



HAL
open science

Découverte de contexte pour une adaptation automatique de services en intelligence ambiante

Yazid Benazzouz

► **To cite this version:**

Yazid Benazzouz. Découverte de contexte pour une adaptation automatique de services en intelligence ambiante. Autre. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2011. Français. NNT : 2011EMSE0617 . tel-00733013

HAL Id: tel-00733013

<https://theses.hal.science/tel-00733013>

Submitted on 17 Sep 2012

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.

NNT : 2011 EMSE 0617

THÈSE

présentée par

Yazid BENZAOUZ

pour obtenir le grade de
Docteur de l'École Nationale Supérieure des Mines de Saint-Étienne

Spécialité : INFORMATIQUE

Découverte de contexte pour une adaptation automatique de services en
intelligence ambiante

soutenue à Grenoble, le 26 Août 2011

Membres du jury

Président :	Jérôme GENSEL	Professeur Univ. Pierre Mendès France, Grenoble
Rapporteurs :	Frédérique LAFOREST	HDR, MC INSA Lyon
	Patrick REIGNIER	HDR, MC Université Joseph Fourier, Grenoble
Examineur(s) :	Pierre MARET	Professeur Université Jean Monnet
	Sylvain GIROUX	Professeur Université Sherbrooke Québec, Canada
Directeur(s) de thèse :	Olivier BOISSIER	Professeur ENS Mines Saint-Etienne
Co-encadrant(s) de thèse :	Philippe BEAUNE	MA ENS Mines Saint-Etienne
	Fano RAMPARANY	Expert R&D, Orange-Labs Grenoble
Invité :	Nicolas SABOURET	HDR MC Univ. Pierre et Marie Curie, Paris

Spécialités doctorales :

SCIENCES ET GENIE DES MATERIAUX
 MECANIQUE ET INGENIERIE
 GENIE DES PROCEDES
 SCIENCES DE LA TERRE
 SCIENCES ET GENIE DE L'ENVIRONNEMENT
 MATHEMATIQUES APPLIQUEES
 INFORMATIQUE
 IMAGE, VISION, SIGNAL
 GENIE INDUSTRIEL
 MICROELECTRONIQUE

Responsables :

J. DRIVER Directeur de recherche – Centre SMS
 A. VAUTRIN Professeur – Centre SMS
 F. GRUY Professeur – Centre SPIN
 B. GUY Maître de recherche – Centre SPIN
 J. BOURGOIS Professeur – Fayol
 E. TOUBOUL Ingénieur – Fayol
 O. BOISSIER Professeur – Fayol
 JC. PINOLI Professeur – Centre CIS
 P. BURLAT Professeur – Fayol
 Ph. COLLOT Professeur – Centre CMP

Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

AVRIL	Stéphane	MA	Mécanique & Ingénierie	CIS
BATTON-HUBERT	Mireille	MA	Sciences & Génie de l'Environnement	Fayol
BENABEN	Patrick	PR 1	Sciences & Génie des Matériaux	CMP
BERNACHE-ASSOLLANT	Didier	PR 0	Génie des Procédés	CIS
BIGOT	Jean-Pierre	MR	Génie des Procédés	SPIN
BILAL	Essaïd	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR 1	Informatique	Fayol
BORBELY	Andras	MR	Sciences et Génie des Matériaux	SMS
BOUCHER	Xavier	MA	Génie Industriel	Fayol
BOUDAREL	Marie-Reine	PR 2	Génie Industriel	DF
BOURGOIS	Jacques	PR 0	Sciences & Génie de l'Environnement	Fayol
BRODHAG	Christian	DR	Sciences & Génie de l'Environnement	Fayol
BURLAT	Patrick	PR 2	Génie industriel	Fayol
COLLOT	Philippe	PR 1	Microélectronique	CMP
COURNIL	Michel	PR 0	Génie des Procédés	SPIN
DAUZERE-PERES	Stéphane	PR 1	Génie industriel	CMP
DARRIEULAT	Michel	IGM	Sciences & Génie des Matériaux	SMS
DECHOMETS	Roland	PR 1	Sciences & Génie de l'Environnement	Fayol
DESRAYAUD	Christophe	MA	Mécanique & Ingénierie	SMS
DELAFOSSÉ	David	PR 1	Sciences & Génie des Matériaux	SMS
DOLGUI	Alexandre	PR 1	Génie Industriel	Fayol
DRAPIER	Sylvain	PR 2	Mécanique & Ingénierie	SMS
DRIVER	Julian	DR 0	Sciences & Génie des Matériaux	SMS
FEILLET	Dominique	PR 2	Génie Industriel	CMP
FOREST	Bernard	PR 1	Sciences & Génie des Matériaux	CIS
FORMISYN	Pascal	PR 1	Sciences & Génie de l'Environnement	Fayol
FRACZKIEWICZ	Anna	DR	Sciences & Génie des Matériaux	SMS
GARCIA	Daniel	MR	Génie des Procédés	SPIN
GIRARDOT	Jean-Jacques	MR	Informatique	Fayol
GOEURIOT	Dominique	MR	Sciences & Génie des Matériaux	SMS
GRAILLOT	Didier	DR	Sciences & Génie de l'Environnement	Fayol
GROSSEAU	Philippe	MR	Génie des Procédés	SPIN
GRUY	Frédéric	MR	Génie des Procédés	SPIN
GUY	Bernard	MR	Sciences de la Terre	SPIN
GUYONNET	René	DR	Génie des Procédés	SPIN
HERRI	Jean-Michel	PR 2	Génie des Procédés	SPIN
INAL	Karim	PR 2	Microélectronique	CMP
KLÖCKER	Helmut	DR	Sciences & Génie des Matériaux	SMS
LAFOREST	Valérie	CR	Sciences & Génie de l'Environnement	Fayol
LERICHE	Rodolphe	CR CNRS	Mécanique et Ingénierie	SMS
LI	Jean-Michel	EC (CCI MP)	Microélectronique	CMP
MALLIARAS	George Grégory	PR 1	Microélectronique	CMP
MOLIMARD	Jérôme	MA	Mécanique et Ingénierie	SMS
MONTHEILLET	Frank	DR 1 CNRS	Sciences & Génie des Matériaux	SMS
PERIER-CAMBY	Laurent	PR 2	Génie des Procédés	SPIN
PIJOLAT	Christophe	PR 1	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR 1	Génie des Procédés	SPIN
PINOLI	Jean-Charles	PR 0	Image, Vision, Signal	CIS
STOLARZ	Jacques	CR	Sciences & Génie des Matériaux	SMS
SZAFNICKI	Konrad	MR	Sciences & Génie de l'Environnement	Fayol
THOMAS	Gérard	PR 0	Génie des Procédés	SPIN
TRIA	Assia		Microélectronique	CMP
VALDIVIESO	François	MA	Sciences & Génie des Matériaux	SMS
VIRICELLE	Jean-Paul	MR	Génie des procédés	SPIN
WOLSKI	Krzysztof	DR	Sciences & Génie des Matériaux	SMS
XIE	Xiaolan	PR 1	Génie industriel	CIS

Glossaire :

PR 0	Professeur classe exceptionnelle
PR 1	Professeur 1 ^{ère} classe
PR 2	Professeur 2 ^{ème} classe
MA(MDC)	Maître assistant
DR	Directeur de recherche
Ing.	Ingénieur
MR(DR2)	Maître de recherche
CR	Chargé de recherche
EC	Enseignant-chercheur
IGM	Ingénieur général des mines

Centres :

SMS	Sciences des Matériaux et des Structures
SPIN	Sciences des Processus Industriels et Naturels
Fayol	Institut Henri Fayol
CMP	Centre de Microélectronique de Provence
CIS	Centre Ingénierie et Santé

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2011 EMSE 0617

Prénom NOM : Yazid BENAZZOUZ

DISSERTATION TITLE : Context discovery for the automatic adaptation of services in ambient intelligence

Speciality : Computer Science

Keywords : Ambient intelligence, context-awareness, adaptive services, dynamic adaptation of services, context learning, assisted living, intelligent environment.

Abstract :

This thesis addresses the problem of dynamic adaptation of services in the context of ambient intelligence applications. Literature study shows how context-awareness plays a central role in the design and implementation of adaptive services. However, its use is still limited to elementary descriptions and predefined situational models. Dynamic adaptation should be capable of following user habits to yield dynamic answers to environmental change, and to support heterogeneous sources of context. To this end, we propose mechanisms to discover contexts and situations that trigger adaptation. These mechanisms rely on data mining techniques, and are integrated within an architecture for dynamic adaptation of services. This work was carried out and applied to ambient intelligence projects for the elderly, providing support and assistance in their daily lives, particularly in the context of the ITEA-MIDAS project.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2011 EMSE 0617

Prénom NOM : Yazid BENAZZOUZ

TITRE DE LA THÈSE : Découverte de contexte pour une adaptation
automatique de services en intelligence ambiante

Spécialité: Informatique

Mots clefs : Intelligence ambiante, sensibilité au contexte, service adaptatif, adaptation dynamique de services, apprentissage du contexte, maintien à domicile, environnement intelligent.

Résumé :

Cette thèse s'intéresse à la problématique de l'adaptation automatique de services dans le domaine de l'intelligence ambiante. L'étude de la littérature montre que la sensibilité au contexte est devenue un élément central pour la conception et la mise en place de services adaptatifs. Cependant, sa prise en compte se limite généralement à des descriptions élémentaires de situations ou à des modèles prédéfinis. Afin de permettre une adaptation aux changements d'habitudes des utilisateurs, à la dynamique de l'environnement et à l'hétérogénéité des sources de perception, nous proposons des mécanismes de découverte de contexte et de situations déclencheurs d'adaptation. Ces mécanismes s'appuient sur des techniques de fouille de données et sont intégrés au sein d'une architecture d'adaptation automatique de services. Ces travaux ont été réalisés et appliqués à des projets d'intelligence ambiante pour de l'assistance à des personnes et plus particulièrement dans le cadre du projet ITEA- MIDAS.

TABLE DES MATIÈRES

TABLE DES MATIÈRES	i
LISTE DES TABLEAUX	vii
LISTE DES FIGURES	ix
DÉDICACE	xiii
REMERCIEMENTS	xv
INTRODUCTION	1
1 Problématique	1
2 Objectif de la thèse	3
3 Structure du manuscrit	4
I État de l’art	7
CHAPITRE 1 : ADAPTATION DE SERVICES	9
1 Déclinaisons de la notion d’adaptation	9
1.1 Adaptable vs. Adaptatif	9
1.2 Personnalisation	11
1.3 Recommandation	12
1.4 Reconfiguration	13
1.5 Relations entre ces concepts	14
2 Motivations pour l’adaptation de services	15
2.1 Assurer la continuité des services à l’utilisateur	16
2.2 Améliorer l’interaction de l’utilisateur avec son environnement	16
3 Mécanismes d’adaptation de services	17
3.1 Adaptation à base de règles	18

3.2	Adaptation à base d'apprentissage automatique	19
3.3	Adaptation à base de comparaison	22
3.4	Adaptation à base de fouille de données	22
4	Défis soulevés par l'adaptation de services	23
4.1	Prise en compte du contexte	23
4.2	Gestion de volumes de données	23
4.3	Représentation de la sémantique	24
4.4	Extraction de connaissances	24
4.5	Traitement en temps réel	25
4.6	Ouverture et flexibilité	25
5	Conclusion	25
CHAPITRE 2 : CONTEXTE		27
1	Vue d'ensemble de la notion de contexte	27
1.1	Définition du contexte	28
1.2	Contexte et Temps	30
1.3	Contexte et Interaction	30
1.4	Contexte et Pertinence	30
2	Obtention du contexte	31
3	Description du contexte	33
3.1	Domaine de définition	33
3.2	Propriétés	34
3.3	Niveaux de description	34
4	Représentation du contexte	35
4.1	Modèles conceptuels	36
4.2	Modèles logiques	37
5	Traitement du contexte	42
5.1	Raisonnement	42
5.2	Fusion de données	43

5.3	Reconnaissance	45
5.4	Prédiction	47
5.5	Découverte	48
6	Relation entre description et traitement du contexte	50
7	Synthèse	50
CHAPITRE 3 : SENSIBILITÉ AU CONTEXTE & ADAPTATION		55
1	Contexte et adaptation	55
2	Services sensibles au contexte	57
2.1	Services de contenu	57
2.2	Services de contrôle	57
2.3	Services de communication	58
2.4	Services de commerce électronique	58
3	Limites de l'adaptation des services au contexte	59
3.1	Limites via-à-vis du contexte	61
3.2	Historique du contexte	62
4	Besoin d'une approche de découverte du contexte	63
5	Synthèse	64
II Proposition		69
CHAPITRE 4 : MODÉLISATION DU CONTEXTE		71
1	Modèle sémantique d'une donnée de contexte	71
1.1	Modèle d'une donnée de contexte	72
1.2	Propriétés d'une donnée de contexte	74
1.3	Fonctions de manipulation des données de contexte	77
1.4	Vecteur d'une donnée de contexte	79
1.5	Données statiques du contexte	79
2	Modèle sémantique du contexte	80
2.1	Modèle du contexte	80

2.2	Contexte pur	81
2.3	Vecteur d'un contexte	82
3	Mesures de similarité	82
3.1	Similarité entre données de contexte	83
3.2	Similarité entre contextes purs	92
3.3	Combinaison des mesures de similarité	97
4	Conclusion	98
CHAPITRE 5 : DÉCOUVERTE DU CONTEXTE		99
1	Fondements de la fouille de données	99
2	Différentes approches de découverte de contextes	100
3	Approche ascendante de découverte du contexte	103
3.1	Découverte guidée de situations	103
3.2	Découverte aveugle de situations	109
4	Approche descendante	112
4.1	Modélisation des situations à l'aide d'ontologies	114
4.2	Algorithme de l'approche descendante	115
5	Conclusion	118
CHAPITRE 6 : ARCHITECTURE D'ADAPTATION DE SERVICES 121		
1	Architecture générale	121
2	Acquisition des données de contexte à partir de capteurs	123
2.1	Formes d'acquisition des données de contexte	124
2.2	Notre choix technique	127
3	Archivage des données du contexte	130
3.1	Solutions de stockage existantes	130
3.2	Notre choix technique	131
4	Découverte du contexte	133
5	Vers une adaptation de service	133
5.1	Pertinence d'une situation fréquente	135

5.2	Détection d'une dérive	137
5.3	Détection d'une correction utilisateur	141
6	Conclusion	142

III Application et validation 145

CHAPITRE 7 : CADRE EXPÉRIMENTAL 147

1	Découverte de situations fréquentes	147
1.1	Description du lieu des expérimentations	148
1.2	Expérimentation sur un volume réduit de données	148
1.3	Expérimentation sur un volume important de données	151
2	Découverte hiérarchique de situations	153
2.1	Description du lieu des expérimentations	154
2.2	Infrastructure réseau et matérielle	154
2.3	Description du scénario	155
2.4	Résultat de l'algorithme hiérarchique	157
3	Modélisation des situations à l'aide d'une ontologie	160
4	Autres travaux d'analyse	160
4.1	Partitionner un large graphe	161
4.2	Fouille visuelle des données de contexte	163
4.3	Regroupement par transformation des données de contexte	166
5	Conclusion	167

CHAPITRE 8 : IMPLEMENTATION 169

1	Aperçu général du cadre d'implémentation	169
2	Mise en oeuvre	171
2.1	Description de l'ontologie	171
2.2	Acquisition	175
2.3	Stockage des données de contexte	177
2.4	Description des données collectées	177

2.5	Découverte de situations fréquentes	179
2.6	Identification d'une situation inhabituelle et adaptation . .	182
3	Conclusion	183

IV Conclusion 185

CONCLUSION 187

4	Contribution	188
5	Bilan	190
6	Perspectives	191

BIBLIOGRAPHIE 193

LISTE DES TABLEAUX

4.I	Liste exhaustive des triplets du modèle \mathcal{M}	73
4.II	Les 5 premiers éléments des vecteurs des données de la figure 4.4	85
4.III	Calcul de la distance syntaxique	86
4.IV	Mesures conceptuelles pour locationData	89
4.V	Mesures conceptuelles pour statusData	89
4.VI	Calcul de la similarité conceptuelle de Zhong	90
4.VII	Calcul de la similarité conceptuelle de WuPalmer	91
4.VIII	Exemple de problème posé par TFIDF	96
6.I	Requête SPARQL pour trouver un service	138
7.I	Evaluation du temps d'exécution	153
7.II	Liste des capteurs par zone	157
7.III	Échantillon des données de contexte	159
7.IV	Les clusters formés par l'algorithme	160
7.V	Échantillon de données	162

LISTE DES FIGURES

1.1	Relation entre les concepts d'adaptation	15
2.1	Représentation des niveaux de description	41
2.2	Une méthode de reconnaissance du contexte	46
2.3	Relation entre description et traitement du contexte	51
3.1	Analyse de services sensibles au contexte	60
3.2	Relations entre techniques et niveaux d'interprétation	64
4.1	Modèle des données de contexte	74
4.2	Partie de la structure de l'ontologie \mathcal{O}	77
4.3	Exemple d'une donnée de contexte	78
4.4	Exemple d'un contexte	81
4.5	Exemple de données de contexte	86
4.6	Partie de la hiérarchie de l'ontologie	87
4.7	TFIDF pour le calcul de la similarité entre contextes purs	95
5.1	Approches de découverte du contexte	102
5.2	Exemple abstrait de contextes purs et de contextes fréquents	105
5.3	Exemple de clustering HAC	108
5.4	Intégration de la relativité dans la localisation et dans le temps	113
5.5	Vue partielle de l'ontologie	114
5.6	Exemple de localisation relative	116
5.7	Exemple de temps relatif	117
6.1	Notre architecture d'adaptation automatique de services	123
6.2	Exemple d'architecture employant des capteurs portés	125
6.3	Exemple d'architecture d'acquisition - PlaceLab	126
6.4	L'acquisition du contexte selon Amigo	128
6.5	Acquisition d'une donnée de contexte	129

6.6	Archivage des données de contexte	132
6.7	Découverte des situations fréquentes	134
6.8	Description dans l'ontologie d'une donnée de contexte	136
6.9	Une partie de l'ontologie décrivant les services de contrôle d'une lampe	137
6.10	Exemple de situation pertinente	139
6.11	Exemple de situation courante	140
7.1	Environnement	149
7.2	Scénario	149
7.3	Construction des situations	151
7.4	dispersion des situations	152
7.5	Résultats du clustering	152
7.6	Infrastructure réseaux du Labo Domus	155
7.7	Une portion du dendrogramme obtenu par agglomerative clustering	159
7.8	Exemple de modélisation d'une situation	161
7.9	Exemple de partitionnement d'un large graphe de données de contexte	163
7.10	Exemple de visualisation de données de contexte avec NetDraw .	165
7.11	Un échantillon de données WireHumidity de PlaceLab	166
7.12	Résultat de clustering avec EM	167
8.1	Architecture de gestion du contexte dans le projet MIDAS	172
8.2	Partie de la structure de l'ontologie	173
8.3	Exemple d'instances de concepts	174
8.4	Exemple d'une instance d'une donnée de contexte	175
8.5	Exemple d'un service de contrôle de lampe	176
8.6	Acquisition des événements capteurs et dispositifs	177
8.7	Représentation interne d'un événement capteur	178
8.8	Modèle logique de stockage JENA dans une Base de données MYSQL	179
8.9	Données expérimentales correspondant au scénario	180
8.10	Situations construites par l'algorithme de découverte	182

8.11 Situations fréquentes trouvées par l'algorithme de découverte . . . 183

Je dédie ce travail de thèse à ma femme Meriem et mon enfant Midou pour le soutien et la patience durant toutes ces années en vue de l'obtention de cette thèse. Je ne saurai jamais suffisamment les remercier pour avoir supporté mon éloignement et ma femme en particulier d'avoir pris la responsabilité d'élever notre fils. Je dédie également ce travail à toute sa famille pour avoir été près d'eux durant mes années d'absences.

REMERCIEMENTS

Je tiens tout d'abord à exprimer ma profonde gratitude ainsi que mes sincères remerciements à mes encadrants de thèse : Philippe Beaune et Olivier Boissier de l'École des Mines de Saint-Etienne et Fano Ramparany d'Orange-Labs de Grenoble, pour le soutien, l'aide, et la participation dans l'accomplissement de ce travail de thèse.

Je souhaite adresser également mes vifs remerciements aux membres du jury qui ont accepté de participer à ma soutenance, en particulier aux rapporteurs pour leurs efforts de relecture et leurs réflexions, et, enfin, à Nicolas Sabouret pour avoir accepté de participer à ce jury.

J'exprime ma reconnaissance à l'École Nationale Supérieure des Mines de Saint-Etienne et au laboratoire Orange-Labs de Grenoble d'avoir mis tous les moyens nécessaires à la réussite de ce partenariat en m'accueillant dans leurs laboratoires et me permettant d'accomplir au mieux mon travail de thèse.

Je remercie également la région Rhône-Alpes d'avoir financé mon séjour au Canada, et le laboratoire Domus (Sherbrooke, Canada) pour m'avoir accueilli pendant 6 mois au sein de son équipe de recherche, et permis ainsi, d'approfondir ma connaissance de l'informatique diffuse.

En dernier, mes remerciements aux personnes qui m'ont aidé de près ou de loin à réaliser cette étude.

INTRODUCTION

*"Les espèces qui survivent ne sont pas les plus intelligentes
ni les plus fortes, mais celles capables de s'adapter."*

Charles Darwin

1 Problématique

Ces dernières années, les développements des technologies de communication et des matériels ont conduit au déploiement de nouveaux services, d'objets communicants au sein de notre environnement. Ceux-ci apportent aide et commodité dans l'accomplissement des activités quotidiennes des usagers dans différents domaines tels que, par exemple, divertissement, communication, santé. Ces avancées technologiques se retrouvent également au sein de secteurs en expansion tels que la domotique, l'assistance et le maintien à domicile, la gestion de l'énergie dans les bâtiments publics, la ville intelligente,

Dans ce paysage, l'adaptation au sein de tels systèmes devient un volet central. Mettre en oeuvre une telle adaptation est loin d'être une tâche facile. De nombreuses questions de recherche doivent être résolues avant de déboucher vers une adaptation automatique.

Dans notre travail nous nous intéressons particulièrement à l'adaptation des services dans les environnements intelligents. Un environnement intelligent ¹ est un espace limité (la chambre, la rue, un immeuble, voire une cité), où sont présents divers capteurs et effecteurs. Ces derniers sont principalement utilisés pour fournir des informations et un soutien matériel aux habitants de cet espace.

La notion de service s'est fortement développée sur le Web où les services web sont les composants de base de tout traitement dans une architecture orientée services. Leur principe repose sur la description, la publication, la recherche et l'invocation. Ils apportent modularité, évolutivité, composabilité et réutilisabilité.

¹<http://www.ispace-labnet.org>

Les services pour les environnements intelligents sont généralement considérées comme des extensions des Services Web existants (par exemple, la messagerie, les services de localisation, la gestion des périphériques, les applications multimédias, ...) tout en reprenant les principes de base. Deux catégories de services peuvent être distinguées : les services à l'utilisateur et les services logiciels. Les services à l'utilisateur offrent des fonctionnalités qui par leur utilisation facilitent ou aident l'utilisateur dans l'accomplissement de ses tâches. L'adaptation de tels services concerne essentiellement la prise en compte des habitudes utilisateurs, les changements de l'environnement dans lequel évoluent ces utilisateurs. Les services logiciels quant à eux sont des programmes informatiques permettant la communication et l'échange de données entre applications dans des environnements distribués. L'adaptation pour de tels services concerne l'amélioration du processus d'exécution des services (planification, développement et maintenance).

Dans ce travail, la notion d'adaptation des services est étudiée par rapport à une vision centrée utilisateur. Dans la suite de ce document, le terme service désigne les services à l'utilisateur. L'adaptation tient donc compte essentiellement des évolutions de l'utilisateur et de son environnement et non des conditions d'exécution des services eux-mêmes.

Afin d'aborder plus particulièrement cette problématique d'adaptation, il est nécessaire de considérer la notion de contexte. Avec les travaux en lien avec la sensibilité au contexte (context-awareness), le contexte concerne l'ensemble des informations nécessaires à la description de la situation déclenchant l'adaptation. Les recherches récentes en informatique ubiquitaire y intègrent l'environnement du ou des utilisateurs/systèmes en considérant principalement des informations issues de capteurs (e.g. contrôle gestuel de la télé en se servant d'une caméra, réglage anticipé du chauffage avant l'arrivée de la personne chez elle). Allant au-delà de la localisation, considérée comme donnée principale de contexte au début des travaux sur cette thématique, le contexte s'est élargi pour inclure également les connaissances sur l'utilisateur du service, ses habitudes de vie, ses activités et son

environnement. Ainsi, les services traditionnellement très peu informés de leur entourage sont devenus plus conscients de l'environnement dans lequel ils évoluent et aussi de l'environnement du demandeur de services. Par ce biais, ils ont renforcé la prise de décision des systèmes ; ils ont permis une meilleure connectivité et ils sont devenus capables de se reconfigurer de manière automatique comme par exemple le choix de la meilleure bande de fréquence afin d'assurer un service optimal à l'utilisateur.

2 Objectif de la thèse

Dans cette thèse nous nous intéressons à la problématique de l'adaptation au contexte des services dans un environnement intelligent. Notre objectif est de proposer un ensemble de méthodes pour la mise en place d'un processus de découverte de contextes qui peuvent correspondre à des situations, des modèles, des intentions utilisateurs ou simplement des corrélations entre les données acquises sur l'utilisateur et son environnement. Ce processus de découverte de contextes s'intègre dans une architecture visant l'adaptation automatique des services.

En nous appuyant sur des méthodes de fouille de données intégrées au sein d'une architecture logicielle de gestion et de découverte du contexte, il s'agit de traiter la complexité d'adaptation des services due principalement à la dynamique de l'environnement, au changement dans les usages de ces services, l'hétérogénéité et le volume important des données qui constituent le contexte de ces services.

L'approche technique envisagée pour traiter cette problématique consiste à analyser un historique des données collectées au cours du temps sur l'utilisateur et son environnement puis utiliser des techniques de fouille de données pour identifier des motifs afin de construire des modèles d'adaptation. Nous proposons deux approches pour exploiter ces historiques de données : une approche ascendante et une approche descendante. L'approche ascendante s'intéresse à la découverte de situations sans connaissance a priori de la situation à trouver. Elle peut être soit

une découverte aveugle sur les données de contexte, soit une découverte guidée par un type particulier de situation à découvrir. Dans ce dernier cas, la prise en compte de la temporalité peut être faite séparément des autres types d'informations disponibles, ou au moyen de mesures de proximité temporelle. L'approche descendante repose sur une ontologie du contexte, que nous avons développée, et que nous utilisons comme moyen de description des situations à regrouper.

Au final, nous contribuons au développement d'outils informatiques permettant de mettre en oeuvre l'architecture finale de la solution proposée et nous la validons sur des scénarios d'usage dans deux projets partenaires d'Orange-Labs.

3 Structure du manuscrit

Nous avons organisé ce matériel en trois parties : état de l'art permettant de mettre en évidence les différentes problématiques de notre thèse, introduction et contributions de la thèse défendue, évaluation et validation de notre proposition sur les scénarios d'usage retenus.

Partie I : État de l'art

Le premier chapitre de cette partie précise et définit ce que nous considérons dans ce document lorsque nous parlons d'adaptation de services. Il présente et discute diverses questions relatives à l'adaptation automatique de services à l'utilisateur dans les environnements intelligents. Après une présentation d'une vue d'ensemble de la notion d'adaptation, nous introduisons les motivations à la base de cette problématique d'adaptation de services dans ce contexte d'intelligence ambiante. Nous présentons ensuite les mécanismes existants à la base de cette adaptation et concluons ce chapitre en présentant les défis qui restent à relever pour une adaptation de services centrée utilisateur.

Le chapitre 2 aborde la notion du contexte en intelligence ambiante. Le contexte est une notion importante dans le processus d'adaptation de services à l'utilisateur

pour caractériser les informations sur l'utilisateur et son environnement. Dans un premier temps nous étudions dans la littérature la manière dont le contexte est modélisé et représenté. A partir de cette première caractérisation, nous étudions la manière dont il est interprété en nous intéressant plus particulièrement aux mécanismes et méthodes à la base de cette interprétation.

Le troisième et dernier chapitre de cette partie discute les limites actuelles de l'adaptation des services au regard des différentes approches de modélisation et d'interprétation du contexte. Nous analysons plus particulièrement la relation entre données de contexte et la connaissance nécessaire à l'adaptation des services et mettons ainsi en avant le besoin d'une approche de découverte du contexte pour l'adaptation automatique des services sur la base des techniques de fouille de données.

Partie II : Contribution

Le premier chapitre de notre contribution (Chapitre 4) décrit la modélisation sémantique des informations sur lesquelles s'appuie notre système de découverte de contexte. Nous décrivons ainsi le contexte en identifiant d'une part les données de contexte et d'autre part les situations.

Le chapitre suivant (Chapitre 5) s'intéresse à l'utilisation des méthodes de fouille de données, particulièrement celles du regroupement de données pour la découverte automatique du contexte. Nous proposons deux approches : une approche ascendante et une approche descendante. Dans l'approche ascendante, nous ne considérons aucune connaissance sur les situations à découvrir. Nous utilisons et nous comparons plusieurs algorithmes de fouille de données reposant sur des mesures de similarité que nous avons adaptées aux caractéristiques et représentations des données de contexte. Nous développons également une méthode pour la découverte des situations fréquentes. Dans l'approche descendante, nous explorons l'idée d'utiliser des ontologies comme moyen de description des situations à découvrir. La méthode décrite est fondée sur la corrélation entre la

description de la situation dans l'ontologie et les regroupements construits à partir des données de contexte. Puis nous discutons ces deux approches pour déterminer la technique de découverte qui sera retenue pour être implémentée et intégrée dans le système d'adaptation automatique de services que nous avons développé.

Le dernier chapitre de cette partie (Chapitre 6) couvre la partie architecture de notre système pour l'apprentissage de l'adaptation au contexte des services. Nous présentons une architecture modulaire en expliquant et justifiant les choix techniques adoptés puis nous détaillons le mécanisme d'apprentissage de l'adaptation au contexte.

Partie III : Applications et validation

Le chapitre 7 décrit le cadre expérimental utilisé pour la validation de notre proposition. Nous détaillons particulièrement deux expérimentations sur la découverte des situations fréquentes et la découverte hiérarchique de situations. Nous montrons ensuite un exemple de modélisation de situation à l'aide d'ontologie suivi d'une description de quelques expérimentations d'algorithmes de regroupement. Ces derniers nous ont aidés dans les choix d'implémentation de notre système.

Le chapitre 8 vient consolider les travaux que nous avons présentés dans les chapitres précédents. Nous faisons état de notre participation au projet ITEA Midas coordonné par Orange-Labs et nous présentons une description du système global du projet au développement duquel nous avons contribué, et spécifiquement dans la définition de scénarios ainsi que dans les parties acquisition, stockage et découverte de situations fréquentes.

Enfin, nous terminons ce document en synthétisant les différentes contributions que nous avons proposées étant donné les objectifs que nous nous étions fixés dans ce travail. Après une analyse et un retour sur les expérimentations réalisées, nous discutons différentes perspectives de recherche que soulève ce travail.

Première partie

État de l'art

CHAPITRE 1

ADAPTATION DE SERVICES

Ce chapitre précise et définit ce que nous considérons dans ce document lorsque nous parlons d'adaptation de services. La section 1 présente une vue d'ensemble de la notion d'adaptation dans les environnements intelligents. La section 2 introduit les motivations à la base de cette problématique d'adaptation de services dans ce contexte d'intelligence ambiante. En section 3, nous présentons les mécanismes existants à la base de cette adaptation et concluons ce chapitre en présentant les défis qui restent à relever pour une adaptation de services centrée utilisateur.

1 Déclinaisons de la notion d'adaptation

Cette section détaille la notion d'adaptation selon les points de vue les plus notables en littérature. Plusieurs concepts s'accrochent à cette notion et il nous semble nécessaire de les présenter pour mieux distinguer et cadrer toute proposition d'une solution d'adaptation automatique de services. Ces concepts sont l'adaptation, la personnalisation, la recommandation et la reconfiguration de services. Nous présentons à la fin de cette section une synthèse graphique qui illustre les relations entre ces différents concepts.

1.1 Adaptable vs. Adaptatif

R. Oppermann (1994) [114] distingue les systèmes *adaptatifs* des systèmes *adaptables*. Il appelle *système adaptable* les systèmes qui permettent à l'utilisateur de modifier certains paramètres de leurs comportements. A l'opposé, il appelle *système adaptatif* les systèmes qui s'adaptent aux utilisateurs automatiquement en fonction des besoins présumés de ces utilisateurs.

Dans ce qui suit, nous allons nous intéresser plus particulièrement à ce second

type d'adaptation. Tout d'abord, il est à noter que les systèmes qui ont un fonctionnement autonome n'emploient pas tous des procédés adaptatifs. Par exemple le système de réception d'un téléphone mobile est autonome parce qu'il est capable de sonner dès la réception d'un appel. Par contre, ce système n'est pas adaptatif car il est incapable de s'adapter selon la situation ¹ de l'utilisateur. Dans le cas d'un système adaptatif, la réception d'un appel pourrait être plus discrète. Si l'utilisateur est en réunion, une icône clignotante s'affiche sur l'écran de l'ordinateur portable en face de l'utilisateur, indiquant un appel entrant. Si l'utilisateur est à la maison en train de regarder la télé alors que le téléphone est dans la chambre et le son de la télé est élevé, l'icône de réception d'appel est alors affichée en face de la personne sur la télé.

Ensuite, l'environnement système et l'environnement utilisateur sont les seules sources d'adaptation. Le premier désigne le cadre matériel, les services logiciels, applications et réseaux dans lesquels fonctionne le système. Le second désigne les conditions de vie de l'utilisateur et les rendus des services, applications ou systèmes.

Ceci dit, selon S. Saowanee (2008) [119], il existe deux procédés d'adaptation de services : l'*autoadaptation* et l'*adaptation contrôlée* par l'utilisateur. Dans le processus d'autoadaptation, le système s'adapte sans aucune interaction entre l'utilisateur et le système. Le processus d'adaptation contrôlée quant à lui, laisse à l'utilisateur le soin de prendre la décision, le système automatisant le reste du changement. Selon l'auteur, l'autoadaptation ne fonctionne que dans un nombre de cas limités, qui se caractérisent par une extrême simplicité à cause de la faible expressivité des langages de description des situations nécessaires à l'adaptation. L'auteur ajoute que même si un service peut s'adapter parfaitement à la hauteur des attentes des utilisateurs, une telle solution est inacceptable pour l'utilisateur, pour des raisons psychologiques. L'utilisateur peut penser que le service n'est pas

¹Dans ce chapitre, le terme *situation* est utilisé dans son sens commun et sera défini au chapitre 2 section 3.3

fiable, à cause de l'absence de pouvoir sur le service et le manque de compréhension des situations où il se sent hors de contrôle, ignorant ce qui s'est passé ou pourquoi cela a conduit à une expérience négative. B. Meyer (1994) [15] débouche sur une conclusion similaire en disant qu'une autoadaptation peut ne pas être efficace, surtout lorsque l'adaptation ne change pas les actions disponibles à l'utilisateur.

Bien que l'autoadaptation donne plus d'autonomie aux systèmes pour préserver l'intelligence des espaces, l'adaptation contrôlée permet plus de souplesse et de liberté aux interactions utilisateurs dans l'environnement. Toutefois, nous pensons que l'autoadaptation peut couvrir l'adaptation contrôlée si les corrections de l'utilisateur sur les décisions prises par le système sont prises en compte automatiquement dans le processus d'autoadaptation future.

1.2 Personnalisation

La personnalisation selon McBurney *et al.* (2007) [94] est une forme d'adaptation. La personnalisation vise à adapter les fonctionnalités et le comportement du système afin qu'il réagisse différemment en fonction du contexte utilisateur et de ses préférences personnelles. Par exemple, dans [107], les auteurs appliquent un filtrage sur un contenu d'informations pour sélectionner celles qui se réfèrent à un lieu particulier de l'utilisateur.

Selon McBurney *et al.* (2008) [120], la personnalisation assure l'adaptabilité du système et facilite la prise en compte des besoins utilisateur. La personnalisation conduit à une adaptation par maintien et construction implicites des préférences utilisateurs en utilisant des mécanismes de surveillance et d'apprentissage appliqués au nom de l'utilisateur. Papadopoulou *et al.* (2008) [45] distinguent la personnalisation comme étant le processus de création, de maintien et d'application des préférences utilisateurs pour la prise de décision. Par contre, si la personnalisation est combinée avec la prise en compte de l'environnement utilisateur, la décision sera probablement affectée à cause d'un changement de l'environnement

qui induira un changement des préférences utilisateur. De ce fait, la personnalisation se fait dynamiquement en accord avec l'évolution et la mise à jour des préférences utilisateur et selon l'état de son environnement.

1.3 Recommandation

Les systèmes de recommandation sont des mécanismes particuliers de personnalisation. Ils ont pour objectif d'aider l'utilisateur à trouver des informations ou services intéressants [109]. L'idée principale est d'enregistrer les préférences utilisateur afin de proposer des produits ou services supposés intéresser l'utilisateur. L'adaptation dans ce cas de figure est vue comme une anticipation sur les choix de l'utilisateur.

La notion de recommandation est souvent ambiguë dans la littérature. En effet, comme le montrent les travaux décrits ci-dessous, ceci est dû au manque d'une explicitation claire de cette notion. Nous citons ici quelques exemples de ces travaux. E. Pignotti *et al.* (2004) [106] parlent de recommandation adaptative et mettent l'accent sur la compréhension du contexte comme un acteur majeur dans la conception de ce type d'adaptation. Par contexte, les auteurs veulent dire la date et l'heure, l'emplacement de l'utilisateur, historique des services utilisés par l'utilisateur et les préférences et profil utilisateur. W. Quan et H. Jianmin (2006) [109] introduisent le concept de recommandation personnalisée pour désigner un système de recommandation personnalisée de services. C'est un ensemble d'outils personnalisés qui permettent de trouver des informations ou des services intéressants pour l'utilisateur. Le travail présenté en [29] relie la recommandation à l'adaptation et à la personnalisation. Leur travail consiste à générer automatiquement du code pour l'interface utilisateur afin de fournir des services personnalisés en adaptant les offres, en recommandant certains produits ou en offrant un accès facile à certains services relatifs à la localisation et aux tâches utilisateurs.

1.4 Reconfiguration

La reconfiguration est également une forme d'adaptation. A la différence des adaptations précédentes celle-ci ne prend pas en compte l'utilisateur mais la transformation des paramètres du système selon les conditions internes ou externes au système. C'est par exemple, la réduction du débit de communication en cas de téléchargement intense sur le réseau de la part de l'utilisateur. Dans les travaux de A. Giovanni *et al.* (2005) [51] une telle stratégie adaptative est appliquée afin de reconfigurer dynamiquement les services en domotique selon les habitudes d'un nouvel utilisateur et l'état des capteurs. Dans [53], G. Grondin présente une autre forme de cette adaptation en prenant un point de vue reconfiguration logicielle, c'est-à-dire qui se conforme à des conditions nouvelles ou différentes. Sa vision est justifiée par le fait que les besoins logiciels sont spécifiques et complexes. A cause de cette complexité, il peut arriver que les spécifications initiales ne soient plus pertinentes après plusieurs mois d'utilisation de l'application ou du système. Pour lui, la reconfiguration diffère de l'adaptation si toute situation nouvelle que le système rencontre a été prévue initialement par le concepteur. Toutefois, nous pensons qu'une énumération des différentes situations ou conditions de fonctionnement du système est une tâche difficile voir impossible sauf pour des systèmes à états finis.

Les travaux de M. Cremene *et al.* (2004) [87] considèrent une reconfiguration automatique de service en terme d'adaptation au niveau architecture par ajout/retrait/remplacement ou par paramétrage de ses composants. Les auteurs avancent que la plupart des travaux existants qui traitent de l'adaptation dynamique la considèrent dans le cadre des composants logiciels. On cite à titre d'exemple le travail fait dans [131]. Cependant, selon ces auteurs les services offrent des possibilités d'adaptation supplémentaires qui ne sont pas traitées dans ces travaux, comme la migration de services.

1.5 Relations entre ces concepts

Cette section discute les relations entre les différents concepts étudiés précédemment afin de préciser ce que nous appelons *adaptation automatique* dans nos travaux.

Nous avons représenté sous forme schématique (figure 1.1) le positionnement des différents concepts qui se rattachent à la notion d'adaptation selon deux axes non orientés. L'axe des abscisses représente les systèmes automatiques selon qu'ils soient adaptables ou adaptatifs, et l'axe des ordonnées représente la source d'adaptation.

Nous retrouvons sur la figure les trois concepts de base discutés dans les précédentes sections : reconfiguration, personnalisation et recommandation. La figure représente l'environnement système comme la principale source d'adaptation pour la reconfiguration au même titre que l'environnement utilisateur pour la personnalisation. Elle distingue par le trait horizontal la part de l'environnement système ou utilisateur prise en compte dans l'adaptation. De plus, le trait vertical sépare les types d'adaptation utilisée quand il s'agit d'automatiser les actions de reconfiguration ou de personnalisation. La recommandation est une forme de personnalisation dont l'objectif est d'anticiper les choix de l'utilisateur. La recommandation s'adresse à un ou plusieurs utilisateurs selon certains paramètres et des préférences qui les concernent. D'où, la recommandation fera un sous-ensemble de l'ensemble des systèmes de personnalisation.

Les services utilisateurs sont embarqués dans l'environnement de l'utilisateur et ils font partie de l'environnement système qui sera pris en compte dans l'adaptation. Mais l'adaptation automatique de services à l'utilisateur n'est pas une reconfiguration car elle n'a pas vocation à changer le service lui-même : elle implique le dispositif qui donne directement un rendu à l'utilisateur, c'est-à-dire qu'elle a un impact sur le rendu du service pour l'utilisateur. L'adaptation automatique de services à l'utilisateur n'est pas non plus une personnalisation même si elle implique certaines données sur l'utilisateur ou son environnement. De ce qui précède, nous

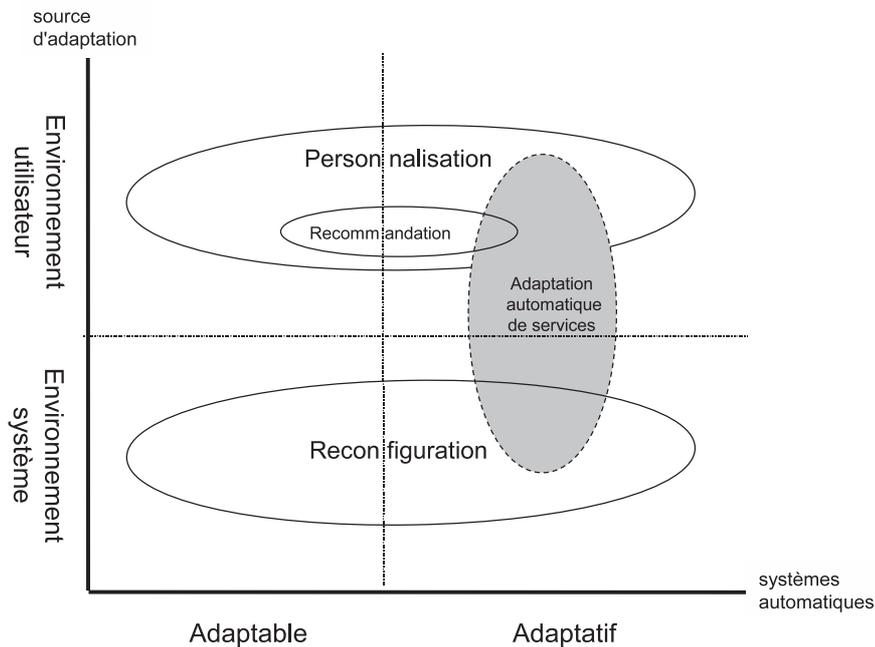


Figure 1.1 – Relation entre les concepts d’adaptation

caractérisons l’adaptation automatique de services comme étant une interaction implicite avec le service afin de changer son comportement à l’égard de l’utilisateur. Les données sur l’utilisateur et son environnement constituent les sources principales de ce changement.

Après avoir donné ces particularités, *nous considérons à présent l’adaptation automatique dans nos travaux comme un procédé adaptatif assuré de manière automatique* et nous pensons qu’elle est du coeur de l’intelligence des systèmes. Dans ce qui suit, nous nous basons sur cet énoncé pour bâtir les fondements de notre travail.

2 Motivations pour l’adaptation de services

Dans cette section, nous analysons les motivations à l’adaptation de services à l’utilisateur. Ces motivations se résument en deux points : assurer la continuité des services à l’utilisateur et améliorer l’interaction de l’utilisateur avec son envi-

ronnement.

2.1 Assurer la continuité des services à l'utilisateur

Les technologies actuelles encouragent la prolifération des espaces intelligents pour les personnes allant d'espaces très réduits comme la voiture à des espaces très larges comme une cité ou une ville. Ce développement est soutenu par la miniaturisation continue des composants électroniques et la baisse de leur coût, et est encouragé par l'émergence de nouveaux besoins utilisateurs comme la réduction de la consommation énergétique, l'aide au domicile des personnes âgées, la surveillance de bâtiments.

L'utilisateur attend de ces espaces un service qui prend en compte ses besoins et assure la tâche demandée. Néanmoins, les services à l'utilisateur sont souvent confrontés à l'absence de mécanismes d'adaptations. Prenons par exemple un service d'appel d'urgence, dans la situation où une personne prépare à manger et utilise une hotte aspirante. A cause du bruit émis, la personne peut manquer un appel urgent. Il est important dans ce cas de trouver une mesure de compensation telle que un service d'affichage de message sur une interface déployée dans la cuisine. Ce genre de solution nécessite des mécanismes d'adaptation qui permettront de basculer de ou vers une solution selon les situations dans l'environnement.

L'adaptation des services à l'utilisateur s'avère un facteur déterminant pour assurer la continuité de ces services et veiller à ce que le service ait bien effectué la tâche demandée.

2.2 Améliorer l'interaction de l'utilisateur avec son environnement

Un des acteurs clés dans la conception des espaces intelligents est d'avoir une interaction plus souple de l'utilisateur avec son environnement. L'adaptation des services à l'utilisateur joue un rôle important dans l'amélioration de cette interaction en facilitant et aidant l'individu dans sa relation avec son environnement physique et social, dans ses dimensions spatiales et temporelles. Par exemple,

la diffusion d'un message téléphonique peut être améliorée en tenant compte de l'endroit, la position et l'orientation de l'utilisateur ; l'appel d'aide d'une personne âgée peut se faire de manière automatique selon la disponibilité et le rapprochement des aidants parmi les voisins, la famille et les amis.

S. J. Louis et A. Shankar (2004) [123] ont montré qu'en tenant compte des informations de l'environnement, l'adaptation améliore l'interaction utilisateur. Elle permet en effet de répondre à certaines questions telles que “devrais-je éteindre ou diminuer le volume de mon lecteur de musique lorsque le téléphone sonne ?” ou encore “dois-je faire une pause de la chanson en cours si la personne quitte la chambre?”. Cette adaptation a notamment pour effet de minimiser l'attention qu'accorde l'utilisateur aux tâches qu'il exécute, comme par exemple, l'attention que l'utilisateur doit consacrer à chaque intervention de modification du statut d'un service [14]. L'adaptation peut aider également à anticiper les activités humaines, par exemple, l'allumage du système de chauffage une heure avant que la personne rentre chez elle [113].

3 Mécanismes d'adaptation de services

La mise en oeuvre de l'adaptation de services nécessite des mécanismes afin de préciser quand et comment s'adapter. Ces mécanismes se distinguent en : identification de la situation pertinente pour une adaptation de services, sélection du service à adapter. Le premier vise à déterminer les situations ou les changements dans l'environnement, et le deuxième vise à sélectionner le bon service à activer afin d'agir sur l'environnement ou de changer son propre comportement dans le but de personnaliser, recommander, contrôler des dispositifs, anticiper, simplifier des tâches, etc. Le service peut être associé au préalable à la situation ou au changement, ou sélectionné automatiquement selon un mécanisme bien défini.

Nous analysons dans ce qui suit différents mécanismes. Notre objectif ici, est de voir si ces mécanismes répondent aux motivations présentées auparavant et de déceler les défis résolus ou qui persistent.

3.1 Adaptation à base de règles

L'adaptation à base de règle consiste à écrire les différentes conditions qui régissent la sélection des services sous forme logique en utilisant un langage de description approprié. Il s'agit donc d'un effort important de la part des développeurs. Cependant, il existe des travaux qui tendent à améliorer cette façon de faire. P. D. Costa (2007) [37] propose dans sa thèse une solution intéressante pour la définition des règles d'adaptation. Il suggère une plateforme pour les développeurs de services avec un cadre de modélisation, base pouvant être élargie et spécialisée en fonction des applications. L'auteur propose une méthode pour déterminer des situations pertinentes pour l'adaptation de service à partir des spécifications initiales de l'application. A partir des détails sur l'utilisation de l'application, un moteur à base de règles est appliqué pour inférer des situations. Ces situations correspondent à des descriptions de conditions d'utilisations pour une ou plusieurs applications.

Ce travail est intéressant du point de vue conceptuel car il permet de réduire le temps de développement des services et facilite la tâche aux développeurs. Cependant, il est difficile pour ce type de système d'assurer une adaptation de services dans des situations qui n'ont pas été déterminées à partir des spécifications initiales. Ceci impose d'utiliser des descriptions bien expressives pour les spécifications initiales afin de couvrir au maximum les situations pertinentes pour l'adaptation.

De plus, dans une approche d'adaptation à base de règles, les développeurs de services sont obligés de prévoir toutes les situations possibles au moment de la construction des règles qui régissent la sélection des services sachant que le service ne peut pas fonctionner correctement en dehors de ces prévisions [38]. Ceci dit, même si on arrive à prévoir toutes les situations nécessaires à l'adaptation, la construction des règles qui régissent la sélection des services pose d'autres difficultés. D'après [87] :

1. Pour chaque service, il est nécessaire d'écrire des règles spécifiques.
2. Si un service doit s'adapter à de nouvelles situations qui n'ont pas été identifiées lors de sa construction, l'intervention du concepteur est nécessaire afin de définir de nouvelles règles ou de réécrire le service.
3. Le lien entre une situation et le service qui doit être activé ou déclenché durant cette situation est connue uniquement du développeur du service et non déterminé automatiquement.

Selon [9], les systèmes à base de règles ont attiré beaucoup d'attention mais leur utilisation comme modèles de prévision dans la planification des activités a été limitée en raison du manque de techniques statistiques permettant de dériver ces règles à partir des données des activités. L'enjeu pour les systèmes à base de ces mécanismes est donc à la fois dans la description des situations de l'utilisateur et de son environnement, et dans leur capacité à maintenir plus de flexibilité et d'ouverture.

3.2 Adaptation à base d'apprentissage automatique

L'apprentissage automatique est l'un des piliers de l'intelligence artificielle par sa capacité d'apprendre à partir de l'expérience. Particulièrement, l'apprentissage par renforcement a pour but de concevoir des systèmes autonomes et adaptatifs. Pour cela, nous allons parler plus en détails de quelques travaux utilisant ce mécanisme puis nous montrons ses avantages et ses inconvénients.

O. Bridiczka *et al.* (2007) [19], emploient un apprentissage supervisé. Le modèle de situation est appris et étiqueté par l'expert, puis l'utilisateur donne un feedback en indiquant le service correspondant durant chaque situation. De manière similaire, S. Zaidenberg (2009) [142] propose que le système détermine le modèle d'adaptation à partir des interactions du système avec l'utilisateur et de l'observation des réactions de l'utilisateur aux réponses du système. Pour cela, elle

utilise un algorithme d'apprentissage par renforcement. Elle définit un environnement équipé d'un système ambiant capable de détecter le contexte de l'utilisateur et d'agir en fonction de cette observation. Un ensemble de prédicats représente un état de l'environnement. La sélection de l'action à entreprendre est basée sur un entraînement de la part de l'utilisateur en utilisant un apprentissage par renforcement avec un entraînement simple de type récompense positive ou négative. Auparavant, A. J. Smiee (1992) [124] a proposé un système similaire pour déterminer le modèle utilisateur de manière implicite, en s'appuyant sur l'historique de l'interaction de l'utilisateur avec le service. Cette nouvelle approche a été validée en utilisant un modèle construit par apprentissage durant l'utilisation d'un service d'information (sport, politiques, finances, etc.). L'auteur utilise les mots contenus dans les articles comme des indicateurs de l'information et construit un réseau de neurones à partir des articles lus ou rejetés durant la session de lecture. Pour les auteurs un mot d'un document peut avoir plusieurs significations selon le sujet abordé ; de ce fait, ils précisent que la sémantique a été le grand handicap de ce travail car le système proposé ne considère pas l'interprétation exacte du mot dans chaque document.

Du côté des applications commerciales, la solution d'Adecco (société suisse)², propose un apprentissage des habitudes du foyer en termes d'utilisation des appareils électriques pour automatiser leur allumage ou leur extinction. L'utilisateur de la solution Adecco a la possibilité d'agir sur les paramètres de configuration et de contrôle localement ou à travers Internet en utilisant une interface logicielle. En prenant ces données, le système peut apprendre à partir des entrées utilisateur pour automatiser des actions. Par exemple, le système peut apprendre que dans certains cas de figure (caractérisés par des données de capteurs) il faut envoyer une alerte par *mail* et non par *SMS*. Adecco propose un système capable d'être configuré par l'utilisateur pour réagir à des situations déterminées. Les corrections de l'utilisateur dans le temps sont prises en compte par le système pour les futures

²<http://www.adecco.ch/home/index.asp>

réactions. Par contre, les données d'entrée pour le système proposé proviennent de l'interface de configuration ou de paramétrage des dispositifs et aussi de capteurs : l'utilisateur intervient pour définir les règles ou les ajuster à ses besoins. Ces règles s'adaptent alors selon les corrections de l'utilisateur dans le temps. Nous mentionnons en dernier que dans ce système, le contrôle des dispositifs n'est pas fait à travers des services au sens informatique.

Ceci dit, un des intérêts pour l'adaptation à base d'apprentissage est la capacité de cette technique à offrir des solutions commerciales. L'avantage le plus clair est que les mécanismes présentés ci-dessus sont moins rigides que ceux présentés dans les sections précédentes, et ce, bien qu'ils nécessitent un entraînement soit pour définir la situation pertinente pour une adaptation de services, soit pour associer la situation pertinente au service à adapter.

S. Zaidenberg (2009) [142] avance un autre avantage en précisant que l'utilisation de ce mécanisme pour l'adaptation dynamique aux situations est possible si on trouve les actions adéquates à chaque situation. Par contre, la technique choisie devra respecter plusieurs contraintes notamment : facilité de l'entraînement car il sera à la charge de l'utilisateur, possibilité d'injecter des connaissances initiales pour que le système ait un minimum de cohérence avec le comportement de l'utilisateur, dès l'initialisation.

Les solutions présentées ci-dessus apportent une amélioration dans l'interaction de l'utilisateur avec son environnement mais elles échouent à assurer la continuité des services si le modèle d'apprentissage rencontre une situation pour laquelle il n'a pas été entraîné. Le temps réel et l'entraînement du modèle restent les difficultés majeures pour ces systèmes. D'un côté, il faut assurer un maximum de fiabilité et de robustesse du système pendant la mise en marche et de l'autre côté si le système est mis à disposition de l'utilisateur novice, cette technologie peut ne pas être maîtrisée.

3.3 Adaptation à base de comparaison

Cette approche est moins utilisée par rapport à l'adaptation à base de règles ou à base d'apprentissage. Elle consiste à comparer les conditions liées à l'exécution d'un service avec les données de la situation en cours. Les conditions sont définies au préalable, par le concepteur, dans la description des services. Le service qui correspond le mieux à la situation courante est sélectionné même si les données sur cette situation sont incomplètes. S. Dietze *et al.* (2009) [43] proposent un algorithme basé sur une mise en correspondance des descriptions des services avec les besoins utilisateurs en utilisant des mesures de similarité. Une approche similaire, utilisée en [107], propose une adaptation par comparaison des profils définis dans l'application avec le profil de l'utilisateur. Une fois le profil sélectionné, les règles d'adaptation déjà définies dans le profil sont appliquées.

Dans cette approche, la description sémantique des situations et des services est le paramètre essentiel pour le bon fonctionnement du système. Vu que ces descriptions sont définies par le concepteur, la continuité des services n'est alors assurée que pour les cas prévus.

3.4 Adaptation à base de fouille de données

Actuellement, nous assistons à un intérêt croissant pour les techniques de fouille de données. Celles-ci ont en effet montré leur capacité à fournir de bons résultats dans plusieurs domaines tels que le WEB et l'analyse des transactions d'achats. Pour le cas de l'adaptation des services, l'approche consiste à explorer les données d'utilisation des services en utilisant ces techniques afin d'identifier des situations potentiellement utiles ou des tendances dans les données profitables pour l'invocation future des services. Les solutions discutées en [95, 108] concernent la découverte des motifs d'usage du service pour les exploiter dans la détection des services similaires ou la recommandation de certains services parce que les clients ont l'habitude d'utiliser des services du même genre. Les tech-

niques employées sont celles provenant du domaine de la fouille de données ou *Data Mining*.

La solution proposée par Edwin *et al.* (2003) [98] vise à améliorer l'usage d'un appareil domotique. Edwin *et al.* proposent d'automatiser certaines actions en cherchant les manipulations régulières dans l'historique des données enregistrées de l'utilisation d'un appareil. Le défaut de cette solution est qu'elle ne s'applique qu'à un seul appareil et elle ne tient pas compte d'autres données de l'environnement comme par exemple les états des autres appareils ou dispositifs.

4 Défis soulevés par l'adaptation de services

La conception de services adaptatifs supportant des environnements et des besoins utilisateurs dynamiques nécessite la résolution de différents défis. A partir des deux sections précédentes, nous allons les résumer ici.

4.1 Prise en compte du contexte

L'ensemble des situations de l'utilisateur et de son environnement est décrit par le terme général "*contexte*". Sa prise en compte est essentielle dans l'interaction de l'utilisateur avec les systèmes. Une des questions de recherche qui reste ouverte est de trouver le moyen de percevoir ce contexte. Les solutions actuelles proposent des capteurs divers pour acquérir les données sur l'utilisateur et son environnement mais elles sont hétérogènes et proviennent de sources diverses et variées. Il en découle que les situations nécessaires à l'adaptation des services deviennent très complexes à cerner et conceptualiser.

4.2 Gestion de volumes de données

Les espaces intelligents sont caractérisés par un volume important de données provenant de capteurs et dispositifs. En plus du nombre de capteurs et dispositifs déployés dans l'environnement, le volume de ces données est augmenté par

la dimension temporelle de ces applications. En conséquence, des solutions de stockage et d'acquisition sont nécessaires afin de permettre l'exploitation de ces données par des mécanismes d'adaptation.

4.3 Représentation de la sémantique

Les services représentent une grande partie des fonctionnalités offertes par les espaces intelligents. Ils vont devoir exploiter les données sur l'utilisateur et son environnement pour s'adapter. Les appareils, les dispositifs dans l'environnement et les Services Web locaux ou distants constituent les sources potentielles de ces données. Il est donc important pour un système d'adaptation automatique de services d'avoir des descriptions expressives sur les sources et les services car la découverte automatique de nouvelles sources et de nouveaux services exige d'un côté d'avoir l'identité de chacune des sources et ses types de données et de l'autre de connaître les propriétés des services et leurs capacités.

La technologie du Web Sémantique fournit des descriptions souvent complexes et de donner du sens aux termes échangés. Cependant, le défi réside dans la définition de ces descriptions et la conception de l'ontologie qui prend en compte l'hétérogénéité sémantique des données échangées.

4.4 Extraction de connaissances

Ce défi est reconnu comme étant la capacité d'identifier, de concevoir les connaissances utiles pour le processus d'adaptation et de montrer comment ces connaissances sont exploitées en temps réel, en particulier dans un environnement très dynamique [112]. T. Chaari *et al.* (2007) [127] clarifient ce défi comme étant une création d'intelligence : analyse des informations du contexte, déduction de la signification ou de la compréhension, et intégration de cette connaissance pour l'adaptation des applications. Dans d'autres travaux ce besoin a été exprimé comme de la fusion d'informations ou de la déduction. Par exemple, Bhusate et Pitt (2009) [16] mentionnent que la fusion des données de capteurs et des actions

utilisateur pour l'adaptation est un défi spécifique pour coordonner l'interaction complexe des utilisateurs, des dispositifs, des capteurs et des systèmes intelligents et offrir des meilleurs services.

4.5 Traitement en temps réel

Adapter les applications et les services à l'exécution est très difficile puisque le temps nécessaire entre la détection, la reconnaissance du modèle d'adaptation et le choix ou la sélection des actions à entreprendre est très court [107].

4.6 Ouverture et flexibilité

Avoir des architectures d'adaptation ouvertes et flexibles est un grand enjeu de recherche. J. Indulska *et al.* (2001) [70] argumentent qu'il n'existe pas d'architecture ouverte, flexible et évolutive d'adaptation. Cela pose aussi la question d'étendre les applications et services existants pour supporter l'adaptation. S. Fritsch *et al.* (2007) [122] mentionnent que l'écriture des applications d'adaptation demeure une tâche ardue pour les développeurs d'applications. Les approches existantes demandent la conception de toutes les pièces d'adaptation au moment du design.

5 Conclusion

Ce chapitre a permis de préciser la notion d'adaptation de services. L'adaptation a un intérêt particulier quand il s'agit de l'utilisateur par sa capacité à lui offrir plus de liberté et de souplesse dans ses interactions et son apport dans la robustesse aux changements de l'environnement. *Nous définissons alors l'adaptation de service à l'utilisateur comme étant la capacité à faire bénéficier le service de l'expérience de son utilisation en tenant compte des situations de l'utilisateur et de son environnement.* La notion de contexte qui se réfère aux situations de l'utilisateur et de son environnement apparaît comme l'enjeu principal d'adaptation

des services à l'utilisateur par rapport aux défis présentés ci-dessus. En effet, la difficulté réside dans la détermination des situations utiles ou pertinentes pour l'adaptation dans un environnement dynamique difficile à caractériser. En faisant le lien entre le contexte et l'adaptation automatique, il nous semble clair que l'interaction de l'utilisateur avec son environnement à travers des services sensibles au contexte et adaptatifs est une approche prometteuse qui mérite d'être creusée. C'est pour cela que nous avons choisi d'étudier plus précisément la notion du contexte. Nous y consacrons le prochain chapitre.

CHAPITRE 2

CONTEXTE

Le chapitre précédent a examiné la problématique d'adaptation. Nous avons remarqué que le contexte est une notion importante pour caractériser les informations sur l'utilisateur et son environnement dans le processus d'adaptation de services. Ce chapitre propose une réflexion sur le contexte avec une attention particulière à sa modélisation et à son interprétation dans les systèmes et applications existantes. Le but est de savoir quelle description du contexte utiliser et quelles techniques existantes permettent d'atteindre cette description à partir de données brutes. La section 1 présente une vue d'ensemble de la notion de contexte. La section 2 montre comment le contexte est fourni aux applications. Dans les sections 3, 4 et 5, nous étudions le contexte du point de vue utilisation dans les applications c'est-à-dire sa description, ses représentations et ses techniques de traitement. Dans la section 6, nous ferons le lien entre traitement et description du contexte puis nous terminons ce chapitre par une synthèse.

1 Vue d'ensemble de la notion de contexte

Le contexte est une notion complexe, qui a été étudiée dans plusieurs disciplines souvent de manière complètement indépendante. Ces disciplines sont, par exemple la linguistique, l'intelligence artificielle [75] et l'intelligence ambiante [41]. En linguistique¹, le contexte est le texte qui entoure un élément de la langue et qui détermine son sens et sa valeur.

Dans le domaine de l'intelligence artificielle, la définition du contexte découle de deux grandes approches : approche dite logique propositionnelle de McCarthy (1987) [75], Guha (1991) [55], McCarthy et Buvac (1994) [76], et approche dite de raisonnement local (Local Models Semantics (LMS) / Muti Context Systems

¹www.universalis.fr/encyclopedie/context-linguistique

MCS) de Giunchiglia (1993) [48], Ghidini et Giunchiglia (2001) [52]. La première approche considère le contexte comme étant les circonstances dans lesquelles est déterminée la valeur de vérité d'un terme. Cette vision est traduite en un ensemble de prédicats permettant de manipuler ce concept dans le but de faire du raisonnement et des inférences logiques. Dans l'approche LMS/MCS, le contexte est un ensemble connu de faits sur lequel est effectué un raisonnement. Ce contexte est nécessairement partiel et approximatif, description incomplète du monde. Une comparaison détaillée de ces deux approches est proposée par Serafini et Bouquet dans [86].

En intelligence ambiante, le contexte a fait son entrée par le biais de l'avancement technologique des capteurs. Dès lors, les chercheurs ont eu tendance à développer des solutions pour son acquisition et son utilisation dans des applications. Ces solutions nécessitent des descriptions claires de ce qui constitue le contexte : depuis des représentations cohérentes jusqu'à des traitements qui permettent de l'exploiter efficacement et intelligemment dans les applications.

Dans ce qui suit, nous nous intéressons particulièrement à la notion de contexte comme traitée en intelligence ambiante et mettons en avant ses aspects importants pour nos travaux.

1.1 Définition du contexte

La manière la plus répandue de définir le contexte est d'énumérer la liste des informations considérées comme du contexte pour les applications. Selon ces applications, le contexte peut ainsi revêtir des sens différents : temps de la journée, durée d'un événement, emplacement d'un objet [63], historique d'une interaction [105], caractéristiques, connaissances et intérêts personnels [100],...

Dans ces différents travaux on peut noter que le contexte se réfère principalement à la description d'une situation [12, 69, 104]. Ceci reste cependant ambiguë car l'étendue de la situation n'est pas précisée. J. Pascoe (1998) [72] et A. Ward *et al.* (1997) [3] voient le contexte comme une situation représentée par un état ou

un ensemble d'états. Un état peut, par exemple, correspondre à l'emplacement des objets dans l'environnement. S. Greenberg (2001) [121] affirme qu'une énumération exhaustive de l'ensemble des états du contexte existants de l'environnement semble cependant difficile. En outre, nous ne pouvons pas toujours savoir quelles informations de l'environnement déterminent un état particulier de ce dernier. Par exemple, la présence de fumée n'induit pas automatiquement un état d'incendie. En conséquence, le déclenchement automatique d'un service d'alerte sur la base d'un tel état semble difficile.

Parmi les définitions présentées ci-dessus, la définition de Dey (2001) [41] qui présente le contexte comme "*Toute information qui peut être employée pour caractériser la situation d'une entité. Une entité est une personne, un endroit, ou un objet qui est considéré pertinent à l'interaction entre un utilisateur et une application, y compris l'utilisateur et les applications eux-mêmes*", est souvent reprise car elle permet de faire un équilibre entre les deux tendances. Elle est l'une des définitions du contexte la plus expressive et est très suivie en littérature. Cette définition présente un intérêt particulier pour nos travaux car elle mentionne clairement l'utilisation de l'information sur une entité pour décrire sa situation. Plus récemment, Truong *et al.* (2005) [7] considèrent le contexte comme une collection d'informations qui caractérise l'interaction entre l'utilisateur et l'application. Cette définition se rapproche de celle de Dey, à la différence que les auteurs mettent plus l'accent sur le rôle du contexte dans l'interaction.

Dans ce qui suit, nous prenons la définition de Dey comme référence et nous ferons un tour d'horizon rapide de travaux sur le contexte pour donner trois grandes lignes déterminantes qui entrent dans la définition du contexte. Ces grandes lignes découlent de notre propre vision du contexte et son utilisation dans nos travaux. Nous les présentons dans les sous-sections suivantes.

1.2 Contexte et Temps

Nous pouvons tout d'abord constater que le contexte ne peut être pris indépendamment de l'espace-temps. Toute situation ou interaction se produit nécessairement dans un interval temporel. La granularité de cet interval est très importante. En effet, elle définit l'étendue des informations considérées comme du contexte. Cet interval peut aller d'un événement très court à la durée d'une activité par exemple. De ce fait, on peut passer facilement d'un nombre très réduit d'informations à une base de données selon qu'on s'intéresse à un événement isolé ou à une suite d'événements. Ce constat rejoint certains avis (Ryan *et al.* (1998) [115]) sur l'importance du temps dans la caractérisation du contexte.

1.3 Contexte et Interaction

La définition de Dey mentionne l'interaction de l'utilisateur avec les applications sans considérer l'interaction entre les applications elles-mêmes. Or, cet aspect peut être utile vu le développement croissant d'infrastructures M2M (Machine to Machine) et Services Web où le contexte relatif à l'interaction entre applications peut contribuer à la réactivité et l'efficacité de ces systèmes à l'égard de l'utilisateur. Cette vision de l'interaction au sens large a été défendue par P. Brézillon (2002) [22] et Coutaz *et al.* (2005) [74] où ils vont jusqu'à considérer le contexte comme un problème interactionnel dans lequel l'interaction elle-même peut être importante.

1.4 Contexte et Pertinence

Enfin, notre dernier constat porte sur la notion de pertinence du contexte. Cette dernière permet de déterminer les informations du contexte qui sont utiles et qui sont impliquées dans la conception d'un système sensible au contexte. Dey mentionne ce besoin dans sa définition sans donner trop de détails sur comment déterminer quelles sont les informations pertinentes. O. Bucur *et al.* (2005) [23] sont

allés plus loin dans la caractérisation et la formulation de la pertinence des informations du contexte en associant au contexte le concept de *finalité* qui signifie l'objectif pour lequel le contexte est utilisé à un moment donné. Cette objectif peut être à court ou à long terme. Dans le cas où l'objectif est à long terme, cela signifie que le contexte est archivé au fil du temps tout en conservant de la pertinence. Prenons l'exemple d'un système qui vise à adapter l'interface utilisateur selon les habitudes d'utilisation de l'interface. Les données sur les utilisations passées restent pertinentes parce qu'elles contribuent à déterminer les prochaines adaptations de l'interface et de prédire les actions futures de l'utilisateur.

Dans le cas où l'objectif est instantané, le contexte courant est utilisé directement dans le processus d'adaptation, par exemple le verrouillage automatique des portes et des fenêtres à la sorties de l'utilisateur de sa maison.

Après ce tour d'horizon, nous verrons plus loin dans ce chapitre que ces questions présentent un grand défi dans la conception des systèmes, des applications et des services sensibles au contexte.

2 Obtention du contexte

Après avoir vu les différentes définitions et avant de poursuivre notre analyse, il semble important de présenter comment le contexte est obtenu par les applications.

L'analyse des travaux existants montre que les systèmes utilisant le contexte emploient deux types d'architectures : architectures à couches [18, 30, 40, 42, 60, 93] et architectures distribuées [23].

L'architecture à couches structure l'architecture en différents niveaux allant de l'acquisition du contexte jusqu'à son utilisation dans les applications. H. Karen (2003) [60] a étudié le contexte du point de vue de la conception des logiciels. Elle propose une couche dédiée à la gestion du contexte appelée CML (Context Management Layer) basée sur un modèle relationnel. Elle est intégrée dans une architecture à plusieurs couches : acquisition, réception, gestion, interrogation et

adaptation. L'acquisition se fait à partir des capteurs. La réception est une interface entre la partie acquisition et la couche gestion : elle sert à transformer les informations hétérogènes du contexte en une abstraction plus adéquate pour le gestionnaire du contexte. L'interrogation est une autre interface qui permet à la couche adaptation de facilement exprimer une requête sur un modèle de contexte en utilisant des faits ou des abstractions de situations.

L'approche distribuée a l'avantage de supporter un très grand nombre de sources de contexte (capteurs par exemple) et d'être robuste à leur défaillance. Cette approche emploie un ensemble d'agents logiciels ayant des capacités de perception de leurs environnements à travers des capteurs. Ainsi, les agents sont capables d'interagir entre eux afin de coordonner leurs actions en prenant en compte le contexte acquis par chacun d'eux sur l'utilisateur et son environnement. Le travail de O. Bucur *et al.* [23] et celui de Chen *et al.* (2004) [59] constituent les premières ébauches vers ce type d'architecture.

L'acquisition est la partie la plus importante de ces architectures car elle conditionne l'utilisation du contexte au niveau applicatif. D'après M. Baldauf *et al.* (2007) [13], une séparation de la détection et de l'utilisation du contexte est nécessaire pour améliorer l'extensibilité et la réutilisabilité des systèmes. M. Baldauf *et al.* (2007) [13], Chen (2004) [30] et Winograd (2001) [128] résument trois approches d'acquisition du contexte : accès direct aux capteurs, accès à travers des intergiciels, et accès à travers des serveurs de contexte. Dans l'approche d'accès direct au capteurs, le client logiciel communique directement avec le capteur et transforme les données brutes dans un format acceptable pour ses besoins. Ce type d'architecture permet une communication facile mais réduit la flexibilité et complique la gestion des différentes données par plusieurs utilisateurs. L'accès à travers des intergiciels permet une vue uniforme de l'espace physique. Il présente aussi les avantages d'intégration, de management, d'ajout, de suppression et d'interrogation des capteurs. Les serveurs de contexte sont des composants qui agissent comme des médiateurs, dans le but de soulager la tâche des clients indivi-

duels. Le serveur du contexte exécute un traitement a priori sur le contexte dans le but de réduire le traitement des acteurs individuels, qui n'ont besoin que d'envoyer une requête à un point d'accès particulier et d'avoir en retour une réponse.

L'acquisition a permis de passer d'un contexte réel non touchable à des descriptions ayant une certaine représentation en vue d'être exploitées par les applications. Nous allons dans les sections suivantes aborder plus en détails trois piliers de son utilisation par les applications : sa description, ses différentes représentations et ses techniques de traitement. Chacun de ces piliers nous permettra de construire une idée sur la description, la modélisation et la technique qui participent à l'adaptation automatique de services.

3 Description du contexte

La littérature propose de multiples descriptions du contexte. Dans ce qui suit, nous les analysons selon trois points de vue : domaine de définition, propriétés, niveaux de description.

3.1 Domaine de définition

Au vu des travaux existants, le domaine de définition du contexte i.e. les frontières des données considérées comme relevant du contexte, est très variable. Par exemple, pour C. Bolchini dans [39] l'étendue du contexte couvre l'environnement utilisateur, ses préférences et ses caractères personnels. Dans [93], le contexte est beaucoup plus étendu. Il comporte les données physiques (température, temps, formes, etc.), sociales (ses amis, ses appels téléphonique), utilisateur (profil, préférences, émotions), système (nombre de processus en exécution, espace mémoire, etc.). Le domaine de définition du contexte dépend généralement des applications elles-mêmes et de la dimension qui est donnée au contexte.

3.2 Propriétés

A partir de la diversité des domaines entrant dans la définition du contexte, on voit donc, comme le constate Karen dans [60] que la description du contexte est fortement hétérogène. De manière générale la description du contexte fait intervenir la notion de certitude. Ceci s'explique par le fait qu'une donnée est toujours soumise à une valeur de confiance surtout si cette donnée est délivrée par des capteurs. Le volume de données et la spatio-temporalité sont également deux propriétés de cette notion. En effet, en considérant les données capteurs comme du contexte dans la plupart des travaux, un large volume de données est alors engendré à cause notamment du nombre de capteurs déployés et du changement plus ou moins fréquent de leurs valeurs au fil du temps.

D'après C. Bolchini en [39], les applications sensibles au contexte montrent l'importance de prendre en compte l'espace (la localisation), la temporalité, la localisation et la temporalité référentielles (proche, très proche, loin, etc.) et l'historique du contexte. L'historique de ces capteurs est utilisé soit pour vérification en cas d'anomalie du système d'acquisition, soit pour le besoin d'expérimentations [61], pour prédire des contextes futurs [93] ou encore pour la reconnaissance du contexte [18].

3.3 Niveaux de description

A partir de ce qui précède nous allons montrer que le contexte peut être vu à différentes échelles. La définition de Dey introduit notamment le niveau *situation*.

Un premier niveau de description est le niveau *données* pour définir le *contexte brut*. Le contexte brut est un ensemble de données initiales qu'on peut organiser selon un axe spatial et un axe temporel. Ces données ne sont pas rattachées à une signification particulière. Par exemple une valeur envoyée par un capteur de température à 16h depuis la cuisine est une donnée brute.

Le deuxième niveau concerne le *contexte significatif*. Il permet d'associer une

signification à la donnée brute du niveau *données*. Par exemple la valeur de température du thermomètre peut être associé à l'information "chaleur élevée".

Le troisième niveau que nous identifions est celui de *situation*. Une situation concerne un ensemble de signifiants attachés aux données et liés par un ensemble de relations.

Le quatrième niveau est celui de *compréhension*. Une compréhension est un ensemble de situations associées entre elles par des liens spacio-temporels.

4 Représentation du contexte

L'analyse de la littérature met en évidence deux modèles de représentation : modèles logiques des données de contexte et modèle conceptuel du contexte. En effet, ces modèles portent sur deux niveaux de description évidents qui sont respectivement *données* et *situation*. Nous détaillons dans cette section chacun des ces modèles afin de faire un choix approprié pour nos travaux. Nous présentons également quelques efforts de création de standards de représentation du contexte par le W3C pour avoir un regard sur ces travaux qui nous aidera dans notre démarche de représentation.

Avant cela, deux travaux intéressants ont constitué le fondement de notre étude sur ces modèles de représentations. Le premier est celui de Cristiana Bolchini (2007) [39]. L'auteur a proposé un cadre d'analyse pour les modèles de contexte et une comparaison jusqu'à ce jour des approches actuelles. Dans cette analyse, les auteurs constatent que malgré la généralité des modèles proposés, on peut identifier un focus bien défini en n'attaquant qu'un sous-problème spécifique du contexte. Ils ajoutent que les systèmes voulant offrir des modèles de contexte qui supportent un nombre important d'applications, souffrent souvent de ne pas être efficaces. Cette conclusion plaide pour un équilibre entre généralité et efficacité pour toute tentative de représentation du contexte.

Le second est celui de T. Strang *et al.* (2004) [125]. Les auteurs ont présenté une étude intéressante dans laquelle ils détaillent six approches de la représenta-

tion du contexte selon les modélisations : clé-valeur, langages à schéma de balises, langage graphique de modélisation (UML, ER, etc.), modélisation orientée-objets, modélisation basée sur la logique et finalement ceux basées sur les ontologies. Cependant, cette analyse n'a pas permis de dégager d'approche claire qui permettent d'atteindre les niveaux de descriptions du contexte et les représentations qui vont avec. Les auteurs se sont contenté de défendre les modèles de représentation à base d'ontologie. Le potentiel de représentation de ces derniers est pourtant déjà bien admis par la communauté scientifique.

4.1 Modèles conceptuels

Les modèles conceptuels du contexte fournissent une description du contexte au niveau *situation*. Les formalismes de représentation utilisés sont ceux de prédicats logiques, de diagrammes UML, entités-relations, d'ontologies ou d'abstractions sémantiques [33]. Ces modèles sont définis lors de la spécification. Le développeur représente les situations lors du développement ou agrège les informations renvoyées par les capteurs pour les associer à une étiquette de situation. Cependant, cette pratique provoque une dépendance forte entre les modèles et les applications, et ne favorise pas la flexibilité dans ces applications. Nous présentons dans ce qui suit quelques travaux utilisant des modèles conceptuels.

Dans [60], les situations sont spécifiées par un ou plusieurs prédicats qui peuvent être combinés pour décrire des situations plus complexes. Ces situations sont décrites à partir des données issues de capteurs conformément au modèle de description. E. D. Dejene (2007) [40] définit un modèle conceptuel appelé *EH-RAM*. Il est basé sur des méta-entités du contexte reliées par des relations et régies par des axiomes, et constitue l'élément de base de représentation du contexte.

Nous pouvons retrouver d'autres travaux dans l'étude de P. D. Costa (2007) [37]. L'auteur a étudié plusieurs systèmes utilisant une modélisation conceptuelle du contexte tels que SOUPA , CoOL et CONON. Cette étude révèle l'absence de

fondements d'ontologies dans ces systèmes. Par ailleurs, le W3C ² propose un standard "*Delivery Context : Interfaces (DCI)*", qui définit la plate-forme et l'interface de programmation pour permettre à des applications Web d'accéder à une hiérarchie de propriétés dynamiques relatives au contexte. Elles concernent les capacités des terminaux, les configurations, les préférences utilisateurs et les conditions environnementales. Cet effort est lié à la définition d'un *contexte de livraison d'un appareil*, description des fonctionnalités des terminaux et des préférences de l'utilisateur ("profile CC/PP") ³. Il peut être utilisé pour guider l'adaptation du contenu présenté à ce dispositif. Ces profils et le vocabulaire CC/PP s'appuient sur RDF (*Resource Description Framework*) et ont des identifiants (URI) utilisés pour faire référence à des capacités spécifiques ou à des préférences.

Malgré la grande qualité de ces travaux, la prise en compte du contexte a été limitée à la représentation d'un ensemble de données sans se soucier de la spécificité de ces données (interaction, temps, pertinence), que nous avons discutée au début de ce chapitre. En d'autres termes, l'application a la charge d'acquérir le contexte courant en accord avec le modèle pré-établi du contexte sans vérifier la pertinence de ces données, ni l'interaction et ses conditions. En effet, ces modèles de représentation conceptuel sont étroitement liés aux applications qui les utilisent. Cet aspect freine le développement rapide des applications car d'un côté c'est à la charge du développeur de définir le contexte pertinent pour l'application et de l'autre côté le temps consacré à la construction des modèles de situations est très important.

4.2 Modèles logiques

Les modèles logiques des données de contexte sont utilisés pour décrire et structurer le contexte brut selon des représentations numériques, symboliques ou structurelles. Dans sa forme numérique, les données correspondent à des valeurs

²<http://www.w3.org/TR/2006/CR-DPF-20061019/>

³<http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/>

numériques provenant principalement des capteurs. La source de données est le seul élément qui permet de les décrire. Par exemple, la valeur (15) envoyée par un thermomètre est une température. Les données de contexte représentées sous une forme symbolique sont des concepts qui désignent des faits reconnus par des capteurs ou autres dispositifs, par exemple, l'activité d'une personne (debout, assis, marcher, courir, intérieur, extérieur, etc.) [71, 85]. Ce type de données ne peut pas être délivré à partir de capteurs ordinaires. Un apprentissage de bas niveau est souvent utilisé. Dans sa forme structurelle, les données de contexte sont inscrites dans des structures de données pour leur permettre de porter par elles-mêmes leur propre description de la donnée. Nous avons identifié trois types de représentation structurée : *attributs-valeurs*, *objet* et *sémantique*. Nous détaillons d'abord ces modèles, puis nous concluons cette section avec une comparaison dans le but d'identifier le modèle de représentation le plus pertinents pour nos travaux.

4.2.1 Attributs-Valeurs

Les données de contexte décrites sous la forme d'attribut-valeur sont organisées en couple (nom de l'attribut - valeur) [21], paramètres-attributs [54] et sous forme de balises-champs [20]. D'autres données sont structurées en lignes ou en colonnes. La plupart d'entre elles sont celles contenues dans les fichiers de traces. Chaque colonne représente une implémentation de la relation entre le nom de l'attribut et la valeur.

4.2.2 Objets

Les données objets du contexte désignent un mode de représentation structurée pour décrire un élément par ses caractéristiques. Par exemple la position fournie par une caméra traqueur est un objet composé de trois valeurs X, Y et Z. Deux raisons conduisent à la représentation des données de contexte comme des objets : d'une part la réalisation d'un raisonnement logique (inférence, déduction etc.), et d'autre part la manipulation facile d'objets en utilisant la conception orientée objet

qui a fait ses preuves dans le domaine du développement logiciel.

4.2.3 Sémantiques

Les données sémantiques du contexte désignent du contenu informationnel composé des valeurs des données, de concepts et de relations qui aident à expliquer ces valeurs. Plusieurs études se sont intéressées à représenter le contexte en utilisant les modèles sémantiques. Les données de contexte dans [28] sont représentées par un n-uplet (sujet, attribut, valeur, temps, certitude). Dans [46], les données de contexte sont des assertions RDF ⁴. Dans [27], les données de contexte sont décrites en termes de propriétés et attributs exprimés dans un modèle de langage Web. D'autres études se sont intéressées au modèle CC/PP ⁵, qui est une norme de W3C pour exprimer les capacités des dispositifs et les profils de préférence. Ce qui est commun [56] à toutes les représentations sémantiques est une modélisation déclarative sous forme graphique qui peut être employée pour représenter la connaissance ou pour soutenir les systèmes automatisés afin de raisonner sur les connaissances.

P. Korpipää and J. Mäntyjärvi (2003) [82] expliquent comment passer de valeurs mesurées à des données sémantiques de contexte. L'objectif est d'utiliser les concepts du monde réel conjointement avec les valeurs mesurées pour faciliter l'exploitation du contexte dans les applications. En outre, l'approche exige que les valeurs soient déjà mesurées en correspondance avec les situations du monde réel puis décrites selon plusieurs paramètres comme ci-dessous :

- Type du contexte : la dimension du contexte représenté par un identifiant unique, par exemple physique (température).
- Valeur du contexte : elle peut être symbolique ou numérique par exemple, froid ou 1 degré.

⁴<http://www.w3.org/RDF/>

⁵Composite Capabilities/Preference Profile (CC/PP)

- timestamp : l'instant de l'événement horodaté.
- Source : un attribut optionnel qui désigne la source émettrice du contexte en question. Dans le cas d'un capteur, cela peut être l'identifiant du capteur.
- Confiance : cet attribut décrit la certitude de l'information du contexte

P. Korpiää and J. Mäntyjärvi ont utilisé RDF comme formalisme de la syntaxe afin d'arriver à des niveaux de description élevés du contexte, exploitables facilement par des applications mobiles.

4.2.4 Comparaison

Nous appuyons notre analyse précédente des modèles de représentation logique par d'autres travaux de la littérature afin de mieux les comparer.

Les travaux [39], [130], [13] attirent l'attention sur l'absence d'utilisation de modèles sémantique dans l'apprentissage du contexte. En effet, M. Baldauf *et al.* (2007) [13] précise dans son étude des systèmes sensibles au contexte l'importance des modèles sémantiques pour définir, manipuler et stocker les données dans un format compréhensible par la machine. Par exemple, les systèmes sensibles au contexte décrits en [130] représentent les informations de contexte en XML, RDF et OWL puisqu'ils sont considérés extensibles et interopérables, en particulier RDF et OWL facilitent la réutilisation du vocabulaire commun. Néanmoins, C. Bolchini *et al.* (2007) [39] constatent que toutes les solutions qui construisent un historique du contexte ne considèrent pas de représentations du contexte basées sur les ontologies. Ils constatent également que les modèles à base d'ontologies ne sont pas dotés de caractéristiques d'apprentissage automatique.

En effet, la représentation sémantique a un potentiel de représentation intéressant. Elle est de plus soutenue par la présence de langages de représentation existants comme RDF et Topic Maps. Nous rejoignons H. Truong and S. Dustdar (2009) [130] sur ces points et nous nous demandons par rapport à ce qu'a dit C. Bolchini *et al.* (2007) [39], s'il est possible de proposer une représentation

sémantique pour décrire le contexte brut et si cette représentation peut convenir aux différents niveaux de description du contexte. C'est dans ce cadre que se situe notre analyse. Nous avons résumé dans la figure 2.1 les niveaux de description et leurs représentations possibles. Il ressort de la figure que les modèles sémantiques du contexte sont présents à différents niveaux de description. Ainsi, il semble adéquat d'avoir des techniques de traitement du contexte qui permettent de passer d'un niveau bas de description à un niveau plus complexe sans changer de représentation. Surtout si cette représentation est dans un format compréhensible à la machine.

Compréhension	Sémantique
Situation	Sémantique
Contexte significatif	Attributs - Valeurs / Objets / Sémantique
Niveau de description	Représentation

Figure 2.1 – Représentation des niveaux de description

5 Traitement du contexte

Dans cette section, nous nous intéressons aux principales techniques de traitement du contexte. L'objectif est de déterminer les techniques permettant d'arriver à un niveau particulier de description à partir du contexte brut. A partir de l'analyse des principales techniques existantes, nous présenterons une vision synthétique des liens entre niveaux de description et techniques de traitement.

5.1 Raisonnement

Le raisonnement symbolique sur le contexte met en place des techniques d'inférence sur des données multiples pour déduire ou détecter d'autres contextes [97]. Il est plus pratique à employer dans le développement des applications sensibles au contexte car il suffit de définir les bonnes règles qui définissent le domaine. Des travaux [17] s'intéressant à ce type de traitement ont fait évoluer l'aspect basique du raisonnement en l'associant à d'autres modèles. L'objectif est d'assurer une meilleure prise en compte du contexte c'est-à-dire chercher à avoir des descriptions du contexte plus riches. Nous détaillons ici un exemple de ces travaux.

E. D. Dejene (2007) [40] propose un modèle hybride qui distingue les données du contexte de la sémantique du contexte. Ces deux aspects sont traités séparément et les résultats sont réunis pour effectuer un meilleur raisonnement durant la prise de décision dans un environnement sensible au contexte. Ce modèle emploie les ontologies et les principes des bases de données pour modéliser à la fois la sémantique et les données. Le raisonnement s'effectue par le biais d'axiomes particuliers. Par exemple, une déclaration qui dit que "dans des conditions normales, un patient est toujours traité par le même médecin" peut être considérée comme un axiome (hypothèse) dans un domaine médical. Un axiome est une phrase, une proposition ou une règle valide. En fonction des relations axiomatiques utilisées, de nouvelles connaissances sont déduites pour le raisonnement. Dans son implémentation, l'auteur utilise entre autres les solutions de raisonnement de l'*API* fournie

par *protégé*⁶ fondé sur un moteur de logique de description et une extension RDF du modèle JENA⁷ faisant du raisonnement en chaînage avant et chaînage arrière.

Néanmoins, les solutions à base de raisonnement décrites en [17, 40, 97], sont très dépendantes du domaine et se focalisent sur les données du contexte courant. L'historique de ces données n'est pas exploité durant le raisonnement. Dans ce cas, il n'est pas possible pour les systèmes à base de raisonnement de déduire ou d'inférer des descriptions plus complexe comme les habitudes par exemple. Nous soulignons en dernier que ces systèmes sont très rigides à cause des liens étroits entre la couche acquisition du contexte et la couche de raisonnement. Ces travaux présentent bien la deuxième couche sans préciser comment le contexte est acquis par l'application.

5.2 Fusion de données

Dans [57], la fusion de données est définie comme un processus portant sur l'association, la corrélation, et la combinaison des données et des informations provenant d'une ou plusieurs sources. La fusion de données construit de nouvelles données ou informations qui ne sont pas nécessairement des interprétations des données de bases. Prenons l'exemple de deux données de contexte fournies par deux capteurs différents de localisation : le premier capteur donne les coordonnées (X, Y, Z), alors que le deuxième donne le nom du lieu ("le salon", par exemple) ; la fusion de ces deux données produit une donnée plus riche sur la localisation (X, Y, Z, Lieu). La fusion de données de contexte est capable d'intégrer des données issues de sources de contexte multiples et variées. Nous donnons dans ce qui suit quelques exemples de travaux relatifs à cette approche.

Dans [1, 66, 68], les auteurs emploient deux méthodes de fusion : fusion multi-sources et fusion temporelle. En premier lieu, le contexte significatif est produit à partir du contexte brut sous la forme de symboles. Ces symboles sont ensuite

⁶<http://protege.stanford.edu>

⁷<http://jena.sourceforge.net/>

exploités par les deux méthodes de fusion.

La fusion multi-sources consiste à générer du contexte à partir des différents symboles issus de différentes sources à un instant donné. Par exemple "courir", "dans une gare" conduit à l'information "la personne est en retard".

La fusion temporelle quant à elle, est une fusion séquentielle de symboles produits à des instants successifs sur une courte période : par exemple "parcourir la table des heures de bus", "marcher vers la sortie", "marcher dans la lumière du jour", conduit à générer "la personne se dirige vers l'arrêt de bus".

L'algorithme utilisé en [68], est un algorithme appelé SCM (Symbol Clustering Map). C'est un algorithme d'apprentissage qui cherche à déterminer les motifs des symboles fréquents qui figurent dans les données en entrées. Cet algorithme d'apprentissage ne peut être utilisé tel quel pour identifier les motifs temporels puisque la similarité ne fonctionne que sur des symboles non ordonnés. L'algorithme modifié, appelé SCM2, peut être quant à lui appliqué sur des séquences ordonnées.

Dans [137], H. Xin *et al.* (2009), une méthode de fusion de données de capteurs pour l'inférence des activités est proposée. La méthode employée est celle de Dempster-Shafer (1967) [101]. C'est une généralisation des probabilités mathématiques permettant une meilleure quantification de l'incertitude. Elle est intégrée dans une architecture à couches pour automatiser les services à la base de données de contexte qui sont obtenues à partir de capteurs ou des fournisseurs de contexte, et qui sont généralement peu compréhensibles et non interprétables directement par le service. Le but est d'obtenir des activités avec un degré de confiance acceptable.

Bien que ces approches soient intéressantes, elles ne permettent pas souvent de construire des descriptions cohérentes du contexte car leur objectif est focalisé sur l'intégration et non l'interprétation des données de base. De plus, dans ces méthodes seul le contexte courant est considéré pertinent pour la fusion et rarement les données passées. Nous verrons dans la prochaine sous-section une méthode

fondée sur l'exploitation de l'historique des données.

5.3 Reconnaissance

La reconnaissance du contexte correspond à l'utilisation des techniques d'apprentissage automatique pour déterminer des description de contexte de niveau supérieur à partir des données provenant généralement de capteurs. Les techniques appliquées fonctionnent généralement en deux phases : une phase hors ligne permet de déterminer les contextes recherchés qui reflètent les modèles d'apprentissage appris à partir de l'historique des données stockées dans des fichiers ou des bases de données ; une deuxième phase, en ligne, permet de détecter les contextes identifiés hors ligne. Les techniques utilisées se rapportent en majorité aux techniques développées dans le domaine de l'intelligence artificielle. Nous allons détailler plusieurs travaux plus profondément que précédemment dans la mesure où ces méthodes sont au coeur de notre travail.

Parmi les travaux existants, on peut citer celui de Mark (2005) [18] où l'auteur s'est intéressé à des contextes relatifs à la personne comme la posture, l'activité, la parole et la localisation, et s'est appuyé sur des techniques de classification automatique durant des expérimentations de type "*classificateur Naive Bayes à base de distributions Gaussiennes de probabilités*", "*classificateur Naive Bayes à base de mixtures de Gaussiennes*", "*arbres de décision C4.5*" et "*modèles de Markov cachés*". La précision de reconnaissance obtenue par ce travail sur des données de capteurs est intéressante. Elle varie entre 91% et 95% pour la posture, de 68% à 92% pour l'activité, de 80% à 81% pour la parole et de 62% à 67% pour la localisation.

Dans [67], les auteurs emploient une méthode basée sur la segmentation de séries temporelles de données pour la reconnaissance d'états de la personne comme par exemple "marcher". Cette méthode consiste à segmenter de manière non supervisée la série chronologique des données produites par les capteurs. Cette méthode est couplée à la programmation dynamique pour minimiser la variance

intra-segment, très lente pour une longue séquence de données. De ce fait, un algorithme appelé GIR (Global Iterative Replacement) donne le même résultat en une fraction du temps nécessaire pour la programmation dynamique. L'algorithme repose sur le principe suivant : ayant K -segments s_1, s_2, \dots, s_k , l'objectif est d'obtenir une segmentation où les segments sont intérieurement homogènes. Le problème revient à trouver les frontières C_i qui minimisent la totalité du coût de la segmentation au lieu de se limiter aux points frontaliers C_{i-1} et C_{i+1} . Comparativement à des situations réelles, les auteurs affirment que la segmentation capture les changements. Par exemple, il est possible d'obtenir la transition de "se lever" à "marcher".

En [132], une méthode est combinée pour la classification des activités les plus basiques de la personne à partir de capteurs comme par exemple "courir", "marcher", etc. La méthode combine deux algorithmes très connus de regroupement de données K-SOM (Kohonen Self-Organizing Map) [126] et K-means [2]. K-SOM est employé de manière non supervisée pour stocker chaque vecteur de données en entrée dans une cellule d'une carte à deux dimensions et k-means est utilisé pour minimiser la somme de la racine de la distance entre un vecteur en entrée et un vecteur prototype W_{ij} comme le montre la figure 2.2.

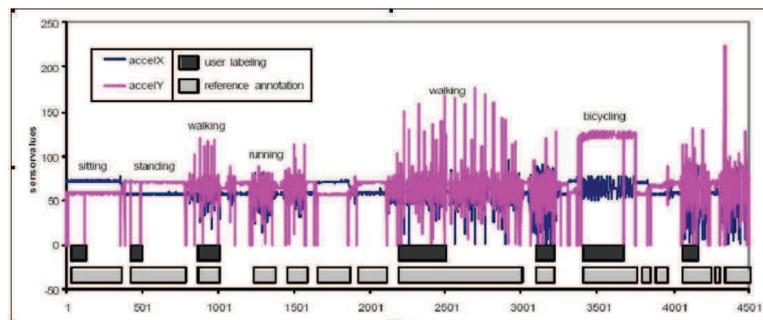


Figure 2.2 – Une méthode de reconnaissance du contexte

Beaucoup d'autres méthodes de reconnaissance [49, 83, 84] ont été exploitées en littérature pour la reconnaissance du contexte. Le logiciel WEKA ⁸ regroupe de

⁸<http://www.cs.waikato.ac.nz/ml/weka/book.html>

nombreuses méthodes de fouille qui peuvent être appliquées à la reconnaissance du contexte.

Après avoir présenté ces méthodes, nous constatons que le contexte est limité aux données capteurs sous une forme numérique. Les algorithmes de reconnaissance donnent généralement de bons résultats. Cela les rend centraux dans le développement d'applications sensibles au contexte en permettant de passer de données brutes de capteurs à des états (i.e. marcher, parler) que nous pouvons qualifier de données significatives facilement exploitables par les applications. Malgré cela, le but de la reconnaissance est de reconnaître un fait pour lequel la méthode est développée et non de pouvoir interpréter les données prises au hasard. Cette particularité limite le champs d'application de cette méthode.

5.4 Prédiction

De manière générale, la prédiction consiste à trouver la relation dans les données de contexte entre un ou plusieurs contextes passés et un contexte futur. Ainsi, il serait possible de prédire ce contexte à partir de la reconnaissance des contextes passés et du contexte courant. Par rapport aux techniques précédentes, cette méthode est très utile pour l'anticipation des actions dans un environnement.

Le travail de R. Mayrhofer (2005) [93] est le plus connu dans ce domaine. L'auteur propose d'inférer des contextes futurs à partir des contextes passés et observés. Dans ce travail remarquable sur la prédiction du contexte, l'auteur distingue la prédiction des contextes à venir et la reconnaissance des contextes qui représentent la situation en cours. La prédiction est achevée par la recherche de modèles communs dans les données de capteurs en utilisant des techniques de classification [93]. Ces modèles sont interprétés comme "des états" d'une machine et agissent comme identificateurs du contexte par exemple : " dès que la personne se réveille le matin, elle monte les volets des fenêtres". L'exemple comprend deux états le premier désigne l'identificateur et le second désigne le contexte prédit. Au fur et à mesure que l'utilisateur change de contexte, des états nouveaux devien-

dront actifs indiquant les contextes à venir.

Nous pouvons citer également la méthode en [129], "Extended Lifelong Growing Neural Gas". Cette technique est assimilée à une classification de plusieurs séries temporelles de données capteurs. Son résultat de prédiction est utilisé par des règles appelées "*règles de contexte*" pour effectuer des actions.

Cependant, le passage d'une méthode non supervisée hors ligne à une méthode supervisée en ligne peut poser des difficultés lors de développement d'applications multi-utilisateurs. En fait, les modèles entraînés deviennent rapidement dépendants des types de capteurs utilisés et de la personne qui a servi à l'entraînement.

D'autres techniques [8] ont été utilisées dans la prédiction du contexte comme l'apprentissage automatique pour la prédiction séquentielle, les réseaux de neurones, les chaînes de Markov et les réseaux de Bayes. Une analyse de ces techniques a été effectuée par A. Boytsov *et al.* (2009) [8] : les auteurs veulent montrer que ces techniques pourraient être des outils suffisamment efficaces, associées à des algorithmes appropriés pour extraire des descriptions du contexte. Les auteurs ne précisent pas ces descriptions et affirment que cette approche est faisable, applicable et donne de bons résultats. Malgré cela, la prédiction reste une méthode particulière dont le champ d'application n'a pas atteint l'espoir attendu.

5.5 Découverte

La découverte du contexte est le processus d'extraction de connaissances à partir de grandes quantités de données de contexte par des techniques provenant du domaine de la fouille de données. Le fait d'utiliser de telles méthodes de ce domaine dans le cadre de la sensibilité au contexte et de les appliquer ou de les améliorer constitue en soi un grand défi notamment par rapport au niveau de description du contexte, ses caractéristiques et son étendue. La fouille peut faire appel aux techniques de reconnaissance et de prédiction citées plus haut. Elle se distingue cependant par l'absence de restriction sur le type de données et sur les

contextes à découvrir. Laisant le champ ouvert à l'exploration des données, une interprétation manuelle ou automatique des résultats retenus est nécessaire.

Les seuls travaux que nous avons classés sous cette thématique sont ceux de Heierman et Diana (2003) [98]. Les auteurs ont cherché à déterminer les épisodes significatifs dans les données séquentielles. Ce sont les épisodes les plus fréquents. L'algorithme choisit un épisode comme significatif s'il se produit un nombre minimum de fois. Ce travail vise à améliorer les solutions en domotique en découvrant les habitudes d'utilisation régulière dans un flux de données recueillies auprès des interactions des utilisateurs avec les appareils électriques comme ceux de l'électroménager. Ces tendances, si elles sont prévisibles, peuvent être automatisées pour l'utilisateur ou même améliorées d'une façon qui permet d'obtenir le même résultat final en moins de temps et moins d'effort.

Toutefois, ce travail se caractérise par le fait que les données sont recueillies à partir d'un dispositif particulier indépendamment du reste de l'environnement, ce qui signifie que l'influence des autres facteurs environnementaux est ignorée. Donc, l'automatisation de l'interaction est quelque part incomplète car seule la manière d'utiliser l'appareil est pris en compte dans le processus d'automatisation. Le contexte de l'utilisateur et de son environnement ne contribuent pas à ce processus. Ce procédé peut avoir des effets indésirables pour l'utilisateur comme par exemple le réglage du chauffage à 20 degrés comme chaque soir à 18h, alors que la personne va passer la nuit chez quelqu'un d'autre ce soir là sans passer par chez lui.

Nous pensons que cette méthode est intéressante pour nos travaux et mérite de l'attention mais elle ne peut pas toute seule répondre à la problématique d'adaptation des services. D'autres aspects sont encore absents auxquels l'état de l'art essaie d'apporter des réponses.

6 Relation entre description et traitement du contexte

Dans cette section, nous mettons en correspondance les techniques de traitement et les niveaux de description du contexte (cf. figure 2.3). Sur cette figure, les flèches avec les petits cercles montrent le passage d'un niveau de description à un autre en utilisant le traitement du contexte correspondant. Par exemple, la reconnaissance peut à partir de données brutes de capteurs déterminer un contexte significatif (la posture d'une personne par exemple). Un contexte significatif peut aussi être déterminé en effectuant un test simple. En effet, ce traitement est une règle du type **si-alors-sinon** qui permet d'attribuer un contexte significatif à une valeur numérique de capteur. Il n'a pas été présenté comme une technique de traitement du contexte vu sa simplicité. Par exemple, si la valeur du capteur est supérieure à 35, cela signifie une forte chaleur. La reconnaissance peut de son côté déterminer une situation, impossible à accomplir par une technique de test, comme une situation de réunion par exemple.

A la base des techniques présentées plus haut et en suivant le raisonnement que nous venons de faire, nous remarquons sur la figure que *situation* est le niveau de description le plus atteignable à partir du contexte brut. Nous choisissons d'exploiter ce niveau de description pour une adaptation automatique de services. Cependant, la situation est un concept très large et difficile à cerner. Dans le prochain chapitre, nous revenons sur ce point pour définir concrètement les situations pertinentes pour l'adaptation de services et nous déterminons également la technique qui permet de l'obtenir à partir du contexte brut.

7 Synthèse

Dans cette section, nous donnons notre avis sur les travaux cités précédemment et nous rassemblons les exigences relatives à la proposition d'une solution informatique d'utilisation du contexte par rapport à notre problématique d'adaptation de services.

Les travaux actuels mettent en avant d'autres insuffisances dans l'utilisation du contexte par les applications. Nous en citons ci-dessous quelques unes.

Dey *et al.* (2001) [42] mentionnent que l'écart entre la sémantique de l'application et le contexte brut conduit à une perte de généralité. Les capteurs sont ainsi difficiles à réutiliser dans d'autres applications et difficiles à utiliser simultanément dans plusieurs applications. Comme le constate Ceri *et al.* [27], le modèle d'une application réelle varie en fonction du domaine d'application et de l'objectif de sensibilité au contexte. Aucun modèle universel n'existe actuellement qui s'appliquerait à toutes sortes d'applications. Les applications adaptatives font ainsi usage de modèles fixes, limitant leur extensibilité à de nouveaux besoins d'adaptation [134].

De manière similaire, Karen *et al.* [80] mettent en évidence l'écart important entre la sortie des capteurs et la description du contexte, utile pour les applications. Cet écart est comblé par les différents types de traitement qui peuvent s'organiser en différentes couches [42]. Le développement de telles applications nécessite des systèmes de gestion du contexte capables de stocker et de partager les données de contexte afin d'extraire plusieurs niveaux de description sans définir de modèles préalables. Selon cette vision, nous présentons un ensemble de besoins en terme de représentation des données du contexte qui serviront de base pour le développement d'un modèle de données de contexte adéquat.

Nous avons mis le doigt dans ce chapitre sur deux points importants et complémentaires qui n'ont pas eu assez d'attention en littérature. Le premier recommande la modélisation sémantique pour les données de contexte non seulement à cause de ses multiples avantages mais aussi parce qu'elle est présente à différents niveaux de description du contexte. De ce fait, le passage d'un niveau à un autre sera plus fluide. Cependant, malgré l'apport de la modélisation du contexte selon l'approche sémantique ou les ontologies, les travaux existants ne sont pas penchés sur la question de l'archivage de ces données car ils manquent de vision sur leur exploitation.

Le deuxième point défend une approche par les données de contexte non seulement à cause des inconvénients des approches métiers du contexte mais aussi parce que l'approche par les données est largement supportée par les techniques de traitements du contexte. Toutefois, il existe peu de techniques qui manipulent des données de contexte sous format sémantique car cette tendance est encore nouvelle.

Nous concluons ainsi qu'une approche par les données sémantiques du contexte se présente comme une candidate intéressante à l'adaptation de services mais pose un vrai défi d'assembler à la fois la sémantique et ses avantages, et l'apprentissage sur les données et d'autres défis que nous résumons dans ce qui suit :

1. **Représentation** : le choix du modèle et du langage de représentation d'une donnée de contexte selon la représentation sémantique nécessite beaucoup de réflexion considérant la technique qui va traiter ces données.
2. **Temporalité** : le temps est une caractéristique importante du contexte comme cela a été précisé dans les sections 1.2 et 3.2.
3. **Qualité** : le modèle de données de contexte devrait prendre en compte l'incertitude des données de contexte. Nous avons discuté ce point dans la section 3.2.
4. **Intégration et fusion** : un rôle clef pour le modèle de données de contexte est de permettre aux données d'être fusionnées ou intégrées à d'autres données et cela à différents niveaux d'abstraction, en se basant sur des ressemblances conceptuelles ou temporelles. D'abord, le traitement de la redondance des données (par exemple, une même donnée pourrait apparaître plusieurs fois à des instants différents).
5. **Flexibilité** : le modèle de données de contexte doit être flexible pour qu'il puisse intégrer les nouveaux besoins des applications.

6. **Stockage et récupération** : vu le nombre important des données de contexte, des solutions de stockage s'avèrent nécessaires. Ces solutions devraient être compatibles avec les solutions existantes afin de bénéficier de l'expérience de ces systèmes dans la manipulation de grands volumes de données.

CHAPITRE 3

SENSIBILITÉ AU CONTEXTE & ADAPTATION

Ce chapitre analyse les limites actuelles de l'adaptation des services sensibles au contexte au regard des différentes approches de modélisation et de traitement du contexte vues dans le chapitre précédent. Nous analysons plus particulièrement la relation entre les données de contexte et la connaissance nécessaire à l'adaptation des services et mettons ainsi en avant le besoin d'une approche de découverte du contexte pour l'adaptation automatique des services sur la base des techniques de fouille de données.

La section 1 met en évidence la sensibilité au contexte pour l'adaptation. Dans la section 2, nous analysons les travaux existants d'adaptation de services sensibles au contexte et nous les classons par type de services. A la base de cette analyse, nous montrons en section 3 les limites de ces travaux. En réponse à ces limites, nous mettons en avant le besoin d'une approche de découverte du contexte pour l'adaptation automatique des services en section 4. Dans la section 5, nous ferons une synthèse de l'état de l'art en donnant quelques éléments qui fondent notre approche de découverte de contexte pour l'adaptation automatique de services que nous développons dans la seconde partie de ce document.

1 Contexte et adaptation

Dans les précédents chapitres, nous avons montré l'importance du contexte pour l'adaptation des services. Dans la littérature, la prise en compte du contexte prend une étendue plus large sous le nom de *sensibilité au contexte*. Nous distinguons alors deux approches selon l'utilisation ou non de la sensibilité au contexte pour faire de l'adaptation.

La première approche est issue des travaux de R. Mayrhofer (2005) en [93] où est défendue l'idée que l'adaptation n'est pas le seul objectif du développement

des systèmes sensibles au contexte. La prévention et la planification sont d'autres champs d'utilisation du contexte. Par exemple, la surveillance de l'état de santé d'un malade à l'aide de capteurs biologiques ou la correction automatique d'un plan de rendez-vous sur un agenda électronique. Néanmoins, cette distinction faite par l'auteur est introduite pour motiver l'utilisation de mécanismes de prédiction dans leurs travaux. Cela ne change en rien le fait que la prévention et la planification soient accomplies en utilisant des mécanismes d'adaptation.

Dans la seconde approche, l'adaptation représente la motivation principale de l'utilisation de la sensibilité au contexte notamment dans le cas des services. Dans [47], les auteurs donnent plusieurs exemples de services tels que la délivrance de messages, le contrôle d'accès, les services pour les personnes dépendantes (orientation dans un bâtiment,...), la gestion de l'infrastructure technique (énergie, barrière etc...). Ces services sont définis par Kirsch-Pinheiro M. *et al.* (2008) [89] comme des services dont la description est enrichie avec les informations du contexte relatives à l'environnement d'exécution des services et à leur adaptation. Cette définition nous semble trop restrictive car enrichir la description des services avec des informations du contexte suppose l'utilisation d'une syntaxe bien définie pour décrire les services. Ce qui veut dire que le contexte qui sert à l'adaptation de ces services est limité aux possibilités offertes par cette syntaxe. Par contre, la définition de [96] qui se focalise sur les services utilisateur en définissant les services sensibles au contexte comme des services qui exploitent les informations de contexte durant la fourniture du service à l'utilisateur, elle s'accorde avec notre vision de l'adaptation définie au chapitre 1 sauf qu'elle ne fait pas apparaître la nature du contexte à utiliser dans l'adaptation.

Dans ce qui suit, nous allons nous intéresser plus particulièrement à la sensibilité au contexte pour l'adaptation des services afin d'en déduire les limites qui freinent l'adaptation des services dans ces travaux.

2 Services sensibles au contexte

Nous analysons dans cette section plusieurs travaux d'adaptation de services sensibles au contexte en donnant plus d'attention au contexte et au mécanisme d'adaptation utilisés. Nous pourrions ainsi compléter notre analyse et en déduire les limites pour l'adaptation de ces services.

Les services sensibles au contexte connaissent un grand essor de nos jours, par exemple dans la sécurité [62], le contrôle de la vie privée [117, 143] et pour l'adaptation de contenus [58]. Nous n'allons pas couvrir tous les champs d'application mais nous nous limitons à l'essentiel de la littérature en distinguant quatre grandes familles de services.

2.1 Services de contenu

Ces services ont pour but de mettre à disposition différents types de données à l'utilisateur comme par exemple, des documents, des vidéos, etc. Le contexte est utilisé par ces services pour aider à délivrer ces contenus selon le contexte de l'utilisateur, par exemple en fonction de sa localisation et ses préférences. P. Ariel *et al.* (2003) [11] ont proposé un service sans fil de consultation de contenus en fonction du temps, de la localisation et de l'environnement physique (météo, ...) de l'utilisateur. Les descriptions du contexte et la délivrance de contenus sont basées sur la technologie XML. Toutefois, ce service n'exploite pas l'historique des consultations précédentes de contenu de l'utilisateur.

2.2 Services de contrôle

Ces services permettent d'agir sur certains dispositifs de manière automatique afin de soulager l'utilisateur de tâches souvent pénibles pour certaines catégories de personnes. Par exemple, l'aide à domicile emploie ces services pour assister les personnes âgées dans leurs tâches quotidiennes. Le travail de F. Kawsar *et al.* (2006) [81] permet de personnaliser des services de contrôles d'éclairage se-

lon les préférences de l'utilisateur et ses besoins. Leur système est basé sur une API (*Application Programming Interface*) de gestion de préférences développée à la suite d'une étude sur les profils d'utilisateurs participants. Le contexte dans ces conditions est limité au profil de personnes qui sont généralement statiques mais qui sont susceptibles de changer car ils contiennent souvent des préférences dynamiques d'utilisateurs.

2.3 Services de communication

Les services de communication sont les plus répandus actuellement, encouragés par l'évolution des moyens de télécommunications tels que la VoixIP. Les auteurs en [139] proposent une personnalisation de services d'informations vocales (par exemple, informations sur le trafic aérien) selon les préférences des utilisateurs et leurs statuts (occupé, en train de lire...) et le langage utilisé. Ce système est basé sur un mécanisme de règles logiques.

Le travail proposé en [24] utilise un mécanisme différent pour faire de la recommandation pour un service de radios sur Internet selon les choix utilisateurs en cours. Ce mécanisme emploie des mesures de similarité pour faire une comparaison entre les choix utilisateurs et les recommandations disponibles. De cette façon le service recommande les radios qui s'accordent avec les choix donnés. On se rapproche plus d'un système de sélection automatique que de recommandation.

Pour ces deux services, le contexte utilisé est très limité et restreint à des informations statiques sur l'utilisateur (profil, choix, préférences, ...) qui sont pourtant susceptibles de changer fréquemment. En outre, les choix ou préférences passés de l'utilisateur ne contribuent en aucune manière à la constitution du contexte.

2.4 Services de commerce électronique

Ce dernier type de services est largement répandu à cause notamment de l'expansion de l'utilisation d'Internet n'importe où et depuis des dispositifs variés.

Par exemple, les services de réservation [91], ventes-achats [99], achats [110], restauration [77], etc.

Les mécanismes d'adaptation utilisés sont similaires à ceux que nous avons présentés au premier chapitre. Nous remarquons cependant que les travaux [110] et [77] exploitent l'historique des informations relatives à l'utilisation des services fournis.

3 Limites de l'adaptation des services au contexte

Dans cette section, nous nous basons sur les travaux vus dans la section précédente pour montrer deux limites principales à l'adaptation des services sensibles au contexte : l'incapacité de prendre en compte différents types de situations et la non exploitation de l'historique du contexte dans la mise oeuvre de l'adaptation.

Le tableau 3.1 résume les travaux [11, 24, 77, 81, 91, 99, 110, 139] en précisant le type d'adaptation, le contexte utilisé, le mécanisme d'adaptation et en ajoutant le critère "*exploitation de l'historique du contexte*".

Nous présentons d'abord deux constatations sur ce tableau puis nous détaillons les limites en deux sous-sections. La première constatation est que ces travaux couvrent bien les types d'adaptation que nous avons présentés au premier chapitre sauf pour le cas de la reconfiguration. En effet, la reconfiguration prend en compte les conditions internes ou externes au système, ce qui n'est pas le cas pour ces travaux. On rappelle bien la séparation faite dans la figure 1.1 qui montre l'adaptation par rapport à l'environnement système ou utilisateur. La deuxième constatation concerne les mécanismes d'adaptation employés par ces travaux. Nous avons bien couvert ces mécanismes au chapitre 1. Cependant, l'adaptation à base de règles paraît comme la plus répandue.

Service	Type d'adaptation	Contexte	Technique d'adaptation	Exploitation de l'Historique du contexte	Référence du travail
Service Web sans fil	Adaptation de contenu	Dispositifs, temps, date, localisation, environnement physique (météo, etc.)	Technologie basée sur XML	Aucun historique utilisé	P. Ariel et al. (2003)
Service au domicile, Télé, éclairage, etc.	Personnalisation	Préférences des utilisateurs, préférences de contrôle	Gestion de préférences basée sur API d'applications	Etude sur les profils des participants	F. Kawsar et al. (2006)
Service Radio internet	Recommandation	Les choix utilisateurs en cours	Calcul de similarité	Aucun historique utilisé	P. Cunningham et C. Hayes (2002)
Service Web, réservation d'hôtel, voyage, etc.	Adaptation de contenu	Localisation, destination du voyage, etc.	Plateforme de développement	Aucun historique utilisé	K. Markus et K. Alfons (2004)
Service Web, VoIP, information, trafic aérien, etc.	Personnalisation de la composition des services	Préférences utilisateurs, état (occupé, libre), langage utilisé	Déclenchement à base de règles	Aucun historique utilisé	Y. Yang et al. (2006)
Services web de commerce électronique	Recommandation	Profil utilisateur, localisation, informations sur la météo	Système à base de génération de règles	Aucun historique utilisé	K. Ohbyung et K. Jihoon (2009)
Service web de restauration	Personnalisation	Informations personnelles utilisateurs, activités, etc.	Arbre de décision	Historique des informations utilisateurs ont utilisés le service	H. Jongyi et al. (2009)
Services web de shopping	Personnalisation de la recommandation	Profil des centres d'intérêts utilisateurs	Analyse de similarités	Historique des transactions	W. Quan and H. Jianmin (2006)

Figure 3.1 – Analyse de services sensibles au contexte

3.1 Limites via-à-vis du contexte

Nous avons montré que le niveau de description "situation" est le niveau le plus atteignable à partir du contexte brut. Seulement, en regardant le contexte dans les travaux du tableau 3.1, il existe un manque clair dans l'exploitation des techniques de traitement du contexte notamment pour les niveaux description de situation et de compréhension. Nous remarquons que les contextes les plus complexes utilisés sont potentiellement des descriptions statiques ou non complètes comme des profils, des préférences ou des activités de l'utilisateur. Il existe cependant plusieurs types de situations qui caractérisent l'utilisateur et son environnement qui ne sont pas prises en compte, comme par exemple les habitudes d'utilisation d'un service par l'utilisateur. Ceci s'explique notamment par le fait que plusieurs types de situations sont quasiment impossibles à déterminer sans l'historique du contexte ou l'utilisation de techniques de traitement appropriées. Il est à noter que nous n'avons pas pu identifier de travaux qui exploitent des compréhensions à des fins de sensibilité au contexte pour les applications ou les services. Nous pensons que ce niveau est plus difficile à décrire que le niveau situation. Des travaux existants [36, 135] montrent cette difficulté à déterminer des descriptions du contexte aux niveaux situation et compréhension.

Nous distinguons alors trois types de situations selon le niveau de description du contexte "*situation*" (**situation fréquente, situation rare, tendance**) ; et deux autres descriptions plus complexes selon le niveau "*compréhension*" (**situations dépendantes, épisode**). Nous les présentons dans ce qui suit.

3.1.1 Situation fréquente

Une *situation fréquente* est une situation qui se produit souvent dans l'environnement et qui tend à devenir une habitude. Une telle situation est liée à un instant ou à un lieu donné, ou les deux à la fois. Un exemple d'une telle situation est : "la personne descend les volets chaque soir autour de 22h".

3.1.2 Situation rare

Une *situation rare* se détermine par rapport à une situation habituelle. Par exemple, la consommation énergétique est 200 watt/heure alors que d'habitude elle est de 50 Watt/heure, le matin, quand il n'y a personne dans la maison. Autre exemple : imaginons par exemple une personne qui invite ses amis chez lui pour dîner sans spécifier cela dans son agenda alors qu'elle le fait habituellement.

3.1.3 Tendance

Une *tendance* est une situation dans laquelle le système juge probable d'intervenir dans les prochaines heures. Un exemple peut être le fait qu'il est fort probable qu'aujourd'hui l'utilisateur prenne un dîner dehors, car il a reçu une invitation et donc il n'est pas nécessaire de commander des pizzas.

3.1.4 Situations dépendantes

Les *situations dépendantes* sont des situations qui se succèdent dans le temps mais qui ne sont pas consécutifs. Par exemple, la dépendance entre un appel téléphonique effectué et l'arrivée d'une personne.

3.1.5 Episode

Un *épisode* est une succession temporelle de situations par exemple, le téléphone mobile sonne, la personne cherche l'endroit où elle a mis son téléphone, la personne répond au téléphone.

3.2 Historique du contexte

L'exploitation de l'historique du contexte est quasiment absente de nombreux travaux et dans un nombre limité de cas elle est limitée aux profils utilisateurs et aux transactions passées. En effet le tableau 3.1 montre cette limitation de la conception de services sensibles au contexte. Auparavant K. Georgia *et al.* (2009)

[88] ont discuté les différentes solutions pour la conception des services sensibles au contexte à des fins de personnalisation ou d'adaptation. L'étude, faite du point de vue ingénierie, a montré que la majorité de ces travaux ne considère pas l'historique du contexte.

Cet avis est aussi partagé par M. Baldauf *et al.* (2007) [13] en analysant des systèmes sensibles au contexte. Cet analyse précise que la majorité des systèmes étudiés (the Context Toolkit, CoBrA, CASS, SOCAM et CORTEX) ont la possibilité d'enregistrer les données du contexte dans une base de données persistante. Par contre, les modèles de ces systèmes sont développés au niveau conceptuel et non extraits de cet historique. Une autre analyse en [93] confirme ce résultat et montre que les projets cités ne traitent pas les motifs fréquents ou périodiques.

4 Besoin d'une approche de découverte du contexte

Au regard des techniques de traitement du contexte, nous nous demandons si l'une d'entre elles peut servir à la détermination des situations à partir de l'historique du contexte et comment les utiliser dans l'adaptation des services. Pour justifier le choix d'une technique de traitement du contexte parmi d'autres, nous reprenons les techniques présentées au chapitre 2 pour les comparer par rapport à leur capacité à trouver les situations que nous venons de présenter. La figure 3.2 étend la figure 2.3 avec les situations par niveau de description du contexte.

Par exemple, en utilisant une technique de reconnaissance sur des données capteurs provenant d'un microphone et de capteurs infrarouges pour la localisation, l'algorithme peut détecter qu'une personne est en situation de réunion ou en discussion avec un groupe de personnes.

Ceci dit, le tableau montre clairement le potentiel de la découverte du contexte comme technique d'extraction des situations à partir de l'historique du contexte. Cela nous semble une approche qui apporte des réponses aux deux limites précédentes.

Prise en compte du contexte

La section précédente apporte une bonne suggestion en présentant la découverte du contexte comme une technique capable de déterminer des situations intéressantes pour l'adaptation de services. Dans la suite de cette thèse nous nous focalisons sur les descriptions du contexte au niveau *situation* sans prendre en compte le contexte au niveau compréhension c'est-à-dire nous ne retenons pas les situations dépendantes et les épisodes. *Nous considérons le contexte comme étant une description d'une situation qu'on peut déterminer automatiquement à partir de données dites de contexte. Les situations qui nous intéressent sont les situations fréquentes et les situations rares identifiées plus haut dans ce chapitre.*

Nous donnons dans ce qui suit les éléments qui fondent notre approche d'adaptation automatique de services.

Adaptation des services au contexte

L'étude des méthodes de traitement du contexte au chapitre 2 montre que les modèles d'adaptation basés sur la reconnaissance du contexte permettent à un service d'agir ou de s'ajuster si la situation pour laquelle le modèle a été entraîné, est détectée. Les modèles d'adaptation basés sur la prédiction permettent au service d'agir ou de s'ajuster quand le service est dans l'état prédit. On peut noter que l'utilisation de ce type de méthodes (reconnaissance et prédiction) pour l'adaptation de services ne permet cependant pas aux services d'identifier par eux-mêmes les raisons pour lesquelles l'adaptation est effectuée.

Pour mieux illustrer, nous présentons un exemple d'activation du mode silencieux d'un téléphone mobile (TM) quand l'utilisateur est en réunion peut se faire de deux façons différentes :

L'option est offerte par le concepteur dans le TM. Elle nécessite un modèle bien entraîné pour reconnaître la situation de réunion à partir de données capteurs tels que, microphone intégré, accéléromètres, etc. Ainsi, quand l'utilisateur est en

réunion le téléphone se met en mode silencieux automatiquement.

L'option n'est pas prévue dans le TM. Le TM analyse l'historique des conditions d'utilisation de l'option silencieux du TM par l'utilisateur (en réunion) puis détermine que l'utilisateur met son TM silencieux quand il est en réunion. Ainsi, les prochaines fois, le TM se met en mode silencieux de manière autonome.

La différence entre les deux procédés est que le deuxième exploite l'historique du contexte de manière automatique pour déterminer les situations pertinentes pour l'adaptation de services. Les modèles pour ce deuxième cas se construisent et se corrigent au fur et à mesure de la vie du système à côté de l'utilisateur, ce qui le rend plus autonome et adaptatif.

Motivations de notre approche

Nous résumons les motivations de cette nouvelle approche en deux points : l'autonomie et la robustesse. L'autonomie signifie que le système dépend uniquement de ses capacités perceptives et non d'un but précis qui lui a été assigné a priori. En d'autres termes, l'utilisation du service et la situation où il est utilisé sont toutes deux prises en compte dans le processus d'adaptation, ni l'action d'utiliser un tel service, ni la situation dans laquelle il sera utilisé ne sont prédéfinies dans le système. Le système est capable tout seul de déterminer dans quelle situation il doit agir et avec quel type de service. En analysant l'historique des données du contexte, le système détermine la pertinence d'un service pour un type de situation particulière.

La robustesse est confrontée aux changements des paramètres d'adaptation. En effet, le changement dans les paramètres d'adaptation ne nécessite pas un nouveau développement du modèle même si un paramètre n'a pas été prévu au départ. Considérant l'exemple précédent du TM, le fait de ne pas considérer la *date* dans le modèle de départ, nécessite normalement un nouveau modèle de reconnaissance de situation. Par contre, si le système exploite l'historique de manière automatique, la situation sera raffinée en déterminant que l'utilisateur met son

mobile en mode silencieux uniquement pendant les réunions du *jeudi*.

Mise en perspective

Il existe une cohérence et une complémentarité très claires entre les réponses apportées aux problématiques ci-dessus. La découverte du contexte permet de retrouver de nouvelles situations qui de leur côté contribuent à un mécanisme d'adaptation des services. Nous proposons dans la suite de ce document un cadre de développement pour l'adaptation au contexte des services en utilisant une approche de découverte du contexte pour un ensemble ouvert d'applications. Nous répondrons à une question de recherche nouvelle qui est : quelles sont les techniques de fouilles de données qui permettent la découverte du contexte et comment les intégrer dans la conception des services adaptatifs et sensibles au contexte ? On se basera sur la sémantique des données pour obtenir un modèle générique de données de contexte.

Deuxième partie

Proposition

CHAPITRE 4

MODÉLISATION DU CONTEXTE

Nous présentons dans ce chapitre la modélisation sur laquelle s'appuie notre système de découverte du contexte pour l'adaptation de services. Contrairement aux travaux présentés au chapitre 2 qui proposent des modèles conceptuels du contexte où les modèles de contexte et les applications sont étroitement liés, l'originalité de cette modélisation est de considérer le contexte à son niveau données, à la manière des technologies qui permettent d'organiser et de structurer de très grandes quantités de données. L'idée derrière cette modélisation est de mettre en place des techniques de fouille de données permettant la découverte automatique des contextes relatifs à l'usage des services dans les espaces intelligents.

La section 1 décrit le modèle d'une donnée de contexte. La section 2 présente le modèle du contexte. La section 3 définit des fonctions de mesures de similarité propres à cette modélisation, qui serviront à regrouper des données de contexte ou des contextes pour la construction de situations. Nous terminons ce chapitre par une conclusion.

1 Modèle sémantique d'une donnée de contexte

Nous proposons dans ce chapitre un modèle de données de contexte basé sur les technologies du web sémantique et sur lequel repose le fonctionnement de notre système d'adaptation automatique de services. Le choix d'une modélisation sémantique du contexte brut est d'intégrer dans une même représentation la donnée et ses caractéristiques de base : sa signification, sa confiance, le moment où elle a été produite, l'entité concernée par la donnée et la source émettrice de la donnée. En d'autres termes la donnée de contexte contient toutes les informations utiles pour la caractériser pleinement. Nous rappelons la discussion qu'on a eue au chapitre 2 à la section 4.2.4, où nous avons montré que la représentation

sémantique est plus appropriée pour la représentation du contexte brut.

Ce modèle a pour but de répondre au problème d'hétérogénéité des données en provenance de sources multiples et de rendre notre système plus robuste à la dynamique des sources dans l'espace intelligent, principalement la disparition ou l'arrivée de nouvelles sources dans l'environnement. Il est aussi construit de façon à ce que le regroupement de données soit possible en utilisant des mesures de similarité sémantiques, syntaxiques et temporelles.

Cette modélisation est également choisie afin qu'elle aide à l'autonomie et à la généralité du système à développer. Nous reviendrons dans le chapitre architecture sur ces deux aspects pour vérifier la capacité de cette modélisation à répondre à ces critères. Nous nous focalisons pour l'instant sur la description du modèle d'une donnée de contexte.

1.1 Modèle d'une donnée de contexte

Le modèle \mathcal{M} que nous allons présenter nécessite une ontologie de contexte \mathcal{O} . Cette ontologie a pour objectif de fournir un vocabulaire commun de l'espace intelligent. Elle décrit de manière générique les données provenant de sources de contexte multiples, ainsi que les services à l'utilisateur qui sont embarqués dans l'espace intelligent. Nous utilisons \mathcal{D} pour désigner l'ensemble de ces données et nous notons d_i une donnée de cet ensemble.

Soit dc une déclaration RDF ¹ de la forme $(sub, pred, obj)$, où sub , $pred$ et obj représentent respectivement le sujet, le prédicat et l'objet de la déclaration. RDF est un langage qui a été initialement conçu pour décrire les informations sur les ressources Web. Le sujet représente la ressource à décrire ; le prédicat représente la propriété applicable à cette ressource, et l'objet représente la valeur attribuée à la propriété pour cette ressource. Cette valeur peut être une ressource ou un littéral. Nous définissons alors le modèle d'une donnée de contexte comme suit :

¹<http://www.w3.org/RDF/>

Définition 1.1.1 (Modèle d'une donnée de contexte). Le modèle $\mathcal{M} = (\mathcal{T}, \mathcal{O})$ d'une donnée de contexte est composé d'une ontologie \mathcal{O} , et d'un ensemble \mathcal{T} de sept déclarations RDF représenté par le tableau 4.I, où chaque sujet et chaque prédicat est défini dans l'ontologie du contexte \mathcal{O}

L'ensemble de ces sept triplets constitue un graphe RDF. Ce graphe décrit une ressource appelée *contextData* sur laquelle sont appliqués les prédicats suivants : *hascontext*, *isContextOf*, *hasSource*, *hasConfidence* et *hasTimestamp*. *cxt :contextData* est la ressource qui désigne la donnée de contexte d_i dans son ensemble. Les valeurs des prédicats *hasSource*, *hasConfidence* et *hasTimestamp* sont des littéraux qui représentent respectivement un identifiant de la source qui a fourni la donnée de contexte, une valeur réelle indiquant le pourcentage de confiance qui lui a été attribué et en dernier une valeur temporelle exprimant la date et l'heure auxquelles la donnée a été perçue ou générée par la source. Les valeurs des prédicats restants *hascontext* et *isContextOf* sont des ressources qui décrivent respectivement la valeur de contexte et l'entité concernée. A chacune de ces ressources est appliqué un prédicat *hasIdentifier* qui prend un littéral comme valeur. La figure 4.1 illustre le modèle d'une donnée de contexte.

Le choix de RDF comme un modèle de représentation de données de contexte est motivé par les avantages suivants : RDF décrit la sémantique des données à l'aide de prédicats ; RDF permet l'échange de ces données sans ambiguïté entre applications. De par sa représentation sous forme de graphe, RDF permet aux

Tableau 4.I – Liste exhaustive des triplets du modèle \mathcal{M}

Sujet	Prédicat	Objet
<i>cxt :contextData</i>	<i>cxt :isContextOf</i>	<i>cxt :entity</i>
<i>cxt :contextData</i>	<i>cxt :hascontext</i>	<i>cxt :contextValue</i>
<i>cxt :contextData</i>	<i>cxt :hasSource</i>	<i>xsd :string</i>
<i>cxt :contextData</i>	<i>cxt :hasConfidence</i>	<i>xsd :float</i>
<i>cxt :contextData</i>	<i>cxt :hasTimestamp</i>	<i>xsd :dateTime</i>
<i>cxt :entity</i>	<i>cxt :hasIdentifier</i>	<i>xsd :string</i>
<i>cxt :contextValue</i>	<i>cxt :hasIdentifier</i>	<i>xsd :string</i>

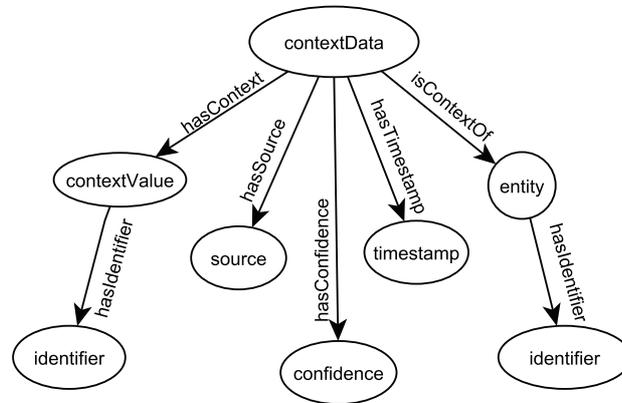


Figure 4.1 – Modèle des données de contexte

données de contexte de former un large graphe sur lequel plusieurs traitements sur les graphes sont possibles dont l’exploration visuelle. De plus, les modèles à base de RDF se présentent comme simples et faciles à utiliser dans les architectures à base de Web services.

1.2 Propriétés d’une donnée de contexte

Les données de contexte sont des descriptions sémantiques du contexte brut qui respectent le modèle précédent. Par exemple, la localisation d’une personne, le statut d’une lampe (allumée, éteinte), etc. Nous définissons alors une donnée de contexte comme suit :

Définition 1.2.1 (Donnée de contexte). *Un ensemble X de 7 triplets RDF est une donnée de contexte dénotée d_i ssi X satisfait la correspondance locale et la correspondance globale au modèle $\mathcal{M} = (\mathcal{T}, \mathcal{O})$.*

La correspondance locale a pour but d’imposer que chaque triplet de la donnée de contexte corresponde à un triplet du modèle. Quant à la correspondance glo-

bale, elle vérifie que la structure de la donnée s'apparie à la structure du modèle.

Proposition 1.2.1 (Correspondance locale). *Soit \mathcal{O} une ontologie du contexte. Soit X un ensemble de 7 triplets RDF, dénotés (s_j, p_j, o_j) pour $j \in [1..7]$. X correspond localement à un modèle $\mathcal{M} = (\mathcal{T}, \mathcal{O})$ ssi il existe une bijection F notée : $F : X \rightarrow \mathcal{T}$ telle que $\forall (sub, pred, obj) \in \mathcal{T}, \exists! (s_j, p_j, o_j) \in X$ vérifiant :*

- $s_j \in subClassOf^*(sub, \mathcal{O})$.
- si $pred \in ObjectProperty(\mathcal{O})$ alors $p_j \in subPropertyOf^*(pred, \mathcal{O})$ ou, si $pred \in DataProperty(\mathcal{O})$ alors $p_j = pred$
- si $obj \in \{cxt : entity\}$ alors $o_j \in subClassOf^*(obj, \mathcal{O})$ ou, si $obj \in \{cxt : contextValue\}$ alors $o_j \in subClassOf^*(obj, \mathcal{O})$ ou, si $obj \notin \{cxt : entity, cxt : contextValue\}$ alors o_j est un littéral de type obj .

* signifie que la fonction est appliquée au concept ou récursivement à l'un de ses sous-concepts.

Proposition 1.2.2 (Correspondance globale). *Soient $(s_{i,j}, p_{i,j}, o_{i,j}), (s_{i,k}, p_{i,k}, o_{i,k}), j, k \in [1..7]$ et $j \neq k$, deux triplets de \mathcal{T} d'une donnée de contexte d_i qui satisfait la correspondance locale d'un modèle $\mathcal{M} = (\mathcal{T}, \mathcal{O})$. La donnée de contexte satisfait aussi la correspondance globale ssi :*

- $\forall j, k / p_{i,j}, p_{i,k} \in \{cxt : isContextOf, cxt : hasContext, cxt : hasSource, cxt : hasConfidence, cxt : hasTimestamp\}$ alors $s_{i,j} = s_{i,k}$
- si $o_{i,j} \in subClassOf^*(cxt : entity)$ et $s_{i,k} \in subClassOf^*(cxt : entity)$ alors $o_{i,j} = s_{i,k}$
- si $o_{i,j} \in subClassOf^*(cxt : contextValue)$ et $s_{i,k} \in subClassOf^*(cxt : contextValue)$ alors $o_{i,j} = s_{i,k}$

En fait pour comprendre ces deux propositions, nous prenons comme exemple le premier triplet du tableau 4.1 du modèle plus haut (*cxt :contextData, isContextOf, cxt :entity*) et nous vérifions si le triplet (*locationData, isLocationOf, Person*) de la donnée de contexte de la figure 4.3² satisfait la correspondance locale. *locationData* est bien une sous-classe de *cxt : contextData* dans l'ontologie en figure 4.2. *isContextOf* est une propriété d'objet dans l'ontologie et *isLocationOf* est une sous-propriété de celle-ci. *Person* est une sous-classe de *cxt : entity*. Donc, le triplet satisfait la correspondance locale. Pour vérifier la correspondance globale sur l'exemple de la figure 4.3, on considère les triplets (*locationData, isLocationOf, Person*), (*locationData, hasLocation, Room*), (*locationData, hasSource, IR02*), (*locationData, hasConfidence, 90*), (*locationData, hasTimestamp, 2009-09-23T09 :01 :38*) : ils ont tous le même sujet donc la première règle de la correspondance globale est vérifiée. De plus, l'objet (*Person*) du triplet (*locationData, isLocationOf, Person*) est une sous classe du concept *Entity* selon l'ontologie de la figure 4.2 et constitue le sujet du triplet (*Person, hasIdentifier, John*), ce qui correspond à la 2ème règle de la correspondance globale, qui est donc vérifiée aussi. Enfin, l'objet (*Room*) du triplet (*locationData, hasLocation, Room*) est une sous classe du concept *ContextValue* selon l'ontologie de la figure 4.2 et constitue le sujet du triplet (*Room, hasIdentifier, KitchenOven*), ce qui correspond à la 3ème règle de la correspondance globale. Cette troisième règle est ainsi avérée et nous pouvons donc conclure que la correspondance globale est entièrement vérifiée.

²kichenOven désigne l'endroit de la cuisine où il y a le four.

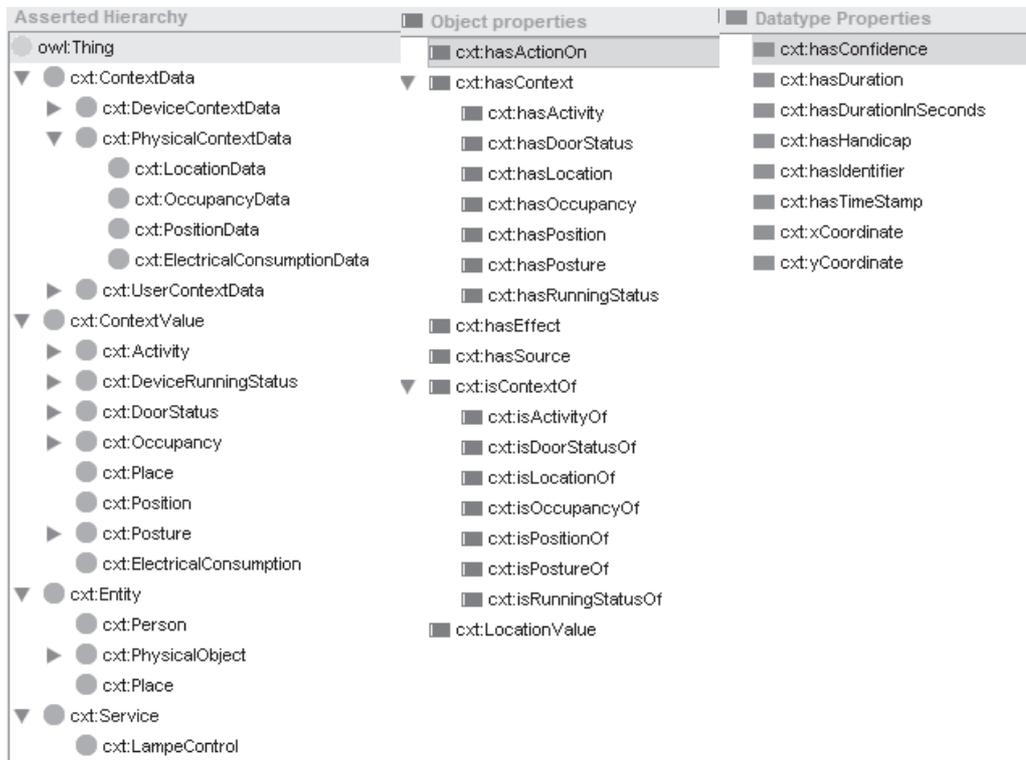


Figure 4.2 – Partie de la structure de l’ontologie \mathcal{O}

1.3 Fonctions de manipulation des données de contexte

Nous définissons un ensemble de fonctions applicables sur les données de contexte que nous utiliserons à la section 3 de ce chapitre pour le calcul de mesures de similarité. Ces fonctions permettent de récupérer les valeurs des objets au sens de RDF de la manière suivante : soit d_i une donnée de contexte instance du modèle $\mathcal{M} = (\mathcal{T}, \mathcal{O})$, on a alors :

- $Entity(d_i)$: rend le concept de l’entité de la donnée.
- $ContextValue(d_i)$: rend le concept de la valeur du contexte de la donnée.
- $EntityIdentifier(d_i)$: rend la chaîne de caractères identifiant l’entité.

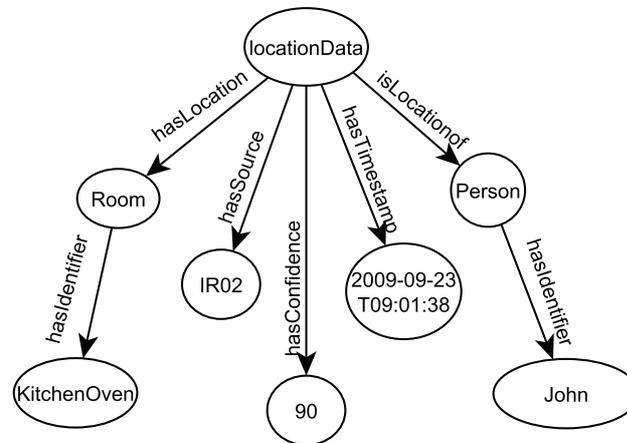


Figure 4.3 – Exemple d’une donnée de contexte

- $ContextValueIdentifier(d_i)$: rend la chaîne de caractère identifiant la valeur du contexte.
- $Source(d_i)$: rend l’identifiant de la source émettrice de la donnée d_i .
- $Timestamp(d_i)$: rend le timestamp auquel a été émise la donnée d_i .
- $Confidence(d_i)$: rend la valeur de confiance attribuée par la source à la donnée d_i .

Exemple 1.3.1. Si on considère la donnée du contexte de la figure 4.3, on peut appliquer ces fonctions comme suit :

- $Entity(cxt : locationData) = Person$
- $ContextValue(cxt : locationData) = Room$
- $EntityIdentifier(cxt : locationData) = "John"$,
- $ContextValueIdentifier(cxt : locationData) = "KitchenOven"$
- $Source(cxt : locationData) = "IR02"$

- $Timestamp(cxt : locationData) = 2009 - 09 - 23T09 : 01 : 38$
- $Confidence(cxt : locationData) = 90$

1.4 Vecteur d'une donnée de contexte

Un vecteur d'une donnée de contexte est constitué des objets (au sens RDF) de cette donnée. La mise sous forme de vecteurs des données de contexte facilitera, dans la suite de ce mémoire, la présentation de certaines définitions et de certains calculs, notamment comme nous le verrons plus loin, en ce qui concerne les mesures de similarité.

Définition 1.4.1 (Vecteur d'une donnée de contexte). Soit d_i une donnée de contexte $\in \mathcal{D}$, on note un vecteur de d_i par $V_{d_i} = (Entity(d_i), ContextValue(d_i), EntityIdentifier(d_i), ContextValueIdentifier(d_i), Source(d_i), Confidence(d_i), Timestamp(d_i))$.

Exemple 1.4.1. Le vecteur de la donnée de contexte de la figure 4.3 est : $V_{locationData} = (Person, Room, "John", "KitchenOven", "IR02", 90, 2009-09-23T09 :01 :38)$

1.5 Données statiques du contexte

Une *donnée statique du contexte* correspond aux données sur la disposition et l'emplacement des entités fixes dans l'espace intelligent ; par exemple, une lampe dans une chambre, une télévision en face d'un canapé, etc. Ces données ne sont pas émises par une source de contexte mais ce sont des connaissances préalables sur l'environnement qui seront utiles dans la construction des contextes comme le montre la donnée de contexte numéro 2 de la figure 4.4. Cependant, pour respecter la représentation du modèle de données de contexte, nous avons choisi trois valeurs particulières pour le temps, la confiance et la source émettrice : le 1er Janvier 1970 à 0 h (valeur à partir de laquelle se fait le décompte dans les systèmes Unix), 100, ADMIN (administrateur) respectivement. Ce choix est dû à la définition préalable de ces données.

2 Modèle sémantique du contexte

Dans la conclusion du chapitre 3, nous avons considéré le contexte comme étant une description d'une situation qu'on peut déterminer automatiquement à partir des données de contexte. En accord avec cette définition, nous proposons de construire des situations sur la base de regroupements de données de contexte en utilisant des liens sémantiques, syntaxiques et temporels. En effet, le modèle d'une donnée de contexte regroupe à la fois des concepts définis par une ontologie, des chaînes de caractères et une valeur temporelle. Dans ce qui suit, nous définissons le modèle du contexte et ses propriétés sur ces fondements.

2.1 Modèle du contexte

En se basant sur les liens syntaxiques, sémantiques et temporels qui peuvent exister entre les données de contexte, nous supposons avoir à disposition une fonction de similarité appelée *Simg* pour le regroupement de ces données (la section 3 de ce chapitre est consacré à ces mesures de similarité). *Simg* est une combinaison des mesures syntaxique, sémantique et temporelle. Ainsi, nous définissons un modèle du contexte comme suit :

Définition 2.1.1 (Contexte). Soit \mathcal{D} doté d'une fonction de similarité *Simg*, C est un contexte de cette espace ssi :

- $C = \{d_i / d_i \in \mathcal{D}, i = 1..N_C\}$ tel que d_i est une donnée de contexte.
- $\forall d_i \in C, \exists d_j \in C / d_j \neq d_i$ et $Simg(d_i, d_j) > \mu$,
 μ est un seuil de similarité acceptable, c'est-à-dire au dessous duquel deux données ne sont pas considérées similaires. C est une valeur qui sera fixée expérimentalement.

Nous nommons ainsi \mathcal{C} , l'ensemble des contextes construits à partir de \mathcal{D} en utilisant la fonction *Simg*. L'exemple de la figure 4.4 montre le contexte "John est

dans la cuisine et la lampe de la cuisine est allumée". Ce contexte est formé de trois données de contexte dont une donnée statique.

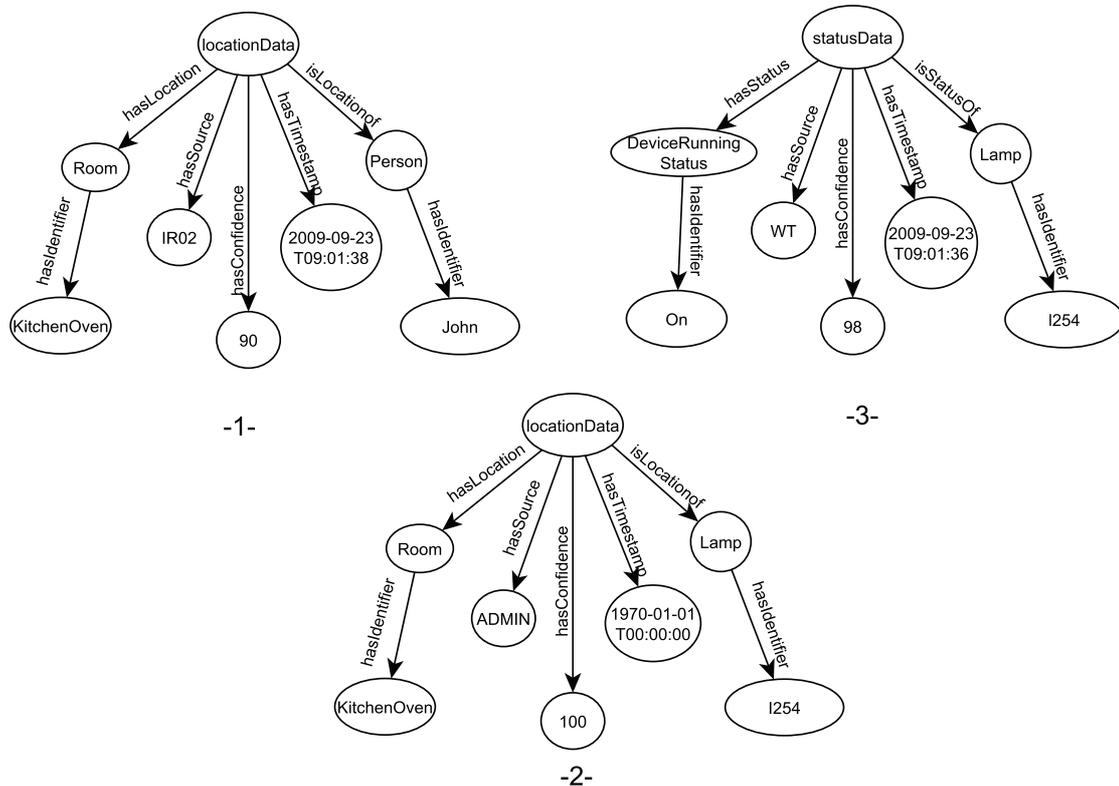


Figure 4.4 – Exemple d'un contexte

2.2 Contexte pur

Plaçons nous dans le cas où nous disposons de $N_{C_{pur}}$ sources de données de contexte. Un contexte pur est alors un ensemble formé de $N_{C_{pur}}$ données de contexte provenant de sources différentes. En d'autres termes, le contexte pur contient une donnée de contexte et une seule en provenance de chaque source. Notamment un contexte pur ne contiendra pas de données statiques de contexte. Nous exploitons ce type particulier de contexte pour la découverte des situations fréquentes au prochain chapitre. Nous le définissons comme suit.

Définition 2.2.1 (Contexte pur). Soit C un contexte $\in \mathcal{C}$, tel que $C = \{d_i/d_i \in \mathcal{D}, i = 1..N_{C_{pur}}\}$, C est un contexte pur ssi $\forall d_i, d_j \in C$ si $i \neq j$ alors $Source(d_i) \neq Source(d_j)$.

Exemple 2.2.1. Les données de contexte -1- et -3- de la figure 4.4 forment un contexte pur.

2.3 Vecteur d'un contexte

Le vecteur d'un contexte contient tous les objets (au sens RDF) appartenant à toutes les données de ce contexte. Il est défini comme le vecteur des vecteurs des données de ce contexte.

Définition 2.3.1 (Vecteur d'un contexte). Soit C un contexte pur $\in \mathcal{C}$, tel que $C = \{d_i/d_i \in \mathcal{D}, i = 1..N_{C_{pur}}\}$. Un vecteur de C est noté $V_C = (V_{d_1}, \dots, V_{d_{N_{C_{pur}}}})$ avec $N_{C_{pur}}$ le nombre de sources de données de contexte.

Exemple 2.3.1. Le vecteur du contexte de la figure 4.4 est : $V_C = ((Person, Room, "John", "KitchenOven", "IR02", 90, 2009-09-23T09 :01 :38), (lamp, DeviceRunningStatus, "I254", "On", "WT", 98, 2009-09-23T09 :01 :36), (lamp, room, "I254", "KitchenOven", "ADMIN", 100, 1970-01-01T00 :00 :00))$

Nous consacrons le reste de ce chapitre aux mesures de similarité qui nous permettront d'effectuer des regroupements sur les données et sur les contextes eux-mêmes. Il est important de les présenter pour faciliter leur compréhension car ces mesures sont propres à notre modélisation.

3 Mesures de similarité

On parlera dans cette section de mesures de similarité entre données de contexte et de mesures de similarité entre contextes purs. Ces mesures sont utilisées pour regrouper des données et des contextes purs. L'objectif est de construire des situations pertinentes pour l'adaptation de services. Une situation est pertinente si elle

contient au moins une donnée qui représente un effet de service (cf. section 5 du chapitre 6).

Les mesures que nous allons détailler dans cette section sont basées sur l'idée développée dans le cadre des mesures de similarité sémantique fondées sur les ontologies [50, 73], les statistiques [79] et la proximité temporelle [92, 140] développée dans le cadre du rapprochement entre événements temporels.

3.1 Similarité entre données de contexte

Nous avons identifié trois types de mesures de similarité entre données de contexte : syntaxique, conceptuelle et proximité temporelle. Nous détaillons chacune d'elles dans les sections suivantes.

3.1.1 Mesure de similarité syntaxique

La similarité syntaxique est calculée sur la base de l'égalité syntaxique entre deux termes. Dans ce travail de thèse, l'égalité syntaxique se réfère à une comparaison de deux chaînes de caractères en utilisant la mesure de distance de Levenshtein [64, 65].

La mesure de distance proposée par Levenshtein est égale au nombre minimal de caractères qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre. En d'autres termes c'est le minimum de la distance d'édition qui transforme un *terme1* en un *terme2*. En pratique, on associe à chacune des opérations un coût. Dans notre cas, on a considéré un coût de 1. La distance de Levenshtein est d'autant plus grande que le nombre de différences entre les deux termes est grand.

Pour calculer la distance syntaxique entre deux données de contexte, on compare les termes appartenant à chacune des données, un par un selon la distance de Levenshtein et ceci uniquement pour les éléments qui sont des chaînes de caractères (les 5 premiers éléments d'un vecteur de données de contexte). Elle est définie comme suit :

Définition 3.1.1 (Mesure de distance syntaxique entre données de contexte).

Soit V_{d_i} et V_{d_j} deux vecteurs associés respectivement aux données de contexte d_i et $d_j \in \mathcal{D}$. Pour calculer la distance syntaxique entre d_i et d_j , on compare les termes appartenant à chacune des données, un par un selon la distance de Levenshtein et ceci uniquement pour les éléments qui sont des chaînes de caractères (les 5 premiers éléments d'un vecteur de données de contexte). Elle est définie comme suit :

$$Dist_{sy}(d_i, d_j) = \sum_{k=1}^5 LevenshteinDist(V_{d_i}(k), V_{d_j}(k)) \quad (4.1)$$

Les deux premiers membres des vecteurs représentent des concepts de l'ontologie de contexte : ils peuvent donc être considérés dans les mesures de similarité soit comme des concepts (avec une mesure conceptuelle, comme nous le verrons ci-dessous), soit comme des chaînes de caractères (avec une mesure syntaxique). Ainsi, dans le premier cas $m=3$; et dans le second, $m=1$.

Pour le calcul de la similarité syntaxique entre deux données de contexte, on utilise la formule suivante :

$$Sim_{sy}(d_i, d_j) = 1 - \frac{Dist_{sy}(d_i, d_j)}{Dist_{sy}^{max}} \quad (4.2)$$

, où $Dist_{sy}^{max}$ est la distance syntaxique maximale parmi les distances syntaxiques entre les données de contexte de \mathcal{D} . De cette façon, la valeur de distance est située dans l'intervalle $[0,1]$.

Cependant, il est clair que cette mesure ignore le rapprochement sémantique entre deux chaînes de caractères. Elle ne considère pas non plus les liens représentés par les prédicats d'une donnée de contexte. Dans les sous-sections suivantes, nous présentons des mesures de similarité qui contournent ces insuffisances.

Exemple de calcul

Les tableaux 4.II et 4.III montre un exemple de calcul de la distance syntaxique entre les données de la figure 4.4 : la somme des distances syntaxique entre les données (1-2),(1-3),(2-3) est respectivement (15,41,32) avec $Dist_{sy}^{max} = 41$. Les similarités respectives sont alors 0.63, 0, 0.21. Les deux données 1 et 2 sont les plus proches syntaxiquement.

3.1.2 Mesure de similarité conceptuelle

La similarité conceptuelle est calculée en termes de distance entre deux concepts selon leurs positions respectives dans l'ontologie.

Dans son mémoire HDR, F. L. Gandon (2008) [50] a présenté une étude intéressante sur la manière de voir les graphes comme des espaces métriques. Il avance qu'une ontologie offre un support à d'autres types de raisonnement que la dérivation logique. Par exemple, la hiérarchie de concepts contenue dans une ontologie peut être vue comme un espace permettant de définir des métriques pour comparer la similarité sémantique de deux notions.

Nous avons exploré cette approche dans notre travail pour définir une similarité conceptuelle entre les données du contexte. Nous nous sommes basés sur deux mesures largement utilisées en littérature (Jiwei Zhong *et al.* (2002) [73] et Rada *et al.* (1989) [111]), et nous avons proposé en conséquence deux mesures de similarité conceptuelle applicables sur les données de contexte. Dans ce qui suit, nous détaillons chacune de ces mesures et nous prenons les données de contexte de la figure 4.5 et la hiérarchie de la figure 4.6 comme exemple. La figure 4.5

Tableau 4.II – Les 5 premiers éléments des vecteurs des données de la figure 4.4

Donnée	Entity	ContextValue	EntityIdentifier	ContextValueIdentifier	Source
1	Person	Room	John	KitchenOven	IR02
2	Lamp	Room	I254	KitchenOven	Admin
3	Lamp	DeviceRunningStatus	I254	On	WT

Tableau 4.III – Calcul de la distance syntaxique

1-2	6	0	4	0	5
1-3	6	18	4	9	4
2-3	0	18	0	9	5

montre deux données de contexte envoyées par la même source IR02 et une autre donnée de contrôle envoyée par une autre source LS1.

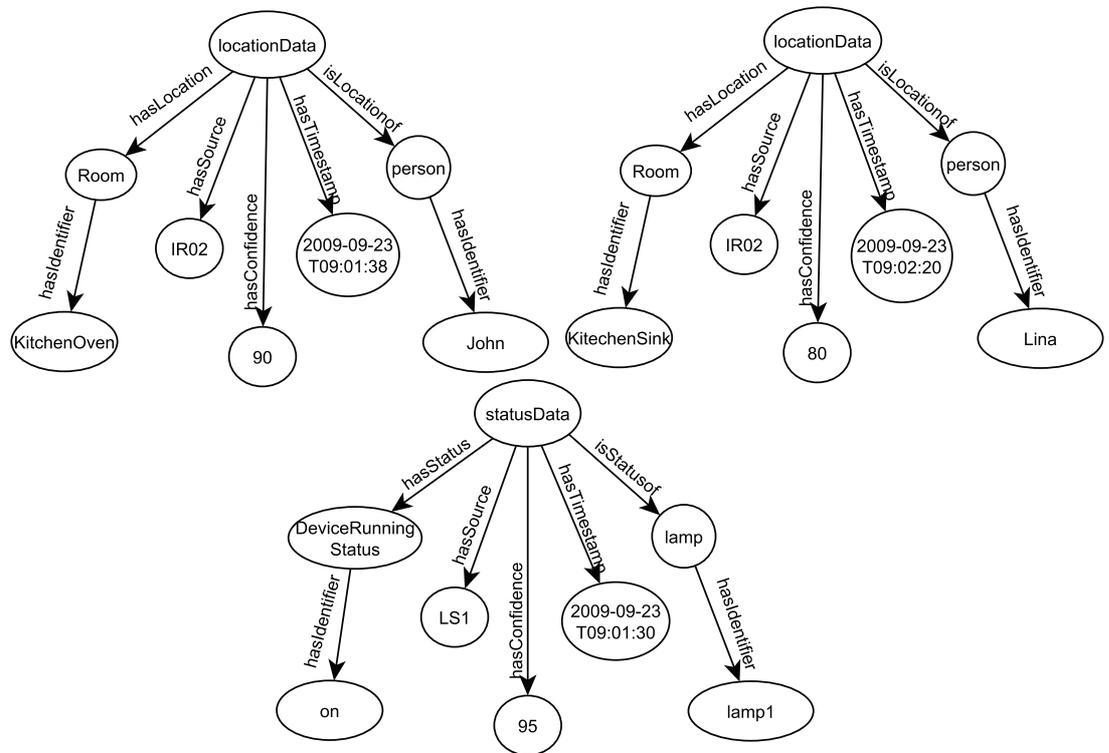


Figure 4.5 – Exemple de données de contexte

3.1.3 Mesure conceptuelle basée sur la mesure de Zhong

Nous donnons tout d'abord un bref aperçu sur la mesure de Zhong puis nous présentons notre mesure qui est une variante de celle de Zhong applicable aux données de contexte.

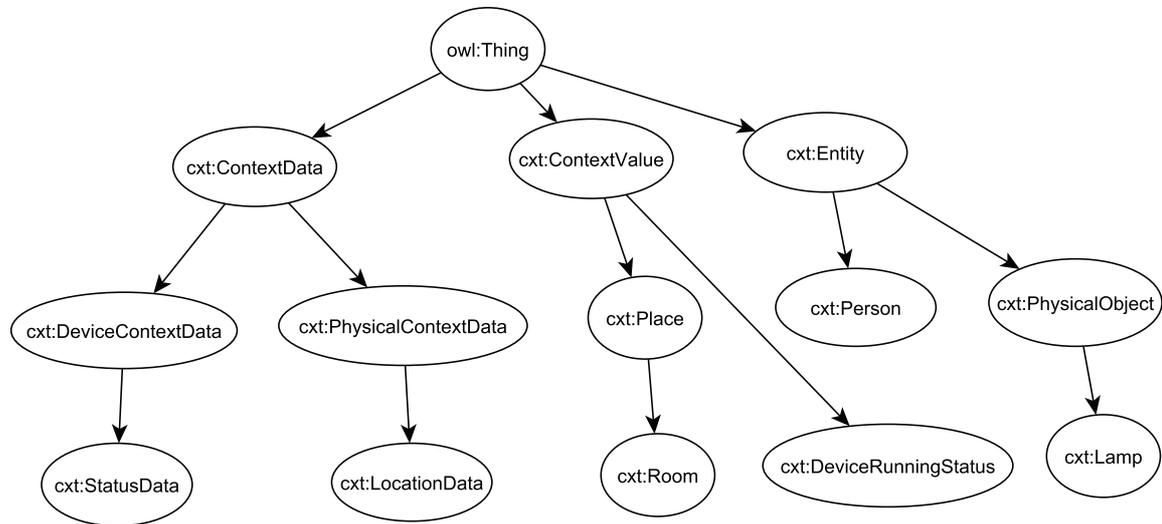


Figure 4.6 – Partie de la hiérarchie de l’ontologie

La mesure de J. Zhong *et al.* (2002) est définie dans [73]. Elle repose sur l’idée que les différences entre les concepts de niveau supérieur sont plus grandes que celles entre les concepts de niveau inférieur de la hiérarchie de l’ontologie. Le modèle considère également la distance entre les *frères* de la hiérarchie plus longue que celle entre les concepts *père* et *fil*s.

Soient cp_1, cp_2 deux concepts et c_{pp} leur parent le plus proche. Chaque concept dans l’ontologie a une valeur appelée *milestone* obtenue par la formule $mv = \frac{1/2}{k^{I(n)}}$, où k est un facteur pré-défini plus grand que 1. Il indique le taux auquel la valeur décroît au long de la hiérarchie. $I(n)$ est la profondeur du noeud n dans la hiérarchie tel que $I(\text{root}) = 0$ avec root la racine de l’ontologie. Dans notre étude la valeur de k est fixée à 2 pour donner plus de poids aux liens hiérarchiques entre les concepts. La distance entre deux concepts sera déterminée par leur milestone

nv et leur parent le plus proche comme suit :

$$Dist_{CC}(cp_1, cp_2) = Dist_{CP}(cp_1, ccp) + Dist_{CP}(cp_2, ccp) \quad (4.3)$$

avec,

$$Dist_{CP}(cp_k, ccp) = nv(ccp) - nv(cp_k), \quad (4.4)$$

cp_k indique un concept quelconque. Il désigne dans l'équation les deux concepts cp_1 et cp_2 . Ainsi, à partir de la distance, la similarité entre deux concepts est déduite de la manière suivante :

$$Sim(cp_1, cp_2) = 1 - Dist_{CC}(cp_1, cp_2) \quad (4.5)$$

Cette mesure convient bien pour comparer des données de contexte qui ne proviennent pas de la même source car pour des données provenant de la même source, la partie ($cxt :contextData$, $cxt :Entity$, $cxt :ContextValue$) est la même, seuls les objets attribués à ces sujets changent comme le montre l'exemple de la figure 4.5. De ce fait, il est évident que la mesure qu'on cherche doit avantager les distances entre frères de même *Entity* ou de même *ContextValue*. Par contre, cette mesure de Zhong favorise les liens père-fils, ce qui s'accorde mieux à comparer des données qui ne se réfèrent pas forcément aux mêmes entités ou aux mêmes valeurs de contexte.

Nous avons utilisé la mesure de Zhong pour définir une similarité conceptuelle entre les données de contexte. Pour appliquer cette mesure aux données de contexte, on compare les termes appartenant à chacune des données, un par un selon cette mesure pour les deux premiers car ils font partie des concepts de l'ontologie. Notre mesure de similarité conceptuelle sera comme suit :

$$Sim_C(d_i, d_j) = \frac{\sum_{k=1}^2 Sim(V_{d_i}(k), V_{d_j}(k))}{2} \quad (4.6)$$

La mesure de similarité conceptuelle que nous avons adoptée correspond à la

moyenne des similarités de Zhong sur les deux concepts (*entity* et *contextValue*) qui font partie de chaque donnée de contexte. Cette valeur est bien comprise dans l'intervalle $[0,1]$ vu la formule de calcul du *milestone*.

Exemple de calcul

Prenons deux données de contexte : *la localisation de John* et *le statut de la lampe*, de la figure 4.5. Nous allons utiliser une partie de la hiérarchie de l'ontologie \mathcal{O} représentée par la figure 4.6 pour montrer un exemple de calcul de la similarité conceptuelle de Zhong. Nous prenons donc les deux concepts correspondant aux éléments *entity* et *contextValue* de chaque donnée, en d'autres termes les deux premiers éléments des vecteurs de chaque donnée de contexte.

Selon cette hiérarchie, Entity et ContextValue sont les deux parents les plus proches respectivement des concepts (Person,Lamp) et (Room,DeviceRunningStatus) avec $l(Entity) = 1/4$ et $l(Contextvalue) = 1/4$. Les deux tableaux 4.IV, 4.V, 4.VI montrent le calcul de la similarité de Zhong. Elle est égale à la moyenne des deux similarités entre chaque paire de concepts des deux données de contexte $Sim_C = (11/16 + 11/16)/2 = 0.68$

3.1.4 Mesure conceptuelle basée sur la mesure de Wu et Palmer

Comme dans la section précédente, nous donnons tout d'abord un bref aperçu sur la mesure de Rada puis nous présentons notre mesure qui est une variante de celle de Rada applicable aux données de contexte.

Rada *et al.* (1989) [111] défendent l'hypothèse selon laquelle la similarité

Tableau 4.IV – Mesures conceptuelles pour locationData

locationData (John)	Person	Room
$l(n)$	2	3
nv	1/8	1/16
cp-ccp	1/8	3/16

Tableau 4.V – Mesures conceptuelles pour statusData

StatusData	Lamp	DeviceRunningStatus
$l(n)$	3	2
nv	1/16	1/8
cp-ccp	3/16	1/8

Tableau 4.VI – Calcul de la similarité conceptuelle de Zhong

locationData	Person	Room
StatusData	Lamp	DeviceRunningStatus
<i>DistCC</i>	5/16	5/16
Sim	11/16	11/16

entre deux concepts correspond au nombre minimum d'arêtes séparant deux concepts dans la hiérarchie de liens de type subsomption. Cette mesure se base donc sur le calcul de la distance entre les noeuds par le chemin le plus court. Elle présente un moyen rapide pour évaluer la similarité sémantique selon la hiérarchie de notre ontologie.

Soit cp_1 et cp_2 deux concepts, on a :

$$DistRada(cp_1, cp_2) = Min(NombreArcsEntre(cp_1, cp_2)) \quad (4.7)$$

Wu et Palmer (1994) [141] proposent une amélioration basée sur le ratio entre la profondeur du plus petit concept ancêtre commun et la profondeur des deux concepts comparés.

Soient cp_1 et cp_2 , c_{pp} des concepts, on a alors :

$$SimWuPalmer(cp_1, cp_2) = 2 * N_3 / (N_1 + N_2 + 2 * N_3) \quad (4.8)$$

avec c_{pp} le plus petit ancêtre commun de cp_1 et cp_2 et, N_1 est égale à la distance de Rada de cp_1 à c_{pp} , N_2 est égale à la distance de Rada de cp_2 à c_{pp} , et N_3 est égale à la distance de Rada de c_{pp} à la racine de la hiérarchie.

Cette mesure s'accorde bien au calcul de la similarité sémantique entre les données de contexte de même source car le chemin entre deux frères de la hiérarchie de l'ontologie est très court. De ce fait, la similarité sémantique entre les données de même source est plus importante que celle entre des données de sources différentes.

Soient V_{d_i} et V_{d_j} , deux vecteurs respectifs de deux données de contexte d_i, d_j .

$$Sim_C(V_{d_i}, V_{d_j}) = \frac{\sum_{k=1}^2 sim(V_{d_i}(k), V_{d_j}(k))}{2} \quad (4.9)$$

Cette mesure est fortement influencée par la densité du réseau sémantique de l'ontologie. Les liens de subsomption sont considérés comme uniformes c'est-à-dire que la longueur d'un lien est égale pour tous les liens de subsomption dans l'ontologie. Par exemple le lien entre chambre et cuisine est considéré de même longueur qu'entre meuble et table.

Exemple de calcul

Nous reprenons le même exemple présenté dans la section précédente pour calculer la mesure de Similarité de WuPalmer. Le tableau 4.VII résume ce calcul. La similarité conceptuelle entre les deux données de contexte *locationData* et *statusData* vaut alors $Sim_C = (2/5 + 2/5)/2 = 0.4$.

3.1.5 Mesure de proximité temporelle

La proximité temporelle entre deux instances de données de contexte est calculée sur les données de contexte non statique en utilisant leurs timestamps individuels. La proximité temporelle entre deux données de contexte est inspirée de la formule définie par [92, 140]. Soit t_1, t_2 deux timestamps de deux données de contexte respectives d_i et d_j , la proximité temporelle entre d_i et d_j est présentée

Tableau 4.VII – Calcul de la similarité conceptuelle de WuPalmer

locationData	Person (N1)	Room (N1)
StatusData	Lamp (N2)	DeviceRunningStatus (N2)
N1	1	2
N2	2	1
N3	1	1
SimWuPalmer	2/5	2/5

par la formule 4.10.

$$tp(d_i, d_j) = e^{-a \frac{|t_1 - t_2|}{T}} \quad (4.10)$$

où T est la durée d'une partition de la série temporelle des données de contexte et a est une valeur entre 0 et 1. Elle désigne le poids de la décroissance temporelle. Par exemple, considérant $a=86$ assure que la décroissance est insignifiante si la différence entre les données de contexte est moins de 7 secondes.

Exemple de calcul

Reprenons le même exemple que précédemment pour le calcul de la proximité temporelle entre les deux données de contexte (locationData et StatusData) qui ont respectivement les timestamps suivants : 2009-09-23T09 :01 :38 et 2009-09-23T09 :01 :30. $tp = e^{-86 \frac{8}{60}}$ sur une période $T = 60$ secondes.

3.2 Similarité entre contextes purs

Nous avons identifié trois mesures de similarité entre contextes : conceptuelle, relationnelle et statistique. Nous nous intéressons particulièrement à la mesure de similarité entre contextes purs car nous les utiliserons pour la découverte de situations fréquentes dans une des méthodes que nous proposerons au prochain chapitre. Nous détaillons ces mesures dans ce qui suit.

3.2.1 Mesure de similarité conceptuelle entre contextes purs

Soit C_i, C_j deux contextes purs $\in \mathcal{C}$ avec $N_{C_{pur}}$ le cardinal de ces ensembles. On désigne par d_k^i une donnée de contexte provenant de la source k et appartenant au contexte C_i . La mesure de similarité conceptuelle entre C_i et C_j est donnée par l'équation suivante :

$$Sim_{CC}(C_i, C_j) = \frac{\sum_{k=1}^{N_{C_{pur}}} Sim_C(d_k^i, d_k^j)}{N_{C_{pur}}} \quad (4.11)$$

Cette mesure est calculée sur la base de la similarité conceptuelle entre données de contexte ayant la même source.

3.2.2 Mesure de similarité statistique entre contextes purs

Notre mesure statistique est basée sur une mesure très répandue en littérature : TFIDF (ou term frequency-inverse document frequency) [79]. Elle présente de bon résultats quand il s'agit de regrouper des données [35]. Nous faisons d'abord un court rappel de cette mesure puis nous montrons comment elle sera appliquée.

TFIDF est une méthode de pondération largement utilisée dans la fouille de textes. Cette mesure statistique permet d'évaluer l'importance d'un terme par rapport à un document tiré d'un corpus. Le poids du terme augmente en fonction de son nombre d'occurrences dans le document. Il varie également en fonction de la fréquence du terme dans le corpus.

TF (*term frequency*) représente la fréquence d'un terme dans un document. Cette valeur est en général normalisée pour éviter les biais liés à la taille d'un document (le nombre d'occurrences serait potentiellement plus élevé dans un large document que dans un autre plus petit). Si on considère un document j et un terme i , alors la fréquence de ce terme se traduit par la formule suivante :

$$t f_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (4.12)$$

Où $n_{i,j}$ est le nombre d'occurrences du terme i dans le document j et k désigne tous les termes du document j .

IDF (*inverse document frequency*) représente la fréquence inverse du document j . C'est une mesure de l'importance du concept i dans un ensemble de documents. Elle consiste souvent à calculer le logarithme de l'inverse de la proportion de l'ensemble des documents qui contiennent le terme i .

$$idf_i = \log \frac{|Document|}{|\{document_j / cp_i \in document_j\}|} \quad (4.13)$$

- $|Document|$: nombre total des documents.
- $|\{document_j/cp_i \in document_j\}|$: nombre de documents où le terme cp_i apparaît.

Au final, la mesure est donnée par la formule $TFIDF = tf * idf$. Cette mesure est utilisée pour le calcul de la similarité entre contextes purs en considérant un contexte pur comme un document. Le choix de cette mesure est acceptable vu le grand nombre de contextes purs manipulés. Les termes d'un document correspondent aux objets (au sens RDF) de toutes les données appartenant à ce contexte et le corpus représente l'ensemble des contextes purs. La figure 4.7 résume cette analogie dans la partie haute de la figure et montre un exemple de celle-ci dans la partie basse en considérant deux contextes en qualité de documents. Le document de gauche reflète le contexte pur (sans la donnée statique de la figure 4.4). Le document de droite reflète le même contexte pur avec une modification de l'état de la lampe à off.

En accord avec cette analogie, nous associons à chaque contexte pur un vecteur de taille égale au nombre de termes distincts de tous les documents de contexte (la source, la confiance et le timestamp sont exclus). Nous appelons ce nombre Ta . Si on considère l'exemple de la figure 4.7, ce vecteur sera de taille $Ta = 9$ car nous avons *On* d'un côté et *Off* de l'autre. Pour chaque terme, nous calculons son tf et son idf . En considérant par exemple ces deux documents comme le corpus, le vecteur du document sur la gauche de la figure contiendra les valeurs $tf * idf$: $(1/9 * \log 1)$, $(0 * \log 2/1)$, $(1/9 * \log 2/1)$ pour les termes *Room*, *On*, *Off* respectivement.

Soient maintenant C_i, C_j deux contextes purs $\in \mathcal{C}$, et Vdc_i, Vdc_j les deux vecteurs de ces contextes purs. Pour calculer la mesure de similarité entre eux sur la base de la mesure TFIDF, nous utilisons alors une distance euclidienne mais d'autres distances mathématiques comme celle de Manhattan pourront être utilisées.

Corpus	Ensemble des contextes purs	
Document	Contexte pur	
/	Donnée de contexte	
/	Sujet	Triplet RDF
/	Prédicat	
Terme	Objet	

1- Analogie entre contexte et document

Room Lamp Person DeviceRunningStatus KitchenOven On John I254	Room Lamp Person DeviceRunningStatus KitchenOven Off John I254
--	---

2- Exemple de deux contextes en forme de documents

Figure 4.7 – TFIDF pour le calcul de la similarité entre contexte purs

$$Sim_{tfidf}(C_i, C_j) = 1 - \sqrt{\sum_{p=1}^{Ta} (Vdc_i(k) - Vdc_j(k))^2} \quad (4.14)$$

Pour le cas des données de contexte qui ont des objets (au sens RDF) de même valeur et provenant de triplets différents, le calcul de TFIDF ne donne pas le résultat espéré. Par exemple, le tableau 4.VIII montre deux documents. Ces documents correspondant à deux contextes purs contenant chacun deux données de contexte (lampe allumée et télévision éteinte), (lampe éteinte et télévision allumée). Nous avons les mêmes termes dans les deux documents, de ce fait les deux vecteurs correspondant sont exactement les mêmes. Ainsi, la similarité calculée est de 1 bien que les deux contextes purs soient différents. Ce problème est connu sous

le nom d'homonymie en recherche d'information. Nous n'avons pas rencontré ce problème lors des expérimentations car il est rare de rencontrer ces cas particuliers à cause de la diversité des sources des données de contexte utilisées. Même si ce problème se pose, il faudra considérer le TFIDF des triplets et non uniquement des objets. De plus, cette mesure sera combinée avec la mesure de similarité relationnelle qui elle tient compte des triplets de données. Nous la verrons dans la prochaine section.

3.2.3 Mesure de similarité relationnelle entre contextes purs

Nous reprenons ici l'idée de la similarité relationnelle pour définir une mesure relationnelle qui a pour objectif de comparer les relations existantes entre les concepts d'un graphe. Les mesures de similarité relationnelle ont été largement discutées en littérature comme en [73, 138]. La technique utilisée est de rechercher "le plus grand sous graphe commun" au sens du graphe RDF par rapport à la taille de l'union des deux graphes. Nous utilisons deux opérations : intersection et union. L'opération d'intersection entre deux graphes quantifie l'information partagée et l'opération d'union quantifie la taille du graphe résultant des deux graphes.

Soient C_i, C_j deux contextes purs de \mathcal{C} , la similarité relationnelle est calculée sur la base de la formule suivante :

$$Sim_R = \frac{NbTriplets(intersection(G_{C_i}, G_{C_j}))}{NbTriplets(union(G_{C_i}, G_{C_j}))} \quad (4.15)$$

Tableau 4.VIII – Exemple de problème posé par TFIDF

Lamp	Lamp
DeviceRunningStatus	DeviceRunningStatus
I254	I221
On	Off
TV	TV
DeviceRunningStatus	DeviceRunningStatus
TV1	TV1
Off	On

L'intersection retourne un graphe ne contenant que des triplets communs, l'union retourne un graphe fusion où les ressources de même URI sont fusionnées et *NbTriplets* est une fonction qui détermine le nombre de triplets dans un graphe. Ces deux graphes ne sont pas forcément fortement connexes. Certains triplets peuvent n'être reliés à aucun autre car un contexte est un ensemble de données sans que ces mêmes données soient attachées les unes aux autres.

Cette mesure peut cependant s'appliquer aux ensembles en général. Elle est une division de deux nombres. Sa valeur augmente rapidement quand le nombre des éléments de l'ensemble d'intersection augmente car de l'autre côté le nombre d'éléments de l'union diminue. L'avantage de son utilisation comme mesure de similarité est sa capacité à tenir compte de tout le triplet lors de l'intersection et l'union. De ce fait, les objets seront reliés via des prédicats à leurs sujets respectives. De cette façon le problème que nous avons rencontré avec la mesure TFIDF ne se posera pas.

3.3 Combinaison des mesures de similarité

Les mesures de similarité que nous avons présentées précédemment ne suffisent pas si elles sont utilisées individuellement. En effet, nous pouvons remarquer qu'elles sont complémentaires dans le sens où chacune d'elles tend à regrouper les données de contexte selon une de leurs caractéristiques. Nous proposons une formule générale qui permet de regrouper les données de contexte en combinant les différents types de mesure.

L'approche par combinaison de mesures de similarité a présenté beaucoup d'intérêt en littérature comme par exemple en [10] et dans la thèse de A. Alasoud (2009) [4]. La combinaison de similarités présentée dans ce chapitre, que ce soit pour le regroupement de données de contexte *Simg* ou pour le regroupement de contextes purs *Simgc*, est une somme pondérée puis normalisée des différentes similarités utilisées comme suit :

$$Sim_g = \frac{\alpha * sim_{sy} + \beta * sim_C + \gamma * tp}{\alpha + \beta + \gamma} \quad (4.16)$$

$$Sim_{gc} = \frac{\alpha' * sim_{CC} + \beta' * sim_{t\ fid\ f} + \gamma' * Sim_R}{\alpha' + \beta' + \gamma'} \quad (4.17)$$

Les poids α , β , γ et α' , β' , γ' sont fixés dans les expérimentations qui seront détaillées au chapitre 7. Ces poids sont égaux à 1 si on ne veut pas avantager une mesure par rapport à une autre.

4 Conclusion

Nous avons présenté dans ce chapitre le modèle de données de contexte que nous utilisons. Ce modèle est utilisé pour découvrir le contexte sur la base du regroupement de données ou des contextes eux-mêmes. Dans les approches existantes, le contexte est utilisé comme modèle conceptuel dans les applications.

Nous avons aussi proposé une ontologie de l'espace intelligent pour la prise en compte de l'hétérogénéité sémantique des données de contexte provenant de sources diverses et variées. La nouveauté dans la construction de cette ontologie est qu'elle englobe dans sa structure la relation entre les services et les concepts utilisés par ces données. De cette manière les services peuvent être détectés directement à partir des données de contexte. Quant aux méthodes de mesure syntaxique, temporelle, conceptuelle, statistique et relationnelle, elles sont propres à notre modélisation et présentent une première ébauche de regroupement de données et de contexte. Nous reviendrons dans la partie *Application et validation* sur les résultats apportés par ces mesures.

La façon dont les éléments ci-dessus peuvent être intégrés constitue le fondement pour atteindre notre objectif d'une approche par les données de contexte pour l'adaptation automatique des services. L'étendue de cette approche et les solutions apportées sont abordées dans le chapitre suivant.

CHAPITRE 5

DÉCOUVERTE DU CONTEXTE

Dans ce chapitre, nous proposons et détaillons deux approches de découverte du contexte. On les appellera approche ascendante et approche descendante. La première approche s'intéresse à la découverte de situations sans connaissance a priori de la situation à trouver. Elle a donné naissance à deux alternatives du traitement du problème :

- soit une découverte guidée par un type particulier de situation (cf. section 3.1 du chapitre 3) à découvrir,
- soit une découverte aveugle sur les données de contexte.

L'approche descendante est fondée sur l'idée d'utiliser une ontologie pour décrire les situations à découvrir et de corréliser ces descriptions avec des groupements de données de contexte. Ce chapitre commence par présenter dans la section 1 les fondements de la fouille de données. La section 2 décrit notre démarche de découverte du contexte. Dans les sections 3 et 4, nous détaillons les approches ascendante et descendante respectivement. Nous terminons ce chapitre par une conclusion à la section 5.

1 Fondements de la fouille de données

La fouille de données (data mining en Anglais) fait référence aux techniques mathématiques, statistiques et d'intelligence artificielle qui sont utilisées pour la découverte de connaissances lors d'un processus d'exploration d'une grande quantité de données. Le clustering est l'une des méthodes les plus utilisées pour effectuer cette découverte. Elle emploie des algorithmes d'apprentissage non supervisé en utilisant des mesures de similarité pour regrouper des données proches (au sens de ces similarités) en clusters.

Du point de vue de la littérature, Laurent Candillier, dans sa thèse "contex-

tualisation, visualisation et évaluation en apprentissage non supervisé", en 2006 [26], a identifié plusieurs approches de clustering, notamment les approches statistique, stochastique, et à base de graphes. Il ressort de cette étude qu'il est difficile d'établir une comparaison des différents algorithmes proposés par ces approches à cause de deux raisons : *la dépendance du clustering à la manière dont il est utilisé*, par exemple chercher un nombre défini de clusters ou avoir une bonne séparation entre les clusters ; de plus *l'abondance des mesures de similarité existantes* rend difficile le choix des mesures à utiliser. De ce fait, notre but n'est pas de détailler ces aspects car la comparaison entre algorithmes de clustering relève d'un champ de recherche très documenté en littérature [78, 90].

Dans ce chapitre, notre objectif est de mettre en place des techniques de découverte du contexte sur la base d'algorithmes de clustering existants et des mesures de similarité combinées (Simg : entre données de contexte, et Simgc : entre contextes purs) que nous avons présentées à la fin du chapitre 4. La raison principale du choix du clustering est sa capacité à corréliser les données sans nécessiter une phase d'entraînement. Cette faculté est considérée comme essentielle pour rendre notre système robuste à la dynamique de l'environnement. Nous nous limitons cependant dans ce chapitre à exploiter les algorithmes de clustering les plus connus et adaptés aux structures des données en entrées des algorithmes. Nous évaluons ces approches lors des expérimentations qui seront détaillées au chapitre 7 afin de déterminer celle qui sera retenue pour l'implémentation d'un démonstrateur.

2 Différentes approches de découverte de contextes

Dans ces travaux de thèse, *la découverte du contexte pour l'adaptation se présente comme un processus d'extraction automatique de situations à partir d'un large volume de données de contexte* mais avant d'entamer la présentation de ces approches, nous souhaitons rappeler quelques précisions importantes pour la suite de ce chapitre.

Nous avons identifié plusieurs types de situations dans l'état de l'art (cf. figure 3.2), deux selon le niveau de description du contexte "compréhension" : (situations dépendantes et épisodes), et trois autres selon le niveau de description du contexte "situation" : (situations fréquentes, rares, tendances). Nous avons également montré au chapitre 3 que la découverte est une technique très prometteuse pour extraire ces types de situations par rapport à d'autres techniques telles que le raisonnement. Cependant, dans nos réalisations, nous nous sommes limité au niveau de description du contexte "situation" en précisant que le *contexte est la description d'une situation qu'on peut déterminer automatiquement à partir de données dites de contexte*. Nous nous sommes focalisés sur les situations fréquentes et rares (inhabituelles).

Ainsi, nous nous sommes intéressés à la découverte des situations fréquentes et des situations rares pour effectuer une adaptation automatique de services. En effet, ces deux types de situations représentent un enjeu pour l'adaptation de services :

- les situations fréquentes permettront à l'adaptation de tenir compte des habitudes d'utilisation des services par les usagers (nous verrons au chapitre 6 comment identifier des dérives et activer des services en conséquence)

- les situations rares sont inhabituelles (cf. chapitre 3 section 3.1.2). Nous nous intéressons à ces situations car elles sont le contraire des situations fréquentes et peuvent nous servir à déterminer une dérive lors d'une adaptation automatique de service.

Nous avons identifié deux approches que nous explorons dans le cadre de cette thèse pour la découverte de ces deux types de situations : approche ascendante et approche descendante. La figure 5.1 résume ces deux approches.

L'approche ascendante de découverte du contexte consiste à faire une découverte automatique des situations fréquentes ou rares à partir de données de contextes sans connaissance a priori des descriptions de ces situations. Cette approche de découverte a donné naissance à deux alternatives de traitement du pro-

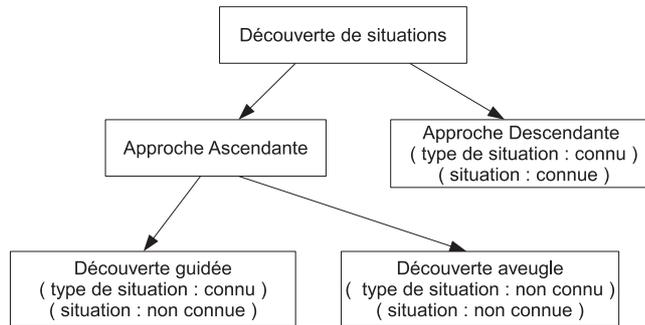


Figure 5.1 – Approches de découverte du contexte

blème. L'une fait une extraction de connaissance *guidée* par le *type de situation* à trouver (par exemple : situations fréquentes, rares). L'autre procède par une *découverte sans objectif préalable* c'est à dire une fouille aveugle dans les données de contexte.

L'approche descendante se base sur l'idée de décrire une classe de situations qui nous intéresse dans une ontologie et de faire une fouille dans les données collectées pour savoir si cette classe de situations a été produite dans l'environnement. La différence par rapport à l'approche ascendante est le fait que dans cette approche, nous définissons a priori la classe de situations et nous voulons vérifier si elle est présente dans les données.

Ces deux approches supposent l'existence de données de contexte collectées dans des bases de données. La période sur laquelle s'effectue la découverte varie d'une période de quelques heures à la période d'une journée ou plusieurs semaines sachant que la longueur de la période influera sur le résultat de la découverte. Dans ce qui suit, nous détaillons ces deux approches. Nous soulignons que nous ne ferons pas la présentation des résultats des expériences liées à ces approches dans ce chapitre pour simplifier sa lecture : nous prenons le soin de présenter les détails de ces expérimentations dans un chapitre de la partie *Application et Validation*.

3 Approche ascendante de découverte du contexte

Nous avons utilisé dans cette approche des algorithmes de clustering existants qui s'accordent avec les deux alternatives proposées par cette approche. Nous présentons chacune d'elles dans les sous-sections suivantes.

3.1 Découverte guidée de situations

Dans la découverte guidée, l'algorithme de clustering est utilisé pour découvrir les situations fréquentes. Une situation fréquente se définit comme un contexte qui apparaît plusieurs fois dans les données de contexte. Sa fréquence d'apparition dans les données est un paramètre important pour sélectionner les situations les plus fréquentes. Ces derniers ont plus de chance à se produire et donc plus pertinent à l'adaptation automatique de service. Cependant une situation qui n'a qu'une fréquence de 1 est considérée comme rare.

Les situations fréquentes ont la particularité de se répéter dans les données de contexte. Elles sont donc un bon moyen pour tester nos mesures de similarité (cf. chapitre 4) et des algorithmes de clustering. De point de vue application, ces situations sont intéressantes pour déterminer les habitudes de l'utilisateur et de son environnement. Par exemple : la télé est toujours allumée pendant le repas, les volets des fenêtres sont toujours fermés pendant l'absence de l'utilisateur.

Nous proposons deux manières pour accomplir cette tâche : un regroupement de contextes et un regroupement de données de contexte. Nous les présentons respectivement ci-dessous.

3.1.1 Regroupement de contextes pour la découverte de situations fréquentes

Cette méthode repose sur l'idée d'une segmentation des données en une suite de contextes purs. Cette segmentation est effectuée selon un critère dit de *cohérence* en respectant l'ordre temporel des données. Le critère de cohérence assure qu'il n'y a pas de valeurs de contexte contradictoires au sein d'un même contexte

pur. Par exemple, un capteur qui dit que la lampe est allumée et un autre qui dit que cette même lampe est éteinte. Un changement désigne par exemple, une lampe qui devient allumée et indique une transition vers un autre contexte pur. Pour rendre les choses plus claires, nous avons illustré par la figure 5.2 un environnement en forme de carré et des capteurs en forme de cercles à deux états (plein, vide). Les sous-figures de 2 à 6 représentent des changements produits dans cet environnement. Au total, nous avons 6 contextes purs dont un contexte fréquent (3,6). Formellement, on définit la règle de cohérence comme suit :

Définition 3.1.1 (cohérence). Soit $C_{C_{pur}}$ un contexte pur, $C_{C_{pur}}$ est cohérent ssi $\forall d_i, d_j \in C_{C_{pur}}$, si $IdentifierEntity(d_i) = IdentifierEntity(d_j)$ alors $IdentifierContextValue(d_i) = IdentifierContextValue(d_j)$.

Une fois que nous disposons d'un ensemble de contextes purs, nous calculons les similarités entre les contextes purs en utilisant la mesure de similarité combinée *Simgc* (conceptuelle, relationnelle et statistique) définie à la fin du chapitre 4. Nous construisons ensuite un large graphe où les noeuds sont les clusters purs et les arêtes sont étiquetées avec la similarité entre chaque paire de contextes purs. Néanmoins, lorsque la similarité = 0, l'arête n'est pas retenue ; le graphe n'est pas complet. A ce stade nous pouvons appliquer un algorithme de clustering sur ce graphe.

Notre technique de découverte de situations fréquentes utilise une méthode existante de clustering appelée MCL (Markov cluster algorithm) [133]. Il a été proposé par Van Dongen en l'année 2000. C'est un algorithme de clustering non supervisé de séparation de graphes, rapide et capable de prendre en entrée de très grands graphes. Notre entrée pour cet algorithme est un fichier où chaque ligne est de la forme (identifiant du cluster pur 1, identifiant du cluster pur 2, mesure de similarité). Par exemple, (C_1 C_2 0.25). Le résultat est en forme de ligne où chaque ligne représente une liste de contexte purs représentant un cluster. Ainsi, chaque cluster comportant plus d'un contexte pur indique un contexte fréquent. Un de ces contextes purs représentera la situation fréquente. Tous les contextes purs d'un

même cluster étant considérés similaires par l'algorithme, l'un quelconque de ces contextes représentera la situation.

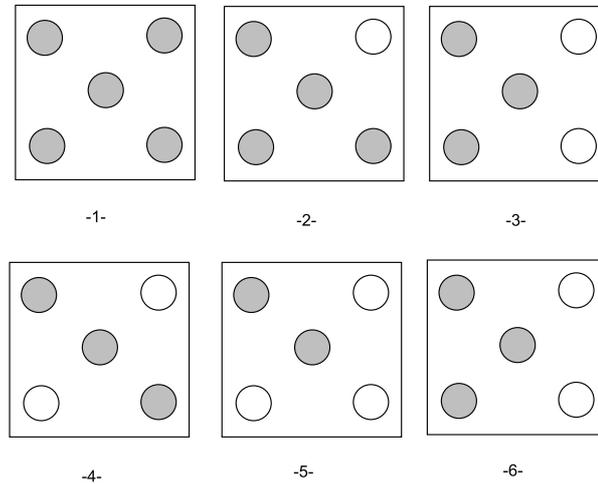


Figure 5.2 – Exemple abstrait de contextes purs et de contextes fréquents

Le principe de fonctionnement de l'algorithme MCL postule qu'il est naturel que les clusters dans un graphe soient caractérisés par la présence d'un grand nombre d'arêtes entre les membres de ce cluster. Ainsi, l'objectif est de trouver des groupes de noeuds fortement connectés. L'algorithme MCL répond à cette question en simulant des parcours aléatoires à l'intérieur du graphe à l'aide de deux opérations appelées expansion et inflation.

L'expansion associe des probabilités à tous les parcours allant d'un noeud i vers un noeud j . Puisque, les chemins au sein d'un cluster sont longs (mesurés en nombre d'arcs composant ces chemins) à cause de la connexité forte entre ses noeuds et les chemins entre des clusters différents sont généralement courts (en nombre d'arcs), les probabilités associées aux paires (i,j) de noeuds appartenant à un même cluster vont être en général relativement grandes car il y a beaucoup de chemins allant de i à j . L'expansion va donc avoir pour effet d'augmenter les probabilités intra-cluster et va diminuer celles inter-cluster. Quant à l'inflation, elle vise à changer les probabilités associées en favorisant les parcours les plus probables par rapport aux parcours moins probables. L'inflation est sensible à un

paramètre $r > 1$. Si cette valeur augmente, l'inflation devient plus forte et augmente en conséquence la finesse des clusters.

Les deux opérations expansion et inflation sont répétées par l'algorithme jusqu'à ce que le graphe soit séparé en plusieurs clusters. En pratique, l'algorithme est très rapide et supporte bien les fortes charges de plusieurs centaines de milliers de noeuds. La valeur de r testée sur un volume réduit de données de contexte n'influe pas sur le résultat. Cependant la variation de cette valeur sur un large volume de données de contexte influe sur la formation des clusters en faisant varier leur nombre.

Cette technique de clustering donne de bons résultats que nous avons validés via l'analyse manuelle et visuelle de la formation des clusters.

3.1.2 Regroupement de données de contexte pour la découverte de situations fréquentes

Nous explorons dans cette section un algorithme de clustering de type hiérarchique dit **HAC** (Hierarchical Agglomerative Clustering) [25] pour la découverte de situations fréquentes. L'idée de cet algorithme est de considérer au départ chaque donnée comme un cluster à un seul élément puis de regrouper ces données dans des clusters plus importants en construisant une hiérarchie ascendante jusqu'à avoir un seul cluster contenant toutes les données de base.

En utilisant HAC sur des données de contexte, nous nous sommes rendu compte que HAC regroupe les données sans se soucier de leur ordre temporel. On a eu alors comme résultat des groupes de données qui se produisent à des temps variés et qui ne représentent pas des situations mais des groupes de données qui proviennent tous de la même source par exemple. La raison de ce résultat provient de la mesure de similarité combinée *Simg*, où nous avons donné le même poids aux mesures syntaxique, conceptuelle et temporelle. Nous avons eu alors l'idée d'avantager la mesure temporelle. Après plusieurs expérimentations sur un échantillon de données de contexte que nous détaillons leur cadre expérimental

au chapitre 7, nous avons remarqué que les cas où on a de bons résultats correspondent à des valeurs de poids vérifiant :

$$Simg = \frac{\alpha * sim_{sy} + \beta * sim_C + \gamma * tp}{\alpha + \beta + \gamma}, \gamma > \alpha + \beta \quad (5.1)$$

Cela signifie qu'une donnée de contexte est regroupée avec les données qui sont produite au même temps, juste avant ou après sinon elle sera considérée comme un singleton qui sera regroupé avec d'autres clusters proches (au sens temporel) en montant dans la hiérarchie.

Ainsi, HAC utilisé sur des données de contexte arrive à construire des situations de manière hiérarchique sous forme d'inclusions organisées en dendrogramme comme le montre figure 5.3. L'axe des abscises contient les clusters singletons et l'axe des ordonnées désigne la distance inter-clusters. Une coupure en horizontal de l'arbre détermine le nombre de clusters à considérer. Supposons que les lettres A à F de cette figure correspondent à des données de contexte. Les contextes se construisent dans l'ordre (Y, Z), (B, C), (B, C, D), (B, C, D, Y, Z) et (A, B, C, D, Y, Z). Cette caractéristique est équivalente à former des niveaux d'abstraction sur les situations. Elle signifie aussi qu'un lien hiérarchique entre deux situations éloignées (B, C, D) et (Y, Z) dans le temps est équivalent à une répétition d'une même situation, puisque les situations qui se ressemblent seront reliées dans la hiérarchie. En résumé, en *préservant l'ordre temporel*, l'algorithme hiérarchique fournit à la fois la relation entre des situations éloignées dans le temps et offre une possibilité d'explorer la construction d'épisodes en se basant sur les abstractions successives de situations sur une période donnée. Ceci dit, il reste une autre difficulté à résoudre pour cet algorithme qui est celle du choix de la meilleure coupure de la hiérarchie.

Il n'est pas approprié pour nous de spécifier a priori le nombre de clusters au départ car nous voulons que cette valeur se détermine de manière automatique selon le volume et la période des données collectées. Dans [103] et dans ¹, plusieurs

¹<http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/BranchCutting/>

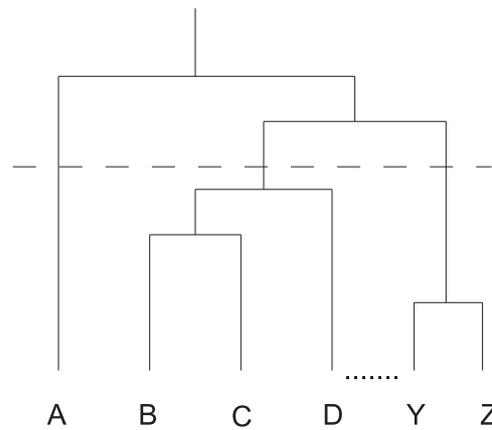


Figure 5.3 – Exemple de clustering HAC

méthodes pour déterminer le niveau de coupure sont présentées. L'une d'elles propose de déterminer le niveau de coupure en cherchant l'endroit, où la distance inter-cluster sur l'axe des ordonnées change brusquement.

Un des avantages de cet algorithme est qu'il est capable de regrouper de nouvelles données sans reprendre tout le traitement car ces données arrivent triées (temporellement) et s'intègrent dans le dendrogramme en intégrant des clusters existants ou en formant de nouveaux clusters. Nous allons à présent décrire le déroulement général de l'algorithme HAC.

Étant donné un ensemble de N données et une matrice de distances (ou similarités) M de taille $N * N$, trois options majeures sont données en littérature pour la formation des clusters : **single-linkage**, **complete-linkage** et **average-linkage**.

En "single-linkage", la distance entre deux clusters est calculée comme la distance entre les deux données les plus proches dans les deux groupes. On la note :

$$D(C_i, C_j) = \min_{x \in C_i, y \in C_j} (x, y) \quad (5.2)$$

En "complete-linkage", la distance entre deux clusters est calculée comme la distance entre les deux éléments les plus éloignés dans les deux groupes. On la

note :

$$D(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \quad (5.3)$$

En "average-linkage", la distance entre deux clusters est calculée comme la moyenne de toutes les distances entre les paires d'éléments x de C_i et y de C_j . On la note :

$$D(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y) \quad (5.4)$$

Nous n'avons pas remarqué une nette différence dans les résultats de clustering quant à l'utilisation de ces options sur quelques échantillons de données de contexte. De ce fait, nous avons retenu l'option single-linkage dans l'expérimentation (cf. chapitre 7). Cependant, l'approche de regroupement de contextes (vue précédemment et qui utilise des contextes purs) donne un meilleur résultat pour la découverte des situations fréquentes : pour un regroupement de données de contexte, l'algorithme HAC produit du bruit dans la situation fréquente. Ce bruit représente les données de contexte étrangères à la situation. Par exemple, si on suppose que (B,C) et (Y,Z) sont deux situations qui se ressemblent alors la donnée (D) est considérée comme du bruit pour les deux situations (B, C, D) et (Y, Z).

Il reste à souligner que l'algorithme HAC présente une complexité de $O(n^2)$, où n est le nombre de données en entrée à l'algorithme.

3.2 Découverte aveugle de situations

A la différence des travaux menés dans les précédentes sous-sections, la découverte aveugle de situations ne met pas de postulat sur le type de situations à découvrir (par exemple fréquentes, rares, etc.). Ainsi, cette section a pour but d'explorer le comportement de certains algorithmes de clustering sur le regroupement des données de contexte en utilisant la mesure de similarité combinée *Simg* : il s'agit d'étudier la construction de regroupements de données et de voir si ces re-

groupements peuvent correspondre à des situations pertinentes pour l'adaptation des services.

Dans les sous-sections suivantes, nous citons les méthodes que nous avons explorées pour la découverte de situations de manière aveugle sur les données en nous basant uniquement sur les mesures de similarité entre données. En effet, la difficulté de cette approche réside dans le sens qui sera donné aux clusters trouvés par cette approche. Vu que la découverte est aveugle, une interprétation manuelle des clusters s'avère nécessaire. Ainsi, nous nous pouvons pas envisager une adaptation automatique de services basée sur ces clusters. Malgré cette difficulté, le fait d'explorer cette approche a permis de mieux comprendre les données et la façon dont des groupes réagissent aux mesures de similarité. Nous présentons dans ce qui suit deux façons de procéder à une découverte aveugle sur les données de contexte.

3.2.1 Fouille d'un large graphe de données de contexte

Notre première idée pour l'exploration d'une fouille aveugle sur les données est de voir les données de contexte comme un large graphe, où les noeuds représentent les données de contexte et les arêtes représentent les similarités entre ces données. On ne considère que les similarités supérieures à un seuil λ . La formule que nous avons utilisée est la suivante :

$$Lien(d_i, d_j) = \begin{cases} 1 & \text{si } Simg(d_i, d_j) \geq \lambda \\ 0 & \text{sinon} \end{cases} \quad (5.5)$$

Après la construction du graphe de données, l'étape suivante est d'utiliser un algorithme de clustering pour partitionner le graphe en plusieurs clusters de façon à ce que la dépendance soit forte au sein d'un même cluster et faible avec les autres clusters. Pour cela, nous avons exploré un algorithme rapide appelé Bunch [116] ou "Module Dependency Graph". Cet algorithme a été initialement développé pour aider les développeurs à maintenir, comprendre et vérifier les sources

de leurs programmes. Cet algorithme est rapide et très utilisé mais pour nous le choix de l'algorithme n'est pas très important en soi car notre objectif est de vérifier s'il est possible de voir les données de contexte comme un large graphe, et si les clusters qui seront trouvés peuvent correspondre à des situations pertinentes pour nos travaux. Cette approche est sensible au seuil λ . Un petit changement de cette valeur conduit à un changement total dans la configuration du graphe et donc sur le résultat du clustering. De plus, les clusters trouvés sont totalement indépendants et donc nous n'avons pas d'informations sur la ressemblance entre clusters comme c'est le cas des liens hiérarchiques produits par l'algorithme HAC de la section précédente.

3.2.2 Fouille numérique des données de contexte

Nous nous intéressons dans cette section à une idée qui vise à transformer les données de contexte en vecteurs de données numériques et à faire du clustering sur les données numériques obtenues. L'avantage de cette transformation est d'un côté de pouvoir explorer plusieurs algorithmes de clustering disponibles et de l'autre côté de vérifier l'impact que peut avoir la non prise en compte de la représentation sémantique des données.

Nous avons proposé une transformation simple des données de contexte en vecteurs de données numériques. Elle consiste dans un premier temps à organiser les données de contexte en lignes et en colonnes dans un tableau td . Chaque ligne correspond à un vecteur de 6 éléments d'une donnée de contexte (entity, ContextValue, IdentifierEntity, IdentifierContextValue, Source, Confidence) sans prise en compte du temps, où les éléments des vecteurs des données de contexte correspondent aux 6 colonnes. Le tableau tn contenant les valeurs numériques correspondant aux vecteurs de données de contexte sera construit comme suit : si $td(i, j)$ est une valeur numérique alors $tn(i, j) = td(i, j)$ sinon nous utilisons la formule suivante :

$$tn(i, j) = \frac{n(td(i, j))}{N}, j = 1..6 \quad (5.6)$$

où $n(td(i, j))$ est le nombre d'apparitions de l'élément $td(i, j)$ dans la colonne j et N est le nombre total des données de contexte. Par exemple, si l'entité *person* d'une donnée d_i apparaît 70 fois dans 400 données de contexte, on aura pour $td(i, 1) = \text{"person"}$, $tn(i, 1) = 70/400$.

A l'issue de cette transformation plusieurs algorithmes de clustering sur des données numériques peuvent être exploités. Nous avons exploré l'algorithme EM (Expectation-Minimization) proposé par A. P. Dempster *et al.* (1977) en [102]. Cet algorithme est reconnu pour ses bons résultats. L'idée de *EM* est de trouver le maximum de vraisemblance des paramètres d'un modèle probabiliste. Une collection d'exemples de départ est constituée de vecteurs à n dimensions. L'algorithme fait l'hypothèse que ces vecteurs sont des tirages aléatoires d'un mélange de K distributions gaussiennes. L'objectif est alors d'estimer les paramètres de chacune des K distributions.

Ignorer la sémantique des données de contexte et du temps a un fort impact. En effet, l'algorithme tend à faire des regroupements de données de même source comme si les autres triplets des données de contexte n'avaient aucun apport. Après analyse, nous avons trouvé que seuls deux valeurs du vecteur changent : correspondant à IdentifierEntity et IdentifierContextValue. Les autres sont quasiment les mêmes puisque nous avons utilisé dans cette approche des sources de contexte qui envoient des données se référant à la même entité et à la même valeur de contexte. Par exemple, la porte du salon est ouverte, la porte du salon est fermée.

4 Approche descendante

Dans cette approche, la situation à découvrir est définie au préalable. Le but de faire une découverte de cette même situation dans les données de contexte est de déterminer si cette situation a été produite dans l'environnement. Par exemple, une

personne rentrant du travail en fin de journée peut souhaiter savoir si son conjoint malade d'Alzheimer a pris ses médicaments.

Les situations que nous avons voulu utiliser dans nos expérimentations ressemblent plus à des activités qu'à une suite de changements dans l'état de l'environnement comme nous l'avons vu lors de la découverte de situations fréquentes plus haut dans ce chapitre. Les activités sont des situations plus difficiles à caractériser car une activité suppose une suite de changements comme par exemple "la prise des médicaments nécessite de les prendre à la main de la table puis de les avaler". De ce fait, nous avons introduit la notion de relativité dans le temps et dans la localisation en ajoutant d'autres concepts et d'autres relations (cf. 5.4) dans l'ontologie de contexte pour qu'on puisse définir ce type de situations. Nous avons intégré une ontologie que nous avons développée conjointement avec le laboratoire DOMUS² [31]. Ci-dessous, nous décrivons ces ajouts et nous montrons un exemple de situation.

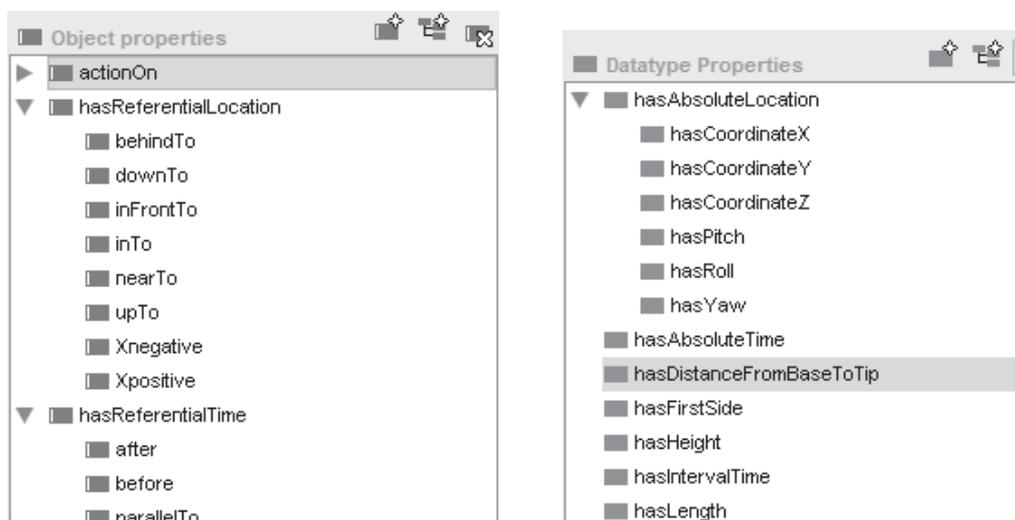


Figure 5.4 – Integration de la relativité dans la localisation et dans le temps

²<http://domus.usherbrooke.ca>

4.1 Modélisation des situations à l'aide d'ontologies

Nous avons introduit la notion de relativité dans l'ontologie. Elle signifie pour ces travaux que la situation qu'on veut définir n'est pas une image statique de l'environnement mais une situation où des actions pourront être faites et changer l'état de l'environnement.

La figure 5.5 montre la structure générale de l'ontologie et ses deux caractéristiques : le temps relatif et la localisation relative qui sont détaillés dans les deux prochaines sous-sections.

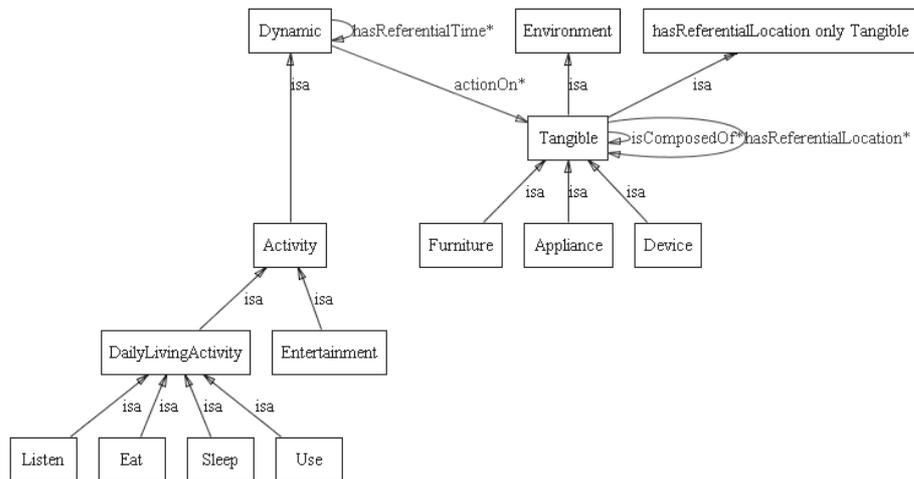


Figure 5.5 – Vue partiel de l'ontologie

4.1.1 Localisation relative

La localisation absolue des objets dans l'espace intelligent engendre l'imprécision, l'insuffisance et l'ambiguïté ; par exemple, dans une maison, le réfrigérateur se trouve dans la cuisine et le four est situé dans la cuisine. Il n'y a pas une idée claire de la disposition relative des deux. Contrairement à la localisation absolue, la localisation relative prend en compte la localisation de l'objet par rapport aux

autres objets dans l'environnement. Cette localisation prend la forme de *upTo*, *DownTo*, *behindTo*, *inFrontTo*, *nearTo*, *Xpositive*, *Xnegative*. *Xpositive* et *Xnegative* désignent respectivement l'orientation droite et gauche, selon l'angle sous lequel l'objet est perçu. Dans la figure 5.6, nous donnons l'exemple "le lit est en face de la table, la chaise est à côté de la table, le lit est derrière la chaise"

En plus de la localisation relative, on utilise la localisation absolue. Elle est déterminée par les coordonnées (X, Y, Z) d'un objet dans l'environnement utilisateur. La localisation relative est intégrée à l'ontologie comme une propriété d'objet ("Object properties") selon la description du langage OWL puisqu'elle met en relations deux entités. La localisation absolue concerne une entité, donc elle est introduite comme une propriété de type de données ("data type properties"). La figure 5.6 montre un exemple de modélisation utilisant la propriété *nearTo*.

4.1.2 Temps relatif

La notion de temps relatif est inspirée du travail de James F. Allen [5, 6] sur sa théorie de la relation temporelle entre les actions. James F. Allen a défini un ensemble de relations de temps qu'on a repris et intégré à notre ontologie afin de décrire les relations temporelles entre les actions d'une même situation.

De ce fait, on arrive à décrire le processus d'accomplissement d'une activité en utilisant des propriétés tels que *before*, *after*, *parallelTo*, *since*, *from*, *Overlaps*. De plus, chaque activité a une durée et un temps absolu qui se réfère au temps de début de l'activité. Les différentes propriétés sont visibles sur la figure 5.4. La figure 5.7 montre un exemple de modélisation utilisant la propriété *before*. Cet exemple illustre une personne qui écoute la radio avant de dormir.

4.2 Algorithme de l'approche descendante

L'idée de cet algorithme repose sur une comparaison entre des regroupements de données de contexte et la description de la situation à découvrir. Cette approche s'avère intéressante car les données de contexte et la description de la situation

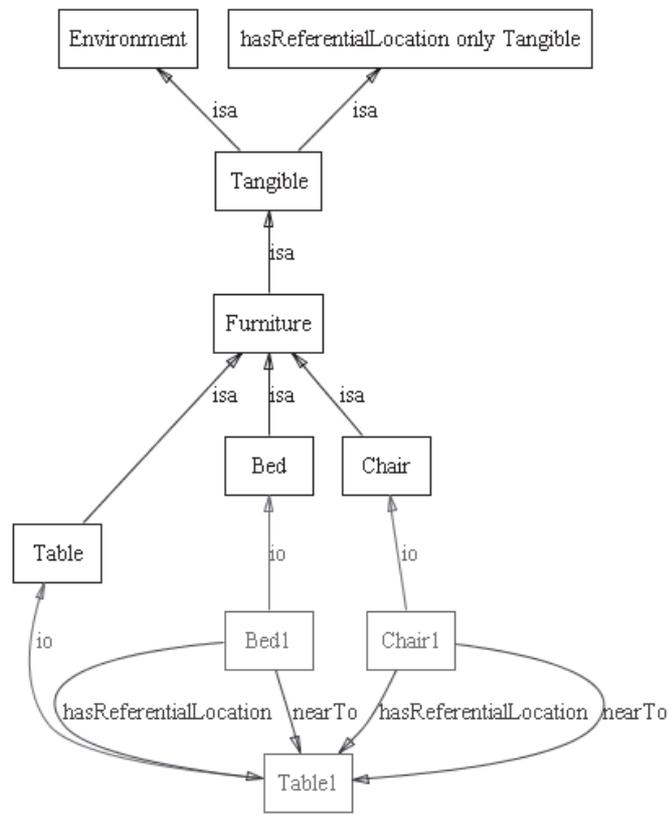


Figure 5.6 – Exemple de localisation relative

sont exprimées toutes les deux de manière sémantique, ce qui permet d'envisager un calcul de distance entre les deux.

La première tâche de notre algorithme est d'identifier l'événement qui peut caractériser la situation dans les données de contexte puis de construire autour de cet événement le groupement qui correspond le mieux à la description de la situation décrite dans cette nouvelle ontologie. Pour cela, à chaque situation décrite dans l'ontologie, nous associons une donnée de contexte dite événement d_e . Par exemple, si nous prenons l'exemple de la figure 5.7 qui veut dire qu'une personne écoute la radio avant de dormir, nous pouvons choisir l'événement *allumer*

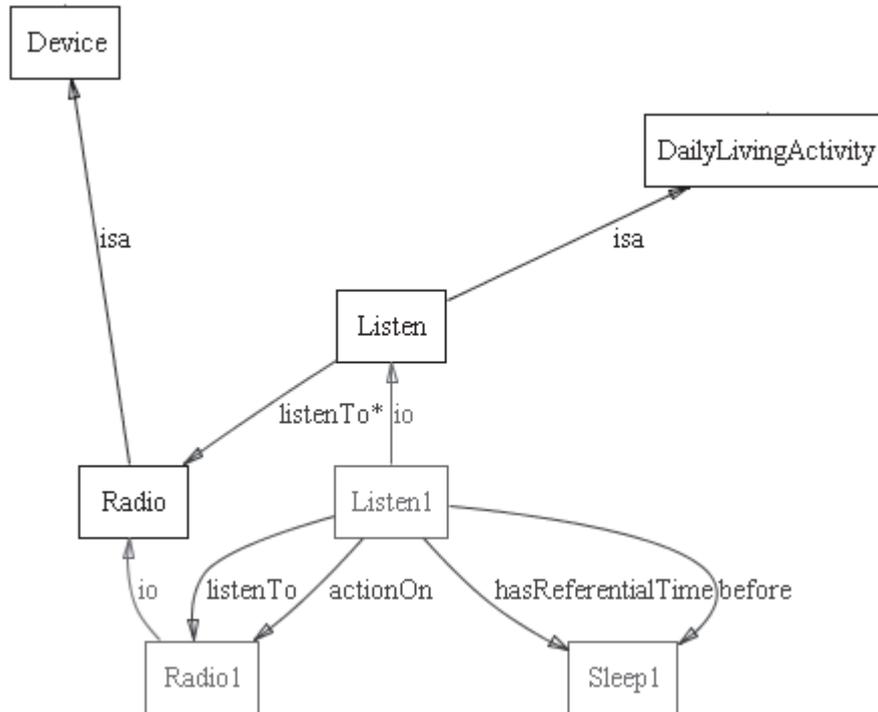


Figure 5.7 – Exemple de temps relatif

la radio pour caractériser la situation à découvrir.

Une fois que cet événement est détecté dans les données de contexte, la deuxième tâche consiste à construire autour de cet événement un cluster de données de contexte de façon à ce que chaque donnée de contexte d_i appartenant à ce contexte vérifie la règle suivante.

$$TimeStamp(d_e) - \mu/2 \leq Timestamp(d_i) \leq Timestamp(d_e) + \mu/2 \quad (5.7)$$

μ désigne la durée de ce groupement. Nous pouvons évaluer ainsi la découverte

sur des regroupements à des intervalles de temps de différentes largeurs autour d'un événement. Ensuite, à l'intérieur de chaque cluster, les triplets redondants sont fusionnés pour ne faire qu'un.

La dernière étape de notre algorithme est de calculer le nombre de triplets communs aux deux descriptions : situation à découvrir et groupement de données. Une correspondance positive existe entre les deux descriptions si :

$$\frac{NbTC}{\min(NbTS, NbTG)} > \rho \quad (5.8)$$

où, NbTC est le nombre de triplets en communs, NbTS est le nombre de triplets de la situation à découvrir, NbTG est le nombre de triplets du regroupement et ρ désigne un seuil. Cette équation reflète le fait que les deux descriptions sont d'autant plus similaires que le nombre de triplets communs est grand.

En résumé, cette approche descendante est intéressante car elle permet de décrire des situations plus riches avec des concepts et des relations qui sont pris en compte dans la structure d'une donnée de contexte. Seulement, cette approche ne permet pas un mécanisme adaptatif de services car la description des situations est laissée à la charge du concepteur mais d'autres mécanismes semi-automatiques peuvent être envisagés, par exemple pour faire une notification si la situation pré-établie est découverte dans les données de contexte. Dans le chapitre des expérimentations, nous présentons un exemple de modélisation de situation et comment elle peut être identifiée selon cet algorithme à partir des données de contexte.

5 Conclusion

Dans ce chapitre, nous avons d'abord présenté une nouvelle notion : la découverte de contexte. La découverte de contexte a pour objectif d'identifier des situations au fur et à mesure du déroulement de l'application sans trop se soucier de la dynamique de l'environnement. Nous avons présenté une ébauche fondée sur le clustering qui sera appuyée par une étude expérimentale dans un chapitre dédié

à cet effet. Nous n'avons pas exploré tous les algorithmes existants à cause de leur nombre élevé mais nous avons distingué les approches de découvertes par rapport à leur capacité à supporter un mécanisme adaptatif de service indépendamment du choix de l'algorithme de clustering lui-même.

Le fait de cerner le problème de découverte du contexte et de le voir sous plusieurs angles, nous a permis de faire les choix d'implémentation adéquats : nous avons sélectionné une méthode de découverte qui supporte des mécanismes automatisés. La découverte de situations fréquentes selon l'approche de regroupement de contexte répond à cette exigence. Nous l'avons retenue pour qu'elle soit implémentée dans un démonstrateur dans le cadre du projet européen MIDAS.

Dans le chapitre suivant, nous explorons l'algorithme de découverte de situations fréquentes dans une architecture d'adaptation automatique de services.

CHAPITRE 6

ARCHITECTURE D'ADAPTATION DE SERVICES

Dans ce chapitre, nous nous intéressons à l'approche ascendante de découverte des situations fréquentes et d'identification de situations inhabituelles pour l'adaptation automatique des services. Comme nous l'avons montré dans le chapitre précédent, la découverte aveugle de situations ne permet pas d'interpréter le type d'une situation trouvée. En revanche, les situations fréquentes présentent un grand intérêt car elles apportent une information sur des situations répétitives voire des habitudes de personnes dans l'espace intelligent. Nous voulons tirer profit des services utilisés ou activés durant les situations fréquentes pour adapter les services en cas de dérive par rapport aux situations déjà reconnues. L'avantage de l'architecture que nous proposons est sa portabilité pour d'autres environnements et sa capacité à suivre les changements qui peuvent intervenir par l'ajout ou la suppression de sources de contexte. Elle est de plus capable de découvrir de nouvelles situations dues aux changements des habitudes de vie de la personne.

Nous proposons une architecture modulaire d'adaptation au contexte des services qui comprend une brique d'acquisition et de stockage des données de contexte dans une base de données, une brique de découverte et une brique dédiée à l'adaptation. Dans la suite de ce chapitre, nous détaillons l'architecture globale et son fonctionnement puis nous expliquons le travail fait pour chaque module et nous justifions les choix techniques faits. Nous terminons par la proposition d'un mécanisme d'adaptation de services.

1 Architecture générale

Le schéma de la Figure 6.1 montre notre architecture d'adaptation automatique des services au contexte. Les capteurs constituent un moyen intéressant de percevoir l'espace intelligent. De nos jours, les systèmes de communication dis-

posent de capteurs de plus en plus diversifiés, autonomes et disposant de capacité de calcul voire de raisonnement leur permettant de prendre en compte des données de toute nature.

Nous avons fait le choix de stocker les données de contexte en provenance de l'environnement dans une base de données. Elles sont alors disponibles pour l'identification des contextes pertinents pour les services embarqués dans l'environnement. Une fois l'identification achevée, l'adaptation consiste en l'utilisation des services en fonction du contexte déjà identifié et de façon automatique. L'identification du contexte s'appuie sur la couche la plus importante de notre architecture qui est la découverte du contexte. Notre idée est qu'une fois les situations découvertes à partir des données de contexte, nous procédons à une identification des services en relation avec les situations découvertes. Cette étape de fonctionnement de notre architecture est détaillée dans la 5ème section.

Notre architecture est structurée en couches. La communication entre ces couches est assurée par une interface propre et des protocoles bien définis. Elle comprend différentes briques logicielles, dont principalement :

1. L'acquisition des données à partir de capteurs, applications ou services Web.
2. La découverte de situations habituelles de l'utilisateur ou de son environnement.
3. La détermination de la pertinence des situations découvertes pour les services embarqués dans l'environnement.
4. La détection en temps réel d'une dérive de la situation courante par rapport à l'une des situations déjà apprises.
5. L'activation la notification, ou l'alerte selon la nature du service déterminé en 3. Pour le cas de l'activation, si l'utilisateur réagit inversement à l'action du système (c'est-à-dire il y a une détection de l'effet inverse de l'action

du service durant un temps de correction $t_{correction}$), le système supprime la situation pertinente et relance le processus de découverte.

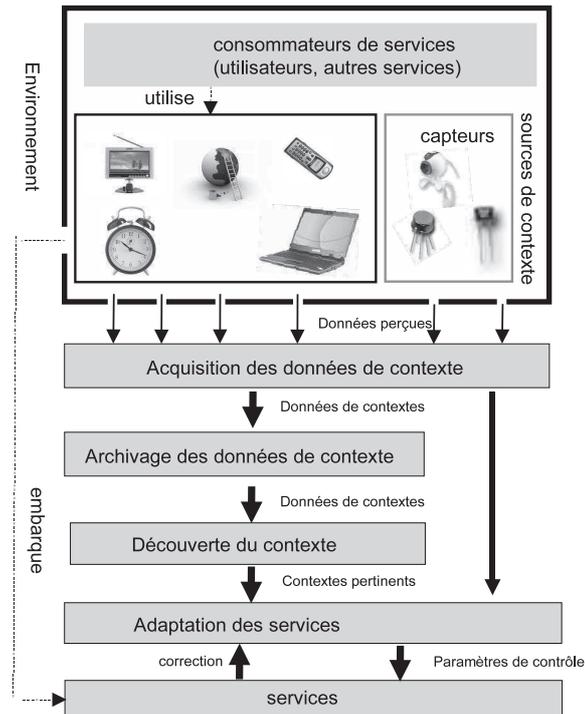


Figure 6.1 – Notre architecture d’adaptation automatique de services

Dans les sections suivantes, nous détaillons chacune des couches de notre architecture en spécifiant les relations entre elles. Nous justifions aussi les choix techniques faits par rapport aux différentes solutions existantes.

2 Acquisition des données de contexte à partir de capteurs

Dans cette section, nous intéressons d’abord aux formes d’acquisition du contexte pour montrer les difficultés qu’elle pose dans le cadre du développement d’un sys-

tème ouvert d'adaptation automatique de services ; puis nous présentons la solution adoptée pour préserver le caractère générique de notre architecture.

2.1 Formes d'acquisition des données de contexte

L'expérience qu'on a eu au sein à la fois d'une grande entreprise de télécommunication et d'un laboratoire disposant d'une plateforme matérielle complète d'acquisition, nous permet de classer l'acquisition de données sur l'utilisateur et son environnement en trois formes d'acquisition : à partir de capteurs portés, à partir de capteur non portés et à partir de services Web. Nous les détaillons dans les sous-sections suivantes.

2.1.1 Acquisition du contexte à partir des capteurs portés

Les capteurs portés sont des capteurs qui sont faciles à porter par l'humain ou à intégrer à son corps. Cette nouvelle technologie vise principalement à acquérir le flux informationnel utilisateur sur sa santé, ses activités et son comportement. Cependant, la fréquence d'arrivée des informations aux sources de contexte peut être trop élevée pour que les sources de contexte puissent construire les modèles de données (c'est notamment le cas lorsque des capteurs doivent renseigner sur les mouvements de l'utilisateur). Une solution peut consister en l'ajout d'un buffer en entrée des sources de contexte concernées.

La Figure 6.2 montre un exemple d'architecture employant des capteurs portés. Quelques exemples d'utilisation de ce type de capteurs pour la détection de situations sont donnés ci-dessous :

- Dans [32], deux accéléromètres sont placés sur la face avant du genou d'une personne pour la reconnaissance de son activité (debout, assis, marcher, courir et faire du vélo).
- Dans [44, 118], les capteurs sont fixés sur le mobile afin de détecter l'activité utilisateur (marcher, assis) et la situation de l'environnement (le mobile

est sur la table, à l'intérieur du bâtiment ou à l'extérieur). Deux accéléromètres sont utilisés pour mesurer l'accélération, et des capteurs d'humidité, de contact, d'illumination, un thermomètre et un microphone sont utilisés pour mesurer l'environnement.

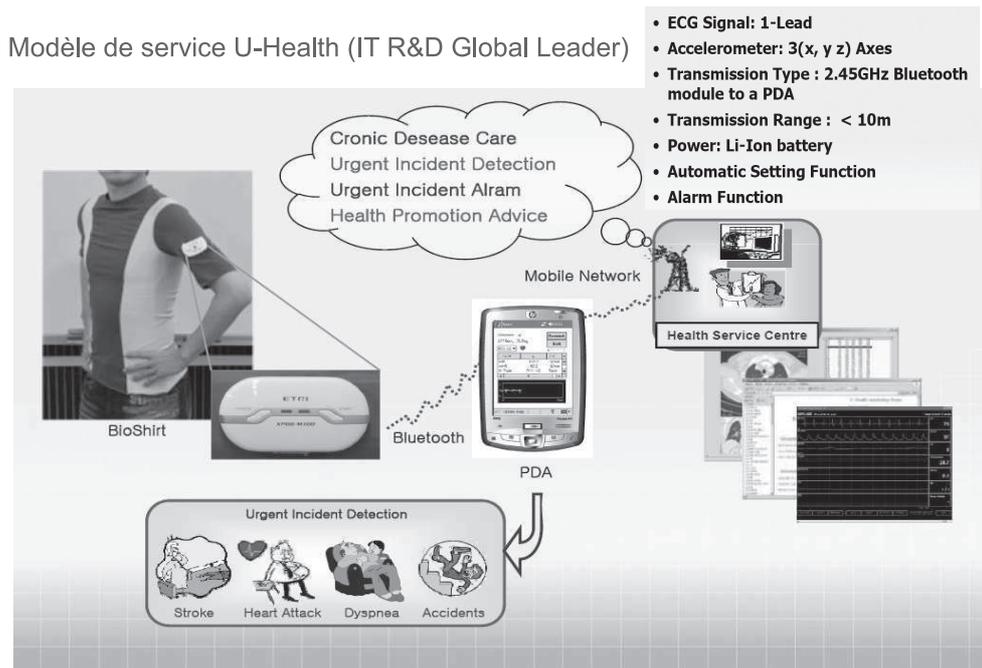


Figure 6.2 – Exemple d'architecture employant des capteurs portés

2.1.2 Acquisition du contexte à partir des capteurs non portés

Cette forme d'acquisition consiste à fixer un ensemble de capteurs dans l'espace intelligent et collecter les différentes données en vue de l'identification, la reconnaissance, la découverte d'informations sur le comportement utilisateur ou pour mener des études sur des types particuliers de capteurs.

Une des grandes réalisations d'infrastructure instrumentée est celle de PlaceLab¹ au MIT en Figure 6.3. Dans cet espace intelligent, une expérience a été

¹<http://www.placelab.org/PlaceLab>, consulté en Nov 2007

menée avec un couple marié à qui on a demandé de mener une vie normale sur une période de 10 semaines. Cet espace était équipé de plus de 900 capteurs (Switch, indicateurs, détecteurs, RFID) et environ 104 heures de données ont été annotées et analysées. Ce travail a exploité une ontologie qui contient 98 activités qui couvrent la majeure partie des aspects de la vie dans une maison. L'information sur les activités couvre même les activités les plus fines comme par exemple, le pliage du linge et le brossage des dents.

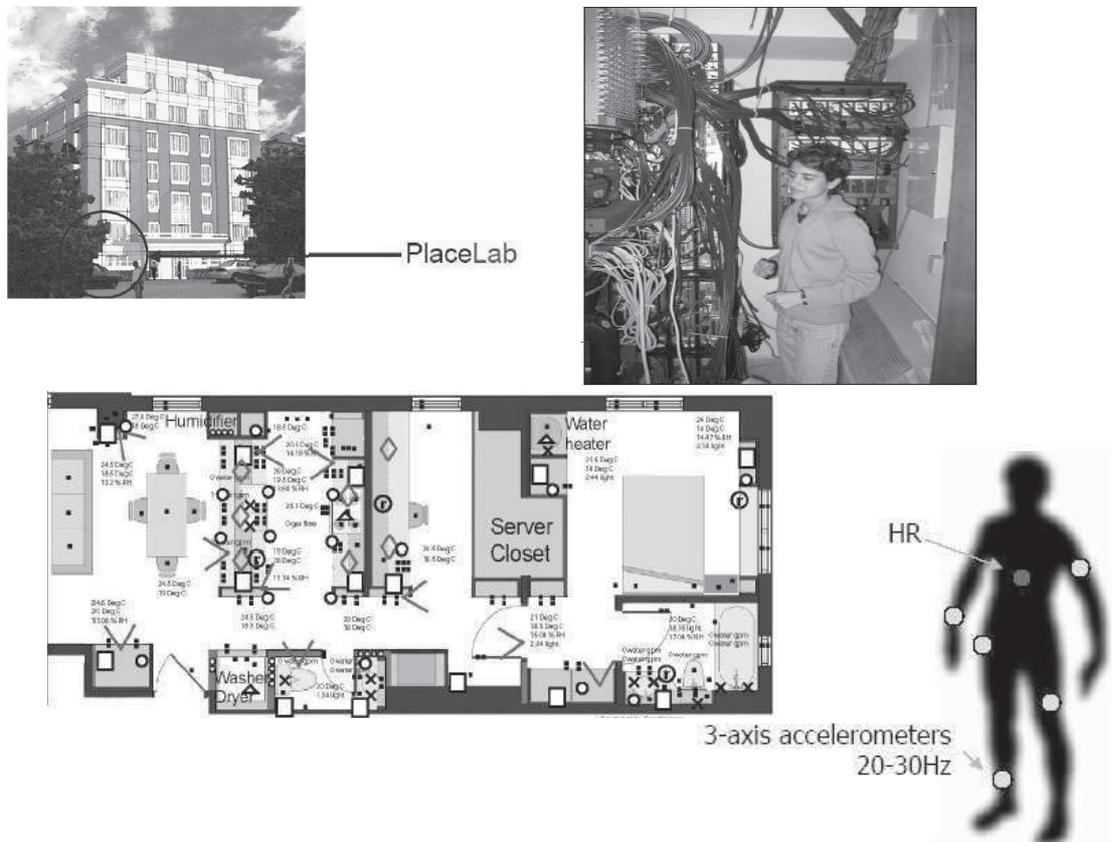


Figure 6.3 – Exemple d'architecture d'acquisition - PlaceLab

2.1.3 Acquisition non capteurs

Dans le cas des données de contexte ne provenant pas de capteurs, les informations sur l'utilisateur et son environnement sont fournies par des services Web, par exemple un serveur de météo ou un serveur de base de données médicale, etc.

2.2 Notre choix technique

Les sections précédentes cachent clairement deux difficultés essentielles dans la mise en oeuvre d'un système d'acquisition du contexte qui sont l'hétérogénéité des sources de contexte et le format des données. Ces deux difficultés peuvent constituer un handicap si on veut développer un système facile à maintenir et assez générique pour être fonctionnel dans divers environnements.

La solution qu'on propose dans cette thèse est celle déjà développée dans le cadre du projet Amigo² auquel nous avons contribué. Amigo est un projet collaboratif européen. Il a proposé un système de gestion de contexte appelé *CMS* pour (Context Management System). L'idée de cette réalisation est l'encapsulation de chaque capteur de l'environnement comme une source de contexte implémentée comme une interface logicielle entre le capteur réel et le système de gestion du contexte. La Figure 6.4 montre les deux types de fonctionnement du CMS.

Le CMS est chargé de collecter et de redistribuer des données liées au contexte parmi les composants du système Amigo. Toute donnée recueillie, ou produite par un service ou composant, peut-être assimilée à une donnée de contexte. Tout composant Amigo peut facilement devenir une source de contexte en implémentant l'API *IContexteSource* qui va mettre ses informations à la disposition des autres composants par la définition de trois méthodes *Subscribe*, *Unsubscribe* et *Query*. Les sources de contexte selon Amigo doivent être publiées comme des services Web. Ces services, sources de contexte, peuvent-être interrogés de manière synchrone (*query-driven*) ou asynchrone (*data-driven*). Dans le premier cas, c'est la

²<http://www.hitech-projects.com/euprojects/amigo/>, Amigo-IST, consulté 2007

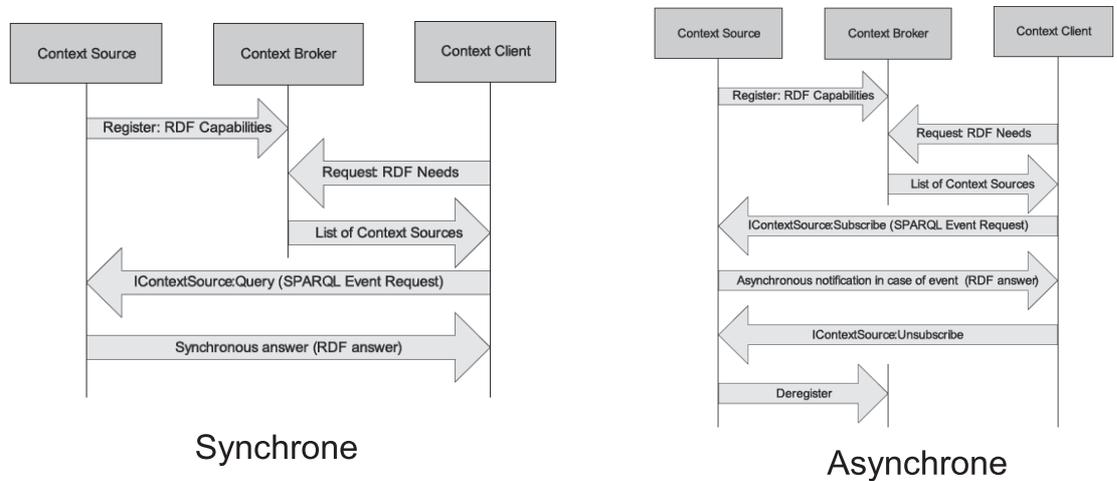


Figure 6.4 – L’acquisition du contexte selon Amigo

méthode Query qui doit être utilisée. Elle retourne les données de contexte requises. La requête doit être formulée avec le langage SPARQL (basé sur une modélisation RDF des données)³. Dans le second, le consommateur d’une donnée de contexte s’abonne auprès du service par la méthode Subscribe. Lorsqu’une donnée est générée, elle est automatiquement envoyée au subscriber. Cet abonnement prend fin avec l’appel de la méthode Unsubscribe.

Ce système d’acquisition a été adapté par notre équipe du laboratoire Orange-Labs pour concorder avec les besoins technologiques des scénarios que nous devons tester ; puis il a été complété par nous même pour être intégré à notre architecture d’adaptation automatique des services. Nous reviendrons sur ce point au chapitre implémentation. La Figure 6.5 montre comment une donnée de contexte est acquise selon le modèle présenté dans le chapitre 4. Chaque capteur réel de l’espace intelligent est encapsulé comme une source de contexte. Ainsi, la donnée récupérée par un consommateur de contexte quelconque est de la même forme que le modèle d’une donnée de contexte. La récupération des données de contexte par

³<http://www.w3.org/TR/rdf-sparql-query/>

notre système utilise la méthode de souscription auprès des sources de contexte afin de recevoir uniquement les changements dans les valeurs de capteurs. Un protocole de communication entre capteurs et sources de contexte a également été défini dans Amigo.

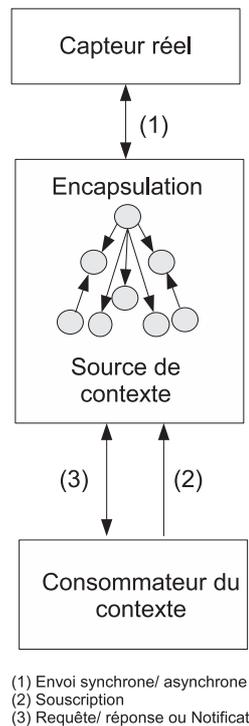


Figure 6.5 – Acquisition d’une données de contexte

Le travail proposé par Amigo pour la collecte et la redistribution des données de contexte présente une approche unique par rapport aux travaux existants. Les autres efforts de recherche se sont plutôt focalisés sur la prise en compte et l’intégration du contexte au sein d’architectures existantes comme on l’a dit dans l’état de l’art. Le travail effectué dans Amigo est plus spécifiquement dédié à être exploité et repris par les applications utilisant le contexte. Dans la section suivante, nous présentons la brique d’archivage des données de contexte. Elle est directement liée au CMS.

3 Archivage des données du contexte

Nous avons montré dans la section précédente comment les données de contexte sont acquises à partir de capteurs et services Web. Les flux de ces données dans certaines applications peuvent être très importants et arrivent à des vitesses élevées à cause du changement dans l'environnement qui induit un changement rapide des états ou des valeurs des capteurs. Si on ajoute à cela la caractéristique temporelle des données et le nombre de sources de contexte déployées, ces facteurs multiplient rapidement la taille des données stockées. Il est clair que ces données ne peuvent pas être stockées au niveau des sources de contexte et la solution la plus logique est de stocker ces données dans une base de données. Cela convient bien aux algorithmes de découverte de contexte qu'on a présenté dans le chapitre précédent.

Comme nous l'avons dit dans la section précédente le fait de ne considérer que les changements dans l'environnement a résolu plusieurs problèmes liés à la redondance des données qui posent essentiellement des difficultés de stockage et de temps d'accès. Cependant, les données enveloppées par des schémas sémantiques comme celles des données de contexte apportent des difficultés supplémentaires de stockage. Il faut trouver les techniques qui permettent de garder conjointement les données et leur sémantique. Cela permettra notamment d'exprimer des requêtes plus significatives (ex. sélectionner les différentes températures enregistrées de la semaine).

3.1 Solutions de stockage existantes

Le problème de stockage d'un volume important de métadonnées a été très adressé en littérature. Des efforts de recherche proposent des solutions complètement nouvelles de gestion de bases de données RDF ⁴ ⁵. D'autres ⁶ exploitent

⁴<http://agraph.franz.com/allegrograph/>, consulté 2008

⁵<http://www.openrdf.org/>, consulté 2008

⁶<http://jena.sourceforge.net/> consulté en 2007

les avancées des techniques traditionnelles de gestion de bases de données notamment les bases de données relationnelles pour mettre en oeuvre des solutions adaptées à ce type de données.

Nous nous intéressons particulièrement aux différentes solutions de stockage des données RDF vu la modélisation de nos données de contexte [14]. Il existe deux solutions dans la littérature. L'une s'intéresse à implanter RDF en natif comme dans le projet Mulgara⁷. L'un des avantages est d'optimiser le stockage et la restitution du contenu exprimé en RDF. En effet, le stockage en natif ne nécessite pas la sauvegarde de données supplémentaires et réduit le temps de recherche dans la base.

La deuxième solution de stockage de données RDF est d'exploiter les solutions existantes en termes de bases de données. Le passage de RDF vers la base de données et vice versa se fait à travers un composant logiciel spécifique qui se charge de la correspondance de et vers la base de données. Une des solutions les plus populaires est Jena. Jena supporte plusieurs systèmes de base de données telle que MySQL, PostgresQL, SQL Server, Oracle et Derby. Jena crée et aménage sa propre table dans la base de données. Le nombre de données stockées dans ces tables est limité par le nombre de noms distincts utilisés pour identifier ces données.

3.2 Notre choix technique

En accord avec ce qui a précédé et les évaluations faites en [34, 136], la solution proposée par Jena semble pour nous plus intéressante car elle permet d'économiser les efforts de tout remettre en oeuvre afin que des solutions puissent répondre au mieux à nos attentes en termes d'espace de stockage, de vitesse et de possibilités d'accès aux données.

Notre module de stockage agit comme un consommateur de contexte qui souscrit aux différentes sources de contexte présentes dans l'espace intelligent. A

⁷<http://docs.mulgara.org>

chaque réception d'une donnée de contexte, le module de stockage exploite les possibilités de stockage offertes par Jena pour archiver en temps réel les données de contexte dans une base de données MySQL. La Figure 6.6 est une extension de la figure d'acquisition du contexte présentée en 6.5. Elle montre l'archivage de la donnée de contexte une fois acquise par la source de contexte. A chaque donnée de contexte correspond un identificateur unique dans la base de données MySQL, généré automatiquement par notre module de stockage.

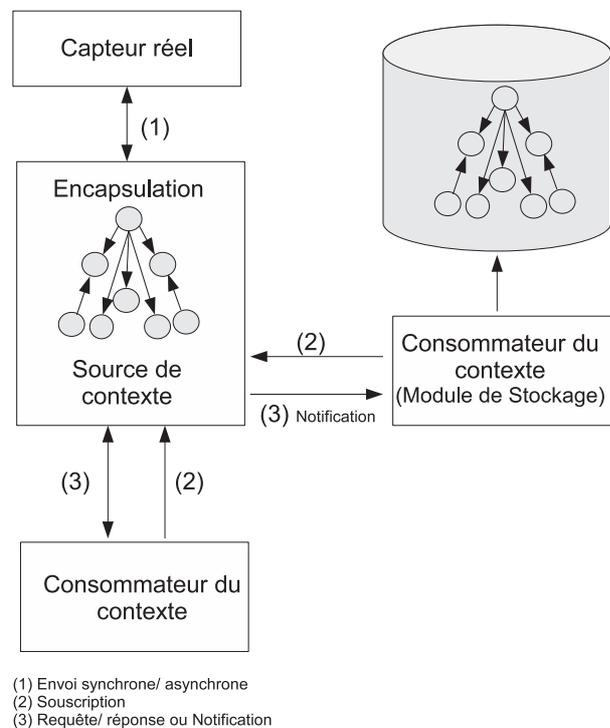


Figure 6.6 – Archivage des données de contexte

Dans la section suivante, nous verrons comment la base des données de contexte est exploitée par l'algorithme de découverte de situations fréquentes.

4 Découverte du contexte

Le module de découverte du contexte de notre architecture est basé sur l'algorithme de découverte de situations fréquentes présenté dans le chapitre précédent. Le but est de trouver des habitudes de l'utilisateur ou de son environnement et adapter automatiquement des services en conséquence.

Une fois les données de contexte stockées dans la base de données, l'algorithme de découverte de situations fréquentes procède régulièrement à une fouille dans les données pour trouver les situations. Le terme régulier signifie une activation automatique du module de découverte selon des périodes de fonctionnement du système qui peuvent s'étaler d'une journée à un mois.

Le fonctionnement du module de découverte se résume comme suit : les données provenant de différentes sources à un instant t constituent une situation. Le changement d'au moins une valeur de ces données est considéré comme un changement de situation. Les situations au cours du temps sont comparées et regroupées par un algorithme de clustering. Chaque regroupement (cluster) correspondra à une occurrence d'une situation habituelle.

La Figure 6.7 montre comment le module de découverte s'intègre en pratique aux modules d'acquisition et de stockage. Dans le cas où des situations sont trouvées, elles sont directement exploitées par un algorithme d'identification des situations pertinentes pour les services fournis au sein de l'environnement. Dans la section suivante, nous détaillons cet algorithme en précisant notre solution vers une adaptation automatique de services.

5 Vers une adaptation de service

Dans cette section, nous proposons un mécanisme d'adaptation capable de faire bénéficier le service de l'expérience de son utilisation. Ce mécanisme devra être implémenté au niveau de la couche adaptation de notre architecture. L'idée d'adaptation développée ici est partagée en deux étapes. Dans la première, ce

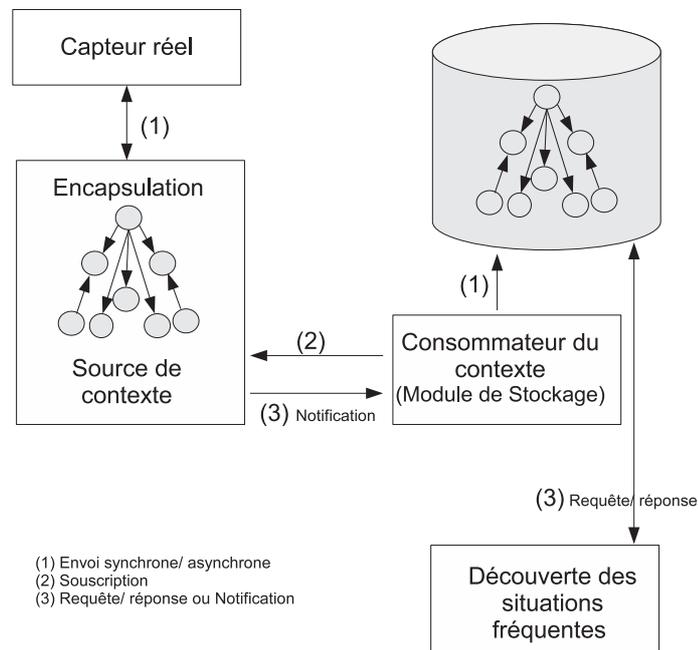


Figure 6.7 – Découverte des situations fréquentes

mécanisme commence par identifier les situations pertinentes pour des services embarqués dans l'espace intelligent de l'utilisateur et qui sont déclarés au sein de l'ontologie de contexte. Ces situations pertinentes sont déterminées parmi les situations fréquentes trouvées par l'algorithme de découverte de situations fréquentes du chapitre 5 section 3.1.1. En effet, toute situation fréquente n'est pas obligatoirement pertinente pour les services embarqués dans l'environnement. Supposant la situation fréquente "la personne monte les volets des fenêtres chaque matin", s'il n'existe pas au moins un service embarqué dans l'espace intelligent de l'utilisateur qui contrôle la remontée des volets, le système ne peut rien automatiser. Nous expliquons ci-dessous comment déterminer ces situations pertinentes.

En second étape, ce mécanisme réagit automatiquement pour corriger la situation en cours (qui représente une situation inhabituelle) en contrôlant, notifiant ou alertant l'utilisateur du système dans le cas d'une dérive par rapport à une si-

tuation pertinente. Une dérive indique un changement dans l'utilisation du service ou dans un état habituel de l'environnement. Dans le cas où l'utilisateur réagit inversement à la correction faite par le système, la situation pertinente correspondante est ignorée et une nouvelle découverte est lancée pour prendre en compte ces nouvelles données. Dans les sous-sections suivantes nous détaillons comment ces tâches peuvent être accomplies par le système.

5.1 Pertinence d'une situation fréquente

Le principe de détermination d'une situation pertinente est basé sur l'idée de regarder dans chaque situation fréquente découverte, les données de contexte qui correspondent à des effets de services. Nous utilisons pour cela l'ontologie définie au chapitre 4 qui décrit de manière générique les données provenant de sources différentes et les services qui peuvent être fournis dans l'espace intelligent. Nous attirons l'attention sur le fait que chaque service modélisé selon cette ontologie a une propriété "hasEffect" qui permet entre autres de faire le lien entre l'effet d'utilisation du service défini dans l'ontologie et les données de contexte perçues de l'environnement. Nous définissons formellement cette propriété comme suit :

Définition 5.1.1 (Effet de service). Soit $e_f \in \text{subClassOf}^*(\text{cxt} : \text{contextValue}, \mathcal{O})$, e_f est un effet d'un service ssi : $\exists s_r, e_n$, tels que $s_r \in \text{subClassOf}^*(\text{Service}, \mathcal{O})$ et $e_n \in \text{subClassOf}^*(\text{cxt} : \text{Entity}, \mathcal{O})$ et :

- $s_r \text{ hasEffect } e_f$
- $s_r \text{ hasActionOn } e_n$

Chaque donnée de contexte faisant partie d'une situation fréquente est mise en correspondance avec la description des services dans l'ontologie en posant une requête à cette ontologie. Si la situation présente un ou plusieurs effets de services, cette situation est considérée comme pertinente pour ces services.

Afin d'illustrer au mieux nos propos, les deux figures 6.8 et 6.9 présentent respectivement une donnée de contexte de l'état d'une lampe *on*, et une partie de

l'ontologie qui décrit les services de contrôle d'une lampe avec les effets correspondants. On suppose que cette donnée fasse partie d'une situation fréquente, on prend les deux valeurs de l'entité et du contexte (*BedroomLight, on*) de la donnée et on vérifie si cela correspond à un effet de service en utilisant une requête exprimée en Sparql ci-dessous. La réponse de la requête est le service : *lampeOn-ControlService* défini dans l'ontologie.

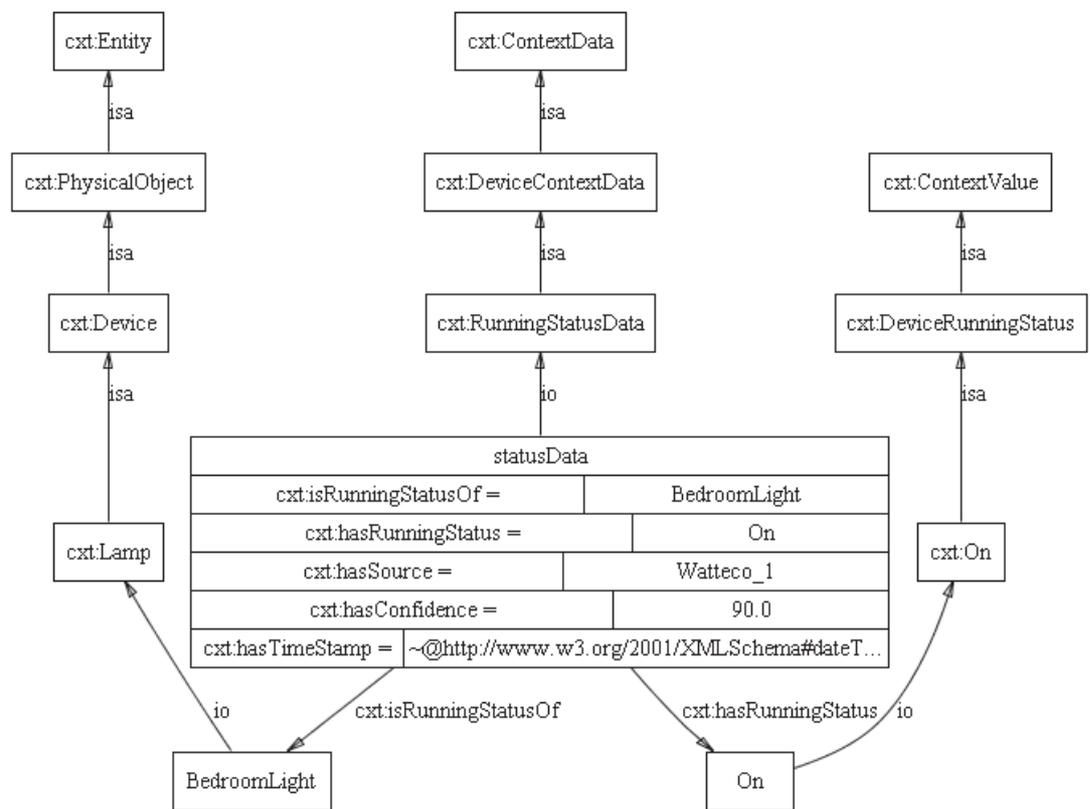


Figure 6.8 – Description dans l'ontologie d'une donnée de contexte

A la fin du parcours de toutes les situations fréquentes découvertes, le résultat de cette étape d'identification est un couple d'informations qui représente la situation et les services utilisés qui se réfèrent à cette situation. On le représente

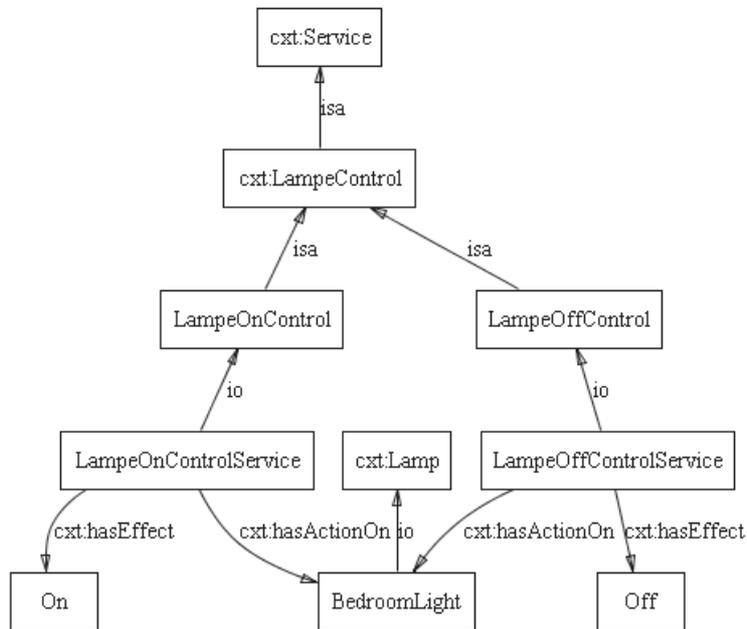


Figure 6.9 – Une partie de l’ontologie décrivant les services de contrôle d’une lampe

formellement comme $\langle S_h, S_{er}^{S_h} \rangle$. Les situations qui ne se réfèrent à aucun des services présents dans l’environnement sont simplement ignorées ou remontées à l’administrateur du système.

5.2 Détection d’une dérive

Une dérive représente un écart entre la situation courante et une situation pertinente. La situation courante décrit l’état présent de l’environnement. Elle correspond à l’ensemble des données de contexte à cet instant.

Pour trouver l’écart entre les deux situations, nous formons deux groupes de données de contexte dans chaque situation. Un groupe G_{def} formé des données qui correspondent à des effets de services et un groupe G_{dr} contenant les don-

Tableau 6.I – Requête SPARQL pour trouver un service

```

PREFIX base : <http://www.owl-ontologies.com/ContextDataOntology.owl#>
SELECT ?service
WHERE ?service base :hasEffect base :on .
?service base :hasActionOn base :BedroomLight

```

nées restantes. L'écart est alors calculé entre les données de contexte formant les groupes G_{dr} de chaque situation. Nous notons ce principe formellement comme suit :

Définition 5.2.1 (Groupe de données effets). *Soit S un contexte pur représentant une situation, on note G_{def} l'ensemble des données de contexte tel que : $G_{def} = \{d_i / d_i \in S, contextValue(d_i) \text{ est un effet de service}\}$. G_{def} est appelé groupe de données effets.*

Définition 5.2.2 (Groupe de données sans effets). $G_{dr} = \overline{G_{def}}$ est le complément de G_{def} , sachant un contexte pur S . G_{dr} est appelé groupe de données sans effets.

Si on regarde l'exemple de la situation pertinente et de la situation courante des figures 6.10 et 6.11, G_{def} de chaque situation contient une seule donnée (RunningStatusData).

Nous détaillons à présent de manière formelle le principe pour déterminer un écart entre deux données de contexte. Nous nous servons de cet écart pour définir une dérive.

Définition 5.2.3 (Écart entre deux données de contexte). *Soit d_i, d_j deux données de contexte avec $Source(d_i) = Source(d_j)$ où le signe = désigne l'égalité syntaxique (cela signifie que les deux données proviennent de la même source). Il existe un écart entre d_i et d_j ssi :*

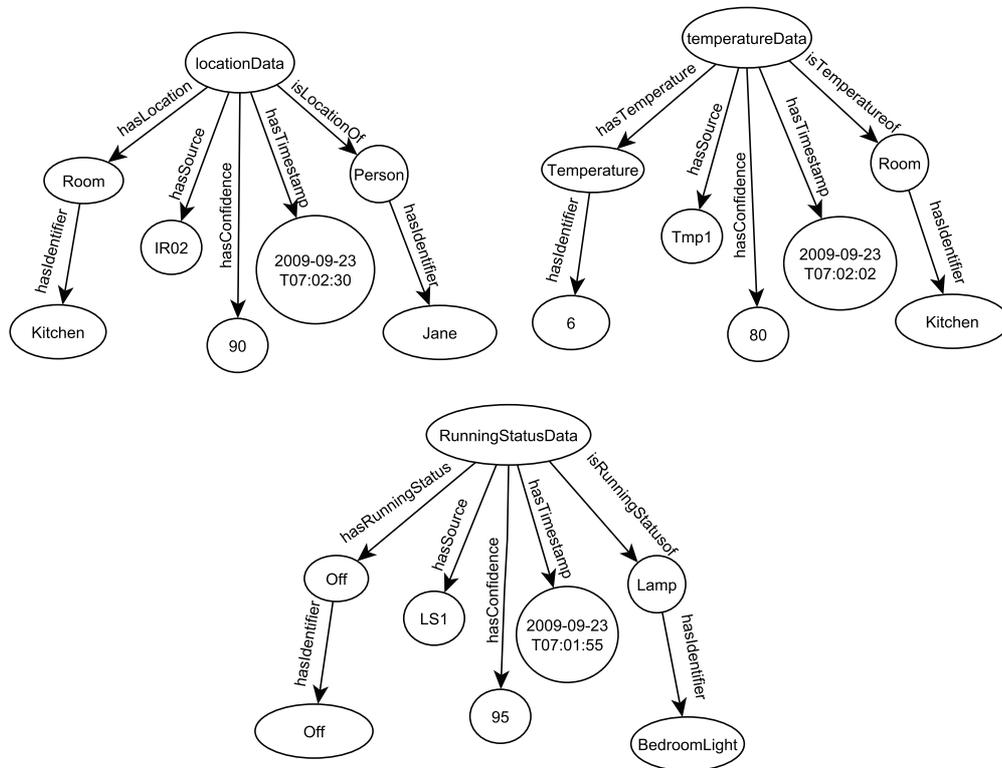


Figure 6.10 – Exemple de situation pertinente

- $\exists f(Timestamp(d_i), Timestamp(d_j)) = \text{vrai}$ tel que f est une fonction de $(Date, Date) \rightarrow (vrai, faux)$, elle indique si oui/non il existe un écart entre les moments de production des deux données,
- $| Confidence(d_i) - Confidence(d_j) | > c_a$, où c_a est la différence tolérée de la confiance,
- $\exists g(Identifier(ContextValue(d_i)), Identifier(ContextValue(d_j))) = \text{vrai}$ tel que g est une fonction de $(cxt : Contextvalue, cxt : ContextValue) \rightarrow (vrai, faux)$, elle indique si oui/non il existe un écart entre les deux valeurs de contexte.

Les figures 6.10 et 6.11 montrent un exemple de situation courante et de situation pertinente. Pour trouver une dérive entre ces deux situations, nous avons

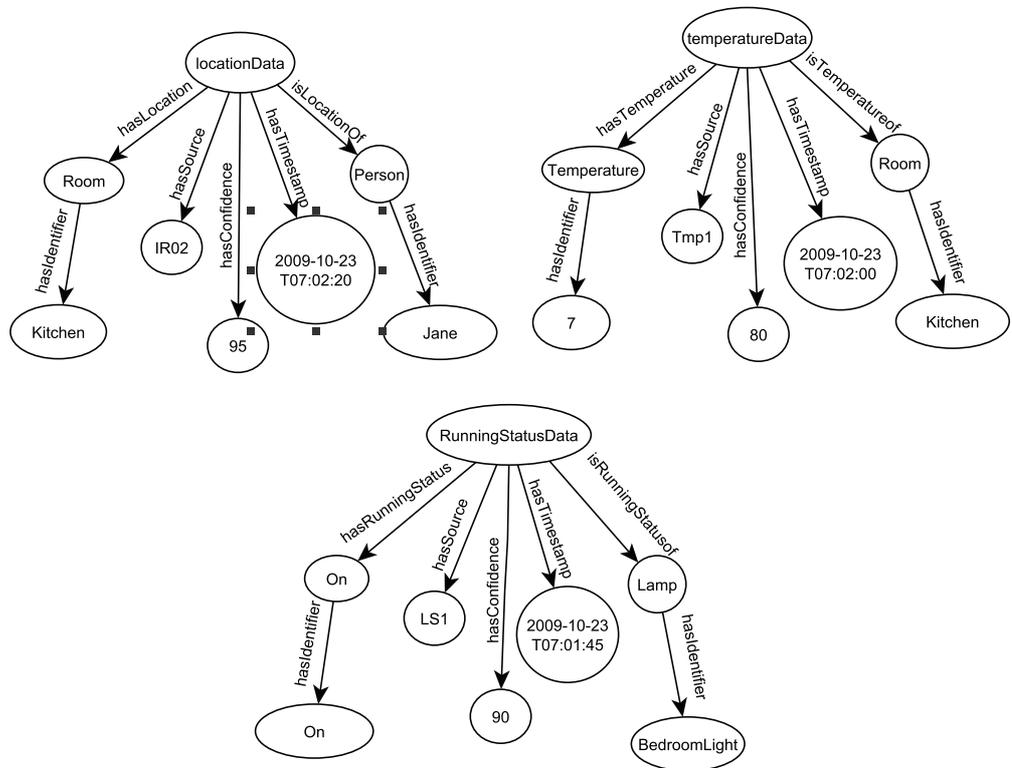


Figure 6.11 – Exemple de situation courante

besoin de déterminer s'il y a un écart entre les deux situations. Nous utilisons pour cela les deux fonctions f et g . La fonction f détermine si un écart existe sur la base du *Timestamp*. Si l'écart avec la situation courante est de quelques secondes par exemple, on suppose qu'il n'y a pas d'écart (cf. les figures 6.10 et 6.10). La fonction g est plus simple. Elle détermine l'écart entre deux valeurs numériques (par exemple la différence de température).

Si on définit des seuils pour la comparaison de la situation courante avec les situations pertinentes, c'est parce qu'une correspondance exacte est pratiquement rare ; par exemple il pourrait y avoir un décalage de quelques secondes dans le timestamp ou dans la valeur de confiance et de même pour l'identifiant d'une valeur de contexte. Néanmoins, ces décalages peuvent être tolérés car ils ne présentent pas un grand éloignement par rapport à une situation initiale c'est-à-dire

une situation pertinente. Une dérive se définit alors comme suit :

Définition 5.2.4 (Dérive). *Soit S_c une situation courante et S_p une situation pertinente telles que S_c et S_p sont des contextes purs de taille N . Il existe une dérive entre S_c et S_p ssi :*

- $\forall d_i \in G_{dr}^{S_c}, d_j \in G_{dr}^{S_p}$ avec $Source(d_i) = Source(d_j)$, alors \nexists écart entre d_i et d_j .
- $\forall d_i \in G_{def}^{S_c}, d_j \in G_{def}^{S_p}$ avec $Source(d_i) = Source(d_j)$, alors $contextValue(d_i) \neq contextValue(d_j)$

La dérive peut être détectée, s'il n'y a pas d'écart entre situation courante et situation pertinente mais que les données de contexte qui représentent des effets de services dans les deux situations, soient inversées. On prend l'exemple suivant pour être plus clair : si dans une situation courante on a *lampe allumée* et dans la situation pertinente on a *lampe éteinte*, alors cela veut dire que d'habitude la lampe est éteinte, mais que ce n'est pas le cas dans la situation courante. Le système réagit en utilisant le service `lampeOffControlService` pour mettre la lampe dans l'état éteint. Ceci en utilisant la requête SPARQL décrites ci-dessus avec les paramètres de la situation pertinente (*BedroomLight, Off*). L'exécution de ce service se fera en respect du protocole et de la configuration matérielle installée. Il peut s'agir d'une simple commande d'un actionneur de type Zigbee par exemple ou d'une description dans un langage plus évolué comme le langage WSDL/SOAP (Web Services Description Language, Simple Object Access Protocol).

5.3 Détection d'une correction utilisateur

La correction utilisateur peut intervenir sur les services de contrôle car pour le cas de la notification ou de l'alerte, l'utilisateur peut simplement les ignorer s'il juge que ce n'est pas important pour le cas courant. Dans le cas où l'utilisateur corrige l'action du système, cela veut dire que la situation pertinente n'est pas totalement vraie. Supposant un utilisateur qui laisse la radio allumée chaque nuit. Le

système va considérer que c'est une situation fréquente. Hors, un jour l'utilisateur a une migraine et n'a pas allumé la radio. Le système va mettre la radio en marche sur la fréquence habituelle de l'utilisateur et attendre un temps $t_{correction}$. Si l'utilisateur corrige cette action en éteignant la radio pendant cette période, le système va supprimer la situation fréquente de l'ensemble des habitudes utilisateur. Puis, il relance la découverte pour la prise en compte de ces nouvelles données. Il peut arriver qu'il découvre une situation fréquente qui dit : "l'utilisateur laisse la radio allumée uniquement les nuits des jours de travail". Dans le cas où l'utilisateur ne réagit pas à l'action prise par le système, l'action est considérée comme correcte.

6 Conclusion

La principale contribution de ce chapitre consiste en la proposition à la fois d'une architecture globale d'adaptation automatique des services qui exploite les résultats de la découverte du contexte, et d'un mécanisme d'adaptation basé sur deux méthodes : l'identification des situations pertinentes à partir des situations fréquentes et la détection d'une dérive de la situation courante par rapport à celles-ci. Cette proposition combine les technologies du Web sémantique, les services web et les algorithmes de fouille de données.

De plus, les systèmes basés sur cette architecture auront l'avantage d'être plus autonomes car ils seront capables de déterminer sans aucune intervention de l'utilisateur, les situations auxquelles il faut réagir et les services à utiliser.

Cependant, certaines limitations demeurent. L'étape d'identification exige un travail très minutieux sur la modélisation de l'ontologie de l'espace et sur le modèle de données de chaque source. C'est cette dépendance qu'on reproche souvent à ce genre de système car le résultat devient rapidement dépendant de la vision du constructeur et de sa manière de percevoir l'environnement. En effet, l'identification des situations pertinentes pour un service s'appuie sur le modèle de données de contexte et l'ontologie de l'espace intelligent pour déterminer les services. Cela peut être problématique lorsque les deux descriptions (au niveau données et au ni-

veau ontologie) sont incompatibles.

A ce stade d'étude, nous avons proposé une modélisation du contexte, des approches de découvertes et une architecture d'adaptation automatique de services. Nous nous appuyons sur ces fondements théoriques pour montrer le cadre expérimental de la découverte du contexte et les détails d'implémentation logicielle de cette architecture.

Troisième partie

Application et validation

CHAPITRE 7

CADRE EXPÉRIMENTAL

Ce chapitre est dédié au cadre expérimental qui a été conduit durant cette thèse. L'objectif des expérimentations décrites dans ce chapitre a été en premier de déceler une méthode de découverte adéquate selon une approche d'apprentissage non supervisé et qui peut être incorporée dans notre architecture d'adaptation automatique de services. En second lieu, l'objectif était de soulever les différents problèmes liés l'implémentation, principalement son adéquation aux contraintes de mise en oeuvre telles que la dynamique des sources dans l'environnement et la robustesse des mesures de similarité.

Nous présentons d'abord les expérimentations essentielles qui concernent l'approche ascendante et l'approche descendante puis les travaux d'expérimentation de moindre importance. Nous mettons l'accent en premier sur la découverte de situations fréquentes identifiée au chapitre 5 comme approche prometteuse à l'adaptation automatique de service. Ensuite, nous détaillons quelques travaux d'expérimentation qui ont permis de faire les bons choix théoriques et pratiques pour la conduite de cette thèse. Nous mettons en conclusion nos commentaires sur le travail conduit dans ces expérimentations.

1 Découverte de situations fréquentes

Ces tests ont été effectués à Orange-Labs (Laboratoire de Grenoble, France). Ils ont pour but d'étudier la découverte de situations fréquentes. La méthode utilisée est celle décrite dans le chapitre 5. Nous exploitons la méthode de clustering MCL en utilisant une méthode hybride faisant intervenir les fonctions de similarité conceptuelle et relationnelle. Nous avons fait deux évaluations : l'une avec un scénario simple et une autre sur un volume important de données. Ces tests vont nous permettre de conclure sur la validité de l'étape de découverte dans l'architec-

ture. Nous présentons ces deux expérimentations dans les sous-sections suivantes après avoir décrit le lieu des expérimentations.

1.1 Description du lieu des expérimentations

Les données ont été recueillies dans les locaux d'Orange Labs R&D pour des besoins multiples parmi lesquels le développement d'une gamme de passerelles M2M en anticipation, l'intégration de diverses technologies radio comme par exemple Zigbee, Wavenis et RFID.

1.2 Expérimentation sur un volume réduit de données

Cette expérimentation avait pour but de mettre en pratique l'algorithme de découverte de situations fréquentes sur un exemple simple pour valider son fonctionnement. Les données de contexte utilisées comme point d'entrée sont tirées du scénario suivant : Un capteur relève la présence d'une personne appelée "John" tandis qu'un autre transmet le statut d'une lampe de la chambre. La figure 7.1 schématise la structure de l'environnement et la figure 7.2 montre le déroulement du scénario dans le temps. On constate sur cette dernière figure les données de contexte en forme de croix qui relèvent d'une réception asynchrone (*envoi d'un signal lorsqu'une modification est effective*). Ce scénario est choisi de façon à voir s'il est possible de découvrir les 3 situations suivantes :

1. John est dans la chambre, la lumière de la chambre est allumée
2. John est dans le salon, la lumière de la chambre est éteinte.
3. John est dans la chambre, la lumière est éteinte

Le tableau de données en figure 7.3 donne un aperçu des éléments qui changent dans les données de contexte afin de montrer le déroulement du scénario. Ces données sont réellement des données sémantiques en respect du modèle présenté en chapitre 4 mais elles sont représentées sous forme de tableau et uniquement en

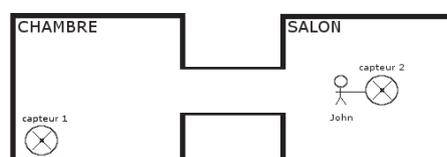


Figure 7.1 – Environnement

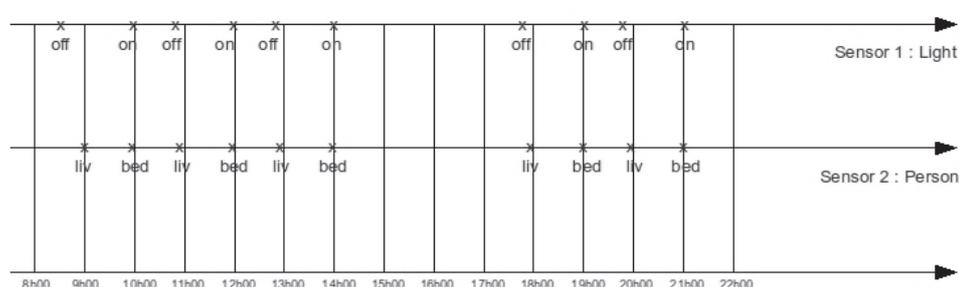


Figure 7.2 – Scénario

partie pour simplifier l’affichage. Chaque ligne du tableau représente une situation. Par exemple, la ligne 9 indique la situation où John est dans le salon et la lampe de la chambre à coucher est éteinte.

En utilisant la méthode de découverte des situations fréquentes présentée au chapitre 5, nous avons obtenu le résultat présenté en 2D et en 3D dans les figures 7.4 et 7.5 respectivement. Le logiciel utilisé pour la visualisation des résultats est BioLayout Express 3D ¹. Ce logiciel est conçu pour la visualisation de données biologiques mais il est utilisé ici car il permet de donner une bonne idée sur le bon déroulement des fonctions de similarités utilisées en observant la dispersion des situations dans l’espace. Les sources étant en libre accès, un travail d’intégration a été nécessaire pour améliorer les représentations des résultats de classification.

¹<http://www.bioblayout.org/>

Notamment une colorisation des différents clusters et la restriction de l'affichage à des groupes particuliers de données.

Les numéros sur ces figures sont les identifications des situations du tableau 7.5. Ils indiquent celles qui ont participé à la construction de chaque groupe. On les retrouve sur le tableau de la figure 7.3. Cependant, les liens entre les différentes situations est peu apparent sur les figures, on les voit en forme de hachures. Ces liens reflètent les mesures de similarité entre les situations.

Le résultat de cette méthode montre les trois clusters qui relèvent de l'analyse des trois contextes simulés lors de la construction du scénario, par exemple, le cluster des situations (17,37,..) de la partie gauche en bas des figures 7.4 et 7.5 indique situation fréquente : *John est dans le salon, la lumière de la chambre est éteinte.*

Situation	Capteur de lumière (sensor1)				Capteur de présence (sensor2)			
	Sensor1	LampeBedRoom	Off	90	null	null	null	nu
0	Sensor1	LampeBedRoom	Off	90	null	null	null	nu
1	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
3	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
4	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
5	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
7	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
8	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
9	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
11	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
12	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
13	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
15	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
16	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
17	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
19	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
20	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
21	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
23	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
24	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
25	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
27	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
28	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
29	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
31	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
32	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
33	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
35	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
36	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
37	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
39	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
40	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
41	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
43	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
44	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
45	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
47	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
48	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
49	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
51	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
52	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
53	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
55	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
56	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
57	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
59	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
60	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
61	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
63	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
64	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
65	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
67	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
68	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
69	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
71	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
72	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90
73	Sensor1	LampeBedRoom	Off	90	Sensor2	LivingRoom	John	90
75	Sensor1	LampeBedRoom	On	90	Sensor2	BedRoom	John	90
76	Sensor1	LampeBedRoom	Off	90	Sensor2	BedRoom	John	90

Figure 7.3 – Construction des situations

1.3 Expérimentation sur un volume important de données

L'algorithme a été mis à l'épreuve sur un volume important de données afin d'étudier son efficacité. On a testé l'algorithme sur une base de données répertori-

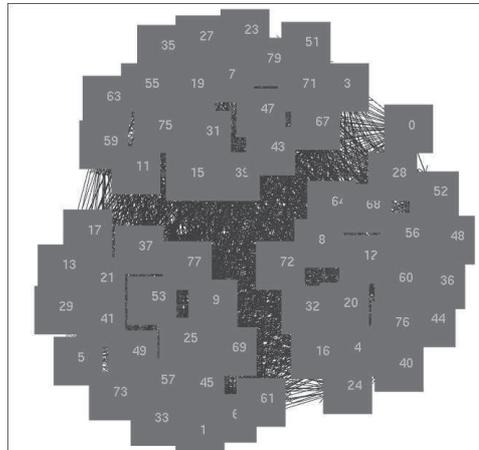


Figure 7.4 – dispersion des situations

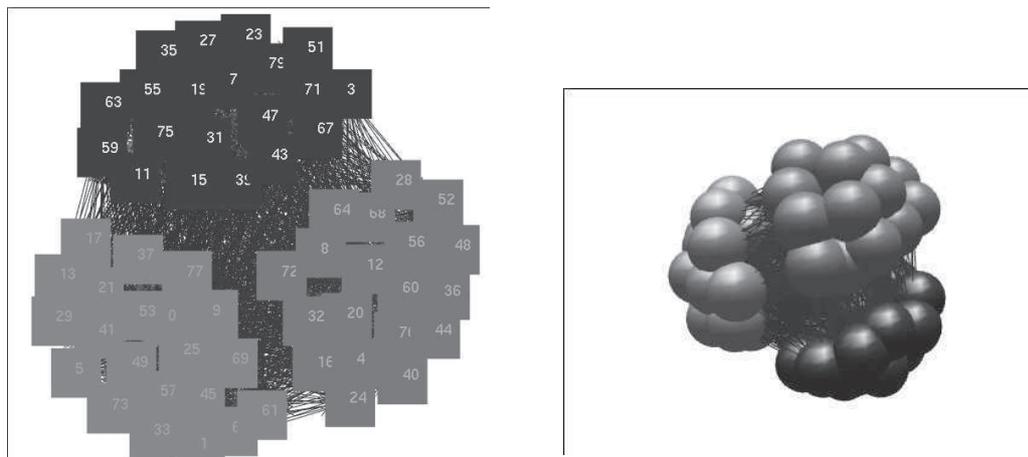


Figure 7.5 – Résultats du clustering

riant des centaines de milliers de données de température, luminosité, humidité, etc. Ces données sont collectées à Orange-Labs. Ces données ont la particularité de changer très rapidement. Pour ne pas à voir à gérer un nombre de situations énorme, nous avons éliminé les situations dont la durée est inférieure à 10 secondes. Ce délai est un paramètre qui permet de distinguer les situations non

significatives. Il peut être ajusté selon les besoins.

Indépendamment de la pertinence du résultat obtenue, nous avons évalué l'efficacité de l'algorithme de découverte de situations fréquentes en termes de temps d'exécution en secondes par rapport au nombre de situations, du nombre de capteurs présents dans l'environnement et du nombre de données de contexte. Le tableau 7.I présente ces résultats ; on voit clairement que le temps d'exécution est principalement occupé par le calcul des similarités. Il est évident que ce temps d'exécution croît en fonction du nombre de données de contexte. En particulier, le temps de calcul des mesures de similarité est très important pour une application temps réelle générant un nombre important de situations.

Tableau 7.I – Evaluation du temps d'exécution

Nombre de données de contexte	Nombre de situations	Nombre de capteurs	Temps du Tri Temporel	Temps de calcul des similarités	Temps de Clustering
260	250	2	4s	450s	2s
1000	450	6	60s	2000s	2s

Nous soulignons que malgré l'importance de vérifier le fonctionnement sur un nombre important de données, la difficulté réside dans l'interprétation des résultats obtenus car cela demande beaucoup de temps d'analyse et d'autres moyens de visualisation pour donner un jugement correct. Néanmoins, les échantillons que nous avons pris de manière aléatoire sur des périodes différentes confirment la construction de situations fréquentes.

2 Découverte hiérarchique de situations

Ce travail est tiré en partie du séjour de recherche qui a été encouragé par les efforts des différents partenaires scientifiques et industriel et financé en partie par la région Rhône-Alpes dans le cadre du programme Explora'doc 2008.

2.1 Description du lieu des expérimentations

Le séjour s'est déroulé au département d'informatique de l'Université de Sherbrooke dans le laboratoire DOMUS². DOMUS est une équipe multi-disciplinaires dotée d'une infrastructure de recherche intéressante. Sur le campus universitaire, le laboratoire dispose d'un appartement de 4 pièces utilisé pour mener des expérimentations relatives aux systèmes d'informations diffus afin de créer des habitats intelligents. La première cible est de concevoir une maison intelligente pour les personnes souffrant de déficits cognitifs afin de leur permettre de réaliser leurs activités quotidiennes en toute sécurité tant dans leur domicile qu'à l'extérieur. Pour réaliser cet objectif, toute une branche d'activités de recherche de DOMUS gravite autour des architectures orientées services, de l'assemblage dynamique et contextuel de services, de la localisation multi-personnes, de la personnalisation, des systèmes multi-agents, de la sensibilité au contexte et de l'agrégation de données provenant de capteurs multiples et hétérogènes pour produire des informations de contexte de plus haut niveau.

2.2 Infrastructure réseau et matérielle

Le laboratoire Domus a un partenariat avec ArianeControls³ pour l'utilisation des appareils de communication sur des lignes électriques. Ces appareils communiquent avec les capteurs et effecteurs du laboratoire. Le dispositif appelé "PRO2" est la partie principale de l'infrastructure domotique. Il contrôle les différents appareils (luminaires, vidéo, audio, etc.) en fonction des différents entrants (capteurs, keypads, etc ...). Les capteurs présents partout dans l'appartement sont de trois types :

1. capteurs d'état d'ouverture de porte,
2. capteurs infrarouge pour la détection de présence,

²<http://domus.usherbrooke.ca>

³<http://www.arianecontrols.com/>

3. Tapis tactile.

Ces capteurs doivent tous être reliés aux "ports I/O" du PRO2. En effet, tous ces capteurs fonctionnent par "contact sec". Ainsi, lorsqu'un de ces capteurs se déclenche, un événement est produit dans le PRO2. La façon dont "PRO2" traite cet événement dépend de la façon dont il est programmé. La Figure 7.6 ci-dessous montre le serveur d'événements ainsi que la structure de l'appartement.

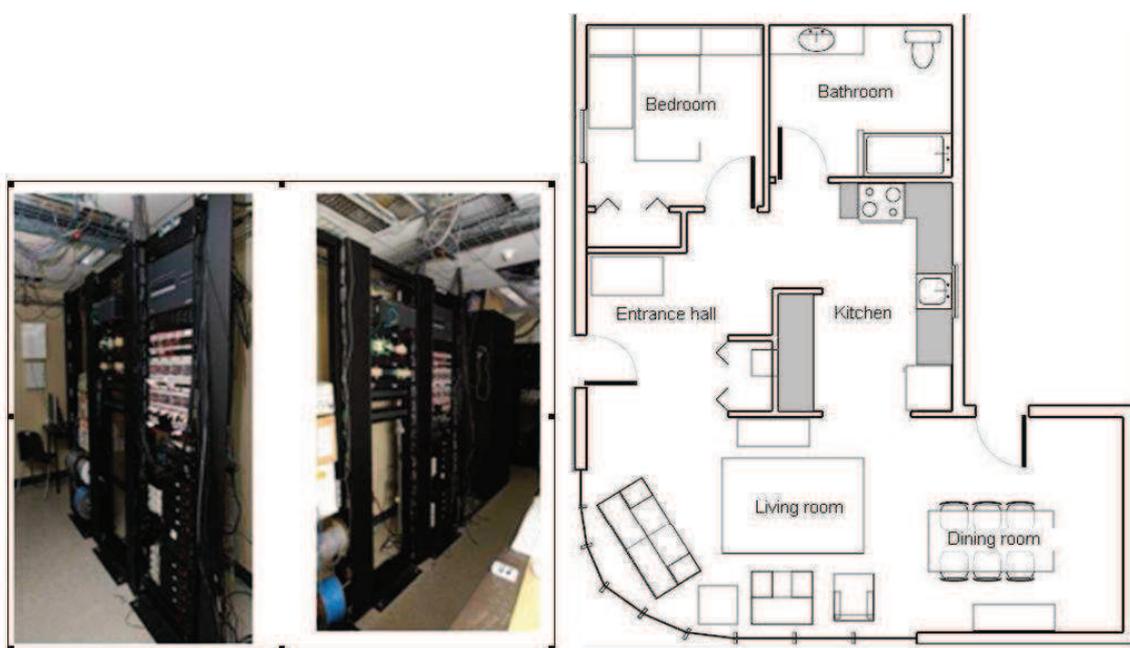


Figure 7.6 – Infrastructure réseaux du Labo Domus

2.3 Description du scénario

Nous avons appuyé notre travail sur le scénario effectué par Kadouche Rachid, membre du laboratoire Domus. L'exploitation de ces données est profitable pour les travaux conduits par les deux parties et nous a permis d'optimiser le travail de préparation de l'espace intelligent et nous a évité de nous confronter à des autorisations en terme de règles d'éthiques qui prennent généralement beaucoup de temps.

Dans ce scénario, le centre d'intérêt était les habitudes personnelles du matin : *toilette et prise du petit déjeuner*. Il y a deux versions du scénario. La première est appelée la série 1. L'utilisateur est invité à exécuter les routines du matin sans contraintes lors de l'exécution de ses tâches. Dans la deuxième (série 2), une contrainte est introduite. Elle lui impose de préparer un thé pour le petit déjeuner. Le script de la recette du thé est affiché sur une interface tangible dans la cuisine. La préparation de la recette du thé prend au plus dix minutes. Six personnes ont participé aux deux versions du scénario pendant l'expérience. Chaque personne a participé à la série 1 dix fois pendant deux semaines consécutives, et une semaine pour la série 2. Une pause de deux semaines est programmée entre les deux séries.

Le temps de l'expérience était d'environ 45 minutes. Le scénario commence avec les mêmes conditions dans l'appartement pour tous les utilisateurs, les portes fermées et les lumières éteintes. L'utilisateur est invité à rester dans la chambre pendant une minute (temps requis pour démarrer le processus d'enregistrement des données). Chaque échantillon des données de test est formaté selon le modèle d'une donnée de contexte vu en chapitre 4 en utilisant un programme spécifique car le système d'acquisition de données ne correspond pas au système d'acquisition Amigo qu'on a présenté dans le chapitre "Architecture". Le total des données est de 60 échantillons pour les séries 1 et 30 échantillons de la série 2.

Pour le besoin de cette expérience, trente-six capteurs fixes ont été utilisés. En outre, le laboratoire est conçu en six zones logiques correspondant aux 6 zones couvertes par les capteurs infrarouges de la figure 7.6. De plus, le laboratoire DOMUS est équipé comme n'importe quel appartement pour vivre. Les détails sur les types de capteurs utilisés sont présentés comme suit :

- Détecteurs infrarouge (IR) de mouvement : ils donnent des données sur la localisation des utilisateurs dans une zone. Il n'y a qu'un seul détecteur IR qui couvre toute la zone de la salle à manger et trois IR sont installés dans la cuisine.

- Détecteur de pression sous forme de tapis tactile : ils sont placés sur le plancher du hall d'entrée. Ils ont pour rôle de détecter le déplacement d'un utilisateur entre la chambre à coucher et le salon. Il y a deux chemins possibles pour se déplacer entre ces deux zones : par la cuisine ou par le hall d'entrée.
- Interrupteurs d'éclairage : ces capteurs envoient un événement à chaque mise en état allumé ou éteint de la lumière.
- Capteurs de contacts de portes : ces capteurs sont placés sur les portes. Ils envoient un événement lié à l'état de la porte (ouverte ou fermée).
- Autres capteurs de contacts : ils sont les mêmes que les contacts de portes. Ils sont placés sur les tiroirs et le réfrigérateur. Ils fournissent un événement lorsque leur état est modifié : ouvert ou fermé.
- Débitmètre : il fournit l'état des robinets et de la chasse d'eau des toilettes. Deux sont placés sur les robinets d'eau chaude et froide de l'évier de la cuisine et un autre sur la chasse d'eau. Ils envoient un événement lorsque le robinet est ouvert ou fermé et quand on tire la chasse d'eau.

Le tableau 7.II résume l'ensemble de ces capteurs par zone d'appartement.

Tableau 7.II – Liste des capteurs par zone

	Hall d'entrée	Salon	Salle à manger	Cuisine	Salle de Bain	Chambre à coucher	Total
IR	0	1	1	3	0	0	5
Détecteur de pression	1	0	0	0	0	0	1
Lampes	0	1	1	1	1	1	5
Capteurs de contacts de portes	0	0	0	0	1	1	2
Autres capteurs de contacts	0	0	0	19	0	0	18
Débitmètres	0	0	0	0	2	2	4

2.4 Résultat de l'algorithme hiérarchique

Dans cette section, nous décrivons les détails des paramètres utilisés par "*Agglomerative clustering*", puis nous présentons les résultats obtenus. Ceci dit, ce

résultat a été précédé par plusieurs tests que nous ne reprenons pas dans cette section car le but est de se focaliser sur les résultats les plus pertinents de chacune des expérimentations.

La mesure de distance utilisée combine trois des mesures conceptuelle, syntaxique et la proximité temporelle décrite dans le chapitre 5. De plus, la distance entre deux clusters est obtenue avec la distance entre les deux données les plus proches dans les deux groupes (*single-linkage*).

La proximité temporelle est calculée avec une valeur $T = 10$ minutes (*la durée d'une période des données de contexte*). La décroissance temporelle a est de 86. Cette valeur assure que la décroissance est stable si la différence entre les instances des données de contexte est inférieure à 7 secondes. Les autres mesures conceptuelle et syntaxique ont été calculées en exploitant la librairie du logiciel libre Simpack ⁴ dans nos programmes d'expérimentations. Simpack est une bibliothèque open source extensible de similarités ou de distances. Cette bibliothèque implémente la majorité des similarités de bases utilisées en littérature.

Le temps de l'expérience était d'environ 45 minutes, valeur arbitraire pour couvrir le temps requis par le scénario ci-dessus. Les données de contexte sont divisées par période de temps fixe (10 minutes). Ce délai est fixé pour permettre d'un côté une bonne analyse des résultats par comparaison aux données réelles enregistrées et annotées et pour obtenir la visualisation complète du dendrogramme. Ce délai permet en outre la présence de données ambiguës (par exemple l'envoi de données de localisation contradictoires par deux capteurs infrarouge dus au chevauchement de zones).

Deux coupes possibles du dendrogramme sont présentées dans la figure 7.7 : à 0,3 et à 0,4. Le tableau 7.IV correspond à un dendrogramme coupé à une similarité minimum de 0,3. Il montre d'un côté les contextes d'utilisateur et les services connexes activés. On mentionne ici que l'identification des services est faite manuellement car l'algorithme d'identification n'était pas encore prêt. Par contre,

⁴<http://www.ifi.uzh.ch/ddis/simpack.html>

on peut constater que les groupes (6,8,10,12,7,9,11) et (13,15,17,14,16) sont à la droite du dendrogramme. Ce dernier groupe respecte l'ordre temporel mais son affichage est décalé pour permettre une meilleure lisibilité. De plus, nous pouvons remarquer quelques incohérences qui sont dues aux conflits entre l'état ouvert et fermé de l'eau froide du robinet pendant l'expérimentation. Cela s'est révélé être dû à la sensibilité du capteur débitmètre. Les services auxquels nous sommes intéressés dans cette expérimentation sont les services de contrôle de la lumière. Le laboratoire DOMUS dispose des boîtes de commandes qui permettent de contrôler automatiquement les lumières dans l'habitat.

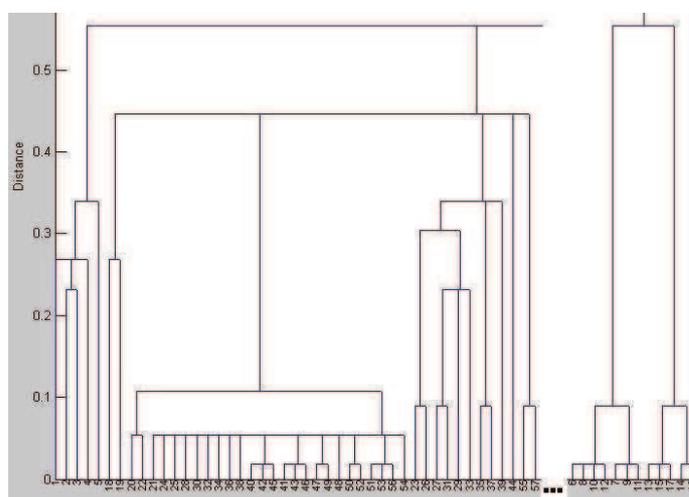


Figure 7.7 – Une portion du dendrogramme obtenu par agglomerative clustering

ID	Context data
ext1SS	status device hasStatus isStatusOf Open Lampe1 1970-01-01T09:39:00 5105 100
ext1SD	status door hasStatus isStatusOf Open Door2 1970-01-01T09:39:03 0215 100
ext2SD	status door hasStatus isStatusOf Open Door1 1970-01-01T09:39:08 0216 100
ext2SS	status device hasStatus isStatusOf Open Lampe2 1970-01-01T09:39:09 5106 100
ext3SD	status door hasStatus isStatusOf Close Door1 1970-01-01T09:39:14 0216 100
ext1ST	status tap hasStatus isStatusOf Open TapeColdWaterWashbasin 1970-01-01T09:39:43 F101 100
...	...

Tableau 7.III – Échantillon des données de contexte

clusters	cluster samples	situation	service
1,2,3,4	lamp bedroom open door bedroom open door bathroom open lamp bathroom open	wake up and going to bathroom	lamp bedroom open lamp bathroom open
6,8,10,12 7,9,11	open tape col water washbasin bathroom close tape col water washbasin bathroom	washing	no service here
13,15,17 14,16	close tape col water washbasin bathroom open tape col water washbasin bathroom	washing	no service here
18,19	open the door bathroom and put the kitchen lamp on	finish washing and go to kitchen	lamp kitchen on
...

Tableau 7.IV – Les clusters formés par l’algorithme

3 Modélisation des situations à l’aide d’une ontologie

La situation dans cette approche correspond à une activité de la personne selon le même scénario que celui présenté dans la section précédente. Nous nous sommes intéressés particulièrement à l’activité préparation du thé. L’idée recherchée derrière la détection de cette activité est de savoir si la personne arrive à suivre les instructions présentées sur l’interface tactile pour accomplir cette tâche.

Nous avons modélisé cette activité à l’aide de notre ontologie sous le logiciel Protégé. Il existe cependant plusieurs abstractions de l’activité selon le niveau de détails qu’on cherche à modéliser. La Figure 7.8 montre un exemple abstraction qui reflète les données de contexte utilisées dans l’expérimentation. La difficulté de la découverte sur ce type de situations est due à la richesse de la description de la situation décrite dans l’ontologie avec des concepts et des relations non pris en compte par la structure des données de contexte.

4 Autres travaux d’analyse

Cette section résume quelques travaux d’analyse que nous avons réalisés parmi d’autres durant l’étude sur les mesures de similarité ainsi que sur la formation des clusters potentiels.

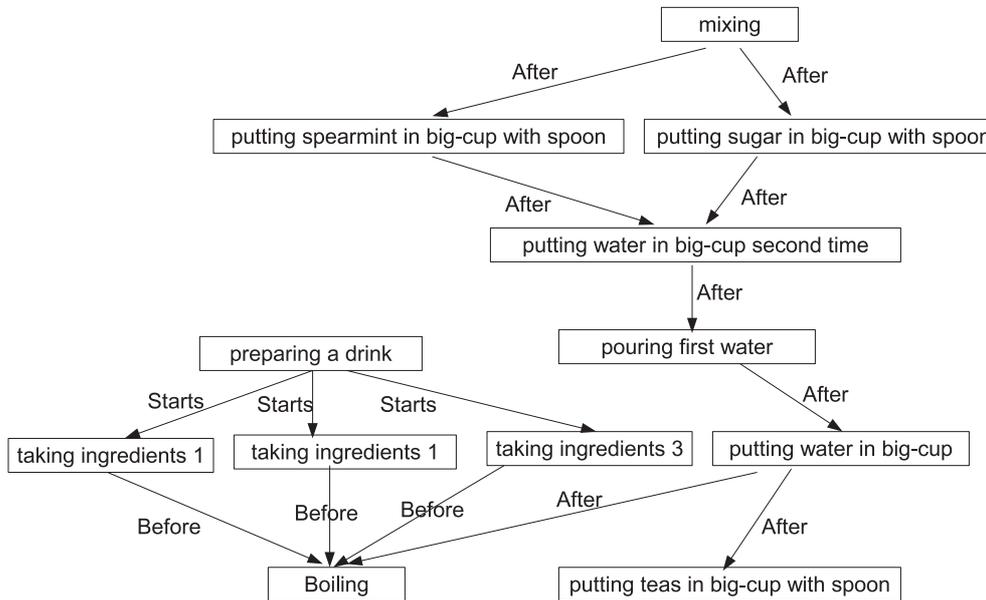


Figure 7.8 – Exemple de modélisation d’une situation

4.1 Partitionner un large graphe

Nous avons dans un premier temps considéré les données recueillies de la même expérimentation décrite à la section 2.3 en se limitant aux données recueillies sur une même personne exécutant le scénario selon un format propre à nos expérimentations (cf. 7.V). Ensuite, ces données ont été structurées automatiquement pour les rendre sous le format d’un fichier en entrée à l’algorithme MDG (cf. chapitre 5). Ce fichier représente un large graphe construit sur la base d’une mesure de similarité combinée des trois mesures (conceptuelle, syntaxique et temporelle) avec des poids (α, β, γ) de 1.

La figure 7.9 montre un exemple de résultat avec cet algorithme. Les cercles en gris portant les noms $d1$ à $d7$ représentent des données de contexte. Chaque rectangle correspond à un regroupement de données selon la localisation (*Bedroom*, *Bathroom*, *Kitchen*). Les flèches reliant ces rectangles indiquent les mesures de similarité entre les données. Par exemple, si $d3$ indique "la lampe du salon allumée"

```

*Node data "timestamp" "source" "type" "location" "status"
"8 :59 :21" "5105" "Lampe" "Bedroom" "Open"
"8 :59 :24" "0215" "Door" "Bedroom" "Open"
"8 :59 :30" "0216" "Door" "Bathroom" "Open"
"8 :59 :32" "5106" "Lampe" "Bathroom" "Open"
"8 :59 :37" "0216" "Door" "Bathroom" "Close"
"8 :59 :50" "F101" "TapeColdWaterWashbasin" "Bathroom" "Open"
"8 :59 :52" "F101" "TapeColdWaterWashbasin" "Bathroom" "Close"
"9 :0 :8" "F101" "TapeColdWaterWashbasin" "Bathroom" "Open"
"9 :0 :15" "F101" "TapeColdWaterWashbasin" "Bathroom" "Close"
"9 :0 :19" "F101" "TapeColdWaterWashbasin" "Bathroom" "Open"
"9 :0 :25" "F101" "TapeColdWaterWashbasin" "Bathroom" "Close"
"9 :1 :29" "0216" "Door" "Bathroom" "Open"
"9 :1 :34" "5101" "Lampe" "Kitchen" "Open"
"9 :1 :34" "IR02" "Infrared" "KitchenOven" "Close"
"9 :1 :37" "IR01" "Infrared" "KitchenSink" "Close"
"9 :1 :38" "IR02" "Infrared" "KitchenOven" "Open"
"9 :1 :38" "0113" "LockerC6" "Kitchen" "Open"
"9 :1 :40" "0113" "LockerC6" "Kitchen" "Close"
"9 :1 :40" "IR01" "Infrared" "KitchenSink" "Open"
"9 :1 :42" "IR01" "Infrared" "KitchenSink" "Close"
"9 :1 :42" "0115" "LockerA4" "Kitchen" "Open"
"9 :1 :46" "IR01" "Infrared" "KitchenSink" "Open"
"9 :1 :47" "0115" "LockerA4" "Kitchen" "Close"
"9 :1 :48" "IR01" "Infrared" "KitchenSink" "Close"
"9 :1 :51" "F103" "TapeHotWaterSink" "Kitchen" "Open"
"9 :1 :51" "IR01" "Infrared" "KitchenSink" "Open"
"9 :1 :57" "F103" "TapeHotWaterSink" "Kitchen" "Close"
"9 :1 :58" "IR01" "Infrared" "KitchenSink" "Close"
"9 :2 :1" "IR01" "Infrared" "KitchenSink" "Open"
"9 :2 :2" "IR01" "Infrared" "KitchenSink" "Close"
"9 :2 :6" "IR01" "Infrared" "KitchenSink" "Open"
"9 :2 :8" "0210" "LockerB1" "Kitchen" "Open"
"9 :2 :9" "IR01" "Infrared" "KitchenSink" "Close"

```

Tableau 7.V – Échantillon de données

et $d6$ indique "la lampe de cuisine allumée" alors ces deux données vont être proches au sens des mesures de similarité (tous les deux indiquent l'état d'une lampe) mais à cause de leurs localisations (salon, cuisine); ces deux données vont être placées dans des groupes différents.

La particularité de cette expérimentation est la formation de clusters par localisation où le caractère temporel des données a été ignoré. Nous sommes arrivés à la conclusion que l'idée de partitionner un large graphe de données de contexte peut être utile mais après une abstraction des données temporelles.

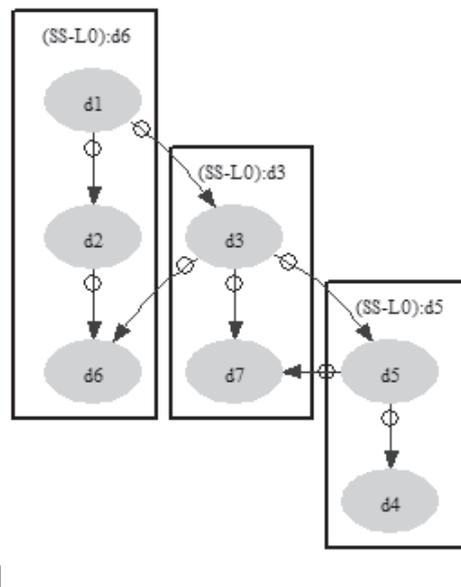


Figure 7.9 – Exemple de partitionnement d'un large graphe de données de contexte

4.2 Fouille visuelle des données de contexte

La fouille visuelle de données nous permet d'avoir un avis préalable sur les formations possibles des groupes. Il existe de nombreux logiciels de visualisa-

tion de données sous forme de graphe comme par exemple "Pajek" ⁵, Tulip ⁶ et Netdraw⁷.

Ces logiciels aident à analyser les données représentées par un graphe sous plusieurs angles par exemple visualiser les relations d'un même noeud, les sous graphes isolés, les chemins, les circuits, etc.

Nous montrons dans la figure 7.10 un exemple de visualisation de données de contexte avec NetDraw. Nos données de contexte sont introduites d'un fichier à plusieurs lignes où chaque ligne représente une donnée de contexte à la manière de la figure 7.V. Puisque, dans nos échantillon de données, une source de contexte ne peut envoyer qu'un seul type de données de contexte, *entity* et *contextValue* seront les mêmes par source de données. C'est pour cela qu'elle n'apparaissent pas dans la liste de ces données afin de nous permettre d'analyser la construction des clusters sur la base des similarités entre les identifiants de ces concepts. Ce logiciel permet plusieurs possibilités d'analyse du graphe. Par exemple, le nombre "K" d'interconnexions d'un sommet indique l'importance de ce noeud dans les données de contexte. Si ce noeud représente une localisation, nous pouvons déduire la fréquence de passage de la personne à cet endroit. Les points de coupure dans le graphe permettent quant à eux de séparer le graphe en plusieurs sous graphes connectés qui représenteront probablement des situations. La figure 7.10 montre un exemple d'analyse sur les données du scénario du laboratoire DOMUS. Chaque noeud représente une donnée de contexte et son timestamp. Dans ce cas de figure, nous avons étudié les relations temporelles entre les données par type d'activité ; par exemple l'activité dans le salon a été plus importante que celle dans la cuisine. Ceci montre l'importance du regroupement sur la base de la localisation. Nous avons aussi fait d'autres analyses des données de PlaceLab comme celles fournies par un capteur d'humidité (cf. figure 7.11 du fichier 1WireHumidity).

Les travaux que nous avons décrits dans cette section, nous ont permis en

⁵<http://vlado.fmf.uni-lj.si/pub/networks/pajek>

⁶<http://tulip.labri.fr>

⁷<http://www.analytictech.com>

outre de s'assurer qu'une mesure combinée est plus intéressante pour nos données de contexte qu'une mesure simple. La leçon tirée de cette expérience recommande d'associer la localisation à la temporalité si on veut être plus précis sur les situations qui se produisent au même instant dans l'environnement. En fait, la localisation se montre comme la spécificité la plus marquante dans les données de contexte de DOMUS en plus de la temporalité. Ce qui est dit partout dans la littérature, et relève également du bon sens.

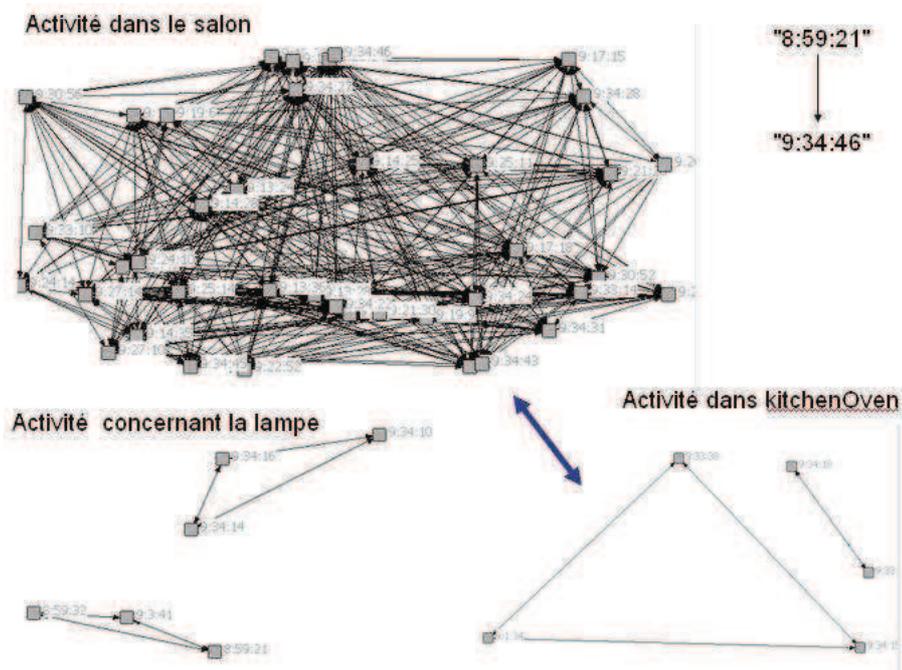


Figure 7.10 – Exemple de visualisation de données de contexte avec NetDraw

```

Node Data
timestamp type device valeur ip
09:00:06:093 2 120000004C002126 16.87749512483423 54
09:00:09:593 2 040000004C095D26 14.55548760372063 68
09:00:12:625 2 1E0000003F607F26 12.06580303329285 51
09:00:20:000 2 9E0000003F4A2A26 13.50901692632956 53
09:00:22:156 2 5B0000004C096B26 17.53441958483936 55
09:00:24:046 2 5D0000004C095E26 16.33196033881575 58
09:00:34:500 2 230000004C001526 14.55947988361151 61
09:00:42:515 2 980000003F4A1F26 15.98048063874044 52
09:00:43:468 2 AF0000003F4A1E26 14.47910705459863 59
09:00:51:234 2 0D0000003F607126 17.04203617407417 69
09:01:00:953 2 040000004C095D26 14.55646746086851 68
09:01:01:015 2 120000004C002126 16.87296477167182 54
09:01:02:796 2 1E0000003F607F26 11.99264929718915 51

```

Figure 7.11 – Un échantillon de données WireHumidity de PlaceLab

4.3 Regroupement par transformation des données de contexte

Une autre analyse a été appliquée sur les données du scénario de préparation du thé. Nous avons exploré l'algorithme EM (Expectation-Minimization) après reformatage des données de contexte en données numériques. Nous avons utilisé le logiciel Matlab ⁸ ainsi que le logiciel Weka ⁹ pour conduire ces tests. La figure 7.12 montre un exemple de résultats avec le logiciel Weka. Les clusters sont affichés en fonction de la localisation (à droite en vertical). Nous avons remarqué que dans plusieurs cas, le regroupement se fait selon la localisation et le temps. En fait, le temps permet de regrouper les données de proximité, ce qui indique indirectement une activité ou une situation à un emplacement particulier.

⁸<http://www.mathworks.fr/>

⁹<http://www.cs.waikato.ac.nz/ml/weka/index.html>



Figure 7.12 – Résultat de clustering avec EM

5 Conclusion

Ce travail expérimental et d'analyse a permis de déboucher sur une solution de découverte de situations pour l'adaptation automatique de services. La découverte de situations fréquentes a été choisie et intégrée à notre implémentation d'architecture. Par contre, la modélisation de situations à l'aide d'une ontologie mérite d'être approfondie. Nous avons évoqué cependant une nouveauté quant à l'utilisation d'une contrainte temporelle dans un algorithme de clustering. Ce problème nécessite une étude expérimentale spécifique afin d'être introduit comme une nouvelle approche dans les algorithmes de clustering avec contraintes. Nous soulignons en dernier que le travail de visualisation aide beaucoup dans la compréhension de la formation des clusters. Ceci dit, nous n'oublions pas de dire que malgré notre satisfaction des résultats obtenus, une expérimentation à grande

échelle sur un cas réel persiste comme un besoin au terme de ces travaux. Orange-Labs conduit avec des partenaires des expérimentations dans des appartements instrumentés avec de nouveaux capteurs non intrusifs. Notre solution pourra être reprise pour une mise à l'échelle. Dans le chapitre suivant, nous donnons plus de détails sur le travail d'intégration de notre solution et son utilisation dans le cadre du projet européen MIDAS.

CHAPITRE 8

IMPLEMENTATION

Nous présentons de ce chapitre une implémentation d'un démonstrateur que nous avons développé à Orange-Labs dans le cadre du projet européen MIDAS. Nous voulons montrer par ce démonstrateur que notre proposition d'une approche de découverte de contexte pour une adaptation de services est réalisable avec notre architecture et donne de bons résultats. Dans la section 1, nous décrivons le cadre de cette implémentation et en section 2 nous donnons des détails de réalisation du démonstrateur.

1 Aperçu général du cadre d'implémentation

Nous avons conduit ce travail dans le cadre du projet européen MIDAS (ITEA 2/07008) ¹ et d'un stage d'ingénieur que nous avons proposé à Orange-Labs ². L'objectif principal de ce projet est de développer des solutions complètes et intelligentes, faisant usage de différents types de capteurs, d'actionneurs et de technologies de communication. L'objectif est de fournir un accompagnement personnalisé à toutes les personnes ayant besoin d'assistance, en fonction de leur propre spécificité (âge, handicap, etc) et d'une manière non intrusive et respectueuse. Le projet vise particulièrement la conception d'interfaces adaptatives et conviviales à la fois indoor et outdoor destinées à vaincre le scepticisme et l'inquiétude naturelle du vieillissement des individus.

Parmi plusieurs scénarios sélectionnés dans le cadre de ce projet, notre travail intervient dans le développement du scénario : *Détermination du niveau d'activité*. Il a pour but d'évaluer le niveau d'autonomie de la personne, d'identifier les situations habituelles et de planifier et déclencher des actions si nécessaire.

¹<http://www.midas-project.com/>

²Stage de Jérémy Garden, INSA de ROUEN, octobre 2009

L'environnement auquel est destiné ce genre d'application comprend une maison équipée d'un ensemble de capteurs et actionneurs et des services comme la notification, la communication avec la famille, la communication avec les voisins ou les équipes professionnelles de santé et un service d'information sur la vie et la santé du patient.

Ce scénario se déroule dans un appartement mettant en scène une dame appelée Jane de 80 ans qui passe une période de la journée à faire des allers-retours entre sa chambre et la cuisine. Plus précisément, au début du scénario, Jane dort dans son lit, puis elle se réveille, va à la cuisine, elle se sert de son four pour cuire ou réchauffer de la nourriture, elle mange et puis retourne à sa chambre pour prendre une courte sieste. Puis elle retourne dans la cuisine pour boire un verre d'eau et retourne à nouveau dans son lit pour reprendre son court repos.

L'appartement est équipé d'un capteur qui permet de suivre l'état du four, c'est à dire si le four est allumé ou éteint, et avec des lumières qui émettent des signaux lorsqu'ils sont sous tension et hors tension. Jane tient son téléphone portable avec elle. Le téléphone mobile intègre un logiciel qui est capable de détecter son emplacement, à savoir si elle est dans sa chambre ou dans sa cuisine. Il intègre également un logiciel qui est capable de détecter la posture de Jane, à savoir si elle est couchée, debout, assise ou en train de marcher.

Maintenant, en observant le comportement de Jane sur une longue période de temps, disons plus d'une semaine, un humain aurait sans doute remarqué que la plupart du temps, si ce n'est à chaque fois, quand Jane se réveille elle allume la lumière, et que la plupart du temps lorsque Jane quitte sa chambre, elle éteint la lumière. Malgré sa simplicité, ce scénario présente plusieurs défis liés aux choix des capteurs permettant sa réalisation et à la découverte des habitudes d'utilisation des appareils par les personnes à partir de ces données.

Notre objectif est d'effectuer une analyse similaire, en appliquant des techniques de fouille de données sur un corpus de données capteurs corrélé avec le comportement de Jane et collecté au cours de la même période de temps. Plus

précisément, nous avons réalisé notre architecture (cf. chapitre 6) et nous avons intégré les capteurs utilisés dans l'environnement de Jane pour répondre à ce besoin.

2 Mise en oeuvre

Dette cette réalisation (cf. 8.1), le CMS-Amigo implémente les contextes sources et les contextes clients comme des services Web à l'aide de la pile DPWS développée par le projet SOA4D³. Selon notre modélisation, les données de capteurs sont collectées automatiquement, stockées dans une base de données (MYSQL-JENA) puis les situations fréquentes de Jane sont découvertes automatiquement. Enfin, les situations fréquentes sont utilisées automatiquement pour personnaliser automatiquement des services de contrôle déployés dans l'environnement.

Les termes (CS, GW, SOAP/RDF, HAB, SOAP/XML, CC) sur la figure désignent respectivement les mots anglais (Context Source, Gateway, SOAP Protocol + RDF, Home Automation Box, SOAP Protocol + XML, Context Consumer).

Dans ce qui suit, nous présentons d'abord l'ontologie que nous avons utilisée puis les modules formant cette architecture.

2.1 Description de l'ontologie

Notre ontologie est développée sous Protégé⁴. Elle décrit de manière générique les données provenant de capteurs ou de Service Web et les services à l'utilisateur qui sont embarqués dans l'espace intelligent. Par exemple, un service web distant peut être une base de données médicale, une application distante peut être le répondeur du bureau de travail.

Dans notre implémentation, l'espace intelligent correspond à l'habitat utilisateur et les sources sont quatre capteurs déployés dans cet habitat : capteur de posture, capteur de localisation, capteur de position et un capteur dit électrique

³<http://www.soa4d.org/>

⁴<http://protege.stanford.edu>

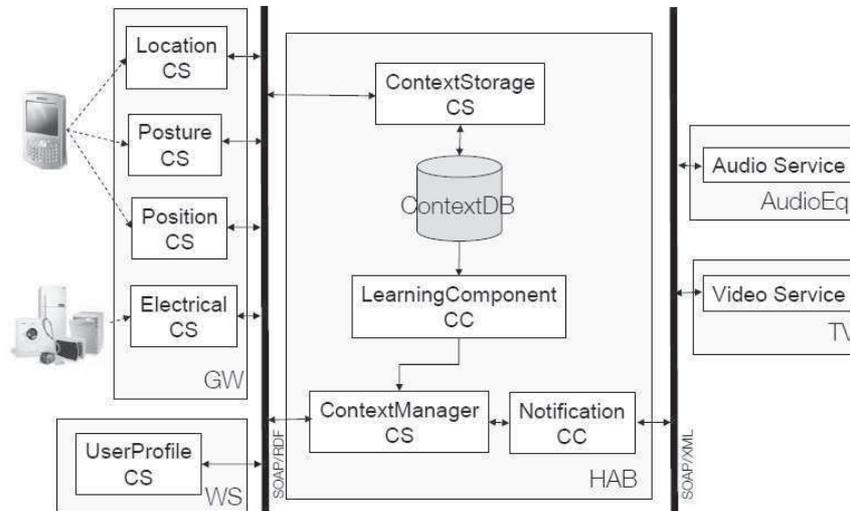


Figure 8.1 – Architecture de gestion du contexte dans le projet MIDAS

qui rend un ou plusieurs états des appareils électriques dans l’habitat. Les services à l’utilisateur sont des services de contrôle de dispositifs, de notification ou d’alerte comme l’allumage du chauffage, l’extinction de l’éclairage...

Les figures 8.2, 8.3, 8.4 montrent quelques parties de notre ontologie. La figure 8.2 illustre une partie de notre ontologie en indiquant à la fois les classes, les propriétés d’objets et les propriétés de données. La structure hiérarchique de la figure 8.3 décrit les liens entre les concepts relatifs à une donnée de contexte décrivant le statut d’un dispositif et les concepts parents : Entity, ContextValue et ContextData. La figure 8.4 montre comment cette même donnée est instanciée à partir de l’ontologie pour donner naissance à une donnée de contexte.

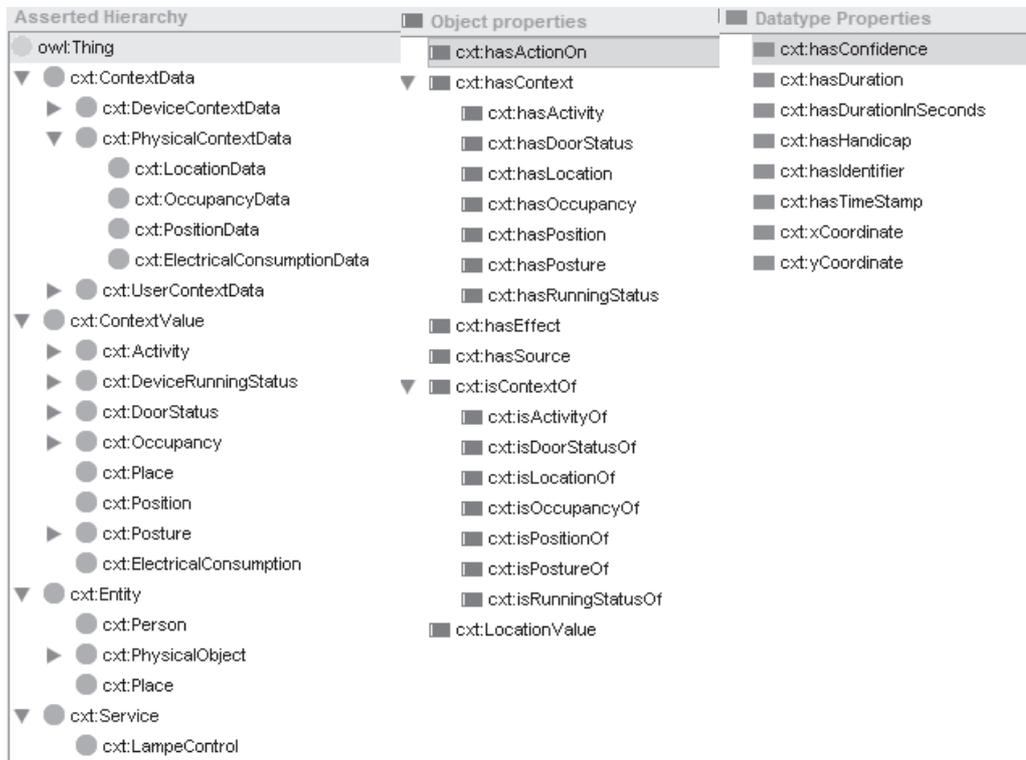


Figure 8.2 – Partie de la structure de l'ontologie

Les services dans notre ontologie sont régis par deux propriétés fondamentales : l'action et l'effet. L'action définit le type de requête et l'entité concernée par ce service. L'effet détermine l'état dans lequel sera le composant contrôlé par ce service. Par exemple l'état de la lampe change de "On" (allumée) à "Off" (éteinte). Nous illustrons ce principe par la figure 8.5 et un exemple basique : "le contrôle d'une lampe".

Nous mentionnons que le but de cette modélisation de service n'est pas de définir ou de donner des descriptions fonctionnelles aux services déployés dans l'environnement à la manière d'OWL-S ou d'autres formalismes. Nous nous sommes contentés dans cette implémentation d'une description simplifiée qui permet d'identifier les services par leurs effets sur l'environnement ; pour utiliser le service lui

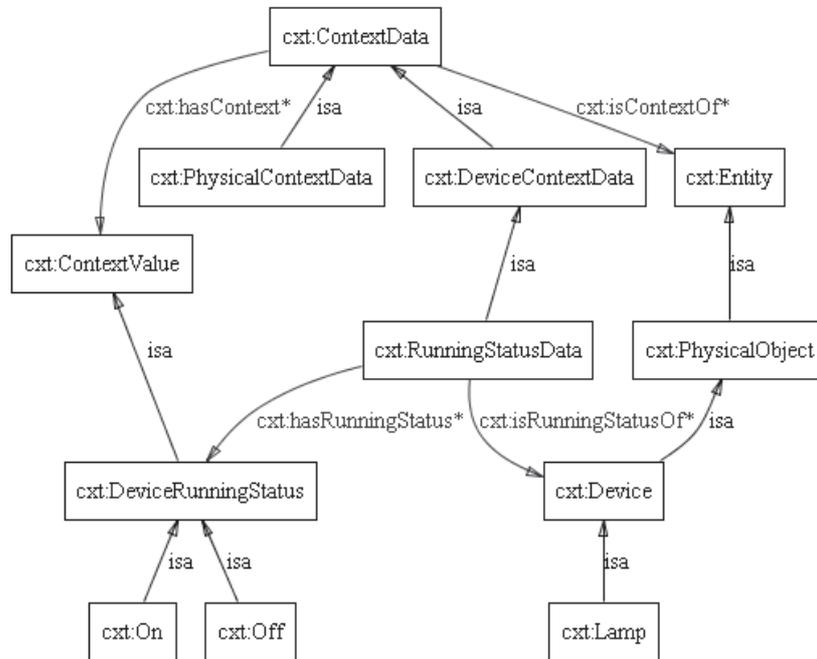


Figure 8.3 – Exemple d’instances de concepts

même cela imposera une représentation fonctionnelle plus descriptive du service selon l’architecture déployée. La description fonctionnelle dont on parle est utile dans le cas d’un processus de découverte (au sens communément admis dans les architectures orientées services) ou dans la composition de ces services. Ces aspects sortent du cadre de cette thèse.

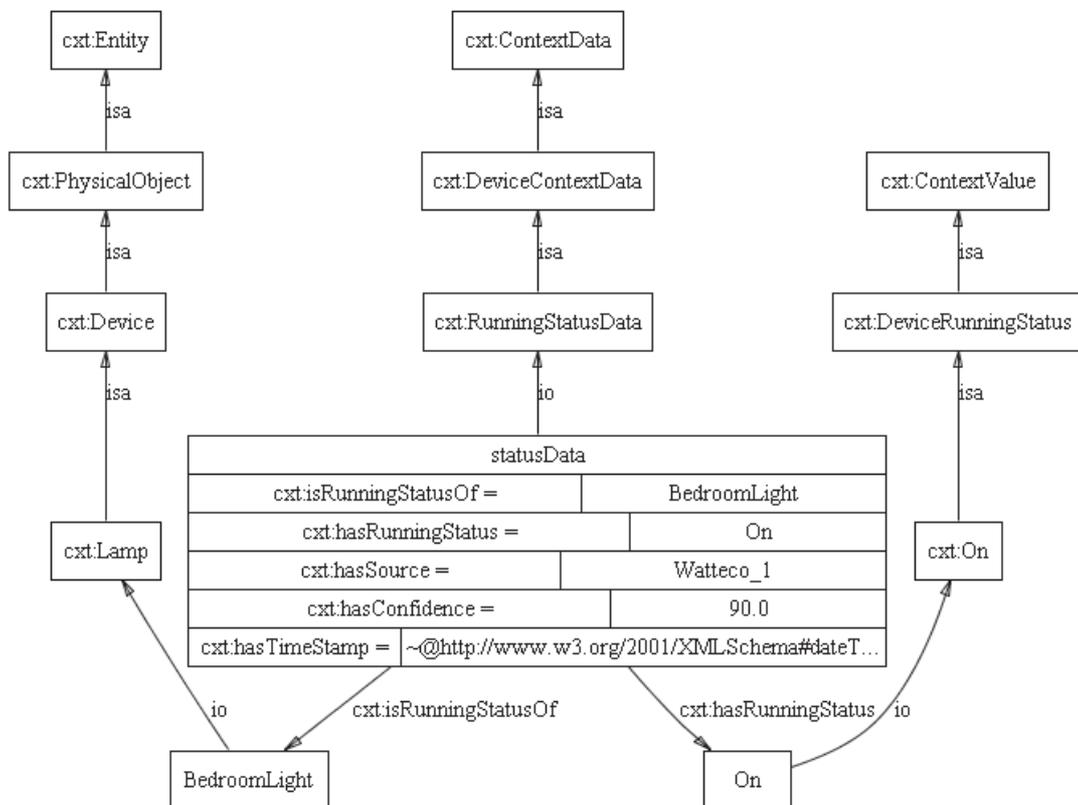


Figure 8.4 – Exemple d’une instance d’une donnée de contexte

2.2 Acquisition

Les données utilisées sont collectées à partir de capteurs réels selon le principe suivant :

- Localisation : un téléphone mobile intègre un logiciel qui est capable de détecter l’emplacement de la personne dans la maison (cuisine, chambre, salon, etc.). La technologie radio utilisée par ce logiciel est le WiFi.
- Posture : le même téléphone mobile est capable de déterminer la posture (assise, couchée, debout, marcher) de la personne en employant des accélé-

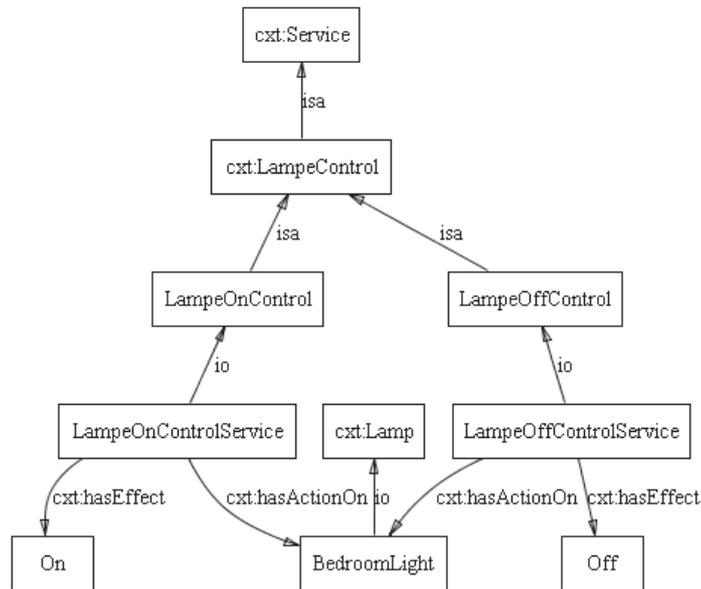


Figure 8.5 – Exemple d’un service de contrôle de lampe

romètres.

- Position : désigne les valeurs (X,Y,Z) envoyées par une caméra de la position d’une personne dans l’espace.
- Statut d’un appareil ou d’un dispositif électrique : un capteur dit Watteco envoie la valeur d’état de l’appareil (On,Off). Ce capteur se place sur une ligne électrique pour détecter *la signature électrique* d’un appareil et déterminer son état.

Les données recueillies par ces capteurs sont sous format XML (cf. figure 8.6). Elles sont modélisées au niveau des sources de contexte selon notre modèle sémantique d’une donnée de contexte (cf. figure 8.7).



Figure 8.6 – Acquisition des événements capteurs et dispositifs

2.3 Stockage des données de contexte

Un consommateur de contexte (*context Consumer* selon CMS-Amigo) appelé ContextStorageCS s'abonne auprès de chaque capteur au démarrage du système. Cet abonnement signifie qu'à chaque réception d'une nouvelle donnée capteur par les contextes sources locationCS, PostureCS, PositionCS et ElectricalCS, cette nouvelle donnée est envoyée automatiquement via SOAP/RDF et selon le format d'une donnée de contexte au ContextStorageCS. Ensuite, ContextStorageCS stocke à la volée ces données dans une base de données MYSQL en utilisant la librairie Jena. La figure 8.8 montre le modèle logique de stockage de Jena dans la base de données MYSQL. En fait, Jena crée plusieurs tables dont chacune a un rôle bien particulier comme le montre la figure 8.8. La table "statements" comprend les triplets RDF. Deux tables sont utilisées pour les littéraux et URI longs. D'autres tables plus spécifiques à certaines fonctionnalités sont utilisées par JENA mais ne sont pas représentées sur la figure.

2.4 Description des données collectées

Une partie des événements capteurs recueillis sont détaillés dans le tableau de la figure 8.9. Ce format simplifié représente les données en forme de tableau selon

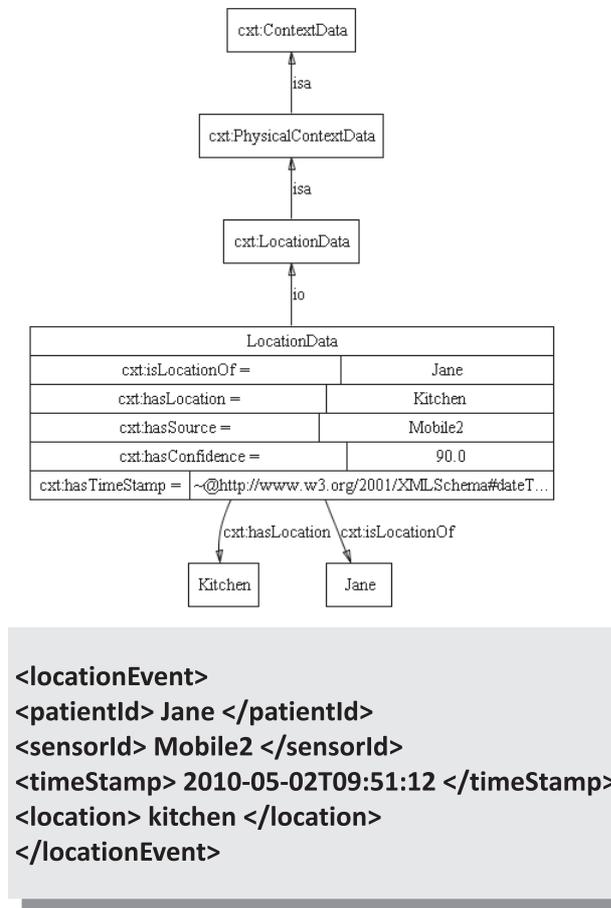


Figure 8.7 – Représentation interne d'un événement capteur

les valeurs que prend un capteur et non selon leur modèle sémantique (données de contexte). Cette présentation est plus facile à comprendre.

Le déroulement des événements se fait dans l'ordre de l'index du tableau. Potentiellement, nous avons 64 configurations possibles selon les valeurs que peuvent prendre les capteurs.

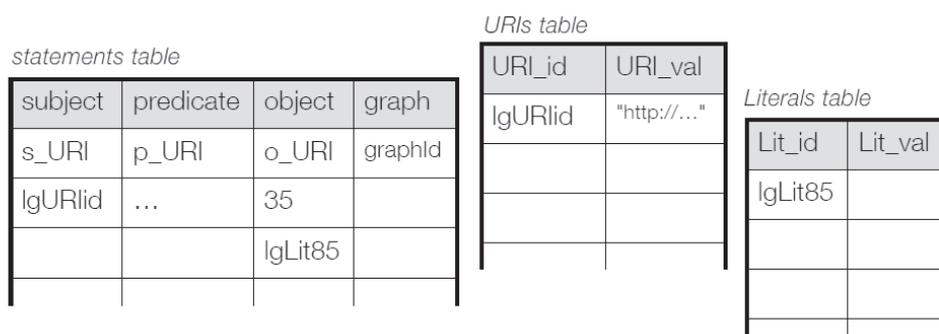


Figure 8.8 – Modèle logique de stockage JENA dans une Base de données MYSQL

2.5 Découverte de situations fréquentes

D'après l'algorithme de découverte du chapitre 5 de la section 3.1.1, nous commençons par appliquer un tri temporel sur les données (ordre donné par l'index du tableau), puis nous utilisons la contrainte de cohérence pour segmenter les données de contexte en contextes purs. Nous trouvons 27 contextes purs dans les données collectées. Ces contextes purs commencent à partir de la ligne 5 jusqu'à la ligne 31 du tableau de la figure 8.10. L'algorithme de clustering MCL prend en entrée ces contextes purs et les similarités entre eux pour en déduire un clustering. Le résultat donné par cet algorithme est donné dans la figure 8.11. Par exemple, le cluster 0 contient les 4 contextes purs 2, 12, 16, 24. Si nous vérifions leurs représentations synthétiques dans le tableau 8.10, nous remarquons qu'ils sont identiques. Ce résultat met en évidence les lieux de Jane au cours des quatre contextes 2, 12, 16, 24. On peut remarquer que ce groupe peut être décrit comme

	Oven		Kitchen Light		Bedroom Light		Location		Posture			
	ON	OFF	ON	OFF	ON	OFF	Kitchen	Bedroom	Running	Standing	Seating	Lying
1		X										
2				X								
3						X						
4							X					
5												X
6					X							
7											X	
8										X		
9						X						
10							X					
11			X									
12	X											
13		X										
14				X								
15							X					
16					X							
17											X	
18						X						
19												X
20					X							
21											X	
22										X		
23						X						
24							X					
25			X									
26				X								
27							X					
28					X							
29											X	
30						X						
31												X

Figure 8.9 – Données expérimentales correspondant au scénario

suit : "La personne est assise sur son lit, tandis que la lumière est allumée". On note cette situation fréquente C0. Elle sera représentée par l'un des contextes purs 2, 12, 16 ou 24. Le cluster centroïde est mieux approprié pour ce choix car c'est lui le plus proche à tous les membres du cluster au sens de la mesure de similarité. Avec une analyse similaire pour tous les clusters trouvés nous établissons les situations fréquentes suivantes :

- C0 : la personne est assise sur son lit, la lumière de la chambre est allumée et le four est éteint.
- C1 : la personne est debout dans sa chambre, la lumière est allumée et le four est éteint.
- C2 : la personne est debout dans sa chambre, la lumière est éteinte et le four

est éteint.

- C3 : la personne est debout dans la cuisine, la lumière de la cuisine est éteinte et le four est éteint.
- C4 : la personne est debout dans la cuisine, la lumière de la cuisine est allumée, et le four est en marche.
- C5 : la personne est dans son lit, la lumière de la chambre est éteinte, et le four est éteint.
- C6 : la personne est allongée sur son lit, la lumière de la chambre est allumée et le four est éteint.
- C7 : la personne est assise sur son lit, la lumière de la chambre est éteinte et le four est éteint.

Oven		kitchen light		Bedroom light		Location		Posture				situations
on	off	on	off	on	off	kitchen	bedroom	running	standing	seating	lying	
												0
												1
												2
												3
												4
												5
												6
												7
												8
												9
												10
												11
												12
												13
												14
												15
												16
												17
												18
												19
												20
												21
												22
												23
												24
												25
												26

Figure 8.10 – Situations construites par l’algorithme de découverte

On pourra remarquer que la fréquence n’est pas la même pour C0 (4) que pour C7 (2). Cela nous permettra de ne retenir que les situations qui ont une fréquence supérieure à certain seuil. Ce résultat correspond bien aux faits réels.

2.6 Identification d’une situation inhabituelle et adaptation

Nous avons testé notre proposition d’adaptation sur des exemples simples préparés manuellement sous Protégé et avec SPARQL selon le principe de la section 5 du chapitre 6. Il reste à développer ce module et l’intégrer dans le démonstrateur. Ce qui permettra de valider la partie adaptation de services.

Clusters	Situations
C0	2 12 16 24
C1	3 11 17 23
C2	4 10 18 22
C3	5 9 19 21
C4	6 7 8 20
C5	0 14 26
C6	1 15
C7	13 25

Figure 8.11 – Situations fréquentes trouvées par l’algorithme de découverte

3 Conclusion

Le travail d’implémentation décrit dans ce chapitre a été effectué sur un cas d’usage en vue d’arriver au développement d’un démonstrateur pour le projet MIDAS. Les capteurs que nous avons utilisés dans ce projet sont un capteur d’états (marche / arrêt) de machines électriques et un terminal mobile avec lequel on peut récupérer l’emplacement, la position et la posture de son propriétaire. Toutefois, la flexibilité du langage de représentation des données que nous avons adopté, permet de prendre en compte un large spectre de sources de contexte comme les services web ou des outils personnels (agenda, répertoire téléphonique,...).

L’implémentation montre qu’il est possible d’ajouter d’autres capteurs sans modifier le module de stockage ni de découverte en multipliant autant de sources de contexte que de capteurs dans l’environnement. L’ontologie est générique et peut être facilement étendue pour prendre en compte d’autres données de contexte. Ainsi, cette implémentation permettra de tester la robustesse de notre architecture

aux remplacements et aux pannes de capteurs dans l'environnement. La panne d'un capteur peut être due par exemple à l'épuisement de sa batterie. Dans ce cas, le capteur ne produit aucune donnée de contexte. Par contre, l'arrivée d'un nouveau capteur engendrera la prise en compte de ses données par le biais d'une source de contexte associée à celui-ci. De ce fait, cette nouvelle donnée peut influencer la construction des situations fréquentes. Ainsi, cette implémentation se voit capable de supporter plusieurs configurations de l'environnement.

Cependant, lors d'un passage à l'échelle avec quelques dizaines ou centaines de capteurs, le temps de réponse du système sera trop grand puisque nous avons montré que le temps d'exécution de l'algorithme de découverte de situations fréquentes est principalement occupé par le tri temporel et le calcul des similarités (cf. section 3.3 du chapitre 7) et que ce temps augmente en fonction du nombre de données de contexte. Cette difficulté pourra être maîtrisée avec une optimisation du code et un choix des périodes à lesquelles le système relance la procédure de découverte.

Du point de vue de la découverte, nous pouvons affirmer que notre approche est capable d'identifier des groupes de situations similaires qui se produisent fréquemment. Bien que nous n'ayons pas poussé la mise en oeuvre de notre approche, nous avons pu constater que certains de ces groupes correspondent à des contextes qui sont pertinents pour le contrôle de l'environnement. Par exemple, le groupe 1 et le groupe 2 correspondent aux cas où la personne quitte sa chambre, et leur description suggère que la lumière de la chambre à coucher peut être éteinte (c'est la seule différence entre la description synthétique des deux groupes).

Au-delà de la domotique, nous croyons que notre approche est applicable à des domaines où des clusters basés sur la similarité devraient être trouvés à partir de structures de données hétérogènes et distribuées. Ainsi les domaines d'application dédiés à la recherche de contenu ou encore SmartCity et SmartBuilding pourraient être des cibles potentielles.

Quatrième partie

Conclusion

CONCLUSION

Les services offerts au sein des environnements intelligents tendent de plus en plus à s'adapter. On peut considérer que la prise en compte du contexte est la première étape vers le déploiement de services dotés de capacités relevant de la cognition artificielle au sens fort, c'est-à-dire des services qui permettent de choisir, en situation, les meilleures actions pour atteindre des buts généraux. Ils s'adaptent aux utilisateurs ou doivent s'auto-configurer pour optimiser leur fonctionnement (empreinte écologique, consommation énergétique). Cette capacité d'adaptation nécessite que ces services soient conscients à la fois de leur contexte d'exécution et de leur contexte d'utilisation.

L'adaptation au contexte exige également de s'intéresser au "terrain" dans lesquels ces services sont ancrés. En effet, l'analyse des situations ou activités (professionnelles, ludiques, domestiques, etc.) pour lesquelles cette adaptation est recherchée permet de déterminer les contextes pertinents qu'il faut prendre en compte dans la conception et l'évaluation de ces services adaptatifs. Aujourd'hui, concevoir des services sensibles au contexte exige donc, d'une part, beaucoup d'efforts de la part des concepteurs, et, d'autre part, un paramétrage délicat des services et de leur comportement. La mise en place de techniques d'apprentissage du périmètre du contexte pertinent à prendre en compte pour de tels services apparaît donc comme un enjeu de recherche critique dans cette problématique afin de diminuer ces différents efforts de conception et de configuration.

Cette réflexion a conduit les recherches décrites dans ce document et nous a amené à proposer un ensemble de techniques permettant aux services de découvrir par eux même les contextes ou les situations auxquelles ils puissent/doivent s'adapter.

Ainsi, nous avons proposé, expérimenté et implémenté dans notre travail une approche de découverte de contexte pour une adaptation automatique de services. Les services que nous avons considérés sont les services à l'utilisateur c'est-à-dire

des fonctionnalités qui facilitent ou aident l'utilisateur dans l'accomplissement de ses tâches. La démarche scientifique qui nous a permis d'aboutir à ce résultat comporte plusieurs étapes : une étude de l'état de l'art sur l'adaptation des services au contexte pour mettre en évidence les défis et les limites de l'adaptation des services au contexte ; une proposition pour résoudre ces problèmes ; une évaluation de la proposition via des expérimentations et une validation des résultats via la réalisation d'un démonstrateur. Nous reprenons l'essentiel de ces étapes dans ce qui suit.

4 Contribution

L'étude de la littérature a montré une ambiguïté quant à l'utilisation de la notion d'adaptation en intelligence ambiante. Elle est souvent assimilée à la personnalisation, à la recommandation et à la reconfiguration. L'adaptation automatique se distingue de celles-ci quand il est nécessaire de considérer l'expérience passée dans le processus d'adaptation. Nous avons ainsi défini l'adaptation, dans le cadre de nos travaux, comme étant la capacité à faire bénéficier le service de l'expérience de son utilisation en tenant compte des situations de l'utilisateur et de son environnement.

Selon cette même étude, nous avons constaté que la prise en compte du contexte consistait en la réaction positive des systèmes à certaines situations ou de modèles pré-définis après avoir été reconnus ou identifiés à partir de dispositifs de perception de l'environnement. Cette pratique soulève de nombreux problèmes de déploiement des services. Ceci est lié à plusieurs facteurs tels que la dynamique de l'environnement, le changement des habitudes d'usage des services et l'hétérogénéité des sources de perception de l'environnement. Nous avons de plus remarqué que l'exploitation de l'historique du contexte est quasiment absente de nombreux travaux et, que dans un nombre restreint de cas, elle se limite aux profils utilisateurs et transactions passées. Notre objectif est donc de faire bénéficier le service de l'expérience de son utilisation au travers d'une analyse de l'histo-

rique du contexte pour déterminer les situations qui représenteront les contextes auxquels adapter les services. En stockant les données de contexte au fil du temps, une fouille permettra de découvrir des situations qui contribueront à l'aide d'un mécanisme d'adaptation à des services adaptatifs.

Le contexte n'est donc plus un modèle préétabli au moment de la conception du système mais une description de situations qui peuvent être découvertes dans les données de contexte et qui peuvent changer au fur et à mesure du cycle de vie du système. Ceci nous a conduit à définir le contexte comme étant une description d'une situation que l'on peut déterminer automatiquement à partir de données dites de contexte et à partir desquelles il est possible d'identifier des situations pertinentes pour l'adaptation de ces services.

Dans la partie proposition, nous avons commencé par définir un modèle sémantique unique pour exprimer les données de contexte. Ce choix est motivé par la volonté de cacher l'hétérogénéité de données en provenance de sources multiples et variées, et de pouvoir échanger ces données facilement entre applications.

En conséquence, nous avons développé une ontologie de contexte qui définit les concepts manipulés par les sources de contexte et leurs relations. Cette ontologie sert d'espace métrique pour le calcul de la mesure de similarité conceptuelle entre les données de contexte. Cette mesure complète d'autres mesures : temporelle, syntaxique, relationnelle et statistique, que nous avons adaptées aux caractéristiques et représentations des données de contexte et des contextes eux-mêmes. En effet, les structures du modèle sémantique d'une donnée de contexte et du modèle du contexte permettent le calcul de ces différentes mesures. Ces mesures sont le support aux algorithmes de regroupement de données de contexte et des contextes pour extraire des situations.

Nous avons appelé le processus d'extraction automatique de contextes à partir d'un large volume de données de contexte " une découverte de contexte ". Afin de répondre aux défis posés par ce processus, nous avons proposé deux approches : une approche ascendante et une approche descendante. Dans l'approche ascen-

dante, nous ne considérons aucune connaissance sur les situations à découvrir. Nous utilisons plusieurs algorithmes de fouille de données reposant sur nos mesures de similarité et comparons leurs résultats. Nous avons développé également une méthode pour la découverte des situations fréquentes. Dans l'approche descendante, nous avons exploré l'idée d'utiliser des ontologies comme moyen de description des situations à découvrir. La méthode décrite est fondée sur la corrélation entre la description de la situation dans l'ontologie et les regroupements construits à partir des données de contexte.

Nous nous sommes ensuite intéressés à l'intégration de ce module de découverte dans une architecture de gestion de contexte pour l'adaptation automatique de services. En effet, les architectures existantes ne permettent pas de telles fonctionnalités ni peuvent supporter l'hétérogénéité des sources du contexte et n'assurent en aucun cas l'adaptation automatique. Nous avons ainsi proposé une nouvelle architecture supportant la découverte du contexte et avons montré comme elle s'interface avec les modules d'acquisition et de stockage du contexte. Nous avons également proposé un mécanisme d'adaptation qui exploite le résultat de la découverte. Ce mécanisme présente à la fois un défi et une approche innovante pour déterminer automatiquement les services concernés par une situation découverte en étudiant les données qui forment cette même situation.

5 Bilan

Les résultats de ces travaux ont été encourageants et nous ont permis de dégager un ensemble de mécanismes et de représentations qui ont été implémentés dans le cadre de développement de notre démonstrateur pour le projet collaboratif européen MIDAS.

Nous avons présenté ci-dessus quelques éléments de solution pour la prise en compte du contexte pour l'adaptation automatique des services. Ces éléments sont le fruit de plusieurs concertations entre le laboratoire de recherche et celui de l'entreprise. C'est aussi le fruit de notre participation à différents projets

au sein d'Orange-Labs (CASPER, SMART-IMMO) et dans le projet collaboratif Européen MIDAS où ces résultats sont exploités et continuent à être transférés. En outre, deux co-encadrements de stages, un à l'École Nationale Supérieure des Mines de Saint-Etienne et un autre à Orange-labs Grenoble ont permis d'avancer sur ces travaux. Nous soulignons en dernier, la participation de la région Rhône-Alpes à ces travaux par le biais d'une offre de bourse de stage de recherche au laboratoire DOMUS de l'Université de Sherbrooke. Le séjour de recherche dans ce laboratoire a permis de travailler sur des données expérimentales acquises à partir d'un environnement semi-réel. De plus, il a permis de confronter nos solutions aux problèmes techniques relevés sur place.

Ces travaux présentent pour nous une grande satisfaction vis les enjeux industriels et sociétaux qu'ils présentent. Orange Labs développe depuis quelques temps des recherches et des applications sur l'habitat intelligent, notamment concernant le maintien à domicile des personnes âgées. Le but est de fournir aux personnes âgées un habitat qui les aide à conserver leur autonomie dans leur domicile. Parmi les services envisagés pour ce genre de problèmes on peut citer la télé-alarme sociale, la surveillance et le monitoring, les dispositifs de rappel, dispositifs de compensation, la réduction du stress.

6 Perspectives

A très court terme, notre objectif est de consolider les travaux effectués notamment sur la partie évaluation des algorithmes de découverte et la détermination de la pertinence des situations découvertes. À plus long terme, nous souhaitons mettre à disposition de l'utilisateur un espace intelligent où les services sont de plus en plus conscients des situations et des comportements qui se produisent dans l'environnement. Nous souhaitons également dépasser le cadre actuel de la découverte du contexte limité aux espaces intelligents pour l'étendre à d'autres environnements tel que le Web. Nous sommes convaincus du besoin d'un regroupement automatique de données sémantique à cause de la convergence des solutions tech-

nologiques de communication et la croissance des besoins en termes de traitement intelligent de données. En effet, le Web contient des informations déjà sous forme sémantique. Si certaines informations peuvent être considérées comme des données de contexte, nous pouvons améliorer l'interaction de l'utilisateur avec les services fournis sur la toile.

Ceci dit, certains thèmes de recherche peuvent constituer nos travaux dans les années à venir. Le premier est celui de l'interprétation automatique de groupes sémantiques de données. Nous ne disposons pas à l'heure actuelle de méthode pour l'interprétation automatique de connaissances découvertes de manière aveugle sur des données. Le second est la proposition d'une approche distribuée pour l'échange de connaissances apprises sur l'utilisation des services par les utilisateurs. De cette manière, l'adaptation des services peut se faire dans les différents espaces de vie de l'utilisateur.

BIBLIOGRAPHIE

- [1] Flanagan John A., Himberg Johan, and Mlántyjlarvi Jani. A hierarchical approach to learning context and facilitating user interaction in mobile devices. VTT Technical Research Centre of Finland, 2003.
- [2] Hartigan J. A. and Wong M. A. A K-means clustering algorithm. *Applied Statistics*, 28 :100–108, 1979.
- [3] Ward A., Jones A., and Hopper A. A new location technique for the active office. *IEEE Personal Communications*, 4(5) :42–47, 1997.
- [4] Ahmed K. Alasoud. *A Multi-Matching Technique for Combining Similarity Measures in Ontology Integration*. PhD thesis, Department of Computer Science and Software Engineering, Concordia University, Québec, Canada, 2009.
- [5] James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2) :123–154, 1984.
- [6] James F. Allen. Planning as temporal reasoning. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 3–14, Cambridge, MA, 1991.
- [7] Truong Binh An, Lee Young-Koo, and Lee Sung-Young. Modeling and reasoning about uncertainty in context-aware systems. In *Proceedings of the IEEE International Conference on e-Business Engineering*, pages 102–109. IEEE Computer Society, 2005.
- [8] Boytsov Andrey, Zaslavsky Arkady, and Synnes Kåre. Extending context spaces theory by predicting run-time context. In *Proceedings of the 9th International Conference on Smart Spaces and Next Generation Wi-*

red/Wireless Networking and Second Conference on Smart Spaces, pages 8–21, Berlin, Heidelberg, 2009. Springer-Verlag.

- [9] Theo A. Arentze, Frank Hofman, and Harry J. P. Timmermans. Deriving rules from activity diary data : A learning algorithm and results of computer experiments. *Journal of Geographical Systems*, 3(4) :325–346, 2001.
- [10] Miguel Arevalillo-Herráez, Juan Domingo, and Francesc J. Ferri. Combining similarity measures in content-based image retrieval. *Pattern Recognition Letters*, 29(16) :2174–2181, 2008.
- [11] Pashtan Ariel, Kollipara Shriram, and Pearce Michael. Adapting content for wireless web services. *IEEE Internet Computing*, 7(5) :79–85, 2003.
- [12] Schilit B., Adams N., and Want R. Context-aware computing applications. *IEEE Workshop on Mobile Computing Systems and Applications*, pages 85–90, 1899.
- [13] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4) :263–277, 2007.
- [14] Yazid Benazzouz, Philippe Beaune, Fano Ramaparany, and Olivier Boisier. Modeling and storage of context data for service adaptation. In Schahram Dustdar (Eds) Quan Z. Sheng, Jian Yu, editor, *Enabling Context-Aware Web Services : Methods, Architectures, and Technologies*, pages 470–490. CRC Press Taylor and Francis Group, 2010.
- [15] Meyer Beth. Retail user assistant : Evaluation of a user-adapted performance support system. In *Proceedings of the 4th International Conference on Human-Computer Interaction*, pages 45–55, London, UK, 1994. Springer-Verlag.

- [16] Arvind Bhusate and Jeremy Pitt. Pervasive adaptation for enhancing quality of experience. In *Proceedings of the 2nd Perada Workshop on Pervasive Adaptation*, pages 3–8, Edinburgh, Scotland, 2009.
- [17] Antonis Bikakis, Theodore Patkos, Grigoris Antoniou, and Dimitris Plexousakis. A survey of semantics-based approaches for context reasoning in ambient intelligence. In *Proceedings of the Workshop of Artificial Intelligence Methods for Ambient Intelligence at the European Conference on Ambient Intelligence*, pages 15–24, 2007.
- [18] Mark Lawrence Blum. *Real-time Context Recognition*. PhD thesis, Department of Information Technology and Electrical Engineering of Swiss Federal Institute of Technology Zurich (ETH), 2005.
- [19] Oliver Brdiczka, James L. Crowley, and Patrick Reignier. Learning situation models for providing context-aware services. In *Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction*, volume 4555 of *Lecture Notes in Computer Science*, pages 23–32. Springer, 2007.
- [20] Peter J. Brown. The stick-e document : A framework for creating context-aware applications. In *Proceedings of the International Conference on Electronic Publishing and Document Manipulation*, volume 2,3, pages 259–272. Electronic Publishing, 1995.
- [21] Peter J. Brown and Gareth J. F. Jones. Exploiting contextual change in context-aware retrieval. In *Proceedings of the ACM symposium on Applied Computing*, pages 650–656. ACM, 2002.
- [22] Patrick Brézillon. Hors du contexte, point de salut. In *Objets Communicants*, Autrans, France, 2002.

- [23] Oana Bucur, Philippe Beaune, and Olivier Boissier. What is context and how can an agent learn to find and use it when making decisions. In *Proceedings of the 4th International Central and Eastern European Conference on Multi-Agent Systems*, volume 3690, pages 112–121. Springer, 2005.
- [24] Hayes C., Massa P., Avesani P., and Cunningham P. An on-line evaluation framework for recommender systems. In *Proceedings of the Workshop on Personalization and Recommendation in E-Commerce*. Springer Verlag, 2002.
- [25] Johnson S. C. Hierarchical clustering schemes. *Psychometrika*, pages 241–254, 1967.
- [26] Laurent candillier. *Contextualisation, visualisation et évaluation en apprentissage non supervisé*. PhD thesis, Université charles de Gaulle Lille 3, 2006.
- [27] Stefano Ceri, Florian Daniel, Maristella Matera, and Federico M. Facca. Model-driven development of context-aware web applications. *ACM Transaction on Internet Technology*, 7(1–32), 2007.
- [28] Tarak Chaari, Ejigu Dejene, Frédérique Laforest, and Vasile Marian Scuturici. Modeling and using context in adapting applications to pervasive environments. In *Proceedings of the IEEE International Conference on Pervasive Services*. IEEE, 2006.
- [29] Tarak Chaari, Mohamed Zouari, and Frédérique Laforest. Ontology based context-aware adaptation approach. In *Book of Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability : Adaptive Technologies and Applications*, pages 26–58, Hershey - New York, 2009. Information science reference.

- [30] Harry Chen. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. PhD thesis, University of Maryland, Baltimore County, 2004.
- [31] Belkacem Chikhaoui, Yazid Benazzouz, and Bessam Abdulrazak. Towards a universal ontology for smart environments. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications and Services*, Kuala Lumpur, Malaysia, 2009.
- [32] Brewster Christopher and O'Hara Kieron. Editorial : Knowledge representation with ontologies : Present challenges-future possibilities. *International Journal of Human Computer Studies*, 65(7) :563–568, 2007.
- [33] Bettini Claudio, Brdiczka Oliver, Henricksen Karen, Indulska Jadwiga, Nicklas Daniela, Ranganathan Anand, and Riboni Daniele. A survey of context modelling and reasoning techniques. *Pervasive Mobile Computing*, 6(2) :161–180, 2010.
- [34] Rod Coffin. Create scalable semantic applications with database-backed rdf stores, store your rdf triples in a database for faster performance and greater scalability. Technical report, 2007.
- [35] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. *American Association for Artificial Intelligence*, 2003.
- [36] Robert Cooley, Pang-Ning Tan, and Jaideep Srivastava. Websift : The web site information filter system. In *Proceedings of Workshop on Web Usage Analysis and User Profiling WEBKDD in conjunction with ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego Ca, 2009.

- [37] Patrikla Dockhm Costa. *Architectural support for context-aware applications : From context models to services platforms*. PhD thesis, Centre for Telematics and information, University of Twente, The Netherlands, 2007.
- [38] Marcel Cremene, Michel Riveill, and Christian Martel. Towards unanticipated dynamic service adaptation. In *Proceedings of the Third International Workshop on Coordination and Adaptation Techniques for Software Entities (in conjunction with ECOOP'06)*, pages 25–34, Nantes, France, 2006.
- [39] Bolchini Cristiana, Curino Carlo A., Quintarelli Elisa, Schreiber Fabio A., and Tanca Letizia. A data-oriented survey of context models. *ACM SIGMOD*, 36(4) :19–26, 2007.
- [40] Dejene Ejigu Dedefa. *Context Modeling and Collaborative Context-Aware Services for Pervasive Computing*. PhD thesis, National Institute of Applied Sciences (INSA de Lyon) France, 2007.
- [41] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5(1) :4–7, 2001.
- [42] Anind K. Dey, Gregory D. Abowd, and Daniel Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2) :97–166, 2001.
- [43] Stefan Dietze, Alessio Gugliotta, and John Domingue. Bridging the gap between mobile application contexts and semantic web resources. In *Book of Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability : Adaptive Technologies and Applications*, pages 217–234, Hershey - New York, 2009. Information science reference.
- [44] Meshkova E., Riihijarvi J., Mahonen P., and Kavadias C. Modeling the home environment using ontology with applications in software configura-

- tion management. In *Proceedings of the international conference on telecommunications*, pages 1–6, 2008.
- [45] Papadopoulou Elizabeth, McBurney Sarah, Taylor Nick, and Williams M. Howard. A dynamic approach to dealing with user preferences in a pervasive system. In *Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications*, pages 409–416, Washington, DC, USA, 2008. IEEE Computer Society.
- [46] Jérôme Euzenat, Jérôme Pierson, and Fano Ramparany. Dynamic context management for pervasive applications. *Knowledge Engineering Review*, 23(1) :21–49, 2008.
- [47] Bry F., Hattori T., Hiramatsu K., Okadome T., Wieser C., and Yamada T. Context modeling in owl for smart building services. In *Proceedings of the GI-Workshop Über Grundlagen von Datenbanken, Wörlitz*, pages 38–42. Institute of Computer Science Martin-Luther-University Halle-Wittenberg, 2005.
- [48] Ciunchiglia F. Contextual reasoning. epistemologia. In *Proceedings IJ-CAI'93 Workshop on Using Knowledge in its Context*, pages 345–364, Chambery, France, 1993.
- [49] Françoise Fessant, Vincent Lemaire, and Fabrice Clérot. Combining several som approaches in data mining : Application to adsl customer behaviours analysis. In *Proceedings of studies in Classification, Data Analysis, and Knowledge Organization*, pages 343–354, 2007.
- [50] Fabien L. Gandon. *Graphes RDF et leur Manipulation pour la Gestion de Connaissances*. 2008.
- [51] Acampora Giovanni, Loia Vincenzo, Nappi Michele, and Ricciardi Stefano. Ambient intelligence framework for context aware adaptive appli-

- cations. In *Proceedings of the Seventh International Workshop on Computer Architecture for Machine Perception*, pages 327–332, Washington, DC, USA, 2005. IEEE Computer Society.
- [52] Fausto Giunchiglia and Chiara Ghidini. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2) :221–259, 2001.
- [53] Guillaume Grondin. *Un modèle d’agents auto-adaptables à base de composants*. PhD thesis, Ecole nationale supérieure des Mines de Saint-Etienne, France, 2009.
- [54] Chen Guanling and David Kotz. Context aggregation and dissemination in ubiquitous computing systems. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, page 105, Washington, DC, USA, 2002. IEEE Computer Society.
- [55] Ramanathan V. Guha. Contexts : a formalization and some applications. Technical Report STAN-CS-91-1399, Stanford Computer Science Department, 1991.
- [56] Peter Alesso H. and Craig F. Smith. *Thinking on the Web : Berners-Lee, Gödel and Turing*. Wiley-Interscience, 2006.
- [57] David L. Hall and James Linas. In *Handbook of Multisensor Data Fusion : Chapter 2*. CRC Press, 2001.
- [58] Bo Han, Weijia Jia, Ji Shen, and Man-Ching Yuen. Context-awareness in mobile web services. In *Proceedings of the Second International Symposium on Parallel and Distributed Processing and Applications*, Lecture Notes in Computer Science, pages 519–528, Hong Kong, China, 2004. Springer.

- [59] Chen Harry, Perich Filip, Chakraborty Dipanjan, Finin Tim, and Joshi Anupam. Intelligent agents meet semantic web in a smart meeting room. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 854–861, Washington, DC, USA, 2004. IEEE Computer Society.
- [60] Karen Henrickse. *A framework for context-aware pervasive computing applications*. PhD thesis, The school of information technology and electrical engineering of Queensland University, 2003.
- [61] Jeffrey Hightower, Anthony LaMarca, and Ian E. Smith. Practical lessons from place lab. *IEEE Pervasive Computing*, 5 :32–39, 2006.
- [62] Seon ho Park, Young ju Han, and Tai myoung Chung. Context-aware security management system for pervasive computing environment. In *CONTEXT*, pages 384–396, 2007.
- [63] Eric Horvitz, Susan Dumais, and Paul Koch. Learning predictive models of memory landmarks, 2004.
- [64] Heikki Hyyrö. A bit-vector algorithm for computing levenshtein and damerau edit distances. *Nord Journal Computing*, 10(1) :29–39, 2003.
- [65] Levenshtein V. I. Binary codes capable of correcting deletions, insertions and reversals. *Soviet, Physics Doklady*, 10 :707–710, 1966.
- [66] Himberg J., Flanagan J. A., and Mäntyjärvi J. Towards context awareness using symbol clustering map. In *Proceedings of the Workshop for self-organizing maps*, pages 249–254, 2003.
- [67] Himberg J., Korpiaho K., Mannila H., Tikanmaki J., and Toivonen H.T.T. Time series segmentation for context recognition in mobile devices. In *Proceedings of the IEEE International Conference on Data Mining*, pages 203–210, 2001.

- [68] Flanagan J.A. Unsupervised clustering of symbol strings. In *Proceedings of the International Joint Conference on Neural Networks*, volume 4, pages 3250–3255, 2003.
- [69] Indulska Jadwiga, Robinson Ricky, Rakotonirainy Andry, and Henricksen Karen. Experiences in using cc/pp in context-aware systems. In *Proceedings of the 4th International Conference on Mobile Data Management*, pages 247–261. Springer-Verlag, 2003.
- [70] Indulska Jaga, Loke Seng Wai, Rakotonirainy Andry, Witana Varuni, and Zaslavsky Arkady B. An open architecture for pervasive systems. In *Proceedings of the Third International Working Conference on New Developments in Distributed Applications and Interoperable Systems*, pages 175–188, Deventer, The Netherlands, The Netherlands, 2001. Kluwer, B.V.
- [71] Mantyjarvi Jani, Himberg Johan, Petri Kangas, Urpo Tuomela, and Pertti Huuskonen. Sensor signal data set for exploring context recognition of mobile devices. In *Proceedings of the Workshop “Benchmarks and a Database for Context Recognition” of 2nd International Conference on Pervasive Computing*, 2004.
- [72] Pascoe Mr. Jason. Adding generic contextual capabilities to wearable computers. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, page 92. IEEE Computer Society, 1998.
- [73] Zhong Jiwei, Zhu Haiping, Li Jianming, and Yu Yong. Conceptual graph matching for semantic search. In *Proceedings of the 10th International Conference on Conceptual Structures (ICCS)*, pages 92–196, London, UK, 2002. Springer-Verlag.
- [74] Coutaz Joëlle, Crowley James L., Dobson Simon, and Garlan David. Context is key. *Communications of the ACM*, 48(3) :49–53, 2005.

- [75] McCarthy John. Generality in artificial intelligence. *Communications of the ACM*, 30(12) :1030–1035, 1987.
- [76] McCarthy John and Buvac Sasa. Formalizing context (expanded notes). Technical report, 1994.
- [77] Hong Jongyi, Suh Eui-Ho, Kim Junyoung, and Kim SuYeon. Context-aware system for proactive personalized service based on context history. *Expert System Application*, 36(4) :7448–7457, 2009.
- [78] Jain A. K., Murty M. N., and Flynn P. J. Data clustering. *A review, ACM Computing Surveys*, 31 :264–323, 1999.
- [79] Sparck Jones K. A statistical interpretation of term specificity and its application. *Journal of Documentation*, 28 :11–21, 1972.
- [80] Henricksen Karen, Indulska Jadwiga, and Rakotonirainy Andry. Modeling context information in pervasive computing systems. In *Proceedings of the First International Conference on Pervasive Computing*, pages 167–180, London, UK, 2002. Springer-Verlag.
- [81] Fahim Kawsar, Kaori Fujinami, Susanna Pirttikangas, and Tatsuo Nakajima. Personalization and context aware services : A middleware perspective. In *Proceedings of the 2nd International Workshop on Personalized Context Modeling and Management for UbiComp Applications*, California, USA, 2006.
- [82] Panu Korpipää and Jani Mäntyjärvi. An ontology for mobile device sensor-based context awareness. In *Proceedings of the 4th International and Interdisciplinary Conference*, volume 2680 of *Lecture Notes in Computer Science*, pages 451–458. Springer, 2003.
- [83] Andreas Krause and Asim Smailagic. Unsupervised, dynamic identification of physiological and activity context in wearable computing. In

Proceedings of the 7th International Symposium on Wearable Computers, pages 88–97. IEEE Press, 2003.

- [84] Andreas Krause, Asim Smailagic, and Daniel P. Siewiorek. Context-aware mobile computing : Learning context-dependent personal preferences from a wearable sensor array. *IEEE Transactions on Mobile Computing*, 5 :113–127, 2006.
- [85] Kristof Van Laerhoven. Combining the self organizing map and k-means clustering for online classification of sensor data. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 464–469, 2001.
- [86] Serafini Luciano and Bouquet Paolo. Comparing formal theories of context in ai. *Artificial Intelligence*, 155(1-2) :41–67, 2004.
- [87] Cremene M., Riveill M., Martel M., Loghin C., and Miron C. Adaptation dynamique de services. In *Proceedings of the Première conférence francophone sur le déploiement et la (Re)Configuration de logiciels*, Grenoble, France, 2004. Actes de DECOR.
- [88] Kapitsaki Georgia M., Prezerakos George N., Tselikas Nikolaos D., and Venieris Iakovos S. Context-aware service engineering : A survey. *Journal of Systems Software*, 82(8) :1285–1297, 2009.
- [89] Kirsch-Pinheiro M., Vanrompay Y., and Berbers Y. Context-aware service selection using graph matching. In *Proceedings of the 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop*, volume 411 of *Lecture Notes in Computer Science*, pages 199–205, Dublin, Ireland, 2008. Springer.
- [90] Meila M. Comparing clusterings - an information based distance. *Journal Multivar Anal*, 98 :873–895, 2007.

- [91] Keidl Markus and Kemper Alfons. Towards context-aware adaptable web services. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 55–65, New York, NY, USA, 2004. ACM.
- [92] Cooper Matthew, Foote Jonathan, Girgensohn Andreas, and Wilcox Lynn. Temporal event clustering for digital photo collections. *ACM Transactions on Multimedia Computing, Communications and Applications*, 1 :269–288, 2005.
- [93] R. Mayrhofer. *An Architecture for Context Prediction*, volume 45 of *Schriften der Johannes-Kepler-Universität Linz*. Trauner Verlag, 2005.
- [94] Sarah McBurney, M. Howard Williams, Nick K. Taylor, and Eliza Papadopoulou. Managing user preferences for personalization in a pervasive service environment. *Advanced International Conference on Telecommunications*, 0 :32, 2007.
- [95] Richi Nayak and Cindy Tong. Applications of data mining in web services. In *Web Information Systems - WISE*, Lecture Notes in Computer Science, pages 199–205. Springer, 2004.
- [96] Katsumi Nihel. Context sharing platform. *Advanced Technologies and solutions toward ubiquitous network society*, special issue :200–204, 2004.
- [97] Petteri Nurmi and Patrik Floréen. Reasoning in context-aware systems, 2004.
- [98] Heierman Edwin O. and Cook Diane J. Improving home automation by discovering regularly occurring device usage patterns. In *Proceedings of the Third IEEE International Conference on Data Mining*, page 537, Washington, DC, USA, 2003. IEEE Computer Society.

- [99] Kwon Ohbyung and Kim Jihoon. Concept lattices for visualizing and generating user profiles for context-aware service recommendations. *Expert System Application*, 36(2) :1893–1902, 2009.
- [100] Reinhard Oppermann, Marcus Specht, and Igor Jaceniak. Hippie : A nomadic information system. In Hans-Werner Gellersen, editor, *Proceedings of the First International Symposium in Handheld and Ubiquitous Computing*, volume 1707 of *Lecture Notes in Computer Science*, pages 330–333, Karlsruhe, Germany, 1999. Springer.
- [101] Dempster A. P. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2) :325–339, 1967.
- [102] Dempster A. P., Laird N. M., and Rubain D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1) :1–38, 1977.
- [103] Langfelder P., Zhang B., and Horvath S. Defining clusters from a hierarchical cluster tree : the dynamic tree cut library for r. *Bioinformatics*, 24(5) :719–720, 2008.
- [104] Moran Thomas P. and Dourish Paul. Introduction to context-aware computing. *Human Computer Interaction*, 16(2) :87–95, 2001.
- [105] Dourish Paul. Seeking a foundation for context-aware computing. *Human computer interaction*, 16(2) :229–241, 2001.
- [106] Edoardo Pignotti, Peter Edwards, and Gunnar Aastrand Grimnes. Context-aware personalised service delivery. In *Proceedings of the European Conference on Artificial Intelligence*, pages 1077–1078, 2004.
- [107] Davy Preuveneers, Koen Victor, Yves Vanrompay, Peter Rigole, Manuele Kirsch, and Yolande Berbers. Context-aware adaptation in an ecology of

- applications. In *Book of Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability : Adaptive Technologies and Applications*, pages 1–25, Hershey - New York, 2009. Information science reference.
- [108] Liang Qianhui and Chung J. Y. Analyzing service usage patterns : Methodology and simulation. In *Proceedings of the IEEE International Conference on e-Business Engineering*, pages 359–362, Washington, DC, USA, 2007. IEEE Computer Society.
- [109] Wen Quan and He Jianmin. Personalized recommendation services based on service-oriented architecture. In *Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing*, pages 356–361, Washington, DC, USA, 2006. IEEE Computer Society.
- [110] Wen Quan and He Jianmin. Personalized recommendation services based on service-oriented architecture. In *Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing*, pages 356–361, Washington, DC, USA, 2006. IEEE Computer Society.
- [111] Roy Rada, Hafedh Bicknell, Ellen Mili, and Maria Blettner. Development and application of a metric on semantic nets. *IEEE Transaction on Systems, Man, and Cybernetics*, 1(19) :17–30, 1989.
- [112] Claudia Raibulet. Adaptive resource and service management in a mobile-enabled environment. In *Book of Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability : Adaptive Technologies and Applications*, pages 235–257, Hershey - New York, 2009. Information science reference.
- [113] Fano Ramparany, Yazid Benazzouz, and Marta Bel Martin. Agenda driven home automation - towards high level context aware systems. *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, 0 :125–130, 2009.

- [114] Oppermann Reinhard, editor. *Adaptive user support : ergonomic design of manually and automatically adaptable software*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1994.
- [115] N. S. Ryan, J. Pascoe, and D. R. Morse. Enhanced reality fieldwork : the context-aware archaeological assistant. In V. Gaffney, M. van Leusen, and S. Exxon, editors, *Computer Applications in Archaeology*, British Archaeological Reports, Oxford, 1998. Tempus Reparatum.
- [116] Mancoridis S., Mitchell B. S., Chen Y., and Gansner E. R. Bunch : A clustering tool for the recovery and maintenance of software system structures. In *Proceedings of the IEEE International Conference on Software Maintenance*, pages 50–59. IEEE Computer Society Press, 1999.
- [117] Yau Stephen S. and Karim Fariaz. A context-sensitive middleware for dynamic integration of mobile devices with network infrastructures. *Journal of Parallel and Distributed Computing*, 64(2) :301–317, 2004.
- [118] Walid S. Saba. Language, logic and ontology : Uncovering the structure of commonsense knowledge. *International Journal of Man-Machine Studies*, 65(7) :610–623, 2007.
- [119] Schou Saowanee. Context-based service adaptation platform : Improving the user experience towards mobile location services. In *Proceedings of the International Conference on Information Networking*, volume 23,25, pages 1–5, 2008.
- [120] McBurney Sarah, Papadopoulou Eliza, Taylor Nick, and Williams Howard. Adapting pervasive environments through machine learning and dynamic personalization. In *Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications*, pages 395–402, Washington, DC, USA, 2008. IEEE Computer Society.

- [121] Greenberg Saul. Context as a dynamic construct. *Human Computer Interaction*, 16(2) :257–268, 2001.
- [122] Fritsch Serena, Senart Aline, and Clarke Siobhan. Addressing dynamic contextual adaptation with a domain-specific language. In *Proceedings of the 1st International Workshop on Software Engineering for Pervasive Computing Applications, Systems, and Environments*, page 2, Washington, DC, USA, 2007. IEEE Computer Society.
- [123] Louis S.J. and Shankar A. Context learning can improve user interaction. In *Proceedings of the IEEE International Conference on Information Reuse and Integration*, pages 115–120, 2004.
- [124] Andrew Jennings Smieeee. Customer adaptive communications services. In *Proceedings of 10 IEEE Region conference TENCOM*, pages 886–890, Melbourne, Australia, 1992.
- [125] Thomas Strang and Claudia Linnhoff Popien. A context modeling survey. In *Proceedings of the Workshop “Advanced Context Modelling, Reasoning and Management” of the Sixth International Conference on Ubiquitous Computing*, 2004.
- [126] Kohonen T., Schroeder M. R., and Huang T. S., editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [127] Chaari Tarak, Ejigu Dejene, Laforest Frédérique, and Scuturici Vasile-Marian. A comprehensive approach to model and use context for adapting applications in pervasive environments. *Journal of systeme software*, 80(12) :1973–1992, 2007.
- [128] Winograd Terry. Architectures for context. *Human-Computer Interaction*, 16(2) :401–419, 2001.

- [129] Gerald Tesauro, David S. Touretzky, and Todd K. Leen. *Advances in Neural Information Processing Systems*. MIT Press, Denver, Colorado, USA, 1995.
- [130] Hong-Linh Truong and Schahram Dustdar. A survey on context-aware web service systems. *International Journal of Web Information Systems*, 5(1) :5–31, 2009.
- [131] Mathieu Vallée, Fano Ramparany, and Laurent Vercouter. Using device services and flexible composition in ambient communication environments. In *Proceedings of the 1st International Workshop on Requirements and Solutions for Pervasive Software Infrastructure (RSPSI)*, Dublin, Ireland, 2006.
- [132] Laerhoven Kristof Van. Combining the self-organizing map and k-means clustering for on-line classification of sensor data. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 464–469, London, UK, 2001. Springer-Verlag.
- [133] Stijn van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.
- [134] Roberto De Virgilio and Riccardo Torlone. Modeling heterogeneous context information in adaptive web based applications. In *Proceedings of the 6th International Conference on Web Engineering*, pages 56–63, New York, NY, USA, 2006. ACM.
- [135] Yan Wang. Web mining and knowledge discovery of usage patterns. Technical report, 2000.
- [136] Kevin Wilkinson, Craig Sayers, Harumi A. Kuno, and Dave Reynolds. Efficient rdf storage and retrieval in jena2. In *Exploiting Hyperlinks*, pages 35–43, 2003.

- [137] Hong Xin, Nugent Chris, Mulvenna Maurice, McClean Sally, Scotney Bryan, and Devlin Steven. Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive Mobile Computing*, 5(3) :236–252, 2009.
- [138] Manuel Montes y Gómez, Alexander Gelbukh, and Aurelio López-López. Comparison of conceptual graphs. In *Proceedings of the 1st Mexican International Conference on Artificial Intelligence*, Lecture Notes in Artificial Intelligence, pages 548–556, Acapulco, Mexico, 2000. Springer.
- [139] Yuping Yang, Fiona Mahon, M. Howard Williams, and Tom Pfeifer. Context-aware dynamic personalised service re-composition in a pervasive service environment. In *Proceedings of the Third International Conference in Ubiquitous Intelligence and Computing*, volume 4159 of *Lecture Notes in Computer Science*, pages 724–735. Springer, 2006.
- [140] Sungju Yun, Anjin Park, and Keechul Jung. Graph-based high level motion segmentation using normalized cuts. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 44, pages 306–311, 2008.
- [141] Wu Z. and Palmer M. Verb semantics and lexical selection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 1994.
- [142] Sofia Zaidenberg. *Apprentissage par renforcement de modèles de contexte pour l’informatique ambiante*. PhD thesis, Grenoble INP France, 2009.
- [143] M. Zuidweg, J. Goncalves Filho, and M. J. van Sinderen. Using p3p in a web services-based context-aware application platform. In E. Halasz, C. Lukovszki, and T. Marosits, editors, *Proceedings of EUNICE 2003 9th Open European Summer School and IFIP WG6.3 Workshop on Next Generation Networks*, pages 238–243, Budapest, 2003. Budapest University of Technology and Economics.

