Huxley. *Le Meilleur Des Mondes.* Distribooks, 2010.

[Jauréguiberry 1997] Francis Jauréguiberry. *L'usage du téléphone portatif comme expérience sociale.* Réseaux, vol. 15, pages 149–165, 1997.

[Jeremy Ginsberg 2009] Rajan S. Patel Lynnette Brammer Mark S. Smolinski & Larry Brilliant Jeremy Ginsberg Matthew H. Mohebbi. *Detecting influenza epidemics using search engine query data.* Nature, vol. 457, pages 1012–1014, 2009.

[Kant 1988] Emmanuel Kant. *Theorie Et Pratique. Sur Un Pretendu Droit De Mentir Par Humanite (Bibliotheque Des Textes Philosophiques) (French Edition).* Vrin, 1988.

[Kessous 2013] Emmanuel Kessous. *L'attention au monde : Sociologie des données personnelles à l'ère numérique.* Armand Colins, 2013.

[Krumm 2007] John Krumm et Eric Horvitz. *Predestination : Where Do You Want to Go Today ?* Computer, vol. 40, no. 4, pages 105–107, 2007.

[Kulynych 1997] Jessica J. Kulynych. *Performing Politics.* Polity, vol. 30, pages 315–346, 1997.

[Lessig 2000] Lawrence Lessig. *Code and Other Laws of Cyberspace*. Basic Books, 2000.

[Levinas 1961] Emmanuel Levinas. *Totalité et Infini : Essai Sur L'exteriorite*. Martinus Nijhoff, 1961.

[Levy 1998] Levy. *Qu'est ce que le virtuel ?* La Découverte, 1998.

[Levy 2004] Neil Levy. *Foucault as a virtue ethicist*. Foucault Studies, vol. 1, pages 20–31, 2004.

[MacIntyre 1991] Alasdair MacIntyre. *Three Rival Versions of Moral Enquiry : Encyclopaedia, Genealogy, and Tradition*. University of Notre Dame Press, 8 1991.

[MacIntyre 1997] Alasdair MacIntyre. *Aprés la vertu : Etude de thÈorie morale*. PUF, 1997.

[MacKinnon 1991] Catharine A. MacKinnon. *Toward a Feminist Theory of the State*. Harvard University Press, 1991.

[Marcuse 1989] Marcuse. *L'Homme unidimensionnel : Essai sur l'idéologie de la société industrielle avancée*. Editions de Minuit, 1989.

[MaxDowell 1979] John MaxDowell. *Virtue and Reason*. Monist, vol. 62, pages 331–350, 1979.

[McLuhan 1967] Marshall McLuhan et Quentin Fiore. *The Medium is the Massage : An Inventory of Effects*. Bantam Books, 1st édition, 1967.

[Métayer 2013] Daniel Le Métayer. *Privacy by design : a formal framework for the analysis of architectural choices*. In CODASPY, pages 95–104, 2013.

[Moor 1985] James H. Moor. *What is computer ethics ?* Metaphilosophy, vol. 16, pages 266–75, 1985.

[Morel 1996] Pierre Morel et Claude Quétel. *Les medecines de la folie*. Hachette LittÈrature, 1996.

[Moreno 1970] Jacob Levy Moreno. *Fondements de la sociomÈérie*. PUF, 1970.

[Nath 2006] Shyam Varan Nath. *Crime Pattern Detection Using Data Mining*. In LWA, pages 338–341, 2006.

[Nissenbaum 1998] Helen Nissenbaum. *Protecting Privacy in an Information Age : The Problem of Privacy in Public*. Law and Philosophy, vol. 17, no. 5/6, pages 559–596, 1998.

[Orwell 1961] George Orwell. *1984*. Signet Classic, 1961.

[Pizzorno 1989] Alessandro Pizzorno. Michel foucault philosophe : Rencontre internationale paris, 9, 10, 11 janvier 1988, chapitre Foucault et la conception libérale de l'individu, pages 236–248. Seuil, 1989.

[Rachels 1975] James Rachels. *Why Privacy is important*. Philosophy and Public Affairs, vol. 4, no. 4, pages 323–333., 1975.

[Rawls 1999] John Rawls. *A Theory of Justice*. Belknap Press of Harvard University Press, revised edition édition, 9 1999.

[Ricoeur 2000] Paul Ricoeur. *La mémoire, l'histoire, l'oubli.* Seuil, 2000.

[Scanlon 1975] Thomas Scanlon. *Thomson on privacy.* Philosophy and Public Affairs, vol. 4, pages 315–322, 1975.

[Schmidt 2013] Eric Schmidt et Jared Cohen. *The New Digital Age : Reshaping the Future of People, Nations and Business.* Knopf, 4 2013.

[Simon 1991] Herbert A. Simon. *Bounded Rationality and Organizational Learning.* Organization Science, vol. 2, pages 125–134, 1991.

[Steinbuch 1957] Karl Steinbuch. *Informatik : Automatische Informationsverarbeitung.* SEL-Nachrichten, vol. 4, page 171, 1957.

[Sweeney 2002] Latanya Sweeney. *k-Anonymity : A Model for Protecting Privacy.* International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 5, pages 557–570, 2002.

[UNESCO 1997] UNESCO. *Déclaration universelle sur le génome humain et les droits de l'homme*, 1997.

[Vallor 2010] Shannon Vallor. *Social Networking Technology and the Virtues.* Ethics and Information Technology, vol. 12, pages 157–170, 2010.

[van Wel 2004] Lita van Wel et Lambèr Royakkers. *Ethical issues in web data mining.* Ethics and Information Technology, vol. 6, pages 129–140, 2004.

[Varenne 2009] Franck Varenne. *Qu'est-ce que l'informatique ?* VRIN, 2009.

[Virilio 1977] Paul Virilio. *Vitesse et politique.* Galilée, 1977.

[Virilio 2002] Paul Virilio. *The visual crash.* MIT Press, 2002.

[Waren 1890] Samuel Waren et Louis Brandeis. *The right to privacy.* Harvard Law Review, vol. 4, pages 193–220, 1890.

[Warnier 2010] Jean Pierre Warnier. *Foucault en Afrique. La microphysique d'une monarchie africaine.* Revue internationale des sciences sociales, vol. 191, pages 103–112., 2010.

[Willcocks 2006] Leslie P. Willcocks. *Michel Foucault in the Social Study of ICTs Critique and Reappraisal.* Social Science Computer Review, vol. 24, pages 274–295, 2006.

Bibliographie informatique

[Afrati 2003] Foto Afrati, Theodore Andronikos, Vassia Pavlaki, Eugenie Foustoucos et Irène Guessarian. *From CTL to datalog.* In PCK50, PCK50, pages 72–85, New York, NY, USA, 2003. ACM.

[Agrawal 2002] R Agrawal, J Kiernan, R Srikant et Y Xu. *Hippocratic databases.* In VLDB, pages 143–154, 2002.

[Atkinson 2003] Colin Atkinson et Thomas Kühne. *Model-Driven Development : A Metamodeling Foundation.* IEEE Software, vol. 20, no. 5, pages 36–41, 2003.

[Atluri 1999] V Atluri, S Jajodia et B George. *Multilevel Secure Transaction Processing.* Kluwer A, 1999.

[Banerjee 2011] Prith Banerjee, R. Friedrich, Cullen Bash, P. Goldsack, Bernardo A. Huberman, J. Manley, C. Patel, Parthasarathy Ranganathan et A. Veitch. *Everything as a Service : Powering the New Information Economy.* IEEE Computer, vol. 44, no. 3, pages 36–43, 2011.

[Basin 2006] David Basin, Jürgen Doser et Torsten Lodderstedt. *Model driven security : From UML models to access control infrastructures.* ACM Trans. Softw. Eng. Methodol., vol. 15, pages 39–91, 2006.

[Bell 1973] David Elliott Bell. *Secure computer system : A refinement of the Mathematical Model.* Rapport technique, The Mitre Corporation, Bedford, MA, 1973.

[Beugnard 1999] Antoine Beugnard, Jean-Marc Jézéquel et Noël Plouzeau. *Making Components Contract Aware.* IEEE Computer, vol. 32, no. 7, pages 38–45, 1999.

[Bézivin 2005] Jean Bézivin. *On the unification power of models.* Software and System Modeling, vol. 4, no. 2, pages 171–188, 2005.

[Bianco 2008] Philip Bianco, Grace Lewis et Paulo Merson. *Service Level Agreements in Service-Oriented Architecture Environments.* Rapport technique, Software Engineering Institute, Carnegie Mellon University, 2008.

[Biba 1977] Kenneth J. Biba. *Integrity Considerations for sexure computer systems.* Rapport technique, The Mitre Corporation, 1977.

[Bottaro 2007] André Bottaro, Anne Gérodolle et Philippe Lalanda. *Pervasive Service Composition in the Home Network.* In AINA, pages 596–603, 2007.

[Boudol 2009] Gérard Boudol. *Formal Aspects in Security and Trust.* chapitre Secure Information Flow as a Safety Property, pages 20–34. Springer-Verlag, Berlin, Heidelberg, 2009.

[Brewer 1989] D. F. C. Brewer et M. J. Nash. *The Chinese Wall Security Policy.* In IEEE Symposium on Security and Privacy, pages 206–214, 1989.

[Callaway 2008] Robert David Callaway. *An autonomic service delivery platform for service-oriented network environments.* PhD thesis, Computer Engineering Raleigh, North Carolina, 2008. AAI3306558.

[Carminati 2006] Barbara Carminati, Elena Ferrari et Patrick Hung. *Security Conscious Web Service Composition.* In ICWS, pages 489–496, 2006.

[Cervantes 2004] Humberto Cervantes et Richard S. Hall. *Autonomous Adaptation to Dynamic Availability Using a Service-Oriented Component Model.* In ICSE, pages 614–623, 2004.

[Chae 2012] Heemoon Chae, Jooik Jung, Jong-Hyuk Lee et Kyong-Ho Lee. *An efficient access control based on role attributes in service oriented environments.* In ICUIMC, page 73, 2012.

[Chollet 2008] Stéphanie Chollet et Philippe Lalanda. *Security Specification at Process Level.* SCC, vol. 1, pages 165–172, 2008.

[Chollet 2009a] Stéphanie Chollet. *Orchetsration de services hétérogènes et sécurisés.* PhD thesis, Université Joseph Fourier, 2009.

[Chollet 2009b] Stéphanie Chollet et Philippe Lalanda. *An Extensible Abstract Service Orchestration Framework.* In ICWS, pages 831–838, 2009.

[Chollet 2011] Stéphanie Chollet, Vincent Lestideau, Philippe Lalanda, Yoann Maurel, Pierre Colomb et Olivier Raynaud. *Building FCA-Based Decision Trees for the Selection of Heterogeneous Services.* In IEEE SCC, pages 616–623, 2011.

[Covington 2001] Michael J. Covington, Wende Long, Srividhya Srinivasan, Anind K. Dev, Mustaque Ahamad et Gregory D. Abowd. *Securing context-aware applications using environment roles.* In SACMAT '01, pages 10–20, 2001.

[Cuppens 2008] Frederic Cuppens et Nora Cuppens. *Modeling contextual security policies.* International Journal of Information Security (IJIS), vol. 7, no. 4, pages 285 – 305, august 2008.

[Dami 1998] Samir Dami, Jacky Estublier et Mahfoud Amiour. *Apel : A Graphical Yet Executable Formalism for Process Modeling.* Autom. Softw. Eng., vol. 5, no. 1, pages 61–96, 1998.

[Denker 2010] Marcus Denker, Jorge Ressia, Orla Greevy et Oscar Nierstrasz. *Modeling Features at Runtime.* In MoDELS (2), pages 138–152, 2010.

[Denning 1976] Dorothy E. Denning. *A lattice model of secure information flow.* Commun. ACM, vol. 19, no. 5, pages 236–243, Mai 1976.

[Desmedt 2011] Yvo Desmedt. *Trojan Horses, Computer Viruses, and Worms.* In Encyclopedia of Cryptography and Security (2nd Ed.), pages 1319–1320. Sp, 2011.

[Escoffier 2007] Clément Escoffier et Richard S. Hall. *Dynamically Adaptable Applications with iPOJO Service Components.* In Software Composition, pages 113–128, 2007.

[Frénot 2012] Stéphane Frénot et Julien Ponge. *LogOS : an Automatic Logging Framework for Service-Oriented Architectures.* In 38th Euromicro Conference on Software Engineering and Advanced Applications, Izmir, Turkey, Septembre 2012.

[Fuggetta 1993] Alfonso Fuggetta. *A Classification of CASE Technology.* IEEE Computer, vol. 26, no. 12, pages 25–38, 1993.

[Gama 2009] Kiev Gama et Didier Donsez. *Towards Dynamic Component Isolation in a Service Oriented Platform.* In CBSE, pages 104–120, 2009.

[García Frey 2010] Alfonso García Frey. *Self-explanatory user interfaces by model-driven engineering.* In Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems, EICS '10, pages 341–344, New York, NY, USA, 2010. ACM.

[Goguen 1984] Joseph A. Goguen et José Meseguer. *Unwinding and Inference Control.* In IEEE Symposium on Security and Privacy, pages 75–87, 1984.

[Gross-Amblard 2011] David Gross-Amblard. *Query-preserving watermarking of relational databases and Xml documents.* ACM Trans. Database Syst., vol. 36, no. 1, page 3, 2011.

[Guruge 2004] Anura Guruge. *Web Services : Theory and Practice.* Digital Press, 1 édition, 3 2004.

[Harrison 1976] Michael A. Harrison, Walter L. Ruzzo et Jeffrey D. Ullman. *Protection in operating systems.* Commun. ACM, vol. 19, no. 8, pages 461–471, Ao ˚ t 1976.

[Hély 2007] David Hély, Frédéric Bancel, Marie-Lise Flottes et Bruno Rouzeyre. *Securing Scan Control in Crypto Chips.* J. Electronic Testing, vol. 23, no. 5, pages 457–464, 2007.

[Hengartner 2006] Urs Hengartner et Peter Steenkiste. *Avoiding Privacy Violations Caused by Context-Sensitive Services.* In Proceedings of the 4TH IEEE International conference on pervasive computing and communications, PERCOM 2006, pages 222–231, 2006.

[Hung 2007] Patrick C. K. Hung et Yi Zheng. *Privacy Access Control Model for Aggregated e-Health Services.* In EDOCW, pages 12–19, 2007.

[Jafarian 2008] Jafar Haadi Jafarian, Morteza Amini et Rasool Jalili. *A Context-Aware Mandatory Access Control Model for Multilevel Security Environments.* In Proceedings of the 27th international conference on Computer Safety, Reliability, and Security, SAFECOMP '08, pages 401–414, Berlin, Heidelberg, 2008. Springer-Verlag.

[Jajodia 1991] Sushil Jajodia et Ravi S. Sandhu. *Towards a Multilevel Secure Relational Data Model.* In SIGMOD Conference, pages 50–59, 1991.

[Jajodia 2004] Sushil Jajodia et Duminda Wijesekera. *A flexible authorization framework for e-commerce.* In Proceedings of the First international conference

on Distributed Computing and Internet Technology, ICDCIT'04, pages 336–345, Berlin, Heidelberg, 2004. Springer-Verlag.

[Jajodia 2007] Sushil Jajodia et Ting Yu. *Basic Security Concepts*. In Secure Data Management in Decentralized Systems, pages 3–20. 2007.

[Jensen 2011] Jostein Jensen et Martin Gilje Jaatun. *Security in Model Driven Development : A Survey*. In Proceedings of the 2011 Sixth International Conference on Availability, Reliability and Security, ARES '11, pages 704–709, Washington, DC, USA, 2011. IEEE Computer Society.

[Jouault 2008] Frédéric Jouault, Freddy Allilaire, Jean Bézivin et Ivan Kurtev. *ATL : A model transformation tool*. Sci. Comput. Program., vol. 72, no. 1-2, pages 31–39, 2008.

[Kleiner 2007] E. Kleiner et T. Newcomb. *On the Decidability of the Safety Problem for Access Control Policies*. Electr. Notes Theor. Comput. Sci., vol. 185, pages 107–120, 2007.

[Kohler 2007] Mathias Kohler, Christian Liesegang et Andreas Schaad. *Classification Model for Access Control Constraints*. In IPCCC, pages 410–417, 2007.

[Kuhn 2010] D. Richard Kuhn, Edward J. Coyne et Timothy R. Weil. *Adding Attributes to Role-Based Access Control*. IEEE Computer, vol. 43, no. 6, pages 79–81, 2010.

[Kulkarni 2008] Devdatta Kulkarni et Anand Tripathi. *Context-aware role-based access control in pervasive computing systems*. In Proceedings of the 13th ACM symposium on Access control models and technologies, SACMAT '08, pages 113–122, NY, USA, 2008.

[Kumar 2002] Arun Kumar, Neeran Karnik et Girish Chafle. *Context sensitivity in role-based access control*. SIGOPS Oper. Syst. Rev., vol. 36, no. 3, pages 53–66, 2002.

[Li 2007] Jun Li et Alan H. Karp. *Access control for the services oriented architecture*. In SWS, pages 9–17, 2007.

[Mao 2009] Ziqing Mao, Ninghui Li, Hong Chen et Xuxian Jiang. *Trojan horse resistant discretionary access control*. In SACMAT, pages 237–246, 2009.

[Marin 2006] Cristina Marin et Philippe Lalanda. *A domain-specific service-oriented development environment*. In IEEE SCC, pages 307–310, 2006.

[Marin 2008] Cristina Marin. *Une approche orientée domaine pour la composition de services*. PhD thesis, Université Joseph Fourier, 2008.

[Maurel 2011] Yoann Maurel, Philippe Lalanda et Ada Diaconescu. *Towards a Service-Oriented Component Model for Autonomic Management*. In IEEE SCC, pages 544–551, 2011.

[McCollum 1990] Catherine D. McCollum, J. R. Messing et LouAnna Notargiacomo. *Beyond the Pale of MAC and DAC-Defining New Forms of Access Control*. In IEEE Symposium on Security and Privacy, pages 190–200, 1990.

[Minami 2010] Kazuhiro Minami et Nikita Borisov. *Protecting location privacy against inference attacks.* In ACM Conference on Computer and Communications Security, pages 711–713, 2010.

[OASIS 2003] OASIS. *A brief introduction to XACML.* http ://docs.oasis-open.org/xacml/3.0, 2003.

[OASIS 2010] OASIS. *XACML v3.0 Core and Hierarchical Role Based Access Control (RBAC) Profile Version 1.0.* http ://docs.oasis-open.org/xacml/3.0/xacml-3.0-rbac-v1-spec-cs-01-en.pdf, 2010.

[Orriëns 2003] Bart Orriëns, Jian Yang et Mike P. Papazoglou. *Model driven service composition.* In ICSOC 2003, pages 75–90. Springer-Verlag, 2003.

[Papazoglou 2003] Mike P. Papazoglou. *Service -Oriented Computing : Concepts, Characteristics and Directions.* In Proceedings of the Fourth International Conference on Web Information Systems Engineering, WISE '03, pages 3–, Washington, DC, USA, 2003. IEEE Computer Society.

[Pareschi 2008] Linda Pareschi, Daniele Riboni, Alessandra Agostini et Claudio Bettini. *Composition and Generalization of Context Data for Privacy Preservation.* In PerCom, pages 429–433, 2008.

[Pedersen 2011] Torben Pedersen. *HTTPS, HTTP over TLS.* In Encyclopedia of Cryptography and Security (2nd Ed.), pages 569–570. 2011.

[Perwej 2012] Yusuf Perwej, Firoj Parwej et Asif Perwej. *An Adaptive Watermarking Technique for the copyright of digital images and Digital Image Protection.* CoRR, vol. abs/1205.2800, 2012.

[Phillips 2003] Charles E. Phillips, Steven A. Demurjian et T. C. Ting. *Safety and Liveness for an RBAC/MAC Security Model.* In DBSec, pages 316–329, 2003.

[Ray 2006] Indrakshi Ray et Mahendra Kumar. *Towards a location-based mandatory access control model.* Computers & Security, vol. 25, pages 10–1016, 2006.

[Rodríguez 2007] Alfonso Rodríguez, Eduardo Fernández-Medina et Mario Piattini. *A BPMN Extension for the Modeling of Security Requirements in Business Processes.* IEICE, vol. E90-D, no. 4, pages 745–752, 2007.

[Samarati 2000] Pierangela Samarati et Sabrina De Capitani di Vimercati. *Access Control : Policies, Models, and Mechanisms.* In FOSAD, pages 137–196, 2000.

[Sandhu 1992] Ravi S. Sandhu. *The Typed Access Matrix Model.* In Proceedings of the 1992 IEEE Symposium on Security and Privacy, SP '92, pages 122–, Washington, DC, USA, 1992. IEEE Computer Society.

[Sandhu 1998] Ravi S. Sandhu et Qamar Munawer. *How to Do Discretionary Access Control Using Roles.* In ACM Workshop on Role-Based Access Control, pages 47–54, 1998.

[Schmeling 2012] Benjamin Schmeling, Anis Charfi, Marko Martin et Mira Mezini. *Towards Conflict-Free Composition of Non-functional Concerns.* In CAiSE, pages 80–94, 2012.

[Schmidt 2007] Stefan Schmidt, Robert Steele, Tharam S. Dillon et Elizabeth Chang. *Fuzzy trust evaluation and credibility development in multi-agent systems.* Applied Soft Computing, vol. 7, no. 2, pages 492 – 505, 2007.

[Seinturier 2012] Lionel Seinturier, Philippe Merle, Romain Rouvoy, Daniel Romero, Valerio Schiavoni et Jean-Bernard Stefani. *A Component-Based Middleware Platform for Reconfigurable Service-Oriented Architectures.* Software : Practice and Experience, vol. 42, no. 5, pages 559–583, Mai 2012.

[Shen 1992] Honghai Shen et Prasun Dewan. *Access Control for Collaborative Environments.* In CSCW, pages 51–58, 1992.

[Shen 2006] Hai-bo Shen et Fan Hong. *An Attribute-Based Access Control Model for Web Services.* In Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT '06, pages 74–79, Washington, DC, USA, 2006. IEEE Computer Society.

[Siponen 2002] Mikko T. Siponen. *Database Security and the Problem of Polyinstantiation : a moral scrutiny.* Australasian J. of Inf. Systems, vol. 10, no. 1, 2002.

[Sirer 2002] Emin Gün Sirer et Ke Wang. *An access control language for web services.* In SACMAT, pages 23–30, 2002.

[Spiekermann 2012] Sarah Spiekermann. *The challenges of privacy by design.* Commun. ACM, vol. 55, no. 7, pages 38–40, 2012.

[Srivatsa 2007] Mudhakar Srivatsa, Arun Iyengar, Thomas A. Mikalsen, Isabelle Rouvellou et Jian Yin. *An Access Control System for Web Service Compositions.* In ICWS, pages 1–8, 2007.

[Sward 2011] Ricky E. Sward et Jeff Boleng. *Service-oriented architecture (SOA)concepts and implementations.* In SIGAda, pages 3–4, 2011.

[Sweeney 2002] Latanya Sweeney. *k-Anonymity : A Model for Protecting Privacy.* International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 5, pages 557–570, 2002.

[Szyperski 2002] Clemens Szyperski. *Component Software : Beyond Object-Oriented Programming.* Addison-Wesley Longman Publishing Co, 2002.

[Szyperski 2003] Clemens A. Szyperski. *Component Technology - What, Where, and How ?* In ICSE, pages 684–693, 2003.

[Thion 2005] Romuald Thion et Stéphane Coulondre. *Intégration du contexte spatio-temporel dans le contrôle d accès basé sur les rôles.* Revue des Sciences et Technologies de l'Information, série Ingénierie des Systèmes d'Information, vol. 10, no. 4, pages 89–117, 2005.

[Thomas 1998] Roshan K. Thomas et Ravi S. Sandhu. *Task-Based Authorization Controls : A Family of Models for Active and Enterprise-Oriented Autorization Management.* In Proceedings of the Eleventh International Conference on Database Security, pages 166–181, 1998.

[Vallecillo 2010] Antonio Vallecillo. *On the Combination of Domain Specific Modeling Languages*. In ECMFA, pages 305–320, 2010.

[Wolter 2007a] Christian Wolter et Andreas Schaad. *Modeling of task-based authorization constraints in BPMN*. In BPM'07, pages 64–79, Berlin, Heidelberg, 2007. Springer-Verlag.

[Wolter 2007b] Christian Wolter, Andreas Schaad et Christoph Meinel. *Deriving XACML policies from business process models*. In WISE'07, pages 142–153, Berlin, Heidelberg, 2007. Springer-Verlag.

[Yeh 2011] Lo-Yao Yeh, Yen-Cheng Chen et Jiun-Long Huang. *ABACS : An Attribute-Based Access Control System for Emergency Services over Vehicular Ad Hoc Networks*. IEEE Journal on Selected Areas in Communications, vol. 29, no. 3, pages 630–643, 2011.

[Yuan 2005] Eric Yuan et Jin Tong. *Attributed Based Access Control (ABAC) for Web Services*. In Proceedings of the IEEE International Conference on Web Services, ICWS '05, pages 561–569, Washington, DC, USA, 2005. IEEE Computer Society.