



HAL
open science

Applications communautaires spontanées dynamiquement reconfigurables en environnement pervasif

Ghada Ben Nejma

► **To cite this version:**

Ghada Ben Nejma. Applications communautaires spontanées dynamiquement reconfigurables en environnement pervasif. Autre [cs.OH]. UPPA, 2015. Français. NNT: . tel-02437272

HAL Id: tel-02437272

<https://univ-pau.hal.science/tel-02437272v1>

Submitted on 13 Jan 2020

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

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

UNIVERSITE DE PAU ET DES PAYS DE L'ADOUR

ECOLE DOCTORALE DES SCIENCES EXACTES ET LEURS APPLICATIONS



THESE DE DOCTORAT EN INFORMATIQUE

Présentée et soutenue par

Ghada BEN NEJMA

Le 22 Décembre 2015

**Applications communautaires spontanées
dynamiquement reconfigurables en environnement pervasif**

Thèse dirigée par :

Philippe ROOSE

Marc DALMAU

Jérôme GENSEL

JURY :

RÉSUMÉ

Depuis quelques années, des évolutions importantes ont lieu en matière d'infrastructures technologiques. En particulier, la démocratisation des dispositifs mobiles (comme les *PCs*, *Smartphones*, *Tablettes*, etc.) a rendu *l'information accessible* par le grand public partout et à tout moment, ce qui est l'origine du concept d'informatique ubiquitaire. L'approche classique des systèmes de l'informatique ubiquitaire, qui répondent aux besoins des utilisateurs indépendants les uns des autres, a été bouleversée par l'introduction de la dimension sociale. Ce rapprochement est à l'origine d'une discipline naissante « *le pervasive social computing* » ou l'informatique socio-pervasive. Les applications socio-pervasive connaissent une véritable expansion. Ces dernières intègrent de plus en plus la notion de *communauté*. Le succès des applications communautaires se justifie par le but poursuivi par ces dernières de répondre aux besoins des communautés et d'offrir un 'chez soi' virtuel, spécifique à la communauté, dans lequel elle va construire sa propre identité et réaliser ses objectifs. Également, la notion de communauté représente une source d'informations contextuelles sociales. Elle est, aujourd'hui, au cœur des problématiques de personnalisation et d'adaptation des applications informatiques.

Dans le cadre de cette thèse, nous étudions d'abord les applications informatiques centrées communautés existantes sous différents aspects et nous relevons plusieurs lacunes au niveau de la notion de communauté, les modèles de communautés, l'architecture des applications communautaires, etc. Pour remédier à ces défauts, nous proposons trois principales contributions :

- Un nouveau type de communauté adapté aux exigences des environnements pervasifs qui vient rompre avec les traditionnelles communautés pérennes thématiques : des communautés éphémères, géolocalisées et spontanées (sans contrainte thématique).
- Un modèle de communauté basé sur les standards du web sémantique pour répondre aux problèmes liés à l'hétérogénéité de conception des communautés.
- Une architecture dynamiquement reconfigurable pour promouvoir les communautés spontanées en aidant les utilisateurs nomades à intégrer des communautés environnantes et à découvrir les services dédiés.

Nous montrons la faisabilité de nos propositions pour la conception et le développement d'applications communautaires spontanées grâce au prototype Taldea. Enfin, nous testons les approches proposées de découverte de communauté et de services à travers plusieurs scénarios caractérisés par la mobilité et l'ubiquité.

Mots clés : Application communautaire ; *Pervasive social computing* ; Technologies du web sémantique ; Découverte de services ; Sensibilité au contexte.

ABSTRACT

Sommaire

RÉSUMÉ.....	3
ABSTRACT	4
Sommaire	5
Table des figures	8
Table des tableaux	10
Table des équations	11
Chapitre 1: Introduction	13
1.1 Contexte	13
1.2 Problématiques défendues et contributions	15
1.2.1 Notion de communauté	15
1.2.2 Modèle de communautés	16
1.2.3 Architecture des applications communautaires	17
1.3 Organisation du mémoire	18
Chapitre 2 : Les communautés : cadre théorique et applications	25
2.1 Cadre théorique	25
2.1.1 Définitions de communauté	25
2.1.2 Caractéristiques d'une communauté	26
2.1.3 Typologie des communautés	29
2.2 Applications communautaires : est-ce l'informatique au service des communautés ou bien l'inverse ?	33
2.2.1 Les applications informatiques comme support à l'expansion des communautés	33
2.2.2 La communauté, un facteur de progrès pour les applications informatiques	35
2.3 Limites des applications centrées communauté	39
2.4 Conclusion.....	42
Chapitre 3 : Le Social Computing comme support des communautés	45
3.1 Fondements théoriques.....	46
3.2 Technologies : de la sémantique pour le Social Computing	47
3.2.1 Représentation sémantique des connaissances	48
3.2.2 Ontologies pour le social computing	49
3.3 Les réseaux sociaux : un exemple d'application du <i>Social Computing</i>	56
3.3.1 Les réseaux sociaux grand public	57
3.3.2 Les réseaux sociaux d'entreprise	57
3.3.3 Les réseaux sociaux basés sur la localisation	58
3.3.4 Les réseaux sociaux éphémères	59
3.3.5 Architecture des réseaux sociaux	61
3.4 Conclusion.....	62
Chapitre 4: Vers une convergence entre le Social Computing et l'informatique pervasive	65
4.1 Sensibilité au contexte.....	66
4.2 Principales plateformes de sensibilité au contexte	70
4.2.1 Plateforme d'observation de contexte.....	71
4.2.2 Plateformes de déploiement adaptatif.....	72

4.2.3	Plateforme d'adaptation dynamique	72
4.3	Utilité de sensibilité au contexte pour le <i>Social Computing</i>	73
4.3.1	L'utilisation du contexte pour assister les échanges sociaux	75
4.3.2	L'utilisation du contexte pour améliorer les applications sociales	76
4.4	Pervasive Social Computing	77
4.5	Conclusion.....	81
Chapitre 5 : Modèle de représentation de la notion de communauté à base d'ontologies		87
5.1	Communauté spontanée	88
5.1.1	Définition	88
5.1.2	Usages potentiels	91
5.2	Ontologie de communauté	92
5.2.1	Utilisation des vocabulaires existants	94
5.2.2	Les extensions définies	98
5.3	Les inférences.....	104
5.4	Conclusion.....	108
Chapitre 6 : Taldea : une architecture dédiée aux applications communautaires dynamiquement reconfigurables		109
6.1	Scénario.....	110
6.2	Exigences	113
6.3	Architecture et principe de fonctionnement	114
6.4	Les composants de l'architecture	115
6.4.1	Les core services et les services sensibles aux communautés	116
6.4.2	Le processeur d'annotation	117
6.4.3	Le moteur de requête	121
6.4.4	Le calcul de proximité	125
6.4.5	Le moteur de règles.....	134
6.4.6	Le contrôleur de confidentialité	135
6.5	L'accès aux communautés	137
6.5.1	La recommandation	137
6.5.2	La recherche.....	140
6.5.3	La création	141
6.6	La découverte des services.....	143
6.6.1	La recommandation	146
6.6.2	La recherche.....	150
6.6.3	La création	153
6.7	Conclusion.....	154
Chapitre 7: Réalisation: Prototype et Evaluation		157
7.1	Préambule : la plateforme de supervision <i>Kalimucho</i>	157
7.1.1	Kalimucho-A.....	159
7.1.2	Kalimucho kernel.....	161
7.2	Réalisation.....	165
7.2.1	Choix technologique	166
7.2.2	Scénario d'illustration : parc botanique	167
7.2.3	Taldea : le prototype	169
7.3	Evaluation.....	176
7.3.1	Mesures d'évaluation	176

7.3.2	Principes d'expérimentation et résultats	177
7.4	Conclusion.....	188
Chapitre 8 : Conclusion et perspectives		189
8.1	Conclusion générale	189
8.2	Perspectives.....	192
Bibliographie.....		196
Publications		205

Table des figures

Figure 1: Le projet MOANO.....	14
Figure 2: Organisation des chapitres.....	21
Figure 3: Les différentes caractéristiques des communautés.....	29
Figure 4: Les objectifs des applications communautaires $A \cup B$	43
Figure 5: Architecture du social computing extraite de (Wang et al. 2007).....	47
Figure 6: La combinaison <i>Sioc</i> , <i>Foaf</i> et <i>Skos</i>	50
Figure 7: L'ontologie <i>Sioc</i>	51
Figure 8: L'ontologie <i>Foaf</i>	52
Figure 9: L'ontologie Owl-Time (Miron 2009).....	53
Figure 10: L'ontologie <i>GeoRSS-simple</i> (Miron 2009).....	54
Figure 11: L'ontologie <i>Owl-S</i>	55
Figure 12: L'évolution des systèmes distribués vers les systèmes ubiquitaires (Satyanarayanan 2001).....	65
Figure 13: Vue d'ensemble des éléments du contexte (Petit 2010).....	67
Figure 14: Architecture des systèmes sensibles au contexte.....	71
Figure 15: Capture d'écran de l'appli <i>Cityvox</i> pour la recommandation de restaurant.....	75
Figure 16: Fêtes de Bayonne (spectacle musical, un spectacle tauromachique).....	76
Figure 17: L'architecture du <i>Pervasive Social Computing</i> (Zhou et al. 2012).....	77
Figure 18: Illustration de la communauté spontanée.....	89
Figure 19: Positionnement de la communauté spontanée.....	90
Figure 20: Extrait de l'ontologie de communauté.....	96
Figure 21: Représentation de l'intérêt de la communauté.....	100
Figure 22: Représentation de la typologie de la communauté.....	101
Figure 23: Représentation du contexte de la communauté.....	102
Figure 24: Exemple de correspondance au niveau des vocabulaires <i>Foaf</i> et <i>Sioc</i>	103
Figure 25: Règles d'inférence en <i>DL</i>	107
Figure 26: Fonctions de base de l'architecture <i>Taldea</i>	110
Figure 27: Illustration de l'utilisation de <i>Taldea</i>	111
Figure 28: Architecture générale de <i>Taldea</i>	115
Figure 29: Exemple de service <i>ad hoc</i> : <i>P2Pchat</i>	117
Figure 30: Exemple des différents types d'annotation (Ma et al. 2009).....	118
Figure 31: <i>TextAnnot</i> service web d'annotation.....	119
Figure 32: Extrait de l'ontologie botanique (Aussenac-Gilles et al. 2013).....	120
Figure 33: Exemple d'annotation sémantique avec <i>TextAnnot</i>	121
Figure 34: Comparaison entre distance minimum et distance de <i>Hausdorff</i>	128
Figure 35: Territoire géographique sous forme d'un cercle.....	131
Figure 36: Emprise géométrique minimale.....	132
Figure 37: L'enveloppe convexe d'un ensemble de points (longitude, latitude) en entrée.....	132
Figure 38: Exemple du problème du point dans un polygone.....	133
Figure 39: Architecture des systèmes à base de règles.....	135
Figure 40: Architectures de la découverte de service sémantique.....	144
Figure 41: Extrait de la description en <i>Owl-S</i> du service " <i>TakePicture</i> "......	146

Figure 42: Exemple de calcul de la distance Hausdorff entre profil de communauté et description de service.....	149
Figure 43: Algorithme de <i>matching</i> du service annoncé et du service demandé selon l'approche Dom (Paolucci et al. 2002).....	150
Figure 44: Exemple d'annotation d'une requête en langage naturelle pour la recherche de service.....	152
Figure 45: Exemple de description d'un service.....	153
Figure 46: Extrait de la description en <i>Owl-S</i> du service " <i>Alarme</i> ".....	154
Figure 47: Architecture générale.....	158
Figure 48: Architecture de la plateforme : Kalimucho-A (Da 2014).....	161
Figure 49: Illustration des actions de reconfigurations assurées par la plateforme <i>Kalimucho</i>	162
Figure 50: Modèle de conteneur de composant métier <i>OSAGAIA</i>	164
Figure 51: Modèle de Connecteur <i>Korrontea</i>	165
Figure 52: Descriptif du parc <i>Mosaïc</i>	167
Figure 53: Connexion à Taldea.....	171
Figure 54: Le module d'accès aux communautés.....	172
Figure 55: L'espace virtuel commun d'une communauté.....	173
Figure 56: Module de découverte des services.....	174
Figure 57: Exemple de services <i>ad hoc</i>	175
Figure 58: Les mesures rappel et précision dans un système d'information.....	177
Figure 59: Similarité _{Wu&Palmer} (voiture, football) calculée via l'outil WS4J.....	178
Figure 60: Courbes d'évaluation correspondant à la recommandation des communautés pour un utilisateur X.....	182
Figure 61: Courbes d'évaluation correspondant à la moyenne des précisions et à la moyenne des rappels pour la recommandation de communautés.....	184
Figure 62: Courbes d'évaluation correspondant à la recommandation des services pour une communauté Y.....	186
Figure 63: Courbes d'évaluation correspondant à la moyenne des précisions et la moyenne des rappels pour la recommandation de services.....	187

Table des tableaux

Table 1: Tableau récapitulatif des communautés et de leurs caractéristiques.....	32
Table 2: Tableau récapitulatif des travaux centrés communauté et de leurs caractéristiques..	38
Table 3: Tableau comparatif entre les réseaux sociaux et les applications communautaires...	63
Table 4: Classification des plateformes de sensibilité de contexte.	73
Table 5: Les enjeux de l'informatique ubiquitaire sur les entités sociales (Jessup et al. 2002).	74
Table 6: Comparaison des projets de social Computing en fonction des éléments du contexte et du type de l'adaptation.	81
Table 7: Extrait du fichier OWL de l'ontologie de communauté.	98
Table 8: Les outils nécessaires pour chaque type de communauté.	100
Table 9: Tableau récapitulatif des principaux concepts du modèle de la communauté.	104
Table 10: Exemple d'inférence du type de la communauté.....	106
Table 11: Règles d'inférence en <i>SWRL</i>	107
Table 12: Forme d'une requête <i>Sparql</i>	122
Table 13: Exemple de requête <i>Sparql</i>	123
Table 14: Modèle de requête <i>Sparql</i>	124
Table 15: Propriétés simples d'une distance.....	129
Table 16: Algorithme de la distance de Hausdorff.	130
Table 17: Algorithme de la proximité géographique.	134
Table 18: Exemple de règles d'accès aux données.....	137
Table 19: Algorithme de recommandation de communauté.	140
Table 20: Exemple de requête de l'utilisateur.	141
Table 21: Exemple de requête <i>Sparql</i> générée pour la recherche de communauté.....	141
Table 22: Une description en <i>Sioc</i> d'un contenu partagé sur l'espace social de la communauté.	142
Table 23: Algorithme de recommandation de service.	148
Table 24: Exemple de requête <i>Sparql</i> générée pour la recherche de service.....	152
Table 25: Règle <i>SWRL</i> pour l'annotation d'un service.....	154
Table 26: Fiche technique du matériel de test.....	167
Table 27: Test de comparaison entre distance minimum et distance de <i>Hausdorff</i>	179
Table 28: Les précisions et les rappels calculés pour la recommandation de communauté pour un utilisateur X.	182
Table 29: Comparaison entre la distance sémantique basée sur <i>Wordnet</i> et la distance sémantique pour l'ontologie botanique.	183
Table 30: Les précisions et les rappels calculés pour la recommandation de service pour une communauté Y.	186
Table 31: Tableau récapitulatif des apports théoriques de la thèse.....	189

Table des équations

Équation 1: Le contenu informationnel.....	125
Équation 2: La mesure de similarité de Wu & Palmer.....	126
Équation 3: Distance minimum pour mesurer la distance entre deux ensembles de concept.....	127
Équation 4: Distance de Hausdorff pour mesurer la distance entre deux ensembles de concept.....	129
.....	129
Équation 5: La moyenne des précisions et la moyenne des rappels.....	183

Chapitre 1: Introduction

Ces dernières années ont été marquées par l'avancée des technologies de l'information, la démocratisation des moyens de communication informatique, l'essor des dispositifs mobiles (Smartphones, tablettes tactiles, ordinateurs portables, capteurs, *etc.*) et le développement du nombre des applications proposées à l'utilisateur. L'environnement informatique personnel en a été bouleversé. Aujourd'hui, les utilisateurs bénéficient des services et des applications accessibles partout et depuis toute sorte de dispositifs et qui facilitent la collaboration, l'interaction sociale, la création et l'échange du contenu. A ce titre, le *pervasive social computing* (Zhou et al. 2012) est une discipline naissante qui a instauré une nouvelle ère d'applications sociales impactées de plus en plus par l'informatique ubiquitaire. Ceci a entraîné la fusion de deux domaines : *social computing* (Parameswaran et al. 2007) et l'informatique ubiquitaire (Weiser 1994) et fait émerger le *pervasive social computing*. Les personnes étant de plus en plus connectées grâce aux dispositifs mobiles, elles participent aux applications sociales partout et à tout moment. Elles sont à la recherche d'interactions sociales avec d'autres utilisateurs dans un environnement ubiquitaire. La notion d'ubiquité des applications sociales vise les problématiques des interactions sociales dans un environnement caractérisé par la mobilité, des dispositifs hétérogènes et un contexte changeant. Apporter de la sensibilité au contexte aux applications sociales enrichit les informations et les fonctionnalités selon la situation de l'utilisateur et ses interactions sociales.

1.1 Contexte

Nos travaux de recherche s'inscrivent dans le cadre du projet ANR projet *MOANO*¹ (Modèles et Outils pour Applications NOMades de découverte de territoire). Ce projet a permis de mettre en commun les avancées dans les domaines : du *pervasive social computing*, des interfaces homme-machine, de l'ingénierie des modèles, des architectures logicielles, de l'ingénierie des connaissances et des systèmes d'informations à caractères spatio-temporels.

¹ <http://moano.liuppa.univ-pau.fr/>

Le projet MOANO est une collaboration entre 4 laboratoires (*LIUPPA/Pau, LIFL/Lille, IRIT/Toulouse, LIG/Grenoble*) et a permis de mêler des compétences hétérogènes et complémentaires. Il a démarré en décembre 2010 pour se terminer en octobre 2014. Le cadre général du projet peut se résumer par la Figure 1:



Figure 1: Le projet MOANO.

D'un point de vue général ce projet a pour finalité de permettre à des utilisateurs finaux de concevoir des applications dédiées à la découverte de territoire et de les générer automatiquement pour un usage en situation de mobilité (contraintes logicielles/matérielles), tout en permettant des usages interactifs multimodaux. Ces applications exploitent des ressources sémantiques facilitant ainsi la recherche d'informations et le déploiement des services selon des axes spatio-temporels et thématiques.

Afin de favoriser l'usage des applications de découverte du territoire (objectifs du projet *MOANO*), nous nous sommes intéressées, dans le cadre de cette thèse, aux applications communautaires afin de faciliter la communication avec les personnes présentes sur le territoire, mais également les échanges et la diffusion de services dédiés.

1.2 Problématiques défendues et contributions

L'expansion des communautés est vérifiée par une série de besoins sociaux comme la recherche d'interactions sociales et de communion (*e.g.* discussions et partages sur des expériences vécues), le besoin en information, le besoin de reconnaissance, la réciprocité anticipée (*i.e.* on rend service en espérant que l'autre fera de même), *etc.* En informatique, les chercheurs s'intéressent de plus en plus à la notion de communauté. Ce succès se justifie par le but poursuivi par ces applications qui est de répondre aux besoins des communautés et d'offrir un 'chez soi' virtuel, spécifique à la communauté, dans lequel elle va construire sa propre identité et réaliser ses objectifs. Également, la notion de communauté représente une source d'informations contextuelles sociales. Elle est, aujourd'hui, au cœur des problématiques de personnalisation et d'adaptation des applications informatiques. Cette tendance vers les communautés révolutionne l'approche traditionnelle dans laquelle les applications sont centrées utilisateur et ne sont sensibles qu'à son environnement physique.

Dans nos travaux, nous nous intéressons aux applications informatiques centrées communautés sous différents aspects : la notion de communauté, les modèles de communautés, l'architecture des applications communautaires, les fonctionnalités offertes, *etc.* L'étude de ces différents aspects nous a amenées à approfondir nos travaux en fonctions de trois axes principaux, dont nous présenterons de manière succincte les différents résultats dans ce qui suit:

1.2.1 *Notion de communauté*

Les applications communautaires actuelles offrent des informations et des services liés au thème qui est défini à l'avance par les développeurs de ces applications et mis à la disposition des utilisateurs. Avec ce genre d'application, le thème est fixé depuis la conception de cette dernière. Puis, la communauté est créée en fonction de ce thème. Si les

centres d'intérêt de l'utilisateur évoluent, ces applications ne s'adaptent pas à ses nouveaux besoins. Il doit abandonner l'application courante et en chercher une autre. En conséquence, nous constatons un besoin croissant des utilisateurs à créer des communautés selon des thèmes qu'ils définissent eux-mêmes. Ce genre de communauté, détachée de toutes contraintes thématiques, imposées par l'application, permet de répondre à des besoins qui n'ont pas pu être résolus au niveau des communautés traditionnelles pérennes (*e.g.* organiser des événements ou des festivités, demander de l'aide, *etc.*) et de faire face à des situations inopinées (*e.g.* les accidents, les incendies, les catastrophes naturelles, les crises, *etc.*). Ce genre de situation influence les attentes des utilisateurs envers les communautés, et rompt avec les communautés pérennes thématiques, pour céder la place aux communautés éphémères, géolocalisée et spontanées.

1.2.2 *Modèle de communautés*

Malgré l'expansion des applications communautaires, il n'y a pas de collaboration et d'interopérabilité entre les applications existantes car chaque application communautaire possède sa propre conception de la notion de communauté et utilise son propre modèle. Les représentations de communauté ne sont pas toutes conçues de la même façon, ni dans les mêmes circonstances spatio-temporelles, ni par les mêmes organisations. Cependant, dans un environnement caractérisé par l'ubiquité, les communautés sont appelées à apparaître, évoluer, interopérer puis disparaître. Ces tâches sont rendues difficiles voire impossibles du fait de l'hétérogénéité de conception de ces communautés.

Par exemple, si les membres d'une communauté souhaitent recourir, dans certains cas, à l'utilisation d'un service fourni par une autre application communautaire, ils sont confrontés à l'absence d'interopérabilité entre les applications communautaires. En revanche, ces applications gagneraient à travailler ensemble, partager des ressources, échanger des données, pour mieux satisfaire l'utilisateur.

Cependant, nous constatons qu'il existe des caractéristiques communes aux communautés de ces différentes applications, d'où l'intérêt de proposer un modèle de communauté basé sur les ontologies, qui fournit une vision unique et homogène de la

communauté. Ce modèle ajoute une couche sémantique à la recherche et l'exploitation des ressources des communautés et enrichit la base de connaissances communautaire en inférant automatiquement de nouvelles informations. L'utilisation des standards du web sémantique permet d'assurer une homogénéité et une interopérabilité des données au sein des communautés. Ce modèle est indépendant du domaine considéré, mais il peut être couplé à des ontologies du domaine.

1.2.3 Architecture des applications communautaires

Parmi les problèmes majeurs des applications communautaires existantes, on peut citer la surcharge cognitive et les contraintes du contexte d'exécution.

La surcharge cognitive qui fait référence à un état dans lequel l'utilisateur est perdu dans l'ensemble des items qui lui sont proposés, concerne principalement deux volets : le contenu et les services. D'un côté, la surcharge d'information affecte principalement les communautés proposées à l'utilisateur et les informations associées. Il n'est pas aisé pour l'utilisateur de trouver les communautés susceptibles de l'intéresser parmi le nombre considérable des communautés existantes. D'un autre côté, l'utilisateur est également confronté à une surcharge de services. Trouver le service qui correspond le mieux à la communauté ou à ses exigences, parmi le nombre croissant des services proposés, est une tâche difficile. Notre objectif est d'aider les utilisateurs à surmonter cette surcharge des communautés et des services.

Dans le cadre de nos activités de recherche, nous cherchons à mettre en œuvre une architecture pour applications communautaires de découverte de communautés et des services dédiés. Les applications communautaires proposées permettent d'aider les utilisateurs nomades à intégrer des communautés environnantes et à découvrir les services dédiés à l'échange et au partage des intérêts, des sujets, des activités, des événements, *etc.* Ce qui différencie l'architecture proposée des autres est qu'elle intègre des aspects innovants comme les communautés spontanées, les services basés sur la localisation, en utilisant les technologies du web sémantique et l'adaptation dynamique des applications.

Bien que la sensibilité au contexte soit essentielle dans la démocratisation de l'informatique ubiquitaire, les applications communautaires usuelles ne sont pas conçues pour s'adapter à un contexte donné. Notre deuxième défi est donc de proposer des applications communautaires qui s'adaptent à leur contexte d'exécution. La prise en considération de l'environnement physique permet de proposer à l'utilisateur des services mieux adaptés aux ressources proposées et aux contraintes physiques rencontrées. Pour que nos applications puissent être adaptées à l'environnement physique, nous utilisons un intergiciel de sensibilité au contexte dédié au déploiement et à l'adaptation des applications.

1.3 Organisation du mémoire

Ce mémoire est structuré en deux parties présentant respectivement l'état de l'art et la contribution. Chaque partie est découpée en trois chapitres auxquels viennent s'ajouter cette introduction et une conclusion. L'organisation de ce manuscrit est illustrée dans la **Figure 2**.

La première partie est développée dans les chapitres suivants :

- Le chapitre 2 "*Les communautés : cadre théorique et applications*" traite de la notion de communauté et des applications communautaires. Dans un premier temps, sont présentées quelques définitions, les différentes caractéristiques des communautés et leur typologie. Dans un second temps, est étudiée la corrélation entre communautés et applications informatiques en présentant les travaux qui ont utilisé la notion de communauté comme un facteur de progrès pour les applications informatiques, et les travaux où les applications informatiques représentent un support à l'expansion des communautés. Ensuite, les limites des applications communautaires étudiées sont examinées.
- Le chapitre 3 "*Le Social Computing comme support des communautés*" positionne les applications communautaires dans le cadre du *social computing* comme un domaine clé à l'articulation entre une réflexion sur les principes des sciences sociales et les défis de l'informatique. Ce chapitre présente l'architecture du *social computing* composée de trois couches : les fondements théoriques, les technologies utilisées et les applications. Nous avons plus particulièrement focalisé notre attention sur certaines

technologies et applications qui nous ont inspirées pour la conception de notre outil communautaire.

- Le chapitre 4 “*Vers une convergence entre le Social Computing et l’informatique pervasive*” introduit la notion de sensibilité au contexte comme la préoccupation des applications informatiques en général, et des applications sociales plus particulièrement dans un environnement ubiquitaire. Dans un second temps, sont présentées les principales plateformes de sensibilité au contexte. Ensuite, l’utilité de sensibilité au contexte pour les applications sociales est démontrée. Finalement, un état de l’art des différents travaux existants ramenant l’ubiquitaire au monde du social avec une analyse des points forts et des points faibles de chaque approche est proposé.

Dans la deuxième partie, nous exposons notre proposition.

- Le chapitre 5 “*Modèle de représentation de communauté à base d’ontologie*” décrit le modèle de communauté spontanée. Tout d’abord, la notion de communauté spontanée et ses usages potentiels est introduite. Par la suite, une modélisation de la notion de communauté à base d’ontologies est proposée. Le but principal de ce modèle est de servir de point de référence pour décrire une communauté. Ce modèle utilise plusieurs ontologies existantes pour représenter une ou plusieurs ressources de la communauté. Par ailleurs, un ensemble d’extensions a été proposé pour donner une description aussi complète et fidèle que possible de tous les aspects relatifs à la communauté. Nous concluons ce chapitre en exposant le processus d’inférence de nouvelles informations pour enrichir la base de connaissances communautaire.
- Le chapitre 6 “*Taldea : une architecture dédiée aux applications communautaires dynamiquement reconfigurables*” introduit les applications communautaires Taldea, ses exigences, ses objectifs, ses fonctionnalités et un scénario potentiel. Nous montrons comment ces applications promeuvent la notion de communauté spontanée et facilitent la découverte de services dans un environnement ubiquitaire. Ensuite, nous présentons l’architecture générale de ces applications avec ses différents composants : *les core services, les services sensibles au contexte, le processeur d’annotation, le moteur de requête, le calculateur de proximité sémantique, le moteur*

des règles et le contrôleur de confidentialité. Puis, nous détaillons le fonctionnement des deux modules principaux de Taldea: *le module d'accès aux communautés* et *le module de découverte de service*. Pour chaque module, les différents composants et techniques utilisés sont exposés.

- Le chapitre 7 “*Réalisation: Prototype et Evaluation*“ débute par présenter *Kalimucho* une plateforme sensible au contexte dédiée au déploiement et à l’adaptation des applications. Ce *middleware* est utilisé par l’application Taldea afin de déployer es services contextuellement. Nous présentons un exemple d’exécution du prototype Taldea dans le contexte de la botanique. Enfin, les résultats des mécanismes de découverte des communautés et des services sont évalués via le prototype Taldea.
- Le chapitre 8 “*Conclusion et perspectives*“ conclut notre rapport et propose quelques perspectives liées à nos travaux.

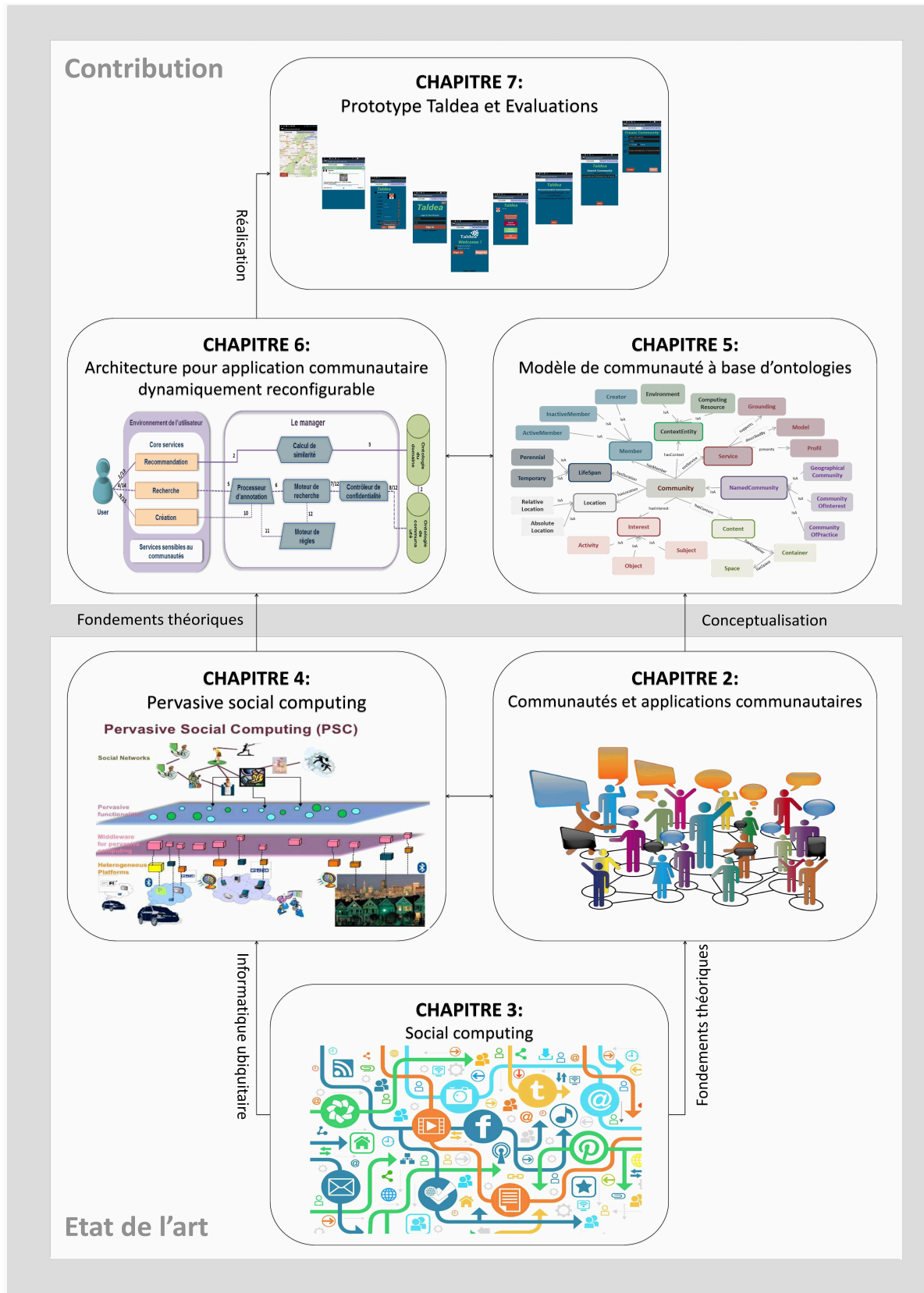


Figure 2: Organisation des chapitres.

Partie 1 : Etat de l'art

Chapitre 2 : Les communautés : cadre théorique et applications

Le but de ce chapitre est d'exposer le cadre théorique de la notion de « communautés » et des applications communautaires. La première partie présente quelques définitions, les différentes caractéristiques des communautés et leur typologie. La deuxième partie donne une vue d'ensemble des applications informatiques centrées communautés.

2.1 Cadre théorique

2.1.1 Définitions de communauté

Il n'existe pas de définition consensuelle autour de la notion de communauté, les définitions varient d'un domaine à un autre. Même dans le domaine informatique, les définitions diffèrent entre les différents sous-domaines que sont l'intelligence artificielle, les systèmes multi-agents, les architectures logicielles, les réseaux informatique, la gestion des connaissances, *etc.* Pour illustrer notre propos, prenons quelques définitions de différentes disciplines:

Pour les réseaux informatiques, Orman *et al.* (Orman et al. 2013) proposent la définition suivante : “ Une communauté est un sous-ensemble de nœuds cohérent avec des liens internes plus abondants, relativement au reste du réseau.”

Dans le domaine des systèmes d'apprentissage, Barab *et al.* (Barab et al. 2004) définissent une communauté comme: “ Un groupe de personnes réunies autour de besoins ou d'objectifs spécifiques. En fait, la communauté a fait ici émerger la problématique de choisir/recommander un élément pour quelqu'un d'autre, et, plus particulièrement, dans le cadre de l'apprentissage.”

Balaji et Sasikala (Balaji et al. 2011) présentent les communautés comme un élément de base d'analyse des réseaux sociaux : “ Aujourd'hui, le nombre des communautés, qu'elles

soient implicites ou explicites, se multiplie très rapidement. Découvrir des communautés dans un tel environnement est devenu un domaine de recherche en pleine expansion.”

Comme le montrent ces définitions, les raisons d’adhérer à une communauté varient d’une personne à une autre. D’une manière générale, nous avons identifié un certain nombre de raisons déterminantes notamment :

- La recherche d’interactions sociales ;
- Le besoin d’information ;
- Le besoin de reconnaissance ;
- La recherche de communion ;
- La réciprocité anticipée (i.e. on rend service en espérant que l’autre fera de même);
- La collaboration ;
- Le repérage des sources d’expertise (i.e. des compétences identifiées au sein de la communauté);
- *Etc.*

2.1.2 Caractéristiques d’une communauté

Avant de présenter les principaux types de communautés existants, nous passons en revue les différentes caractéristiques d’une communauté. Les travaux de *Benghozi et al.* (Benghozi et al. 2001) distinguent les caractéristiques communes et les différences entre les communautés. Certaines caractéristiques communes peuvent être objectives et faciles à identifier comme:

- *le regroupement de personnes ;*
- *l’existence de relations stables et durables entre les membres (proximité sociale, affective, géographique, etc.) ;*
- *des ressources partagées (matérielles, informationnelles, cognitives, relationnelles ou symboliques) ;*
- *l’existence d’un intérêt commun entre les membres.*

Ces caractéristiques nous seront utiles pour la modélisation des communautés. D'autres caractéristiques sont davantage latentes, subjectives et plus difficiles à identifier, citons entre autres :

- *les conditions et le contrôle d'accès à la communauté ;*
- *le sentiment d'appartenance des membres ;*
- *le phénomène de "multi-appartenance" ;*
- *la notion de frontière ou de limite de la communauté.*

Il existe, par ailleurs, d'autres caractéristiques qui diffèrent d'une communauté à une autre et qui peuvent en définir le type, citons:

- *Le statut* : le statut de la communauté peut être formel, officiel, certifié, tenu pour vrai par des organisations gouvernementales ou non, ou être non formel, c'est-à-dire un groupement de personnes organisé en dehors de tout système et non certifié par une autorité.
- *La taille* : les communautés sont de toutes tailles, tout dépend de leur type. Le nombre des membres varie selon les caractéristiques de chaque communauté, telles que les moyens de communication, la thématique, les objectifs, *etc.* La taille peut être restreinte ou grande.
- *La répartition géographique* : les membres de la communauté peuvent être situés sur le même territoire ou peuvent être dispersés partout dans le monde.
- *La durée de vie* : en théorie, la durée de vie d'une communauté est indéterminée. En créant une communauté, on n'en prévoit pas le terme. Mais, en pratique, on trouve des communautés permanentes qui existent aujourd'hui depuis longtemps, ainsi que des communautés temporaires qui sont créées dans un but bien défini, puis sont abandonnées après la réalisation ou la non validité de leurs objectifs (elles deviennent obsolètes). Nos travaux s'intéressent plus particulièrement aux communautés temporaires, éphémères, spontanées. En effet, c'est le type de communauté qui s'accorde le mieux avec les applications communautaires ubiquitaires. Ces dernières assistent les utilisateurs dans la création de communautés qui répondent à une situation ponctuelle (besoins, objectifs, nouvel emplacement géographique...).

- *Les objectifs* : ils varient d'une communauté à une autre. En principe, les objectifs à atteindre sont clairement identifiés avant ou lors de la création de la communauté. Par exemple, la communauté des « *gamers* » a pour objectif de permettre l'échange sur des sujets en lien avec les jeux vidéo.
- *Les modes de communication* : l'interaction entre les membres de la communauté peut être physique (contact direct avec les autres) ou virtuelle (à travers le réseau, en utilisant les technologies de communication).
- *Le type des membres* : peut être déterminé à partir de leurs expériences et leurs connaissances. Dans les premières formes de communautés, les membres partageaient le même rang social ou les mêmes préoccupations intellectuelles, comme c'est le cas pour les communautés de noblesse ou les communautés scientifiques. Avec la nouvelle génération de communautés, les membres peuvent avoir des connaissances proches, des expériences similaires, des intérêts communs, un territoire partagé, *etc.*, comme ils peuvent être totalement différents et seulement partager un point commun. Parmi les catégories des membres, on peut citer : l'expert, membre actif, membre non actif, *etc.*
- *La nature des relations entre les membres* : elle peut être économique, sociale, scientifique, *etc.* Elle peut également relever de relations basées sur la proximité géographique.

Toutes ces caractéristiques sont variables d'une communauté à une autre. C'est à travers les valeurs de ces caractéristiques que l'on peut déterminer la typologie de la communauté. Nous proposons dans la Figure 3 un schéma récapitulatif des différentes caractéristiques d'une communauté.

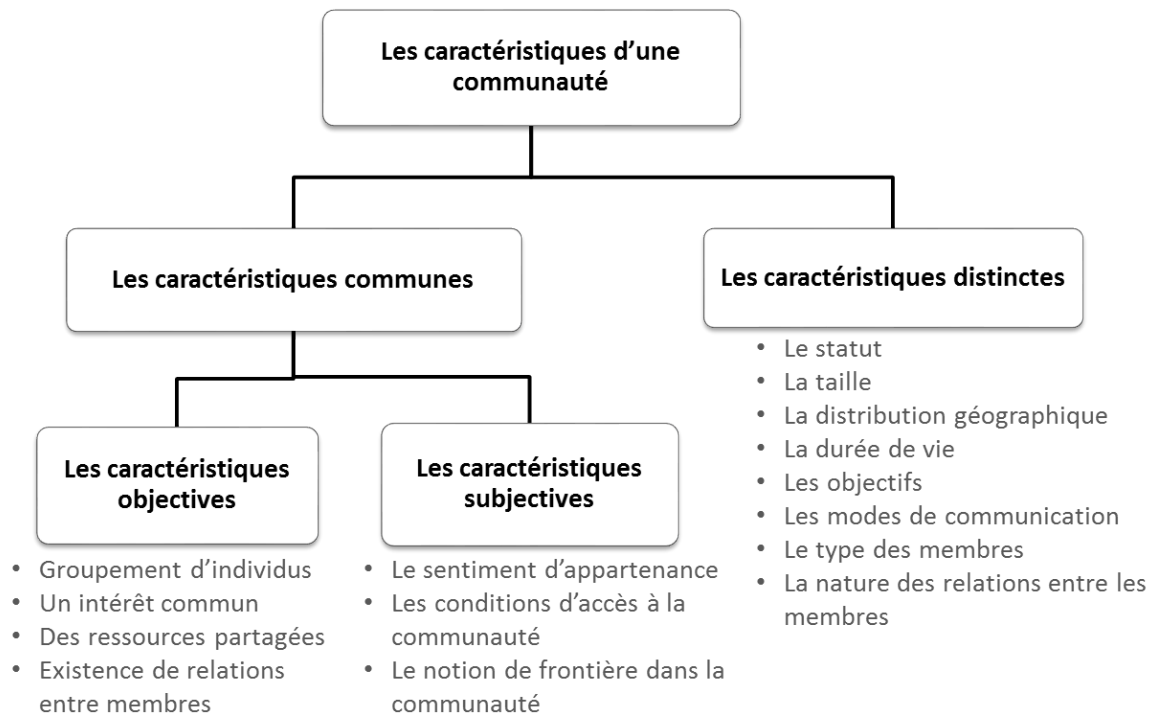


Figure 3: Les différentes caractéristiques des communautés.

La Figure 3 représente les caractéristiques communes et les caractéristiques distinctes qui définissent le type d'une communauté. Dans la section suivante, nous présentons ces différents types de communautés.

2.1.3 Typologie des communautés

Comme précédemment expliqué, les définitions d'une communauté varient, mais une caractéristique commune se dégage de toutes les définitions citées ci-dessus : *une communauté est un groupement d'individus*. Ce groupement peut être fondé sur un intérêt commun ou sur une pratique, il peut être physique, virtuel, géographique. Au contraire de la définition de la communauté, la typologie fait l'objet d'un consensus dans plusieurs travaux de recherche.

2.1.3.1 Communautés d'intérêt

La notion de communauté d'intérêt a été formalisée pour la première fois par Licklider et Taylor dans le cadre d'un projet de recherche en informatique mené par l'ARPA (*le département américain de la défense*). Pour eux, une communauté d'intérêt « *repose non pas sur une localisation commune mais sur un intérêt commun* » (Licklider et al. 1968). L'intérêt en question peut couvrir des domaines tels que la science, la religion, le patrimoine, *etc.* Nous présentons, dans ce qui suit, quelques exemples de communautés d'intérêt existantes, en ciblant les thèmes fondamentaux autour desquels elles sont construites.

- *Les communautés sociologiques* : historiquement, ce sont les premières formes de communautés. Une telle communauté est un groupe social partageant la même appartenance socioculturelle ou une identité de groupe. On parle ici de communautés religieuses, communautés de noblesse, communautés de métier...
- *Les communautés scientifiques* : ce terme désigne l'ensemble des personnes qui appartiennent ou s'intéressent à la recherche scientifique. Le terme de communauté scientifique est d'usage fréquent. Elles tiennent une place non négligeable parmi les autres communautés.
- *Les communautés écologiques* : ce genre de communauté traite de problèmes liés à l'environnement, à l'habitat écologique, à l'alimentation bio et au bien-être.

Le concept de communauté d'intérêt représente un type de communauté, mais les instances de ces concepts sont assez nombreuses et variées en termes de cible ou de centre d'intérêt. Nous n'avons cité ci-dessus que quelques exemples de communauté d'intérêt parmi bien d'autres. Dans ce qui suit, nous présentons un autre type de communauté : la communauté de pratique.

2.1.3.2 Communautés de pratique

Ce concept a été développé par Wenger (Wenger 1998). Il définit une communauté de pratique comme « *un groupe de personnes qui travaillent ensemble et qui sont conduites à inventer constamment des solutions locales aux problèmes rencontrés dans leurs pratiques* ».

professionnelles. Après un certain temps, et au fur et à mesure que ces personnes partagent leurs connaissances, leurs expertises, ils apprennent ensemble ». Considérant cette définition, nous pouvons en conclure qu'une communauté de pratique est un ensemble de personnes engagées dans la même occupation ou dans la même carrière, qui œuvrent ensemble pour réussir une tâche, comprendre un nouveau phénomène, mener à bien une tâche ou afin d'atteindre des objectifs à travers leurs pratiques respectives.

Dans la littérature, il apparaît une différence entre une communauté de pratique et une communauté d'apprentissage. Riel and Polin 2004 (Riel et al. 2004) définissent une communauté d'apprentissage comme un groupe d'individus qui œuvrent ensemble dans un temps déterminé pour réussir une tâche ou pour comprendre un nouveau phénomène ou pour compléter une tâche collaborative. Nous considérons la communauté d'apprentissage comme une sorte de communauté de pratique puisque c'est un groupe d'individus qui travaillent et apprennent ensemble, mais dans le cadre d'une formation (formalisée, institutionnalisée ou non).

2.1.3.3 Communautés géographiques

La géolocalisation est l'objet d'un fort engouement actuellement dans le domaine des communautés. Les communautés géographiques se caractérisent par le partage d'un même territoire délimité géographiquement. Ainsi, elles se caractérisent par un groupe d'individus se trouvant dans la même zone géographique, possédant un intérêt commun et qui coopèrent pour réaliser les objectifs qui ont présidé à la création de la communauté. Les relations entre les membres de la communauté ne sont plus basées sur les liens sociaux ou sur des idées, mais plutôt sur des relations de proximité géographique, c'est-à-dire que leur point commun principal est la proximité géographique. Une communauté qui s'intéresse à un lieu donné (*e.g.* Los Angeles) n'est considérée comme communauté géographique que lorsque ses membres sont sur le même lieu, sinon elle est considérée comme une communauté d'intérêt. Les amateurs de ce genre de communauté profitent des technologies de l'information et de la communication. Ils bénéficient notamment des applications communautaires orientées géolocalisation pour communiquer et interagir.

À l’opposé, en raison de l’éloignement géographique des personnes, un nouveau type de communauté est apparu, lié davantage à la révolution technologique : les communautés virtuelles.

2.1.3.4 Communautés virtuelles

Ce type de communauté connaît un succès croissant. Ce n'est pas un phénomène récent, même si c'est aujourd'hui que l'on en parle le plus. Dans son ouvrage sur les communautés virtuelles, Howard Rheingold a défini les communautés virtuelles comme « *les regroupements socioculturels qui émergent du réseau lorsqu'un nombre suffisant d'individus participent à ces discussions publiques pendant assez de temps en y mettant suffisamment de cœur pour que des réseaux de relations humaines se tissent au sein du cyberspace* » (Rheingold et al. 1995). Elles sont également désignées comme des communautés en réseau, c’est-à-dire que les membres de ces communautés communiquent par voie électronique, ils adoptent les technologies de l'information et de la communication pour interagir. Avec l’évolution des moyens de communication à distance, la communauté virtuelle n’est plus un type de communauté mais plutôt une caractéristique des communautés. Dans la Table 1, nous résumons les différentes communautés existantes et les principales caractéristiques qui les différencient :

Type de communauté	Objectifs	Distribution géographique	Mode de communication	Type de membres
Communauté d'intérêt	Comprendre, interpréter et chercher des solutions aux problèmes liés aux centres d'intérêt.	Dispersé et/ou regroupé géographiquement	Physique et/ou virtuel	Indifférent
Communauté de pratique	Réussir une tâche ou comprendre un nouveau phénomène ou trouver une solution dans les pratiques professionnelles.	Dispersé et/ou regroupé géographiquement	Physique et/ou virtuel	Exercent la même pratique
Communauté virtuelle	Communiquer à travers une voie électronique pour réaliser des objectifs relatifs à un centre d'intérêt.	Dispersé et/ou regroupé géographiquement	Virtuel	Utilisateurs des TIC
Communauté géographique	Communiquer, échanger et valoriser les informations géographiques dans un environnement interactif.	Groupé géographiquement	Physique et/ou virtuel	Personnes demeurant sur le même territoire géographique

Table 1: Tableau récapitulatif des communautés et de leurs caractéristiques.

Nous avons identifié, dans la Table 1, quatre types majeurs de communautés. Mais cette classification reste théorique, et, en réalité, les communautés virtuelles et les communautés géographiques peuvent être considérées comme des communautés complémentaires ou des nouvelles caractéristiques de la communauté d'aujourd'hui.

2.2 Applications communautaires : est-ce l'informatique au service des communautés ou bien l'inverse ?

En informatique, les chercheurs s'intéressent de plus en plus aux communautés virtuelles compte tenu de leur succès. A titre d'exemple, nous citons Google Group², Yahoo Groups³, Facebook Group⁴, Twitter Group⁵, Orkut⁶, Friendster⁷, *etc.* Ce succès se justifie par le but poursuivi par ces applications informatiques de répondre aux besoins des communautés et d'offrir un 'chez soi' virtuel, spécifique à la communauté, dans lequel elle va construire sa propre identité et fixer ses objectifs.

Par ailleurs, les communautés représentent une nouvelle dimension de contexte et une source d'information contextuelle sociale. Elles sont au cœur des problématiques de personnalisation et d'adaptation des applications informatiques. Cette sensibilité au contexte social de l'utilisateur révolutionne l'approche traditionnelle où les applications sont centrées utilisateur et sensibles à son environnement physique.

2.2.1 Les applications informatiques comme support à l'expansion des communautés

Les applications communautaires sont devenues très populaires grâce à l'essor des dispositifs mobiles tels que les *Smartphones* et les tablettes tactiles. Quel que soit le domaine, la finalité est la même : réunir des utilisateurs à travers un réseau informatique afin qu'ils puissent réaliser les objectifs de leur communauté. Ces applications offrent différents services répondant aux besoins des membres de la communauté et s'articulent autour d'un intérêt ou

² <https://plus.google.com/getstarted/suggest?fww=1>

³ <https://fr.groups.yahoo.com/neo>

⁴ <http://www.facebookgroups.com/>

⁵ <https://twitter.com/solocalgroup>

⁶ <https://orkut.google.com/>

⁷ <http://www.friendster.com/>

d'une pratique commune. On peut résumer le rôle des applications communautaires par les points suivants :

- Favoriser la production, l'exploitation, la recherche et le stockage de toutes les informations produites par la communauté;
- Organiser les échanges au sein de la communauté (les protocoles et les formats, etc.);
- Améliorer la cohésion de la communauté;
- Accumuler les connaissances de la communauté;
- Mobiliser des acteurs sur la création du contenu et assurer leur fidélisation;
- Assurer la communication entre les membres de la communauté ;
- Annoter et contextualiser l'information partagée au sein de la communauté ;

Plusieurs travaux de recherche se sont intéressés aux communautés sous différents angles et de différentes manières, ce qui a favorisé le succès des applications communautaires aujourd'hui. Parmi ces travaux on peut lister :

Maret *et al.* (Maret et al. 2014) qui proposent un modèle et une architecture pour un *framework* permettant aux utilisateurs de dispositifs mobiles de créer et de participer à des communautés virtuelles. Le modèle est basé sur RDF, le langage de base du web sémantique. Quant à la plateforme, elle est basée sur une approche multi-agent, pour gérer les interactions entre les utilisateurs. Les auteurs de ce travail proposent des communautés *ad hoc* principalement destinées aux *Smart Cities*.

Christo El *et al.* (Christo El et al. 2012) s'intéressent aux communautés virtuelles comme un outil opérationnel pour l'échange et le partage d'informations entre personnes handicapées, mais également avec les personnes défendant les droits des personnes handicapées. Cette application communautaire assure la collaboration entre les différents participants, ainsi que l'alimentation de la base de connaissances des communautés des personnes handicapées. C'est un outil pour défendre les droits de cette communauté.

Cabitza *et al.* (Cabitza et al. 2006) visent les applications collaboratives qui gèrent les informations et les comportements caractérisant les communautés. La particularité de

l'application proposée est qu'elle prend en compte les différents niveaux de participation des membres des communautés. Parmi les cas d'utilisation de ce genre d'application, on peut citer une organisation de *workshop* où l'on incite les gens à y participer.

Garrot *et al.* (Garrot et al. 2007) proposent un outil support aux communautés de tuteurs, appelé *TE-Cap*. Ce dernier met en relation des outils de communication et un outil de gestion des connaissances produites par les échanges au sein de la communauté. Il permet de classer et rechercher toutes les informations produites par la communauté à partir d'une interface proposant des thèmes propres aux pratiques liées à cette communauté.

L'idée clé du travail de Nguyen *et al.* (Nguyen et al. 2011) est d'exploiter les bases de connaissances des communautés virtuelles géolocalisées qui sont considérés aussi comme des connaissances du territoire. Un *framework*, appelé *PlaceComm*, est proposé pour construire des applications sensibles au contexte où les connaissances produites au sein des communautés sont partagées et centralisées via une ontologie.

Les travaux présentés ci-dessus ont contribué d'une façon ou d'une autre, à travers des modèles, des architectures, des outils, *etc.*, à offrir à la communauté un espace virtuel permettant d'organiser ses échanges, de capitaliser les connaissances produites et de proposer des services qui répondent aux besoins de ses membres. Les nouvelles technologies ont fait évoluer les communautés. Dans le sens inverse, comment les communautés contribuent-elles au progrès de l'informatique ?

2.2.2 La communauté, un facteur de progrès pour les applications informatiques

Sommairement, quatre types d'entités influencent les applications informatiques : les entités physiques, les entités système, les entités utilisateurs et les entités sociales. La communauté est considérée comme une entité sociale révélatrice des informations importantes qui contribuent au développement des applications informatiques. Parmi les travaux qui ont mis en lumière les communautés et leur rôle pour le progrès des applications informatiques, on peut citer :

L'article de Abrouk *et al.* (Abrouk et al. 2010) présentent une méthode de découverte de communautés implicites sur le web collaboratif par observation des usages (*i.e.* les actions des utilisateurs). Découvrir ces communautés implicites est un véritable gain pour différents acteurs : les agents de publicité, les concepteurs du système et les utilisateurs finaux, *etc.* La méthode de détection de communauté s'appuie sur une analyse statistique des étiquettes des ressources, et de l'utilisation de ces ressources par les utilisateurs. Cette approche a été validée sur une base de données filmographique (*MoviLens*). Ce qui a permis d'extraire des communautés de genre : communauté des films pour enfants, communauté des films de romance, communauté des documentaires, *etc.*

Fauvet (Fauvet 2008) se base sur l'hypothèse qu'une communauté rassemble un ensemble de services répondant à un même ensemble de besoins fonctionnels. Ainsi, une communauté peut être perçue comme offrant un niveau d'abstraction intermédiaire entre des applications clientes et les services qu'elles désirent utiliser. L'objectif de son approche est de substituer un service web par un autre de la même communauté de service, répondant mieux à des besoins fonctionnels et non fonctionnels (*e.g.* des critères de qualité). Un modèle de communauté, ainsi qu'une architecture logicielle, sont proposés.

L'augmentation de la création et du partage des documents multimédia est exponentielle. L'annotation de ces documents facilite leur organisation mais cette tâche reste toujours difficile. Chakravarthy *et al.* (Chakravarthy et al. 2006) proposent une approche utilisant des communautés pour faciliter la tâche d'annotation et permettre une réutilisation des connaissances annotées. Cette approche se base sur une ontologie de domaine pour l'annotation, et une interface à travers laquelle un membre de la communauté peut introduire ces annotations.

Daqing *et al.* (Daqing et al. 2012) étudient les comportements individuels et les interactions au sein des communautés en exploitant les traces numériques dans les espaces cyber-physiques. Les traces numériques sont générées principalement à partir de trois sources d'information : les applications Internet et Web, l'infrastructure statique, les appareils mobiles et les capteurs portables. L'exploitation des traces numériques des communautés permet de mieux comprendre nos vies, nos organisations et nos sociétés et peut apporter des innovations

dans des domaines comme la santé, la sécurité publique, la gestion des ressources de la ville, la surveillance de l'environnement et le transport.

Selon le même principe Leprovost *et al.* (Leprovost et al. 2012a) se basent sur les interactions des utilisateurs au sein de communautés virtuelles pour identifier les utilisateurs centraux ou influents, ainsi que les thématiques populaires.

Coutand *et al.* (Coutand et al. 2005) proposent un système qui gère les communautés dans un environnement ubiquitaire. Les informations produites par les communautés sont interprétées pour fournir à l'utilisateur le bon service au bon moment. Ce système fournit des services destinés aux communautés et tire profit des interactions au sein de ces dernières pour enrichir le contexte de l'application, et par la suite, faire les adaptations nécessaires.

Wang *et al.* (Wang et al. 2004) révèlent la notion de communauté de périphériques basée sur la proximité géographique des équipements. Avec une interconnexion de type *ad hoc*, les objets communicants interagissent entre eux. Un modèle a été construit pour représenter les interactions au sein de ce genre de communautés. Ce travail permet de montrer comment la notion de communauté de périphériques peut supporter les actions des utilisateurs.

Ci-dessous, nous présentons dans la Table 2 les caractéristiques fondamentales des différents travaux qui ont abordé la notion de communauté dans le domaine de l'informatique. Ces travaux se distinguent par leur architecture, leur système d'information, leur sensibilité au contexte, les fonctionnalités orientées communauté et le support.

	Système d'information	Architecture	Sensibilité au contexte	Fonctionnalité	Support
(Maret et al. 2014)	Modèle de données sémantique	Distribuée	Contexte d'exécution (État des devices)	Joindre des communautés dans des Smart Space	Open Framework pour des communautés
(Leprovost et al. 2012a)	Sémantique	Distribuée	Non	Comprendre les communications dans les communautés en ligne	Web collaboratif
(Chakravarthy et al. 2006)	Sémantique	Centralisée	Non	Annotation des documents multimédia	Application d'annotation
(Fauvet 2008)	Semi-structuré	Centralisée	Contexte d'exécution (Qualité de service)	Sélection dynamique des services web	Application web
(Zhang et al. 2010)	Données multimodales et hétérogènes	Centralisée	Contexte d'exécution (Données provenant des capteurs)	Révéler les habitudes des communautés	Framework pour l'intelligence sociale et communautaire
(Christo El et al. 2012)	Base de connaissances	Centralisée ou Distribuée	Non	Promotion des droits des personnes handicapées	Virtual Knowledge Network
(Leprovost et al. 2012b)	Sémantique	Distribuée	Non	Découverte des communautés	Web social
(Coutand et al. 2005)	Données de contexte	Centralisée	Contexte d'exécution (Capacités des dispositifs mobiles)	Répondre aux besoins des groupes dans un environnement mobile	Applications et services
(Cabitza et al. 2006)	Syntaxique	Distribuée	Contexte des communautés	Gérer la participation des membres de la communauté	Application collaborative
(Daqing et al. 2012)	Traces numériques	Centralisée	Contexte d'exécution (Données provenant des capteurs)	Etudier les traces numériques pour révéler des liaisons et des comportements sociaux	Framework
(Garrot et al. 2007)	Syntaxique	Centralisée	Non	Instrumentation des communautés de pratique	Portail (TE-cap)
(Nguyen et al. 2011)	Sémantique	Centralisée	Contexte de l'utilisateur Contexte géographique Contexte du périphérique	Communautés virtuelles géographiques	Framework
(Wang et al. 2004)	Syntaxique	Distribuée	Contexte social Contexte du périphérique	Soutenir les groupes sociaux dans un environnement Ubicom	Applications

Table 2: Tableau récapitulatif des travaux centrés communauté et de leurs caractéristiques.

Après avoir recensé les caractéristiques et les fonctionnalités des différentes approches centrées communautés, nous présentons dans la suite les limites que nous avons pu constater en étudiant ces différents travaux.

2.3 Limites des applications centrées communauté

Nous avons observé certaines limitations qui touchent les applications communautaires dans différents domaines (*e.g.* les systèmes d'information, la sécurité informatique, l'intelligence artificielle, *etc.*) :

- *La contrainte thématique* : les applications communautaires offrent des informations et des services liés au thème de l'application qui est défini à l'avance par les développeurs de l'application et mis à la disposition des utilisateurs. Dans ce genre d'application, le thème est fixé depuis la conception. Puis, la communauté est créée en fonction de ce thème. Si l'utilisateur change de centre d'intérêt, ces applications ne s'adaptent pas à ses nouveaux besoins. Il doit abandonner l'application courante et en chercher une autre.
- *L'absence d'interopérabilité et de collaboration* : si les membres d'une communauté recourent, dans certains cas, à l'utilisation d'un service fourni par une autre application communautaire, ils sont confrontés à un problème d'interopérabilité et de collaboration entre les applications communautaires. La collaboration consiste à fournir une représentation sémantique unifiée des descriptions relatives aux communautés, interprétable par les membres et par les machines, permettant ainsi de favoriser le partage et l'échange de l'information relative aux communautés. L'interopérabilité permet de faciliter l'exploration et l'extraction des informations issues de différentes sources hétérogènes. Chaque application traite la communauté selon sa propre conception. La définition et les caractéristiques d'une communauté sont différentes d'une application à une autre. En revanche, ces applications gagneraient à travailler ensemble, partager des ressources, échanger des données, pour mieux satisfaire l'utilisateur. Compte tenu de la diversité des définitions, des typologies des communautés, et des besoins en interopérabilité entre les différentes applications communautaires, un modèle consensuel de communautés est indispensable pour permettre aux applications d'interopérer. L'objectif de ce modèle est de cerner les caractéristiques des communautés, définir un vocabulaire commun

regroupant l'ensemble des concepts décrivant ces dernières afin de fournir une vision unique et homogène de la communauté.

- *La dispersion du capital informationnel de la communauté* : le développement sans précédent des technologies de l'information et de la communication rend plus facile l'échange des informations au sein des communautés. L'intervention humaine des membres des communautés transforme ces informations brutes en connaissances acquises sur la communauté et sur son environnement. Les connaissances produites à partir des interactions sociales sont appelées *des fragments sociaux* par Deparis *et al.* (Deparis *et al.* 2011). Aujourd'hui, les problèmes majeurs des applications en ligne, sont la dispersion du capital informationnel de la communauté sur les plateformes Cloud et les plateformes *Social media*, la difficulté d'acquérir ou d'analyser les informations en provenance de différentes sources, le manque de fiabilité et l'incomplétude des informations fournies par les utilisateurs des réseaux sociaux. En contrepartie, l'utilisation des applications locales, assure une capitalisation des informations échangées au sein de la communauté pour enrichir la base de connaissances communautaires, une rapidité de recherche et de diffusion des informations dans des situations critiques comme des accidents ou des catastrophes naturelles. La capitalisation de ces fragments sociaux est devenue essentielle pour enrichir la base de connaissances des différents intervenants (*e.g.* la communauté, l'organisation, le territoire géographique, *etc.*). Quelques projets en cours visent à utiliser les connaissances sociales produites par les communautés comme : SPIPOLL⁸ qui a pour but d'obtenir des données quantitatives sur les insectes ou OPERATION ESCARGOT⁹ qui invite les utilisateurs à identifier les escargots et les limaces dans un jardin. Les observations sont ensuite transmises aux musées scientifiques. De même, le programme SAUVAGES DE MA RUE¹⁰ vise à permettre aux citoyens de reconnaître les espèces de plantes qui poussent dans leur environnement et de les envoyer à des chercheurs grâce à un outil communautaire. Malgré ces expériences, l'exploitation des fragments sociaux au sein des communautés reste encore modeste.

⁸ <http://www.spipoll.org/>

⁹ <http://www.noeconservation.org/index2.php?rub=12&srub=31&ssrub=322&goto=contenu>

¹⁰ <http://sauvagesdemarue.mnhn.fr/sauvages-de-ma-rue/presentation>

Les applications communautaires doivent travailler mieux pour profiter des connaissances produites les échanges sociaux.

- *Le non contrôle des données* : les communautés en ligne offrent de nouveaux moyens de communication, d'échange et de partage social. La fréquence des interactions sociales est assez importante et le degré de confidentialité des données est insatisfaisant. L'atteinte à la vie privée des membres de la communauté concerne les documents multimédia (photo, vidéo, *etc.*) échangés, les données personnelles (âge, e-mail, *etc.*), la position géographique, *etc.* Ces données collectées peuvent être utilisées de différentes manières : la profilisation (de vos opinions, de votre caractère, de vos préférence et de vos intérêts), les programmes de surveillance, les actions de marketing, la location de fichiers, le chantage, *etc.* Plusieurs espions sont à l'affût comme le lobby des géants de l'Internet, les hackers, les curieux, *etc.* Par conséquent, les notions de vie privée et de contrôle de données deviennent un défi ultime. Pour protéger sa vie privée, l'utilisateur est fortement invité à utiliser une connexion sûre, choisir un espace de stockage de donnée sécurisé, adopter de bonnes pratiques (nettoyage, chiffrement), utiliser des outils fiables, *etc.* La protection des données est devenue un sujet majeur des travaux de recherche. Dans le livre (Camenisch et al. 2011) les auteurs traitent ce sujet sous ses différents aspects (les besoins, les acteurs, les menaces, les mesures, les outils, *etc.*).
- *L'aspect statique des systèmes* : actuellement, la plupart des applications communautaires sont des applications statiques au sens du contenu et du comportement (*i.e.* déploiement des services). Pour le contenu, les informations de l'utilisateur (intérêts, préférences, *etc.*) sont mises à jour explicitement par celui-ci. Néanmoins, certaines applications adoptent des mises à jour dynamiques du contexte de l'utilisateur dont son emplacement géographique. Toutefois, le travail sur le déploiement dynamique des services reste encore modeste. Les applications sont pourtant destinées à changer de comportement pour suivre le changement du contexte d'exécution de l'utilisateur en ajoutant et supprimant des services. L'évolution du contenu et du comportement rendent ce genre d'applications plus adapté à l'utilisateur, flexible et personnalisable.

- *La perception partielle de la communauté* : les applications communautaires sont limitées à une perception partielle des caractéristiques et de la typologie des communautés citées ci-dessus. Ainsi, se pose un problème de vision générale et d'extensibilité en termes de nouveaux éléments de communautés.
- *L'inaptitude à gérer des communautés dans des situations critiques* : les applications sensibles au contexte en environnement ubiquitaire permettent de fournir à un utilisateur nomade les informations et les services dont il a besoin n'importe où et à n'importe quel moment. Ce genre d'application gère les interactions d'un groupe d'utilisateurs dans divers domaines applicatifs (commerce mobile, apprentissage ubiquitaire, secteur médical, etc.). Néanmoins, dans le cas de circonstances inattendues ou critiques, la diffusion d'information lors de situations inopinées (e.g. des catastrophes naturelles, des accidents, des incendies, etc.) est complexe. Quand on est confronté à ce genre de situation, on peut faire appel à des communautés éphémères ou temporaires dont l'intérêt est de collaborer pour résoudre la situation. Une telle communauté disparaît lors d'un changement de situation de ses membres ou suite à la satisfaction ou à l'obsolescence du besoin pour lequel elle a été créée. De telles applications doivent relever plusieurs défis comme : inciter des utilisateurs à rejoindre la communauté créée, gérer l'hétérogénéité matérielle et logicielle des dispositifs des utilisateurs, fournir le bon service au bon moment et assurer la rapidité de réponse. Ceci devra être réalisé sans pour autant compromettre la vie privée de l'utilisateur ou causer une surcharge d'information.

2.4 Conclusion

Dans la première partie de ce chapitre, nous avons passé en revue les définitions des communautés, leurs différentes caractéristiques et leur typologie. Dans la seconde partie, nous avons étudié la corrélation entre communautés et applications informatiques en présentant les travaux qui ont utilisé la notion de communauté comme un facteur de progrès pour les applications informatiques et les travaux où les applications informatiques représentent un support à l'expansion des communautés. Ensuite, nous avons examiné les limites des applications communautaires existantes.

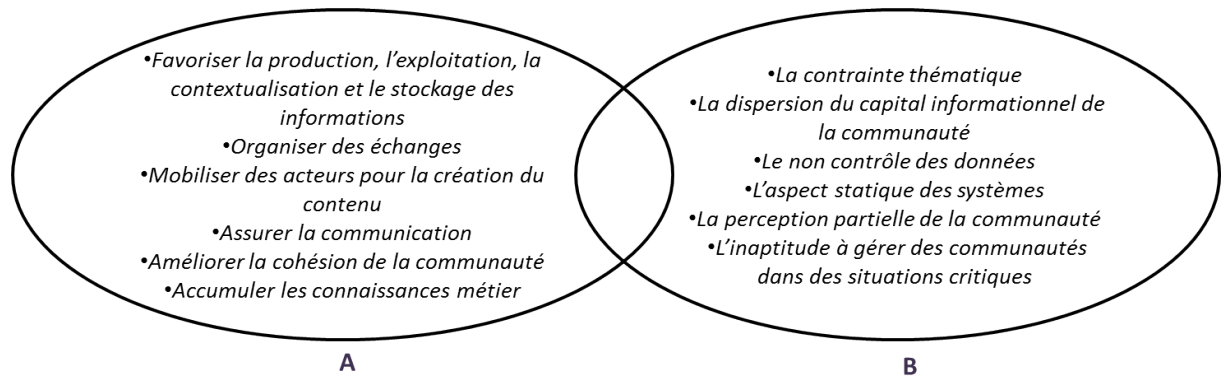


Figure 4: Les objectifs des applications communautaires ($A \cup \bar{B}$).

Dans la Figure 4, nous rappelons les objectifs généraux des applications communautaires (**A**) et nous précisons les difficultés à surmonter (**B**) afin de déterminer l'aspect général de notre application communautaire ($A \cup \bar{B}$).

Chapitre 3 : Le Social Computing comme support des communautés

Aujourd'hui, grâce aux avancées technologiques notamment dans les domaines de la communication, des réseaux, du traitement et du stockage des données, des capteurs, *etc.* des milliers de communautés virtuelles se propagent sur internet formant un élément d'un grand puzzle : *le social computing*. C'est une discipline naissante de l'informatique et du développement technologique qui a connu une expansion ces dernières années. Le *social computing* est devenu le carrefour de plusieurs domaines de recherches dont notamment l'informatique et les sciences sociales. Les laboratoires *Intel*, dans leur livre blanc '*ISTC for Social Computing*', définissent le *social computing* comme « *le phénomène social des technologies de l'information et des médias numériques. On n'utilise plus les technologies numériques pour comprendre le social mais on utilise plutôt les sciences sociales pour comprendre la vie numérique* ». En se basant sur les sciences sociales, les chercheurs se sont intéressés à ce que l'on appelle l'empreinte numérique des utilisateurs. L'analyse de ces empreintes numériques révèle plusieurs informations utiles sur le comportement humain, les interactions sociales, les personnes d'influence, les sujets d'actualités, *etc.*

Parameswaran and Whinston donnent une vue d'ensemble de ce domaine. Ils le présentent comme : « *un grand nombre de nouvelles applications et services qui facilitent l'action collective et l'interaction sociale en ligne avec un riche échange d'informations multimédias et une évolution des connaissances globales. Les exemples incluent les blogs, les wikis, les social bookmarking, les réseaux sociaux et les communautés. Le social computing permet aux utilisateurs, avec des connaissances technologiques relativement modestes, de manifester leur créativité, de s'engager dans l'interaction sociale, d'apporter leur expertise, de partager du contenu, de construire collectivement de nouveaux outils, de diffuser l'information et d'assimiler le pouvoir de négociation collective.* » (Parameswaran et al. 2007).

Le principal objectif du *social computing* est de créer les outils et l'infrastructure nécessaire pour supporter les communautés en ligne dans leur communication et leur échange du contenu multimédia. Les géants de l'informatique se précipitent pour avancer dans ce domaine. Plusieurs produits ont été proposés sur le marché pour promouvoir les communautés et assurer les échanges sociaux. Nous citerons notamment *Facebook Groups*¹¹, *Google Groups*¹², *GroupTweet*¹³, *Yahoo Groups*¹⁴, *Orkut*¹⁵, *Friendster*¹⁶, etc. Depuis son avènement, le *social computing* a déjà participé à des prises de décision dans plusieurs domaines d'application comme l'économie, la politique (e.g. le printemps arabe), etc. Ce domaine en pleine croissance a contribué à solliciter l'invention et l'innovation technologique.

Dans la suite de ce chapitre, nous nous contentons de présenter les fondements théoriques de ce domaine, les technologies qui nous intéressent dans notre travail et quelques domaines d'application.

3.1 Fondements théoriques

Passer d'une culture d'utilisateur consommateur face à un système statique à celle d'un utilisateur producteur en interaction avec d'autres utilisateurs producteurs est devenu la base du *social computing*. Le principe de ce domaine est d'étudier comment les technologies de l'information et de la communication (TIC) et le social s'influencent l'une l'autre. C'est pourquoi ce domaine est considéré comme pluridisciplinaire dans la mesure où il regroupe les sciences sociales et les sciences informatiques.

Wang *et al.* (Wang *et al.* 2007) énumèrent les domaines utiles au *social computing* « pour concevoir des systèmes sociotechniques, le *social computing* doit apprendre de la sociologie, de l'anthropologie¹⁷ et des théories psychologiques et organisationnelles. Du point

¹¹ <http://www.facebookgroups.com/>

¹² <https://groups.google.com/>

¹³ <http://grouptweet.com/>

¹⁴ <https://fr.groups.yahoo.com/neo>

¹⁵ <https://orkut.google.com/>

¹⁶ <http://www.friendster.com/>

¹⁷ <http://fr.wikipedia.org/wiki/Anthropologie>

de vue informatique, le social computing exploite les technologies du web, des bases de données, des systèmes multi-agent, des réseaux, du multimédia et du génie logiciel.»

Soulignons l'importance des domaines de spécialisation du génie informatique comme l'intelligence artificielle, l'architecture logicielle, les réseaux, *etc.*, qui ont contribué à former le *social computing* d'aujourd'hui. La figure ci-dessous, présente l'architecture générale du *social computing*.

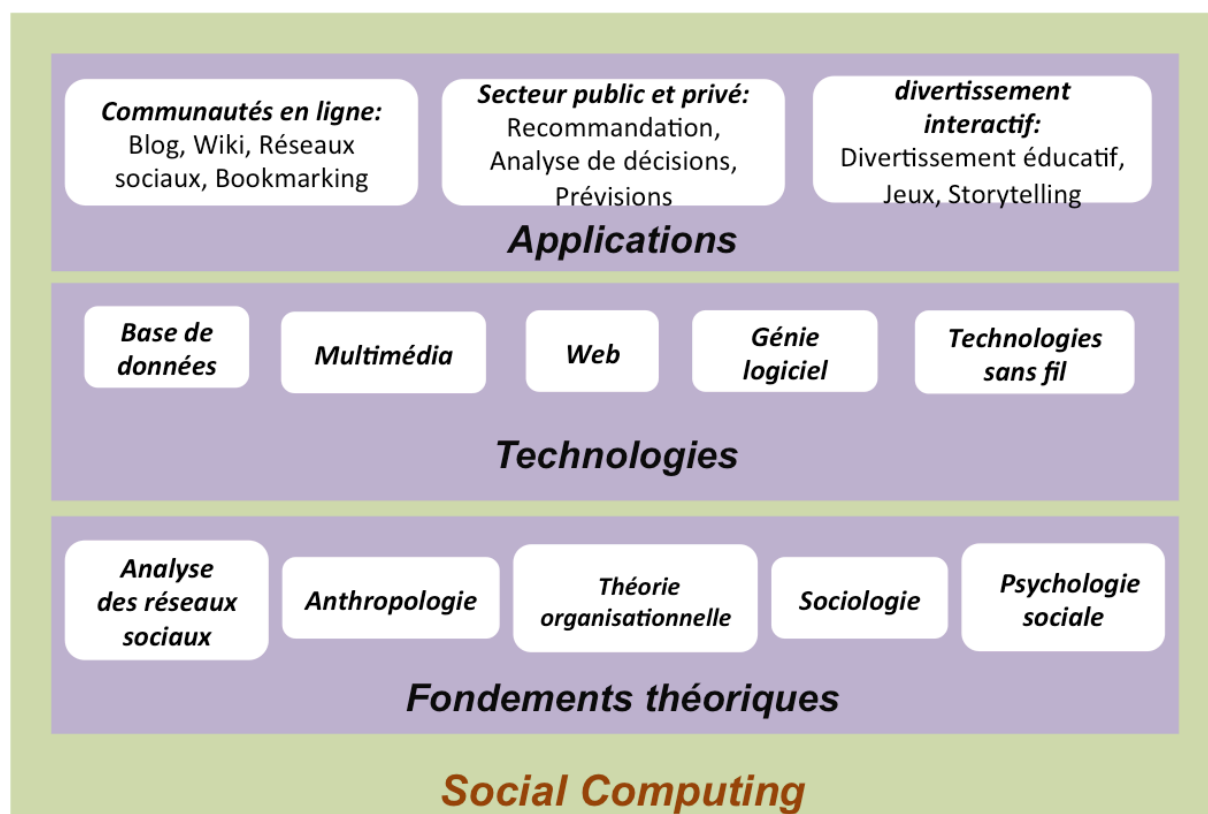


Figure 5: Architecture du social computing extraite de (Wang et al. 2007).

3.2 Technologies : de la sémantique pour le *Social Computing*

Dans cette section, nous présentons une vue générale des technologies liées au *social computing*, en revenant plus particulièrement sur l'ensemble des connaissances et des technologies qui permettent de développer l'outil communautaire que nous visons. Nous exposons l'approche de représentation sémantique des connaissances et les technologies du

web sémantique, en revenant plus particulièrement sur le vocabulaire et les ontologies qui se sont imposés dans le domaine du *social computing*.

3.2.1 Représentation sémantique des connaissances

Tim Berners Lee¹⁸, principal inventeur du World Wide Web (*WWW*), a publié un article dans *Scientific American (Burners-Lee 2010)* décrivant les réseaux sociaux comme « des jardins clos ». Chaque plateforme sociale cultive son jardin de données de façon privative. Comme tout autre domaine, les réseaux sociaux, et, plus généralement, les applications du *social computing* ont besoin de représenter sémantiquement les connaissances pour assurer l'interopérabilité et la collaboration. Les informations produites à travers les interactions sociales gagnent à être représentées selon les principes du web sémantique.

Selon le *World Wide Web Consortium (W3C)*¹⁹, organisme de normalisation des technologies du web sémantique, « le web sémantique fournit un cadre de travail commun qui permet à des données d'être partagées et réutilisées à travers l'application, l'entreprise, et les frontières de la communauté ». Le domaine du web sémantique regroupe l'ensemble des outils, des procédés et des technologies visant à rendre l'information utilisable et accessible par des programmes informatiques. En pratique, le web sémantique permet de décrire les ressources, ainsi que leurs relations à travers des marqueurs sémantiques, afin d'extraire, d'exploiter et d'interpréter le contenu. Plusieurs langages sont aujourd'hui en concurrence, comme *RDF*²⁰, *OWL*²¹, *Topic maps*²², pour décrire les différents formalismes de représentation des connaissances : les thésaurus, les taxonomies, les ontologies, *etc.* Un formalisme de représentation des connaissances est un ensemble structuré de concepts et de liens représentant le sens d'un contenu et servant à faciliter son exploitation automatique ou semi-automatique par une machine ou un être humain. Ces trois structures de données partagent l'organisation hiérarchique de concepts mais ont des usages et des objectifs différents. La taxonomie est une liste de termes contrôlés organisés de façon hiérarchique,

¹⁸ <http://www.w3.org/People/Berners-Lee/>

¹⁹ <http://www.w3.org/2001/sw/>

²⁰ <http://www.w3.org/RDF/>

²¹ <http://www.w3.org/OWL/>

²² <http://www.topicmaps.org/>

représentant les concepts d'un domaine de connaissance et permettant de faciliter la recherche de termes à partir de relations. Le thésaurus est une liste organisée de termes contrôlés et normalisés enrichie par des relations associatives et permettant de faciliter la recherche de termes à partir de différentes relations. Selon le fondateur du *W3C* (Gruber 1993), l'ontologie est « une spécification formelle explicite d'une conceptualisation partagée ». En d'autres termes, l'ontologie est un modèle de description de connaissances basé sur des concepts avec types, propriétés et relations permettant de faciliter la recherche sur différents types d'entités sémantiques (concepts, relations, propriétés).

L'échange et le partage de données entre les différents acteurs du monde du *social computing* ont créé le besoin de représenter sémantiquement et de façon uniforme les différentes entités produites, échangées et partagées. Reposer sur une sémantique commune permet, à terme, d'effectuer des recherches sémantiques de ressources. Dans ce cadre, plusieurs vocabulaires et ontologies ont été proposés pour faciliter, entre autres, la sémantisation du *social computing*.

3.2.2 Ontologies pour le social computing

L'utilisation de vocabulaires et d'ontologies de représentation des connaissances rend les données interopérables entre les applications sociales, les groupes, les personnes, les sites, *etc.* Cependant, les entités utilisées dans les applications sociales sont variées, complexes et hétérogènes. Ces entités peuvent représenter des personnes physiques ou morales, du temps, de l'espace, des services, des ressources physiques ou logicielles, du contenu, *etc.* Un certain nombre de vocabulaires du web sémantique sont déjà disponibles pour représenter un champ d'action spécifique, ses entités connexes et ses relations (*e.g. Foaf* pour les profils des utilisateurs, *Owl-Time* pour les entités temporelles, *etc.*).

Certains vocabulaires sont des standards, d'où notre intérêt pour les utiliser dans notre modèle de communautés à base d'ontologies. Assembler les différents vocabulaires dans un seul modèle permet de prendre en compte les différentes entités, composantes d'une application du *social computing*. Les organismes de normalisation comme *W3C* ont fait en sorte d'optimiser le rapprochement de ces vocabulaires.

3.2.2.1 Représentation sémantique pour les réseaux sociaux

Les réseaux sociaux sont une source riche d'informations permettant à toute personne de créer et d'échanger du contenu. Définir une ontologie qui décrit l'ensemble des objets échangés sur les réseaux sociaux permet de normaliser les échanges et d'interconnecter les "jardins clos" de manière simple et efficace. Une combinaison très utilisée pour décrire les acteurs des réseaux sociaux, ainsi que les contenus échangés, est la combinaison *Foaf*²³, *Sioc*²⁴ et *Skos*²⁵ (cf. Figure 6).

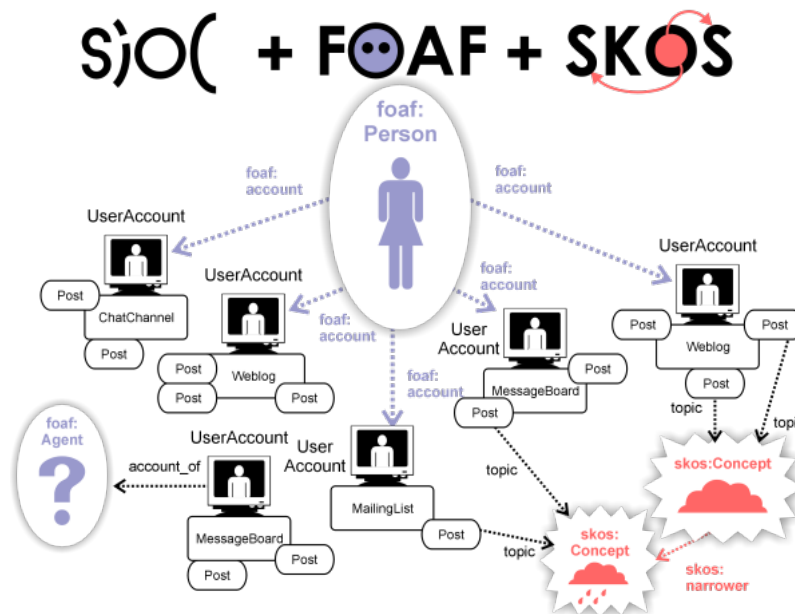


Figure 6: La combinaison *Sioc*, *Foaf* et *Skos*.

La Figure 6 illustre les principales interactions entre les trois vocabulaires, mais il existe bien d'autres ontologies qui collaborent avec celles-ci comme *Dublin Core*²⁶, *RSS*²⁷, etc.

²³ <http://www.foaf-project.org/>

²⁴ <http://www.w3.org/Submission/Sioc-spec/>

²⁵ <http://www.w3.org/2004/02/skos/>

²⁶ <http://dublincore.org/>

²⁷ <http://web.resource.org/rss/1.0/>

Sioc (Semantically-Interlinked Online Communities) est un vocabulaire permettant de décrire les concepts du web social et leurs relations (Breslin et al. 2005) . Il repose sur le langage *RDF*²⁸ (Resource Description Framework). L'ontologie principale (cf. Figure 7), extensible par d'autres modules, décrit des concepts comme *UserGroup*, *Space*, *Item*, *Container*, *Post*, *UserAccount*, *Role*, etc.

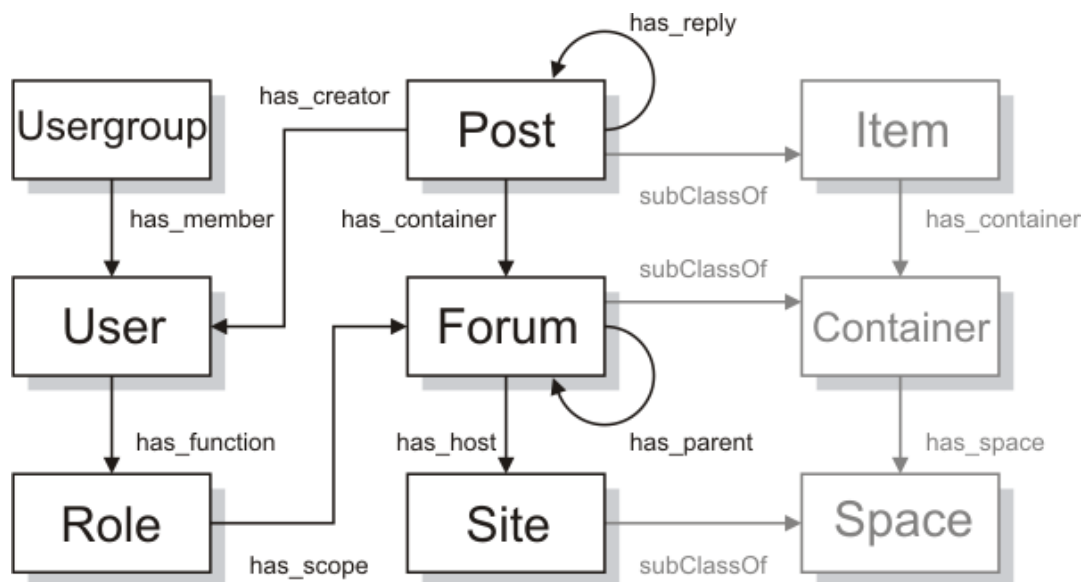


Figure 7: L'ontologie *Sioc*.

Sioc fait partie du jeu de modélisation sémantique des applications du *social computing* tout en collaborant avec d'autres vocabulaires comme *Foaf* pour gérer les utilisateurs, et *Skos* pour définir des thésaurus d'un domaine particulier.

Foaf (*Friend of a Friend*) est aussi un vocabulaire *RDF*. C'est la première brique de sémantisation des réseaux sociaux. Il permet de décrire les personnes et les liens qu'elles entretiennent entre elles. Cette ontologie (cf. Figure 8) décrit le concept de personne et les concepts connexes comme ses groupes, ses organisations, ses documents, ses contacts, etc.

²⁸ <http://www.w3.org/RDF/>

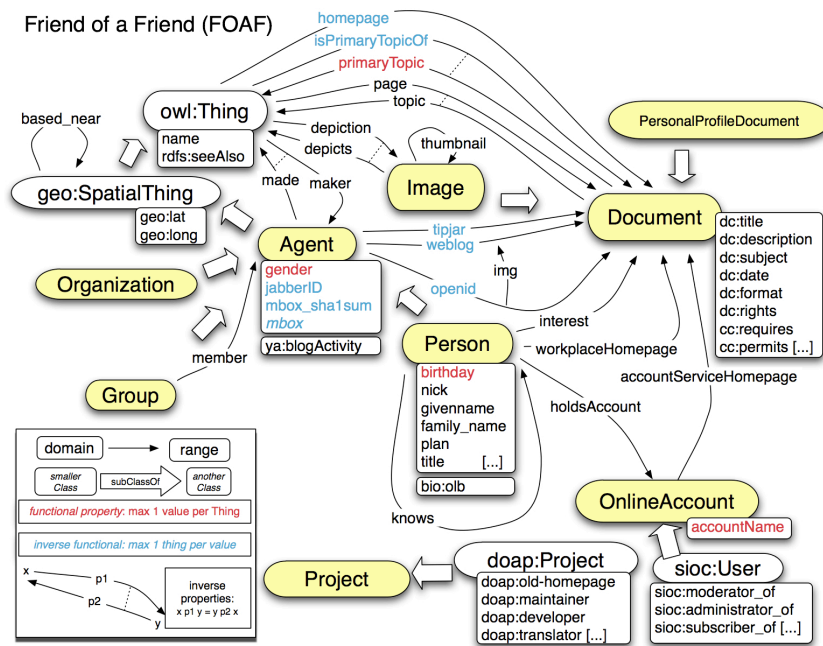


Figure 8: L'ontologie Foaf.

Dans le même cadre, *Skos* (Simple Knowledge Organization System) est un vocabulaire en *RDF* permettant de formaliser la publication des thésaurus (cf. section 3.2.1.1), des classifications ou d'autres types de formalisme de représentation. *Skos* permet d'aider à réaliser une représentation conceptuelle d'un domaine particulier utilisé par les applications sociales.

La facilité d'utilisation, ainsi que la standardisation, sont les principaux arguments qui plaident en faveur de cette combinaison de vocabulaires.

3.2.2.2 Représentation sémantique du temps

La notion de temps est fondamentale pour la description de la dimension temporelle de ressources partagées au niveau des applications sociales pour faciliter leur exploitation ultérieure.

Le vocabulaire *OWL-Time*²⁹ a été proposé pour décrire les entités temporelles et leurs propriétés pour le web sémantique. Ce vocabulaire prend en compte les deux notions suivantes : l'instant et l'intervalle. *Instant* et *Interval* représentent deux sous classes de la classe *TemporalEntity*. Un instant représente un moment infinitésimal tandis qu'un intervalle représente un espace de temps compris entre deux instants : l'instant initial et l'instant final. D'autres concepts ont été définis pour décrire la durée d'une entité temporelle *DurationDescription* et *DateTimeDescription*, les intervalles implicites, etc. De même, des propriétés-liens ont été définis pour décrire les relations entre les différents concepts de l'ontologie (*hasBeginning*, *ProperInterval*, etc.). La Figure 9 illustre les composants associés à l'ontologie *Owl-Time*.

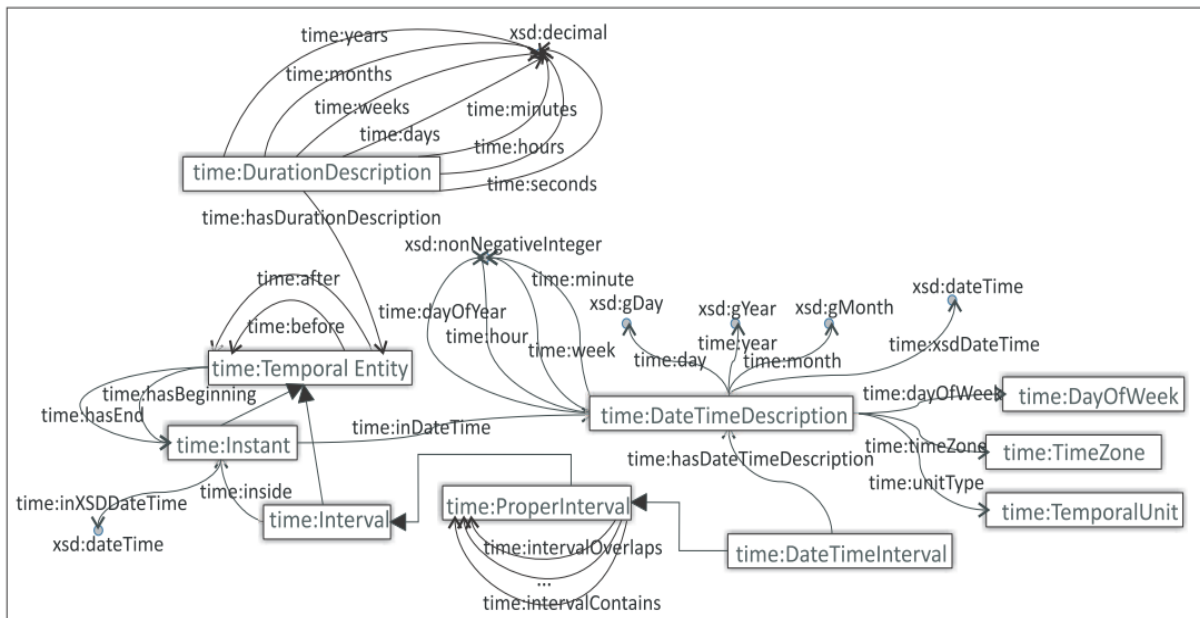


Figure 9: L'ontologie Owl-Time (Miron 2009).

Le temps et la localisation sont des entités fortement corrélées. Tout comme le temps, la représentation sémantique de l'espace a fait l'objet de plusieurs vocabulaires dédiés.

²⁹ <http://www.w3.org/TR/owl-time/>

3.2.2.3 Représentation sémantique de l'espace

Plusieurs ontologies ont été proposées pour représenter l'entité spatiale comme l'ontologie *WGS84 Geo Positioning*³⁰, un vocabulaire *RDF* qui permet de représenter des concepts spatiaux clés comme la latitude, la longitude, la localisation des objets dans l'espace. Pour des applications nécessitant des niveaux d'abstraction de l'entité spatiale plus élevés et plus sophistiqués, d'autres représentations sémantiques de l'espace ont été proposées. Parmi celles-ci *GeoRSS* est recommandée par le W3C comme vocabulaire de référence pour décrire les données spatiales. La version allégée du modèle *GeoRSS* est *GeoRSS-Simple*³¹ (cf. Figure 10) qui permet de décrire les propriétés géospatiales des objets d'une façon simple et facile à intégrer dans les applications sociales. Le concept principal de l'ontologie est *gml:_Feature* qui regroupe tous les types d'éléments spatiaux (e.g. *lat*, *long*, *floor*, *radius*, etc.) et qui est relié au concept *gml:_Geometry* qui spécifie des géométries concrètes comme *box*, *point*, *line* et *polygon*.

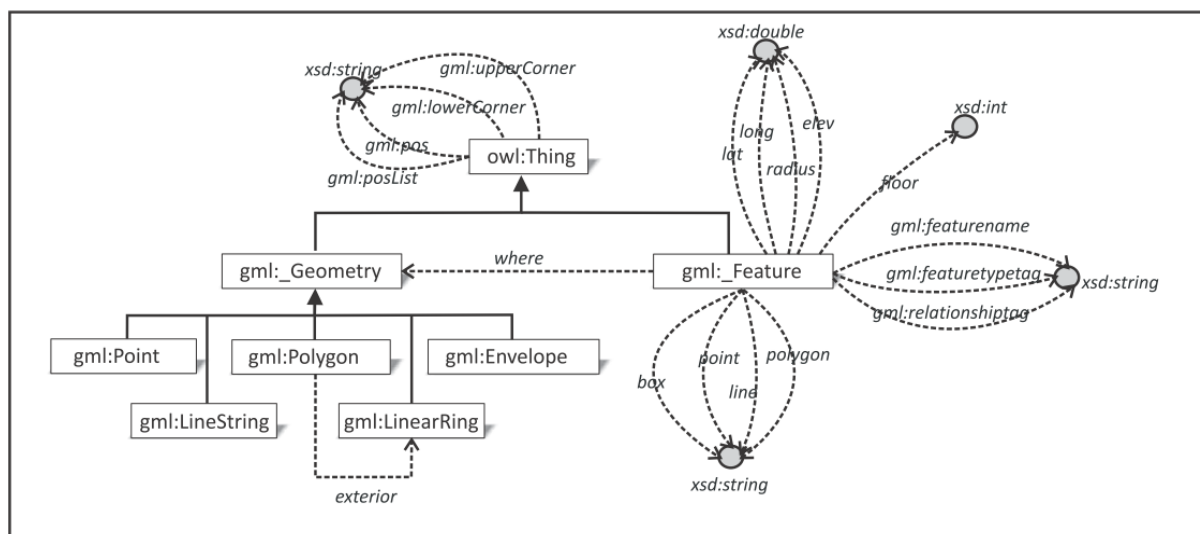


Figure 10: L'ontologie *GeoRSS-simple* (Miron 2009).

³⁰ <http://www.w3.org/2003/01/geo/>

³¹ <http://mapbureau.com:8080/neogeo/neogeo.owl>

3.2.2.4 Représentation sémantique de services

*Owl-S*³² (*Semantic Markup for Web Services*) est une ontologie pour décrire la sémantique des services, basée sur le langage *OWL*³³ (recommandation du *W3C*) et fondée sur les principes de *WSDL*³⁴ (*Web Services Description Language*). L'ontologie supérieure (*i.e. upper ontology*) de *Owl-S* fournit trois types d'information essentiels (*cf.* Figure 11) : le profil du service (*ServiceProfile*) qui décrit ses capacités, le modèle du service (*Service Model*) qui décrit son fonctionnement, et le *Service Grounding* c'est-à-dire les informations nécessaires pour accéder au service.

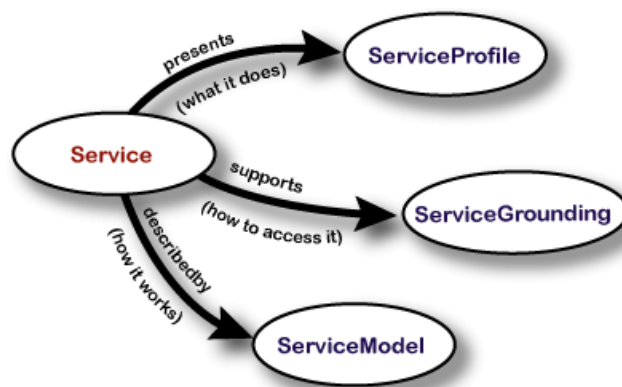


Figure 11: L'ontologie *Owl-S*.

Le profil du service contient les informations suivantes:

- Les propriétés non fonctionnelles comme le nom du service, les contacts et la description textuelle.
- Les propriétés non fonctionnelles "*IOPE*" (inputs, outputs, préconditions, effets).
- La classification selon une taxonomie industrielle et une description de qualité.

Le modèle de service contient des informations sur le modèle de processus :

- *AtomicProcess* permet d'invoquer le service et d'obtenir le résultat en une seule interaction.

³² <http://www.daml.org/services/owl-s/>

³³ <http://www.w3.org/2001/sw/wiki/OWL>

³⁴ <http://www.w3.org/TR/wsd1>

- *CompositeProcess* est une collection de processus atomiques et composites.
- *SimpleProcess* permet de fournir une vue d'un processus atomique ou une représentation simplifiée d'un processus composite.

Le *Grounding* contient des informations sur les protocoles de transport et les formats de message.

Owl-S permet de répondre à un certain nombre de problèmes des services web comme la découverte automatique des services (*i.e.* répondre aux contraintes spécifiées par le client), l'invocation automatique des services (*i.e.* invocation par un agent ou programme informatique), la composition et l'interopérabilité.

D'autres langages de description et de modélisation de services ont été proposés comme *WSMO* (Domingue et al. 2005), *WSML* (De Bruijn et al. 2006), *WSDL-S* (Cardoso et al. 2008), *SWSO* (Battle et al. 2005), *etc.* Mais *Owl-S* utilise des logiques de description pour les services qui offrent une grande puissance d'expression.

3.3 Les réseaux sociaux : un exemple d'application du *Social Computing*

Aujourd'hui, les applications du *social computing* couvrent plusieurs domaines et dimensions : des sites spécialisés dans un format multimédia spécifique (*e.g.* *Youtube* pour les vidéos, *Flickr* pour les images, *Last.fm* pour la musique), des outils de publication (blogs, wikis), des outils de discussion (forums, messageries) et des réseaux sociaux (*Facebook*, *Google+*), *etc.*

Désormais ancrés dans notre quotidien, les réseaux sociaux ne cessent d'évoluer et de varier de jour en jour pour s'adapter à de nouvelles exigences. Stéphane Dufournet définit les réseaux sociaux comme « *les différentes activités informatiques qui intègrent la technologie, l'interaction sociale, et la création de contenu* » (Dufournet 2014).

L'intérêt porté aux réseaux sociaux concerne des acteurs différents ayant différents besoins. C'est pourquoi les formes des réseaux sociaux se multiplient et se diversifient. Les réseaux sociaux peuvent être catégorisés selon leurs objectifs/usages, leurs caractéristiques et

leurs architectures. Dans cette section, nous proposons une classification des réseaux sociaux selon leurs usages et leurs architectures.

3.3.1 Les réseaux sociaux grand public

La vocation première des réseaux sociaux de masse est de connecter des personnes physiques ou morales par des liens sociaux virtuels (*e.g. Facebook*), de publier du contenu (*e.g. SlideShare*), de filtrer ce contenu (*i.e. Social News*), de gérer des marque-pages (*i.e. Social Bookmarking*), de s'adresser aux communautés spécifiques (*e.g. les réseaux sociaux professionnels*). Les réseaux sociaux destinés aux communautés à caractère professionnel sont ouverts au grand public mais pour un usage exclusivement professionnel. Ils permettent de développer les carrières et les opportunités professionnelles des individus inscrits (*e.g. Viadéo, LinkedIn*).

Certains travaux de recherche s'intéressent aux réseaux sociaux grand public pour analyser les données échangées (Abrouk et al. 2010; Al Mutawa et al. 2012) et intégrer des aspects innovants à l'instar des réseaux sociaux sémantiques proposés par (Becker et al. 2008; Chao et al. 2012).

3.3.2 Les réseaux sociaux d'entreprise

Forts de leurs succès auprès des grand public, les réseaux sociaux sont de plus en plus utilisés dans les organisations (comme les entreprises, les associations, *etc.*). Ils permettent de connecter les gens entre eux, de favoriser les échanges et de faire appliquer les bonnes pratiques au sein des communautés. Mais, les organisations s'aperçoivent qu'elles perdent une partie importante de leur capital informationnel lorsqu'elles n'exploitent pas les échanges sociaux. L'acquisition, la collection et l'analyse de l'information sociale dispersée sur les réseaux sociaux et les *clouds* sont des tâches difficiles pour les organisations étendues.

Pour toutes ces raisons, lesdites organisations choisissent de mettre en place des plateformes dont l'objectif est de faciliter l'échange et la capitalisation de l'information sociale. Plusieurs solutions ont été proposées comme les réseaux sociaux d'entreprise (*RSE*).

Les réseaux sociaux d'entreprise sont des réseaux privés, destinés à un groupe de personnes appartenant à la même organisation ou la même entreprise. « Les collaborateurs d'une organisation bénéficieront des fonctionnalités d'un réseau social d'entreprise ou d'une plateforme de collaboration, tandis que l'organisation pourra capitaliser les fragments sociaux produits. » (Deparis et al. 2011).

Les réseaux sociaux d'entreprise sont définis comme des plates-formes sociales et collaboratives permettant aux collaborateurs internes, externes et aux communautés de mieux se connaître et donc de mieux travailler ensemble. Certains travaux de recherche s'intéressent à ces outils sociaux au sein des entreprises comme la plateforme sociale de *Deparis et al.* (Deparis et al. 2014) pour améliorer la collaboration, la communication interne et la capitalisation des connaissances au sein de l'entreprise. Si l'usage des *RSE* apporte de nouveaux modes d'interaction au sein des entreprises, les *RSE* restent cependant un espace de travail collaboratif traditionnel comme les portails intranet.

3.3.3 Les réseaux sociaux basés sur la localisation

Les services basés sur la localisation sont en croissance et investissent plusieurs domaines: santé, tourisme, consommation, *etc.* Ces services permettent d'identifier la localisation d'une personne, d'un lieu, d'un objet. Ils ont permis de combler le fossé entre le monde physique et le monde virtuel en partageant les expériences du monde physique en ligne, ou en employant l'information en ligne dans le monde physique. Ces services reposent sur des technologies d'acquisition de la localisation *outdoor* (*GPS, GSM, CDMA, etc.*) ou *indoor* (*Wi-Fi, RFID, supersonic, etc.*). L'information de localisation possède plusieurs formats : points, trajectoire, région, *etc.* et elle est représentée sous plusieurs formes : absolu (Coordonnées latitude-longitude) ou relative (100 mètres au nord du stadium) ou symbolique (centre de commerce, maison, *etc.*).

La particularité des réseaux sociaux basés sur la localisation (*LBSN*) est d'intégrer des fonctions de géolocalisation permettant aux utilisateurs de se localiser (*e.g. check-in*) et de géo-localiser des objets, des amis et, même, d'organiser des rendez-vous.

Nous avons retenu la définition des LSBN proposée par (Zheng 2012) : « un réseau social basé sur la localisation ne signifie pas seulement l'ajout de la géolocalisation à un réseau social existant afin que les gens puissent partager des informations de localisation dans la structure sociale. Mais, il correspond plutôt à une nouvelle structure sociale composée d'individus avec des liens d'interdépendance dérivés de leur localisation dans le monde physique ainsi que de contenus géolocalisés (e.g. photos géolocalisées). »

Aujourd'hui, la plupart des réseaux sociaux grand public comme Facebook Places³⁵, Foursquare³⁶, Skout³⁷ intègrent des services de localisation. De nombreuses recherches se sont orientées vers les LBSN comme MOSS (un espace dynamique d'échange social basé sur des services de localisation (Zhdanova et al. 2008)) ou encore GeoLife (un réseau social basé localisation qui peut faire des recommandations et prévoir les intérêts de l'utilisateur en se basant sur sa localisation et ses liens sociaux (Zheng et al. 2010)). *Bfriend* (Smailovic et al. 2012) est parmi les premières plateformes de réseautage social *ad hoc* basées sur la localisation et le *Graphe Facebook*³⁸. L'idée clé de cette plateforme est d'envoyer à ses utilisateurs une notification lorsqu'un de leurs amis est à proximité. Nous accordons une attention particulière au travail de (Zheng 2012) qui passe en revue l'ensemble des concepts, des caractéristiques et la philosophie des LBSN.

En examinant ces plateformes basées sur les LBSN, nous constatons que la notion de temps est présente implicitement dans l'esprit des services basés sur la localisation. D'autres travaux ont mis en avant cette notion du temps pour les réseaux sociaux.

3.3.4 Les réseaux sociaux éphémères

En raison des atteintes à la vie privée sur les réseaux sociaux, les utilisateurs se sont orientés vers les réseaux sociaux éphémères (i.e. *Temporary Social Network TSN*). Sarah Evans³⁹, une consultante Social Media, définit les *TSN* comme « des réseaux sociaux dont le contenu ne reste pas en ligne d'une manière permanente. Il n'y a aucune trace, aucun lien ou

³⁵ <https://www.facebook.com/places/>

³⁶ <https://fr.foursquare.com/>

³⁷ <http://www.skout.com/>

³⁸ <https://www.facebook.com/notes/facebook/recherche-dans-le-graphe-facebook/516286921737662>

³⁹ <https://medium.com/@prsarahevens>

la possibilité de partager à nouveau. Le contenu échangé ou partagé disparaît une fois qu'il a été vu. »

Les utilisateurs, en particulier les adolescents, ne souhaitent plus que leurs données persistent éternellement en ligne. Les TSN ont connu une popularité imprévue grâce à la volatilité des données et leur convivialité d'usage, comme c'est le cas de Snapchat (Poltash 2013), Gryphn⁴⁰, Burn Note⁴¹, Wickr⁴², Silent circle⁴³, LobbyFriend⁴⁴, Fast society⁴⁵. Les chercheurs se sont intéressés à ce phénomène, en déterminant les facteurs de différenciation par rapport aux réseaux sociaux existants et en étudiant leurs caractéristiques et leurs usages.

fBomb (Northrop et al. 2013) est une application Facebook qui permet aux utilisateurs de définir les dates d'expiration des messages et des publications échangés. Cette application permet également d'étudier le comportement des utilisateurs vis-à-vis de leurs données sur les réseaux sociaux.

CALBA (Xu et al. 2013) est un *Framework* conçu pour les TSN dont l'objectif est de proposer, pour des utilisateurs mobiles dans un territoire défini, les vendeurs et les publicités adéquats en fonction de la position géographique et des préférences des utilisateurs, tout en respectant la volonté des utilisateurs d'éviter d'être surchargés par les notifications.

(Laforest et al. 2014) proposent un réseau social spontané et éphémère (RSSE) pour permettre la production collaborative de contenus numériques élaborés lors d'une manifestation, en l'occurrence d'une manifestation sportive de type course à pied. Ces RSSE reposent sur une architecture distribuée pair à pair et sur des réseaux mobiles ad hoc, et exploitent les ressources et les services offerts par les terminaux mobiles. À l'instar des réseaux sociaux traditionnels, ces RSSE permettent de mettre en relation des individus nomades en fonction de leurs affinités et/ou de leurs centres d'intérêt afin que ceux-ci puissent collaborer à la réalisation de tâches données au sein d'un RSSE.

⁴⁰ <http://armortext.co/>

⁴¹ <https://burnnote.com/>

⁴² <https://www.mywickr.com/en/index.php>

⁴³ <https://silentscircle.com/>

⁴⁴ <http://www.lobbyfriend.com/>

⁴⁵ <http://fastsociety.com/>

Le domaine des réseaux sociaux temporaires est en pleine expansion. Au-delà des éléments qui les différencient, ils offrent tous la possibilité d'avoir une expérience sociale privée et éphémère. Le principe de base de ce genre de réseau nous amène à poser les questions suivantes : comment s'assurer de la volatilité des données ? Est-ce que cette volatilité donne un faux sentiment de sécurité pour les utilisateurs sachant qu'il existe toujours des failles (e.g. faire des captures d'écran) ?

3.3.5 Architecture des réseaux sociaux

Plusieurs chercheurs ont proposé de nouvelles architectures décentralisées répondant ainsi aux problèmes des réseaux sociaux centralisés (e.g. *Elgg*⁴⁶, *Noosfero*⁴⁷), énumérés dans des travaux comme (Yeung et al.) et (Carpenter 2012), comme le non contrôle des données et l'atteinte à la vie privée.

Pour surmonter les limites des réseaux sociaux centralisés, plusieurs solutions ont été proposées dont notamment les réseaux sociaux décentralisés ou distribués. Cette architecture permet d'interroger et de mettre à jour des sources de données réparties. L'architecture distribuée permet de renforcer la sécurité des données en les répartissant sur plusieurs serveurs. Plusieurs travaux ont tenté de proposer une architecture distribuée pour les réseaux sociaux comme *LifeSocial.Kom* qui est une plateforme sociale *P2P* qui offre les mêmes fonctionnalités que les réseaux sociaux communs (Graffi et al. 2011). *PeerSoN* (Buchegger et al. 2009) est une approche *P2P* couplée à une approche de cryptage pour assurer une sécurité maximale des données échangées. Tribler (Pouwelse et al. 2008) est un système de partage de fichiers *P2P* qui exploite les liens sociaux pour la découverte et la recommandation de contenus. Une architecture libre, distribuée et basée sur les technologies sémantiques. *An architecture of a distributed semantic social network* est proposée par (Tramp et al. 2011) pour répondre aux limites des réseaux sociaux centralisés.

En revanche, aujourd'hui, le pourcentage d'utilisation des réseaux sociaux version mobile ne cesse de croître. Les réseaux sociaux mobiles attirent 98% des utilisateurs

⁴⁶ <https://www.elgg.org/>

⁴⁷ <http://noosfero.org/>

d'*Instagram*, 86% des utilisateurs de *Twitter*, et 68% des utilisateurs de *Facebook*. Les réseaux sociaux mobiles (i.e *native social mobile*) permettent de développer des interactions sociales et d'échanger des données en situation de mobilité. Selon *Monica Lam*, directrice du laboratoire *MobiSocial* de *Stanford*, (Lam 2013), les réseaux sociaux mobiles commencent à perturber les réseaux sociaux classiques. La force de cette architecture alternative des réseaux sociaux s'inscrit dans un contexte de mobilité marqué par l'utilisation des technologies mobiles comme la téléphonie, les services basés sur la localisation et la réalité augmentée. De plus en plus, la frontière entre les réseaux sociaux mobiles et les réseaux sociaux version web devient floue car les applications mobiles utilisent les réseaux sociaux classiques pour découvrir du contenu et des liens d'amitié, et, inversement pour profiter des fonctionnalités mobiles.

Dans cette section, nous avons présenté une classification des réseaux sociaux selon leurs usages et leurs architectures. D'autres classifications des réseaux sociaux ont été proposées comme celle d'Alain Lefebvre (Lefebvre 2005), cette classification organise les réseaux sociaux en deux catégories : les réseaux sociaux implicites construits à la base autour des contenus et les réseaux sociaux explicites construits autour des utilisateurs.

3.4 Conclusion

Ces dernières années, la problématique du *social computing* a suscité des recherches académiques et industrielles. Après avoir situé notre recherche dans le cadre du *social computing* comme un domaine clé à l'articulation entre une réflexion sur les principes des sciences sociales et les défis de l'informatique, nous avons présenté l'architecture du *social computing* composée de trois couches : les fondements théoriques, les technologies utilisées et les applications. Par ailleurs, nous avons plus particulièrement focalisé notre attention sur certaines technologies et applications qui nous ont inspirées pour la conception de notre outil communautaire. En ce qui concerne les technologies qui nous intéressent, nous avons insisté sur les technologies du web sémantique. Concernant les applications liées au *social computing*, nous avons porté un intérêt particulier aux réseaux sociaux sous leurs différentes formes et spécialement aux réseaux sociaux temporaires et aux réseaux sociaux basés sur la localisation. Les réseaux sociaux connaissent depuis plusieurs années une croissance

continue, en parallèle le nombre de communautés en ligne connaît une croissance exponentielle. On perçoit parfois une absorption du communautaire par les réseaux sociaux ou une orientation des communautés vers le réseautage social. On constate très nettement que la ligne de séparation entre les réseaux sociaux et les applications communautaires est devenue de plus en plus floue. Nous présentons dans la Table 3, une liste des principales différences entre les réseaux sociaux et les applications communautaires.

	Réseaux sociaux	Applications communautaires
Partage d'information	sans attente de retour.	attente de retour.
Nature de communauté	communautés implicites.	communautés explicites.
Ressemblance entre les membres	peuvent être très différents et sans points communs.	partagent un intérêt commun.
Type de relations	relations implicites déduites à partir du comportement des membres.	relations explicites déterminées par les membres.
Capitalisation de l'information sociale	non	oui
Structure	en réseau	chevauchement
Exemples d'applications	<i>Social News, Social bookmarking, etc.</i>	<i>Foursquare, Yuback, etc.</i>

Table 3: Tableau comparatif entre les réseaux sociaux et les applications communautaires.

L'impact des évolutions technologiques se manifeste par la tendance des interactions sociales à s'orienter vers la personnalisation, l'ubiquité, *etc.* Les utilisateurs sont aujourd'hui confrontés à de nouveaux défis comme l'ubiquité, l'utilisation de dispositifs mobiles hétérogènes (taille d'écran, modes d'interaction, mémoire, batterie, interface réseau, *etc.*), le changement du contexte général d'utilisation (changement de position géographique, contrainte d'énergie, bande passante et type de connexion), *etc.* La réflexion menée en matière de *social computing* peut par ailleurs être rattachée à celle portant sur le *pervasive computing*. Ce domaine a bouleversé le *social computing* en instaurant une nouvelle ère pour les applications sociales qui a provoqué la fusion des deux domaines et fait émerger le *pervasive social computing*. Ce dernier est abordé dans le chapitre suivant.

Chapitre 4: Vers une convergence entre le Social Computing et l'informatique pervasive

Depuis quelques années, des évolutions importantes ont eu lieu en matière d'infrastructures technologiques. En particulier, la démocratisation des dispositifs mobiles (comme les PCs, *Smartphones*, Tablettes, etc.) qui a rendu l'information accessible par le grand public à tout moment et en tout lieu, ce qui est l'origine du concept d'informatique ubiquitaire. Mark Weiser a défini l'informatique ubiquitaire comme « un monde où les gens sont entourés de terminaux informatiques interconnectés via des réseaux qui nous aident dans tout ce que nous entreprenons » (Weiser 1994). Dans des environnements fortement changeants et hétérogènes, les applications ubiquitaires progressent au détriment des systèmes traditionnels (*cf.* Figure 12). L'adaptation, la personnalisation et la transparence de ce genre d'application leur confère un atout significatif par rapport aux applications distribuées ou mobiles.

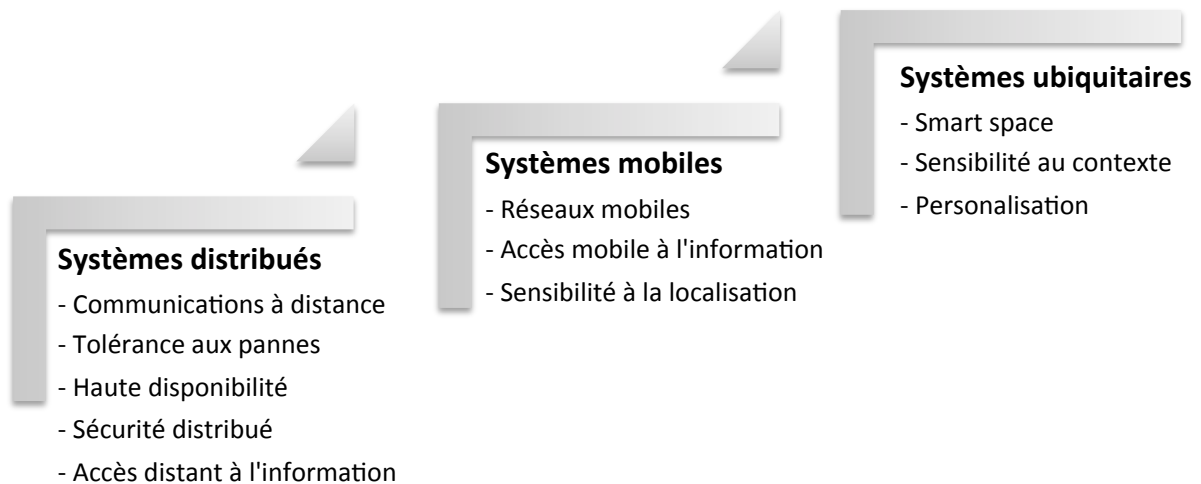


Figure 12: L'évolution des systèmes distribués vers les systèmes ubiquitaires

(Satyanarayanan 2001).

Aujourd'hui, les applications du *social computing* sont impactées de plus en plus par l'informatique ubiquitaire. Ce domaine a bouleversé le *social computing* en instaurant une

nouvelle ère des applications sociales, d'où la fusion des deux domaines qui a fait émerger le *pervasive social computing*. Les personnes sont de plus en plus connectées grâce aux dispositifs mobiles. Elles participent aux applications sociales partout et à tout moment et sont à la recherche d'interactions sociales avec d'autres utilisateurs dans un environnement ubiquitaire.

La notion d'ubiquité des applications sociales vise les problématiques des interactions sociales dans un environnement caractérisé par la mobilité, des dispositifs hétérogènes et un contexte changeant. Les applications dans un environnement ubiquitaire se reconfigurent selon les changements du contexte. La sensibilité au contexte est devenue le pilier principal de l'informatique ubiquitaire. Apporter de la sensibilité au contexte aux applications sociales enrichit les informations sur la situation de l'utilisateur et ses interactions sociales.

Ce chapitre introduit la notion de sensibilité au contexte comme la préoccupation des applications sociales, plus particulièrement, et toutes les applications en général dans un environnement ubiquitaire. Dans un second temps, les principales plateformes de sensibilité au contexte sont présentées dans la section 4.2. Ensuite, la section 4.3 présente l'utilité de la sensibilité au contexte pour les applications sociales. Finalement, la dernière section présente un état de l'art des différents travaux existants ramenant l'ubiquitaire au monde du social avec une analyse des points forts et des points faibles de chaque approche.

4.1 Sensibilité au contexte

Le contexte joue un rôle-clé dans l'informatique ubiquitaire. De nombreuses définitions ont été proposées. Celle qui correspond le mieux à notre vision est celle de Dey (Dey 2001) qui a défini le contexte comme « *toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité est une personne, un endroit ou un objet qui peut être pertinent pour l'interaction entre un utilisateur et une application, y compris l'utilisateur et l'application.* ».

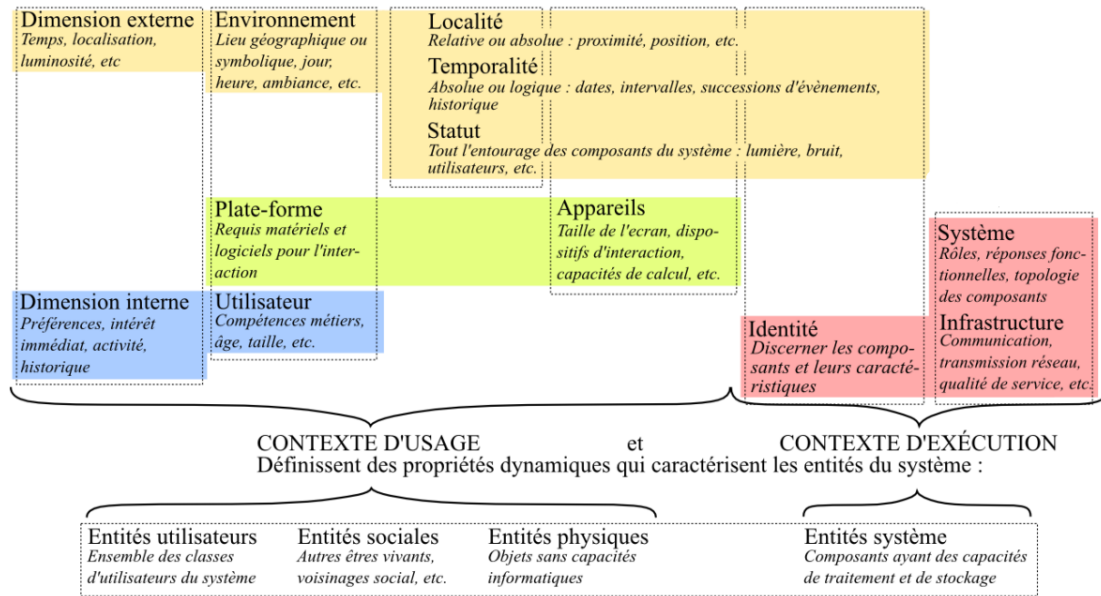


Figure 13: Vue d'ensemble des éléments du contexte (Petit 2010).

La Figure 13 illustre une taxonomie des éléments du contexte proposée par (Petit 2010) qui a classé les éléments du contexte en deux sous-ensembles : le contexte d'usage et le contexte d'exécution. **Le contexte d'usage** regroupe les entités utilisateurs, les entités sociales (y compris les communautés), et les entités physiques. **Le contexte d'exécution** représente l'espace du système incluant l'ensemble de ses composants et l'infrastructure de communication.

En s'appuyant sur la définition de Dey (Dey 2001) du contexte, plusieurs définitions des applications sensibles au contexte ont été proposées comme celle de (Taconet 2011) : « *les applications sensibles au contexte sont des applications dont la structure et le comportement varient en fonction du contexte. Elles utilisent les observations de contexte pour offrir des informations ou des services pertinents pour l'utilisateur.* ». L'étude des différentes définitions de sensibilité au contexte nous a permis de déduire qu'un système sensible au contexte (*i.e.* appelé adaptatif) doit collecter des informations contextuelles pour viser au moins une des trois fonctions suivantes :

- Présenter des informations et des services pertinents à l'utilisateur.

- Adapter son comportement et exécuter automatiquement des services sans l'intervention de l'humain.
- Associer à l'information stockée des données liées à son contexte d'utilisation.

Le domaine de sensibilité au contexte a pris essentiellement deux voies : **l'acquisition du contexte** et **l'adaptation**. La première voie est celle de la capture, de l'interprétation, de la modélisation et du stockage du contexte. La deuxième voie est celle de l'adaptation et du changement du comportement de l'application en fonction du contexte.

Les trois principales raisons pour lesquelles il est nécessaire de considérer des systèmes qui changent de structure et de comportement (appelé adaptatifs) sont :

- remplacer ou optimiser un module d'application,
- adapter l'application au changement du contexte d'exécution, et
- étendre l'application en ajoutant de nouveaux modules.

Des travaux comme celui de (Sancho 2010) proposent une taxonomie de l'adaptation qui énumère les différents mode d'adaptation dans les approches existantes :

- Prise de décisions statique *versus* dynamique : c'est-à-dire les changements de comportement sont prédéfinis dans le code ou inférés via des règles externes.
- Adaptation interne *versus* externe : soit la politique d'adaptation est intégrée dans l'application (*i.e.* dans la logique applicative) soit elle est gérée par un gestionnaire d'adaptation qui surveille et prend les décisions d'adaptation.
- Adaptation basée sur les modèles *versus* adaptation sans modèles : s'appuyer sur des modèles d'observation du contexte et des modèles pour l'adaptation permet de faciliter la gestion de la sensibilité au contexte. Les approches dirigées par les modèles permettent de réduire l'effort de programmation de différentes versions de la même application en proposant des outils de génération de code. Autrement, les informations nécessaires à la sensibilité au contexte existent dans le code (*i.e.* adaptation lors du développement) avant la mise en œuvre de l'application en générant plusieurs versions d'une même application.

- Adaptation réactive *versus* proactive : l'adaptation réactive agit dès qu'un changement de contexte survient, alors que l'adaptation proactive agit sur une prédiction de changement de contexte.
- Adaptation du contenu et adaptation de la présentation : pour la première, on change le contenu en fonction de l'utilisateur (*e.g.* proposer des dessins animés pour les enfants et des films pour les adultes), et pour le deuxième type d'adaptation, on change la façon de présenter le contenu (*e.g.* présenter une vidéo en mode audio description pour les non-voyants). A l'origine, l'adaptation du contenu et l'adaptation de la présentation (*i.e.* appelée adaptation du système d'information) sont apparues pour résoudre le problème de surcharge cognitive où l'utilisateur est perdu dans l'ensemble des informations qui lui sont proposées. L'objectif de ce genre d'adaptation du contenu est d'améliorer la flexibilité des systèmes d'informations. Dans la littérature, les systèmes optant pour l'adaptation du contenu et de présentation sont appelés systèmes d'adaptation d'informations.
- Adaptation de fonctionnalités : qui consiste à adapter dynamiquement les fonctionnalités (*i.e.* les services) d'une application en fonction de son contexte d'usage.
- Adaptation comportementale et adaptation structurelle : l'adaptation comportementale consiste à changer le comportement des composants d'un système lors de l'exécution, c'est-à-dire adapter le fonctionnement interne du composant en fonction du changement. En revanche, l'adaptation structurelle consiste à changer la composition (*e.g.* le nombre, la nature et la connexion des composants) d'un système lors de son exécution.

Dans notre travail, nous nous intéressons à l'adaptation à différents niveaux :

- L'adaptation orientée utilisation (*i.e.* adaptation du contenu et des fonctionnalités) qui dépend des communautés auxquelles il est connecté.
- L'adaptation sensible aux contraintes de l'application et des périphériques (*i.e.* adaptation structurelle) qui dépend du contexte d'exécution de l'application.

La section qui suit décrit les principales plateformes de sensibilité au contexte.

4.2 Principales plateformes de sensibilité au contexte

La particularité des systèmes sensibles au contexte est qu'ils réagissent aux différents changements de ce dernier afin de fournir à l'utilisateur une application adaptée à son contexte. Conformément à sa définition, *une application sensible au contexte ou adaptative* doit pouvoir acquérir, enrichir, exploiter et s'adapter au contexte. Étant donné la lourdeur de ces tâches et la difficulté de réutiliser ces mécanismes ou de les étendre, des architectures, appelées intergiciels pour l'ubiquitaire, ont été proposées. Ces intergiciels intègrent les principales fonctions de sensibilité au contexte (capture, acquisition, raisonnement, adaptation d'application, *etc.*) lors du déploiement de l'application ou pendant son exécution. Ces intergiciels permettent de créer des applications à base de composants (regroupant un certain nombre de fonctionnalités). Un composant est une unité logicielle qui possède un état constitué par les données qu'il traite, une implantation qui est le code réalisant ce traitement et des interfaces fournies et requises qui sont des ports d'entrée/sortie pour interagir avec le monde extérieur. Cette unité logicielle est liée à une infrastructure (conteneur ou *Framework*) qui lui permet d'échanger des messages avec d'autres composants. Parmi les infrastructures à composants les plus utilisées, on trouve : *CORBA*⁴⁸, *OSGi*⁴⁹, Enterprise Java Beans⁵⁰, *Fractal du consortium Object Web*⁵¹ et *Kalimucho* (Da et al. 2014). La programmation orientée composants est apparue pour répondre aux limites de la programmation orientée objets comme la difficulté de réutilisation du code en raison de la granularité fine des objets. La programmation par composants sépare l'interface et l'implémentation du composant, ce qui permet son encapsulation et sa réutilisation.

Pour construire des applications à base de composants sensibles au contexte, plusieurs plateformes (appelées aussi *middleware*) ont été proposées. Ces plateformes permettent de recevoir depuis des capteurs de l'environnement des informations sur le contexte d'usage de l'utilisateur et sur le contexte d'exécution de l'application, de conduire des raisonnements et de prendre les décisions d'adaptation. Comme le montre la Figure 14, la prise en charge de la

⁴⁸ <http://www.corba.org/>

⁴⁹ <http://www.osgi.org/Main/HomePage>

⁵⁰ <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>

⁵¹ <http://fractal.ow2.org/documentation.html>

sensibilité au contexte peut être définie par trois fonctions : l'observation du contexte, l'adaptation lors du déploiement, et l'adaptation lors de l'exécution.

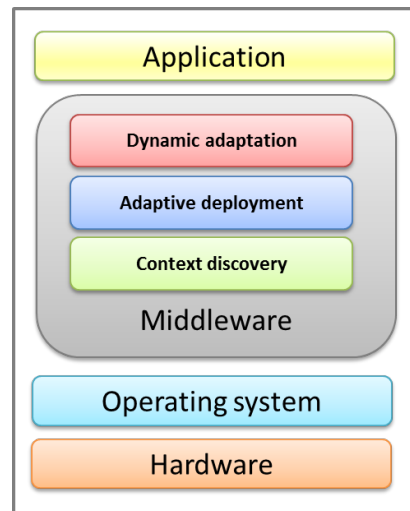


Figure 14: Architecture des systèmes sensibles au contexte.

Dans ce qui suit, nous présentons des exemples de plateforme pour chacune des trois fonctions.

4.2.1 Plateforme d'observation de contexte

CA3M (Taconet 2011) (*Middleware for Context awareness based on a MetaModel*) est un intergiciel d'observation de contexte qui propose les fonctionnalités suivantes :

- Il aide le concepteur de l'application ubiquitaire dans la spécification des éléments du contexte à observer à partir de différentes sources.
- Il permet de prendre en charge les opérations de gestion de contexte et fait l'interface entre un gestionnaire de contexte et l'application ubiquitaire.

Cet intergiciel a été testé avec le gestionnaire de contexte COSMOS (Conan et al. 2007) pour une interprétation de contexte.

4.2.2 Plateformes de déploiement adaptatif

CADeComp (Taconet 2011) est un intergiciel pour un déploiement sensible au contexte d'une application à base de composants. Cette plateforme assure l'adaptation du déploiement au contexte grâce à un ensemble de composants adaptatifs et un modèle de données permettant la définition de la variabilité des paramètres de déploiement en fonction des informations du contexte. Le déploiement des composants de l'application varie selon le type des terminaux, l'assemblage des composants et le placement de ces derniers. La collecte des données de contexte et la prise de décision de l'adaptation permettent de construire un plan de déploiement qui décrit l'architecture de l'application à déployer, ses instances de composants, leur implémentation, leur placement, leur dépendance et les connexions à mettre en place.

D'autres propositions ont été faites comme OTA- 3DMA (Fjellheim 2006) et AxSel (Hamida et al. 2008).

4.2.3 Plateforme d'adaptation dynamique

MUSIC (Rouvoy et al. 2009) est un intergiciel permettant la reconfiguration d'applications mobiles en fonction du contexte. *MUSIC* se base sur l'adaptation par planification qui « réfère à la capacité de reconfiguration d'une application en réponse aux changements de contexte en exploitant les connaissances de sa composition et des métadonnées de QoS associées à chacun des services la constituant » (Rouvoy et al. 2009). Le processus d'adaptation est réalisé par un gestionnaire d'adaptation qui repose sur la collaboration de six services : Adaptation Reasoner, Adaptation Controller, Template Builder, Configuration Executor, Configuration Planner, Configuration Controller. *MUSIC* repose sur la conception d'un modèle de déploiement pour chaque configuration.

Un autre intergiciel concurrent *Wcomp* (Tigli et al. 2009) permet de construire des applications sous forme de graphes de services web.

Nous avons cité précédemment les exemples les plus connus d'intergiciels de sensibilité au contexte. Nous présentons un tableau récapitulatif de la classification de ces différentes plateformes (*cf.* Table 4).

	Intergiciel d'observation du contexte	Intergiciel de déploiement adaptatif	Intergiciel d'adaptation dynamique
CA3M (Taconet 2011)	X		
CADeComp (Taconet 2011)		X	
OTA- 3DMA (Fjellheim 2006)		X	
AxSel (Hamida et al. 2008)		X	X
MUSIC (Rouvoy et al. 2009)	X	X	X
Wcomp (Tigli et al. 2009)		X	X
Kalimucho (Da et al. 2014)	X	X	X

Table 4: Classification des plateformes de sensibilité de contexte.

Nous avons proposé cette classification car certaines plateformes ont été proposées pour répondre à une fonction spécifique de sensibilité de contexte (*i.e.* découverte du contexte, déploiement adaptatif, adaptation dynamique), mais cela n'empêche pas qu'elles peuvent réaliser d'autres fonctions. Ces différentes approches adoptent différents modes d'adaptation (*e.g.* adaptation basée sur l'architecture, sur le paramétrage ou sur les aspects) et différentes techniques. Le lecteur est invité à consulter (Da 2014) pour des descriptions détaillées de ces différentes plateformes de sensibilités de contexte .

Une partie de nos travaux se base sur les techniques développées dans le domaine de l'informatique ubiquitaire, et, plus particulièrement, sur l'adaptation dynamique des applications sociales. Dans la section suivante, nous évoquons l'utilité de l'adaptation pour les applications sociales en exposant des exemples d'utilisation pour définir l'aspect général des applications sociales adaptatives.

4.3 Utilité de sensibilité au contexte pour le *Social Computing*

Les attentes en termes de mobilité, de sensibilité au contexte, d'accès omniprésent à l'information ont bouleversé le domaine du *social computing*. Les interactions sociales ont

beaucoup évolué depuis l'avènement du *pervasive social computing*, non seulement l'émergence d'une nouvelle façon de communiquer, de collaborer, de coordonner, et de s'organiser mais aussi des conséquences importantes sur les entités sociales (individu, groupe, organisation) (cf. Table 5). Certains travaux comme (Jessup et al. 2002) étudient l'impact de l'informatique ubiquitaire sur les entités sociales et inversement (Coutand et al. 2005).

Granularité	Les enjeux du <i>pervasive social computing</i>
Individu	-Quel est l'impact de l'informatique ubiquitaire sur les normes du social ? -Y'aura t'il une apparition de nouvelles formes d'interaction sociales ?
Groupe	-Quel est l'influence de l'ubiquitaire sur les communautés ? -Comment les interactions sont redéfinies entre membres de groupes?
Organisation	-Le modèle d'entreprise est-il influencé par l'informatique pervasive ? -Améliore-t-il la communication au sein de l'organisation ?

Table 5: Les enjeux de l'informatique ubiquitaire sur les entités sociales (Jessup et al. 2002).

La Table 5 présente les enjeux de l'informatique ubiquitaire sur les différentes entités sociales et identifie les points d'impact que nous abordons dans la suite de ce chapitre.

Nous avons montré dans la section 4.1 que la sensibilité au contexte consiste en la découverte du contexte de l'utilisateur afin d'adapter les applications à ces attentes. La considération du contexte pour les applications sociales intervient :

- i) au niveau du contenu proposé ;
- ii) au niveau de la structure et du comportement de l'application.

Nous avons constaté que certaines informations du contexte peuvent être utilisées pour assister l'utilisateur dans ces échanges sociaux (par exemple identifier automatiquement un ami présent sur une photo au lieu d'introduire manuellement l'information par l'utilisateur) et

que l'adaptation peut améliorer la structure et le comportement de l'application (par exemple lorsque la bande passante devient insuffisante, basculer d'une visioconférence vers une audioconférence).

4.3.1 L'utilisation du contexte pour assister les échanges sociaux

Joly *et al.* (Joly et al. 2009) pensent que l'utilisation de certaines informations du contexte peut aider ou remplacer des saisies manuelles des informations de contexte comme la localisation qui peut aider à entamer une communication.

"We believe that sharing some of context' information can replace, or at least assist, manual entry of shared contextual artifacts that are socially useful to start communications".

Exploiter des informations du contexte permet de stimuler les échanges sociaux en les annotant et favorise la communication. Parmi les exemples les plus connus, on cite les applications qui proposent des recommandations basées sur l'avis des amis qui ont visité les mêmes lieux (bons plan, restaurants, sorties, hébergement, shopping, etc.).

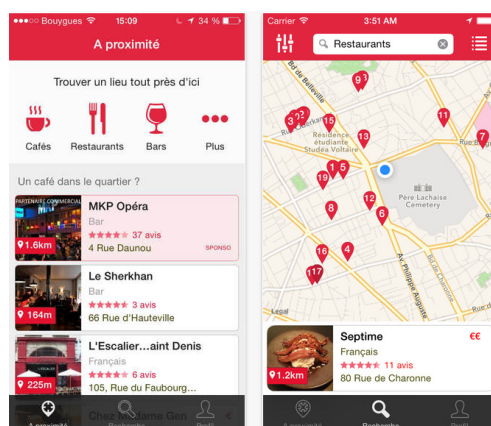


Figure 15: Capture d'écran de l'appli Cityvox pour la recommandation de restaurant.

L'application Cityvox (cf. Figure 15) permet de recommander des restaurants selon la position géographique de l'utilisateur parmi les restaurants commentés par la communauté Cityvox. Elle facilite l'accès au contenu (liste des restaurants) juste en exploitant une information de contexte (position GPS). Un autre exemple est l'application Tinder, une application de rencontres basée sur la géolocalisation qui offre des opportunités de communiquer et de rencontrer des personnes.

Outre l'aide à la découverte des environs et l'établissement de nouvelles relations pour améliorer l'expérience sociale de l'utilisateur, la sensibilité au contexte permet aussi d'améliorer le comportement et la structure de l'application sociale.

4.3.2 *L'utilisation du contexte pour améliorer les applications sociales*

La sensibilité au contexte peut améliorer les interactions entre l'utilisateur, son environnement et l'application. L'adaptation de l'application sociale s'applique sur les fonctionnalités, la structure et le comportement. En ce qui concerne les fonctionnalités de l'application, prenons l'exemple d'une application à propos des Fêtes de Bayonne. Les services de l'application proposés à l'utilisateur sont déduits à partir de sa position géographique et de ses centres d'intérêt. Si l'utilisateur passe près d'un spectacle de rue de tauromachie (lâcher de taureaux), des invitations à la prudence sont émises. Dans le cas où il passe par un spectacle musical, un service d'enregistrement audio lui est proposé. Si l'utilisateur décide de faire une vidéoconférence avec un de ses amis pour lui faire écouter de la musique, mais qu'un problème de bande passante surgit en pleine communication, le service de vidéoconférence sera remplacé par un service d'audioconférence pour assurer la continuité de la communication.



Figure 16: Fêtes de Bayonne (spectacle musical, un spectacle tauromachique).

Nous avons montré les deux aspects (information et application) sur lesquels peut agir la sensibilité au contexte dans le domaine du *social computing*. Dans la section qui suit, nous présentons les différents travaux tentant d'introduire l'ubiquitaire dans les applications sociales.

4.4 Pervasive Social Computing

Le terme *Pervasive Social Computing* a été introduit par (Ben Mokhtar et al. 2009). Mais il apparaît clairement qu'il n'existe pas une définition de référence pour ce domaine. Une initiative a été proposée par (Zhou et al. 2012) pour le définir comme :

“Un moyen pour augmenter l'intelligence sociale de l'Homme afin lui présenter les informations sociales souhaitées (personnes, paramètre de communication, etc.) durant son interaction avec l'environnement physique. “

Le *Pervasive Social Computing* regroupe cinq domaines de recherche : le traitement du signal social, les interactions multimodales, les réseaux sociaux, les médias sociaux, et l'informatique pervasive. Son objectif principal est d'améliorer l'expérience sociale des utilisateurs à travers leurs interactions, leur conscience de groupes, leurs échanges de contenus, leurs comportements sociaux, dans un environnement ubiquitaire.

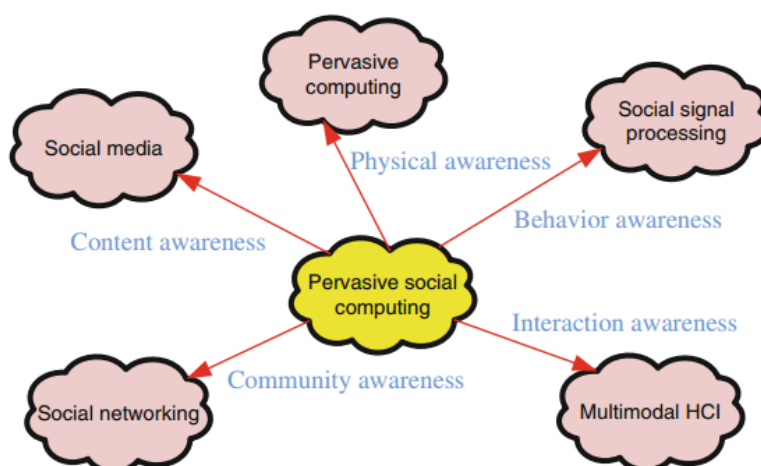


Figure 17: L'architecture du *Pervasive Social Computing* (Zhou et al. 2012).

En rapprochant le contexte d'exécution (cf. section 4.1) du contexte social qui correspond aux entités sociales (utilisateur, communautés, organisations, etc.), le contexte social pervasif peut être défini comme :

“le context socio-pervasif est l'ensemble des informations résultats de l'interaction directe ou indirecte avec des personnes équipées des appareils omniprésents avec des capteurs connectés au même service de réseau social”. (Schuster et al. 2013)

Le contexte social pervasif est caractérisé par quatre dimensions : temporalité, localité, entité sociale, entité système d'où la taxonomie STiPI qui a été proposée pour classifier les éléments de ce contexte (Schuster et al. 2013) selon les dimensions citées.

Depuis son apparition, plusieurs applications ont été proposées pour valider l'approche *pervasive social computing* comme *Foursquare*⁵², *Tinder*⁵³, etc. Dans le cadre de la recherche académique, le *pervasive social computing* fait l'objet d'une attention grandissante. La sensibilité au contexte est utilisée dans les applications sociales pour adapter le contenu proposé à l'utilisateur et assister ses interactions sociales d'un côté, adapter l'application pour répondre aux contraintes physiques de l'autre. Nous présentons dans ce qui suit, un état de l'art des principaux travaux qui ont combiné le social et l'ubiquitaire, et dont l'objectif est d'adapter le contenu ou d'adapter le comportement et la structure de l'application.

Parmi ces travaux, se trouvent les travaux présentés dans les sections précédentes comme *CALBA* (Xu et al. 2013) (cf. section 3.3.4), *Bfriend* (Smailovic et al. 2012) et *GeoLife* (Zheng et al. 2010) (cf. section 3.3.3) pour adapter le contenu proposé à l'utilisateur selon sa localisation et *MobiLife* (Coutand et al. 2005) (cf. section 2.2) pour adapter les services proposés en fonction du contexte social de l'utilisateur.

Les travaux de (Quercia et al. 2010) proposent un algorithme de recommandation des événements sociaux dans une zone urbaine spécifique tout en exploitant les informations du contexte fournis par le Smartphone comme la position géographique.

Quant à l'application *Whozthat* (Beach et al. 2008) elle utilise la proximité des dispositifs mobiles pour échanger les identifiants des réseaux sociaux au niveau local, à travers la technologie sans fil et les réseaux sociaux en ligne. Une fois que le système identifie l'utilisateur, il peut adapter le contenu en fonction de l'identité et des préférences de la personne ou même en fonction des préférences *de son groupe*.

⁵² <https://fr.foursquare.com/>

⁵³ <http://www.gotinder.com/>

Sur le même principe, MobiClique (Pietiläinen et al. 2009) utilise une approche opportuniste⁵⁴ afin de développer un réseau social mobile dont les échanges entre utilisateurs s'effectuent par le moyen des technologies Bluetooth et/ou Wi-Fi. Chaque nœud possède un profil qui contient les informations personnelles à rendre publiques. Les fonctionnalités principales du système sont : la découverte du voisinage, l'identification des nœuds présents et les échanges opportunistes de données avec les destinataires choisis.

On peut distinguer les travaux qui utilisent la sensibilité au contexte pour adapter le contenu comme :

(Menkens 2009) qui présente une plateforme pour applications communautaires regroupant différentes informations du contexte (localisation, temps, communauté, profil de l'utilisateur), de différentes sources (*smartphones*, portée radio, *GPS*), afin de permettre à l'utilisateur de communiquer, de collaborer et d'accéder aux informations dans les meilleures conditions. Un scénario potentiel de l'utilisation de cette plateforme a été proposée dans le cadre de l'université. Une application *CAMCAP* assure les interactions sociales entre les étudiants (communication, évènement, échange de document multimédia) en bénéficiant de la sensibilité au contexte assurée par la plateforme. Un autre *middleware* a été proposé appelé *MobiSoC* (Gupta et al. 2009) pour le déploiement des applications sociales sur des dispositifs mobiles. Il capture le contexte géo-social d'une communauté pour inférer des modèles sociaux (i.e. *Social Patterns*) pour révéler les tendances et les pratiques sociales.

Dans les travaux présentés dans la suite, l'adaptation du contenu intègre en même temps des aspects de l'adaptation structurelle ou comportementale du système :

Park et al. (Park et al. 2005) proposent un modèle de résolution de conflit de contexte pour un groupe ou une communauté dans un environnement ubiquitaire. Autrement dit, dans un environnement multi-utilisateur, on se retrouve dans une situation de conflit entre les préférences ou les intentions des utilisateurs (e.g. problème de lumière éteinte ou allumée

⁵⁴ http://en.wikipedia.org/wiki/Opportunistic_Mobile_Social_Networks

dans une chambre pour 2 personnes). Le modèle proposé adapte les applications impliquées dans la situation de conflit afin de la résoudre.

Moss (Zhdanova et al. 2008) est une infrastructure qui vise, à la fois, à proposer du contenu et à adapter les fonctionnalités de l'application (*i.e.* services) pour améliorer l'expérience sociale de l'utilisateur, tout en prenant en compte ses préférences, ses communautés en cours, ses règles de respect de la vie privée et son contexte physique. Par exemple, une mère partage son agenda et sa liste des tâches avec son mari. Moss gère ce service de telle sorte que si c'est l'heure de récupérer les enfants de l'école et que le mari ne se trouve pas loin ou si la mère est occupée, une notification est envoyée au mari pour réaliser la tâche et une autre est envoyée à la mère pour l'informer de la prise en charge ou non de la tâche.

SocialAware (Gartrell 2008) propose un *Framework* pour applications sociales adaptatives. Pour valider ce *Framework*, deux prototypes ont été implémentés. Le premier permet de créer des *playlists* de musique selon les préférences des utilisateurs dans le même espace géographique. Le second, affiche des bandes annonces de films selon les préférences des utilisateurs dans un espace de projection commun.

Le tableau suivant offre une comparaison entre les projets représentatifs que nous avons présentés. La comparaison porte sur les points suivants qui nous semblent essentiels :

- Les éléments du contexte sur lesquels porte l'adaptation (temporalité, localité, système, social).
- Le type d'adaptation (du contenu, des fonctionnalités, structurelle, comportementale).

	Eléments du contexte				Type d'adaptation			
	Temporalité	Localité	système	social	De Contenu	de fonctionnalités	structurelle	comportementale
CALBA (Xu et al. 2013)	✗	✗			✗	✗		
Mobilife (Coutand et al. 2005)				✗		✗		✗
Bfriend (Smailovic et	✗	✗		✗	✗			

al. 2012)								
GeoLife (Zheng et al. 2010)		✗		✗	✗			
(Quercia et al. 2010)		✗		✗	✗			
Whozthat (Beach et al. 2008)		✗		✗	✗			
MobiSoC(Gupta et al. 2009)		✗		✗	✗			✗
(Park et al. 2005)		✗	✗	✗				✗
(Menkens 2009)	✗	✗		✗		✗		
Moss (Zhdanova et al. 2008)	✗	✗	✗	✗	✗	✗		
MobiClique (Pietiläinen et al. 2009)			✗	✗	✗			
SocialAware (Gartrell 2008)		✗		✗	✗	✗	✗	✗

Table 6: Comparaison des projets de social Computing en fonction des éléments du contexte et du type de l'adaptation.

Les efforts d'adaptation se sont intensifiés au niveau des applications sociales mais très peu d'applications considère les quatre dimensions du contexte social pervasif: temporalité, localité, entité sociale, entité système. Surtout, on s'aperçoit des insuffisances au niveau d'utilisation des évolutions en termes d'adaptation en fonction des contraintes système par les applications sociales qualifiées de 'pervasives'.

4.5 Conclusion

Cette conclusion présente une synthèse des grandes lignes de l'état de l'art sur lesquelles nous nous basons dans cette thèse.

Dans la partie état de l'art, nous avons commencé par une étude de la notion de communauté, ses définitions, les caractéristiques communes et les différences, la typologie. Cela nous a permis d'obtenir une vue d'ensemble afin de définir, concevoir et mettre en œuvre un modèle de communauté pour nos applications communautaires. Ensuite, nous avons

étudié la corrélation entre communauté et applications informatiques en présentant les travaux qui ont utilisé la notion de communauté comme un facteur de progrès pour les applications informatiques et les travaux où les applications informatiques représentent un support à l'expansion des communautés. Cette étude nous a permis aussi d'identifier le gain associé à l'informatisation de la communauté et les limites des applications communautaires existantes. La combinaison des objectifs généraux des applications communautaires et les difficultés à les surmonter nous ont donné une vision sur les défis et les objectifs de nos applications communautaires.

Les applications centrées communautés se situent dans le cadre d'un domaine en vogue : le *social computing*. Il recouvre plusieurs disciplines comme les sciences sociales, les sciences informatiques, bénéficie de plusieurs technologies et concerne différents domaines d'applications (social media, divertissement éducatif, *etc.*). Dans ce manuscrit, nous nous sommes présentés les fondements théoriques, ainsi que les technologies qui se rapportent à nos applications communautaires. L'exemple emblématique de ce domaine est celui des réseaux sociaux. Nous avons étudié et réalisé une classification selon leurs usages et leurs architectures, ce qui nous a permis de dresser un panorama représentant leurs valeurs montantes et leurs limites.

Suite à l'avènement du paradigme ubiquitaire, un bouleversement en profondeur dans le domaine du *social computing* a eu lieu. Cette rencontre fructueuse entre le *social computing* et l'informatique ubiquitaire a permis d'augmenter les possibilités et de concrétiser des usages des applications sociales d'une part et d'agréger, d'enrichir, de recréer, de contextualiser les informations et les services échangées d'autre part. Nous avons étudié les différents aspects de l'informatique ubiquitaire (comme la notion de contexte et les plateformes de sensibilité au contexte), et les principaux travaux dans lesquels convergent le monde ubiquitaire et le monde social. Ensuite, nous avons établi une comparaison selon les éléments du contexte sur lesquels porte l'adaptation des applications sociales (temporalité, localité, système, social) et le type d'adaptation considérée (de contenu, de fonctionnalités, structurelle, comportementale). L'examen de ses différentes applications sociales pervasives, nous a servi de point de départ pour concevoir notre architecture dédiée aux applications communautaires pervasives.

Partie 2 : Contribution

Chapitre 5 : Modèle de représentation de la notion de communauté à base d'ontologies

Le *pervasive social computing* prend de plus en plus d'ampleur grâce notamment à l'essor des dispositifs informatiques mobiles et au succès des applications sociales. Ces dernières intègrent de plus en plus la notion de communauté. Pour autant, les représentations de communauté ne sont pas toutes conçues de la même façon, ni dans les mêmes circonstances spatio-temporelles, ni par les mêmes organisations. Néanmoins, les membres des communautés ont besoin de collaborer et d'échanger des informations. Chaque application communautaire possède sa propre conception de la notion de communauté et utilise son propre modèle. Cependant, dans un environnement caractérisé par l'ubiquité, les communautés sont appelées à apparaître, évoluer, interopérer puis disparaître. Ces tâches sont rendues difficiles voire impossibles du fait de l'hétérogénéité de conception de ces communautés.

A l'heure actuelle, les ontologies constituent une solution pour la représentation et le partage des connaissances. Les applications communautaires que nous proposons reposent sur l'utilisation d'un modèle de communauté à base d'ontologies. Le but principal de ce modèle est de servir de point de référence pour décrire une communauté. Nous utilisons plusieurs ontologies existantes pour représenter une ou plusieurs ressources de la communauté. Par ailleurs, un ensemble d'extensions est proposé pour donner une description aussi complète et fidèle que possible de tous les aspects relatifs à la communauté. Ce modèle est indépendant du domaine considéré mais il peut être couplé à des ontologies du domaine. C'est dans ce contexte que se situe la première partie de notre contribution concernant les applications communautaires.

Dans notre travail, nous nous intéressons aux communautés potentiellement présentes dans un environnement ubiquitaire. Étant donné que les communautés existantes sont thématiques et plus au moins statiques face à un environnement fortement dynamique et

changeant (*i.e.* changement du contexte), nous proposons un nouveau type de communauté adapté aux exigences des environnements pervasifs : les communautés spontanées.

Ce chapitre présente notre modèle de communauté spontanée. Tout d'abord, nous introduisons la notion de communauté spontanée et ses usages potentiels. Par la suite, nous proposons une modélisation de la notion de communauté à base d'ontologies. Cette représentation permet la compréhension et l'interprétation des informations associées aux communautés par les utilisateurs et les logiciels. Finalement, nous exposons un processus d'inférence de nouvelles informations pour enrichir la base de connaissances communautaires.

5.1 Communauté spontanée

5.1.1 Définition

Bien que les applications communautaires existantes constituent une avancée importante vers l'expansion des communautés, elles offrent des informations et des services liés à des thèmes (*i.e.* le sujet autour duquel s'organise la communauté) définis à l'avance par les développeurs et mis à la disposition des utilisateurs. La plupart de ces applications se limite à un seul thème comme la visualisation 3D des cartes (e.g. *YesCitez*), le trafic routier (e.g. *Eklairer*), le partage des *tags* sur une carte (e.g. *HappyStep*), le monde de l'art (e.g. *Commic'Art*), etc. Rares sont les applications qui considèrent plus d'un thème. Mais que se passe-t-il si l'utilisateur ne trouve pas de réponse à ses questions dans les applications qui lui sont proposées ? Que se passerait-il si on lui donnait la possibilité de créer une communauté sur le thème qui l'intéresse ?

Si l'utilisateur change de centre d'intérêt, ces applications ne s'adaptent plus à ses nouveaux besoins. Il doit abandonner l'application et en chercher une autre. De plus, la création des communautés ne peut pas être spontanée puisque le thème est figé dès la conception de l'application et la communauté est créée en fonction de ce thème. À l'opposé, nous constatons un besoin croissant des utilisateurs de créer des communautés selon une thématique qu'ils définissent eux-mêmes.

Dans notre travail, nous proposons un nouveau type de communautés : les communautés spontanées que nous définissons comme :

Un regroupement spontané d'individus ayant un intérêt commun lié à une situation circonstancielle relative à un territoire géographique et limité dans le temps.

Une communauté spontanée désigne un ensemble d'utilisateurs qui partagent des intérêts ou des pratiques communes, se retrouvent dans une situation inopinée caractérisée par une durée de vie éphémère et délimitée géographiquement et interagissent via une application sociale.

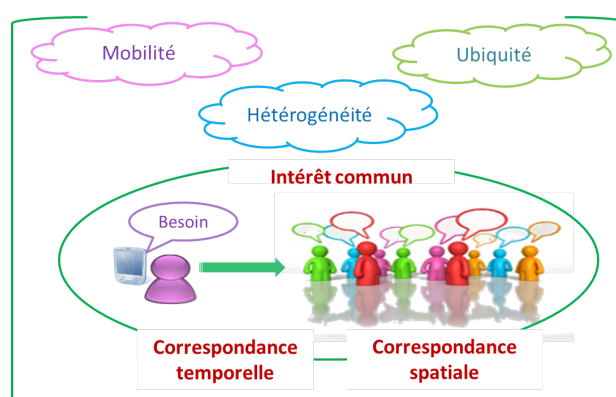


Figure 18: Illustration de la communauté spontanée.

Ce type de communauté représente la croisée de plusieurs types de communautés décrits précédemment (cf. section 2.1.3) : communauté virtuelle, communauté d'intérêt, communauté de pratique, communauté géographique. Il est caractérisé par :

- La communauté est *spontanée* c'est-à-dire détachée de toutes contraintes thématiques imposées par l'application. Elle est caractérisée par un ou plusieurs centres d'intérêts définis par les membres en fonction de leurs intérêts ou de leur situation présente (besoins, objectifs, nouvel emplacement géographique, *etc.*) et non par le domaine de l'application.
- La communauté est *éphémère* car sa durée de vie est délimitée dans le temps. Cette éphémérité s'accorde avec l'idée des communautés spontanées dans un environnement

ubiquitaire et suit la tendance actuelle des réseaux sociaux éphémères (*cf.* section 3.3.4). Une communauté peut disparaître après une durée définie par ses membres, ou suite à un abandon (i.e. arrêt des échanges sociaux) ou encore suite à la satisfaction ou l'obsolescence du besoin pour lequel elles ont été créées.

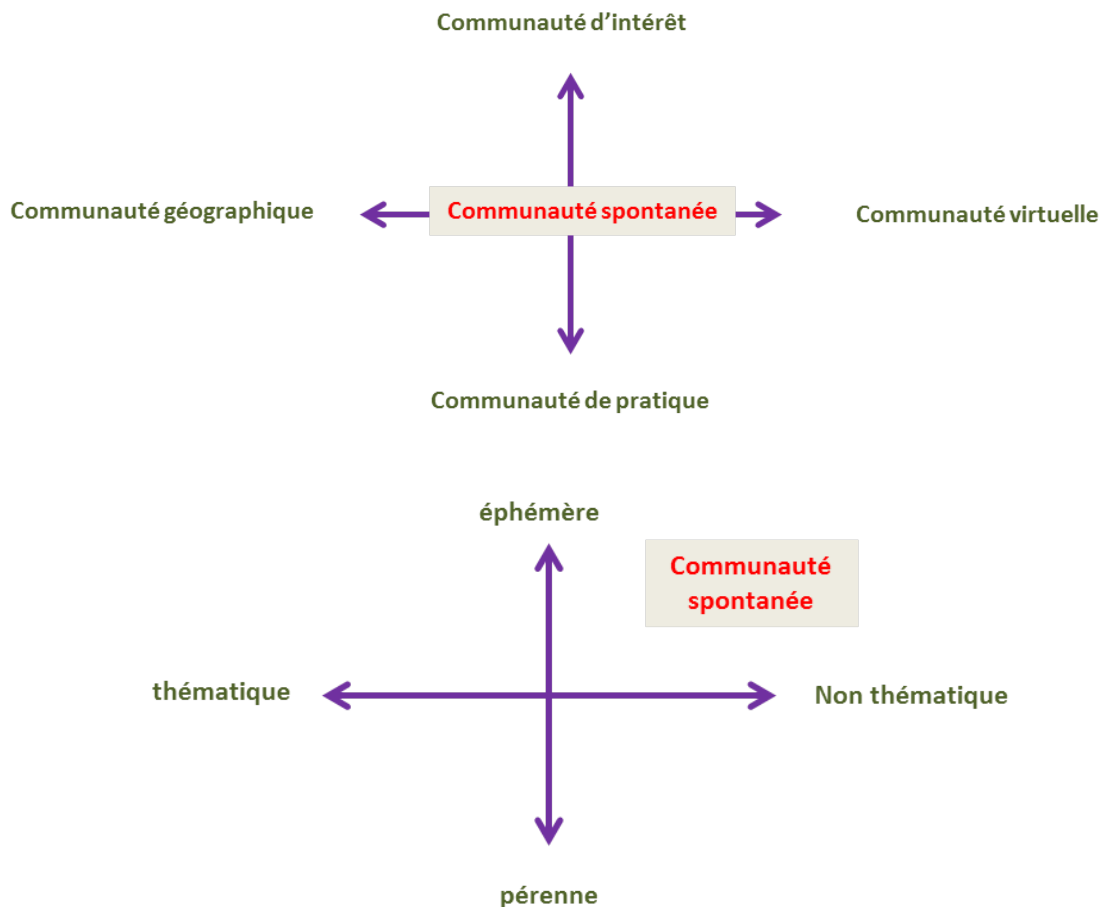


Figure 19: Positionnement de la communauté spontanée.

Nos applications communautaires s'intéressent plus particulièrement à ce type de communauté, c'est-à-dire à ces communautés temporaires, éphémères, géolocalisées et spontanées. En effet, c'est, selon nous, le type de communauté qui s'accorde le mieux avec des situations d'ubiquité et/ou de mobilité.

5.1.2 Usages potentiels

La communauté spontanée permet de répondre aux besoins des utilisateurs nomades, dans des situations précises temporaires ou des situations circonstancielles imprévues qui n'ont pas pu être satisfaits par des communautés établies.

En plus des raisons habituelles d'adhésion à une communauté établie identifiées dans la section 2.1.1, d'autres motifs s'ajoutent incitant l'utilisateur à rejoindre des communautés spontanées tels que :

- *Le repérage et la localisation des objets ou des personnes ;*
- *Le développement des relations sociales avec les personnes sur un même territoire géographique ;*
- *L'échange et le partage de documents multimédias avec les personnes présentes aux alentours ;*
- *L'organisation d'événements ou de festivités ;*
- *Une situation de crise,*
- *etc.*

Plusieurs scénarios d'usage permettent d'illustrer l'utilité de la notion de communauté spontanée et de démontrer la diversité de son utilisation. Nous distinguons deux grandes catégories de scénarios où le type de communauté proposé est approprié aux besoins des utilisateurs mobiles. La première catégorie correspond aux manifestations de type festival, événement sportif, conférence, exposition, *etc.* L'objectif est de réunir des personnes autour des activités de la manifestation et de les assister dans l'échange de contenus multimédias afin de promouvoir et de mieux organiser l'événement. La deuxième catégorie correspond aux situations inopinées comme les catastrophes naturelles, les accidents, les incidents, les incendies, *etc.* Utiliser des communautés éphémères spontanées pour mettre en relation des personnes permet de faciliter la diffusion d'informations en cas d'urgence. Il est quasi-impossible d'assurer des préventions a priori de tous les événements imprévus. En revanche, se servir de communautés spontanées a posteriori, permet de collecter, gérer et transmettre les informations et les services aux personnes dans des délais réduits.

Nous adoptons un scénario dans le cadre du projet *MOANO* (cf. section 1.1) dans un parc botanique où les communautés spontanées permettent de mettre en relation tous les intervenants (visiteurs, jardiniers, éco-gardes, intervenants externes, *etc.*) afin d'animer les activités du parc, d'intervenir en cas de crise (*e.g.* accident, incendie, *etc.*), d'inciter les échanges sociaux, d'enrichir la base de connaissances du parc. Ce scénario sera détaillé dans la section 6.1.

Dans la section suivante, nous présentons une modélisation de la communauté et de ses caractéristiques à base d'ontologies. Nous définissons un vocabulaire commun regroupant l'ensemble des concepts décrivant ces communautés et fournissant une vision unique et homogène de la communauté permettant aux applications communautaire de collaborer et d'interopérer.

5.2 Ontologie de communauté

En raison de la masse actuelle d'informations, il est nécessaire de faire en sorte que les données échangées dans un domaine soient structurées et représentées par un vocabulaire unifié et contrôlé pour faciliter leur organisation, leur interprétation par des composants logiciels, et leur exploitation ultérieure. Les technologies de représentation sémantique des connaissances sont devenues incontournables ce qui explique la forte propension des concepteurs et managers de systèmes d'information à les utiliser.

Se baser sur des ontologies pour modéliser un système d'information procure plusieurs avantages que (Brisson 2006) a résumé en trois points principaux :

- *La collaboration* : différentes personnes peuvent avoir une vue différente du même domaine, les ontologies sont un moyen de fournir une représentation unifiée, consensuelle et partageable des connaissances.
- *L'interopérabilité* : les ontologies permettent l'intégration de l'information à partir de sources différentes pour constituer une seule source homogène.
- *La modélisation et la réutilisation* : les ontologies structurent les informations en ensembles facilement réutilisables, et leur caractère formel permet d'appliquer des

opérations de raisonnement pour vérifier la cohérence des informations ou pour déduire de nouvelles informations.

En outre, il n'y a pas de collaboration et d'interopérabilité entre les différentes applications communautaires existantes, car elles ne se basent pas sur un modèle consensuel ou un vocabulaire commun. Si les membres d'une communauté souhaitent recourir, dans certains cas, à l'utilisation d'un service fourni par une autre application communautaire, ils sont confrontés à l'absence d'interopérabilité entre les applications communautaires. En revanche, ces applications gagneraient à travailler ensemble, partager des ressources, échanger des données, pour mieux satisfaire l'utilisateur. Chaque application traite la communauté selon sa propre conception en fonction de son usage et du service offert. Cependant, nous constatons qu'il existe des caractéristiques communes aux communautés de ces différentes applications, d'où l'intérêt de proposer un modèle de communauté à base d'ontologies.

Une ontologie constitue à la fois une représentation structurée de l'information et un ensemble de règles d'inférences. D'une part, cette représentation permet d'aboutir à des raisonnements automatisés, d'autre part elle permet d'assister l'utilisateur dans ses actions et de favoriser le partage et la réutilisation des informations par l'homme et la machine. L'intégration des ontologies dans les applications communautaires permet de profiter des avantages suivants :

- L'utilisation d'un vocabulaire commun par les différents acteurs.
- La recherche sémantique d'informations.
- L'inférence de nouvelles connaissances.
- La formalisation des échanges entre composants logiciels et humains.
- La collaboration et l'interopérabilité entre les applications.
- Le traitement automatique des données.
- La réutilisation du savoir et la capitalisation des connaissances extraites à partir de sources hétérogènes.

Compte tenu de la diversité des définitions et des typologies des communautés, nous proposons une conceptualisation de la communauté qui est l'objet central de nos applications mobiles. Celle-ci fournit un support formel décrivant explicitement une communauté, et peut servir de modèle pour d'autres applications communautaires afin de minimiser l'hétérogénéité et de faciliter l'interopérabilité et la collaboration entre elles.

5.2.1 Utilisation des vocabulaires existants

L'évolution de l'utilisation des ontologies est soutenue par l'existence de plusieurs vocabulaires et technologies confirmés par la communauté des concepteurs (*e.g.* W3C). A ce titre, nous nous appuyons sur les nombreux efforts de recherche menés dans le domaine des ontologies pour concevoir notre ontologie de définition et de caractérisation des communautés. L'objectif principal dans la construction de cette ontologie est de représenter sémantiquement des descriptions relatives aux communautés interprétables par les membres et par les machines en utilisant des standards comme *OWL*⁵⁵, afin d'effectuer des recherches sémantiques en utilisant des langages d'interrogation comme *Sparql*⁵⁶ et d'inférer des informations complémentaires relatives aux communautés créées.

En nous basant sur une étude des diverses communautés (*cf.* section 2.1), nous avons extrait huit concepts de base permettant de définir le squelette de l'ontologie : l'intérêt commun, le membre, la durée de vie, la localisation, le type de communauté, le contenu, le service et le contexte.

- **L'intérêt commun** : c'est un sentiment d'attachement envers un objet, un sujet ou une activité. Il représente le lien et le point commun fondamental entre les membres de la communauté.
- **Les membres** : c'est l'ensemble des personnes participant au moins une fois à la constitution ou à la persistance d'une communauté. Nous avons distingué trois types de membres: les créateurs de la communauté, les membres actifs et les membres inactifs.

⁵⁵ <http://www.w3.org/2004/OWL/>

⁵⁶ http://www.w3.org/2009/sparql/wiki/Main_Page

- **La durée de vie** : c'est la période durant laquelle la communauté est active (*i.e.* persistance des échanges sociaux).
- **La localisation** : c'est le territoire géographique de la communauté. Ce dernier constitue le point commun qu'on peut percevoir entre les membres. Ce territoire peut être figé (dès la création de la communauté, une zone géographique fixe lui est attachée) ou dynamique (une zone géographique délimitée et calculée en fonction des positions des membres). La dimension spatiale est aussi utilisée pour l'annotation de l'information échangée au sein de la communauté.
- **La typologie de la communauté** : c'est le type de la communauté : communauté d'intérêt, communauté de pratique, *etc.*
- **Le contenu** : c'est l'information échangée entre les membres de la communauté, elle peut être sous différentes formes (post, article, commentaire, *etc.*) et différents formats (son, image, vidéos, *etc.*). Cette information peut être annotée par des méta-informations à caractère spatio-temporel.
- **Les services** : c'est l'ensemble de services permettant de réaliser les objectifs et répondre aux besoins des membres d'une communauté. Ces services facilitent l'interaction entre les membres de communauté et répondant aux besoins en relation avec un centre d'intérêt commun. Chaque service est annoté par l'ensemble des communautés qui l'ont utilisé.
- **Le contexte** : la notion de contexte est liée à la communauté. C'est une interprétation des événements contextuels et de leurs relations (*e.g.* incendie). L'entité contexte est un élément clé dans l'identification des situations nécessitant éventuellement une adaptation de l'application. Un modèle de contexte proposé par Keling Da (Da 2014) a été adopté. Ce modèle, associé à des raisonnements, permet de transformer les informations contextuelles de bas niveau en informations de plus haut niveau plus facilement interprétables. La conséquence étant d'assurer la continuité de service et ainsi d'adapter en permanence au contexte du moment.

Nous présentons ci-dessous l'ontologie de communauté que nous avons définie. Cette ontologie contient des concepts et des propriétés permettant de décrire formellement les communautés, importe des vocabulaires existants et y ajoute des extensions.

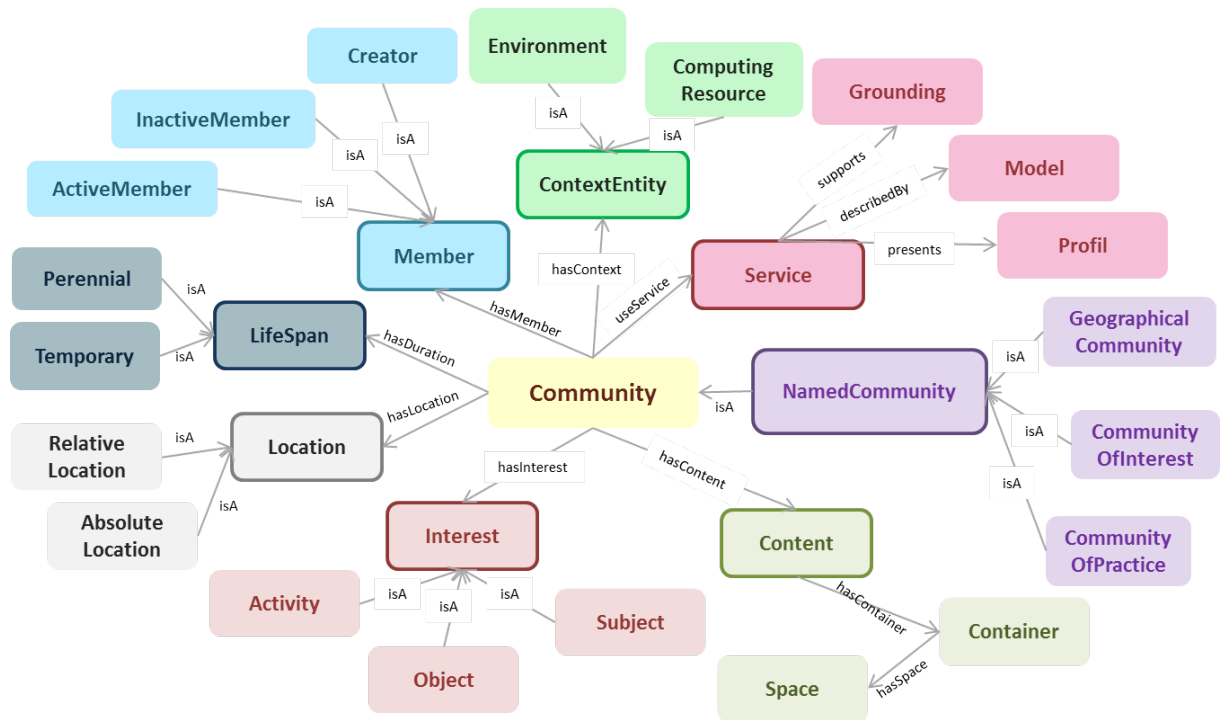


Figure 20: Extrait de l'ontologie de communauté.

La Figure 20 montre un extrait simplifié de notre ontologie. Cette dernière reprend les huit entités indispensables, selon nous, à la description d'une communauté dont l'objectif principal est la gestion des communautés et ses ressources.

En partant de l'idée que l'utilisation des standards et des vocabulaires existants est préconisée, nous avons importé les vocabulaires suivants *Foaf*, *Sioc*, *Owl-S*, *GeoRSS*, *Owl-Time* pour représenter respectivement les membres (*member*), le contenu (*content*), les services (*service*), la localisation (*location*) et la durée de vie (*lifeSpan*).

Nous avons choisi d'utiliser *Foaf* (*the friend of a friend*) pour décrire le profil des membres au sein la communauté. C'est un standard du web sémantique qui vise à décrire des personnes, leurs intérêts, leurs liens et leurs activités. *Sioc* (*Semantically-Interlinked Online Communities*) est un vocabulaire qui a pour but de décrire les interactions et les informations échangées entre les communautés en ligne. Il est couramment associé à *Foaf* pour décrire les personnes impliquées. Dans le contexte de nos travaux, il est utilisé pour décrire le contenu et les échanges entre les membres d'une communauté. *Owl-Time* est proposé par le *W3C* comme

référence pour la description du contenu temporel des pages Web ou pour la description des propriétés temporelles des ressources en général, nous l'avons utilisé pour décrire la durée de vie de la communauté. Pour le territoire de la communauté, nous avons utilisé *GeoRSS-simple*, devenu une recommandation du *W3C* comme vocabulaire de référence pour la description des propriétés géospatiales des ressources Web. *GeoRSS-simple* permet d'associer une entité (*community*) à un type de géométrie (*_geometry* : *point*, *line*, *box*, et *polygone*). *Owl-S* est une ontologie permettant de décrire les propriétés et les capacités des services utilisés par une communauté. Les détails de ces vocabulaires ont été examinés à la section 3.2.2.

Nous présentons dans la Table 7 un exemple d'instanciation de notre ontologie de communauté. À cet effet, nous avons choisi un scénario dans lequel une personne appelée *Alice*, qui possède un compte d'utilisateur dont l'identifiant est *Alice_Compte*, se connecte à la communauté *Fan_des_Orchidées*, et publie un post *Post_Orchidée* dans l'espace social de cette communauté. Pour des raisons de simplification, nous supposons que *Fan_des_Orchidées* est une communauté qui ne compte que deux membres: *Alice* et *Fabrice*. À l'heure actuelle, sa durée de vie est de 1h32. La communauté se situe sur un territoire géographique dont l'identifiant est *Parc_Botanique_territoire_1*. Le polygone *Polygon_1* définit les coordonnées géographiques de ce territoire. Le service *Prendre_Photo* est attaché à cette communauté, car il est utilisé par au moins un de ses membres pour assurer le partage et l'échange de contenu.

```

<owl:NamedIndividual rdf:about="http://www.communties.org/ontology#Alice">
  <rdf:type rdf:resource="&contact:Person"/>
  <foaf:familyName>Fiori</foaf:familyName>
  <foaf:firstName>Alice</foaf:firstName>
  <foaf:holdsAccount rdf:resource="http://www.communties.org/ontology#Alice_Compte"/>
  <foaf:knows rdf:resource="http://www.communties.org/ontology#Albert"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://www.communties.org/ontology#Albert">
  <rdf:type rdf:resource="&contact:Person"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://www.communties.org/ontology#Alice_Compte">
  <rdf:type rdf:resource="&Sioc:UserAccount"/>
  <Sioc:member_of rdf:resource="http://www.communties.org/ontology#Fan_des_Orchidées"/>
  <Sioc:creator_of rdf:resource="http://www.communties.org/ontology#Post_Orchidée"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://www.communties.org/ontology#Fun_des_Orchidées">
  <rdf:type rdf:resource="&Sioc:Usergroup"/>
  <has_duration rdf:resource="http://www.communties.org/ontology#Duration1"/>

```

```

<Sioc:has_member rdf:resource="http://www.communties.org/ontology#Alice_Compte"/>
  <Sioc:has_member rdf:resource="http://www.communties.org/ontology#Fabrice_Compte"/>
<has_location rdf:resource="http://www.communties.org/ontology#Parc_Botanique_territoire_1"/>
  <use_service rdf:resource="http://www.communties.org/ontology#Prendre_Photo"/>
  <has_poste rdf:resource="http://www.communties.org/ontology#Post_Orchidée"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://www.communties.org/ontology#Post_Orchidée">
  <rdf:type rdf:resource="&Sioc:Post"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://www.communties.org/ontology#Prendre_Photo">
  <rdf:type rdf:resource="&owls:Service"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://www.communties.org/ontology#Duration1">
  <rdf:type rdf:resource="&time:DurationDescription"/>
<time:hours>1</time:hours>
<time:minutes>32</time:minutes>
<time:seconds>15</time:seconds>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://www.communties.org/ontology#Parc_Botanique_territoire_1">
  <rdf:type rdf:resource="&gml:_Feature"/>
  <georss:where rdf:resource="http://www.communties.org/ontology#Polygon_1"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://www.communties.org/ontology#Polygon_1">
  <rdf:type rdf:resource="&gml:Polygon"/>
  <georss:posList> 50.578668 2.980399
50.578690 2.980320
50.578700 2.980500
50.578935 2.986775
50.578668 2.980399
</georss:posList>
</owl:NamedIndividual>

```

	Concept
	Relation
	Individu
	Extension

Table 7: Extrait du fichier OWL de l'ontologie de communauté.

Nous présentons en vert les concepts, en rose les relations, en jaune les instances, en bleu les relations ajoutées pour relier les ontologies importées. D'autres concepts et propriétés permettant de mieux décrire une communauté ont été ajoutés à la combinaison des vocabulaires utilisés. Dans ce qui suit, nous donnons de plus amples détails sur les extensions utilisées.

5.2.2 Les extensions définies

Les vocabulaires que nous avons utilisés pour représenter notre modèle de communauté ont su s'imposer dans le domaine du web sémantique et du *social computing*

grâce à leur simplicité et à leur potentiel descriptif. Malgré les avantages de l'utilisation de ces vocabulaires, cela n'empêche pas de constater certaines lacunes comme :

- a) La représentation de l'intérêt d'un mono-utilisateur et pas de représentation d'intérêt commun pour un groupe d'utilisateur ou une communauté par l'ontologie *Foaf*.
- b) L'absence de représentation de la typologie des communautés.
- c) La non prise en compte du contexte d'usage et du contexte d'exécution (Petit 2010) pour la communauté.
- d) Bien que des efforts aient été déployés pour aligner les ontologies (comparaison et mise en correspondance des ontologies) *Foaf* et *Sioc* (e.g. *foaf:group* et *Sioc:userGroup*, *Sioc:userAccount* et *foaf:onlineaccount*), il existe encore des correspondances (équivalence de concepts ou de relations, etc.) (cf. Figure 24).

Pour répondre point par point à ce que nous considérons comme des insuffisances au niveau de la combinaison des vocabulaires utilisés pour décrire une communauté, nous avons ajouté des extensions aux niveaux des concepts et des relations du modèle.

Pour le premier point, la différence entre une communauté et un groupe de personnes est parfois difficile à faire et beaucoup de travaux les confondent. Une communauté est un groupe de personnes unies par des liens d'intérêts communs (Larousse 2014). Les deux projets *Foaf* et *Sioc*, décrivent la notion de groupe d'utilisateurs comme étant un ensemble de *UserAccount*. L'intérêt commun qui réunit ces utilisateurs n'est pas explicitement décrit. En effet, l'intérêt d'un utilisateur est représenté à travers la relation *foaf:interest* qui relie la personne et un document. Ceci est insuffisant car une personne peut avoir des intérêts variés autres que les documents. C'est pourquoi nous avons ajouté le concept *Interest* (cf. Figure 21) pour représenter le sujet commun qui unit les membres de la communauté. Cet intérêt peut être un objet, un sujet ou une activité.

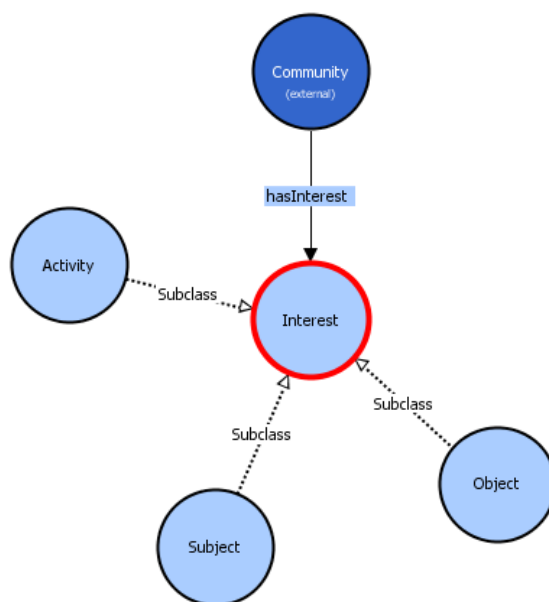


Figure 21: Représentation de l'intérêt de la communauté.

Pour le deuxième point, le concept de communauté reste un peu général et abstrait. En dépit des efforts des sociologues pour classifier les communautés et leur impact sur le social, les typologies des communautés (cf. section 2.1.3) n'ont pas été modélisés au niveau des vocabulaires existants. En revanche, l'information du type de la communauté peut apporter des informations concernant les services et les outils nécessaires pour réaliser les objectifs de mise en place de la communauté (permettant de résoudre le problème de démarrage à froid).

Type de communauté	Outils nécessaires
Communauté d'intérêt	Outils nécessaires pour échanger des informations et des documents multimédia autour de l'intérêt commun (e.g. <i>Apple Iphoto, Picasa, Adobe Photoshop Album, ACDSee, etc.</i>)
Communauté de pratique	Technologies et outils de communication tels que la messagerie et les applications de VoIP (e.g. <i>Gmail, Yahoo messenger, Kphone, Asterisk, etc.</i>)
Communauté géographique	Outils de repérage géographique comme le GPS, capteurs de mouvement (e.g. accéléromètre) ou indiquant la direction (e.g. gyroscope, boussole).

Table 8: Les outils nécessaires pour chaque type de communauté.

Ceci explique l'intérêt que nous avons à représenter cette typologie de communautés dans notre ontologie. La Figure 22 présente les principaux types : communauté d'intérêt

(CoI), Communauté de pratique (CoP), Communauté géographique (geographic community), Communauté d'apprentissage (learning community).

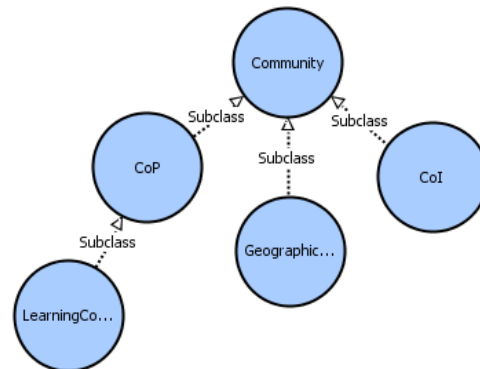


Figure 22: Représentation de la typologie de la communauté.

Pour le troisième point, l'ignorance de l'environnement extérieur de la communauté représente une contrainte pour son adaptation et son évolution. Les systèmes sensibles au contexte (*cf.* section 4.1) visent essentiellement un utilisateur, et non pas une communauté, comme étant une entité sociale distinguée par un contexte commun qui peut être pertinent pour l'interaction entre communauté et application. Du fait que la prise en considération du contexte de la communauté nous est indispensable, nous reprenons une conceptualisation du contexte (*cf.* Figure 23) proposée dans le cadre de la plateforme de sensibilité de contexte *Kalimucho* développée au sein de notre équipe (Da 2014). L'ontologie de contexte adoptée permet de considérer un état du contexte actuel de la communauté et de supporter les fonctionnalités d'adaptations de la plateforme. Elle contient les concepts d'Entité de Contexte ("*ContextEntity*"), de communauté ("*Community*"), d'Environnement ("*PhysicalEnvironment*", et "*ComputingEnvironment*"), et de Ressource de calcul ("*ComputingResource*"). Tous les éléments de ce modèle sont des sous-catégories du concept "*ContextEntity*". Le centre de ce modèle est la communauté puisque son but est de décrire le contexte de la communauté, l'environnement physique où il se situe, et les ressources auxquelles il a accès.

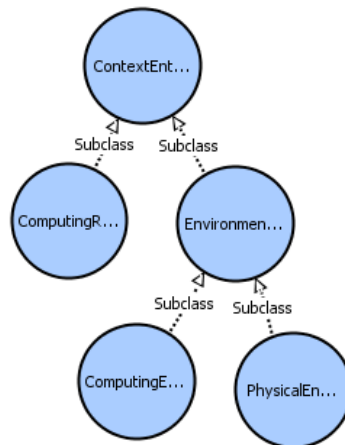


Figure 23: Représentation du contexte de la communauté.

Pour le dernier point, bien que des efforts aient été déployés pour aligner les vocabulaires *Foaf* et *Sioc* (e.g. *foaf:group* et *Sioc:userGroup*, *Sioc:userAccount* et *foaf:onlineaccount*), il reste toutefois encore quelques lacunes à combler (cf. exemple dans la Figure 24). Il en est de même des autres vocabulaires, surtout de ceux qui possèdent des correspondances (i.e. équivalence de concepts ou de relations, etc.). Par exemple, pour les ontologies *Foaf* et *Sioc*, il existe des équivalences au niveau des 'data properties'. Dans notre cas, nous avons fait un choix arbitraire entre les correspondances (équivalence de concepts ou de relations, etc.).

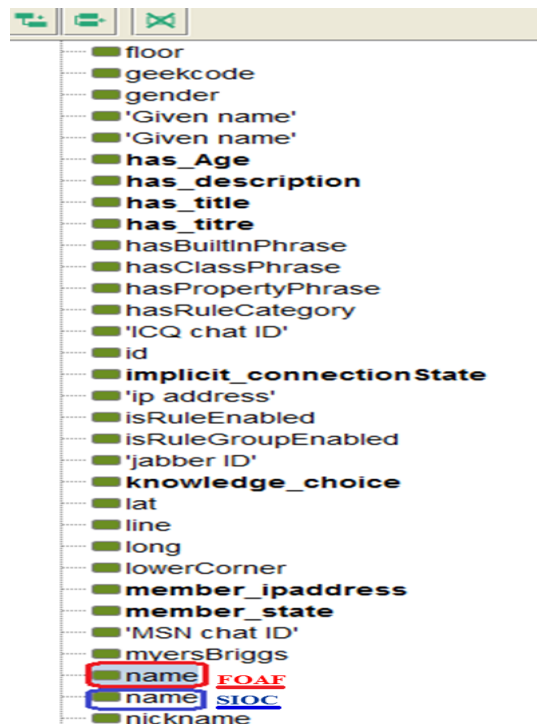


Figure 24: Exemple de correspondance au niveau des vocabulaires *Foaf* et *Sioc*.

Nous avons présenté une modélisation sémantique en concevant une ontologie de communautés. Ce modèle est indépendant du domaine (*e.g.* médecine, botanique, droits, *etc.*) mais il peut être couplé à des ontologies du domaine. Dans notre cas, une ontologie qui contient des connaissances propres au domaine de la botanique est utilisée pour le scénario présenté dans la section 6.1.

Nous avons opté pour une démarche graduelle en commençant par importer les vocabulaires confirmés existants, puis en ajoutant des extensions pour donner une description complète et aussi fidèle que possible de tous les aspects relatifs à la communauté. Bien entendu, il existe d'autres ontologies qui nous semblent intéressantes comme *Weighted Interests*⁵⁷, *BondGrid* (Bai et al. 2006), *SWRC (Semantic Web for Research Communities)* (Sure et al. 2005), *Skos (Simple Knowledge Organization System)*⁵⁸, *BIBO (Bibliographic Ontology)*⁵⁹, *Delivery Context Ontology*⁶⁰, *etc.* Toutefois, en raison de notre volonté de

⁵⁷ <http://smiy.sourceforge.net/wi/spec/weightedinterests.html>

⁵⁸ <http://www.w3.org/2004/02/skos/>

⁵⁹ <http://bibliontology.com/>

simplification du modèle, nous nous sommes volontairement limitées aux vocabulaires cités précédemment. Néanmoins, le modèle de communauté proposé reste extensible, un avantage accordé grâce à l'utilisation des ontologies. Nous présentons dans la Table 9, une comparaison entre les vocabulaires existants et les extensions élaborées au niveau de la représentation des concepts de la communauté.

Principaux Concepts	Vocabulaire existant	Extension
<i>Communauté</i>	X	
<i>Typologie de communauté</i>		X
<i>Intérêt</i>		X
<i>Profil du membre</i>	X	
<i>Contexte</i>		X
<i>Durée de vie</i>	X	
<i>Localité</i>	X	
<i>Service</i>	X	
<i>Contenu échangé</i>	X	

Table 9: Tableau récapitulatif des principaux concepts du modèle de la communauté.

5.3 Les inférences

Un autre avantage des ontologies est l'inférence. C'est le mécanisme de raisonnement qui s'appuie sur les connaissances acquises et un ensemble de règles qui formalisent la partie logique du raisonnement pour déduire de nouveaux faits. L'inférence est définie par Kayser (Kayser 1997) comme « *une façon générique de désigner l'ensemble des mécanismes par lesquels des entrées (perceptives ou non) sont combinées à des connaissances préalables afin d'obtenir des comportements élaborés* ». L'un de nos objectifs est d'enrichir la base de connaissances de la communauté qui contient les informations introduites par l'utilisateur (*i.e.* instances de l'ontologie de la communauté) par des informations inférées par le raisonneur.

Les langages de représentation de connaissances (*e.g.* *RDF Schema*, *OWL*) sont basés sur les logiques de descriptions *LD* (Napoli 1997), une famille de formalismes pour la représentation et l'inférence des connaissances. Les logiques de descriptions reposent sur trois notions de base : les concepts, les relations et les individus (*i.e.* instantiation des concepts). Les *LD* permettent de décrire ces éléments suivant deux niveaux: le niveau terminologique *T-*

⁶⁰ <http://www.w3.org/TR/dcontology/>

BOX et le niveau assertional *A-BOX*. Le niveau terminologique (*T-BOX*) comprend la description des concepts et des relations. Le niveau assertional (*A-BOX*) est constitué des individus, de leur description et des règles qui leur sont attachées. L'inférence s'effectue au niveau terminologique ou assertional. Les règles d'inférence pouvant être appliquées par des logiciels appelés *raisonneurs* ou *moteurs d'inférence* capables de traiter une ontologie OWL pour déduire des faits qui n'y sont pas explicitement déclarés. Parmi les raisonneurs existants, citons Fact⁶¹, Pellet⁶², Racer⁶³. Les principaux services d'inférence proposés par les raisonneurs se présentent aux niveaux de la *T-BOX* et de la *A-BOX*.

Inférence au niveau de la *T-BOX*

- *Satisfiabilité* : il s'agit de vérifier qu'une classe admet des instances. Une classe insatisfaisable (*i.e.* incohérente) entraîne une inconsistance.
- *Subsorption* : permet de savoir si une classe est une sous-classe d'une autre ou non. Il s'agit de construire une hiérarchie de classes inférées.

Inférence au niveau de la *A-BOX*

- *Cohérence*: vérifier si les assertions définies restent cohérentes avec la *T-BOX*.
- *Réalisation* : trouver le concept le plus spécifique décrivant un individu. Ce concept doit être un concept de la *T-BOX*.

Une hypothèse importante du raisonnement avec les *LD*, connu sous le nom du *monde ouvert*, est que la sémantique n'est pas définie *a priori* mais constatée *a posteriori*. Autrement dit, ce principe stipule qu'une information manquante peut être vraie. Par exemple, si on déclare trois concepts *B*, *C* et *D* comme sous concepts d'un concept *A*, on ne peut pas affirmer que les trois concepts sont les seuls enfants du concept *A*; il faudra le déclarer explicitement afin que le raisonnement s'appuie sur le fait que *A* ne possède que ces trois sous-concepts. Parfois, il est souhaitable de fermer la description des entités de telle sorte que l'on puisse

⁶¹ <http://owl.cs.manchester.ac.uk/fact++/>

⁶² <http://clarkparsia.com/pellet/>

⁶³ <http://www.sts.tu-harburg.de/~r.f.moeller/racer/>

affirmer qu'une déclaration est vraie si sa négation ne peut pas être prouvée. Pour notre exemple, on peut exprimer qu'il y a seulement trois sous-concepts du concept A et pas d'autres.

La Table 10 décrit un exemple d'inférences de nouvelles connaissances pour enrichir la description d'une communauté. Le type d'une communauté est une connaissance que peut générer automatiquement un raisonneur. À partir d'un ensemble de règles, le moteur d'inférence déduit qu'une instance du concept '*Community*' qui possède une relation avec une instance du concept '*Interest*', est aussi une instance d'un concept plus spécifique qui correspond au type de la communauté. Dans l'exemple ci-dessus, une communauté qui a pour intérêt un objet naturel (*i.e.* fleurs) est une communauté d'intérêt. Quant à la communauté de pratique, son intérêt est une activité qui peut être une coopération, une pratique, *etc.*

Extrait de la TBox de la communauté
<i>Community</i> : { <i>CommunityOfInterest</i> , <i>CommunityOfPractice</i> , <i>GeographicalCommunity</i> , <i>LearningCommunity</i> }
<i>Interest</i> { <i>Activity</i> , <i>Object</i> , <i>Subject</i> }
<i>Activity</i> { <i>Help</i> , <i>Cooperation</i> , <i>Festivity</i> , <i>Manifestation</i> , <i>Practice</i> }
<i>Object</i> { <i>Natural</i> , <i>Artifact</i> }
<i>Subject</i> { <i>Law</i> , <i>Actuality</i> , <i>theme</i> , <i>Isdea</i> }
<i>CommunityOfInterest</i> \subseteq <i>Communities</i> $\cap \exists$ <i>hasInterest</i> (<i>Object</i> , <i>Subject</i>)
Extrait de la ABox de la communauté
<i>Flowers Lovers</i> : <i>Community</i>
<i>hasInterest</i> (<i>Flowers</i> , <i>Natural</i>)
<i>Natural</i> : <i>Object</i>
Inférence
<i>Flowers Lovers</i> \rightarrow <i>CommunityOfInterest</i>

Table 10: Exemple d'inférence du type de la communauté.

Dans la Figure 25, se trouvent les descriptions des types de communautés suivants : Communauté d'intérêt, communauté géographique et communauté de pratique. La typologie est représentée par un ensemble d'axiomes qui sert de point de départ au raisonnement.



Figure 25: Règles d'inférence en DL.

Le langage *SWRL* (*Semantic Web Rule Language*) a été proposé pour pallier certaines insuffisances d'inférence au niveau des logiques de description. *SWRL* est un langage de règles combinant le langage *OWL-DL* et le langage *RuleML*. Il étend l'ensemble des axiomes de *OWL* en y intégrant des clauses de Horn. En général, une règle *SWRL* possède la forme suivante :

Règle SWRL: affirmation sous la forme d'un antécédent \rightarrow conséquent décrivant des inférences possibles.

Les règles définies par *SWRL* sont composées d'un antécédent (corps) et d'une conséquence (en-tête), et si les conditions définies dans l'antécédent sont vérifiées alors les conditions définies dans la conséquence sont exécutées. Les deux parties constituant une règle (antécédent et conséquence) sont des expressions de la forme:

$$a(x, y) \vee b(y, z) \vee c(x) \vee \dots \rightarrow n(x, z)$$

a, b, n : prédicats binaires (rôles)

c : prédicat unaire (concept atomique)

x, y : variables, instances ou littéraux.

Table 11: Règles d'inférence en SWRL.

OWL-DL est un langage simple dont l'expressivité reste limitée. Par exemple, dans notre ontologie, on a les faits suivants :

Un utilisateur U est créateur d'une ressource R .
L'utilisateur U est membre d'une communauté C .
→ Communauté C est concédante de la ressource R .

Avec *OWL-DL*, on ne peut pas exprimer une relation composée comme celle décrite ci-dessus. Donc, on ne peut pas en déduire que la communauté C qui concède les droits d'utilisation de la ressource R car le créateur U de la ressource est un de ces membres. L'utilité des règles *SWRL* est d'exprimer des relations difficiles voire impossibles à exprimer avec *OWL-DL* et relier les instances concernées.

5.4 Conclusion

Afin de concevoir des applications communautaires pervasives, sensibles au contexte, et exploitant des informations à caractère spatio-temporel, nous avons présenté, dans ce chapitre, la notion de communauté spontanée qui est le type de communauté qui s'accorde le mieux à des situations d'ubiquité et/ou de mobilité. Ensuite, nous avons présenté une ontologie qui modélise les communautés, cerne leurs caractéristiques, définit un vocabulaire commun regroupant l'ensemble des concepts décrivant ces dernières, fournit une vision unique et homogène de la communauté permettant d'ajouter une couche sémantique à la recherche et la consultation de l'information, et permet d'enrichir la base de connaissances communautaire en inférant automatiquement de nouvelles informations. L'utilisation des standards du web sémantique permet d'assurer une homogénéité et une interopérabilité des données au sein des communautés.

Dans le chapitre suivant, nous présentons l'architecture et le fonctionnement de nos applications communautaires à base d'ontologies de communautés qui facilitent les échanges sociaux et la découverte des services au sein des communautés.

Chapitre 6 : Taldea : une architecture dédiée aux applications communautaires dynamiquement reconfigurables

Dans ce chapitre, nous proposons l'architecture *Taldea* qui entend dépasser les limitations des applications communautaires précédemment décrites (*cf.* section 2.3), en intégrant l'ensemble des propositions pour la gestion des communautés, ainsi que pour la découverte et le déploiement dynamique des services des applications communautaires. Notre système a pour objectif de produire des applications qui répondent aux conditions requises et aux objectifs de la plupart des applications communautaires (*cf.* section 2.2.1), tout en intégrant la notion de *spontanéité* et de sensibilité au contexte. *Taldea* utilise plusieurs ontologies, notamment l'ontologie de communauté proposée dans le chapitre précédent, ainsi qu'une voire plusieurs ontologies de domaines.

Bien que nous présentions *Taldea* comme une architecture dédiée aux applications communautaires, il s'agit plus précisément d'un environnement qui produit des applications supportant les communautés tout au long de leur cycle de vie en fournissant les services adaptés. L'architecture *Taldea* est composée essentiellement de deux modules comme le montre la Figure 26. Le premier module permet de faciliter l'accès et l'interaction d'un utilisateur avec des communautés. Le deuxième présente une approche permettant d'assister les communautés dans la découverte de services adaptés à leurs intérêts et à leur contexte.

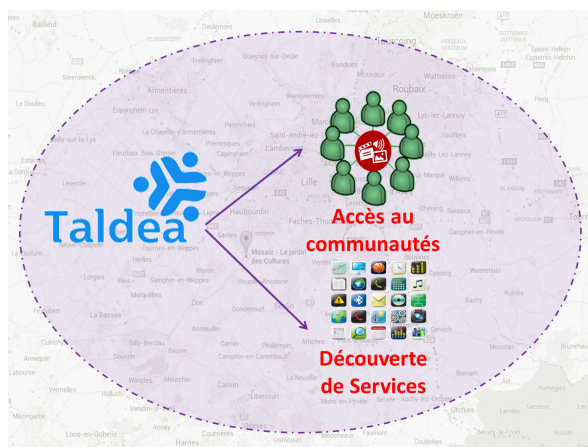


Figure 26: Fonctions de base de l'architecture *Taldea*.

6.1 Scénario

L'objectif principal de *Taldea* consiste à favoriser les interactions des utilisateurs nomades avec les communautés environnantes, ainsi qu'à faciliter la découverte des services adaptés à leurs besoins.

Plusieurs scénarios possibles pourraient montrer l'utilité de la communauté spontanée (*cf.* section 5.1.2) dans des situations temporaires ou critiques comme les conférences, les expositions, les festivals, les événements sportifs, les catastrophes naturelles, *etc.*, ainsi le fonctionnement des applications *Taldea* supportant ce type de communautés dans des environnements contraignants.

Parmi les scénarios envisagés, supposons qu'un botaniste équipé de son *Smartphone* visite un parc botanique. Dès son entrée dans le parc, il se connecte à l'application *Taldea* au travers des technologies de géorepérage⁶⁴ (permettant de définir une zone géographique et de lancer des alertes via l'application lorsque l'appareil quitte ou entre dans cette zone). Pour sa première connexion à l'application, l'utilisateur indique ses centres d'intérêt et son profil sera ensuite dynamiquement enrichi à partir des différentes recherches qu'il réalisera. Une fois

⁶⁴ <http://www.transport-intelligent.net/produits-services/article/le-geofencing>

connecté à l'application, l'utilisateur peut accéder aux communautés et utiliser les services associés de trois façons : la recommandation, la recherche et la création.

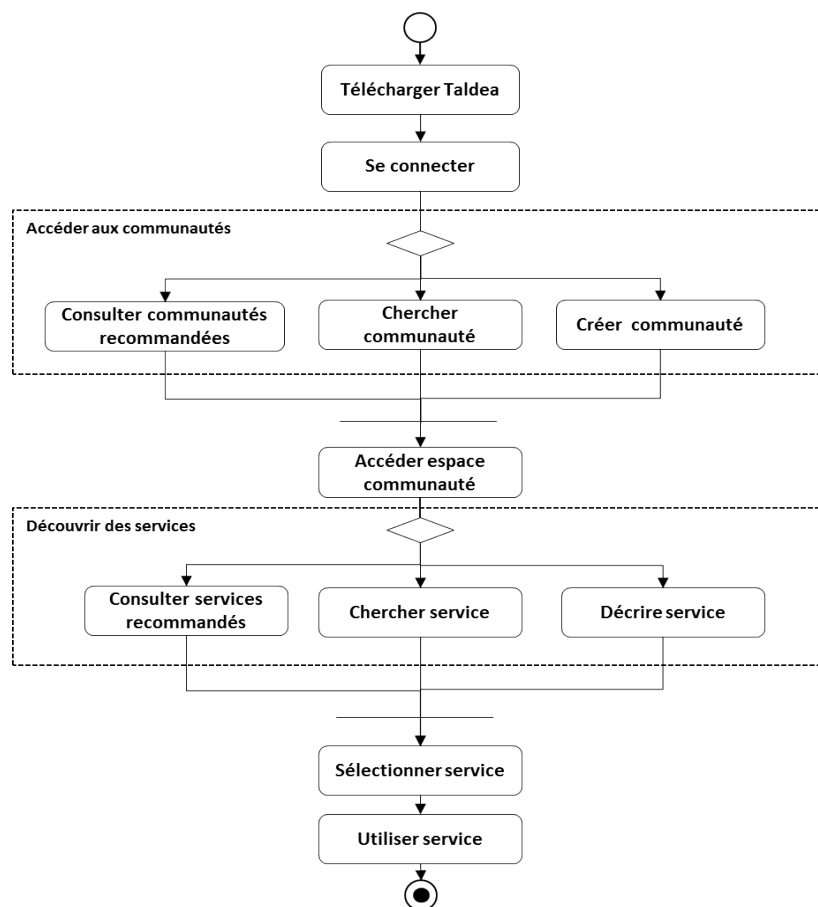


Figure 27: Illustration de l'utilisation de *Taldea*.

L'application lui recommande, parmi les communautés existantes, celles qui possèdent des liens de proximité sémantique avec ses intérêts (proximité obtenue à partir de l'ontologie du domaine à laquelle se réfère *Taldea*) et/ou une proximité géographique entre sa position géographique et le territoire de la communauté. L'utilisateur peut également réaliser une recherche parmi les communautés selon le thème qu'il désire (par exemple, communauté des arts plastique). Les propositions qui lui sont faites par l'application peuvent être ordonnées selon plusieurs critères comme le nombre des membres ou la popularité (un système de notation de communautés à travers lequel chaque utilisateur peut attribuer une note, *etc.*). Supposons que l'utilisateur se connecte à l'une des communautés recommandées (*e.g.*

amateurs d'oiseaux), il se dirige vers l'espace virtuel partagé de la communauté. Dans la communauté, les échanges entre les membres s'effectuent via des publications ou des messages et chaque publication est automatiquement annotée par des données à caractère spatio-temporel (*e.g.* coordonnées *GPS*). En outre, les utilisateurs de l'application peuvent communiquer entre eux en temps différé à travers des messages, ou en temps réel sur l'espace de discussion de la communauté.

En termes de fonctionnalités, l'application *Taldea* a recommandé à l'utilisateur des communautés en fonction de son profil et de sa localisation, mais elle lui offre également la possibilité d'en chercher une par lui-même. En revanche, s'il ne trouve pas de réponse correspondant à son besoin dans les communautés existantes, ou si cette réponse est tardive et met en cause son efficacité, il peut créer une communauté spontanée.

La nouvelle communauté est recommandée à tous les utilisateurs de *Taldea* (qui peuvent être membres d'autres communautés) ayant indiqué qu'ils acceptaient de participer à des communautés spontanées et ayant un lien sémantique avec leurs centres d'intérêts. Cette acceptation peut, éventuellement, utiliser un filtrage spatio-temporel (*i.e.* proche du territoire de la communauté). Lors de la création de la communauté, le créateur peut définir une durée de vie limitée au terme de laquelle la communauté sera clôturée. Dans le cas contraire, dès que le besoin pour lequel la communauté a été créée sera satisfait ou deviendra obsolète, le créateur de la communauté pourra, à tout moment, prendre la décision de la clôturer.

Pour la nouvelle communauté, *Taldea* recommande à ses membres les services utilisés par les communautés sémantiquement proches sur la base de calculs de similarité sémantique (*e.g.* *visite virtuelle*). Si ces propositions ne sont pas satisfaisantes, ou si de nouveaux besoins apparaissent, les membres de la communauté ont la possibilité de rechercher de nouveaux services en formulant une requête en langage naturel (*e.g.* *catalogue du parc*).

Par ailleurs, l'utilisateur peut attacher à un service ou un ensemble de services une description afin de faciliter des recherches futures et des déploiements ultérieurs. Comme, par exemple, le cas où l'utilisateur associe les annotations « *accident, catastrophe naturelle, crise, etc.* » à la description du service « *Alarme* ».

Une fois qu'un service est sélectionné par l'utilisateur parmi les résultats de la recherche ou de la recommandation, le déploiement, ainsi que les éventuelles reconfigurations du service, sont assurés, par la plateforme logicielle *Kalimucho* (développée au sein de notre équipe de recherche). Il s'agit d'une plate-forme de déploiement d'applications réparties reconfigurables sur tout type de machine. L'application *Taldea* elle-même est réalisée à partir de composants interconnectés que cette plate-forme supervise. Les reconfigurations de services permettent leur adaptation automatique au contexte. Par exemple, le parc propose un service de description des plantes sous différents médias que les utilisateurs peuvent visualiser sur leur *Smartphone*. Un service de diffusion de vidéo sera automatiquement déployé sur le périphérique de l'utilisateur. Mais, en raison d'une faible bande passante, le service de vidéo n'est pas utilisable, *Kalimucho* génèrera une reconfiguration de ce service consistant en l'affichage d'une séquence d'images sous-titrées pour assurer la continuité de service.

6.2 Exigences

Pour satisfaire aux enjeux du scénario présenté dans la section précédente, *Taldea* doit satisfaire aux exigences suivantes :

- *Gestion des communautés* : création de communautés, accès aux communautés existantes, recherche de communautés, etc.
- *Gestion des échanges des communautés* : gestion des informations éphémères et pérennes échangées au sein des communautés, utilisation des bases de connaissances pour capitaliser les connaissances produites par les communautés.
- *Traitement du contenu multimédia* : gestion du contenu multimédia et fourniture de services adaptés à son exploitation dans un environnement mobile.
- *Déploiement dynamique de services* : acquisition et exploitation du contexte d'usage et du contexte d'exécution afin de déployer dynamiquement les services adaptés.
- *Gestion des profils des utilisateurs* : gestion des informations comme les centres d'intérêt, les préférences et les contraintes de sécurité et de confidentialité.
- Protection de la vie privée : sous sa forme la plus simple, raisonnement sur un ensemble de règles de contrôle d'accès permettant de protéger les données et le respect de la vie privée des utilisateurs.

Rien n'empêche de considérer d'autres exigences, mais au niveau de ce travail, nous nous limitons à la liste présentée ci-dessus.

6.3 Architecture et principe de fonctionnement

Nous présentons dans la Figure 28, le mécanisme d'accès aux communautés et le mécanisme de déploiement dynamique de services (introduit dans la Figure 26). Chacun de ces deux modules comprend trois services de base : la recommandation, la recherche et la création (R, S, C).

Dans ce système, (1) l'utilisateur se connecte. Pour sa première connexion, un profil contenant les centres d'intérêt qu'il a introduits est créé. Dans le cas contraire, l'application récupère son profil. (2) Ses centres d'intérêt sont projetés sur l'ontologie du domaine, ainsi que les centres d'intérêt des communautés existantes. (3) Ensuite un calcul de similarité est utilisé pour identifier les communautés sémantiquement et géographiquement proches de l'utilisateur. Si l'utilisateur n'est pas satisfait de ces propositions, (4) il peut envoyer une requête en langage naturel pour chercher des communautés. (5) Sa requête est envoyée à l'interface d'annotation (6) qui génère des annotations avec des concepts. (7) Une requête est formulé à partir de ces résultats pour exploiter l'ontologie en utilisant le raisonneur. (8) L'ensemble des communautés renvoyées représente celles dont les centres d'intérêt correspondent à une thématique recherchée et qui vérifient les règles de contrôle de confidentialité. (9) Si le résultat obtenu ne répond pas aux besoins de l'utilisateur, ce dernier peut créer de son propre chef une communauté spontanée en introduisant des informations (*e.g.* intérêts, description textuelle, *etc.*). (10) Ces informations en langage naturel sont annotées. (11) De nouvelles connaissances peuvent être inférées pour enrichir la description de la communauté. (12) Une instanciation de l'ontologie de communauté est effectuée à partir du résultat de l'annotation des informations introduites par l'utilisateur et des informations inférées. Une fois la communauté créée, elle est munie d'un ensemble de services pour assurer les échanges et les interactions entre les membres. Pour la découverte des services adaptés aux activités de la communauté, les 3 services de base (R, S, C) sont aussi utilisés. (13) Il est possible de recommander à la communauté des services correspondant à ses intérêts, (14) de même les membres peuvent mener une recherche pour des services

particuliers et, (15) finalement, il est possible d'associer une description à un service ou un ensemble de services pour favoriser les futurs déploiements.

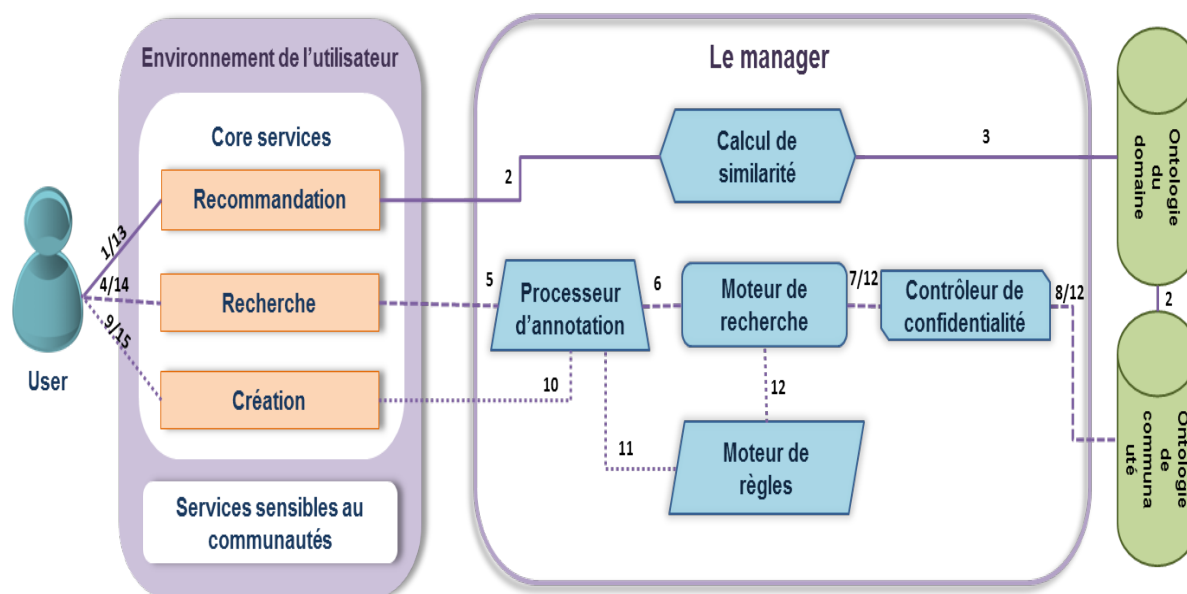


Figure 28: Architecture générale de Taldea.

6.4 Les composants de l'architecture

Taldea possède une architecture hybride qui combine les propriétés des architectures centralisées et distribuées et tire avantages de chacune. L'architecture contient deux types d'entités structurelles (*i.e.* qui constituent la structure) : Le **Manager de communautés** et l'**environnement de l'utilisateur**.

Le **Manager de communautés** est le coordinateur qui interagit avec les bases de connaissances. Un ensemble de fonctionnalités se trouve au cœur de ce *Manager* permettant d'exécuter des calculs, d'alimenter, d'extraire, de mettre à jour et d'inférer des connaissances depuis l'ontologie de communauté, l'ontologie de service et l'ontologie du domaine. Le *Manager* peut être centralisé ou distribué sur plusieurs dispositifs en fonction de la disponibilité des ressources matérielles. Nous présentons ci-dessous la liste des fonctionnalités qui sont détaillées dans les sections suivantes:

- Générer des requêtes formelles;

- Interroger les ontologies ;
- Annoter des ressources ;
- Peupler les bases de connaissances ;
- Calculer les distances sémantiques entre concepts d'une ontologie ;
- Gérer les règles d'inférence et les règles de confidentialité.

L'**environnement de l'utilisateur**, lui, permet de découvrir et d'interagir avec les communautés. Deux types de services peuvent être déployés sur le *Dispositif* : les *core services* et les services sensibles aux communautés (ou services ad hoc). Dans ce qui suit, nous présentons les différents composants de l'architecture proposée.

6.4.1 Les core services et les services sensibles aux communautés

Les core services : ce sont les services de base, indispensables pour atteindre l'objectif principal de l'utilisation de l'application communautaire. Ils sont les mêmes pour tous les utilisateurs indépendamment de leur profil et de leur contexte. Parmi ces services, on trouve le service d'authentification, d'accès aux communautés, de gestion de profil, de gestion de l'espace social (*i.e.* l'espace de partage entre les membres de communautés), *etc.* Ces services interagissent suite à des actions explicites de l'utilisateur (*e.g.* modifier le profil) ou des événements du contexte (*e.g.* connexion via géorepérage).

Les services ad hoc ou les services sensibles aux communautés : ils permettent la réalisation des objectifs de la communauté, supportent ses activités et facilitent les interactions entre les membres. Dans ce manuscrit, nous présentons comme exemple les services suivants : *TakePicture* (pour prendre des photos), *P2Pchat* (messagerie instantanée), *WayPoint* (localisation des points d'intérêt sur une carte). La Figure 29 illustre le fonctionnement du service *P2Pchat* qui suit le modèle *P2P* hybride. Ce dernier utilise le **Manager des communautés** comme un index central pour chercher ou connecter des pairs (*i.e.* utilisateurs). La communication et l'échange sont effectués à travers une architecture *P2P* décentralisée pure.

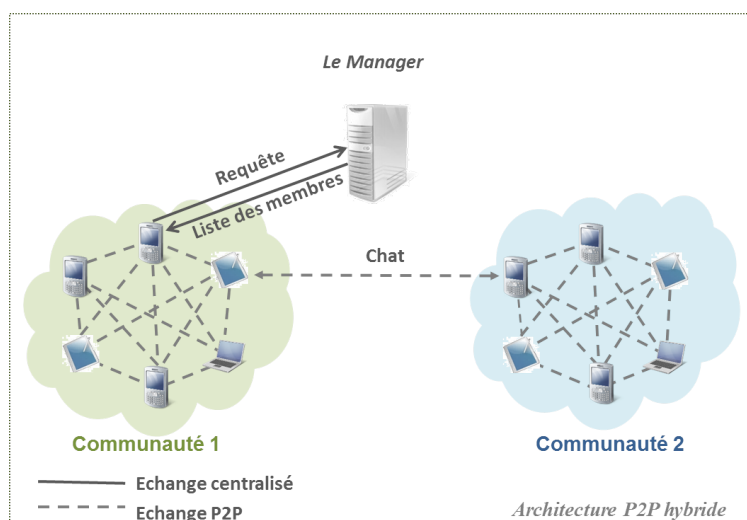


Figure 29: Exemple de service *ad hoc* : P2Pchat.

En raison des ressources limitées des dispositifs mobiles et du nombre croissant des services, nous proposons, dans ce travail, une approche de découverte et de déploiement contextuel des services *ad hoc*. La conscience du contexte est un atout permettant à l'application orientée services, dans un environnement caractérisé par l'ubiquité, de s'adapter aux contraintes du contexte d'usage et du contexte d'exécution (cf. section 4.1). L'ère où l'on déploie les services de l'application sans appliquer un raisonnement en fonction du contexte, est bien terminée.

Les services présentés (*core service* ou service *adhoc*) sont déployés sur le dispositif de l'utilisateur. Les composants du **Manager** des communautés, à savoir le processeur d'annotation, le moteur de requête, le calculateur de proximité, le moteur des règles et le contrôleur de confidentialité, sont décrits en détail dans les sections suivantes.

6.4.2 Le processeur d'annotation

Cette section présente l'un des composants du *Manager des communautés* : le processeur d'annotation qui permet d'annoter sémantiquement un texte entré par l'utilisateur au regard d'une ontologie de domaine.

Les systèmes d'annotation sémantique se situent généralement dans le cadre du Web sémantique. Ils recourent à un modèle formel de connaissances, communément celui des ontologies. L'annotation sémantique permet de donner une sémantique formelle et explicite aux requêtes de l'utilisateur en langage naturel, en les mettant en correspondance avec des éléments de l'ontologie. Le résultat de l'annotation est exprimé en termes de concepts de l'ontologie, de rôles conceptuels, de relations entre instances, d'instances de concepts, d'axiomes ontologiques. La Figure 30 illustre un exemple d'annotation du fragment de texte suivant :

Marie lit une pièce de théâtre de Molière.

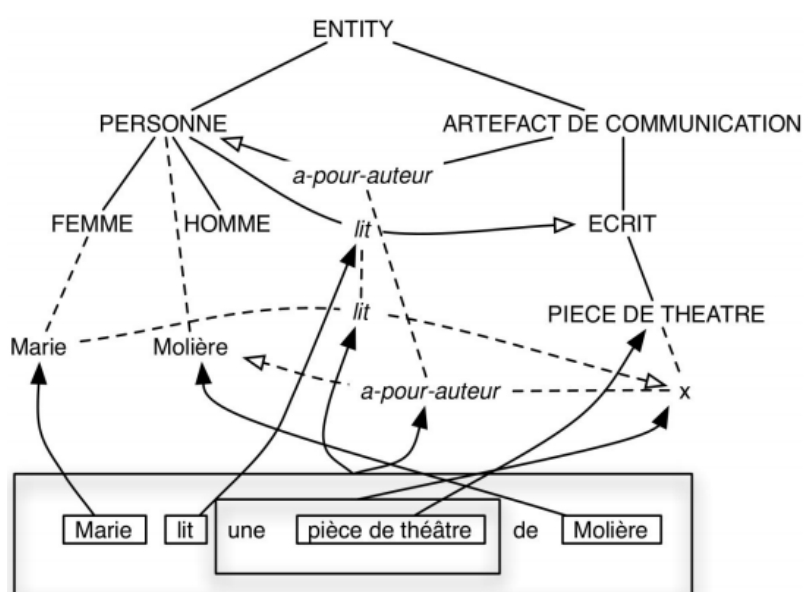


Figure 30: Exemple des différents types d'annotation (Ma et al. 2009).

L'exemple de (Ma et al. 2009) permet d'illustrer les différents types de d'annotation, à savoir :

- Des instances de concepts ou ce que l'on appelle aussi des entités nommées telles que les noms de personnes, les noms d'organisations ou d'entreprises, les noms de lieux, les quantités, les distances, les valeurs, les dates, *etc.* (e.g. Marie).
- Des concepts (e.g. pièce de théâtre).

- Des rôles conceptuels c'est-à-dire des termes qui peuvent être rattachés à des concepts dans le processus d'annotation (*e.g.* lit).
- Des relations entre instances c'est-à-dire un fragment qui est annoté comme une relation entre instances de concepts (*e.g.* une pièce de théâtre de Molière).
- Des axiomes ontologiques. Il n'y pas d'exemple dans le fragment ci-dessus, mais si on prend la phrase suivante : *les pièces de théâtre sont toujours écrites par quelqu'un*, en l'annotant, on peut dénoter l'axiome ontologique exprimant une restriction de cardinalité du rôle *a-pour-auteur*.

Ces différents résultats d'annotation ne sont pas souvent considérés simultanément. Selon les cas et les outils, on met en évidence un ou plusieurs types d'annotation parmi les cinq précités. De nombreux outils d'annotation sémantique sont disponibles en *Standalone* comme SemEx (Guissé et al. 2011), Excom (Alrahabi et al. 2009), *etc.* ou via des services Web tels que DBpedia Spotlight⁶⁵, Wikimeta⁶⁶, OpenCalais⁶⁷, *etc.* Ils se différencient par la nature, l'encodage et l'utilisation des ressources ontologiques. Notre objectif à travers l'utilisation de l'annotation sémantique est d'effectuer des recherches sémantiques, plutôt que de simples recherches de chaînes de caractères, afin d'améliorer la précision de recherche par la compréhension de la signification des requêtes. Pour notre travail, nous recourons au service web *TextAnnot*⁶⁸. C'est un service web qui a été développé au sein de notre équipe dans le cadre du projet *ANR MOANO*.



Figure 31: *TextAnnot* service web d'annotation.

⁶⁵ <http://dbpedia-spotlight.github.io/demo/>

⁶⁶ www.wikimeta.com/fr

⁶⁷ <http://new.opencalais.com/>

⁶⁸ <http://themat2i.univ-pau.fr:8080/TextAnnot-WWW/annotation.jsp>

Ce service assure l'annotation sémantique et la recherche d'informations dans des documents textuels en se basant sur une ontologie de domaine. Une ontologie du domaine botanique a été proposée pour faciliter la compréhension de ce domaine (objectif du projet *MOANO*) (Aussenac-Gilles et al. 2013). La Figure 32 présente le noyau d'ontologie construit à partir d'une analyse manuelle d'une collection de documents de jardinage.

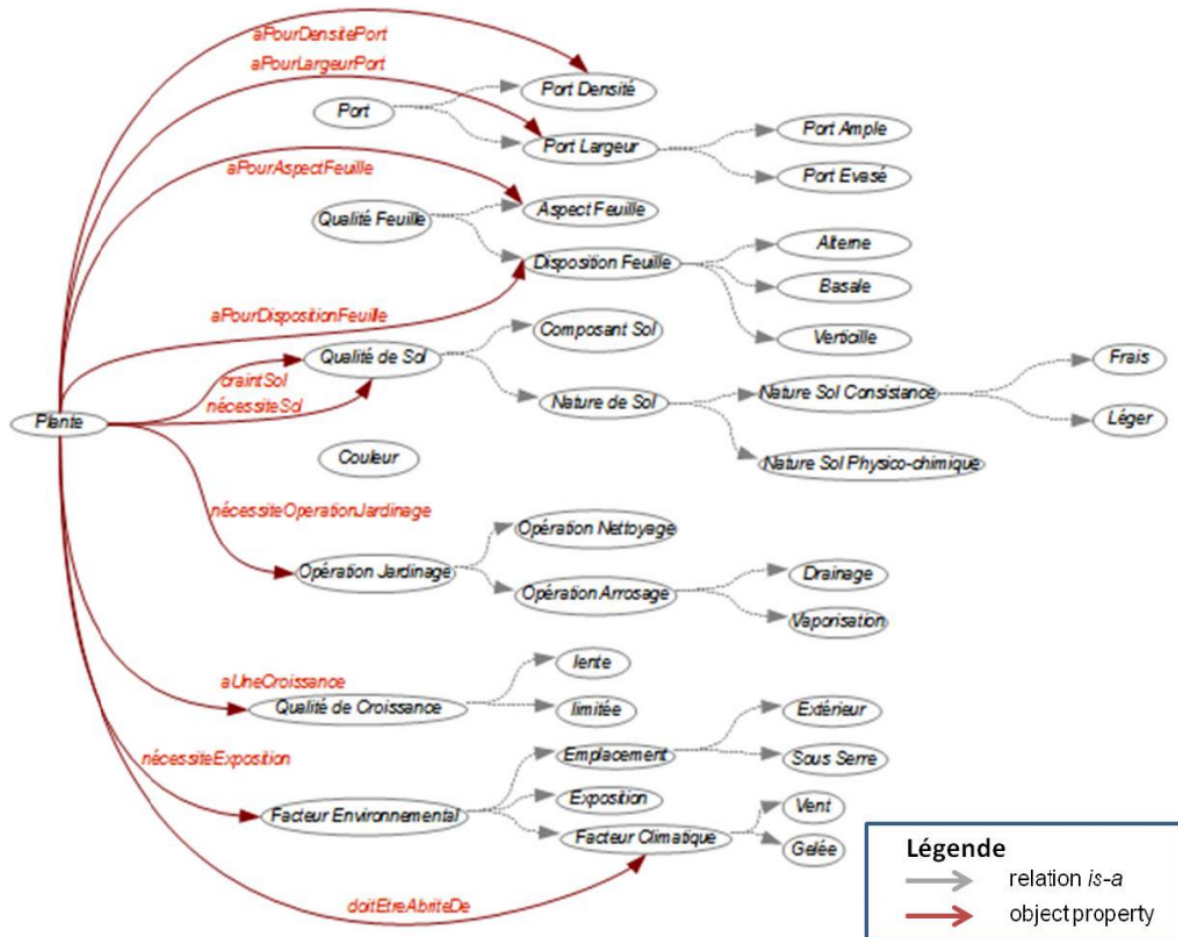


Figure 32: Extrait de l'ontologie botanique (Aussenac-Gilles et al. 2013).

Nous présentons dans la suite un scénario d'utilisation du service web articulé autour de la botanique. Nous considérons le fragment de texte suivant :

Quelle consistance de sol exigent les végétaux ?

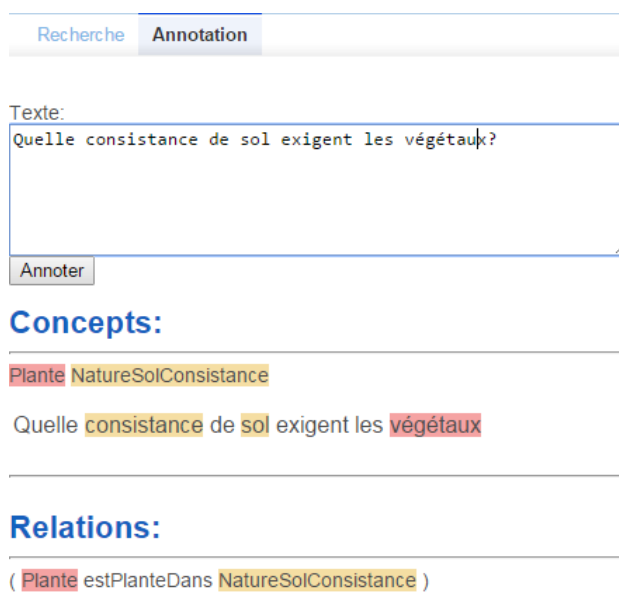


Figure 33: Exemple d'annotation sémantique avec *TextAnnot*.

L'annotation de cet extrait de texte met en valeur les concepts *Plante* et *NatureSolConsistance* et la relation *EstPlanteDans* qui relie *Plante* et *NatureSolConsistance*.

Les résultats obtenus suite à des évaluations (Aussenac-Gilles et al. 2013) sont prometteurs même si cela n'empêche pas quelques problèmes liés à la langue comme la négation, le non traitement de texte dépassant 1024 caractères, *etc.* Ces désavantages ne sont pas problématiques dans notre cas d'utilisation.

Le deuxième composant du *Manager* de communautés, qui utilise les résultats du processeur d'annotation, est le moteur de requête qui génère des requêtes formelles pour interroger la base de connaissances communautaires.

6.4.3 Le moteur de requête

Taldea permet d'interroger la base de connaissances communautaires formée de l'ontologie du domaine (Aussenac-Gilles et al. 2013) et de l'ontologie communautaire (*cf.* section 5.2) en utilisant des requêtes en langage naturel. Notre objectif est d'épargner à l'utilisateur la complexité des langages de requête formelles comme *Sparql* (exprimer des requêtes formelles implique de connaître et de respecter la syntaxe du langage utilisé, de

comprendre un modèle de graphes et de connaître le schéma de la base interrogée). Le moteur de requête utilise les résultats de l'annotation de la requête en langue naturelle pour générer des requêtes plus formelles. Il joue ainsi le rôle de générateur de requêtes *Sparql*.

Sparql (*Protocol And Rdf Query Language*) est un langage de requête recommandé par le W3C depuis 2008. *Sparql* permet de rechercher, d'ajouter, de modifier ou de supprimer des données *RDF*⁶⁹ disponibles à travers diverses sources de données. Il est adapté à la structure spécifique des graphes *RDF*, et s'appuie sur les triplets qui les constituent. Les requêtes *Sparql* permettent non seulement de requêter *RDF* (à travers la clause *Select*), mais aussi de construire de nouveaux graphes ou de compléter des graphes existants (à travers la clause *Construct*). Une requête *Sparql* possède la forme suivante :

<p>PREFIX indique l'espace de noms</p> <p>SELECT identifie les variables qui doivent apparaître dans les résultats d'interrogation</p> <p>FROM indique l'ensemble de données (le graphe) à utiliser</p> <p>WHERE fournit le motif de graphe élémentaire auquel comparer</p> <p>CONSTRUCT construire un nouveau graphe ou complète le graphe interrogé</p> <p>UNION union de deux motifs de graphe élémentaires</p> <p>FILTER rajoute des conditions devant être satisfaites</p> <p>OPTIONAL des motifs supplémentaires peuvent étendre la solution</p> <p>DESCRIBE Retourne un graphe RDF décrivant les ressources trouvées</p> <p>ASK Retourne un booléen indiquant si un motif d'interrogation correspond ou non</p>
--

Table 12: Forme d'une requête *Sparql*.

⁶⁹ http://fr.wikipedia.org/wiki/Resource_Description_Framework

L'exemple ci-dessous montre une requête *Sparql* permettant de rechercher depuis *DBpedia*⁷⁰ (base de données en format *RDF* construite à partir des données dérivées de Wikipédia) les communes d'Île de France de plus de 100 000 habitants.

```
PREFIX db-owl: <http://dbpedia.org/ontology/>
SELECT * WHERE {
  ?ville db-owl:region <http://fr.dbpedia.org/resource/île-de-France>.
  ?ville rdf:type db-owl:Settlement .
  ?ville db-owl:populationTotal ?population.
  FILTER (?population > 100000)
}
```

Table 13: Exemple de requête *Sparql*.

Plusieurs systèmes existants proposent des solutions pour la génération de requêtes *Sparql* allant des interfaces offrant des fonctionnalités simples comme l'auto-complétion ou l'assistance graphique pour la construction de requêtes (*e.g. Flint*⁷¹), jusqu'aux travaux qui cherchent à générer semi automatiquement ou automatiquement des requêtes formelles à partir des requêtes en langage naturel (Pradel et al. 2013). Dans ces travaux de recherche, la génération de requêtes consiste essentiellement en les étapes suivantes (Pradel et al. 2013) :

1. appariement des mots de la requête initiale aux entités sémantiques de la base de connaissances ;
2. construction de graphes de requêtes liant les entités détectées à l'étape précédente ;
3. classement des requêtes construites ;
4. sélection de la bonne requête par l'utilisateur.

Notre initiative de génération de requête suit les mêmes étapes, mais avec un processus d'appariement entre requête en langue naturelle et base de connaissances qui est différent (reconnaissance d'entités nommées contre annotation sémantique dans notre cas) et des

⁷⁰ <http://wiki.dbpedia.org/>

⁷¹ <http://openuplabs.tso.co.uk/demos/sparqleditor>

graphes de requêtes qui sont également différents (graphes de requêtes *ad hoc* construits en cohérence avec la base de connaissances considérée).

Le principe est simple : la première étape est l'annotation de la requête en langue naturelle par des concepts de l'ontologie du domaine. La deuxième étape consiste en la construction de la requête *Sparql*, à partir des modèles de requêtes prédéfinis et des concepts résultats de l'annotation.

```
PREFIX onto :< http://www.communities.org/ontologies/communities.owl#>
SELECT ?User
WHERE {
  ?User rdf:type onto:User.
  ?User onto:hasInterest onto:ConceptResultOfTheAnnotation.}

```

Table 14: Modèle de requête *Sparql*.

Ce modèle de requête permet de générer une requête *Sparql* qui permet de sélectionner les utilisateurs qui ont pour centre d'intérêt le concept résultat de l'annotation de la requête initiale. Ces modèles prédéfinis ont été utilisés pour chercher des communautés et des services. Nous fournissons plus d'exemples de génération de requêtes *Sparql* dans les sections 6.5.2 et 6.6.2.

Bien que les résultats des travaux de génération de requêtes formelles soient prometteurs, ils sont toujours confrontés à des problèmes liés au langage naturel (*e.g.* incohérence, ambiguïté, *etc.*), à la forte dépendance aux domaines, *etc.* Ces problèmes ne font pas l'objet de notre contribution. Etant donné qu'il n'existe pas un outil de génération requête indépendant des domaines et permettant de répondre à nos besoins, nous avons conçu notre propre générateur de requête basés sur des graphes de requêtes *ad hoc*.

Le troisième composant du Manager de communauté est présenté dans la section suivante.

6.4.4 Le calcul de proximité

Ce composant de l'architecture permet de mesurer le degré de proximité sémantique et de proximité géographique utilisé pour la recommandation des communautés et des services.

7.1.1.1 La proximité sémantique

Dans le domaine de la recherche d'information, les mesures de similarité sont utilisées pour évaluer la proximité sémantique entre la requête et le document. Dans le contexte de nos applications communautaires, nous évaluons la similarité sémantique entre les profils des utilisateurs et des communautés et des services en rattachant les éléments des profils aux concepts de l'ontologie du domaine. Dans la littérature, plusieurs approches ont été proposées pour mesurer la similarité sémantique entre deux concepts d'une ontologie dans plusieurs domaines d'application comme le traitement automatique du langage naturel, la bio-informatique, la détermination de la similarité des services web, *etc.* On peut distinguer trois grandes approches : celles basées sur les arcs, celles basées sur les nœuds, et les approches hybrides (Slimani et al. 2007).

- i. *Les approches basées sur les arcs* : se servent de la structure hiérarchique de l'ontologie pour déterminer la similarité sémantique entre les concepts. Ces approches reposent sur l'idée suivante : plus le chemin entre deux nœuds est court plus ces nœuds sont semblables. Parmi les travaux appartenant à cette famille on peut citer : la mesure de similarité de *Wu & Palmer* (Wu et al. 1994) et la mesure de *Ehrig et al.* (Ehrig et al. 2005).
- ii. *Les approches basées sur les nœuds* : la similarité sémantique entre deux concepts est mesurée par la quantité de l'information qu'ils partagent. Cette approche se base sur le calcul du contenu informationnel d'un concept dans le corpus en tenant compte de sa spécificité ou généralité. La notion de contenu informationnel (*CI*) a été proposée pour la première fois par *Resnick* (Resnik 1995). Elle traduit la pertinence d'un concept dans le corpus. Il est calculé à partir de la fréquence d'apparition d'un concept (*c*) et les concepts qu'il subsume dans le corpus. La formule est définie par :

$$CI(c) = -\log(P(c))$$

Équation 1: Le contenu informationnel.

Où $P(c)$ est la probabilité de trouver une instance du concept (c). La probabilité d'un concept (c) est calculée en divisant le nombre des instances de (c) par le nombre total des instances. Parmi les travaux basés sur le calcul du contenu informationnel on peut citer : la mesure de *Resnick* (Resnik 1995) et la mesure de *lin* (Lin 1998).

- iii. *Les approches hybrides* : Ces approches combinent les approches basées sur les arcs (i.e. distances) et celles basées sur le calcul du contenu informationnel. Parmi les travaux qui adoptent les deux approches précédentes combinées, on trouve la mesure de *Jiang* (Jiang et al. 1997).

Dans notre travail, nous avons adopté l'approche basée sur les arcs pour évaluer la similarité sémantique entre les éléments des profils (utilisateurs, communauté) et (communauté, service). Notre choix est motivé par le fait que les deux autres types d'approches (i.e. approches basées sur les nœuds et approches hybrides) sont basés sur le calcul du contenu informationnel des concepts de l'ontologie du domaine. Or, ce calcul est non révélateur de sens dans le cas des communautés spontanées car ses dernières sont détachées de tout corpus de documents (elles ne sont pas créées autour des concepts courants ou fréquents dans un corpus). C'est pourquoi, nous avons choisi la mesure de *Wu et Palmer* (Wu et al. 1994). Elle présente l'avantage d'être simple à implémenter et de présenter de bonnes performances comparativement à d'autres mesures de similarité (Slimani et al. 2007).

La mesure de Wu et Palmer (Wu et al. 1994) : La similarité sémantique entre deux concepts ($c1$) et ($c2$) est calculée à partir de la distance (nombre d'arcs) qui sépare les deux concepts dans la hiérarchie et de leur position par rapport à la racine. La formule de calcul est la suivante :

$$wup(c1, c2) = \frac{2 * profondeur(c)}{dist(c1, c) + dist(c2, c) + 2 * profondeur(c)}$$

Équation 2: La mesure de similarité de Wu & Palmer.

Où (c) est le concept le plus spécifique qui subsume les deux concepts ($c1$) et ($c2$), $profondeur(c)$ est le nombre d'arcs qui séparent (c) de la racine et $dist(c_i, c)$ le nombre d'arcs qui séparent (c_i) de (c).

La mesure de Wu & Palmer est utilisée pour évaluer la similarité sémantique entre deux concepts (éléments des profils (utilisateurs, communauté) et (communauté, service) après leur projection sur l'ontologie du domaine). Mais l'utilisateur, la communauté, ainsi que le service peuvent avoir plus d'un concept dans leurs profils respectifs. Notre problème pour mesurer la similarité sémantique entre concepts des profils peut se ramener à mesurer la similarité sémantique entre deux ensembles de concepts. A notre connaissance, il existe un bon nombre de mesures de similarité sémantique entre deux concepts mais pas entre deux ensembles de concepts.

Pour ce faire, nous avons utilisés la notion de distance en mathématique entre deux ensembles de points. Quand on parle de distance, nous entendons généralement la distance la plus courte. Par exemple, si un ensemble de points A est à une distance D d'un autre ensemble de point B , nous supposons que D est la distance la plus courte entre tous points de A et tous points de B . Cette distance est appelée distance minimum (*i.e. minimum function*) et formulée comme suit :

$$Dmin(A, B) = \min_{a_i \in A} \left(\min_{b_j \in B} (1 - wup(a_i, b_j)) \right)$$

Équation 3: Distance minimum pour mesurer la distance entre deux ensembles de concept.

Sachant que $(1-wup(a_i, b_j))$ est la distance qui sépare a_i de b_j en utilisant la mesure de similarité de Wu et Palmer.

Cette équation se lit de la façon suivante : « pour chaque point a_i de l'ensemble A , il faut trouver la plus petite distance par rapport à tous les points de B et finalement garder la plus petite distance parmi celles calculées pour tous les points de A . »

Cette distance minimum est peu satisfaisante dans plusieurs applications car elle ne prend en compte que les deux points les plus proches des deux ensembles et ignore les distances entre les autres points. Autrement dit, la distance minimum ne considère pas l'ensemble des points mais seulement les deux points les plus proches.

Pour cette raison, nous avons opté pour la distance de *Hausdorff* qui remédie à cet inconvénient. La distance de Hausdorff, une mesure de similarité très utilisée pour mesurer l'éloignement de deux ensembles d'un espace métrique sous-jacent. Elle est définie comme « la distance maximale qui sépare un ensemble du point le plus proche de l'autre ensemble » (Rote 1991).

La Figure 34 compare les deux distances (distance minimum et distance de *Hausdorff*). La distance minimum donne le même résultat $D_{min}(A, B) = D_{min}(C, D)$ malgré l'évidence de la disparité des distances entre les couples (A, B) et (C, D). En revanche, la distance de Hausdorff considère tous les points dans le calcul et retourne la distance maximale qui sépare un ensemble du point le plus proche de l'autre ensemble, soit un résultat plus précis et plus significatif.

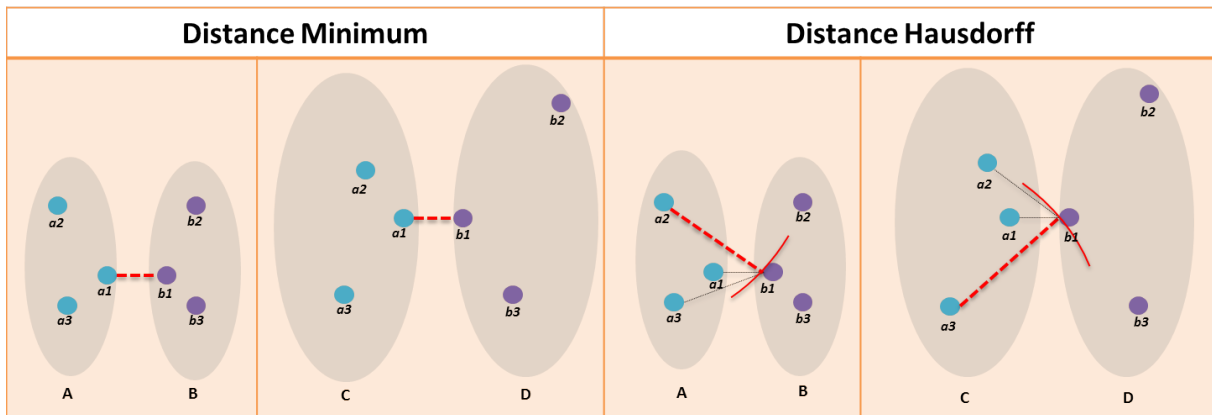


Figure 34: Comparaison entre distance minimum et distance de *Hausdorff*.

Dans notre travail, nous proposons la définition suivante du problème pour la mesure de similarité sémantique entre deux ensembles de concepts:

On note a_i et b_j des éléments des ensembles A et B tels que $a_i \in A$ et $b_j \in B$. Ce sont également des concepts C de l'ontologie du domaine Ω tel que a_i et $b_j \in C(\Omega)$.

Nous considérons $(1 - wup(a_i, b_j))$ comme la distance sémantique entre les deux concepts a_i et b_j . En effet, wup est une mesure de similarité sémantique, tandis que la mesure de *Hausdorff* utilise des distances. Sachant que

la distance sémantique = 1 - mesure de similarité. L'utilisation de la mesure de Hausdorff est acceptable car on considère que tous les éléments $C(\Omega)$ possèdent le même poids. La distance de Hausdorff est une distance sur l'ensemble $C(\Omega)$. Elle est présentée par l'équation suivante :

$$DH(A, B) = \max \left\{ \max_{a_i \in A} \left(\min_{b_j \in B} (1 - wup(a_i, b_j)) \right), \max_{b_j \in B} \left(\min_{a_i \in A} (1 - wup(a_i, b_j)) \right) \right\}$$

Équation 4: Distance de Hausdorff pour mesurer la distance entre deux ensembles de concept.

On désigne par $DH(A, B)$ la distance de Hausdorff entre deux ensembles fermés bornés non vides A et B d'un espace métrique E . La distance de Hausdorff vérifie les propriétés simples d'une distance en mathématique :

Nom	Propriété
Symétrie	$\forall (A, B) \in E^2, DH(A, B) = DH(B, A)$
Séparation	$\forall (A, B) \in E^2, DH(A, B) = 0 \rightarrow A = B$
Inégalité triangulaire	$\forall (A, B, C) \in E^3, DH(A, C) \leq DH(A, B) + DH(B, C)$

Table 15: Propriétés simples d'une distance.

Avec la distance Hausdorff, on peut confirmer que n'importe quel élément $a_i \in A$ est au maximum à une distance égale à $DH(A, B)$ de n'importe quel élément de $b_j \in B$. On présente dans la **Table 16**, l'algorithme qui illustre le calcul de la distance de Hausdorff.

ALGORITHME : Hausdorff Distance

```

1. Fonction MaxMin (A, B)
2. dh (A, B) = 0
3. Pour tout point  $a_i \in A$ 
   LaPlusPetiteDistance = infini
   Pour tout point  $b_j \in B$ 
     distanceij = 1 - wup (ai, bj)
     if distanceij < LaPlusPetiteDistance donc
       LaPlusPetiteDistance = distanceij
   if LaPlusPetiteDistance > dh (A, B) donc
     dh (A, B) = LaPlusPetiteDistance
4. Debut
    
```

5. $dh^A = \text{MaxMin} (a_i, b_j)$
6. $dh^B = \text{MaxMin} (b_j, a_i)$ (Pour la propriété d'asymétrie)
7. $DH(A, B) = \max \{dh^A (a_i, b_j), dh^B (b_j, a_i)\}$

8. **Fin**

Table 16: Algorithme de la distance de Hausdorff.

7.1.1.2 La proximité géographique

A la différence de la proximité sémantique utilisée pour la recommandation des communautés et des services, la proximité géographique est utilisée seulement pour mesurer la correspondance entre la localisation d'un utilisateur et le territoire géographique d'une communauté pour affiner la recommandation des communautés.

Pour calculer la proximité géographique, on revient sur la notion de territoire de la communauté (cf. section 5.2.1). Ce territoire est défini par un espace délimité géographiquement qui regroupe l'ensemble des membres des communautés. Ce territoire peut être fixe ou dynamique :

Un territoire géographique fixe correspond à une zone géographique figée dès la création de la communauté.

Un territoire géographique dynamique désigne une zone géographique calculée en fonction des localisations des membres de la communauté. A chaque intervalle de temps prédéfini, on délimite les contours de l'enveloppe convexe des positions géographiques des membres de la communauté, c'est-à-dire la zone géographique qui englobe tous les membres. On formulera la problématique de la proximité géographique comme suit :

Proximité_géographique (localisation user, territoire communauté)

Proximité_géographique est une fonction qui permet de savoir si un utilisateur se trouve dans le territoire d'une communauté donnée ou non. Cette fonction utilise les positions géographiques reçues par les dispositifs mobiles des membres de la communauté (pour calculer le territoire dynamique) et de l'utilisateur.

En tout premier lieu, l'examen de quelques éléments de référence conduit à comprendre notre raisonnement pour examiner la proximité géographique. Plusieurs outils comme Google Maps API⁷² permettent de définir un territoire géographique (cercle ou polygone) à partir des attributs en entrée. L'exemple suivant montre un exemple de territoire géographique sous forme de cercle délimité à partir d'un point GPS et d'un rayon.

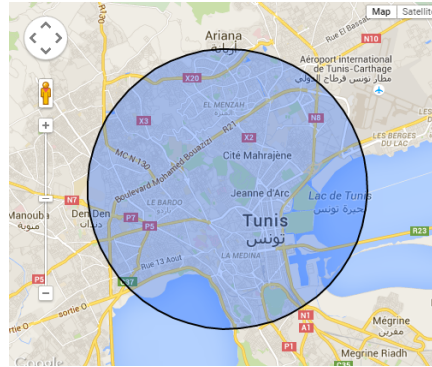


Figure 35: Territoire géographique sous forme d'un cercle.

Pour la délimitation du territoire dynamique de la communauté nous avons considéré le problème de l'emprise géométrique minimale (*i.e. Minimum Bounding Geometry*) (*cf.* Figure 36). Il s'agit d'une emprise géométrique qui délimite une entité ou un groupe d'entités en entrée (comme des points, des lignes et des polygones) pour produire des entités en sortie (comme des cercles⁷³, des enveloppes convexes⁷⁴, *etc.*).

⁷² <http://www.geocodezip.com/>

⁷³ Le problème du cercle minimum consiste à trouver le cercle le plus petit contenant un ensemble de points d'un plan.

⁷⁴ L'enveloppe convexe d'un objet ou d'un regroupement d'objets géométriques est l'ensemble convexe le plus petit parmi ceux qui le contiennent.

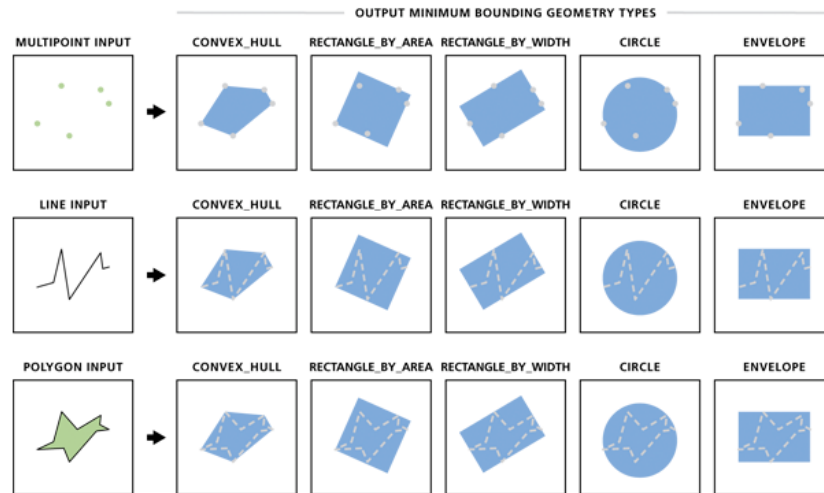


Figure 36: Emprise géométrique minimale⁷⁵.

Dans notre travail, nous optons pour le problème de l’enveloppe convexe car la représentation d’un territoire sous forme de polygone est supportée par l’encodage *GeoRSS* utilisé pour décrire les services (cf. section 5.2.1). De plus c’est la forme la plus répandue au niveau de la communauté système d’information géographique. Plusieurs outils offrent la fonctionnalité de calculer l’enveloppe convexe à partir d’un ensemble de points (longitude, latitude) donné en entrée comme *ArcGIS*⁷⁶, *Mapbox*⁷⁷ ou *Google Maps API*⁷⁸.

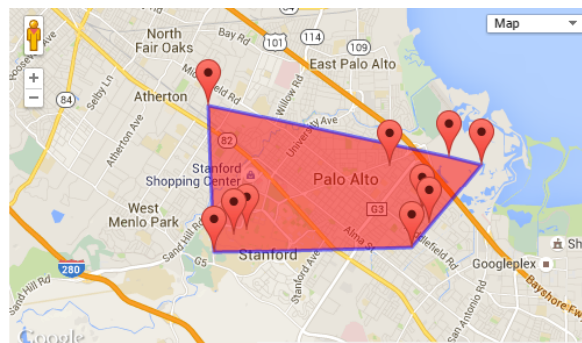


Figure 37: L'enveloppe convexe d'un ensemble de points (longitude, latitude) en entrée.

⁷⁵ <http://pro.arcgis.com/fr/pro-app/tool-reference/data-management/minimum-bounding-geometry.htm>

⁷⁶ <https://www.arcgis.com/features/>

⁷⁷ <https://www.mapbox.com/>

⁷⁸ http://www.geocodezip.com/v3_map-markers_ConvexHull.asp

Ainsi, deux techniques sont au cœur de notre algorithme de calcul de proximité géographique : l'enveloppe convexe représentant le territoire de la communauté et le géorepérage pour vérifier l'existence ou non de l'utilisateur dans ce territoire.

Les algorithmes de calcul de l'enveloppe convexe comme le parcours de *Graham*⁷⁹, la marche de *Jarvis*⁸⁰, ou l'algorithme *Andrew's Monotone Chain*⁸¹, permettent de générer un polygone à partir d'un ensemble de points.

Ensuite, le géorepérage se base sur les algorithmes de type point dans un polygone (*PIP*⁸²) qui permettent de vérifier si un point particulier est à l'intérieur d'un polygone ou en dehors de celui-ci. Le problème PIP est résolu en comptant combien de fois une ligne tracée à partir du point donnée croise le bord du polygone. Si la ligne et le polygone se croisent un nombre de fois pair (*cf.* point 2 de la Figure 38) ou pas du tout, le point est à l'extérieur. S'ils se croisent un nombre de fois impair (*cf.* point 1 et 3 de la Figure 38), le point est à l'intérieur.

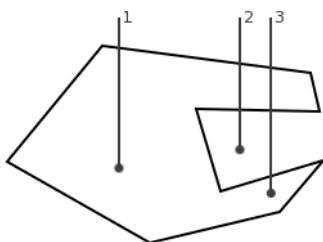


Figure 38: Exemple du problème du point dans un polygone.

Dans ce qui suit, nous proposons l'algorithme de calcul de proximité géographique présenté dans la Table 17 qui permet de vérifier l'existence d'un utilisateur dans le territoire de la communauté ou non. Dans cet algorithme, nous proposons de parcourir les membres d'une communauté donnée et de collecter leurs positions géographiques (*cf.* Table 17, ligne 5-7). Puis, nous calculons l'enveloppe convexe qui regroupe toutes les positions récupérées. Le

⁷⁹ https://fr.wikipedia.org/wiki/Parcours_de_Graham

⁸⁰ https://fr.wikipedia.org/wiki/Marche_de_Jarvis

⁸¹ Algorithm Implementation/Geometry/Convex hull/Monotone chain

⁸² https://en.wikipedia.org/wiki/Point_in_polygon

résultat de cette opération est un polygone représentant le territoire de la communauté (cf. Table 17, ligne 8). Une fois que l'on a l'enveloppe convexe, on peut tester si la position de l'utilisateur (sous forme d'un point ou couple de coordonnées (longitude, altitude)) se trouve à l'intérieur ou l'extérieur du polygone (cf. Table 17, ligne 9). La fonction retourne 1 si l'utilisateur est dans le territoire de la communauté, 0 sinon.

ALGORITHME : Proximité_géographique

Debut

```
1. LatLong LocalisationUser
2. LatLong PositionsMembres []
3. Polygone TerritoireCommunauté
4.   Parcourir les membres d'une communauté donnée
5.     PositionsMembres [i]= membre.getLocalisation()
6.   Fin Parcourir
7.   TerritoireCommunauté.CalculEnveloppeConvexe (PositionsMembres [])
8.   Si (TerritoireCommunauté.contient (LocalisationUser)) alors
9.     Retourner 1;
10.  Sinon
11.    Retourner 0 ;
12.  Fin si
```

Fin

Table 17: Algorithme de la proximité géographique.

Pour le cas de la recommandation de communauté, la pertinence globale d'une communauté par rapport à un utilisateur est une pondération entre la proximité sémantique et la proximité géographique. Le but de la pondération est de pouvoir ajuster, si on le souhaite, le poids de chaque valeur de proximité en fonction des besoins.

Dans ce qui suit, nous présentons le quatrième composant du *Manager* de communautés.

6.4.5 Le moteur de règles

Le moteur de règles est appelé aussi raisonneur ou moteur d'inférence. Un moteur de règles est un moteur d'appariement ou de rapprochement entre des faits et des règles afin de

traiter les faits selon l'objectif exprimé par les règles ou d'inférer de nouvelles connaissances ou encore de vérifier l'intégrité sémantique de la base de connaissances.

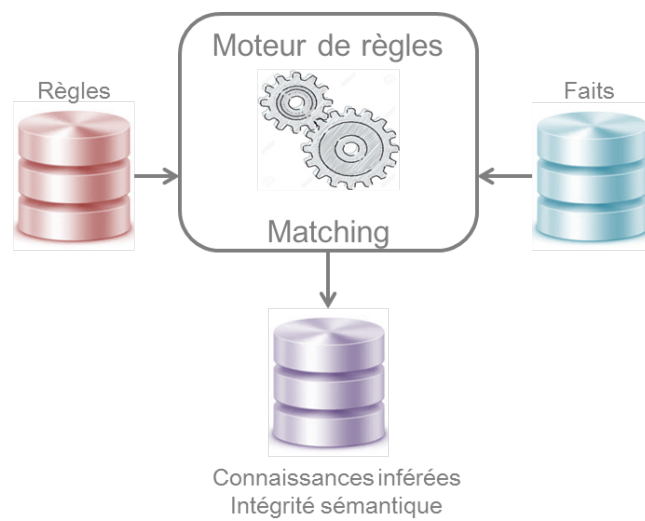


Figure 39: Architecture des systèmes à base de règles.

Les services offerts par les raisonneurs permettent de vérifier la satisfiabilité, de faire de la subsumption, de vérifier la cohérence et de classer les individus (*cf.* section 5.3). Le moteur de règles exécute une ou plusieurs règles provenant de la base de connaissances (liées à l'ontologie). Deux types de règles sont utilisés par notre système : les règles basées sur la logique de description (LD) et les règles *SWRL* (*cf.* section 5.3).

Après avoir défini les règles d'une base de connaissances, le raisonneur peut faire des inférences en créant automatiquement des connaissances qui enrichissent la base de connaissances communautaires. Des exemples d'inférence de connaissances sont présentés dans la section 5.3. Le dernier composant du *Manager* de communauté est présenté ci-après.

6.4.6 Le contrôleur de confidentialité

La tendance vers l'individualisation, la personnalisation et l'adaptation a fait émerger plusieurs problématiques concernant la vie privée et les données personnelles. Soutenir la protection de vie privée et le contrôle de données lors de la constitution et l'exploitation des

ressources par les entités sociales est la préoccupation majeure de plusieurs travaux de recherche comme :

- *D-Foaf* (Kruk et al. 2006) un système de gestion d'identités et des droits d'accès aux données, basé sur l'ontologie *Foaf* et déployé sur les réseaux sociaux.
- (Carminati et al. 2011) Un *Framework* de contrôle d'accès aux données déployable sur les réseaux sociaux et basé sur les technologies du web sémantique.
- Ou encore (Fong et al. 2009) un modèle de contrôle d'accès et de protection de la vie privée dédié à *Facebook* et basé sur des règles.

Au niveau de notre architecture, nous nous intéressons au contrôle d'accès aux données. Tout d'abord, Taldea garantit un accès personnel à l'espace social en passant par le formulaire d'authentification. Ensuite, la question de la protection des données est traitée par le développement d'un contrôleur de confidentialité qui surveille l'accès aux ressources en s'inspirant de modèle très connu des réseaux sociaux (privé, amis, public). Dans ce travail, on ne s'intéresse pas aux autres aspects de sécurité des données (*e.g.* cryptographie).

Notre contrôleur de confidentialité vérifie si une entité (personne, communauté) demandant l'accès à une ressource (contenu, service) a les droits nécessaires pour le faire. Le modèle adopté permet de régler les paramètres d'accès aux ressources selon les préférences des membres de la communauté.

1. **Privé** : l'accès aux ressources de la communauté est limité aux membres de la communauté.
2. **Amis** : procéder par vérification des règles pour autoriser une entité (personne, communauté) à accéder à une ressource. Pour exécuter les règles qui formalisent la logique d'accès aux données, les informations relatives aux entités sociales et aux ressources (*i.e.* le profil des entités sociales, les relations entre entités, les relations entre entité et ressource, *etc.*) doivent être représentées selon un formalisme de représentation de connaissances. Dans la Table 18, un exemple de règles *SWRL* qui permet de limiter l'accès aux ressources est présenté.

Règle 1	$EstCreateur(?m, ?r), estMembre(?m, ?x) \rightarrow estConcédant(?x, ?r)$
Règle 2	$Communaute(?x), Communaute(?y), estConcédant(?x, ?r), attribuerDroitsA(?x, ?y) \rightarrow exploiter(?y, ?r)$

Table 18: Exemple de règles d'accès aux données.

La **règle 1** accorde les droits d'exploitation d'une ressource r à une communauté x si un de ses membres m est le créateur de cette ressource.

La **règle 2** permet à une communauté x concédant les droits d'une ressource r d'attribuer des droits de (lecture attribuerDroitsR, écriture attribuerDroitsW, tous attribuerDroitsA) à une communauté y .

3. Public : L'accès est ouvert à tout le monde, c'est-à-dire à tous les utilisateurs de Taldea.

Depuis l'explosion des réseaux sociaux éphémères, il est important de prendre en compte la durée de vie des données qui présente un impact positif sur la vie privée. Les données personnelles ne persistent plus éternellement en ligne. A cet effet, le concédant des droits d'une ressource peut définir sa durée de vie.

Dans les sections suivantes, nous décrivons les interactions des différents composants de l'architecture présentés précédemment pour assurer l'accès aux communautés et la découverte des services.

6.5 L'accès aux communautés

6.5.1 La recommandation

L'objectif principal de la recommandation est de gérer la surcharge d'information en suggérant des *items* (*i.e.* contenus ou services) à l'utilisateur. Dans notre cas, la recommandation permet de fournir à l'utilisateur une liste des communautés et de services susceptibles de l'intéresser. Il existe différentes formes de recommandation (Poirier 2011) dont notamment :

- *La recommandation éditoriale* qui met en avant les items les plus populaires, les nouveautés, les mieux notés, *etc.* Ce genre de recommandation est utilisé lorsque l'on n'a pas d'information sur l'utilisateur.

- *La recommandation sociale* qui se base sur le comportement passé des utilisateurs similaires, en effectuant une corrélation entre des utilisateurs ayant des préférences et des intérêts similaires. Parmi les démarches les plus connues pour ce type de recommandation, on trouve le filtrage collaboratif (*i.e.* utiliser l’appréciation d’un groupe d’utilisateur sur les items).
- *La recommandation contextuelle* qui permet de proposer des items proches de l’item consulté. Le principe de cette approche est de rapprocher des items selon les caractéristiques et les propriétés de l’item lui-même (genre, couleur, auteur, *etc.*)
- *La recommandation personnalisée* qui, comme son nom l’indique, est une recommandation réalisée en fonction de l’utilisateur. Elle permet de découvrir les items susceptibles d’être intéressants pour l’utilisateur, selon ses habitudes, sa personnalité, mais aussi son humeur du moment, sa culture, sa localisation et plusieurs autres facteurs (*i.e.* les items sont recommandés selon leur proximité du profil de l’utilisateur).

Dans notre travail, nous nous sommes intéressés à la *recommandation personnalisée* pour les communautés et à la *recommandation sociale* pour les services. Pour le mécanisme de recommandation de communauté, la pertinence globale d’une communauté par rapport à un utilisateur est calculée en fonction de la similarité sémantique (qui mesure la correspondance entre le profil de l’utilisateur et les profils des communautés en se basant les intérêts de chacun), et la proximité géographique (qui mesure la correspondance entre la localisation d’un utilisateur et le territoire géographique d’une communauté). Le mécanisme de recommandation de service consiste à recommander pour une communauté les services qui ont été utilisés par des communautés sémantiquement proches.

Plusieurs méthodes de recommandation ont été proposées dans la littérature comme les méthodes basées sur le *clustering* ou les modèles bayésiens, les méthodes basées sur le calcul de proximité (similarité Cosinus, similarité de Pearson, *etc.*). Nous décrivons à présent la méthode de recommandation de communauté que nous avons adoptée, basée sur la mesure de similarité sémantique de *Wu & Palmer*, sur la distance de *Hausdorff* (*cf.* section 7.1.1.1) et sur la proximité géographique (*cf.* section 7.1.1.2).

Dans notre travail, nous proposons la définition suivante du problème pour la mesure de similarité sémantique entre les intérêts de l'utilisateur et les intérêts de la communauté :

- On note une communauté $CK \in C$ possédant un ensemble de centre d'intérêt I^k .
- On note un utilisateur $UH \in U$ possédant un ensemble de centres d'intérêt CI^h .

Les centres d'intérêt I_i^k et CI_j^h sont respectivement des éléments des ensembles I^k et CI^h tels que $I_i^k \in I^k$ et $CI_j^h \in CI^h$. Ce sont également des concepts C de l'ontologie du domaine Ω tel que I^k et $CI^h \in C(\Omega)$.

L'idée de base de la recommandation est de calculer la distance Hausdorff entre les intérêts de l'utilisateur et les intérêts de chaque communauté $DH(CI^h, I^k)$, puis de proposer à l'utilisateur les communautés les plus proches (en termes de distance) de son profil. En prenant l'exemple d'une communauté qui possèdent l'ensemble des intérêts suivant $I = \{i_1, i_2\}$ et un utilisateur qui possède à son tour les centres d'intérêt suivants $CI = \{ci_1, ci_2, ci_3\}$. On calcule la distance $(1-wup(i_j, ci_k))$ pour toutes les combinaisons de points des deux ensembles. Ensuite, on compare le minimum de $\{(1-wup(i_1, ci_1)), (1-wup(i_1, ci_2)), (1-wup(i_1, ci_3))\}$ avec le minimum de $\{(1-wup(i_2, ci_1)), (1-wup(i_2, ci_2)), (1-wup(i_2, ci_3))\}$ et l'on retient le maximum des deux valeurs obtenues comme valeur de $DH(CI^h, I^k)$.

Nous avons ainsi défini un algorithme pour recommander des communautés à un utilisateur donné en fonction des centres d'intérêt indiqués dans son profil et de sa position géographique. Au début de l'algorithme présenté dans la Table 19, nous parcourons la liste des communautés existantes (cf. Table 19 ligne 1). Ensuite, nous projetons les centres d'intérêt de l'utilisateur et les intérêts de la communauté sur l'ontologie du domaine (cf. Table 19 ligne 2). Puis, nous calculons la similarité sémantique entre les concepts centre d'intérêt de l'utilisateur et les concepts intérêt des communautés existantes à l'aide la distance de Hausdorff (cf. Table 19 ligne 3). Nous évaluons alors la proximité géographique (cf. Table 19 ligne 4). Le score de pertinence de la communauté en question par rapport à l'utilisateur est déterminé comme la somme pondérée de la similarité sémantique et la proximité géographique (cf. Table 19, ligne 5). Seules les communautés dont le score de pertinence dépasse un certain seuil sont incluses dans le résultat. (cf. Table 19, ligne 6). Les valeurs du

seuil et les paramètres de pondération (α et β) sont fixées de manière empirique et comprises dans l'intervalle [0..1].

ALGORITHME : Recommandation de communauté

Debut

1. **Parcourir** les communautés existantes
2. Projeter les centres d'intérêt de l'utilisateur et les intérêts de la communauté sur l'ontologie du domaine.
3. $Similarité_sémantique = 1 - DH(EnsembleInteretUtilisateur, EnsembleInteretCommunaute);$
4. $Proximité_géographique (localisation\ user, territoire\ communauté) ;$
5. $ScorePertinence = \alpha \cdot Similarité_sémantique + \beta \cdot Proximité_géographique ;$
6. $RecommanderCommunautes(ScorePertinence > Seuil);$
7. **Fin parcourir**

Fin

Table 19: Algorithme de recommandation de communauté.

6.5.2 La recherche

Nous avons proposé une conceptualisation de la communauté, en tant qu'objet central de nos applications, qui fournit un support formel décrivant explicitement une communauté et pouvant servir de modèle pour d'autres applications communautaires. Dans notre travail, nous utilisons deux ressources sémantiques (ontologie du domaine et ontologie de communauté) pour effectuer une recherche sémantique sur les communautés.

Avec Taldea, l'utilisateur peut introduire des requêtes en langage naturel afin de rechercher les communautés dont les centres d'intérêt correspondent à une thématique souhaitée. La différence par rapport au service de la recommandation réside dans le fait que la recherche renvoie pour résultats des communautés dont la similarité sémantique correspond à 1 par rapport à des besoins de l'utilisateur exprimés sous forme de requête en langage naturel. Contrairement à la recommandation qui renvoie pour résultat des communautés dont la similarité par rapport au profil est comprise entre 0 et 1.

La recherche sémantique est une suite de plusieurs opérations. Elle utilise les trois composants suivants de l'architecture : le processeur d'annotation, le moteur de requête, le contrôleur de confidentialité.

La première étape consiste à recourir au processeur d'annotation. La requête de l'utilisateur est annotée avec des concepts de l'ontologie du domaine (dans notre cas, l'ontologie botanique). La deuxième étape est la génération d'une requête formelle par le moteur de requête. Une requête en *Sparql* est formulée à partir des modèles de requêtes prédéfinis et des concepts résultats de l'annotation.

Prenons l'exemple d'un utilisateur qui recherche des communautés s'intéressant aux orchidées. La requête textuelle de l'utilisateur est annotée via le processeur d'annotation. Le résultat de l'annotation est le concept 'orchidée'. Ensuite, une requête *Sparql* est formulée avec le concept 'orchidée' et en suivant le modèle de requête *Sparql* prédéfinis (cf. section 6.4.3).

Des communautés qui s'intéressent aux orchidées.

Table 20: Exemple de requête de l'utilisateur.

```
PREFIX onto :< http://www.communities.org/ontologies/communities.owl#>  
Select ?communauté  
Where {  
  ?communauté rdf:type onto:Communités.  
  ?communauté onto:hasInterest onto:"orchidée". }  
}
```

Table 21: Exemple de requête *Sparql* générée pour la recherche de communauté.

La base de connaissances est interrogée en utilisant la requête générée. Le résultat est envoyé au contrôleur de confidentialité pour vérifier si l'utilisateur a bien les droits nécessaires pour consulter la ou les communautés résultats.

6.5.3 La création

L'application communautaire Taldea assiste l'utilisateur dans la création d'une communauté à travers un formulaire. Les données introduites sont annotées (cf. section 6.4.2)

et les concepts résultats peuplent l'ontologie de la communauté. Les données introduites sont des informations qui concernent la communauté (*e.g.* son nom, son intérêt, *etc.*) et des valeurs concernant ses paramètres de confidentialité. Après l'instanciation de la communauté avec ces informations, le moteur de règles est mis en œuvre. L'objectif principal de ce processus est d'enrichir la base de connaissances constituée des instances de l'ontologie des communautés. La base de connaissances contient les informations introduites par l'utilisateur, ainsi que des informations inférées par le raisonneur.

Une fois la communauté créée, l'utilisation des vocabulaires contrôlés (*e.g.* *Foaf*, *Sioc*) assure une structuration et une homogénéité de l'information échangée. Dans notre application communautaire, l'ontologie *Sioc* (*cf. section 5.2.1*), ainsi que l'ensemble des classes associées (*i.e.* *Sioc:item*, *Sioc:space*, *Sioc:container*, *etc.*), représentent la formalisation de l'information échangée entre les membres de la communauté. La classe *Sioc:item* possède l'avantage d'être générique, ce qui permet de générer un code qui pourra facilement s'adapter à d'autres contextes (*e.g.* les différents réseaux sociaux). Il est possible d'utiliser des classes plus précises dans des contextes spécifiques comme *BlogSpot*, *BoardPost*, *Comment*, *InstatMessage*, *MailMessage*, *MicroblogPost*, *WikiArticle*, *etc.* Le tableau suivant décrit un contenu partagé sur l'espace social de la communauté créée « *Jour de l'arbre* ».

```
<Sioc:Item rdf:about="http://www.communties.org/ontology# invitation pour
jour de l'arbre" rdf:type="Sioc:Post">
<Sioc:id>c44e74aaeb6837713a</Sioc:id>
<Sioc:name>invitation pour jour de l'arbre</Sioc:name>
<Sioc:has_creator rdf:resource=" http://www.communties.org/ontology#Ghada
"/>
<Sioc:has_container rdf:resource="http://www.communties.org/ontology#Jour
de l'arbre "/>
<Sioc:content rdf:resource=" Besoin de votre aide les botanistes, on vous
invite à planter ou à entretenir des arbres à l'occasion du jour d'arbre
afin de limiter les effets de la désertification."/>
<Sioc:has_reply rdf:resource=" Je vous rejoins de suite."/>
<Sioc:attachment rdf:resource="http://link_to_the_document" rdfs:label="Les
photos"/>
<Sioc:topic rdfs:label="Sujet" rdf:resource=" Arbre "/>
</Sioc:Item>
```

Table 22: Une description en *Sioc* d'un contenu partagé sur l'espace social de la communauté.

Nous utilisons le vocabulaire *Sioc* pour décrire un contenu (un item) partagé sur l'espace social d'une communauté. Un item possède un identifiant '*Sioc:id*', un titre '*Sioc:name*', un créateur qui renvoie au membre qui a posté le message '*Sioc:has_creator*', l'espace commun sur lequel ce contenu a été partagé '*Sioc:has_container*', la description du contenu qui peut être directe ou un renvoi à une ressource '*Sioc:content*', les réponses à ce message '*Sioc:has_reply*', les documents attachées à ce message '*Sioc:attachment*', etc.

Pour échanger du contenu et des documents multimédia, la communauté créée a besoin de services. Nous proposons une stratégie de découverte de service en fonction du *contexte social* de l'utilisateur, c'est-à-dire des communautés dont il est membre.

6.6 La découverte des services

Comme présenté au début de ce chapitre, l'architecture Taldea repose essentiellement sur deux modules (cf. Figure 26). Dans les sections précédentes, nous avons présenté le premier module d'accès aux communautés qui permet de faciliter l'accès et l'interaction d'un utilisateur avec les communautés. Dans cette section, nous présentons une approche de découverte des services permettant d'assister les communautés et leurs membres dans la découverte des services les mieux adaptés à leurs besoins.

Contrairement aux autres approches de découverte de services (qui se limitent aux propriétés fonctionnelles du service (entrées, sorties, effets, préconditions, etc.), nous soulignons ici le rôle du contexte social de l'utilisateur en tant que source d'informations pour la découverte des services utilisés pour favoriser l'échange et le partage des informations entre les utilisateurs.

L'utilisateur, une fois connecté à une communauté, peut sélectionner parmi les services associés à cette communauté, ceux qui vont être déployés sur son dispositif. Pour une nouvelle communauté ou de nouvelles exigences, un grand nombre de services peuvent être disponibles. Trouver le service qui correspond le mieux à la communauté ou à ses nouvelles exigences, est une tâche difficile. Notre objectif est d'aider les communautés à surmonter la surcharge de services. Dans ce travail, nous proposons une approche de découverte des services relativement similaire à celle de la découverte de communauté, composée de trois

services de base : la recommandation, la recherche et la création (R, S, C). Ensuite, les services sélectionnés par l'utilisateur vont être déployés via la plateforme Kalimucho (Dalmau et al. 2013). Nos applications communautaires sont composées de services interconnectés supervisés par la plateforme. Un service est matérialisé par un composant logiciel assurant une fonctionnalité particulière et accessible via son interface. Nos services sont qualifiés de sémantiques car leur description est assurée via les vocabulaires contrôlés du web sémantique (*i.e. Owl-S*), fournissant aux applications les informations nécessaires à leur découverte et leur utilisation. La découverte de services sémantiques est un paradigme important pour le développement des applications distribuées. Elle est définie comme: *“La comparaison par paires d'une description de service et d'une requête de service pour déterminer le degré de leur correspondance sémantique.”* (Klusch et al. 2009b).

Plusieurs travaux dans la littérature ont adopté des architectures, des langages de description et des méthodes différentes pour la découverte sémantique de services. Les architectures de découverte sémantique de services peuvent être centralisées ou distribuées (*cf.* Figure 40). Les composants traditionnels de l'architecture orientée service ont été révisés pour donner lieu à de nouveaux composants : annuaire de service sémantique, moteur d'appariement, requête de service, annonce de service.

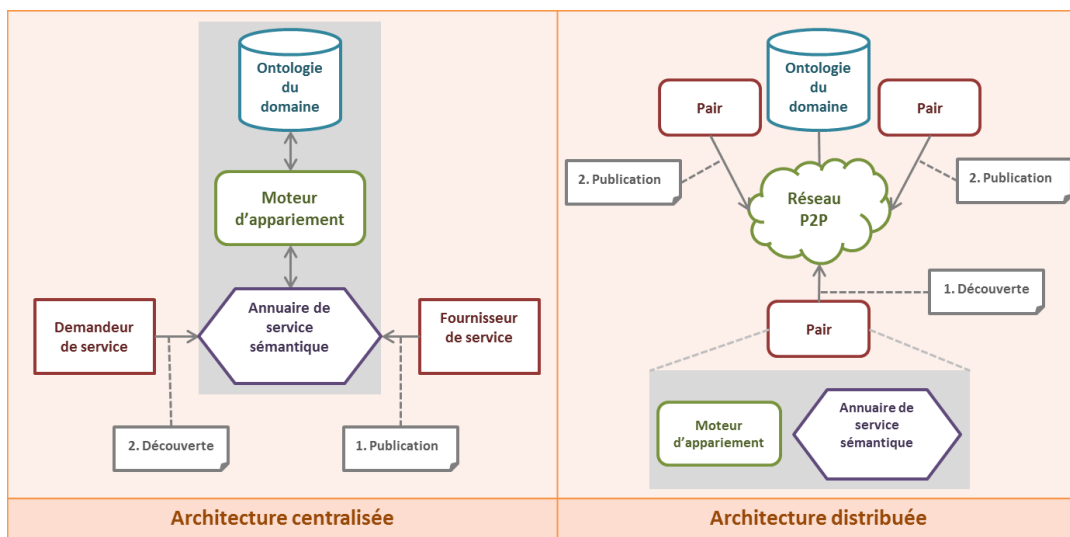


Figure 40: Architectures de la découverte de service sémantique.

Les deux conditions qui élèvent nos services au rang de services sémantiques sont l'utilisation d'un langage de description sémantique et l'annotation avec de l'information sémantique. Dans notre travail, nous avons utilisé le standard *Owl-S* (cf. section 3.2.2.4) parmi plusieurs langages comme *WSML* (De Bruijn et al. 2006), *WSML* (De Bruijn et al. 2006), *WSDL-S* (Akkiraju et al. 2005), *WSMO* (Domingue et al. 2005), etc.

Les méthodes d'appariement (*i.e. matching*) entre la description du service demandé et la description du service proposé peuvent reposer sur des approches basées sur la logique comme la similarité sémantique, l'appariement sémantique multi-niveau (Paolucci et al. 2002), ou sur des approches non basées sur la logique comme la théorie des graphes, la fouille de données, la similarité linguistique, etc. ou encore sur des approches hybrides.

Lors du processus d'appariement, différentes parties de la description du service peuvent être utilisées en fonction du langage utilisé. Par exemple pour une description en *Owl-S*, elle se compose de trois parties : le *service profile*, le *process model*, et le *grounding*, qui décrivent respectivement « que fait le service », « comment le service fonctionne » et « comment accéder au service ». L'appariement entre la description du service et la requête peut se faire sur la partie '*black-box*', c'est-à-dire le profil qui décrit les propriétés fonctionnelles et non fonctionnelles des services (cf. section 3.2.2.4), ou sur la partie plus transparente '*glass-box*' c'est-à-dire le *process model* qui décrit la sémantique des données échangées et comment fonctionne un service pour permettre l'invocation, la composition, la surveillance, la récupération, etc.

Plusieurs *Matchmakers* (un anglicisme désignant l'outil mettant en relation la requête de service et la description de service) ont été proposés, parmi lesquels on peut citer Yasa-m (Chabeb et al. 2010), SAWSDL (Wei et al. 2011) et OWLS-MX (Klusch et al. 2009a), etc. Ces outils utilisent des éléments différents de la description du service et des méthodes différentes dans leur processus de *matching*. On trouvera dans Klush (Klusch 2008) une étude approfondie des différentes approches de *matching* de services dans la littérature avec leurs différentes architectures, langages et description du service utilisées.

6.6.1 La recommandation

À partir des approches étudiées, nous avons constaté que la découverte de services sémantiques se base sur les propriétés fonctionnelles des services comme ses entrées, ses sorties, ses effets, ses préconditions, *etc.*, ou sur des propriétés non fonctionnelles comme la qualité de service, les règles de vie privée, la catégorie, *etc.* Toutes ses propriétés sont modélisées via des langages de description comme *WSMO*, *WSML*, *WSDL-S*, *Owl-S*, *etc.* (cf. section 3.2.2.4).

Dans cette section, nous proposons une approche de recommandation de service basée sur des propriétés non fonctionnelles du service autres que celles présentées précédemment. Autrement dit, le *matching* entre le profil des communautés et le profil des services est basé sur les entités sociales, propriétés du profil du service. Dans ce contexte, les entités sociales correspondent aux communautés qui ont utilisés le service.

Plusieurs langages de description de service représentent l'entité sociale dans la description du service. Par exemple, le standard *Owl-S* présente le composant du profil '*contactInformation*' qui peut référencer des entités sociales (individu ou groupe de personnes) en recourant à des ontologies tierces comme *Foaf*, *Vcard*, *etc.* Taldea exploite cette sémantique exprimée au niveau de l'ontologie *Owl-S* pour annoter un service par l'ensemble des communautés qui l'ont utilisé.

```
<!-- http://www.communities.org/ontology#TakePicture -->
<owl:NamedIndividual rdf:about="http://www.communities.org/ontology#TakePicture">
  <rdf:type rdf:resource="http://www.daml.org/services/owl-s/1.2/Profile.owl#Profile"/>
  <Profile:contactInformation rdf:resource="http://www.communities.org/ontology#FanVivace"/>
  <Profile:contactInformation rdf:resource="http://www.communities.org/ontology#Fleurcommunity"/>
  <Profile:textDescription rdf:resource="http://www.owl-ontologies.com/OntologieBotanique10.owl#CategorieFleur"/>
  <Profile:textDescription rdf:resource="http://www.owl-ontologies.com/OntologieBotanique10.owl#Fleur"/>
  <Profile:textDescription rdf:resource="http://www.owl-ontologies.com/OntologieBotanique10.owl#CategorieFleurVivace"/>
  <Service:presentedBy rdf:resource="http://www.communities.org/ontology#Taldea_TakePicture"/>
</owl:NamedIndividual>

<!-- http://www.communities.org/ontology#Taldea_TakePicture -->
<owl:NamedIndividual rdf:about="http://www.communities.org/ontology#Taldea_TakePicture">
  <rdf:type rdf:resource="http://www.daml.org/services/owl-s/1.2/Service.owl#Service"/>
  <Service:presents rdf:resource="http://www.communities.org/ontology#TakePicture"/>
</owl:NamedIndividual>
```

Figure 41: Extrait de la description en *Owl-S* du service "*TakePicture*".

La Figure 41 montre un extrait de la description en *Owl-S* d'un service nommé « Taldea_TakePicture », représenté par le profil « TakePicture » et référencé par les communautés « FanVivace » et « FleurCommunity ».

Comme présenté dans la section 6.5.1, la recommandation sociale s'efforce de prédire ce que l'utilisateur aimera en cherchant des utilisateurs qui ont les mêmes intérêts ou comportements.

L'idée de la recommandation de services est de recommander pour une communauté les services qui ont été utilisés par des communautés sémantiquement proches.

La nouveauté dans ce que nous proposons réside dans le fait que l'on recommande des services en se basant sur le contexte social. Dans notre cas, le contexte social est représenté par l'ensemble des concepts correspondant aux centres d'intérêt des communautés ayant utilisé ce service. Ainsi, la recommandation de service est basée sur une corrélation entre des communautés ayant des centres d'intérêt sémantiquement proches.

Pour l'approche de recommandation de service basée sur l'aspect social de sa description, nous avons utilisé les mêmes techniques que pour la recommandation des communautés en termes de calcul de similarité (mesure de similarité de *Wu & Palmer* et distance de *Hausdorff*). Nous formulons le problème de la recommandation des services pour une communauté comme suit:

- On dénote une communauté $C^k \in C$ possédant un ensemble d'intérêts I^k .
- On dénote un service $S^h \in S$ possédant un ensemble d'annotations A^h .

Où I_i^k et A_j^h sont respectivement des éléments des ensembles I^k et A^h tel que $I_i^k \in I^k$ et $A_j^h \in A^h$. Ce sont des concepts C de l'ontologie du domaine Ω tel que I^k et $A^h \in C(\Omega)$.

Nous utilisons la distance de Wu et Palmer ($1 - wup(I_i^k, A_j^h)$) pour calculer la distance sémantique entre chaque pair de concepts (I_i^k, A_j^h) et la distance de Hausdorff pour calculer la distance entre les deux ensembles de concepts I^k et A^h qui représentent respectivement les intérêts de la communauté et les annotations du service.

Selon l'algorithme présenté dans la Table 23, les intérêts de la communauté et les annotations des services sont projetés sur l'ontologie du domaine Ω pour effectuer les calculs. Ensuite, on parcourt l'ensemble des services proposés. Pour chaque service, on calcule la distance de Hausdorff entre l'ensemble de ces annotations et l'ensemble des intérêts de la communauté. Les services retenus sont les services qui possèdent une distance inférieure à un seuil prédéfini. La valeur du seuil est fixée de manière empirique et comprise dans l'intervalle [0..1].

ALGORITHME : Recommandation de service

Debut

1. Projeter les intérêts de la communauté et les annotations des services sur l'ontologie du domaine
2. **Parcourir** les profils des services
3. Calculer DH (EnsembleInteretCommunaute, EnsembleAnnotationService)
4. Recommander services (DH (EnsembleInteretCommunaute, EnsembleAnnotationService) < Seuil)
5. **Fin parcourir**

Fin

Table 23: Algorithme de recommandation de service.

Afin d'expliquer notre approche de *matching*, nous présentons un exemple dans la Figure 42 où nous nous référons à l'ontologie de la Figure 42 (a).

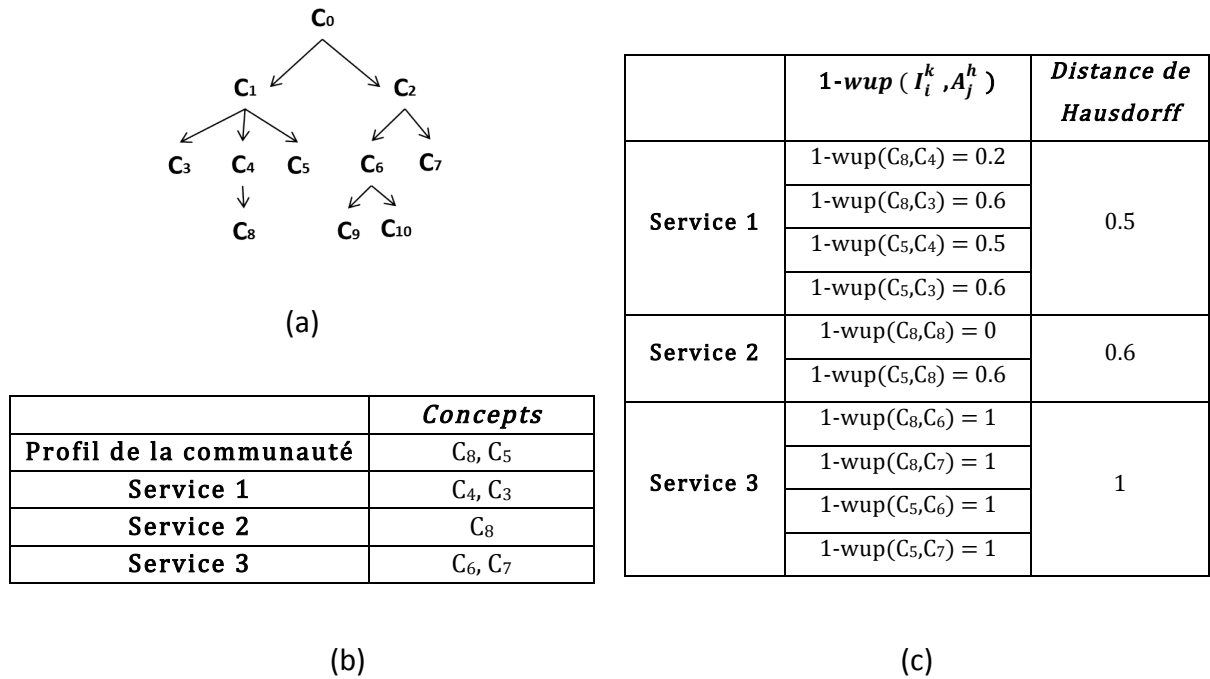


Figure 42: Exemple de calcul de la distance Hausdorff entre profil de communauté et description de service.

Dans la Figure 42 (b), nous présentons une communauté qui possède deux concepts comme centres d'intérêt (C_8, C_5), et trois services avec des annotations différentes sous forme de concepts. Comme le montre la Figure 42 (c), nous avons calculé la distance de Hausdorff entre chaque paire (profil de communauté et description de service). Pour le premier service, nous calculons la distance sémantique entre le premier concept du profil de la communauté et les annotations du service ($1-wup(C_8, C_4)$) et ($1-wup(C_8, C_3)$) et nous retenons le minimum. De même, nous retenons le minimum entre les distances *wup* pour les paires (C_5, C_4) et (C_5, C_3). Pour le résultat final, nous retenons le maximum entre les deux valeurs de minimum obtenues (cf. **Équation 4**). Le même calcul est ensuite effectué pour le reste des services. Si on classe les services par valeurs de distances : S1 est le plus proche, ensuite S2 (même s'il possède une annotation identique au concept du profil de la communauté mais qui représente sa seule annotation) et enfin S3.

6.6.2 La recherche

La première approche que nous avons choisie pour la recherche de service se base sur l’approche de base du domaine de *matching* de services : l’algorithme de Paolucci et al. (Paolucci et al. 2002). Le mécanisme de *matching* se base sur les paramètres du profil du service (*input*, *output*). L’idée est de comparer les paramètres de l’annonce de service avec les paramètres de la requête et de calculer le degré de correspondance. Afin d’expliquer les différents degrés de correspondance, nous allons nous référer à la **Figure 43** (a). On considère C_A le paramètre du service annoncé et C_Q le paramètre du service souhaité, c’est-à-dire la requête. La notion d’équivalence ‘**Exact**’ entre concepts représentant les paramètres correspond au fait que le concept C_A est équivalent et de même type que le concept C_Q , c’est-à-dire $C_A \equiv C_Q$. ‘**Plug-in**’ correspond au fait que l’annonce renvoie un paramètre qui englobe le paramètre du service requis $C_A \supseteq C_Q$. ‘**Subsume**’ correspond au fait que le concept du service annoncé est moins général que le concept du service requis $C_A \subseteq C_Q$. ‘**Fail**’ : correspond au fait qu’il n’y a aucune correspondance $C_A \neq C_Q$. Des valeurs peuvent ainsi être attribuées aux paramètres (cf. **Figure 43** (a)) selon leur degré de correspondance (*Exact*, *Plug-in*, *Subsume*, *Fail*). La Figure 42 (a) représente un fragment de notre ontologie de référence sur laquelle se base le calcul. Dans la Figure 43 (b), nous proposons un exemple de service requis Q et quelques exemples de services proposés. Pour le premier service **S1**, il possède un *plug-in* ($C_4 \supseteq C_8$), un *exact* ($C_9 \equiv C_9$) et un *fail* ($C_7 \neq C_4$), donc sa note est la somme des valeurs d’un *plug-in*, d’un *exact* et d’un *fail* ($2/3 + 1 + 0 = 5/3$). Pour **S2**, il possède un *plug-in* ($C_1 \supseteq C_8$), un *fail* ($C_5 \neq C_9$) et un *subsume* ($C_8 \subseteq C_4$). Et finalement **S3**, deux *fail* ($C_3 \neq C_8$) ($C_4 \neq C_7$) et un *plug-in* ($C_6 \supseteq C_9$).

	Description logique	Signification	Valeur
Sim (C_A, C_Q)	$C_A \equiv C_Q$	Exact	1
	$C_A \supseteq C_Q$	Plug-in	2/3
	$C_A \subseteq C_Q$	Subsume	1/3
	$C_A \neq C_Q$	Fail	0

(a)

	Inputs	Outputs
Q	C_8	C_9, C_4
S1	C_4	C_9, C_7
S2	C_1	C_5, C_8
S3	C_3	C_6, C_7

(b)

Figure 43: Algorithme de *matching* du service annoncé et du service demandé selon l’approche Dom (Paolucci et al. 2002).

Nous avons constaté que les utilisateurs finaux ont des difficultés à formuler la requête du service requis (Input, Output, *etc.*) et que l'approche de Paolucci et al. et les approches dérivées sont destinées à des utilisateurs plus avancés (exprimer des requêtes de services implique connaître les composants de l'ontologie de service à savoir le *service profile* ou le *process model* ou le *grounding* et respecter la syntaxe du langage utilisé). Nous avons donc mesuré la nécessité de proposer une approche conviviale et facile pour chercher des services par un utilisateur final. L'hypothèse de base est de permettre à l'utilisateur d'effectuer une recherche en langage naturel.

Nous notons que la description des services proposés en *Owl-S* comporte une partie exprimée en langue naturelle « *profile textual description* ». Ce composant du profil du service est utilisé lors de la recherche. L'approche proposée est composée des étapes suivantes :

- Interroger les services annoncés (*i.e.* la description des services proposés) par l'utilisateur à travers une requête en langage naturelle.
- *Matching* de la requête de l'utilisateur et de la description des services annoncés.
- Présentation du résultat du *matching* à l'utilisateur.
- Choix de l'utilisateur.
- Vérification des règles autorisant une communauté à accéder à un service.
- Déploiement adaptatif du service choisi.

Le processus du *matching* est relativement similaire à celui de la recherche de communauté (*i.e.* *matching* entre requête de l'utilisateur et les descriptions des communautés existantes) et utilise les mêmes techniques (*i.e.* annotation, génération de requête *Sparql*) et les mêmes composants de l'architecture (*i.e.* processeur d'annotation, moteur de requête). Le *matching* entre requête et description de service est réalisé comme suit:

- Extraction des concepts de l'ontologie du domaine de la requête initiale en recourant au processeur d'annotation (*cf.* Figure 44) ;

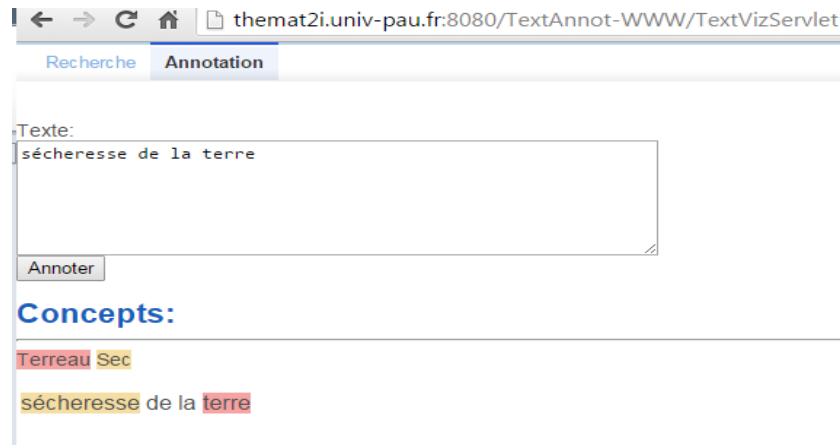


Figure 44: Exemple d’annotation d’une requête en langage naturelle pour la recherche de service.

- Génération de requête *Sparql* par le moteur de requête (cf. Table 24);

```
PREFIX onto :< http://www.communities.org/ontologies/communities.owl#>
Select ?service
Where {
  ?service rdf:type onto:Services.
  ?service onto:presents ?profil.
  ?profil onto :textDescription ?annotation.
Filter (
  ?annotation = "Terreau" &&
  ?annotation = "Sec"
  )
}
```

Table 24: Exemple de requête *Sparql* générée pour la recherche de service

- Interrogation de la *A-box* de l’ontologie des services (cf. Figure 45).

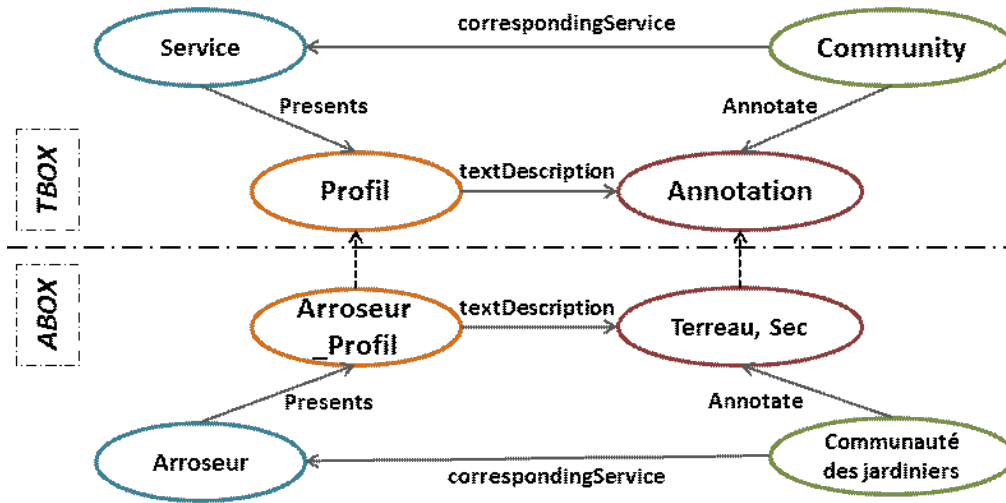


Figure 45: Exemple de description d'un service.

6.6.3 La création

De même que pour la création de communauté, l'application communautaire assiste l'utilisateur dans la création/ajout de descriptions sémantiques à des services permettant de faciliter leur exploitation ultérieure. Pour ce faire, nous nous basons sur les annotations générées par le moteur d'annotation à partir des descriptions introduites par les membres des communautés représentant le contexte ou la situation dans laquelle le service est utilisé.

Avant toute intervention au niveau de la description de service, les paramètres de confidentialité sont vérifiés. Seules les communautés concédantes des droits nécessaires peuvent modifier la description du service. Ensuite, les annotations extraites (qui peuvent référencer plusieurs concepts d'une ontologie de domaine)instancient automatiquement le *"profile textual description"* du service afin d'améliorer l'expressivité de la description du service. L'annotation d'un service par des concepts de l'ontologie du domaine sans préciser la nature des informations sémantiques (elle ne distingue pas s'il s'agit d'une précondition, d'un effet ou un d'un résultat de l'opération) est appelée une annotation à gros grain.

La Table 25 illustre une règle en *SWRL* permettant d'annoter automatiquement une description (c'est-à-dire le profil p) d'un service s par des annotations a exprimées par une communauté donnée c .

<pre>Community(?c), Service(?s), Profil(?p), Annotation(?a), presents(?s,?p), licensor(?c,?s), annotate(?c, ?a), correspondingService(?c,?s) → textDescription(?p,?a).</pre>
--

Table 25: Règle *SWRL* pour l'annotation d'un service.

Considérons l'exemple de la Figure 46. Un membre d'une communauté peut annoter le service "*Alarme*", représenté par le profil "*Alarme_profile*" avec les annotations : "*incendie (fire)*", "*catastrophe naturelle (natural disaster)*", "*crise (crisis)*".

```
<!-- http://www.communities.org/ontology#Alarme_profile -->
<owl:NamedIndividual rdf:about="http://www.communities.org/ontology#Alarme_profile">
  <rdf:type rdf:resource="http://www.daml.org/services/owl-s/1.2/Profile.owl#Profile"/>
  <Profile:contactInformation rdf:resource="http://www.communities.org/ontology#gardeners"/>
  <Profile:textDescription rdf:resource="http://www.owl-ontologies.com/OntologieBotanique10.owl#fire"/>
  <Profile:textDescription rdf:resource="http://www.owl-ontologies.com/OntologieBotanique10.owl#natural_disaster"/>
  <Profile:textDescription rdf:resource="http://www.owl-ontologies.com/OntologieBotanique10.owl#crisis"/>
  <Service:presentedBy rdf:resource="http://www.communities.org/ontology#Alarme"/>
</owl:NamedIndividual>

<!-- http://www.communities.org/ontology#Alarme -->
<owl:NamedIndividual rdf:about="http://www.communities.org/ontology#Alarme">
  <rdf:type rdf:resource="http://www.daml.org/services/owl-s/1.2/Service.owl#Service"/>
  <Service:presents rdf:resource="http://www.communities.org/ontology#Alarme_profile"/>
</owl:NamedIndividual>
```

Figure 46: Extrait de la description en *Owl-S* du service "*Alarme*".

6.7 Conclusion

Dans ce chapitre, nous avons tout d'abord introduit l'application communautaire *Taldea*, ses exigences, ses objectifs, ses fonctionnalités et le scénario potentiel. Nous avons montré comment cette application promeut la notion de *communauté spontanée* et facilite la découverte de services dans un environnement ubiquitaire. Nous avons ensuite présenté l'architecture générale de l'application avec ses différents composants : les *core* services, les services sensibles au contexte, le processeur d'annotation, le moteur de requête, le calculateur de proximité sémantique, le moteur des règles et le contrôleur de confidentialité. Puis, nous avons détaillé le fonctionnement des deux modules principaux de *Taldea*: le module d'accès

aux communautés et le module de découverte des services. Pour chaque module, nous avons exposé les différents composants et techniques utilisés. Chacun de ces deux modules comprend trois services de base : la recommandation, la recherche et la création (R, S, C). Le premier service R , permet de recommander les *items* : communauté ou service sans intervention explicite de l'entité sociale cible (utilisateur ou communauté) tout en rapprochant le profil de l'item et le profil de l'entité sociale. Le deuxième service S permet aux entités sociales une recherche en langage naturel des *items* afin d'éviter la complexité des langages formels. Le dernier service C , offre la possibilité aux utilisateurs de créer des communautés spontanées avec des descriptions sémantiques, ainsi que de créer des descriptions sémantiques pour les services accessibles dans leur environnement ubiquitaire. Pour la réalisation des applications communautaires ubiquitaires, le travail de déploiement dynamique et d'adaptation a été confié à la plateforme *Kalimucho* dont le rôle est la gestion du contexte d'exécution de l'application (acquisition, modélisation, raisonnement, prise de décision d'adaptation), et la mise en œuvre des adaptations par déploiement ou reconfiguration des services de l'application.

Dans le chapitre suivant, nous présentons la plateforme *Kalimucho*, le prototype Taldea et les évaluations des résultats à travers les critères utilisés dans le domaine de la recherche d'information.

Chapitre 7: Réalisation: Prototype et Evaluation

Notre défi est de proposer des applications communautaires qui s'adaptent à l'utilisateur en fonction de son contexte social (*i.e.* communautés), ainsi que de son environnement physique.

La prise en considération de l'environnement physique permet de proposer à l'utilisateur des services mieux adaptés aux ressources proposées et aux contraintes rencontrées. Pour que nos applications puissent être adaptées à l'environnement physique, nous avons besoin d'un intergiciel sensible au contexte (*cf.* section 4.2) dédié au déploiement et à l'adaptation des applications. Dans ce travail, nous avons utilisé la plateforme logicielle *Kalimucho* développée au sein de notre équipe de recherche.

7.1 Préambule : la plateforme de supervision *Kalimucho*

Kalimucho est une plateforme logicielle hébergée sur tout type de dispositif physique (*PC, Smartphone, Tablette*). Elle peut surveiller l'utilisation des ressources matérielles (*batterie, mémoire, CPU*) et le contexte d'exécution (réseau), afin de mettre en place une politique d'adaptation basée sur la reconfiguration de l'application répartie.

Taldea est composée de services interconnectés supervisés par la plateforme *Kalimucho*. Un service est matérialisé par un ensemble de composants logiciels reliés par des connecteurs. Ces services peuvent être réalisés de différentes façons à partir d'assemblages différents de composants, chacun correspondant à une qualité de service différente (Louberry et al. 2011).

Afin de mieux illustrer l'utilité de la plateforme pour nos applications communautaires, reprenons l'exemple proposé en section 6.1: le parc propose un service de description des plantes sous différents médias que les utilisateurs peuvent visualiser sur leur *Smartphone*. Plusieurs configurations possibles du service sont assurées par la plateforme *Kalimucho* (affichage d'une vidéo, affichage d'une séquence d'images, *etc.*). La meilleure qualité de service proposée (*QoS*) est de fournir une vidéo en couleur. Mais, en raison d'un

faible débit, ce service peut être reconfiguré afin d'assurer la continuité de service en, par exemple, réduisant la résolution de la vidéo, ou en proposant un affichage en noir et blanc, ou l'affichage d'une séquence d'images sous-titrées, *etc.*

L'adaptation des applications nécessite la mise en place des étapes suivantes : acquisition, modélisation et interprétation du contexte, identification des besoins d'adaptation, prise de décision d'adaptation et mise en œuvre de cette décision. Toutes ces opérations sont assurées par la plateforme *Kalimucho* (Da 2014) qui est constituée de deux parties : *Kalimucho-A* (prise de décision d'adaptation automatique) et *Kalimucho kernel* (mise en œuvre de l'adaptation ; *middleware*).

Notre démarche pour proposer à l'utilisateur des applications communautaires spontanées dynamiquement reconfigurables est la suivante (illustrée dans la Figure 47) : l'architecture *Taldea* permet de découvrir les services adaptés à l'utilisateur et à son contexte social (*i.e* ses communautés) (*cf.* étape 1 de la Figure 47). *Kalimcho-A* évalue la qualité de service et définit le schéma de reconfiguration du service sélectionné (*cf.* étape 3 de la Figure 47) à partir des informations de contexte fournies par *Kalimucho Kernel* (*cf.* étape 2 de la Figure 47). Ce dernier met en œuvre les schémas de reconfiguration définis afin d'assurer un service continu en dépit des défaillances matérielles, ou réseau et de l'hétérogénéité des appareils (*cf.* étape 4 de la Figure 47).

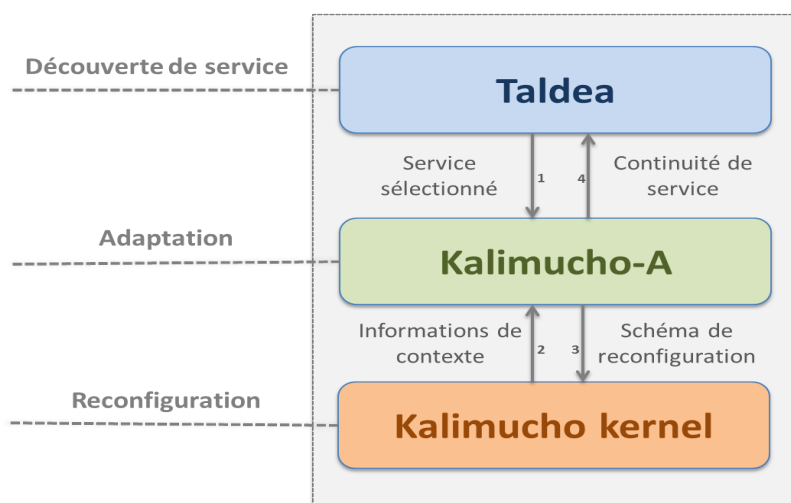


Figure 47: Architecture générale.

L'approche de découverte de service (détaillée dans la section 7.6), consiste en général à assister les membres des communautés dans la découverte des services susceptibles de les intéresser. *Taldea* assure la découverte de service, soit par recommandation en proposant aux membres d'une communauté donnée les services qui ont été utilisés par des communautés sémantiquement proches (cf. section 6.6.1), soit en les aidants à chercher des services au travers de requêtes formulées en langage naturel (cf. section 6.6.2).

Dans les sections qui suivent, nous présentons la plateforme *Kalimucho* et ses deux parties (Kalimucho-A et Kalimucho kernel) pour le déploiement et l'adaptation des services sélectionnés.

7.1.1 *Kalimucho-A*

La couche *Kalimucho-A* (*Kalimucho-Adaptation Platform*) hébergée sur chaque dispositif physique (PC, Smartphone, tablette, etc.) surveille l'utilisation des ressources matérielles (batterie, mémoire, CPU, etc.) et le contexte d'exécution (réseau, besoins des utilisateurs, règles d'usage de l'application, etc.), afin de mettre en place une politique d'adaptation basée sur la modification de l'architecture de l'application (son déploiement).

La plateforme *Kalimucho-A* permet de prendre des décisions de reconfiguration dynamique en toute connaissance de cause. Elle peut ainsi assurer la continuité de service tout en prenant en compte la pérennité globale de l'application. Une telle solution permet au concepteur de définir un ensemble de fonctionnalités dont certaines sont substituables, et à l'utilisateur de disposer de ces fonctionnalités selon ses besoins. Il est donc nécessaire de capturer tous les changements de contexte qu'ils soient liés aux besoins des utilisateurs, aux ressources ou à la mobilité, puis d'interpréter ces changements afin de réagir de la meilleure façon.

L'architecture de la plateforme *Kalimucho-A* contient plusieurs parties indépendantes qui exploitent des connaissances sémantiques (stockées dans des ontologies). Kali2Much est le *middleware* de gestion de contexte. Il contient quatre types d'entités : "*KaliSensor*", "*Context Collector*", "*Context Transformer*" et "*Context Searching Service*". Les données contextuelles brutes sont collectées par des "*KaliSensors*" (cf. étape 1 de la Figure 48). Selon

les besoins des consommateurs de ces données, la plateforme peut configurer les "*Context Collectors*" pour leur permettre d'accéder à ces informations par consultation ou par notification (cf. étape 2 de la Figure 48). Des "*Context Transformers*" peuvent être mis en place pour modifier les données avant qu'elles ne soient délivrées aux consommateurs (cf. étape 3 de la Figure 48). Ces dispositifs de collecte d'informations contextuelles peuvent être déployés sur chaque hôte de l'utilisateur. Le service "*Context Searching Service*" permet aux consommateurs de rechercher des sources d'informations contextuelles de bas niveau (*raw data*) à partir de descriptions sémantiques (cf. étape 4 de la Figure 48).

La plateforme *Kalimucho-A* utilise donc "*Kali2Much*" pour collecter les informations de contexte de bas niveau qu'elle transmet au "*Kali-Reasoning Service*" pour les interpréter et produire des informations de contexte de plus haut niveau sémantique (contexte sémantique) (cf. étape 5 de la Figure 48). Ces informations de contexte de haut niveau produisent par la suite des situations d'adaptation.

Toutes ces informations sont stockées dans des modèles sémantiques basés sur des ontologies à savoir un modèle de contexte (*KaliCOnTo*), un modèle de situation (*KaliMOSA*), une ontologie pour la recherche de contexte (cf. étape 6 de la Figure 48) et une ontologie de solution pour la recherche de solutions d'adaptation (cf. étape 7 de la Figure 48).

Dès qu'une situation d'adaptation est identifiée, le service "*Kali-Adapt*" reçoit une notification qui déclenche la recherche d'une solution applicable dans le contexte actuel (cf. étape 8 de la Figure 48). Pour finir, la reconfiguration est réalisée par l'envoi de commandes aux différentes plateformes "*Kalimucho kernel*" hébergées sur les hôtes concernés (cf. étape 9 de la Figure 48), provoquant ainsi les modifications de l'architecture de l'application.

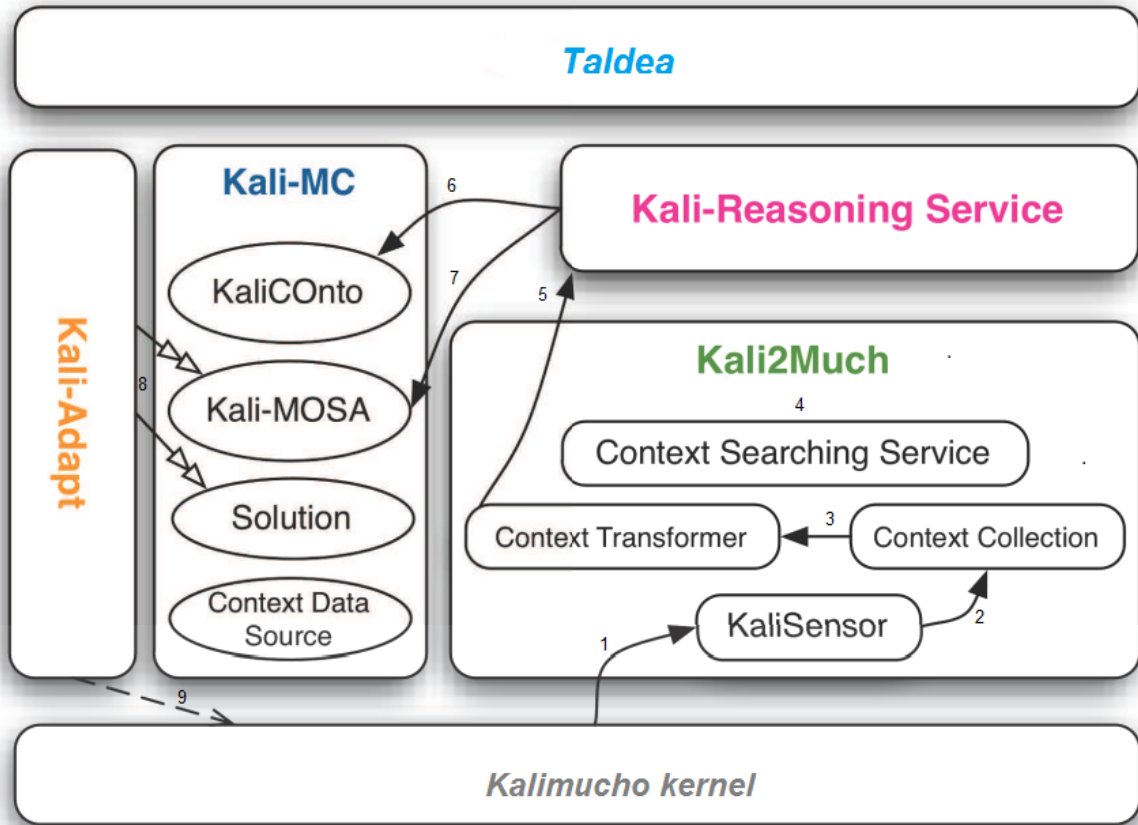


Figure 48: Architecture de la plateforme : Kalimucho-A (Da 2014).

Tous les détails concernant l'architecture et le fonctionnement de *Kalimucho-A* sont présentés dans la thèse de Keling Da (Da 2014).

7.1.2 *Kalimucho kernel*

Kalimucho-A s'appuie sur la plateforme *Kalimucho kernel* qu'elle étend. *Kalimucho-A* utilise *Kalimucho kernel* comme support de gestion de composants logiciels dynamiquement supervisés (déploiement, suppression, migration à chaud) et comme *middleware* de communication.

Les déploiement et redéploiement dynamique de *Taldea* nécessitent qu'elle ait une architecture supervisable. Notre application est donc composée de plusieurs services et chaque service peut être réalisé par un ou plusieurs assemblages de composants reliés par des

connecteurs. L'état d'une application est alors constitué de l'ensemble des états des composants, des périphériques, des connecteurs et de l'environnement. La plate-forme doit recueillir toutes ces données afin de les traiter et de mener les actions de reconfiguration qui conviennent. Adapter l'application consiste alors à ajouter, supprimer ou modifier des services. Nous disposons ainsi de trois moyens d'action permettant de modifier la structure d'une application :

- *La migration de service* : consiste à modifier la distribution d'un service sans en modifier les composants. Les composants sont déplacés (cf. Figure 49, composant C4), avec leurs états, d'un périphérique vers un autre, sans remplacement, ajout ni suppression.
- *Le déploiement ou redéploiement d'un service* : consiste à proposer un nouvel assemblage de composants pour réaliser ce service ou un service équivalent à travers la suppression d'un ou de plusieurs composants existants (cf. Figure 49, composant C1) ou l'ajout d'un nouveau composant (cf. Figure 49, composant C8) ainsi que des déconnexions/reconnexions de ces composants entre eux.
- *La modification d'un service* : consiste à remplacer un ou plusieurs de ses composants (cf. Figure 49, composant C2) par d'autres (cf. Figure 49, composant C7).

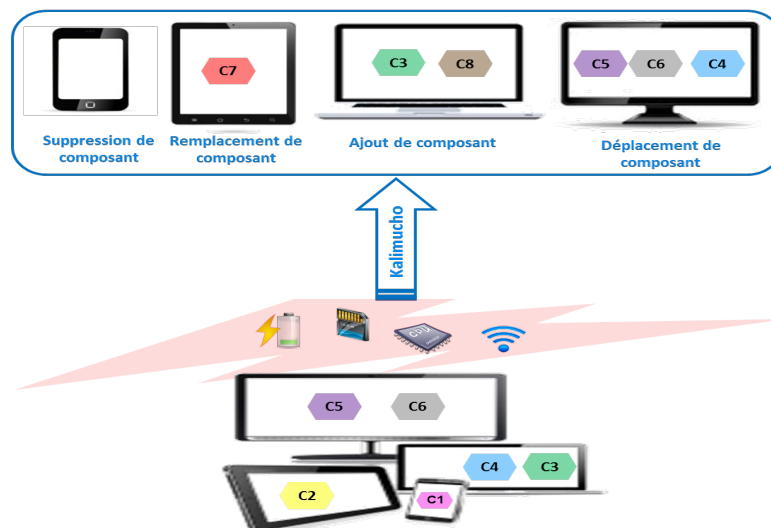


Figure 49: Illustration des actions de reconfigurations assurées par la plateforme *Kalimucho*.

Kalimucho implémente un modèle de composants appelé *Osagaia*, possédant un cycle de vie supervisable par la plateforme et permettant la migration à chaud. La plateforme *Kalimucho* établit les connexions entre composants via des connecteurs de première classe appelés *Korrontea* implémentant une partie métier. Ces derniers permettent le traitement à la volée des données transmises.

Kalimucho a connaissance des composants et connecteurs déployés et peut récupérer les informations de contexte que ceux-ci lui transmettent. C'est en fonction de ces informations que les décisions de reconfiguration pourront être prises. *Kalimucho-A* utilise *Kalimucho Kernel* pour collecter le contexte lié aux états des composants et des connecteurs, pour gérer les communications par réseau ainsi que pour déployer, supprimer et migrer des composants au *runtime*.

Afin de recueillir des informations sur le fonctionnement des composants et la circulation des flux de données entre ces composants, la plateforme les encapsule dans des conteneurs qui, de plus, apportent une solution à la gestion de l'hétérogénéité matérielle et logicielle des périphériques ainsi qu'à la mobilité des périphériques.

7.1.2.1 OSAGAIA : conteneur de composants métier

Le conteneur *Osagaia* possède trois types d'unités fonctionnelles : l'Unité d'entrée (*UE*), l'Unité de sortie (*US*) et l'Unité de contrôle (*UC*). Les unités d'entrée et de sortie peuvent être connectées à un ou plusieurs connecteurs simultanément. Elles permettent au composant métier (*CM*) qui contient la logique métier de l'application de lire et écrire des données provenant ou à destination d'autres composants métier. Le composant métier peut ainsi lire des données via les *UE*, effectuer son traitement et écrire les résultats dans les *US*. L'Unité de contrôle (*UC*) permet à la plate-forme de superviser le conteneur.

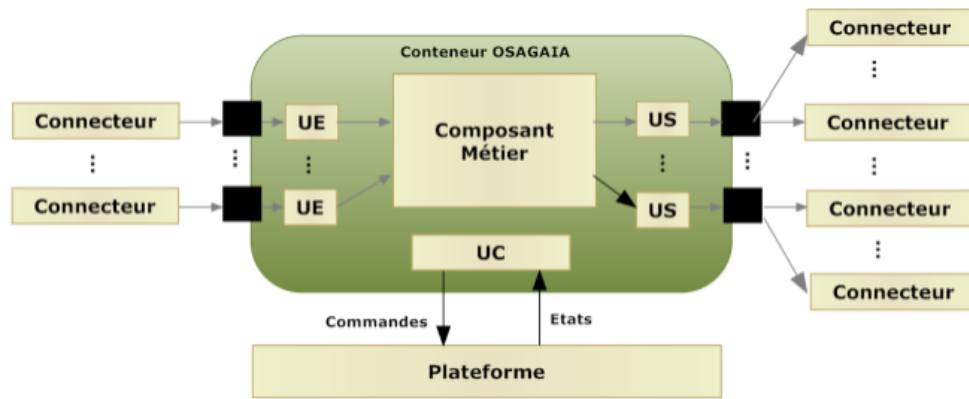


Figure 50: Modèle de conteneur de composant métier OSAGAIA.

Le concepteur de l'application écrit les composants métiers (CM) qui seront encapsulés par la plateforme dans un conteneur *Osagaia* qui en contrôlera le cycle de vie. Le principe retenu est celui que l'on trouve pour les *applets*, les *midlets* ou les activités sur Android : ce cycle de vie correspond à l'appel de méthodes que le développeur doit surcharger.

7.1.2.2 KORRONTEA : conteneur de connecteurs

Pour relier les E/S des composants, le mécanisme de connecteur est généralement admis dans le domaine des architectures logicielles à base de composants. *Korrontea* fournit un conteneur pour les connecteurs. La fonctionnalité principale d'un connecteur est de relier deux composants et de faire circuler l'information entre eux. De la même manière qu'un composant *Osagaia*, un connecteur est une entité de première classe. Il ne se limite pas à la mise en œuvre d'un ou de plusieurs modes de communication (Client/Serveur, Pipe & Filter, etc.). Nous souhaitons également qu'il puisse agir sur l'information elle-même de façon à réaliser l'adaptation de données à la volée. Il est donc nécessaire de le doter, lui aussi, d'un composant métier.

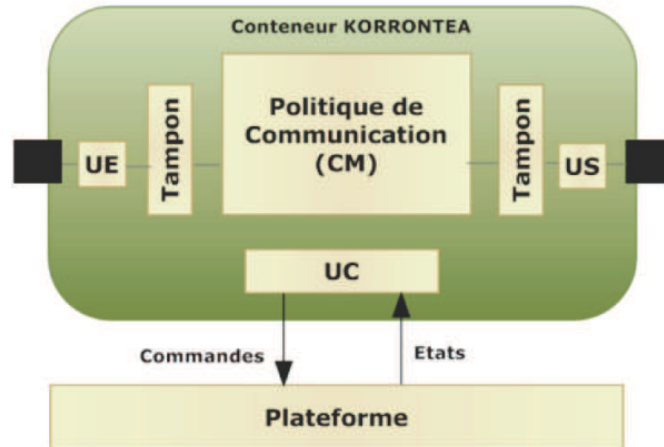


Figure 51: Modèle de Connecteur *Korrontea*.

Le connecteur *Korrontea* informe la plateforme de la circulation des données dans l'application. Il lève des alarmes lorsque des données s'accumulent dans ses tampons, mais également lorsque la circulation des données devient fluide après une accumulation. Ceci permet à la plateforme de surveiller la circulation de données dans un hôte ou entre deux hôtes sur le réseau. Les connecteurs fonctionnent en mode synchrone : pour chaque donnée envoyée un acquittement est renvoyé. Aucune nouvelle donnée ne peut être envoyée avant que l'acquittement a été reçu. Ce mécanisme est nécessaire parce que la plateforme *Kalimucho* doit contrôler et mesurer la circulation des données. Ce qui signifie qu'aucune donnée ne peut être accumulée hors de son *middleware* (par exemple dans les *buffers* des *sockets*). Un connecteur peut être utilisé pour relier deux composants sur la même machine (on parle alors de connecteur interne) ou deux composants placés sur des machines différentes (on parle alors de connecteur distribué). En raison de l'hétérogénéité des réseaux (*wifi*, *bluetooth*, *3G*, *etc.*), il est possible que deux machines devant être reliées par un connecteur ne puissent pas communiquer directement. Dans ce cas, la plateforme se charge de trouver un ou plusieurs hôtes pouvant servir de passerelle entre ces deux machines.

7.2 Réalisation

Taldea est une architecture dédiée aux applications communautaires dynamiquement reconfigurables. Elle permet d'aider les utilisateurs nomades à intégrer des communautés

environnantes et à découvrir les services dédiés à l'échange et le partage des intérêts, des sujets, des activités, des événements, *etc.* Ce qui différencie Taldea des autres applications communautaires est qu'elle intègre des aspects innovants comme les communautés spontanées, le déploiement de services basés sur la localisation, les technologies du web sémantique, l'adaptation et le changement du comportement de l'application en fonction du contexte.

Dans cette section, nous présentons en premier lieu nos choix technologiques, ensuite un scénario d'illustration, et finalement le fonctionnement du prototype *Taldea* dans le cadre du scénario retenu.

7.2.1 Choix technologique

Un utilisateur mobile change souvent de besoins et de contexte et souhaite disposer d'applications capables de réagir à ses exigences. A ce titre, nous proposons des applications communautaires adaptatives. Dans la littérature, plusieurs plateformes (comme *MUSIC* (Floch et al. 2013), *CAMPUS* (Wei et al. 2013), *COSMOS* (Conan et al. 2007) permettent de créer des applications dynamiques. Nous avons choisi la plateforme *Kalimucho* car cette technologie nous permet d'assurer également la continuité de service de la plateforme elle-même, grâce aux fonctions de migration/déplacement de composants à chaud.

Dans un environnement mouvant, les services de la plateforme peuvent eux-mêmes être perdus ou ne plus pouvoir fonctionner correctement. Contrairement aux plateformes citées précédemment, les services de la plateforme *Kalimucho* peuvent, eux-mêmes, faire l'objet de reconfigurations contextuelles.

Taldea a été développée sur le système d'exploitation *Android*. En 2015, *Android* est le système d'exploitation le plus utilisé dans le monde avec plus de 80% de parts de marché. *Android* couvre une bonne partie des terminaux mobiles, *Smartphones* et tablettes compris. Dans nos expériences, nous avons utilisé plusieurs modèles de téléphones et tablettes. Les terminaux datent de 2012 à 2015, avec du moins au plus puissant : un *Samsung Galaxy S2* et un *Samsung Galaxy S4*. Pour l'illustration qui suit, nous utilisons le *Samsung Galaxy S4*. *Kalimucho* permet à un périphérique *Android* d'exécuter des applications pervasives à base de

composants. Ces derniers sont présents sur un dépôt de composants existant sur un périphérique fixe non contraint en ressources. La complexité et le nombre élevé de calculs et de communications demandés par les services qui génèrent l'adaptation causait un épuisement énergétique prématuré si le dépôt de composant était déployé sur un périphérique contraint. C'est pourquoi il est, de préférence, déployé sur des périphériques fixes. Dans notre cas, nous avons utilisé le serveur *Okabe* de l'*IUT* de Bayonne. Les caractéristiques du *Smartphone* et du serveur sont détaillées dans la Table 26.

	Samsung Galaxy S4	Serveur Okabe
Système d'exploitation	Android 4.2 Jelly Bean	Linux kernel 2.6.32
Processeur	Qualcomm Snadragon 600 - 1.9 GHz	Intel ® Xeon ® E5504 (4M Cache, 2.00 GHz)
Mémoire interne	2048 Mo	4040 Mo

Table 26: Fiche technique du matériel de test.

7.2.2 Scénario d'illustration : parc botanique

Le scénario que nous utilisons pour illustrer le fonctionnement se déroule dans le parc botanique *Mosaïc* de Lille (dans le cadre du projet *ANR MOANO*) où se côtoient une grande variété d'intervenants (visiteurs, jardiniers, éco-gardes, intervenants externes, *etc.*). Ces intervenants sont ou peuvent être équipés de dispositifs mobiles (*ordinateur portable, Smartphone, etc.*) hétérogènes (taille d'écran, mode d'interaction, mémoire, batterie, interface réseaux, *etc.*).



Figure 52: Descriptif du parc *Mosaïc*.

Envisageons le scénario suivant : Myriam visite le parc *Mosaïc* pour découvrir la flore et la faune locale de la région. Cherchant à établir des interactions sociales avec les intervenants du parc, elle télécharge et se connecte à l'application *Taldea*. Cette dernière l'aide à trouver des communautés en mettant en jeu plusieurs critères comme le temps, la position géographique et les intérêts de l'utilisatrice qu'elle est. Trois possibilités permettent à Myriam de rejoindre des communautés : en consultant la liste des communautés recommandées, en réalisant une recherche selon le thème qu'elle désire, ou encore en créant une nouvelle communauté.

Comme la visiteuse du parc est une passionnée de fleurs, elle se connecte à l'une des communautés recommandées « *FleursDeMosaïc* ». Elle consulte les multiples échanges partagés sur l'espace virtuel de la communauté. Toutefois, perdue dans le nombre d'informations partagées, elle décide de chercher une communauté qui s'intéresse à la fleur vedette du parc « l'orchidée ». Les résultats de cette recherche l'ont conduite à se connecter à la communauté « *Amateurs d'orchidées* ».

Ce jour correspond à *la journée nationale de l'arbre*, la visiteuse décide de créer une nouvelle communauté qu'elle nomme « *Jour de l'arbre* » pour inviter les personnes à planter ou entretenir des arbres. Peu de temps après, elle a reçu plusieurs retours. Plusieurs personnes intéressées se sont déplacées sur les lieux pour proposer leur aide et célébrer cette fête. Pour sa nouvelle communauté, des services qui ont été utilisés par des communautés sémantiquement proches lui ont été recommandés comme le service « *Join Me* » qu'ont utilisé les autres membres pour la rejoindre.

Si un jardinier, qui est à son tour un membre de la communauté « *Jour de l'arbre* », souhaite aider cette communauté en utilisant le service d'arrosage (qui permet de déclencher l'arrosage), il peut introduire la requête suivante en langage naturel: « *arrosage arbuste* ». De plus, s'il souhaite enrichir la description textuelle de ce service afin de faciliter les recherches futures et les déploiements ultérieurs, il peut l'annoter par : « *plantation d'un nouvel arbuste* ».

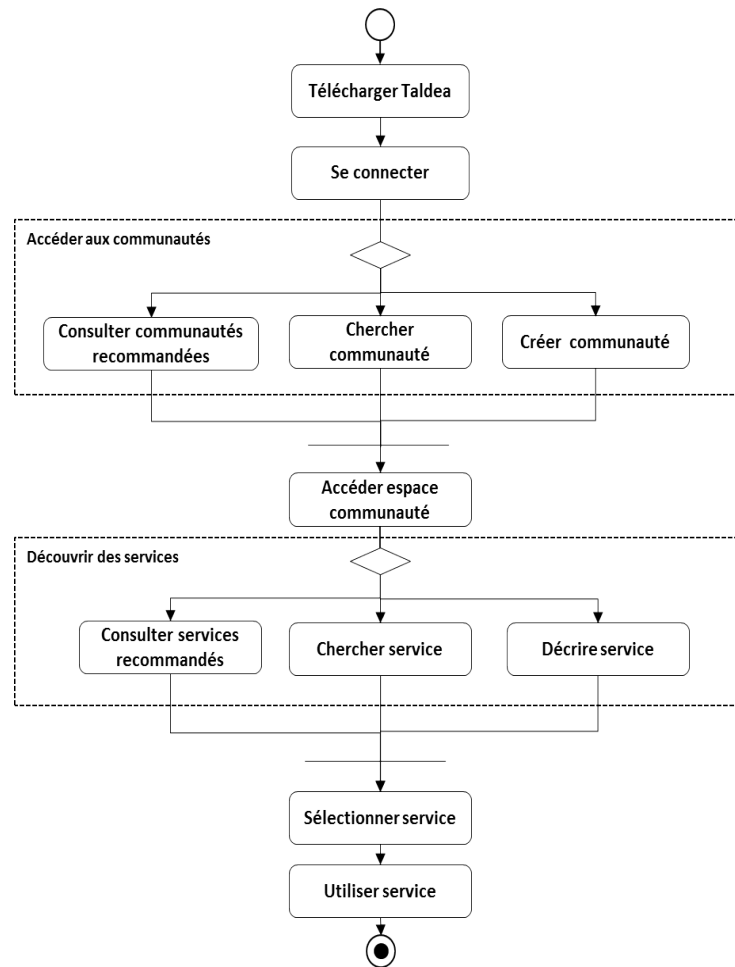
Supposons que le périphérique du jardinier ne possède plus suffisamment de batterie, ce sont tous les services qu'il offre qui cessent de fonctionner impliquant une rupture de service. Ici, la plateforme *Kalimicho* intervient pour proposer des solutions telles que la délocalisation de services en cours d'exécution afin d'anticiper les déconnexions dues au déchargement complet de la batterie.

7.2.3 *Taldea : le prototype*

Nous avons présenté dans les sections précédentes un scénario d'application et les technologies utilisées pour la mise en œuvre de notre prototype. Dans la suite, nous nous penchons sur le prototype lui-même.

Comme nous l'avons expliqué dans la section 6.3, l'architecture *Taldea* contient deux types d'entités structurelles: Le **Manager de communautés** et l'**environnement de l'utilisateur**. Ce découpage est effectué par rapport aux fonctionnalités. Le Manager de communautés est le coordinateur, qui interagit avec les bases de connaissances. L'environnement de l'utilisateur représente l'ensemble des services qui permettent à l'utilisateur de découvrir et d'interagir avec les communautés. Deux types de services peuvent être déployés sur le dispositif de l'utilisateur: les *core services* (e.g. service d'authentification, service de gestion de profil, services d'accès aux communautés, services de découverte de services, etc.) et les services sensibles aux communautés ou services *ad hoc* (e.g. *chat*, *Take_picture*, *who_is_around*, etc.).

Dans ce qui suit, nous décrivons la réalisation de l'application *Taldea* du point de vue de l'utilisateur. Étant donné que les deux points de vue (le *Manager* de communautés et l'environnement de l'utilisateur) sont complémentaires, nous exposons quelques captures d'écran de *l'Environnement utilisateur* (le côté apparent du prototype pour l'utilisateur) et nous revenons sur ses interactions avec le *Manager de communautés* (le coordinateur qui oeuvre en coulisses). Pour ce faire, nous reprenons le scénario présenté ci-dessus comme exemple d'utilisation de notre prototype et nous rappelons la Figure 27 (présentée dans le chapitre 6) qui illustre le fil conducteur de l'utilisation de *Taldea*.



7.2.3.1 Core Services

Dans cette section, nous présentons l'ensemble des *core* services, c'est-à-dire les services de base pour une utilisation minimale de l'application. Ils sont les mêmes pour tous les utilisateurs, indépendamment de leur profil et de leur contexte.

Reprenons le scenario de Myriam dans le parc botanique. Après avoir téléchargé et exécuté l'*apk* (un fichier *Android Package*) de Taldea, elle dispose de l'interface d'accueil présentée par la Figure 53 (a) qui lui permettra de choisir l'ontologie du domaine et, par suite, de créer un compte (*cf.* Figure 53 (b)) ou de s'authentifier (*cf.* Figure 53 (c)). Dans son profil, elle a indiqué qu'elle s'intéresse aux fleurs. Elle a toujours la possibilité d'éditer et de modifier son profil (*cf.* Figure 53 (d)).

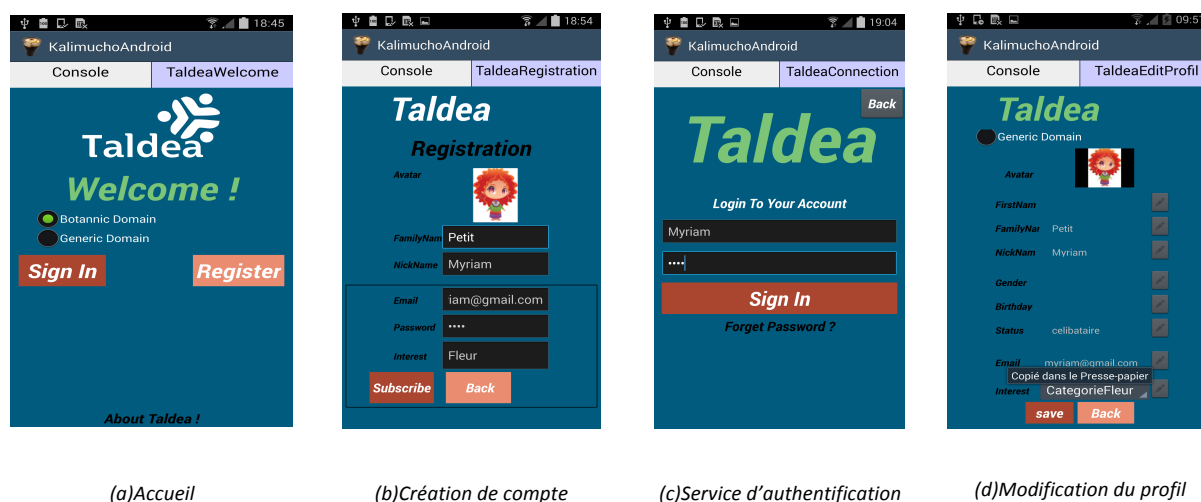
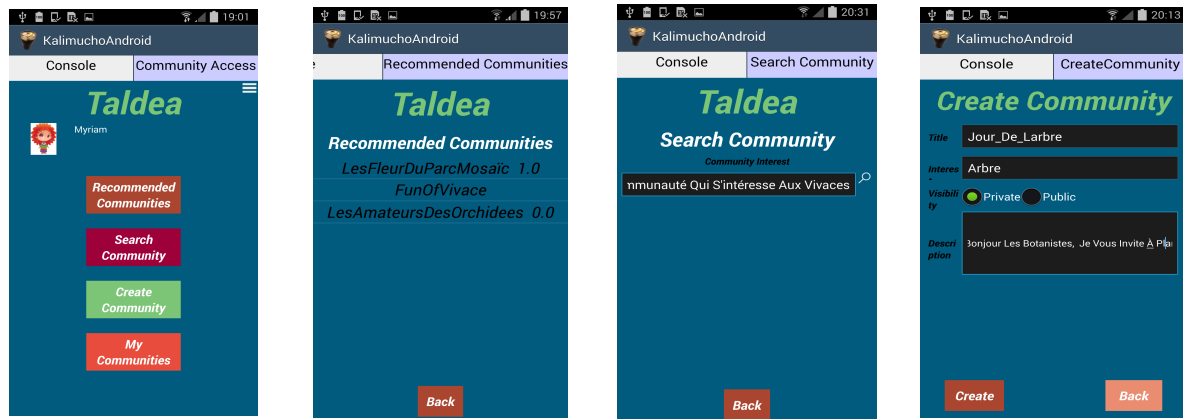


Figure 53: Connexion à Taldea.

Du côté du Manager de communautés, si l'utilisateur possède déjà un compte, son profil est récupéré dans l'ontologie d'application. Sinon, un nouveau profil est créé en utilisant le service d'annotation approprié à l'ontologie du domaine sélectionnée par l'utilisateur pour annoter sémantiquement les centres d'intérêt exprimés en langage naturel par l'utilisateur.

La Figure 54 (e) présente le module d'accès aux communautés. Trois possibilités sont offertes à l'utilisatrice pour accéder à une communauté : soit par recommandation (*cf.* Figure 54(f)), soit par recherche (*cf.* Figure 54(g)), soit en créant une nouvelle communauté (*cf.* Figure 54 (h)). En accédant aux recommandations, elle identifie les communautés susceptibles de l'intéresser, suite à un rapprochement sémantique entre son profil et les descriptions des communautés existantes. Egalement, elle peut chercher une communauté en introduisant une requête en langage naturel afin de trouver des communautés dont les centres d'intérêt correspondent à une thématique donnée. Si elle n'est pas satisfaite des résultats obtenus ou exprime de nouveaux besoins, elle peut créer une communauté spontanément.



(e) Interface accès aux communautés

(f) Service de recommandation de communauté

(g) Service de recherche de communauté

(h) Service de création de communauté

Figure 54: Le module d'accès aux communautés.

Du côté du Manager de communautés, pour la recommandation, le rapprochement entre le profil de Myriam et les profils des communautés s'effectue en pondérant la proximité sémantique et la proximité géographique. A l'aide du calculateur de proximité, la proximité sémantique entre les centres d'intérêt de l'utilisateur et ceux des communautés, ainsi que la proximité géographique entre la position de l'utilisateur et les territoires des communautés sont évalués.

Pour la recherche, nous faisons appel au processeur d'annotation pour annoter la requête de l'utilisateur. Ensuite, nous recourons au générateur de requête pour produire une requête *Sparql* en se basant sur les concepts résultats de la phase d'annotation afin d'interroger l'ontologie de communauté.

Enfin, pour la création, nous recourons au service d'annotation pour la compréhension des entrées de l'utilisateur, puis au moteur de règles pour enrichir la description de la communauté créée. Enfin, le moteur de règles est utilisé pour produire une requête permettant d'instancier l'ontologie de la communauté.



Une fois connecté à une communauté, l'utilisateur est dirigé vers l'espace virtuel commun de la communauté (*cf.* Figure 55(i)), à partir duquel il peut partager, annoter,

commenter (cf. Figure 55 (j)), consulter (cf. Figure 55 (k)) et localiser sur une carte (cf. Figure 55 (l)) les éléments multimédias partagés.



Figure 55: L'espace virtuel commun d'une communauté.

*Du côté du Manager de communautés, dans notre modèle de communauté à base d'ontologies, nous utilisons l'ontologie *Sioc*, ainsi que l'ensemble des classes associées (*i.e.* *Sioc:item*, *Sioc:space*, *Sioc:container*, etc.) qui représentent la formalisation de l'information échangée entre les membres d'une communauté. Une fois l'espace virtuel de la communauté consulté, toutes les instances de la classe *Sioc:item* sont chargées ainsi que les informations associées (par exemple *Sioc:Comment*).*

Le menu  disponible en haut de l'interface (m) permet de consulter la liste des services recommandés à la communauté concernée. Quant au menu , il propose d'accéder de nouveau à la liste des *core* services dont :

- le service qui assiste l'utilisateur dans la recherche d'un service particulier selon la demande qu'il a émise en langage naturel (cf. Figure 56 (n)).
- Le service qui lui permet d'annoter et d'enrichir la description d'autres services (cf. Figure 56 (o)).



Figure 56: Module de découverte des services.

Du côté du Manager de communautés, la découverte des services fait appel à trois services de base: la recommandation, la recherche et la création de description de service. Pour l'approche de recommandation de service, nous avons utilisé les mêmes techniques de calcul de similarité (mesure de similarité de Wu & Palmer et distance de Hausdorff) pour évaluer la proximité sémantique entre le profil de la communauté et le profil du service.

Le processus de *matching* entre la requête de l'utilisateur et la description des services est relativement similaire à celui de la recherche de communauté et utilise les mêmes techniques (*i.e.* annotation, génération de requête *Sparql*) et les mêmes composants de l'architecture (*i.e.* processeur d'annotation, moteur de requête).

Pour la création/ajout de description de services, la description du service en *Owl-S* est enrichie par les annotations générées par le moteur d'annotation, à partir des descriptions introduites par les utilisateurs.

Avant toute intervention au niveau de la description de service, les paramètres de confidentialité, définis au préalable, sont vérifiés. Ensuite, les annotations extraites (qui peuvent référencer plusieurs concepts d'une ontologie de domaine)instancient automatiquement le profil du service afin d'améliorer l'expressivité de la description du service et de faciliter les exploitations ultérieures.

7.2.3.2 Services ad hoc

Les services *ad hoc* permettent la réalisation des objectifs des communautés, supportent leurs activités et facilitent les interactions entre les membres. Dans ce travail, nous présentons deux exemples de services *ad hoc* :

- *TakePicture* pour prendre des photos avec la caméra du dispositif mobile et les partager par la suite sur l'espace de la communauté (cf. Figure 57 (p)).
- *P2Pchat* pour discuter avec un membre connecté, il suffit de double-cliquer sur le membre pour ouvrir la fenêtre de discussion (cf. Figure 57 (q)).

Quand un service est sélectionné parmi ceux proposés à l'utilisateur (à travers une recommandation ou suite à une recherche), le déploiement dynamique ainsi que la reconfiguration en cas de besoin sont assurés par la plateforme *KalimUCHO*.

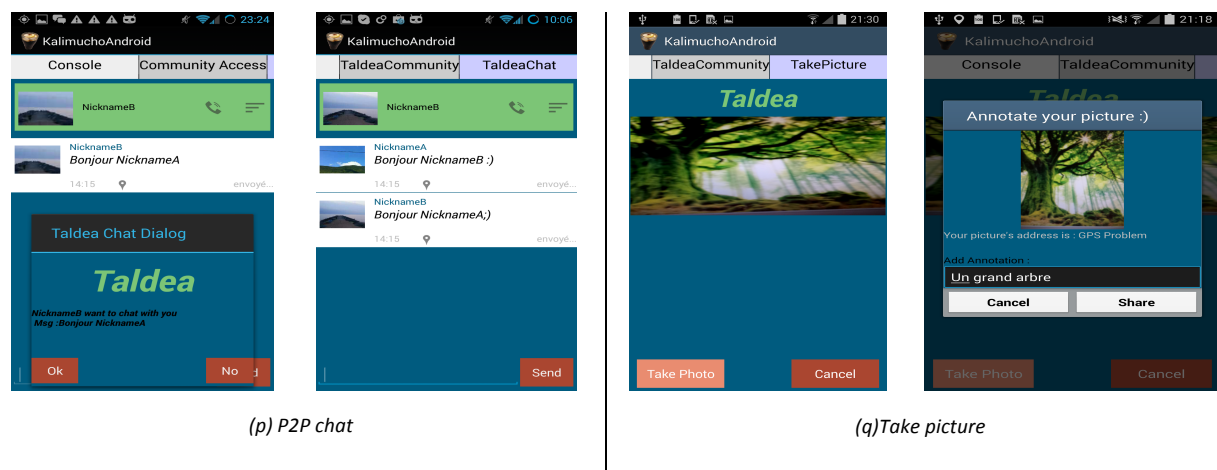


Figure 57: Exemple de services *ad hoc*.

Par la suite, nous nous intéressons à l'évaluation de notre approche de découverte de communautés et de services dédiés.

7.3 Evaluation

Notre préoccupation n'est pas d'évaluer les performances de la plateforme *Kalimucho* sur des terminaux *Android*, ni en termes de déploiement dynamique ni en termes de reconfiguration. Ce point a fait l'objet de travaux antérieurs (Louberry et al. 2011).

Dans cette section, nous abordons les expérimentations effectuées pour évaluer les mécanismes d'accès aux communautés et de découverte des services. D'abord, nous présentons les mesures d'évaluation sur lesquelles nous nous basons. Ensuite, nous décrivons nos expérimentations, avec la prise en compte d'utilisateurs réels.

7.3.1 Mesures d'évaluation

Afin d'évaluer les mécanismes de découverte de communauté et des services dédiés, nous nous sommes basées sur des mesures de performance définies dans les systèmes de recherche d'information, car nos algorithmes interrogent des éléments comme les communautés et les services dans une base de connaissances. Deux mesures sont le plus couramment utilisées : la précision et le rappel. Elles permettent de déterminer l'efficacité du système pour retrouver les items pertinents (*i.e.* communauté ou service), et ignorer les items non pertinents (*cf.* Figure 58).

Le taux de rappel désigne le rapport entre le nombre d'items pertinents trouvés (P_R) par le système sur le nombre total d'items pertinents (P) de la collection. Elle est exprimée par la formule suivante:

$$\text{Rappel} = \frac{|P_R|}{|P|}$$

Le taux de précision est défini comme le nombre d'items pertinents trouvés (P_R) sur le nombre total d'items trouvés (R). Il est exprimé par:

$$\text{Précision} = \frac{|P_R|}{|R|}$$

Ces deux mesures sont utilisées pour évaluer la pertinence de la méthode de recommandation. Quant à la recherche de communautés et de services, les requêtes *Sparql*

générées exécutées dans le cadre des informations structurées (*i.e. Owl*), permettent d'obtenir un taux de précision de 100%. Ce taux s'explique par le fait que le moteur *Sparql* utilisé ne renvoie pas d'items ne vérifiant pas le filtre de la requête et que ce filtre décrit avec précision les items pertinents. Ainsi, seul le rappel peut être calculé pour la recherche de communauté/service.

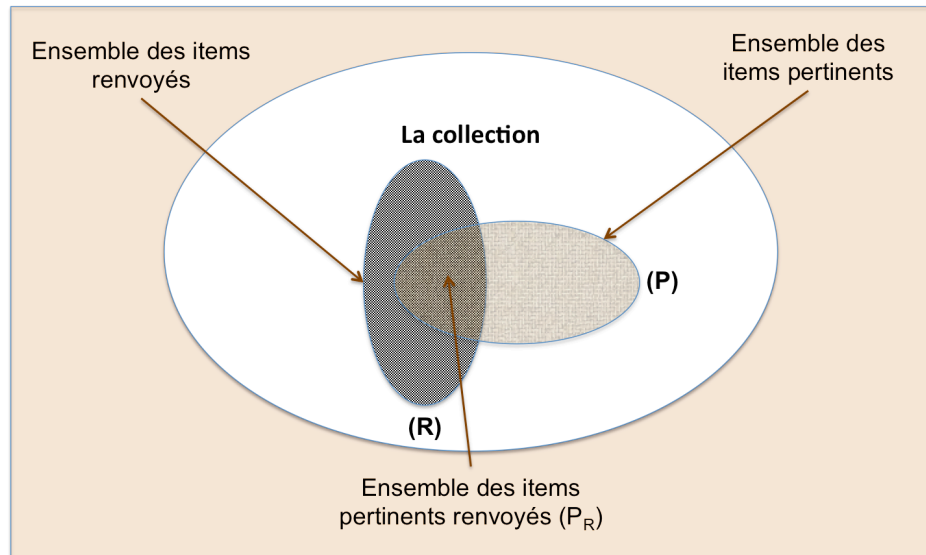


Figure 58: Les mesures rappel et précision dans un système d'information.

Après avoir présenté les mesures d'évaluation retenues. Nous décrivons, dans ce qui suit, les principes et les démarches pour l'expérimentation. Ensuite, nous exposons les résultats d'évaluation de nos approches de découverte de communautés et de services dédiés.

7.3.2 Principes d'expérimentation et résultats

Avant d'exposer les résultats en termes de précision et de rappel, nous présentons une comparaison élaborée sur les deux approches de calcul de distance entre deux ensembles de concepts : distance minimum et distance de *Hausdorff*. Cette expérience vient justifier notre choix de méthode utilisée pour le calcul de distance (présenté dans la section 7.1.1.1).

Pour cette comparaison, nous reprenons l'exemple de calcul de la distance entre deux ensembles de concepts représentant les centres d'intérêt d'un utilisateur et les intérêts des communautés (utilisée dans le mécanisme de recommandation des communautés).

La première approche consiste à calculer la distance de *Hausdorff* $Dist_{Hausdorff}$ (cf. Table 27, colonne 6) et la deuxième méthode consiste à utiliser la fonction *minimum* des distances $Dist_{Min}$ (cf. Table 27, colonne 8). Chacune de ces deux approches nous permet de classer les cinq communautés proposées en fonction de la distance renvoyée. Dans les colonnes 7 et 9 de la Table 27, nous présenterons respectivement le classement des communautés recommandées selon la $Dist_{Hausdorff}$ et la $Dist_{Min}$.

Pour ce test de comparaison de distances, nous avons pris un utilisateur X possédant deux centres d'intérêt 'Football' et 'Voiture' et cinq communautés différentes. Chacune des communautés proposées possède un ensemble d'intérêts différents (par exemple la communauté *Com1* possède deux intérêts 'Football' et 'Musique'). Prenons l'exemple de calcul de distance entre les intérêts de l'utilisateur X et les intérêts de *Com1*. La mesure de similarité de Wu & Palmer est utilisée pour calculer la distance entre un centre d'intérêt utilisateur CI_U et un centre d'intérêt communauté CI_C ($distance_{Wu\&Palmer} = 1 - similarité_{Wu\&Palmer}$). La $similarité_{Wu\&Palmer}$ est calculée entre concepts de l'ontologie *Wordnet* en utilisant l'outil WS4J⁸³. Par exemple, la $distance_{Wu\&Palmer}$ entre l'intérêt 'Voiture' de l'utilisateur X et l'intérêt 'football' de *Com1* est égale à $1 - 0.1739 = 0.8261$ (cf. la Figure 59).

WS4J Demo
 WS4J (WordNet Similarity for Java) measures semantic similarity/relatedness between words.

Type in texts below, or use:

1.	Input mode	<input checked="" type="radio"/> Word <input type="radio"/> Sentence
2.	Word 1	<input type="text" value="car#n#1"/>
3.	Word 2	<input type="text" value="football#n#1"/>
4.	Submit	<input type="button" value="Calculate Semantic Similarity"/>

Summary

wup(car#n#1 , football#n#1) = 0.1739

Figure 59: Similarité_{Wu&Palmer} (voiture, football) calculée via l'outil WS4J.

⁸³ <https://code.google.com/p/ws4j/>

Une fois que toutes les distances $_{Wu\&Palmer}(CI_U, CI_C)$ sont calculées, la distance entre les deux ensembles de concepts (intérêts utilisateur et intérêts des communautés) est calculée selon les deux méthodes (la $Dist_{Hausdorff}$ ou la $Dist_{Min}$). La valeur de distance entre les deux ensembles change selon la méthode de calcul utilisée, et, par conséquent, le classement des communautés proposé à l'utilisateur également.

			Intérêts utilisateur		Distance Hausdorff ($Dist_{Hausdorff}$)	Rang ($Dist_{Hausdorff}$)	Distance Minimum ($Dist_{Min}$)	Rang ($Dist_{Min}$)
			Voiture	Football				
Intérêts Communautés (Com)	Com1	Football	0.8261	0	0.7778	5	0	1
		Musique	0.7778	0.6471				
	Com2	Football	0.8261	0	0.0370	1	0	1
		Voiture sportive	0.0370	0.8333				
	Com3	Voiture sportive	0.0370	0.8333	0.1429	2	0.0370	2
		Sport d'équipe	0.8182	0.1429				
	Com4	Sport	0.8095	0.1000	0.2381	4	0.1000	3
		Transport	0.2381	0.7895				
	Com5	Voiture sportive	0.0370	0.8333	0.2000	3	0.0370	2
		Volley	0.8261	0.2000				
distances $_{Wu\&Palmer}$ (intérêts utilisateur, intérêts des communautés) = 1 - Similarité $_{Wu\&Palmer}$ calculée en utilisant l'outil WS4J présenté dans la Figure 59.								

Table 27: Test de comparaison entre distance minimum et distance de Hausdorff.

La Table 27 montre qu'en utilisant la méthode $Dist_{Min}$ (cf. Équation 3), les communautés *Com1* et *Com2*, tout comme les communautés *Com3* et *Com5*, possèdent des rangs à égalité car elles possèdent les mêmes valeurs de distance minimum (cf. Table 27, colonne 9). Cela est dû au fait que la $Dist_{Min}$ ne considère pas l'ensemble des concepts mais seulement les deux concepts les plus proches entre ceux de l'utilisateur et ceux de la communauté. En revanche, l'utilisation de la $Dist_{Hausdorff}$ (cf. Équation 4) remédie à cet inconvénient et propose des valeurs de distances plus précises, et, par conséquent, un classement des communautés plus significatif (cf. Table 27, colonne 7).

Selon la méthode $Dist_{Min}$, les ensembles qui possèdent la même distance minimum mais d'autres distances dissemblables sont considérés à une même distance les uns des autres. Ainsi, nous constatons que l'approche $Dist_{Min}$ valorise la distance la plus petite et dévalorise les autres distances. Tel est le cas pour les communautés *Com1* et *Com2* où la méthode $Dist_{Min}$ a considéré les valeurs de distances égales à zéro et a négligé les écarts entre les valeurs de distances entre les autres concepts comme (0.7778, 0.0370) et (0.6471, 0.8333).

Par contre, la distance de *Hausdorff* examine l'ensemble des distances et considère une distance plus significative : 'la distance maximale qui sépare un ensemble du point le plus proche de l'autre ensemble'. Selon les valeurs de $Dist_{Hausdorff}$, la *Com2* est la plus proche sémantiquement du profil de l'utilisateur X. On peut affirmer, donc, que n'importe quel intérêt de la *Com2* est au pire à une distance égale à 0.0370 d'un intérêt de l'utilisateur X.

7.3.2.1 Principes d'expérimentation

Notre expérience d'évaluation consiste à évaluer l'approche de recommandation de communautés et de services en termes de rappel et de précision. Notons cependant que pour nos évaluations, nous comparons les résultats du même système mais en nous basant sur deux types d'ontologie différentes : une ontologie spécialisée (*i.e.* ontologie Botanique) et *WordNet* utilisée comme une ontologie générique.

Nous n'avons pas trouvé de données réelles adaptées à notre cadre d'étude (des jeux de données comme *MovieLens*, *Foursquare*). Par conséquent, nous utilisons des données synthétiques produites avec notre prototype. Pour cela, nous générons des profils de communautés et des profils de services pour chaque ontologie utilisée. La génération des profils est une instanciation de l'ontologie de communauté et de l'ontologie des services (présentés dans la section 5.2). Nous avons essayé de diversifier la nature des centres d'intérêt pour les communautés (liées à la botanique ou générales) et d'enrichir la description des services avec des annotations sémantiques relatives aux ontologies du domaine.

Comme l'affirme (Genêt 2005) « *Seul un passage au banc d'essai d'utilisateurs pouvait nous fournir des indications objectives quant au degré de compréhensibilité et à la facilité d'utilisation du logiciel* ». C'est pourquoi, au cours de l'évaluation, nous avons soumis

le prototype à un panel de dix utilisateurs. Ces utilisateurs sont des doctorants de LIUPPA⁸⁴. Nous avons placé chaque utilisateur en situation de mobilité en lui expliquant le contexte de nos travaux (la découverte de territoire et la botanique) et en l'aidant à utiliser le prototype. Nous leurs avons demandé d'évaluer les résultats de la recommandation des communautés et la recommandation des services dans deux contextes différents (un contexte lié à la botanique et un contexte général non attaché à un domaine particulier).

Avant l'étape d'évaluation, nous avons demandé à chaque utilisateur de créer deux profils. Le premier profil contient des centres d'intérêt décrits en langage naturel et relatifs au domaine de la botanique. Le deuxième contient des centres d'intérêt détachés de toutes contraintes thématiques car ils sont annotés en utilisant *Wordnet*. Ensuite, nous avons demandé à chaque utilisateur d'évaluer les cinq premiers résultats retournés afin de comparer les taux de précision et les taux de rappel en utilisant les deux ontologies différentes. L'utilisateur donne la note de 1 s'il est satisfait de l'item retourné et 0 sinon.

Dans ce qui suit, nous présentons, en premier lieu, les expériences que nous avons menées sur la recommandation de communautés. En deuxième lieu, l'expérimentation que nous avons menée sur la recommandation des services.

7.3.2.2 Les résultats de la recommandation de communautés

Dans ce qui suit, nous comparons les résultats, pour chaque ontologie utilisée, du processus de la recommandation de communautés basés sur les profils créés par les utilisateurs et les profils des communautés générés synthétiquement (*i.e.* profils générés via le prototype et non pas des jeux de données de test).

Prenons l'exemple d'un utilisateur X, qui nous a indiqué qu'il s'intéresse aux 'orchidées'. Après le processus d'annotation, le système intègre ce centre d'intérêt dans son profil. Ensuite, nous lui avons demandé de consulter les communautés recommandées.

⁸⁴ <http://liuppa.univ-pau.fr/live/>

Pour évaluer les résultats retournés, nous avons parcouru le résultat communauté par communauté en notant 0 dans le cas où la communauté n'intéresse pas l'utilisateur, et 1 le cas inverse. Comme le montre la Table 28, nous avons calculé pour l'approche de recommandation de communautés les taux de précision et de rappel des cinq premières communautés retournées pour l'utilisateur X.

Les précisions et les rappels calculés pour la recommandation de communauté				
Communauté	Précision (botanique)	Précision (Wordnet)	Rappel (botanique)	Rappel (Wordnet)
1	1	1	0.33	0.33
2	1	0.5	0.66	0.33
3	1	0.66	1	0.66
4	0.75	0.75	1	1
5	0.6	0.6	1	1

Table 28: Les précisions et les rappels calculés pour la recommandation de communauté pour un utilisateur X.

Dans la deuxième colonne de la Table 28, les précisions sont calculées pour les résultats de la recommandation en utilisant le calcul de similarité sur l'ontologie botanique. Dans la troisième colonne du même tableau, les précisions sont calculées pour les résultats de la recommandation en utilisant le calcul de similarité sur l'ontologie *Wordnet*. En suivant le même principe de changement d'ontologie utilisé pour le calcul de similarité, nous avons calculé les différents taux de rappel. Nous présentons ces différents taux de précision et rappel via les courbes présentées en Figure 60.

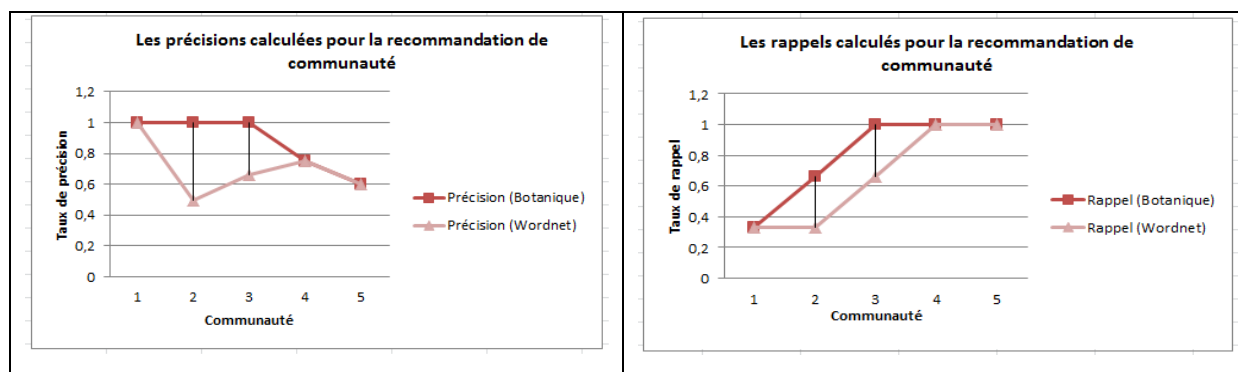


Figure 60: Courbes d'évaluation correspondant à la recommandation des communautés pour un utilisateur X.

Nous remarquons que la courbe qui correspond aux précisions calculées pour les résultats en utilisant l'ontologie botanique est au-dessus de celle qui correspond aux précisions calculées pour les résultats en utilisant l'ontologie *Wordnet*. Autrement dit, l'utilisation de l'ontologie botanique permet de trouver des communautés pertinentes dès les premiers résultats retournés (les trois premières communautés retournées), ensuite les deux courbes se ressemblent et se croisent aux alentours du 4ème résultat.

Cela est dû au fait que l'ontologie du domaine est plus spécialisée que l'ontologie *Wordnet* pour un domaine particulier et permet d'obtenir des résultats de calcul de similarité plus précis pour les concepts. Cette constatation s'appuie, également, sur des exemples comme ceux présentés dans la Table 29. Par exemple, les concepts '*fleur*' et '*arbre*' sont plus proches dans un contexte général que dans le contexte botanique. Par ailleurs, il existe une corrélation entre les fleurs '*glaiëuls*' et la couleur '*rose*' selon l'ontologie botanique, ce qui n'est pas le cas de l'ontologie *Wordnet*. En conclusion, si on est dans un contexte bien particulier et qu'on cherche des résultats plus précis, il est préférable d'utiliser une ontologie pour un domaine spécialisé.

	<i>Wordnet</i>	<i>Ontologie botanique</i>
Sim _{WU&PALMER} (Arbre, Fleur)	0.7826	0.6667
Sim _{WU&PALMER} (Glaiëuls, Rose)	0.1905	0.2857

Table 29: Comparaison entre la distance sémantique basée sur *Wordnet* et la distance sémantique pour l'ontologie botanique.

Ci-dessus, nous avons présenté les évaluations d'un seul utilisateur qui nous paraissent les plus importantes et les plus représentatives. Dans ce qui suit, nous présentons les expérimentations faites sur une dizaine d'utilisateurs. Nous avons calculé pour chacune des deux mesures la moyenne caractérisant le panel choisi des utilisateurs (*cf.* Équation 5).

$Moyenne\ de\ précision = \frac{\sum_{i=1}^i\ precision}{i}$	$Moyenne\ de\ rappel = \frac{\sum_{i=1}^i\ rappel}{i}$
--	--

Où *i* représente le nombre des utilisateurs.

Équation 5: La moyenne des précisions et la moyenne des rappels.

Nous présentons ces moyennes de précisions et de rappels dans la Figure 61. Nous remarquons, d'un côté, que les courbes de moyennes de précisions pour un panel de dix utilisateurs possèdent la même allure que la courbe de précisions pour l'utilisateur représentatif X, et, de l'autre, une diminution de l'écart entre les deux courbes de moyennes de rappels (comparé aux courbes de rappels pour l'utilisateur X). Le rapprochement entre les deux courbes de rappel n'est pas surprenant car on travaille sur un jeu de données réduit en quantité et on utilise les mêmes techniques de rapprochement. Au final, pour les deux approches de test, le système renvoie la même portion de communautés pertinentes au regard du nombre global des communautés pertinentes. En d'autres termes, pour le panel considéré des utilisateurs, le système parvient à retourner des items estimés pertinents dès les premiers résultats (taux de précision) en utilisant l'ontologie du domaine contre des résultats moins précis pour les premiers rangs en utilisant une ontologie plus générique. Mais au final, le système parvient à trouver le même nombre d'items pertinents par rapport au nombre total des items pertinents (taux de rappel), car il utilise les mêmes techniques de calcul de similarité pour les deux ontologies.

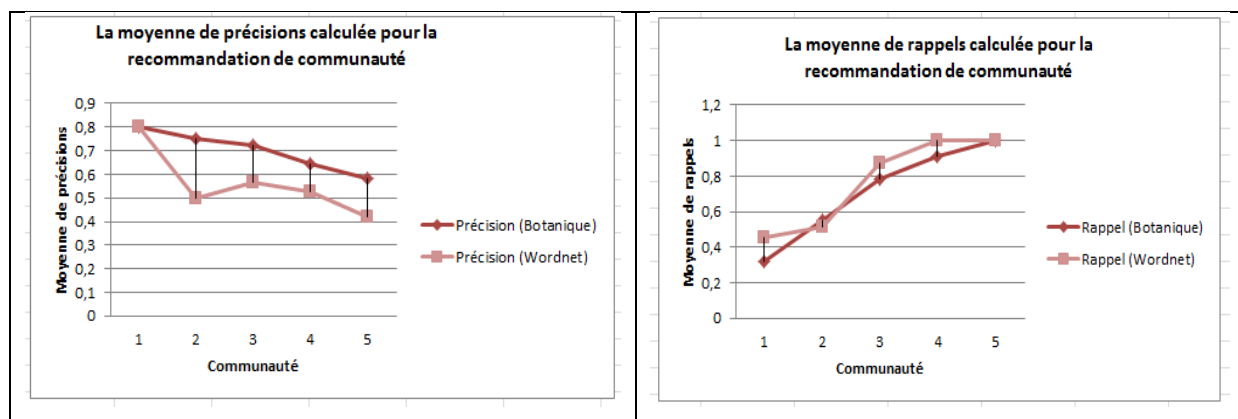


Figure 61: Courbes d'évaluation correspondant à la moyenne des précisions et à la moyenne des rappels pour la recommandation de communautés.

Suite à l'étude des différentes évaluations des utilisateurs, nous avons mentionné que le mécanisme de recommandation de communautés a donné des résultats positifs et a diminué la surcharge cognitive (*i.e.* excès des éléments proposés à l'utilisateur). Les remarques des utilisateurs ont toutes été positives et ils ont explorés toutes les fonctionnalités proposées.

Nous avons constaté aussi que l'utilisation d'une ontologie de domaine améliore les taux de précision et de rappel pour la recommandation des communautés en particulier, et des données en général. En revanche, nous avons constaté au cours des expérimentations quelque cas particuliers : (i) lorsque le profil de l'utilisateur n'a pas de lien sémantique avec les profils des communautés existantes, cela peut empêcher le rapprochement sémantique, (ii) les utilisateurs évaluent le résultat selon d'autres intérêts qui n'ont pas été pris en compte dans leur profil.

Après avoir présenté nos évaluations de l'approche de découverte de communautés, nous présentons, dans ce qui suit, les résultats de l'évaluation de l'approche de découverte de services.

7.3.2.3 Les résultats de la recommandation des services

Pour l'évaluation de la recommandation des services, nous avons suivi la même démarche que pour celle des communautés. Nous avons demandé à chaque utilisateur de créer deux nouvelles communautés : la première liée au domaine de la botanique, la deuxième détachée de toutes contraintes thématique. Pour chaque nouvelle communauté, les services qui ont été utilisés par des communautés sémantiquement proches sont recommandés aux membres de la nouvelle communauté. Ensuite, nous avons demandé d'évaluer les cinq premiers services recommandés afin de comparer les taux de précision et les taux de rappel en utilisant deux ontologies différentes.

Prenons l'exemple d'une communauté Y qui possède comme intérêt '*Musique*'. Cette dernière possède par exemple des liens sémantiques forts avec des communautés qui s'intéressent aux '*arts*' et aux '*films*'. Les services utilisés par ces communautés sémantiquement voisines lui sont recommandés. Ensuite, nous avons demandé au membre X d'évaluer les services recommandés selon le même principe que pour la recommandation de communauté (1 pour un service pertinent et 0 sinon). Comme le montre la Table 30, nous avons calculé, pour la recommandation de services, les taux de précision et de rappel des cinq premiers services renvoyés.

Les précisions et les rappels calculés pour la recommandation de service				
Service	Précision (botanique)	Précision (Wordnet)	Rappel (botanique)	Rappel (Wordnet)
1	1	1	0.5	0.5
2	0.5	1	0.5	1
3	0.33	0.66	0.5	1
4	0.5	0.5	1	1
5	0.4	0.4	1	1

Table 30: Les précisions et les rappels calculés pour la recommandation de service pour une communauté Y.

A l'encontre des résultats obtenus pour la recommandation des communautés, les taux de rappel et de précision obtenus suite à l'utilisation de l'ontologie botanique sont moins satisfaisants que ceux produits suite à l'utilisation de *Wordnet*. Comme illustré dans la Figure 62, la courbe précision en utilisant *Wordnet* est au-dessus de la courbe précision en utilisant l'ontologie botanique. Les courbes de rappel se comportent de la même manière.

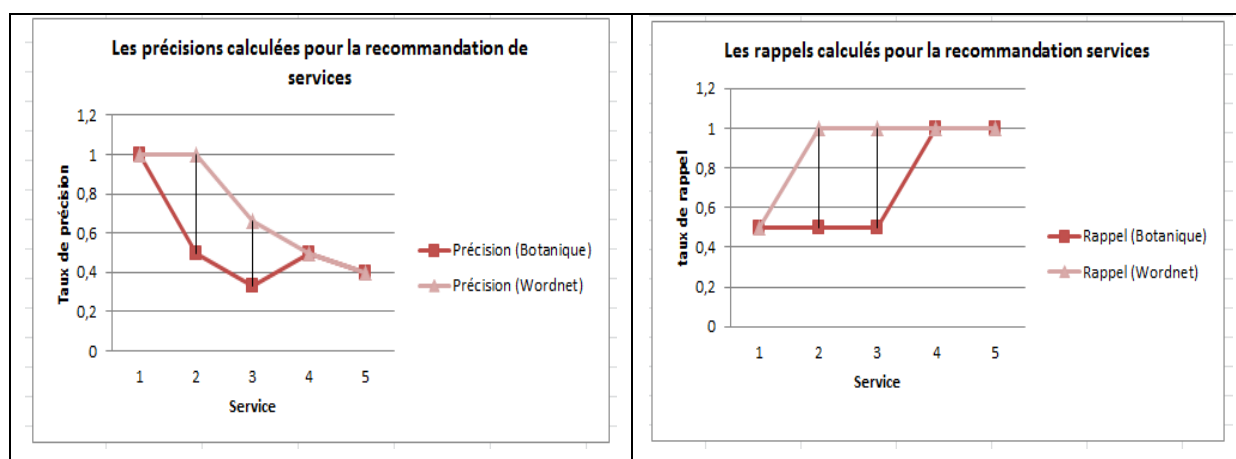


Figure 62: Courbes d'évaluation correspondant à la recommandation des services pour une communauté Y.

Nous rappelons que l'ontologie de service, utilisée pour décrire sémantiquement les propriétés et les capacités des services, introduit d'autres ontologies pour décrire les concepts utilisés dans le domaine "métier" auquel s'applique le service (botanique, tourisme, transport, etc.). Pour nos expérimentations, nous avons utilisé deux ontologies différentes pour annoter la description du service : *Wordnet* et l'ontologie botanique. Les résultats d'évaluation

paraissent meilleurs en utilisant une ontologie plus générique. Ceux-ci peut s'expliquer par le fait que, si un utilisateur est intéressé par un contenu, il sera probablement intéressé par du contenu sémantiquement proche. Ceci n'est pas toujours le cas des services. Par exemple, si deux communautés s'intéressent au même sujet 'cuisine' (une similarité sémantique égale à 1), l'une peut s'intéresser au service permettant de lire des vidéos de cuisine et l'autre au service permettant de produire des recettes. Nous constatons que l'utilisation d'une ontologie du domaine spécialisée améliore le rapprochement du contenu mais pas celui des services. Ce constat a été confirmé en calculant la moyenne des précisions et des rappels pour un panel de communautés.

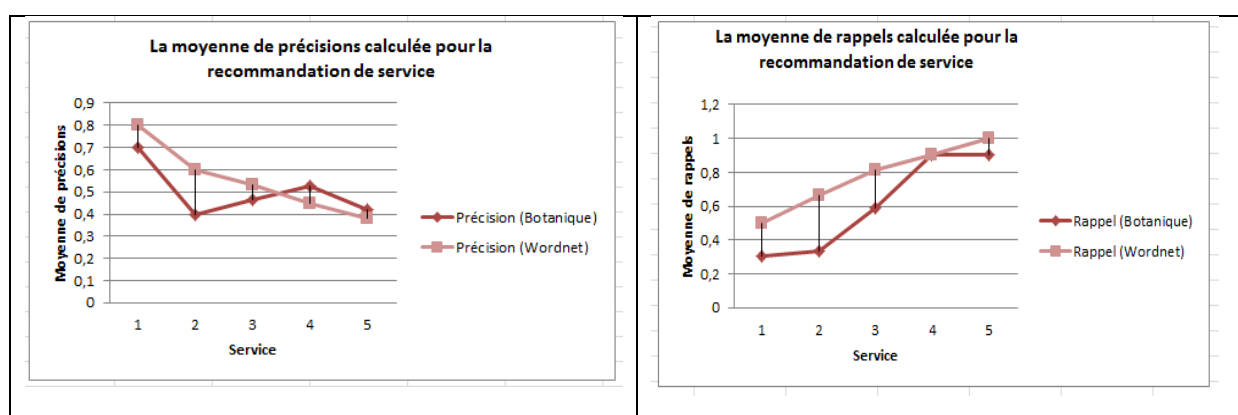


Figure 63: Courbes d'évaluation correspondant à la moyenne des précisions et la moyenne des rappels pour la recommandation de services.

Pour la recommandation de services, nous avons adopté l'approche de recommandation sociale. C'est-à-dire, se baser sur le comportement passé (utilisation des services) des communautés similaires (*i.e.* effectuer une corrélation entre des communautés ayant des intérêts proches sémantiquement) pour recommander des services à une communauté donnée. Après une observation des résultats, nous nous demandons si cette recommandation sociale est plus intéressante qu'une recommandation personnalisée (indépendamment des autres communautés) réalisée en fonction du profil de la communauté et plus précisément sa typologie. En d'autres termes, les services seront recommandés à la communauté en fonction de son type inféré (communauté d'intérêt, communauté géographique, communauté de pratique, *etc.*).

7.4 Conclusion

Dans ce chapitre, nous avons commencé par présenter *Kalimucho* la plateforme sensible au contexte dédiée au déploiement et à l'adaptation des applications *Taldea* en fonction de l'environnement physique. Ensuite, nous avons montré un exemple d'exécution du prototype *Taldea* dans le contexte de la botanique. Nous avons illustré comment notre application communautaire permet d'aider les utilisateurs nomades à intégrer des communautés environnantes et à découvrir les services dédiés à l'échange et au partage des intérêts, des sujets, des activités, des événements, *etc.* Enfin, nous avons évalué, via le prototype *Taldea*, le mécanisme de découverte de communautés et de services.

Après avoir soumis notre travail à cette évaluation préliminaire, nous présentons dans le chapitre suivant une synthèse des travaux présentés, et nous donnons les perspectives et les directions de recherches possibles pour prolonger ce travail.

Chapitre 8 : Conclusion et perspectives

8.1 Conclusion générale

Le présent travail se positionne dans le domaine pluridisciplinaire du *pervasive social computing*. Dans cette thématique, nous nous sommes focalisées sur la question des applications centrées communautés dans un environnement ubiquitaire. Les applications usuelles présentent un certain nombre de limites (cf. **Table 31**). Les apports théoriques de notre contribution concernent les trois aspects évoqués dans la problématique définies dans le chapitre introduction et développés tout au long de ce manuscrit : la notion de communauté, le modèle de communautés, l'architecture des applications communautaires (cf. **Table 31**).

	<i>Limites des approches existantes</i>	<i>Apports de l'approche Taldea</i>
Notion de communauté	Thématique	Spontanée
	Pérenne	Éphémère géolocalisée
Modèle de communautés	Dépendant du domaine	Indépendant du domaine
	Conflit de données	Interopérabilité
	Syntaxique	Sémantique
Architecture des applications communautaires	Surcharge cognitive	Adaptation de contenu et de fonctionnalités
	Statique	Dynamiquement reconfigurable

Table 31: Tableau récapitulatif des apports théoriques de la thèse.

Dans la partie état de l'art, nous avons commencé par une étude de la notion de communauté, ses définitions, les caractéristiques communes et les différences, la typologie. Cela nous a permis de dégager une vue d'ensemble afin de définir, concevoir et mettre en œuvre un modèle de communauté pour nos applications communautaires. Ensuite, nous avons étudié la corrélation entre communauté et applications informatiques en présentant les travaux

qui ont utilisé la notion de communauté comme un facteur de progrès pour les applications informatiques et les travaux où les applications informatiques représentent un support à l'expansion des communautés. Cette étude nous a permis également d'identifier le gain associé à l'informatisation de la communauté, et les limites des applications communautaires existantes. La combinaison des objectifs généraux des applications communautaires et les difficultés à les surmonter nous ont donné une vision sur les défis et les objectifs de nos applications communautaires.

Les applications centrées communautés se situent dans le cadre d'un domaine en plein essor : le *social computing*. Ce domaine recouvre plusieurs disciplines comme les sciences sociales, les sciences informatiques, et bénéficie de plusieurs technologies alors qu'il concerne différents domaines d'applications (social media, divertissement éducatif, *etc.*). Dans ce manuscrit, nous avons présenté les fondements théoriques ainsi que les technologies qui se rapportent aux applications communautaires. L'exemple emblématique de ce domaine est celui des réseaux sociaux. Nous avons étudié et réalisé une classification selon leurs usages et leurs architectures, ce qui nous a permis de dresser un panorama de leurs atouts et de leurs limites.

Suite à l'avènement du paradigme ubiquitaire, un bouleversement en profondeur dans le domaine du *social computing* a eu lieu. Cette rencontre fructueuse entre le *social computing* et l'informatique ubiquitaire a permis d'augmenter les possibilités et de concrétiser les usages des applications sociales d'une part, et d'agréger, d'enrichir, de recréer, de contextualiser les informations et les services échangées d'autre part. Nous avons étudié les différents aspects de l'informatique ubiquitaire (comme la notion de contexte et les plateformes de sensibilité au contexte) et les principaux travaux dans lesquels convergent le monde ubiquitaire et le monde social. Ensuite, nous avons établi une comparaison s'appuyant sur les éléments du contexte sur lesquels porte l'adaptation des applications sociales (temporalité, localité, système, social) et le type d'adaptation considérée (de contenu, de fonctionnalités, structurelle, comportementale). L'examen de ces différentes applications sociales pervasives a servi de point de départ pour concevoir notre architecture dédiée aux applications communautaires pervasives.

Dans la partie contribution, nous avons présenté notre proposition organisée selon trois axes interconnectés :

La proposition de la notion de communauté spontanée: nous avons proposé un nouveau type de communautés qui s'accorde mieux avec des situations d'ubiquité et/ou de mobilité. Ce sont les communautés spontanées que nous définissons comme « *un regroupement spontané d'individus ayant un intérêt commun lié à une situation circonstancielle relative à un territoire géographique et limité dans le temps.*». Une communauté spontanée désigne un ensemble d'utilisateurs qui partagent des intérêts ou des pratiques communes, se retrouvent dans une situation inopinée caractérisée par une durée de vie éphémère et délimitée géographiquement et interagissant via une application sociale. Ce type de communauté est caractérisé par :

- La *spontanéité* c'est-à-dire le fait d'être détachée de toutes contraintes thématiques imposées par le domaine de l'application. Elle est caractérisée par un ou plusieurs centres d'intérêt définis par les membres eux-mêmes et leur situation présente (besoins, objectifs, nouvel emplacement géographique, *etc.*).
- Une *durée de vie éphémère* c'est-à-dire limitée par un intervalle de temps réduit au-delà duquel elle peut disparaître. Cette caractéristique rend possible la définition d'une politique de contrôle d'accès aux informations valable sur une période donnée.
- Un *territoire délimité géographiquement* qui constitue le point commun qu'on peut percevoir entre les membres. Ce territoire permet de mesurer la correspondance entre la localisation d'un utilisateur et le territoire géographique d'une communauté.

La conception d'un modèle de représentation de communauté à base d'ontologies : nous avons proposé un modèle à bases d'ontologies qui représente la notion de communauté. Ce modèle se positionne à un niveau conceptuel générique afin d'être appliqué à n'importe quel domaine donné, et de garantir le caractère spontanée de la communauté. À travers ce modèle, nous avons fourni une vision unique et homogène de la notion de communauté permettant d'ajouter une couche sémantique à la recherche et à la consultation des informations associées et d'enrichir la base de connaissances communautaire en inférant automatiquement de nouvelles informations. Ce modèle importe des vocabulaires existants et

y ajoute des extensions aux niveaux des concepts et des relations. L'utilisation des vocabulaires du web sémantique permet d'assurer l'homogénéité et l'interopérabilité des données au sein des communautés.

La définition d'une architecture dédiée aux applications communautaires dynamiquement reconfigurables : nous avons proposé l'architecture Taldea dont l'objectif est de produire des applications pour promouvoir les communautés spontanées. Notre architecture se base sur le modèle de communauté proposé.

Afin de répondre à la problématique de surcharge cognitive des communautés et des services dédiés, Taldea propose deux mécanismes : i) un premier module d'accès aux communautés qui permet de faciliter l'accès et l'interaction d'un utilisateur avec les communautés. ii) un deuxième module permettant d'assister les communautés et leurs membres dans la découverte des services les mieux adaptés à leurs besoins.

Pour la réalisation des applications communautaires ubiquitaires, le travail de déploiement dynamique et d'adaptation a été confié à la plateforme logicielle Kalimucho dont le rôle est la gestion du contexte d'exécution de l'application (acquisition, modélisation, raisonnement, prise de décision d'adaptation), et la mise en œuvre des adaptations par déploiement ou reconfiguration des services de l'application.

Ce travail a été validé, d'une part, en proposant un prototype implémentant l'architecture proposée. D'autre part, nous avons évalué les mécanismes d'accès aux communautés et de découverte des services.

8.2 Perspectives

Plusieurs perspectives peuvent être dégagées au terme de nos travaux. Certaines concernent l'évaluation et l'amélioration de notre proposition, d'autres ciblent de nouvelles extensions.

Evaluation approfondie : à ce jour, seules les approches de découverte de services et de communautés sont évaluées par des membres de notre équipe de recherche en effectuant des tests simples. Nous souhaitons tester nos approches sur d'autres cas ou scénarios avec des données réelles. D'autres expériences à mener consisteraient à faire varier les valeurs du seuil

utilisées pour la recommandation de communauté et des services, ainsi que les valeurs des paramètres (α et β) de pondération entre la proximité sémantique et la proximité géographique (utilisateur, communauté) dans le but de déterminer la bonne configuration pour que de meilleurs résultats soient retournés à l'utilisateur.

Implantation décentralisée : nous avons présenté *Taldea* comme une architecture hybride qui combine les propriétés des architectures centralisées et distribuées. Dans la réalisation présentée dans le chapitre 8, le *Manager* de communauté est placé dans un serveur central auquel se connectent les dispositifs des utilisateurs pour découvrir des communautés et des services. Même si les échanges, via les services, entre les membres de la communauté se déroulent d'une façon décentralisée, la gestion des communautés et des services se fait en mode centralisé, ce qui résulte une architecture hybride. Il serait plus efficace de faire évoluer cette architecture vers une version décentralisée dans laquelle le *Manager* de communauté est réparti sur les dispositifs des utilisateurs. Il serait peut être intéressant de s'inspirer de l'architecture des réseaux sociaux décentralisés, et, plus précisément, les réseaux sociaux pair-à-pair ou les réseaux mobiles *ad hoc*. Cela permettra de renforcer la robustesse du système en évitant le nœud central représentant le point critique et d'effacer le besoin d'avoir une infrastructure préalable pour les communautés. Tout comme l'architecture, l'ontologie de communauté peut être distribuée. La décomposition d'une ontologie a pour résultat un ensemble de sous-ontologies regroupées dans une ontologie distribuée. Avec l'architecture actuelle, aucune ontologie n'existe sur les dispositifs des utilisateurs, tout est centralisé sur un serveur central. La décomposition de l'ontologie préserve toujours la sémantique et les services d'inférence de l'ontologie originale, mais permettra de plus de travailler en mode semi-connecté (si seule une partie de l'ontologie est transmise au dispositif de l'utilisateur) ou en mode déconnecté (où toute l'ontologie existe sur le dispositif de l'utilisateur).

Plus de protection des données personnelles : dans notre architecture, la question de la protection des données est traitée par le développement d'un contrôleur de confidentialité qui surveille l'accès aux ressources en s'inspirant de modèle très connu des réseaux sociaux (privé, amis, public). Ce contrôleur de confidentialité vérifie si une entité (personne, communauté) demandant l'accès à une ressource (contenu, service) a les droits nécessaires pour le faire. Le modèle adopté permet de régler les paramètres d'accès aux ressources selon

les préférences des membres de la communauté. L'ouverture vers des sources d'information comme les réseaux sociaux engendrera plus de problèmes au niveau de la protection des données. Par conséquent, nous envisageons des méthodes plus avancées de protection des informations personnelles et des ressources des communautés en couplant la représentation de ces données avec des politiques d'accès aux données approuvées.

Nouvelles extensions : nous soulignons ici les perspectives qui nous semblent envisageables pour l'évolution des propositions réalisées.

i) Nous entendons effectuer le calcul de similarité sémantique entre concepts dans le cadre d'alignement d'ontologies. En réalité, il existe plusieurs ontologies qui décrivent le même domaine et qui nécessitent d'être alignées pour recouvrir les différents points de vues, d'où la nécessité de proposer une méthode de calcul de similarité sémantique permettant de supporter des ontologies hétérogènes en entrée et découvrir de manière automatique les liens sémantiques entre leurs éléments. La mise en correspondance des différentes ontologies permettra d'améliorer les approches d'appariement (utilisateur, communauté) et (communauté, service).

ii) Une autre perspective intéressante consiste en l'application d'un processus de filtrage sur des types d'information autres que les communautés comme sur les contenus échangés au sein d'une communauté, par exemple. L'échange massif du contenu intra-communauté nous permet d'envisager, dans un avenir proche, l'exploitation de leurs métadonnées. Une ressource échangée (un document par exemple) est annotée par des métadonnées thématiques (concepts de l'ontologie du domaine) ou des métadonnées contextuelles (données spatio-temporelles). L'idée est de suggérer du contenu à un membre de la communauté en effectuant un rapprochement entre les annotations thématiques et son profil ou les annotations contextuelles et son contexte actuel. Par exemple, recommander à un utilisateur les photos qui ont été prises dans la même localisation.

iii) Dans notre approche, nous souhaiterons aussi améliorer l'acquisition des profils des utilisateurs. Jusqu'à présent le profil est créé explicitement par l'utilisateur puis enrichi suite à l'observation de ces interactions avec le système (plus précisément ses requêtes). Des

données comme les centres d'intérêt sont renseignés par l'utilisateur lui-même, puis annotées via un service d'annotation. Nous envisageons d'inférer les données du profil de l'utilisateur en se servant de ses différents profils sur les réseaux sociaux (par exemple, son profil Facebook). Egalement, il nous semble avantageux d'importer les communautés implicites sur les réseaux sociaux, détectées à l'aide d'un des algorithmes proposés dans la littérature (Leprovost et al. 2012b), et de concrétiser leurs rapports à travers des échanges explicites via nos applications communautaires.

iv) Enfin, une perspective importante de notre travail est l'utilisation des communautés spontanées dans les collectes de données participatives mettent en œuvre le Crowdsourcing (*i.e.* production participative), c'est-à-dire impliquer les communautés, dans une approche économique, dans la production du contenu collaboratif pour un organisme. Afin d'offrir une plus-value fonctionnelle et ne pas se limiter à des applications de type formulaires, nous proposons d'adapter les applications à ce type d'activité pour permettre aux utilisateurs finaux (membres de communautés) de publier des tâches (appelés producteurs) afin qu'elles soient accomplies par d'autres utilisateurs (appelés consommateurs). À savoir qu'un producteur (non spécialistes) exprime une demande en concevant graphiquement une application que des consommateurs utiliseront pour répondre à sa demande. Le modèle d'application généré par le producteur devra alors permettre l'adaptation dynamique de l'application selon le contexte (environnemental, matériel, profil) de chaque consommateur et offrir des interactions complexes (entre consommateurs) et multimodales.

Bibliographie

- Abrouk, L., Gross-Amblard, D., and Leprovost, D. "Découverte de communautés par analyse des usages," *Extraction et gestion des connaissances-Atelier Web Social*, 2010, pp. A5-5--A5-16.
- Akkiraju, R., Farrell, J., Miller, J. A., Nagarajan, M., Sheth, A. P., and Verma, K. "Web service semantics-wsdl-s," 2005.
- Al Mutawa, N., Baggili, I., and Marrington, A. "Forensic analysis of social networking applications on mobile devices," *Digital Investigation* (9, Supplement:0) 2012, pp S24-S33.
- Alrahabi, M., and Desclés, J.-P. "EXCOM: Plate-forme d'annotation sémantique de textes multilingues," *TALN 2009*) 2009.
- Aussenac-Gilles, N., Buscaldi, D., Comparot, C., and Kamel, M. "Enrichissement d'ontologies grâce à l'annotation sémantique de pages web," *EGC*, 2013, pp. 229-234.
- Bai, X., Yu, H., Wang, G., Ji, Y., Marinescu, G. M., Marinescu, D. C., and Bölöni, L. "Intelligent grids," in: *Grid Computing: Software Environments and Tools*, Springer, 2006, pp. 45-74.
- Balaji, S., and Sasikala, S. "TD-Signet: Community Mining With Wsd Based On Implied Graph Structure In Social Networks," *International Journal of Engineering Science and Technology* (3:6) 2011, pp 4588-4596.
- Barab, S., Kling, R., and Gray, J. H. *Designing for virtual communities in the service of learning* Cambridge University Press, 2004.
- Battle, S., Bernstein, A., Boley, H., Grosz, B., Gruninger, M., Hull, R., Kifer, M., Martin, D., McIlraith, S., and McGuinness, D. "Semantic web services ontology (swso)," *Member submission, W3C*) 2005.
- Beach, A., Gartrell, M., Akkala, S., Elston, J., Kelley, J., Nishimoto, K., Ray, B., Razgulin, S., Sundaresan, K., and Surendar, B. "Whozthat? evolving an ecosystem for context-aware mobile social networks," *Network, IEEE* (22:4) 2008, pp 50-55.
- Becker, C., and Bizer, C. "DBpedia Mobile: A Location-Enabled Linked Data Browser," *LDOW* (369) 2008.

Bibliographie

- Ben Mokhtar, S., and Capra, L. "From pervasive to social computing: algorithms and deployments," Proceedings of the 2009 international conference on Pervasive services, ACM, 2009, pp. 169-178.
- Benghozi, P.-J., Bitouzet, C., Soulier, E., and Zacklad, M. "Le mode communautaire: vers une nouvelle forme d'organisation," 3ème Colloque ICUST e-usages, 2001.
- Breslin, J. G., Harth, A., Bojars, U., and Decker, S. "Towards semantically-interlinked online communities," in: *The Semantic Web: Research and Applications*, Springer, 2005, pp. 500-514.
- Brisson, L. "Intégration de connaissances expertes dans le processus de fouille de données pour l'extraction d'informations pertinentes," Université Nice Sophia Antipolis, 2006.
- Buchegger, S., Schiöberg, D., Vu, L.-H., and Datta, A. "PeerSoN: P2P social networking: early experiences and insights," Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, ACM, 2009, pp. 46-52.
- Burners-Lee, T. "Long Live the Web: A Call for Continued Open Standards and Neutrality," *Scientific American*) 2010.
- Cabitza, F., Locatelli, M. P., Sarini, M., and Simone, C. "CASMAS: Supporting collaboration in pervasive environments," Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on, IEEE, 2006, pp. 10 pp.-295.
- Camenisch, J., Leenes, R., and Sommer, D. *Digital Privacy: PRIME-Privacy and Identity Management for Europe* Springer, 2011.
- Cardoso, J., Miller, J., and Emani, S. "Web Services Discovery Utilizing Semantically Annotated WSDL," in: *Reasoning Web*, C. Baroglio, P. Bonatti, J. Małuszyński, M. Marchiori, A. Polleres and S. Schaffert (eds.), Springer Berlin Heidelberg, 2008, pp. 240-268.
- Carminati, B., Ferrari, E., Heatherly, R., Kantarcioglu, M., and Thuraisingham, B. "Semantic web-based social network access control," *computers & security* (30:2) 2011, pp 108-115.
- Carpenter, C. J. "Narcissism on Facebook: Self-promotional and anti-social behavior," *Personality and individual differences* (52:4) 2012, pp 482-486.
- Chabeb, Y., Tata, S., and Ozanne, A. "Yasa-m: A semantic web service matchmaker," Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, IEEE, 2010, pp. 966-973.

Bibliographic

- Chakravarthy, A., Lanfrachi, V., and Ciravegna, F. "Community-based Annotation of Multimedia Documents," 2006.
- Chao, W., Guo, Y., and Zhou, B. "Social networking federation: A position paper," *Computers & Electrical Engineering* (38:2) 2012, pp 306-329.
- Christo El, M., Mihaela, D.-P., Marcia, R., Julien, S., Pierre, M., and Natalia, B. "Enabling Virtual Knowledge Networks for Human Rights Monitoring for People with Disabilities," *International Journal of Virtual Communities and Social Networking (IJVCSN)* (4:2) 2012, pp 1-17.
- Conan, D., Rouvoy, R., and Seinturier, L. "Scalable processing of context information with COSMOS," *Distributed Applications and Interoperable Systems*, Springer, 2007, pp. 210-224.
- Coutand, O., Droegehorn, O., David, K., Nurmi, P., Floréen, P., Kernchen, R., Holtmanns, S., Campadello, S., Kanter, T., and Martin, M. "Context-aware group management in mobile environments," *IST Mobile Summit*, Citeseer, 2005.
- Da, K. "Plateforme d'adaptation autonome contextuelle à base de connaissances," Pau, 2014.
- Da, K., Dalmau, M., and Roose, P. "Kalimucho: middleware for mobile applications," *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, ACM, 2014, pp. 413-419.
- Dalmau, M., and Roose, P. "Kalimucho : Plateforme logicielle distribuée de supervision d'applications," 2013.
- Daqing, Z., Zhu, W., Bin, G., and Zhiwen, Y. "Social and Community Intelligence: Technologies and Trends," *Software, IEEE* (29:4) 2012, pp 88-92.
- De Bruijn, J., Lausen, H., Polleres, A., and Fensel, D. *The web service modeling language WSML: an overview* Springer, 2006.
- Deparis, É., Abel, M.-H., Lortal, G., and Mattioli, J. "Information management from social and documentary sources in organizations," *Computers in Human Behavior* (30) 2014, pp 753-759.
- Deparis, É., Abel, M., and Mattioli, J. "Modeling a social collaborative platform with standard ontologies," *Signal-Image Technology and Internet-Based Systems (SITIS)*, 2011 Seventh International Conference on, IEEE, 2011, pp. 167-173.
- Dey, A. K. "Understanding and using context," *Personal and ubiquitous computing* (5:1) 2001, pp 4-7.

Bibliographie

- Domingue, J., Roman, D., and Stollberg, M. "Web service modeling ontology (WSMO)-An ontology for semantic web services," Position paper at the W3C Workshop on Frameworks for Semantics in Web Services, 2005, pp. 9-10.
- Dufournet, S. "Typologie des réseaux sociaux," <http://www.idnum.fr/>, 2014.
- Ehrig, M., Haase, P., Hefke, M., and Stojanovic, N. "Similarity for ontologies-a comprehensive framework," *ECIS 2005 Proceedings*) 2005, p 127.
- Fauvet, M.-C. "Sélection dynamique de services Web—une approche à base de communautés; projet Web Intelligence-Cluster ISLE-Région Rhône-Alpes," France, 2008.
- Fjellheim, F. "Over-the-air deployment of applications in multi-platform environments," Software Engineering Conference, 2006. Australian, IEEE, 2006, p. 10 pp.
- Floch, J., Frà, C., Fricke, R., Geihs, K., Wagner, M., Lorenzo, J., Soladana, E., Mehlhase, S., Paspallis, N., and Rahnama, H. "Playing MUSIC—building context-aware and self-adaptive mobile applications," *Software: Practice and Experience* (43:3) 2013, pp 359-388.
- Fong, P. W., Anwar, M., and Zhao, Z. "A privacy preservation model for facebook-style social network systems," in: *Computer Security—ESORICS 2009*, Springer, 2009, pp. 303-320.
- Garrot, É., George, S., and Prévôt, P. "Instrumentation d'une communauté de pratique virtuelle: illustration avec le portail TE-Cap," Proceedings of 2nd International Workshop On Learning and working in CoPs, 2007.
- Gartrell, C. M. *Socialaware: Context-aware multimedia presentation via mobile social networks* ProQuest, 2008.
- Genêt, M. "Favoriser la perception des communautés en ligne et la diffusion des connaissances en résumant les informations publiées sur des pages personnelles," 2005.
- Graffi, K., Gross, C., Stingl, D., Hartung, D., Kovacevic, A., and Steinmetz, R. "LifeSocial. KOM: A secure and P2P-based solution for online social networks," Consumer Communications and Networking Conference (CCNC), 2011 IEEE, IEEE, 2011, pp. 554-558.
- Gruber, T. R. "A translation approach to portable ontology specifications," *Knowledge acquisition* (5:2) 1993, pp 199-220.
- Guissé, A., Lévy, F., and Nazarenko, A. "Un moteur sémantique pour explorer des textes réglementaires," *Actes des 22èmes Journées Francophones d'Ingénierie des Connaissances*) 2011, pp 451-458.

Bibliographic

- Gupta, A., Kalra, A., Boston, D., and Borcea, C. "MobiSoC: a middleware for mobile social computing applications," *Mobile Networks and Applications* (14:1) 2009, pp 35-52.
- Hamida, A. B., Le Mouël, F., Frénot, S., and Ahmed, M. B. "A graph-based approach for contextual service loading in pervasive environments," in: *On the Move to Meaningful Internet Systems: OTM 2008*, Springer, 2008, pp. 589-606.
- Jessup, L. M., and Robey, D. "The relevance of social issues in ubiquitous computing environments," *Communications of the ACM* (45:12) 2002, pp 88-91.
- Jiang, J. J., and Conrath, D. W. "Semantic similarity based on corpus statistics and lexical taxonomy," *arXiv preprint cmp-lg/9709008* 1997.
- Joly, A., Maret, P., and Daigremont, J. "Context-awareness, the missing block of social networking," *International Journal of Computer Science and Applications* (4:2) 2009.
- Kayser, D. "La représentation des connaissances," 1997.
- Klusch, M. "Semantic web service coordination," in: *CASCOM: Intelligent Service Coordination in the Semantic Web*, Springer, 2008, pp. 59-104.
- Klusch, M., Fries, B., and Sycara, K. "OWLS-MX: A hybrid Semantic Web service matchmaker for OWL-S services," *Web Semantics: Science, Services and Agents on the World Wide Web* (7:2) 2009a, pp 121-133.
- Klusch, M., and Kaufer, F. "WSMO-MX: A hybrid Semantic Web service matchmaker," *Web Intelligence and Agent Systems* (7:1) 2009b, pp 23-42.
- Kruk, S. R., Grzonkowski, S., Gzella, A., Woroniecki, T., and Choi, H.-C. "D-FOAF: Distributed identity management with access rights delegation," in: *The Semantic Web-ASWC 2006*, Springer, 2006, pp. 140-154.
- Laforest, F., Le Sommer, N., Frénot, S., De Corbière, F., Mahéo, Y., Launay, P., Gravier, C., Subercaze, J., Reimert, D., and Brodu, E. "C3PO: a Spontaneous and Ephemeral Social Networking Framework for a collaborative Creation and Publishing of Multimedia Contents," *procedia computer science*, Elsevier, 2014, pp. 1-6.
- Lam, M. S. "How mobile disrupts social as we know it," in: *Proceedings of the 2013 international conference on Intelligent user interfaces*, ACM, Santa Monica, California, USA, 2013, pp. 315-316.
- Lefebvre, A. *Les Réseaux sociaux: pivot de l'Internet 2.0* M21 Editions, 2005.
- Leprovost, D., Abrouk, L., Cullot, N., and Gross-Amblard, D. "Temporal semantic centrality for the analysis of communication networks," in: *Web Engineering*, Springer, 2012a, pp. 177-184.

Bibliographic

- Leprovost, D., Abrouk, L., and Gross-Amblard, D. "Discovering implicit communities in web forums through ontologies," *Web Intelligence and Agent Systems* (10:1) 2012b, pp 93-103.
- Licklider, J. C., and Taylor, R. W. "The computer as a communication device," *Science and technology* (76:2) 1968, pp 1-3.
- Lin, D. "An information-theoretic definition of similarity," ICML, 1998, pp. 296-304.
- Louberry, C., Roose, P., and Dalmau, M. "Kalimucho: Contextual deployment for qos management," *Distributed Applications and Interoperable Systems*, Springer, 2011, pp. 43-56.
- Ma, Y., Audibert, L., and Nazarenko, A. "Ontologies étendues pour l'annotation sémantique," 20es Journées Francophones d'Ingénierie des Connaissances, 2009, pp. 205-216.
- Maret, P., Laforest, F., and Lanquetin, D. " A semantic web model for ad hoc context-aware communities : Application to the Smart Place Scenario.," in: *International Conference on Enterprise Information Systems*, Portugal, 2014, pp. PP 591-598.
- Menkens, C. "Towards a context aware mobile community application platform," *Information Technology: New Generations*, 2009. ITNG'09. Sixth International Conference on, IEEE, 2009, pp. 1504-1509.
- Miron, A. D. "Découverte d'associations sémantiques pour le Web Sémantique Géospatial-le framework ONTOAST," Université Joseph-Fourier-Grenoble I, 2009.
- Napoli, A. "Une introduction aux logiques de descriptions,") 1997.
- Nguyen, T., Loke, S. W., Torabi, T., and Lu, H. "PlaceComm: A framework for context-aware applications in place-based virtual communities," *Journal of Ambient Intelligence and Smart Environments* (3:1) 2011, pp 51-64.
- Northrop, E. E., and Lipford, H. R. "Poster: Ephemerality in Social Media,") 2013.
- Orman, K., Labatut, V., and Cherifi, H. "An empirical study of the relation between community structure and transitivity," in: *Complex Networks*, Springer, 2013, pp. 99-110.
- Paolucci, M., Kawamura, T., Payne, T. R., and Sycara, K. "Semantic matching of web services capabilities," in: *The Semantic Web—ISWC 2002*, Springer, 2002, pp. 333-347.
- Parameswaran, M., and Whinston, A. B. "SOCIAL COMPUTING: AN OVERVIEW," *Communications of the Association for Information Systems* (19) 2007.

Bibliographie

- Park, I., Lee, D., and Hyun, S. J. "A dynamic context-conflict management scheme for group-aware ubiquitous computing environments," *Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International*, IEEE, 2005, pp. 359-364.
- Petit, M. "Approche spatiale pour la caractérisation du contexte d'exécution d'un système d'information ubiquitaire," *Ecole nationale supérieure d'arts et métiers-ENSAM*, 2010.
- Pietiläinen, A.-K., Oliver, E., LeBrun, J., Varghese, G., and Diot, C. "MobiClique: middleware for mobile social networking," *Proceedings of the 2nd ACM workshop on Online social networks*, ACM, 2009, pp. 49-54.
- Poirier, D. "Des textes communautaires à la recommandation," *Université d'Orléans*, 2011.
- Poltash, N. A. "Snapchat and sexting: a snapshot of bearing your bare essentials," *Rich. JL & Tech.* (19) 2013, pp 14-14.
- Pouwelse, J. A., Garbacki, P., Wang, J., Bakker, A., Yang, J., Iosup, A., Epema, D. H., Reinders, M., Van Steen, M. R., and Sips, H. J. "TRIBLER: a social-based peer-to-peer system," *Concurrency and Computation: Practice and Experience* (20:2) 2008, pp 127-138.
- Pradel, C., Haemmerlé, O., and Hernandez, N. "Passage de la langue naturelle à une requête SPARQL dans le système SWIP," *IC-24èmes Journées francophones d'Ingénierie des Connaissances*, 2013.
- Quercia, D., Lathia, N., Calabrese, F., Di Lorenzo, G., and Crowcroft, J. "Recommending social events from mobile phone location data," *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, IEEE, 2010, pp. 971-976.
- Resnik, P. "Using information content to evaluate semantic similarity in a taxonomy," *arXiv preprint cmp-lg/9511007* 1995.
- Rheingold, H., and Lumbroso, L. *Les communautés virtuelles* Addison-Wesley France, 1995.
- Riel, M., and Polin, L. "Online learning communities: Common ground and critical differences in designing technical environments," *Designing for virtual communities in the service of learning* 2004, pp 16-50.
- Rote, G. "Computing the minimum Hausdorff distance between two point sets on a line under translation," *Information Processing Letters* (38:3) 1991, pp 123-127.
- Rouvoy, R., Barone, P., Ding, Y., Eliassen, F., Hallsteinsen, S., Lorenzo, J., Mamelli, A., and Scholz, U. "Music: Middleware support for self-adaptation in ubiquitous and service-oriented environments," in: *Software engineering for self-adaptive systems*, Springer, 2009, pp. 164-182.

Bibliographie

- Sancho, G. "Adaptation d'architectures logicielles collaboratives dans les environnements ubiquitaires. Contribution à l'interopérabilité par la sémantique," Université des Sciences Sociales-Toulouse I, 2010.
- Satyanarayanan, M. "Pervasive computing: Vision and challenges," *Personal Communications, IEEE* (8:4) 2001, pp 10-17.
- Schuster, D., Rosi, A., Mamei, M., Springer, T., Endler, M., and Zambonelli, F. "Pervasive social context: Taxonomy and survey," *ACM Transactions on Intelligent Systems and Technology (TIST)* (4:3) 2013, p 46.
- Slimani, T., BenYaghlane, B., and Mellouli, K. "Une extension de mesure de similarité entre les concepts d'une ontologie," International conference on sciences of electronic, technologies of information and telecommunications, 2007, pp. 1-10.
- Smailovic, V., and Podobnik, V. "Bfriend: Context-aware ad-hoc social networking for mobile users," MIPRO, 2012 Proceedings of the 35th International Convention, IEEE, 2012, pp. 612-617.
- Sure, Y., Bloehdorn, S., Haase, P., Hartmann, J., and Oberle, D. "The SWRC ontology–semantic web for research communities," in: *Progress in Artificial Intelligence*, Springer, 2005, pp. 218-231.
- Taconet, C. "Intergiciels pour la sensibilité au contexte en environnement ubiquitaire," *Mémoire HDR en informatique, Université d'Evry-Val-d'Essonne* 2011.
- Tigli, J.-Y., Lavirotte, S., Rey, G., Hourdin, V., Cheung-Foo-Wo, D., Callegari, E., and Riveill, M. "WComp middleware for ubiquitous computing: Aspects and composite event-based Web services," *annals of telecommunications-Annales des télécommunications* (64:3-4) 2009, pp 197-214.
- Tramp, S., Frischmuth, P., Ermilov, T., Shekarpour, S., and Auer, S. "An architecture of a distributed semantic social network," *Semantic Web* 2011.
- Wang, B., Bodily, J., and Gupta, S. K. "Supporting persistent social groups in ubiquitous computing environments using context-aware ephemeral group service," Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on, IEEE, 2004, pp. 287-296.
- Wang, F.-Y., Carley, K. M., Zeng, D., and Mao, W. "Social computing: From social informatics to social intelligence," *Intelligent Systems, IEEE* (22:2) 2007, pp 79-83.
- Wei, D., Wang, T., Wang, J., and Bernstein, A. "SAWSDL-iMatcher: A customizable and effective Semantic Web Service matchmaker," *Web Semantics: Science, Services and Agents on the World Wide Web* (9:4) 2011, pp 402-417.

Bibliographic

- Wei, E. J., and Chan, A. T. "CAMPUS: A middleware for automated context-aware adaptation decision making at run time," *Pervasive and Mobile Computing* (9:1) 2013, pp 35-56.
- Weiser, M. "Ubiquitous computing," ACM Conference on Computer Science, 1994, p. 418.
- Wenger, E. *Communities of practice: Learning, meaning, and identity* Cambridge university press, 1998.
- Wu, Z., and Palmer, M. "Verbs semantics and lexical selection," Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, 1994, pp. 133-138.
- Xu, W., Chow, C.-Y., and Zhang, J.-D. "CALBA: capacity-aware location-based advertising in temporary social networks," Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2013, pp. 354-363.
- Yeung, C.-m. A., Liccardi, I., Lu, K., Senevirante, O., and Berners-Lee, T. "Decentralization: The Future of Online Social Networking MSNWN Position Paper."
- Zhang, D., Guo, B., Li, B., and Yu, Z. "Extracting social and community intelligence from digital footprints: an emerging research area," in: *Ubiquitous Intelligence and Computing*, Springer, 2010, pp. 4-18.
- Zhdanova, A. V., Davies, M., Jorns, O., Stavroulaki, V., Demestichas, P., Gonzalez, M., Moessner, K., Carrez, F., Rajasekaran, H., and Iacono, L. L. "MOSS: Mobile Social Spaces," 1st International Workshop on Blending Physical and Digital Spaces on the Internet (OneSpace2008) at the 1st Future Internet Symposium (FIS'08), Vienna, Austria, 2008.
- Zheng, Y. "Tutorial on location-based social networks," Proceedings of the 21st international conference on World wide web, WWW, 2012.
- Zheng, Y., Xie, X., and Ma, W.-Y. "GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory," *IEEE Data Eng. Bull.* (33:2) 2010, pp 32-39.
- Zhou, J., Sun, J., Athukorala, K., Wijekoon, D., and Ylianttila, M. "Pervasive social computing: Augmenting five facets of human intelligence," *Journal of Ambient Intelligence and Humanized Computing* (3:2) 2012, pp 153-166.

Publications

Le travail lors de cette thèse a abouti aux publications suivantes :

Ghada Ben Nejma, Philippe Roose, Marc Dalmau, Jérôme Gensel – *Service Discovery for Spontaneous Communities in pervasive environment* – Special Session on Decentralized Social Networks – Web Information Systems Engineering (Wise 2015), November 1-3, **2015**, Miami, Florida (*accepted*).

Ghada Ben Nejma, Philippe Roose, Dalmau Marc, Jérôme Gensel, Ghorbali Mohamed Amine – *Design and development of semantic application for communities*– SMAP 2014 – 9th International Workshop on Semantic and Social Media Adaptation and Personalization – pp. 54-59, ISBN: 978-1-4799-6813-8 – DOI:10.1109/SMAP.2014.34 – IEEE Editor – November 6-7, **2014**, Corfu, Greece.

Ghada Ben Nejma, Philippe Roose, Marc Dalmau, Jérôme Gensel, Mohamed Amine Ghorbali, – *Taldea Demo: a Spontaneous Community-aware Application in Pervasive Environment* - MoWNet'2014 (2014 International Conference on Selected Topics in Mobile and Wireless Networking conference), Springer Ed. LNCS 9228 doi:10.1016/j.procs.2014.10.039 LNCS, September 8-9, **2014**, Rome, Italy.

Ghada Ben Nejma, Philippe Roose, Marc Dalmau, Jérôme Gensel – *A Semantic Social Software for Mobile Environment* – 1st International Workshop on Social Networks and Digital Ecosystems (SN&DE), 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, October 28-30, University of Technology of Compiègne, DOI 10.1109/3PGCIC, **2013**, Compiègne, France.

Ghada Ben Nejma, Philippe Roose, Marc Dalmau, Jérôme Gensel and Mohamed Amine Ghorbali – *Taldea: a Tool for Fostering Spontaneous Community* – 1st International Workshop on Semantic Social Network (SSN'13) held at 13th International Conference, ICWE 2013, July 8-12, **2013**, Aalborg, Denmark.

Ghada Ben Nejma, Philippe Roose, Jérôme Gensel, Marc Dalmau – *Taldea : une application communautaire avec géolocalisation* – INFORSID 2013 – pp. 165-180 – ISBN : 2-906855-28-6, Mai 29-31, **2013**, Paris, France.

Ghada Ben Nejma, Philippe Roose, Marc Dalmau, Jérôme Gensel – *Taldea, un outil d'aide à la création de communautés spontanées avec géolocalisation- Revue ISI – revue ISI*, vol. 18, n° 1/2013, *Social, localisation et mobilité* coordonné par T. Delot et F. Sèdes – janvier-février **2013**.

Ghada Ben Nejma, Philippe Roose, Jérôme Gensel and Marc Dalmau – *Communauté spontanée géolocalisée* – Short paper – INFORSID 2012 -ISBN 2-906855-27-8, pp. 299-308, 29-31, Mai 29-31, **2012**, Montpellier, France.