ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

ADAPTATION DYNAMIQUE DES SERVICES DANS L'INFORMATIQUE UBIQUITAIRE PAR UTILISATION DE SIMILARITÉ CONTEXTUELLE

PAR
Djamel GUESSOUM

THÈSE PAR ARTICLES PRÉSENTÉE À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION
DU DOCTORAT EN GÉNIE
Ph. D.

MONTRÉAL, LE 24 NOVEMBRE 2016

©Tous droits réservés, Djamel Guessoum, 2016

©Tous droits réservés

Cette licence signifie qu'il est interdit de reproduire, d'enregistrer ou de diffuser en tout ou en partie, le présent document. Le lecteur qui désire imprimer ou conserver sur un autre media une partie importante de ce document, doit obligatoirement en demander l'autorisation à l'auteur.

PRÉSENTATION DU JURY

CETTE THÈSE A ÉTÉ ÉVALUÉE

PAR UN JURY COMPOSÉ DE :

M. Chakib Tadj, directeur de thèse
Département de génie électrique à l'École de technologie supérieure

M. Moeiz Miraoui, codirecteur de thèse
Institut supérieure des sciences appliquées et de technologie à l'Université de Gafsa, Tunisie

M. Talhi Chamseddine, président du jury
Département du génie logiciel et des TI à l'École de technologie supérieure

M. Wong Tony, membre du jury
Département du génie de la production automatisée à l'École de technologie supérieure

M. Bessam Abdulrazak, examinateur externe
Département d'informatique à l'Université de Sherbrooke

ELLE A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 08 NOVEMBRE 2016

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

La persévérance, le travail assidu et l'engagement à atteindre ses buts ainsi que l'accompagnement et la confiance de mes directeurs ont été les piliers de cet humble accomplissement que j'espère sera le fondement pour une transition positive dans ma vie personnelle et professionnelle.

Mes plus sincères remerciements et ma plus profonde gratitude vont à mon directeur de recherche Monsieur Chakib Tadj. Son support continu, sa disponibilité sans limites et ses directives de maître ont été la clé de cet accomplissement.

Mes sentiments les plus sincères et mes remerciements les plus cordiaux vont à mon codirecteur Mr. Moeiz Miraoui, pour son support continu, ses précieux conseils et ses directives pertinentes qui ont guidé mon parcours.

Un grand merci à M. Chamseddine Talhi de m'avoir fait l'honneur de présider ce jury. Un grand merci également aux professeurs M. Tony Wong et M. Bessam Abdulrazak qui ont accepté d'examiner ce travail.

Je dédie ce travail à toute ma famille. À mes parents, ma très chère mère Rachida et mon père Ferhat (A.y), sources inépuisables de générosité. À ma très chère fille Hadile, qui j'espère accomplira un jour ses rêves. À tous mes frères et à ma sœur pour leur support sans limites, à tous mes amis et collègues.

ADAPTATION DYNAMIQUE DES SERVICES DANS L'INFORMATIQUE UBIQUITAIRE PAR UTILISATION DE SIMILARITÉ CONTEXTUELLE

Djamel GUESSOUM

RÉSUMÉ

L'informatique diffuse ou ubiquitaire est une technologie récente qui a émergé de la convergence de plusieurs technologies : l'informatique conventionnelle où le PC est l'entité centrale de traitement de l'information, les systèmes distribués, les réseaux de télécommunications, l'électronique, les systèmes embarqués et finalement Internet. L'idée principale est que la technologie puisse s'adapter aux préférences de l'humain tout en restant transparente. L'objectif de l'informatique ubiquitaire est de permettre l'interconnexion de tous les domaines de la vie en permettant la circulation ubiquitaire des données. L'interaction et l'adaptation transparente de l'environnement avec l'utilisateur est fondée sur le concept de contexte et la conception d'applications sensibles à ce contexte et à ses changements. Les informations du contexte sont acquises à partir de capteurs physiques ou virtuels. Elles sont stockées et mises sous une forme adéquate au modèle du contexte. Elles sont ensuite utilisées par des niveaux d'abstraction supérieurs pour produire les actions requises adaptées à l'utilisateur.

L'adaptation des services fournis à un utilisateur dans un system informatique diffus ou SID est le but de notre travail. En informatique diffuse, l'adaptation des services est un processus dynamique, où les services sont offerts à un utilisateur d'une façon individuelle, d'une manière réactive, en réponse à un changement d'état du contexte ou proactive qui prédit un changement du contexte et réagit en conséquence. Notre présent travail est essentiellement un pas dans cette direction. Il consiste à présenter une approche d'adaptation des services par l'exploitation des similarités entre un contexte courant et des contextes de référence à l'aide de mesures de similarités sémantiques. Le contexte sera défini, modélisé et utilisé comme un facteur : 1) de sélection et d'amélioration des services fournis et 2) de prédiction du contexte future afin de permettre la proactivité de l'adaptation des services.

Mots clés : Informatique ubiquitaire, informatique diffuse, contexte, contexte de référence, sensibilité au contexte, similarité sémantique, services, raisonnement par cas, prédiction de la localisation, Clustering.

DYNAMIC SERVICES ADAPTATION IN UBIQUITOUS COMPUTING USING CONTEXTUAL SIMILARITIES

Djamel GUESSOUM

ABSTRACT

Pervasive computing or ubiquitous computing is a recent technology that has emerged from the convergence of several technologies: conventional computing where the PC is the central entity of information processing, distributed systems, telecommunications networks, electronics, embedded systems and finally internet. The main idea is that the technology can adapt to the preferences of the human while remaining transparent. The goal of ubiquitous or pervasive computing is the interconnection of all areas of life by enabling ubiquitous traffic data. The interaction and seamless adaptation to the environment with the user is based on the concept of context and the design of context-aware applications sensitive to the changes in the context. The context information is acquired from physical or virtual sensors. They are stored and modeled in a suitable form to the context model. They are then used by higher levels of abstraction to produce the requested actions adapted to the user.

The adaptation of services provided to a user in a pervasive computing system or PCS is the purpose of our present work. In Pervasive computing, the adaptation of services is a dynamic process, where services are provided to a user in an individual manner in a reactive way, reacting to a change in the context or proactively by predicting a change of the context and reacts consequently. Our present work is essentially a step in this direction which consists in presenting an adaptation of services approach by exploiting the similarities between a current context and reference contexts using semantic similarity measures. The context will be defined and modeled and will be used as a factor: 1) for selection and improvement of services and 2) for prediction of future context to enable proactive adaptation of services.

Keywords: ubiquitous computing, pervasive computing, context, reference context, context-awareness, semantic similarity, services, case based reasoning, location prediction, clustering.

TABLE DES MATIÈRES

Page

INTRODUCTION 1
0.1 Cadre de recherche 1
0.2 Description de la problématique 3
0.3 Objectifs de la recherche 5
0.4 Méthodologie 7
0.5 Originalité des travaux proposés et contributions 10
0.5.1 Définition du contexte de référence 11
0.5.2 Mesure de similarité entre un contexte courant et un contexte de référence 11
0.5.3 Application des mesures de similarité à l'adaptation dynamique de services 12
0.5.4 Proposition d'une pondération des informations contextuelles de type catégoriques 13
0.5.5 Amélioration de la mesure de similarité de Wu et Palmer entre taxonomies 14
0.5.6 Reconnaissance de l'activité d'un utilisateur à partir des capteurs d'un dispositif électronique (smartphone) 14
0.5.7 Prédiction de la localisation 14
0.6 Organisation de la thèse 15

CHAPITRE 1 REVUE DE LA LITÉRATURE 17
1.1 Introduction 17
1.2 Définitions du contexte dans l'informatique diffuse 17
1.3 Modélisation du contexte 22
1.3.1 Approches de modélisation du contexte 24
1.3.1.1 Modèle spatial de modélisation du contexte 24
1.3.1.2 Modèles orientés Objet- Rôle 25
1.3.1.3 Modèles ontologiques du contexte 25
1.3.1.4 Comparaison des approches de modélisation 28
1.4 La sensibilité au contexte 28
1.4.1 Sensibilité au contexte et adaptation des services 30
1.4.2 Première génération de systèmes sensibles au contexte 33
1.4.3 Deuxième génération de systèmes sensibles au contexte 33
1.4.4 Troisième génération de systèmes sensibles au contexte 34
1.5 Adaptation dynamique de services dans l'informatique diffuse 37
1.5.1 Définitions 38
1.5.2 Adaptation dynamique vs. adaptation statique 40
1.6 Les mesures de similarité en informatique diffuse 41
1.6.1 La similarité 41
1.6.2 Les mesures de similarité sémantique 43

1.6.2.1 Les mesures de similarité sémantique appliquées aux ontologies 43
1.6.2.2 Mesures de similarité vectorielle 53
1.6.2.3 Mesures de similarités et adaptation de services 55
1.7 Conclusion 56

CHAPITRE 2 SURVEY OF SEMANTIC SIMILARITY MEASURES IN PERVASIVE COMPUTING 59
2.1 Introduction 60
2.2 Dynamic adaptation of services in pervasive computing 61
2.3 Notion of semantic similarity 62
2.3.1 Notion of distance and similarity 63
2.4 Application of semantic similarity in pervasive computing 69
2.4.1 Semantic similarity measures and context 70
2.4.1.1 Impact of context 71
2.4.1.2 Semantic similarity between contexts 73
2.4.2 Recommendation of services in a PCS 74
2.4.2.1 Context-aware services 77
2.4.3 Semantic similarity measures and applications 80
2.4.4 Service discovery 81
2.5 Conclusion 87

CHAPITRE 3 A MEASURE OF SEMANTIC SIMILARITY BETWEEN A REFERENCE CONTEXT AND A CURRENT CONTEXT 89
3.1 Introduction 90
3.2 Related work 92
3.3 Context in pervasive computing 94
3.4 The reference context 97
3.4.1 Context variable 98
3.5 Semantic similarity measures 99
3.5.1 A measure of semantic similarity between a current context and a reference context 99
3.5.2 Weighting of contextual variables 100
3.5.3 Measures of semantic similarity between quantitative and quantifiable variables 102
3.5.4 Measures of semantic similarity between categorical variables 103
3.5.5 Overall semantic similarity 104
3.6 Case study 105
3.7 Conclusion 110

CHAPITRE 4 CONTEXTUAL CASE BASED REASONING APPLIED TO A MOBILE DEVICE 111
4.1 Introduction 112
4.2 Related work 116
4.3 Our Approach 118
4.3.1 Case Representation 119

4.3.2 Context acquisition 121
4.3.3 Service filtering 123
4.3.3.1 Case Retrieval and Similarity Measure 124
4.3.3.2 Modified similarity measure 125
4.4 Experimental Method 126
4.4.1 Daily Routine Locations 127
4.4.2 Activity Recognition Parameters 128
4.4.3 Similarity Calculation 129
4.4.4 Application 131
4.5 Conclusion 136

CHAPITRE 5 CONTEXTUAL LOCATION PREDICTION USING SPATIO-TEMPORAL CLUSTERING 137
5.1 Introduction 138
5.2 Related work 140
5.3 Context prediction techniques 145
5.4 Clustering 148
5.4.1 DBSCAN and temporal filtering 149
5.5 Experimentation 151
5.5.1 Prediction process 157
5.6 Conclusion 161

CONCLUSION…. 163

BIBLIOGRAPHIE 167

LISTE DES TABLEAUX

Page

Tableau 1.1 Comparaison des plateformes d'applications sensibles au contexte de troisième génération. 27

Tableau 1.2 Exemple de modèles de systèmes sensibles au contexte. 32

Tableau 2.1 Semantic similarity measures. 68

Tableau 2.2 Semantic similarity measures applied to service recommendations. 79

Tableau 2.3 Table summarizing the studies on semantic similarities in pervasive computing. 85

Tableau 3.1 Context types, variables, and attributes. 96

Tableau 3.2 Attributes of contextual variables by location. 107

Tableau 3.3 Overall semantic similarity. 108

Tableau 3.4 Weight of contextual variables by location. 108

Tableau 4.1 Case structure. 121

Tableau 4.2 Initial Case Base 128

Tableau 4.3 Similarity of Physical Activities. 129

Tableau 4.4 Similarities between Locations "Location_Entertainment" 131

Tableau 4.5 Example of a new case 132

Tableau 4.6 Filtering rules. 133

Tableau 4.7 Retrieval and Filtering Results 135

Tableau 5.1 Location prediction accuracy with noise (Bold) and without noise (user #5542). 158

Tableau 5.2 Average location prediction accuracy (user #5578). 161

LISTE DES FIGURES

Page

Figure 0.1	Méthodologie et structure du travail.	7
Figure 0.2	Processus de l'adaptation par mesure de similarités sémantiques	13
Figure 1.1	Types de contexte.	21
Figure 1.2	Classification des variables contextuelles.	22
Figure 1.3	Application tenant compte du contexte.	29
Figure 1.4	Processus général des systèmes sensibles au contexte.	31
Figure 1.5	Architecture CoBra.	35
Figure 1.6	Architecture de la plateforme SOCAM.	36
Figure 1.7	Architecture de la plateforme CMF.	37
Figure 1.8	Relation entre les concepts d'adaptation.	40
Figure 1.9	Adaptation dynamique vs. statique.	41
Figure 1.10	Approches de mesure de similarité sémantique.	44
Figure 1.11	Exemple de structure taxonomique.	45
Figure 1.12	Exemple d'une ontologie.	47
Figure 2.1	Layered model of semantic similarity measures between ontologies and intra-ontologies.	66
Figure 2.2	Application of semantic similarity measures in pervasive computing systems.	70
Figure 2.3	Service recommendation systems and context-aware service recommendation systems.	77
Figure 2.4	Context-aware applications architecture.	80
Figure 2.5	Service discovery in a PCS and semantic similarity measures.	83
Figure 3.1	Classification of context variables.	99
Figure 3.2	Weighting of the contextual variables	101
Figure 3.3	Scalar variable for temperature.	103
Figure 3.4	Weight changes according to no and nt/n.	109
Figure 4.1	Relationship between problem and solution spaces in CBR.	115

Figure 4.2 Context and the CBR cycle....117
Figure 4.3 Context acquisition and service filtering. 119
Figure 4.4 Hierarchic structure of a case.....123
Figure 4.5 Wu and Palmer similarity measure 125
Figure 4.6 Taxonomies of the symbolic attributes “location” and “phone usage”. 130
Figure 4.7 Retrieved and filtered services.....134
Figure 5.1 Contextual information for location prediction. 145
Figure 5.2 The DBSCAN algorithm.....150
Figure 5.3 DBSCAN and temporal filtering. 151
Figure 5.4 User trajectories. 152
Figure 5.5 Collected points for user #5542. 153
Figure 5.6 Speed histogram.....154
Figure 5.7 Cluster parameters. 155
Figure 5.8 GPS data preprocessing and DBSCAN clustering.....155
Figure 5.9 Cluster PI Saturday. 157
Figure 5.10 Construction of next cluster. 158
Figure 5.11 Saturday (point of interest #6) ROC (receiver operating characteristic). 159
Figure 5.12 Average day-AUC for Bayes Net, J48/C4.5, and KNN. 160

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

SID	Système Informatique Diffus
CBR	Case Based Reasoning
DBSCAN	Density-based spatial clustering of applications with noise
KNN	K-Nearest Neighbors
MDC	Mobile Data Challenge
SVM	Support Vector Machines
sim	Similarity function
w	Weight of a contextual variable
IOF	Inverse Occurrence Frequency
OF	Occurrence Frequency
FFT	Fast Fourier Transform
SD	Standard Deviation
VAR	Variance
GPS	Global Positioning System
CML	Context Modelling Language
ORM	Object-Role Modelling
COBRA	Common Object Request Broker Architecture
SOCAM	Service-oriented Context-Aware Middleware
CMF	Context Management Framework
XML	EXtensible Markup Language
OWL	Web Ontology Language
DL	Description Logics
RDF	Resource Description Framework
W3C	World Wide Web Consortium
NLP	Natural language processing
dis.	Function de distance
LCS	Least Common Subsumer
IC	Informational Content
log	Function Logarithmique

PCS	Pervasive Computing System
Imp(c_n)	Impact d'une variable contextuelle c_n
TF-IDF	Term Frequency–Inverse Document Frequency
cos	Function cosinus
DTD	document type définition
SDK	Software Development Kit
Acc	Acceleration
WLAN	Wireless Local Area Network
WiFi	Wireless Fidelity
PI	Point d'Intérêt
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve

INTRODUCTION

0.1 Cadre de recherche

Les technologies de l'information sont devenues une partie intégrante de la vie de tous les jours qui a commencé avec l'apparition de l'ordinateur personnel et n'a cessé d'évoluer avec l'apparition des ordinateurs portables et des tablettes. L'évolution la plus spectaculaire est survenue avec l'évolution des réseaux d'ordinateurs et des télécommunications qui ont permis la mobilité et le partage de l'information. À la fin du 20ieme siècle, une nouvelle vision commence à prendre forme prônant la présence ubiquitaire et transparente de « l'intelligence » dans l'environnement des utilisateurs et l'adaptation de celle-ci à ses besoins. Ceci a conduit à l'apparition de l'informatique diffuse. Cette nouvelle branche est l'intégration des technologies de l'informatique distribuée, de l'informatique mobile, des télécommunications et de l'électronique.

L'idée principale est que la technologie puisse s'adapter aux préférences de l'utlisateur tout en restant transparente. Plusieurs appellations ont été proposées comme "pervasive computing", "ambient intelligence", "everyware", " physical computing", "the Internet of Things. Mais la définition la plus utilisée est « ubiquitous computing". Chaque appellation met l'accent sur un aspect et toutes s'accordent sur l'omniprésence de l'informatique dans la vie de tous les jours. Elle favorise la création d'environnements intelligents tels que des maisons qui s'adaptent aux besoins de ses occupants, des campus universitaires ou des guides touristiques. Les applications n'ont pas de limites et les horizons sont très vastes.

L'interaction avec l'utilisateur et l'adaptation transparente de l'environnement avec celui-ci sont fondées sur le concept de contexte et la conception d'applications sensibles à ce contexte et à ses changements. Les informations du contexte sont acquises à partir de capteurs physiques ou virtuels. Elles sont stockées et mises sous une forme adéquate au modèle du contexte. Elles sont ensuite utilisées par des niveaux d'abstraction supérieurs pour produire les actions requises adaptées à l'utilisateur.

Pour développer les prochains systèmes pour l'informatique diffuse, trois défis majeures doivent être adressés (Yong Bin Kang, 2006) :

1. Modéliser les informations du contexte

Modéliser un contexte est une tâche difficile vu le caractère dynamique de celui-ci et de celui de l'utilisateur qui peut avoir des objectifs variables dans le temps.

2. Comprendre les interactions complexes

Il existe plusieurs types d'interactions qui doivent être comprises pour modeler un contexte d'une manière fidèle : 1- interactions entre l'utilisateur et l'environnement, 2- l'interaction entre plusieurs ressources environnementales 3- interactions entre plusieurs utilisateurs 4- interactions entre groupes sociaux (famille, amis etc.) 4-combinaison de toutes ces interactions.

3. Fournir les meilleurs services

Les services fournit ne doivent en aucun cas être en conflit avec les préférences de l'utilisateur et ainsi le défi est de pouvoir acquérir ces préférences d'une manière la plus précise possible.

Notre présent travail est un pas dans cette direction. Nous présenterons dans cette thèse des contributions liées à ces défis, essentiellement au troisième chapitre où nous répondons à la question suivante : Étant donné un contexte courant, quels sont les meilleurs services à fournir à un utilisateur?

Pour ceci nous avons choisi deux approches différentes : 1- la première est une approche de comparaison et d'ajustement de services au présent exploitant les informations contextuelles et 2- la deuxième approche consiste à prédire le contexte futur d'un utilisateur afin de permettre l'adaptation proactive d'une application sensible au contexte.

0.2 Description de la problématique

L'adaptation des services fournis à un utilisateur dans un Système Informatique Diffus (SID) est le but de notre travail. En informatique diffuse, l'adaptation des services est un processus dynamique, où les services sont offerts à un utilisateur d'une façon individuelle, d'une manière réactive, en réponse à un changement d'un état du contexte ou proactive qui prédit un changement du contexte et réagit en conséquence (Germán S., 2010). Notre présent travail est essentiellement un pas dans cette direction qui consiste à présenter des applications du processus d'adaptation des services où le contexte est pris en compte comme facteur : 1-de sélection et d'amélioration des services fournis et 2- de prédiction du contexte future afin de permettre la proactivité de l'adaptation des services. Cette adaptation dynamique des services utilise différents types de mécanismes qui sont résumés ci-dessous (Yazid B., 2011) :

1. **Adaptation à base de règles :**

Ce mécanisme d'adaptation dynamique des services consiste à écrire l'ensemble des règles logiques qui conditionnent le déclenchement d'un service dans un contexte donné.
Ce mécanisme est simple à mettre en œuvre mais le développeur doit avoir une connaissance de toutes les situations de déclenchement des services. Il doit aussi s'impliquer dans les nouvelles situations et il doit écrire pour chaque service des règles spécifiques.

2. **Adaptation à base d'apprentissage automatique :**

Avec ce type d'adaptation dynamique de services, les informations du contexte sont détectées et la sélection du service approprié est basée sur un entrainement de la part de l'utilisateur en utilisant un apprentissage approprié.

Ce mécanisme est flexible et donne la possibilité de fournir des solutions commerciales mais nécessite une phase d'entrainement avec une difficulté de mise en œuvre en temps réel.

3. Adaptation à base de fouille de données :

Cette approche consiste à explorer les données actuelles du contexte pour détecter des situations connues ou des tendances afin de fournir les services appropriés.

Les techniques employées sont celles provenant du domaine de la fouille de données. Il peut être utilisé dans le domaine du Web (p. ex. : la recherche Web) et dans la recommandation de services.

4. Adaptation à base de comparaison :

Cette approche consiste à comparer les conditions liées à l'exécution d'un service avec les données de la situation courante. Le service qui correspond le mieux à la situation courante est sélectionné.

Ce type de mécanisme d'adaptation est rarement mentionné dans la littérature et sa particularité est la discontinuité des services pour les cas non prévus.

Les mécanismes d'adaptation cités relèvent les problèmes suivants :

1. Les mécanismes d'adaptation à base de comparaison par utilisation de similarités sémantiques sont rarement ou jamais été testés;
2. Les services fournis à l'utilisateur sont des services individuels, liés au changement d'un état du contexte et tiennent seulement compte du contexte immédiat de l'utilisateur qui est défini par une seule variable contextuelle (p.ex.: quand la lumière s'éteint (variable=luminosité) le téléphone change son degré de luminosité en conséquence), ignorant le contexte général qui est défini par un ensemble de variables (p.ex.: le contexte «maison» est défini par l'ensemble des variables contextuelles (type d'espace, période de la journée, voisinage, température ambiante, etc.);
3. Le contexte et les informations contextuelles exploités ne sont pas toujours ceux qui sont les plus aisément acquis.

Notre travail est conçu pour répondre à ces problèmes en proposant une adaptation de services à base de comparaison de contextes par mesure de similarités sémantiques entre le contexte courant avec un certain nombre de contextes de références, où les services sont

fournis d'une manière collective et individuelle et où le contexte général est toujours pris en considération.

Afin d'atteindre ce but, nous devrons adopter ou clarifier un certain nombre de concepts comme la définition du contexte, les différents types de modélisation et les services à fournir et définir de nouveaux concepts liés à notre travail comme le contexte de référence.
Nous devons aussi répondre à des questions pertinentes comme : «Sur quelle base les contextes de références vont être choisis».

La suite de ce rapport aborde ces points ainsi que la planification des tâches qui mèneront aux résultats escomptés.

0.3 Objectifs de la recherche

L'objectif principal de ce travail de recherche consiste à développer un système d'adaptation dynamique des services selon le contexte dans un système informatique diffus.

Cette approche consiste essentiellement à comparer entre les données d'un contexte de base ou de référence pour lequel les services à fournir sont prédéfinis avec les données du contexte courant. La comparaison se fait par l'utilisation de mesures de similarités sémantiques. Comme il sera montré dans la revue de la littérature, cette approche n'est pas nouvelle mais elle n'a pas été suffisamment abordée. Un de nos objectifs sera de montrer quelques applications de cette approche dans le domaine de l'informatique ubiquitaire et de :

1. Faire une modélisation adéquate des informations contextuelles qui favorise leurs partages entre les équipements qui composent le système informatique diffus. À cet effet, la modélisation par ontologies sera effectuée car elle satisfait aux objectifs de ce travail à savoir le partage des informations contextuelles et la possibilité de faire des raisonnements logiques sur les données détectées pour déduire de nouvelles informations;

2. Proposer ou adopter une définition du contexte parmi la multitude de définitions proposées dans la littérature et argumenter le choix;
3. Proposer ou modifier une mesure de similarité sémantique entre variables contextuelles;
4. Appliquer les concepts précédents pour l'adaptation dynamique (sensible au contexte) réactive et proactive des services.

0.4 Méthodologie

Pour mener les travaux de recherche à bien, la méthodologie montrée dans la (Figure 0.1) est adoptée :

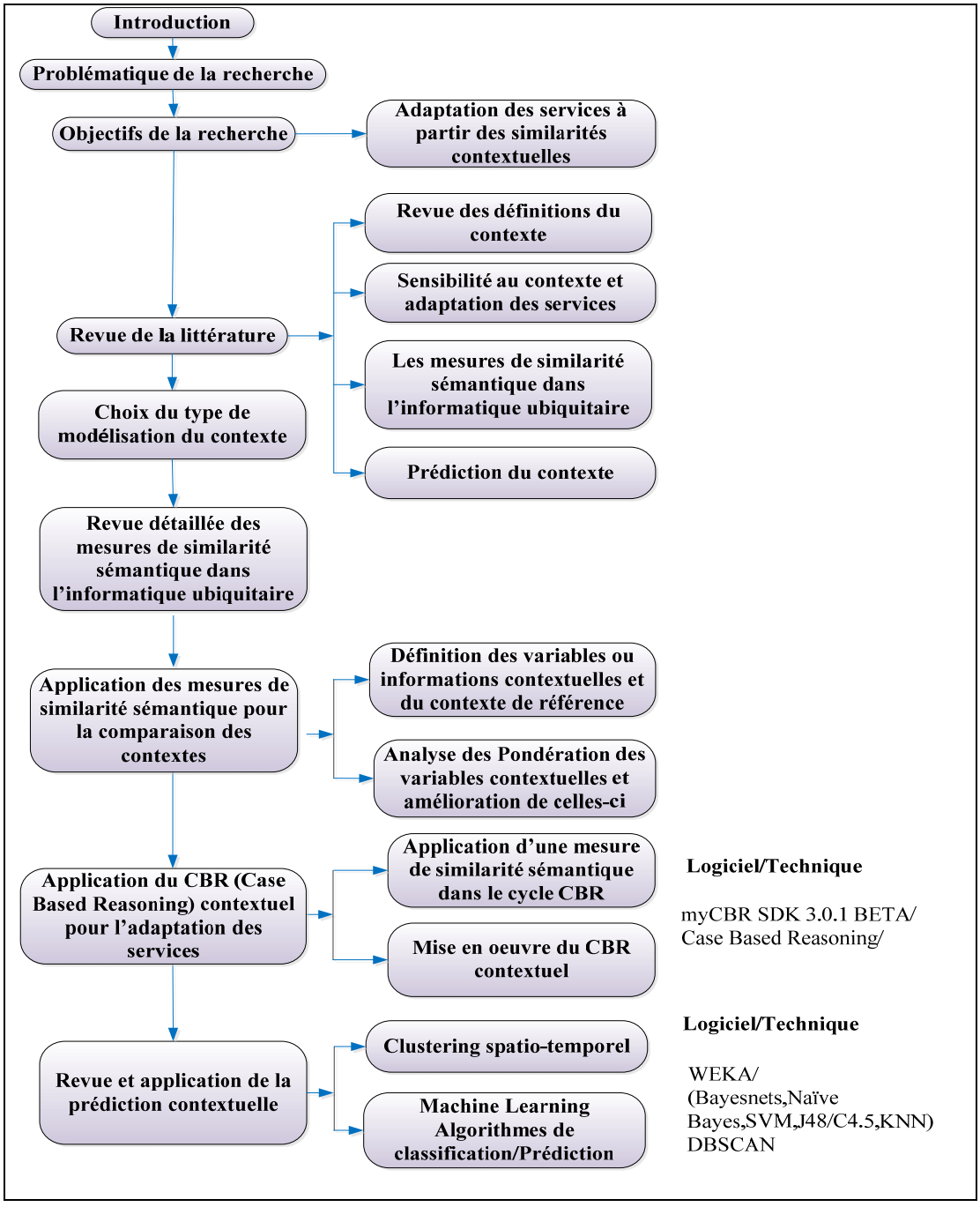

Figure 0.1 Méthodologie et structure du travail

Notre méthodologie se base sur une revue détaillée de la littérature dans les domaines liés à nos objectifs. Elle est composée des étapes suivantes :

1. Définition du contexte

Suite à une étude complète des définitions du contexte proposées dans la littérature, nous avons pris l'essence de ces définitions et leurs domaines d'applications et nous avons examiné les différentes alternatives pour soit les adapter à notre travail ou soit en proposer une nouvelle qui peut être une combinaison de certaines de ces définitions.

2. Choix des modèles de référence pour un équipement dans un SID

Cette étape consiste à définir, pour un équipement particulier, des contextes de base et plus particulièrement le ou les critères selon lesquels ces contextes sont choisis. Il sera question de répondre à la question suivante :
Sur quels critères un contexte de référence doit-il être défini ?

- sur la localisation de l'utilisateur;
- sur son activité;
- sur son profil ou identité;
- sur une certaine combinaison de ces critères.

Dans (Chen, G., and Kotz D., 2000), le contexte a trois formes : 1) contexte selon la localisation de l'utilisateur, 2) le contexte selon l'activité de l'utilisateur et 3) le contexte selon l'identité et le profil de l'utilisateur.

Dans notre travail de recherche, la localisation de l'utilisateur sera le facteur sur lequel les contextes de références seront choisis parce qu'elle peut être déterminée avec précision et qu'elle peut inférer plusieurs autres contextes liés à des services spécifiques. Ce choix n'exclut pas le fait que l'activité de l'utilisateur peut être inférée par le choix des variables contextuelles.

3. Modélisation du contexte

À cette étape, les méthodes de modélisation du contexte dans un SID seront analysées et un type de modélisation permettant l'application des mesures de similarités sémantiques sera adopté.

Nous adopterons, un modèle ontologique du contexte dont les avantages seront détaillés et une ontologie pourrait être proposée afin de satisfaire les définitions de contexte et de service précédentes.

4. Revue et application des mesures de similarité sémantique

Dans cette étape et après avoir adopté un type de modélisation du contexte et avoir défini les informations contextuelles ainsi que leur pertinence dans le domaine de l'informatique ubiquitaire, une revue complète des mesures de similarité sémantique dans la littérature sera faite et les avantages et inconvénients de chacune d'elles seront montrés.

Une mesure de similarité sémantique qui répond à certaines exigences sera choisie pour être appliquée dans le processus d'adaptation dynamique des services et il sera question de l'améliorer s'il y'a lieu.

5. Sensibilité au contexte et adaptation dynamique des services

À cette étape, nous estimons que les concepts principaux ont déjà été définis tels que le contexte, les informations contextuelles et les services. Nous entamons ensuite une analyse des systèmes d'adaptation existants afin de déduire le type d'adaptation dynamique à adopter.

Le Case Based Reasoning (CBR) est le processus d'adaptation qui est favorisé a priori car il représente bien notre approche d'adaptation dynamique des services qui est basée sur la comparaison de contextes en plus CBR ne demande pas un modèle complet du système, favorise un apprentissage incrémental qui n'est pas statique, ne requiert pas une phase

d'apprentissage, utilise un minimum de ressources et s'intègre facilement avec l'aspect dynamique et incertain de l'informatique diffuse.

Les points communs du CBR avec notre approche sont : les cas sauvegardés sont l'équivalent des contextes de référence, le nouveau cas est l'équivalent du contexte courant.

6. Revue et application de la prédiction contextuelle

Durant cette étape, nous aborderons le domaine de l'adaptation des services et la sensibilité au contexte proactive, prenant en compte nos choix sur la pertinence des informations contextuelles, la simplicité ainsi que la précision de leur acquisition.

0.5 Originalité des travaux proposés et contributions

Le présent travail prend son importance dans l'application des mesures de similarités sémantiques à l'adaptation dynamique de services en informatique diffuse. Ainsi, plusieurs contributions vont ressortir à la fin de ce travail à savoir :

1. Définition des contextes de référence, applications des similarités contextuelles entre contexte courant et contexte de référence pour offrir des services à un utilisateur et amélioration du cycle de raisonnement par cas ou Case Based Reasoning (CBR) par l'introduction du contexte;
2. Proposition d'une pondération des variables contextuelles de type catégoriques ;
3. Amélioration de la mesure de similarité de *Wu* et *Palmer* entre taxonomies;
4. Connaissance de l'activité d'un utilisateur à partir des capteurs du smartphone;
5. Prédiction de la localisation future d'un utilisateur à partir des informations contextuelles accessibles et du clustering spatio-temporel des données GPS et effet du bruit dans la prédiction.

0.5.1 Définition du contexte de référence

Un contexte de référence ou de base est un contexte dont les informations contextuelles (environnementale, utilisateur, système) qui le décrivent ainsi que les services à fournir à l'utilisateur ou au système sont prédéfinies. Dans plusieurs travaux le contexte de référence est l'équivalent de « Situation » qui est une interprétation sémantique d'un niveau supérieure du contexte (S. Dobson et al., 2006).

Le choix de ces contextes de référence dépendra des points suivants :

1. La probabilité *P* d'occurrence d'un contexte particulier dans une durée prédéterminée, (plus *P* est grand plus le contexte est candidat pour être considéré comme contexte de référence);
2. Les contextes de références doivent être assez dissimilaires pour que les services fournis à l'utilisateur soient de natures différentes (un seuil minimal de similarité sémantique *S* doit être déterminé);
3. Les contextes de référence ne doivent pas être choisis par rapport à l'état mental (heureux, triste,..) ou physique (soif, faim, fatigue,….) de l'utilisateur parce qu'ils changent souvent et ne peuvent être capturés avec précision.

0.5.2 Mesure de similarité entre un contexte courant et un contexte de référence

L'hétérogénéité des informations/variables contextuelles, impose l'application d'une série de mesures de similarités sémantiques partielles, qui se fait entre variables du même genre. La mesure de similarité globale entre le contexte courant et les contextes de référence sera la somme pondérée de ces mesures de similarités partielles.

Soient les deux contextes C_r (Contexte de référence) et C_c (Contexte courant), définis par *n* variables scalaires/vectorielles et *m* variables catégoriques:

$$\begin{cases} C_r = \{Vrs_1, Vrs_2, \dots\dots\dots Vrs_n, Vrc_1, Vrc_2, \dots\dots\dots\dots Vrc_m\} \\ C_c = \{Vs_1, Vs_2, \dots\dots\dots Vs_n, Vc_1, Vc_2, \dots\dots\dots\dots Vc_m\} \end{cases} \quad (0.1)$$

Où : $\mathrm{Vrs_i}$ et $\mathrm{Vrc_i}$ sont respectivement une variable scalaire/vectorielle et une variable catégorique du contexte de référence. $\mathrm{Vs_i}$ et $\mathrm{Vc_i}$ sont respectivement une variable scalaire/vectorielle et une variable catégorique du contexte courant.

Partant de l'Hypothèse « *Deux contextes sont similaires si les variables contextuelles du même genre qui les caractérisent sont similaires* », déterminer la similarité entre ces deux contextes revient donc à déterminer la similarité entre les variables du même genre des deux contextes. La similarité sémantique entre les variables contextuelles des deux contextes est :

$$sim(C_r, C_c) = \begin{cases} 1 & \text{si localisation est la même} \\ \frac{\sum_1^n ws_i \mathrm{Sim}(\mathrm{Vrs_i}, \mathrm{Vs_i}) + \sum_1^m wc_i \mathrm{Sim}(\mathrm{Vrc_i}, \mathrm{Vc_i})}{\sum_1^n ws_i + \sum_1^m wc_i} & \text{sinon} \end{cases} \quad (0.2)$$

Où : $sim(Vrs_i, Vs_i)$ est la similarité sémantique entre la i^{ieme} variable scalaire/vectorielle du contexte de référence C_r et la i^{ieme} variable scalaire/vectorielle du contexte courant C_c.
$sim(Vrc_i, Vc_i)$ est la similarité sémantique entre la i^{ieme} variable catégorique du contexte de référence C_r et la i^{ieme} variable catégorique du contexte courant C_c.
ws_i : Poids de la i^{ieme} variable scalaire/vectorielle.
wc_i : Poids de la i^{ieme} variable catégorique.

$$\sum_1^n ws_i = 1 \, , \sum_1^m wc_i = 1, \qquad sim(C_r, C_c) \in [0,1] \quad (0.3)$$

0.5.3 Application des mesures de similarité à l'adaptation dynamique de services

L'application des mesures de la similarité sémantique dans le domaine de l'informatique diffuse n'est pas très répandue malgré leur développement dans les modèles ontologiques et leurs applications dans plusieurs autres domaines.

Dans le présent travail, les mesures de similarités sémantiques sont introduites comme outil d'aide à la décision afin de proposer le service pertinent à un contexte particulier (Figure 0.2).

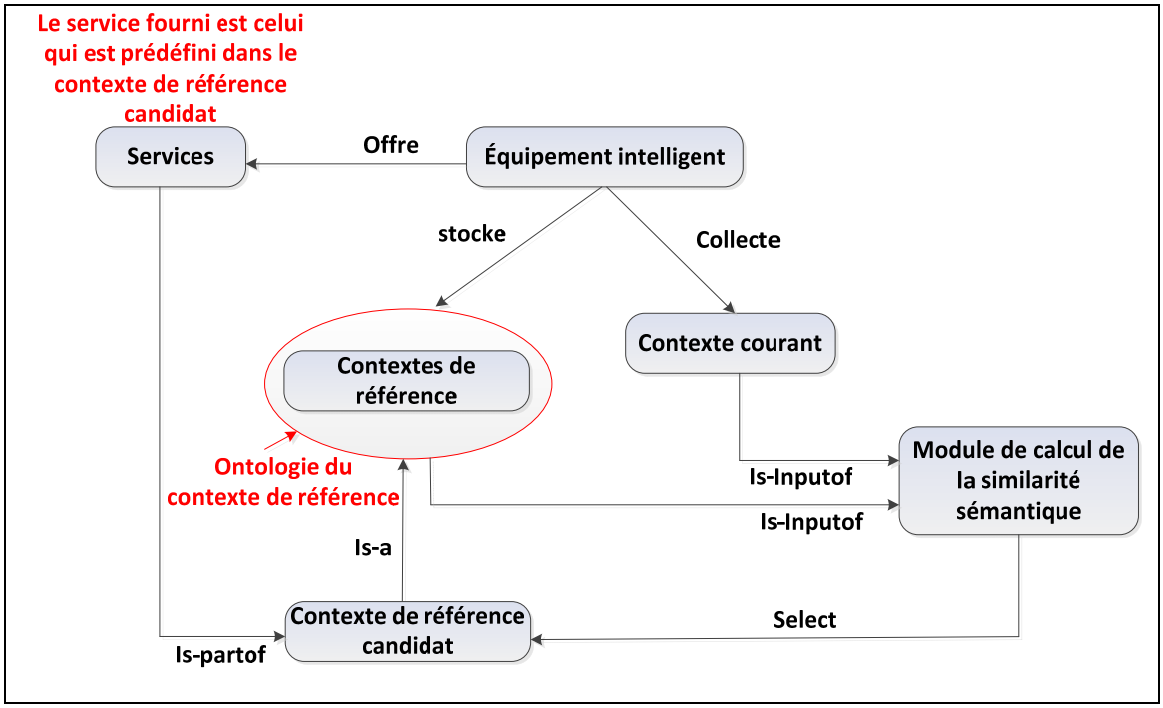

Figure 0.2 Processus d'adaptation par mesure de similarités sémantiques.

0.5.4 Proposition d'une pondération des informations contextuelles de type catégoriques

Dans la littérature, la pondération des variables de type catégorique est inversement proportionnelle au nombre de ces variables (Overlap measure, Eskin measure, IOF, OF, Smirnov, Gambaryan,…). Cette pondération est ainsi identique à toutes les variables catégoriques et ne peut identifier la contribution de chaque variable dans la mesure de la similarité sémantique. Nous proposons une pondération qui montre la contribution de chaque variable de type catégorique dans la caractérisation du contexte de référence.

0.5.5 Amélioration de la mesure de similarité de *Wu* et *Palmer* entre taxonomies

Les mesures de similarité sémantiques appliquées aux ontologies dans leur majorité ont des avantages et des inconvénients et dépendent toujours de la topologie de l'ontologie et de la conception du concepteur. Nous avons choisi une mesure de similarité sémantique simple appliquée aux ontologies et qui peut être employée sur des équipements avec ressources limitées (la mesure de *Wu* et *Palmer*) que nous avons modifiée afin que la similarité de deux concepts dans la même hiérarchie ne puisse pas être plus petite que deux concepts appartenant à différentes hiérarchies.

0.5.6 Reconnaissance de l'activité d'un utilisateur à partir des capteurs d'un dispositif électronique (smartphone)

La reconnaissance de l'activité physique d'un utilisateur de smartphone (marcher, courir, assis, debout, monter des escaliers, descendre des escaliers) est en générale faite à partir de ses capteurs inertiels (accéléromètre, gyroscope, magnétomètre) et l'extraction des caractéristiques fréquentielles et statistiques de ses signaux (FFT, Peak, médiane, SD,…).
Nous avons montré qu'il est possible de faire la reconnaissance de l'activité physique à partir de l'accéléromètre seulement et des caractéristiques -mean, standard deviation (SD) et variance (VAR).

0.5.7 Prédiction de la localisation

Prédire la localisation future d'un utilisateur à partir des informations contextuelles accessibles par GPS (localisation courante, temps, jour de la semaine et vitesse de déplacement) et du clustering spatio-temporel DBSCAN des données GPS et effet du bruit dans la prédiction.

0.6 Organisation de la thèse

Cette thèse est ainsi organisée :

1. Le premier chapitre présente une introduction, suivie par une revue de la littérature dans les domaines concernés par notre recherche : définition du contexte, modélisation du contexte, sensibilité au contexte et adaptation des services et mesures de similarité sémantique.

Les quatre chapitres qui suivent sont des travaux acceptés/publiés/soumis.

2. Le deuxième chapitre est un article qui a été publié dans le journal « International Journal on Smart Sensing and Intelligent Systems » :
Djamel Guessoum, Moeiz Miraoui, and Chakib Tadj. "Survey of semantic similarity measures in pervasive computing." International journal on smart sensing and intelligent systems 8, no. 1 (2015): 125-158.

Cet article, a été initié par le fait que la littérature sur les mesures des similarités sémantiques et leurs applications dans beaucoup de domaines est abondante et aussi par le fait que ces mesures ont été appliquées en informatique ubiquitaire mais nous n'avons trouvé aucune enquête qui rassemble toute cette littérature dans ce domaine dans un seul travail. Cet article est une enquête qui rassemble et catégorise les applications des mesures de similarité sémantique dans le domaine de l'informatique ubiquitaire.

3. Le troisième chapitre est un article qui a été publié dans le journal « Journal of Ambient Intelligence and Smart Environments » (IOS Press).
Guessoum, Djamel, Moeiz Miraoui, Atef Zaguia, and Chakib Tadj. "A measure of semantic similarity between a reference context and a current context." Journal of Ambient Intelligence and Smart Environments 8, no. 6 (2016): 697-707.

Dans cet article, nous avons présenté une application des mesures de similarité sémantique comme un mécanisme d'adaptation de services entre des contextes de références et un contexte courant. Ces mesures sont appliquées sur des informations contextuelles du même type (quantitatif et catégorique). Nous avons également proposé une pondération des informations contextuelles du type catégoriques et nous avons comparé cette pondération avec d'autres connues dans la littérature.

4. Le quatrième chapitre est un article qui a été soumis dans « International Journal of Ambient Computing and Intelligence » (IGI Global) le 24 Mars 2016:
Djamel Guessoum, Moeiz Miraoui, Chakib Tadj, "Contextual Case-Based Reasoning Applied to a Mobile Device ».

Dans cet article, nous présentons une application des mesures de similarité sémantique avec le contexte de l'utilisateur pris en considération pour l'adaptation des services. Cette application utilise la méthode de raisonnement par cas qui est une méthode de raisonnement ayant des avantages quant à son application sur des équipements à ressources limitées et dont le cycle est proche de notre cadre de recherche et où il est requis de calculer la similarité sémantique entre un nouveau cas et des cas sauvegardés du passé de l'utilisateur.

5. Le cinquième chapitre est un article qui a été publié dans « International Journal of Pervasive Computing and Communications ».
Djamel Guessoum, Moeiz Miraoui and Chakib Tadj. "Contextual location prediction using spatio-temporal clustering." International Journal of Pervasive Computing and Communications 12, no. 3 (2016): 290-309.

Dans cet article, nous présentons une application de prédiction de la localisation future d'un utilisateur à partir des informations contextuelles courantes pour permettre l'adaptation proactive des services.

Le dernier chapitre est consacré à la conclusion, le bilan et l'évaluation de notre travail de recherche ainsi que les perspectives et les travaux futurs qui peuvent en découler.

CHAPITRE 1

REVUE DE LA LITERATURE

1.1 Introduction

L'objectif de la revue de la littérature qui suit est surtout de situer les concepts essentiels de notre travail dans les travaux récents. La revue sera donc faite sur les points suivants :

- la définition du contexte;
- la sensibilité au contexte;
- l'adaptation dynamique des services en informatique diffuse;
- les mesures de similarité et l'adaptation des services dans un SID;
- l'adaptation des services dans SID avec ressources limitées.

Les travaux choisis sont essentiellement ceux des dix dernières années seulement, vu le grand nombre de travaux qui ont été fait et le développement rapide de l'informatique diffuse.

1.2 Définitions du contexte dans l'informatique diffuse

En informatique diffuse, la notion de contexte est très importante. Sa définition et sa modélisation permettent aux applications sensibles au contexte d'être plus performantes. Elles permettent aussi l'application des mesures de similarité entre les instances d'un contexte ou entre contextes différents afin d'offrir les services les plus appropriés à l'utilisateur. Les définitions du contexte dans la littérature sont aussi variées que le nombre de domaines où elles sont appliquées, pour cela, nous allons montrer les définitions du contexte les plus connues et en adopter une. Cela constitue la base de notre travail. Nous procédons par la suite au choix d'un type de modélisation du contexte selon des exigences particulières.

Nous privilégeons le modèle ontologique pour modéliser les informations contextuelles. Ce choix est supporté par la simplicité, l'expressivité ainsi que la possibilité de partage des

connaissances et particulièrement par le fait de l'existence des algorithmes d'inférence qui s'appliquent à ce type de modèle.

Nous proposerons un type de contexte qui est décrit par des informations contextuelles prédéfinies et où les services fournis à l'utilisateur sont aussi prédéfinis. Ce contexte sera appelé «contexte de référence». Ce type de contexte servira ensuite à construire une méthodologie de mesure de similarité sémantique entre n'importe quel contexte courant et un certain nombre de contextes de «référence».

Le contexte est défini dans le dictionnaire comme « l'ensemble des circonstances ou faits qui entourent un événement ou une situation particulière ». Malgré l'aspect générique de cette définition, elle reste une description qui représente l'essence des définitions proposées dans la plupart des travaux.

Avant de définir le contexte, il faudrait préciser ses caractéristiques. Ainsi, la majorité des chercheurs s'accordent pour définir le contexte selon les quatre axes suivants (Petit M., 2005) :

- il n'y a pas de contexte sans contexte : La notion de contexte doit se définir en fonction d'une finalité. Par exemple, on cherche à adapter dynamiquement les capacités interactives d'un système;
- le contexte est un espace d'information qui sert l'interprétation : la capture du contexte n'est pas une fin en soi, mais les données capturées doivent servir un objectif;
- le contexte est un espace d'informations partagé par plusieurs acteurs : ici, il s'agit de l'utilisateur et du système;
- le contexte est un espace d'information infini et évolutif : le contexte n'est pas figé, mais se construit au cours du temps.

Les deux caractéristiques essentielles qui ressortent de ce qui a précédé, sont le caractère dynamique du contexte et l'objectif de sa définition.

Les définitions du contexte présentées mettent l'accent sur ces deux aspects. En effet, (Brézillon P. et al., 1999) ont défini deux concepts liés au contexte : 1) L'ensemble des connaissances contextuelles (p.ex. : Temps, localisation) pouvant servir dans un problème de décision. Elles sont latentes et ne peuvent être utilisées sans qu'un objectif n'émerge, et 2) Le contexte qui est le produit de l'émergence d'un objectif ou d'une intention et qui utilise une grande partie des connaissances contextuelles. (Marc Dalmau et al, 2009) ont montré que l'objectif de la définition du contexte en informatique diffuse est de permettre à une application sensible au contexte de découvrir et de réagir à des modifications de situations.
Schilit, Adams en 1994, a considéré que le contexte possède trois aspects importants qui consistent en des réponses aux questions suivantes : Où es-tu ? Avec qui es-tu ? De quelles ressources disposes-tu à proximité ?

Schmidt, A. et al. (1999) ont ainsi catégorisé le contexte en six espaces : les trois premiers ont un lien avec le facteur humain : l'information qui concerne l'utilisateur (p.ex. Habits, conditions biophysiques….), l'environnement social (p.ex. La proximité avec d'autres personnes) et les tâches de l'utilisateur (p.ex. Les tâches actives de l'utilisateur, ….), les trois autres catégories sont liées à l'environnement physique : la localisation, l'infrastructure (p.ex. Les ressources, la communication), et les conditions physiques (p.ex. Le bruit, la luminosité, les conditions climatiques…..)

La définition la plus citée est celle de Dey (Dey et al., 2001) qui définit le contexte par : "toute information pouvant être utilisée pour caractériser la situation d'une entité (personne, objet physique ou informatique" . Il est clair que cette définition ressemble à celle de Schilit par le fait que le contexte est un ensemble d'informations collectées à partir d'un environnement utilisateur (personne), d'un environnement physique (objet physique) ou d'un environnement système et dont l'objectif de collection est la caractérisation de ces environnements.

Brézillon en 1999 a défini le contexte par : "Le contexte est ce qui n'intervient pas directement dans la résolution d'un problème mais contraint sa résolution". Cette définition est très générale car elle ne précise pas la nature de ce qui peut contraindre la résolution d'un

problème. Miraoui et Tadj en 2007 ont proposé une définition orientée service du contexte qui délimite les informations pertinentes qui servent à offrir les services appropriés à un utilisateur. Cette définition tient compte seulement des attentes de l'utilisateur et peut être utilisée dans beaucoup d'applications. Le contexte est défini comme « Toute information dont le changement de sa valeur déclenche un service ou change la qualité (forme) d'un service ».

(Yazid Benazzouz, 2011) a catégorisé les sources des informations contextuelles comme : les préférences de l'utilisateur, l'historique de ses habitudes, son environnement comme la température ambiante ou sa localisation géographique et son environnement système tel que le cadre matériel, les applications et réseaux dans lesquels fonctionne le système.

De tout ce qui a précédé, nous pouvons dire que le contexte est définitivement un ensemble d'informations qui caractérisent un environnement (utilisateur, environnemental, système) et que la collection de ces informations doit servir à un objectif.

La Figure 1.1 qui montre la taxonomie générale des informations contextuelles, inspirée du travail de (Lavirotte S. et al., 2005) est adoptée dans ce qui suit pour sa simplicité et sa globalité. Seules les informations pertinentes doivent être choisies pour servir notre objectif initial qui est de fournir des services appropriés à un utilisateur dans un environnement d'un système informatique diffus.

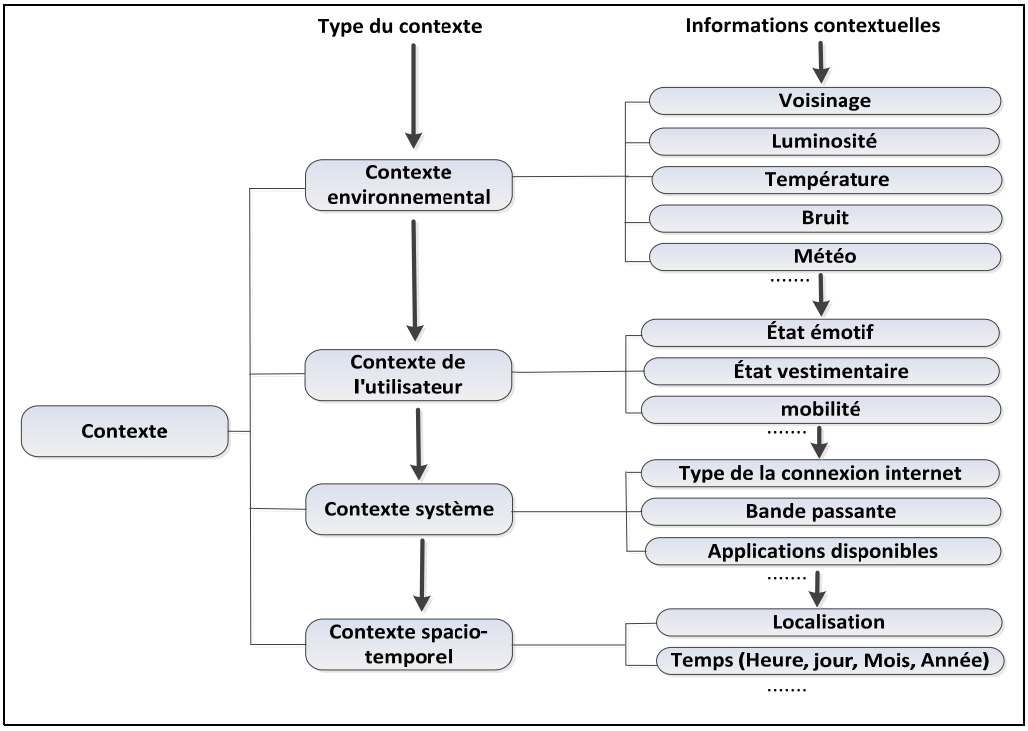

Figure 1.1 Types de contexte
Tirée de Lavirotte S. et al., (2005, p.10)

Les informations contextuelles

L'ensemble des informations qui caractérisent un contexte proviennent de plusieurs sources d'informations, capteurs physiques dans l'environnement de l'utilisateur, de l'équipement intelligent, des capteurs virtuels, internet ou même des fournisseurs des télécommunications et sont ainsi très hétérogènes. La classification des informations contextuelles a été abordée dans plusieurs travaux tels que (Gowda, K. C. et al.,1992), (Liwei Liu et al., 2010). Nous adopterons les trois classes d'informations contextuelles suivantes (Figure 1.2) :

1. Variables quantitatives, exprimées sous forme scalaire ou vectorielle, p.ex. : Température=30℃, (Latitude, Longitude, Altitude);
2. Variables quantifiables, exprimées sous forme qualitative ou ordinale, p.ex.: grand, petit, premier, …). La quantification est l'interprétation de la qualité sous forme d'une quantité ou la projection d'une variable ordinale dans un espace métrique, p.ex. : Chaud ↔ T > 30℃ et premier est projetée sur un axe linéaire ou deuxième vient juste après premier;

3. Variables catégoriques qui sont des variables qui ne peuvent pas être exprimées par le premier ou le deuxième type de variables, et ainsi elles ne sont pas quantifiables. Ce type de variables est décrit par un ensemble de caractéristiques (p.ex. : debout, assis …..).

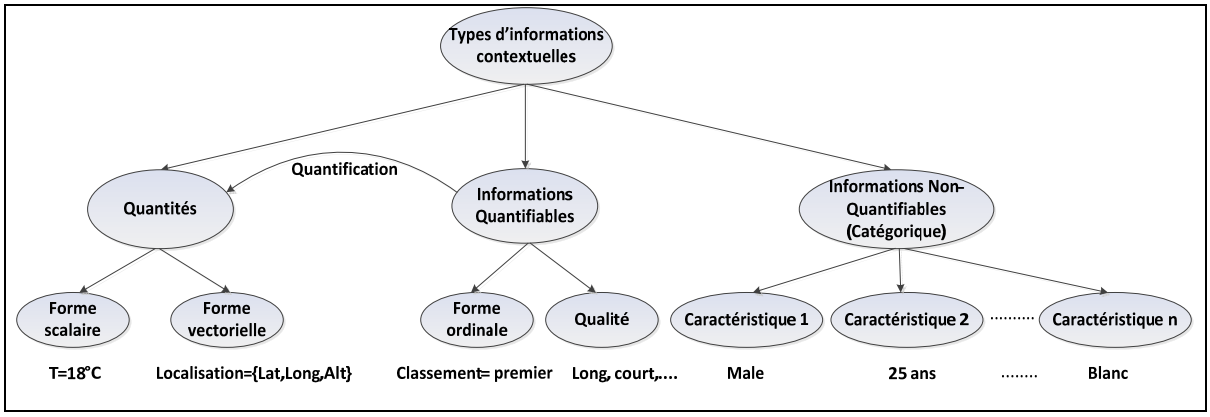

Figure 1.2 Classification des variables contextuelles

1.3 Modélisation du contexte

La modélisation du contexte est l'instrument avec lequel l'information est recueillie, organisée et présentée. Elle sert à réduire la complexité des applications sensibles au contexte, en plus de permettre la réutilisation et le partage des informations contextuelles entres les applications sensibles au contexte pour réduire le coût de la collection, de l'évaluation et de la maintenance des informations contextuelles (Bettini C. et al., 2010).

Les modèles qui existent différent par la méthode de capture des informations contextuelles, dans leur expressivité et la clarté de leurs structures et le plus important, leur support pour le raisonnement sur les informations contextuelles.

Dans la littérature, plusieurs enquêtes sur les modèles existants (Bettini C. et al., 2010), (Ferit Topcu, 2011), ont été faites. Les modèles les plus fréquents ont été décrits, évalués par rapport à une série d'exigences proposées, expliquées dans (Strang, T. & Linnhoff-Popien C., 2004) et sont présentées ci-dessous :

- composition distribuée **:** Le modèle de contexte doit s'adapter à une structure distribuée où une instance centrale gère les fonctionnalités essentielles de la structure;
- validation partielle **:** Un modèle de contexte doit pouvoir être validé partiellement afin de réduire sa complexité;
- richesse et qualité de l'information **:** Le modèle doit pouvoir s'adapter avec le fait que l'information recueillie à travers les capteurs soit variable dans le temps;
- ambiguïté et incomplétude **:** L'information recueillies est parfois ambigüe et incomplète (p.ex. : coordonnées GPS);
- niveau de formalité **:** Le modèle du contexte doit permettre le partage de l'information et sa réutilisation en spécifiant d'une façon adéquate l'information contextuelle;
- applicabilité aux environnements existants **:** Le modèle du contexte doit être réalisable dans l'infrastructure existante.

Claudio Bettini et al. (2010) ont proposé que chaque nouveau modèle du contexte doit tenir compte :

- l'hétérogénéité et la mobilité **:** Le modèle doit traiter des informations hétérogènes (sémantique, temporelles, spatiales, etc.);
- les relations et les dépendances **:** doit tenir compte des dépendances entre les informations;
- relation avec le temps **:** le modèle doit conserver un historique des informations pour être exploitées;
- l'imperfection **:** le modèle doit traiter avec les imprécisions et les ambigüités des informations du monde physique;
- raisonnement : le modèle doit pouvoir s'adapter à de nouvelles situations et même inférer des informations qui peuvent être utilisées par des couches supérieures;
- convivialité des formalismes de modélisation **:** le modèle du contexte utilise un formalisme clair et facile à être exploité par les concepteurs;
- accès aux informations du contexte **:** le modèle doit donner un accès facile aux informations du contexte pour une application sensible au contexte.

1.3.1 Approches de modélisation du contexte

Dans la littérature, plusieurs modèles ont été proposés selon le type d'application, le type d'informations et les données collectées et surtout selon une série d'exigences que l'application sensible au contexte doit satisfaire comme la convivialité du formalisme, la validation partielle et la relation du modèle avec le temps (Bettini, C. et al, 2010). Les modèles les plus représentatifs sont présentés ci-dessous :

1.3.1.1 Modèle spatial de modélisation du contexte

La plupart des modèles spatiaux organisent l'information du contexte par l'emplacement physique de l'utilisateur qui peut prendre deux formes : coordonnées symbolique et coordonnées géométriques (Ferit Topcu, 2011).

Les coordonnées symboliques qui sont des identifiants de l'emplacement de l'utilisateur (par exemple une chambre ou un salon dans une maison) et sont utilisées quand le positionnement explicite n'est pas requis. Par exemple, pour régler la luminosité dans une maison, nous avons seulement besoin d'identifier si on est au salon ou dans une chambre.

Les coordonnées géométriques sont des points dans un espace métrique, par exemple un GPS. Ils ont l'avantage de faciliter la mesure de distance. Un exemple d'utilisation des coordonnées géométriques serait de fournir des services selon l'emplacement géographique de l'utilisateur comme la recherche web.

Elles permettent un raisonnement sur la localisation des objets dans un espace ainsi que les relations spatiales entre elles à l'aide de trois types de requêtes :1) La position : extrait la position d'un objet, 2) La distance : extrait les objets qui se trouvent à une certaine distance et 3) Le voisin le plus près, qui extrait une liste d'un ou plusieurs objets voisins d'un objet (Bettini C. et al., 2010).

1.3.1.2 Modèles orientés Objet- Rôle

Ce type de modèle a été créé pour fournir un degré de formalisme suffisant pour supporter le traitement par requête (query-processing) et permettre un degré de raisonnement et d'inférences ainsi que la possibilité d'avoir une structure adéquate aux tâches analytiques et de conception de l'ingénierie logicielle (software engineering).

Un exemple de langage utilisé par ce type de modèles est le CML (Context Modelling Language). Celui-ci fournit une notation graphique qui permet de modéliser une application sensible au contexte. Il est basé sur ORM Object-Role Modelling, développé pour la modélisation conceptuelle des bases de données en plus de fournir les extensions suivantes :

- traite les informations imparfaites et les conflits entre assertions;
- gère les dépendances entre les éléments du contexte;
- exploite les informations passées et gère l'historique du contexte;
- saisit les différentes classes et sources d'information du contexte.

1.3.1.3 Modèles ontologiques du contexte

a) Définition

Une ontologie est une forme de représentation des connaissances d'un domaine particulier. Cette représentation repose sur un ensemble de classes ayant des propriétés, des relations qui lient ces classes et des individus qui peuplent ces classes ayant des valeurs.
Les ontologies ont été classifiées par rapport à leur généralité (J. Ye et al., 2007) :

- les ontologies génériques : décrivent des concepts généraux et sont indépendantes de n'importe quel domaine particulier (p.ex. espace, temps,..);
- ontologies de domaine : décrivent des concepts reliés à un domaine spécifique (p.ex. physique, biologie,…);
- les ontologies d'application : décrivent des concepts nécessaires pour une application spécifique. Elles dépendent des ontologies de domaine et des ontologies génériques.

Généralement les modèles du contexte sont réalisés pour une application spécifique et de ce fait ils ne peuvent être réutilisables dans d'autres applications et manquent d'interopérabilité. Les approches de modélisation basées sur les ontologies représentent les connaissances et les concepts ainsi que les relations entre elles (exemple : les modèles CoBrA -Common Object Request Broker Architecture- et SOCAM -Service-oriented Context-Aware Middleware-).

Le formalisme de choix pour les modèles à bases d'ontologies est le langage OWL-DL (Web Ontology Language-Description Logics) ou RDF (Resource Description Framework). La syntaxe des deux est basée sur XML.OWL-DL est un sous-langage de l'OWL qui a été développé par le Web Ontology Working Group.

Avec OWL-DL, il est possible de modéliser un domaine en définissant des classes, des individus, caractéristiques des individus et les relations entre objets (Ferit topcu, 2011).

b) Avantages de la modélisation avec les ontologies

Les ontologies fournissent une spécification formelle d'une conceptualisation partagée (Guarino N., 1998), ayant la possibilité d'être lues par les machines, et construites à partir d'un consensus d'une communauté de concepteurs. Elles représentent une source de connaissances structurée et fiable (Sánchez, D., et al., 2012). Plusieurs outils sont disponibles pour publier et pour partager les ontologies par le consortium World Wide Web $W3C$ (p.ex. le langage RDF et le langage OWL).

(Strang, T. & Linnhoff-Popien C., 2004), après l'application d'une série d'exigences ont conclu que les ontologies sont l'outil de modélisation le plus expressif et répond à la plupart de ces exigences. Selon (Korpipää P. and Mäntyjärvi J., 2003), l'ontologie doit avoir des concepts et des relations entre les concepts faciles à comprendre, flexible pour permettre l'ajout de nouveaux éléments de contexte et enfin elle doit maintenir un équilibre entre sa généricité et son expressivité afin d'inclure le maximum de types de contextes avec les détails pertinents sans être spécifique à un seul type de contexte.

Parmi les avantages des modèles ontologiques et au-delà de la clarté de la représentation, de l'organisation hiérarchique des informations et de l'interopérabilité entre plusieurs applications, citons :

- le partage des connaissances : l'unification des terminologies et des concepts permet le partage de l'information à travers les systèmes informatiques diffus;
- le support au raisonnement : par l'utilisation des algorithmes de raisonnement sur les propriétés et les relations interclasses pour inférer la consistance de l'ontologie, ainsi que de nouvelles connaissances et permettre de construire des structures taxonomiques;
- la réutilisation des connaissances : à partir des ontologies bien définies, de nouvelles ontologies peuvent être créées.

Une comparaison proposée par (Tarek C., 2007) de quelques plateformes d'applications sensibles au contexte basées sur des ontologies est présentée dans la Table 1.1.

Tableau 1.1 Comparaison des plateformes d'applications sensibles au contexte de troisième génération

	Type d'architecture	**Méthode d'acquisition**	**Modèle du contexte**	**Interprétation du contexte**	**Historique et stockage du contexte**	**Sécurité et confidentialité**
CoBra	Basée sur des agents	Module d'acquisition	Ontologies (OWL)	Moteur d'inférences et base de connaissances	Distribué	Politiques avec le langage **Rei**
CMF	Centrée sur un gestionnaire de contexte	Serveurs de ressources	Ontologies (RDF)	Service d'interprétation	Non disponible	Non disponible
SOCAM	Middleware distribué	Fournisseurs de contexte	Ontologies (OWL)	Moteur d'inférences	Disponible dans une base de données	Non disponible

1.3.1.4 Comparaison des approches de modélisation

Les modèles spatiaux du contexte permettent un raisonnement sur les relations spatiales entre objets et causent trois requêtes typiques : position, portée, plus proche voisin (Frank, A. U., 2011). CML a l'avantage de faciliter l'analyse et la conception d'applications sensibles au contexte ainsi que de fournir un support pour évaluer les informations imparfaites et historiques. Son inconvénient majeur est sa faiblesse à l'encontre des contextes ayant une structure hiérarchique ou des informations contextuelles ont plus d'importance que d'autres.
Les modèles à bases d'ontologies offrent l'avantage d'être plus expressives et réutilisables. Ils permettent le partage des connaissances. Cependant, leur inconvénient principal est la charge de calcul élevée liée au raisonnement avec OWL-DL.

1.4 La sensibilité au contexte

La sensibilité au contexte dans un système informatique diffus est l'interaction en temps réel entre une application sensible au contexte et les informations qui proviennent du contexte afin d'offrir à l'utilisateur les services appropriés. L'origine du terme « *context awarness* » ou « sensibilité au contexte » est attribuée à Schilit et Theimer qui la définissent comme «la capacité d'une application mobile et/ou un utilisateur de découvrir et de réagir à des modifications de situations» (Marc Dalmau et al, 2009). Les informations du contexte peuvent changer le comportement de l'application. La réponse de l'application peut parfois agir sur le contexte lui-même (Figure 1.3).

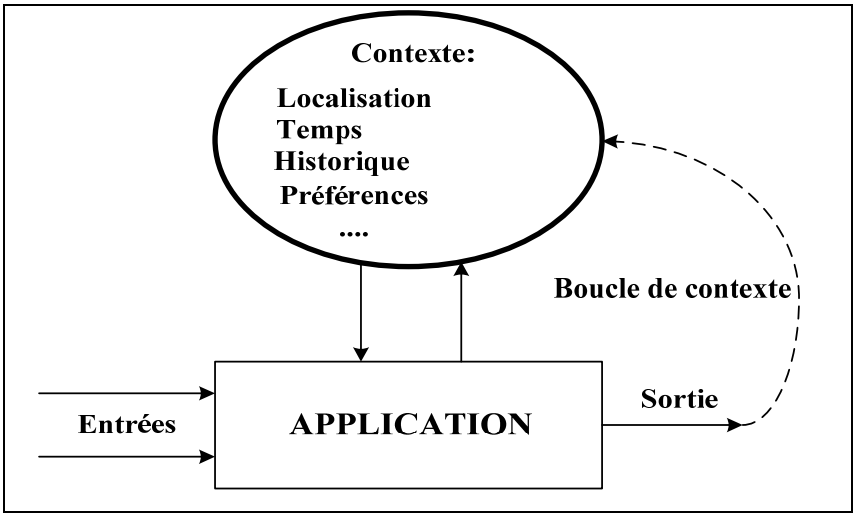

Figure 1.3 Application tenant compte du contexte
Tirée de Petit, M. (2005, p.7)

Dans cette partie, nous présentons quelques travaux qui ont traité la définition de la sensibilité au contexte. (G. Chen and Kotz D., 2000) distinguent deux états de sensibilités au contexte : 1) la sensibilité active au contexte où une application s'adapte automatiquement au contexte découvert en changeant son comportement et 2) la sensibilité passive au contexte où une application présente seulement le nouveau contexte ou le contexte mis à jour à un utilisateur intéressé ou qui fait qu'un contexte soit présent continuellement pour un retrait ultérieure. Cette définition comme toutes les autres lie la notion de sensibilité au contexte dans un SID à l'adaptation dynamique/statique d'une application sensible au contexte.

Une deuxième forme de la définition précédente est celle de (Dey, 2001). Pour ce dernier, un système est sensible au contexte s'il utilise le contexte pour fournir les informations pertinentes et/ou les services à l'utilisateur. La pertinence dépend de la tâche de l'utilisateur.
Une définition plus générale est celle de (Constantin Schmidt, 2007) qui a défini la sensibilité au contexte en général par la « capacité d'une entité à s'adapter utilement ou de réagir à un contexte ». Le fait de '*s'adapter utilement'* ne précise pas le type de services à offrir à l'utilisateur comme résultat de l'adaptation de l'application sensible au contexte.

Nous parlons de « sensibilité au contexte » dans une application lorsque celle-ci est capable d'être sensible à son contexte d'exécution (utilisateur, matériel et environnement) et à son évolution (Marc Dalmau, 2009). Cette définition est une autre forme de la définition de (Weiser M., 1991) de l'informatique ubiquitaire où elle est définie comme « Ensemble de machines qui s'adaptent à l'environnement humain au lieu de forcer les humains à s'adapter à leur environnement ».

À partir des définitions précédentes, il est clair que la sensibilité au contexte est liée au concept d'« adaptation » ou de « réaction » à un contexte ou à son changement.

1.4.1 Sensibilité au contexte et adaptation des services

Les progrès considérables de l'informatique diffuse depuis son apparition dans les années 90 du 20ieme siècle a permis à la recherche dans ce domaine de se diversifier et plusieurs disciplines sont apparues dont la sensibilité au contexte.

Un système sensible au contexte est un système qui s'adapte dynamiquement aux besoins de l'utilisateur pour offrir les services appropriés. L'origine du terme « context awareness » ou «Sensibilité au contexte» est attribuée à Schilit et Theimer qui prétendent que la sensibilité au contexte est «la capacité d'une application mobile et/ou un utilisateur de découvrir et réagir à des modifications de situations» (Marc Dalmau et al., 2009).

La sensibilité au contexte est centrée sur le concept de contexte qui a été définit de plusieurs façons suivant le domaine d'application. Pour exploiter les informations du contexte dans l'implémentation d'applications transparente à l'utilisateur, plusieurs modèles du contexte sont apparus et chaque modèle devait satisfaire à des exigences prédéfinies par les chercheurs afin qu'il puisse être validé. Les modèles les plus récents sont ceux qui adoptent les ontologies comme approche de modélisation parce qu'ils permettent au système de faire des inférences à partir des informations collectées pour déduire de nouvelles informations qui ne

peuvent être collectées par les capteurs physiques ou logiques et ainsi vont enrichir la description du contexte pour fournir des services plus appropriés à l'utilisateur.

Les technologies émergentes comme les télécommunications et les réseaux d'ordinateurs plus performants a permis l'implémentation de nouvelles architectures dans les systèmes sensibles au contexte, ainsi du système autonome ou le traitement de l'information se fait au niveau d'une seule entité locale, sont apparus les architectures distribuées avec l'utilisation des réseaux ad-hoc dans la distribution du traitement de l'information et les architectures centralisées qui permettent d'avoir une entité centrale ayant une puissance considérable.

a) Traitement de l'information dans un système sensible au contexte et architecture d'un système sensible au contexte

Dans la littérature il y'a plusieurs architectures de systèmes sensibles au contexte formées de composantes responsables de la représentation, de la gestion, du raisonnement et de l'analyse de l'information du contexte. Malgré la multitude de ces architectures, toutes utilisent le même processus de traitement de l'information illustre sur la Figure 1.4.

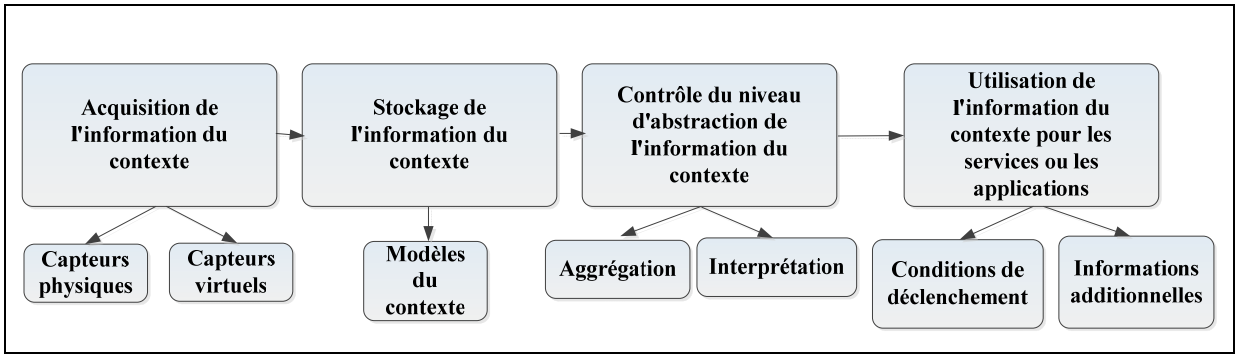

Figure 1.4 Processus général des systèmes sensibles au contexte
Tirée de Sangkeun, Lee et al., (2011, p.2)

La première étape est l'acquisition de l'information du monde physique à travers des sources variées telles que les capteurs. Le système stocke ensuite ces informations pour être utilisées par le modèle de représentation choisi. Le système contrôle ensuite le degré d'abstraction de

l'information par interprétation ou agrégation des données du contexte. Finalement, le système utilise les données abstraites du contexte dans des applications sensibles au contexte.

Acquisition et stockage des informations du contexte

Les types d'informations à acquérir du contexte sont diverses et peuvent être acquises de diverses manières telles que les capteurs physique (exemple –Contexte/Capteur- : Lumiere/photodiode, Audio/Microphone, Mouvement/ Accéléromètres, Position/GPS,……..) **(**Baldauf et al., 2007), les capteurs virtuels, ou de la fusion des deux.

L'information acquise du contexte est stockée sous une forme qui dépend du modèle du contexte (p.ex.: attribut-valeur, information ontologique) (Table 1.2).

Tableau 1.2 Exemple de modèles de systèmes sensibles au contexte
Tirée de Sangkeun, Lee et al., (2011, p.4)

Modèle du contexte	Exemples
Modèle Attribut-Valeur	Context-Toolkit
Modèle basé sur la logique	Approche de McCarthy
Modèle base sur Mark-up programmes	CC/PP,UAProf, CSCP,GPM
Modèle orientés objets	Hydrogen
Modèle graphique	Modèle spatial
Modèle ontologique	SOCAM, CoBra, CoCA

Control du degré d'abstraction du contexte

Le système sensible au contexte n'utilise pas directement les informations acquises du monde physique mais par contre il doit les transformer sous forme de données utilisables. La plupart de ces systèmes réalisent l'abstraction du contexte par deux méthodes ; l'interprétation et l'agrégation.

- l'interprétation du contexte : c'est une méthode qui interprète les données et les transforment sous forme de données sémantiques de niveau supérieur tel que : une donnée GPS est interprétée comme un nom de rue;
- l'agrégation des données : c'est le fait de réduire beaucoup de données en quelques mots clés de haut niveau tel que convertir une série de lecture de températures par des mots clés comme « chaud » et « froid ».

Utilisation de l'information du contexte pour des services ou des applications

Les informations acquises du contexte peuvent être utilisées comme condition de déclenchement d'actions quand le contexte satisfait une situation spécifique ou comme information additionnelle pour améliorer la qualité d'un service fourni.

1.4.2 Première génération de systèmes sensibles au contexte

Au début des années 90, les systèmes sensibles au contexte avaient un modèle du type spatial où l'information partagée était reliée à l'emplacement de l'utilisateur (p.ex.: service de guide touristique). Citons quelques exemples de systèmes de première génération : 1- The Active Badge Location System : c'est un système sensible à l'emplacement de l'utilisateur et le service fourni dépend du l'emplacement de l'utilisateur, 2- Cyberguide : Ce système sert comme guide pour des utilisateurs pendant une visite touristique dépendamment de leur emplacement.

1.4.3 Deuxième génération de systèmes sensibles au contexte

La seconde génération des systèmes sensibles au contexte commence à utiliser des informations plus variées et apparait des architectures plus complexes qui permettent de fournir plus de services. Exemple de deux systèmes représentatif de cette génération: 1- Context-Toolkit qui a une structure distribuée avec un modèle qui utilise des données du type attribut-valeur et stocke les informations du contexte avec leur historique. Il fournit un service de base de protection de la vie privée des utilisateurs et 2- Hydrogen qui est composé

de trois couches : couche d'adaptation, couche de gestion et couche application. Ces trois couches sont présentes dans tous les équipements du contexte et ainsi la présence d'un serveur central n'est pas requise.

1.4.4 Troisième génération de systèmes sensibles au contexte

L'apparition du langage de modélisation sémantique OWL en 2003 et son adoption par plusieurs applications sensibles au contexte comme outil de modélisation a conduit aux systèmes de troisième génération d'opter pour les ontologies comme approche de modélisation. Exemple de de systèmes de la troisième génération :

CoBrA : Context Broker Architecture (CoBrA) (Figure 1.5) est une architecture orientée agents pour les systèmes sensibles au contexte dans des environnements intelligents. le « *context broker* » est l'agent central de cette architecture et est responsable de : a) Fournir un modèle centralisé du contexte, b) acquérir les informations du contexte, c) assurer l'interprétation de l'information contextuelle, d) résoudre les conflits d'interprétation des informations contextuelles et e) protéger la confidentialité des utilisateurs (Chen H., 2005).

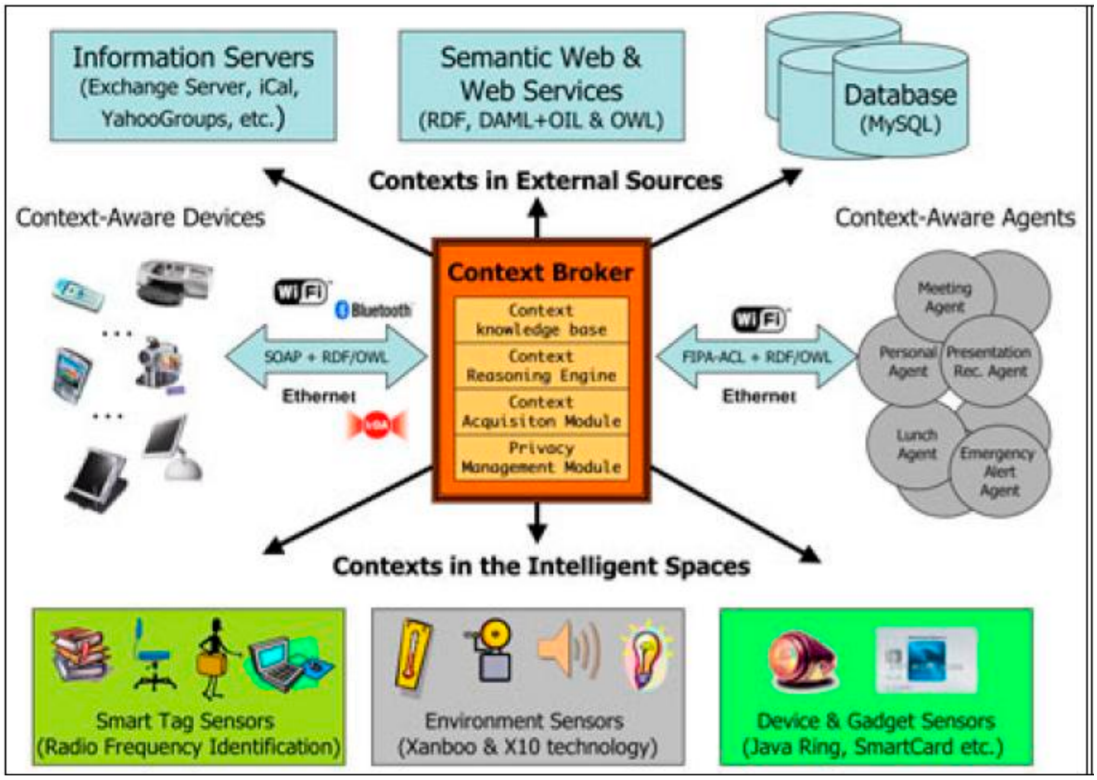

Figure 1.5 Architecture CoBra
Tirée de Chen H. (2005, p.5)

SOCAM : C'est un projet qui vise à assurer le développement et le prototypage rapide de services sensibles au contexte dans des environnements intelligents. Ce projet propose un middleware distribué qui convertit les divers espaces physiques d'où le contexte est capturé en un espace sémantique où le contexte peut être partagé et fourni à des services sensibles au contexte. Ce middleware se base sur des ontologies pour modéliser le contexte. L'architecture de SOCAM comprend les composants de la Figure. 1.6 et qui sont :

- un fournisseur de contexte qui collecte les informations contextuelles, ensuite il les transforme en des représentations OWL;
- un interpréteur de contexte qui fournit les services de raisonnement logiques sur les représentations OWL;
- la base de données du contexte qui stocke les informations décrivant l'environnement de l'application;

- les services sensibles au contexte utilisent les informations stockées dans la base de données pour modifier leur comportement selon le contexte courant;
- le service de localisation de services qui sert aux applications et aux utilisateurs de localiser les interpréteurs de contexte (Gu, T. et al., 2004).

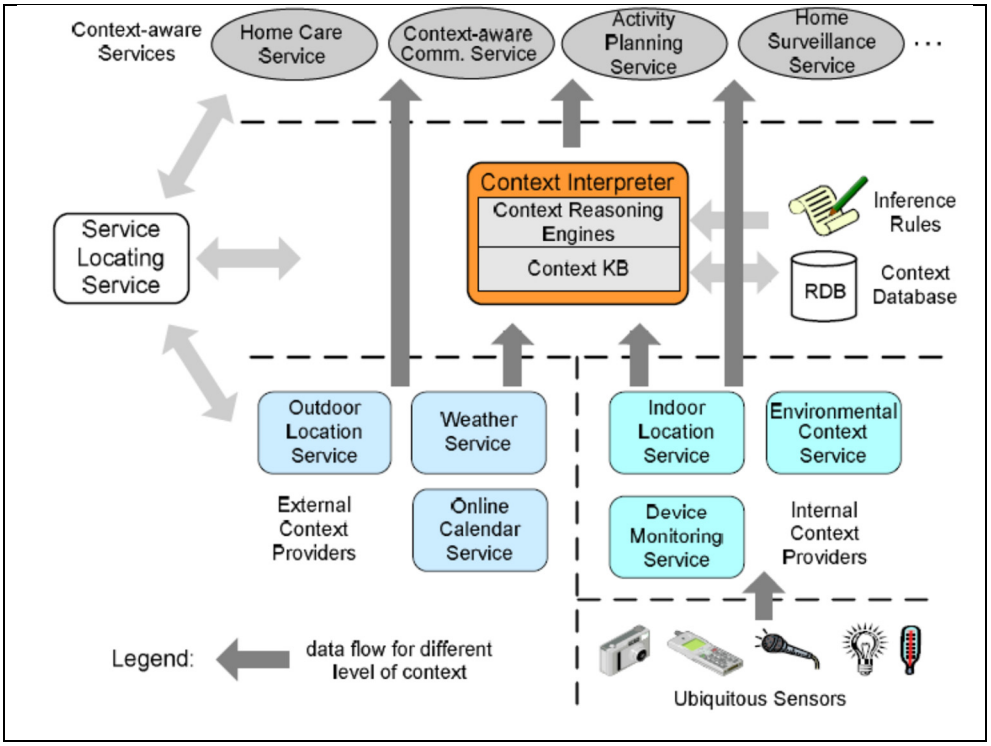

Figure 1.6 Architecture de la plateforme SOCAM
Tirée de Gu, T., (2004, p.5)

Context Fabric : Le context management framework (CMF) est une plateforme analogue au context toolkit de *Dey*. Elle est composée de quatre entités principales : Resource Server (capture de contexte), Context Recognition Service (interprétation du contexte), Context Manager (dissémination du contexte) et application Figure 1.7. Comme la plateforme CoBrA, le CMF se base sur un gestionnaire de contexte centralisé qui communique avec tous les autres modules de la plateforme. En effet, dans CMF, le gestionnaire de contexte récupère les informations contextuelles à l'aide du module Ressource Server. Ensuite, il les interprète en utilisant le module Context Recognition Service. Enfin, il les diffuse à l'application. CMF utilise la logique floue pour acquérir les informations du contexte de haut niveau afin d'enrichir le contexte et la description de l'environnement de l'utilisateur (Tarek C., 2007).

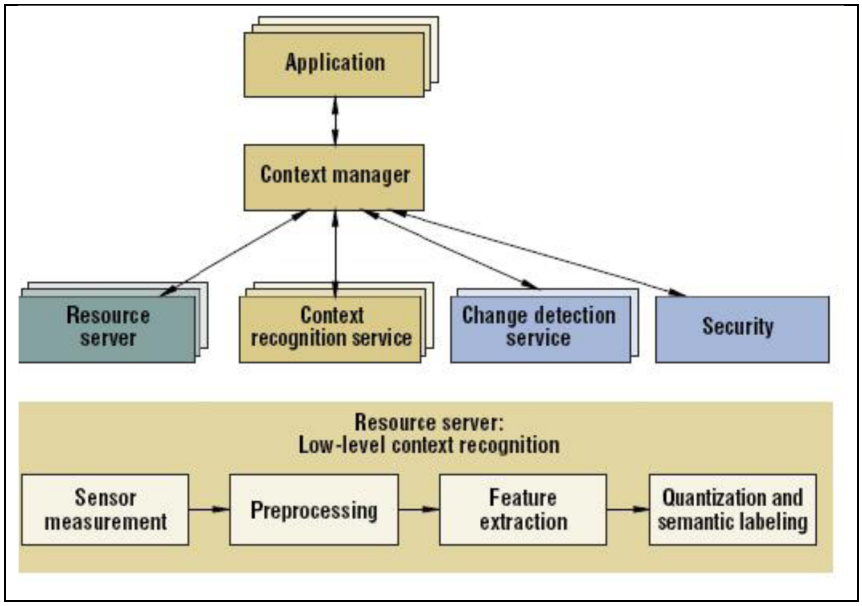

Figure 1.7 Architecture de la plateforme CMF
Tirée de Tarek C. (2007, p.42)

1.5 Adaptation dynamique de services dans l'informatique diffuse

L'environnement d'un SID est centré sur les besoins ainsi que les préférences de l'utilisateur qui est amené à utiliser différents équipements « intelligents » tels que les téléphones portables, les tablettes, les PC ou autres, dans des situations « contextes » différents. Dans un tel environnement, l'utilisateur s'attend à recevoir les services appropriés qui répondent à ses besoins sans son intervention, à chaque fois que le contexte courant change de forme. Ces services doivent être en phase avec ses préférences personnelles ainsi qu'avec les exigences de son environnement. Pour répondre à ces exigences, les applications/équipements dans un tel environnement doivent avoir la particularité d'être « sensibles au contexte » ayant un mécanisme de découverte, de raisonnement et de réaction aux informations contextuelles courantes. L'aspect réactif de l'application/équipement sensible au contexte sans l'implication directe de l'utilisateur définit l'adaptation dynamique dans l'environnement d'un SID.

L'interaction et l'adaptation transparente de l'environnement avec l'utilisateur est basée sur le concept de contexte et la conception d'applications sensibles à ce contexte et à ses

changements. Les informations du contexte sont acquises à partir de capteurs physiques ou virtuels. Elles sont stockées et mises sous une forme adéquate au modèle du contexte ensuite elles sont utilisées par des niveaux d'abstraction supérieurs pour produire les actions requises adaptées à l'utilisateur.

1.5.1 Définitions

L'adaptation dans le domaine informatique est définie dans Le grand dictionnaire terminologique comme « Opération qui consiste à apporter des modifications à un logiciel ou à un système informatique, à la fin de son développement, dans le but d'améliorer ses performances dans un contexte précis d'utilisation ».

Plusieurs définitions sont proposées dans la littérature. Certains auteurs ont essayé de catégoriser les types ainsi que les mécanismes d'adaptation dynamique. La plus générique reste celle proposée par (Efstratiou C., 2004) qui généralise le concept d'adaptation d'un équipement mobile et d'une application sensible au contexte dans un système diffus en postulant qu'une application ou un système est dit adaptatif quand il change son comportement en réponse à un changement de contexte (ce changement peut être dans le contexte ou dans les ressources de l'équipement). (Mohamed Zouari, 2011) avait défini l'adaptation dynamique d'une application sensible au contexte par son aptitude à changer son comportement en cours d'exécution en fonction des fluctuations de son environnement et des changements des exigences des utilisateurs.

Une autre approche a été adoptée dans beaucoup d'autres travaux tels que celui de (Jérôme Simonin and Carbonell N., 2007), qui ont catégorisé l'adaptation dynamique de services selon la finalité de l'adaptation et ont distingué deux types d'adaptation : l'adaptation au profil utilisateur et l'adaptation à l'environnement. Cette approche implique le contexte utilisateur ainsi que l'environnement comme sources d'informations pour une adaptation appropriée des services. Dans les travaux qui vont suivre, les services seront plus explicites. Nous citerons le travail de (Daniela Nicklas and Karen Henricksen, 2008) qui ont catégorisé

l'adaptation des applications sensibles au contexte en quatre classes : 1) la sélection des informations et des services, 2) la présentation des informations et des services, 3) l'exécution automatique d'un service pour un utilisateur et 4) marquage d'un contexte avec une information pour un retrait ultérieur. (Moeiz Miraoui et al, 2009) ont catégorisé l'adaptation en quatre classes: 1) adaptation de contenu, 2) adaptation de comportement, 3) présentation ou adaptation d'interface et 4) adaptation logicielle. Dans la même logique, (Yazid Benazzouz, 2011) a catégorisé l'adaptation en trois classes à savoir, la personnalisation, la recommandation et la reconfiguration de services.

La personnalisation des services est liée directement aux préférences de l'utilisateur et a comme source d'information contextuelle l'environnement de l'utilisateur tel que la température ambiante ou la localisation géographique. La recommandation est une forme particulière de la personnalisation. Elle utilise ses préférences stockées (son historique) pour recommander les services les plus satisfaisants au goût de l'utilisateur. La reconfiguration est finalement une autre forme d'adaptation qui prend en compte l'environnement système par exemple libérer l'espace mémoire pour une application qui en a besoin. À noter que la reconfiguration ne prend pas en compte l'environnement de l'utilisateur.

Selon (Yazid Benazzouz, 2011) les deux seules sources pour l'adaptation des services dans un système informatique diffus sont « l'environnement système et l'environnement utilisateur » (Figure 1.8). L'environnement utilisateur désigne les conditions de vie de l'utilisateur et c'est l'ensemble des informations contextuelles reliées à son environnement physique telles que : la température ambiante, la date, la nuit/jour, etc.L'environnement système est le cadre matériel, les services logiciels, applications et réseaux dans lesquels fonctionne le système.

Dans les catégorisations précédentes, nous pouvons noter que les mêmes concepts sont présents ; seules les délimitations des catégories diffèrent.

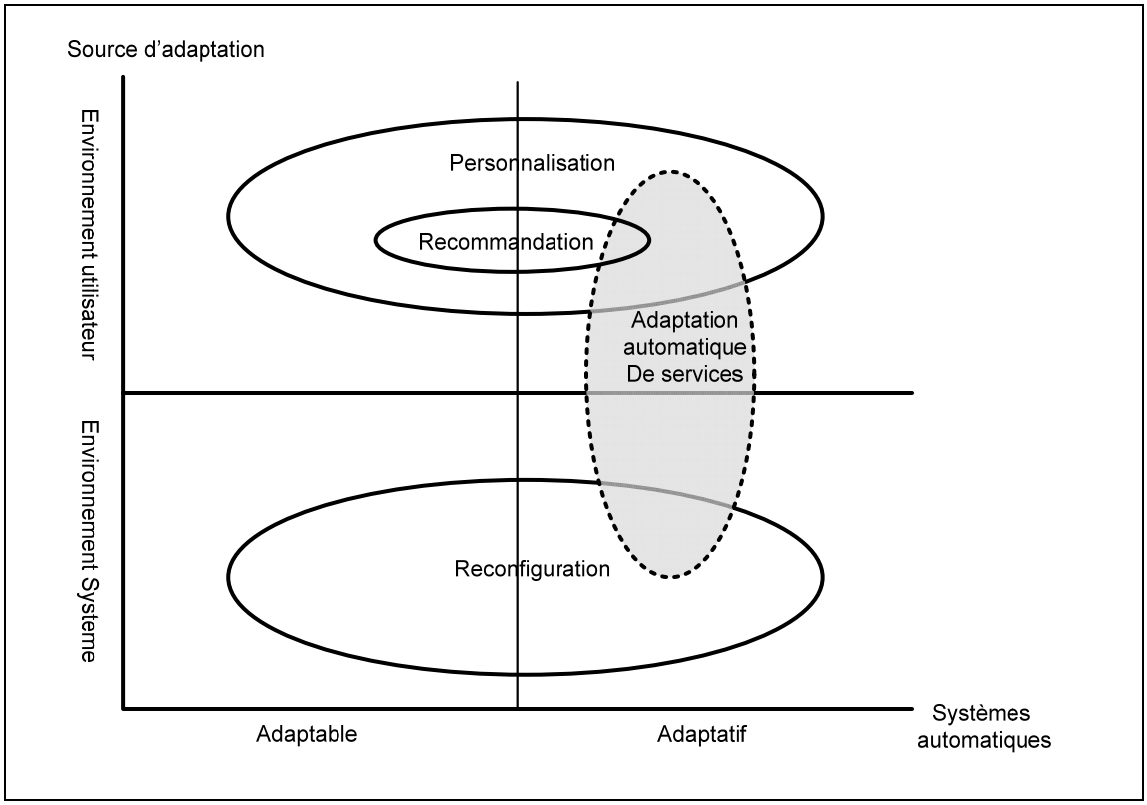

Figure 1.8 Relation entre les concepts d'adaptation
Tirée de Yazid, Benazzouz, (2011, p.15)

1.5.2 Adaptation dynamique vs. Adaptation statique

L'adaptation de services dans un système informatique diffus est classée dynamique ou statique selon l'implication de l'utilisateur ou non au processus d'adaptation. Ainsi (Jérôme Simonin and Carbonell N., 2007) ont classé les systèmes diffus en deux classes à savoir : les systèmes adaptables et les systèmes adaptatifs (Figure 1.9).

Le système est dit adaptable quand il donne la possibilité à l'utilisateur de changer certains de ses paramètres pour satisfaire ses préférences. Un système est dit adaptatif quand il s'adapte de façon automatique au contexte courant de l'utilisateur.

Le premier cas est initié par l'utilisateur et est en général basé sur une caractérisation préétablie des préférences de l'utilisateur. Il est ainsi statique. Par contre, dans le deuxième

cas, l'adaptation est dynamique et se fait durant l'interaction de l'utilisateur avec l'environnement d'une manière totalement transparente, contrôlée par le système.

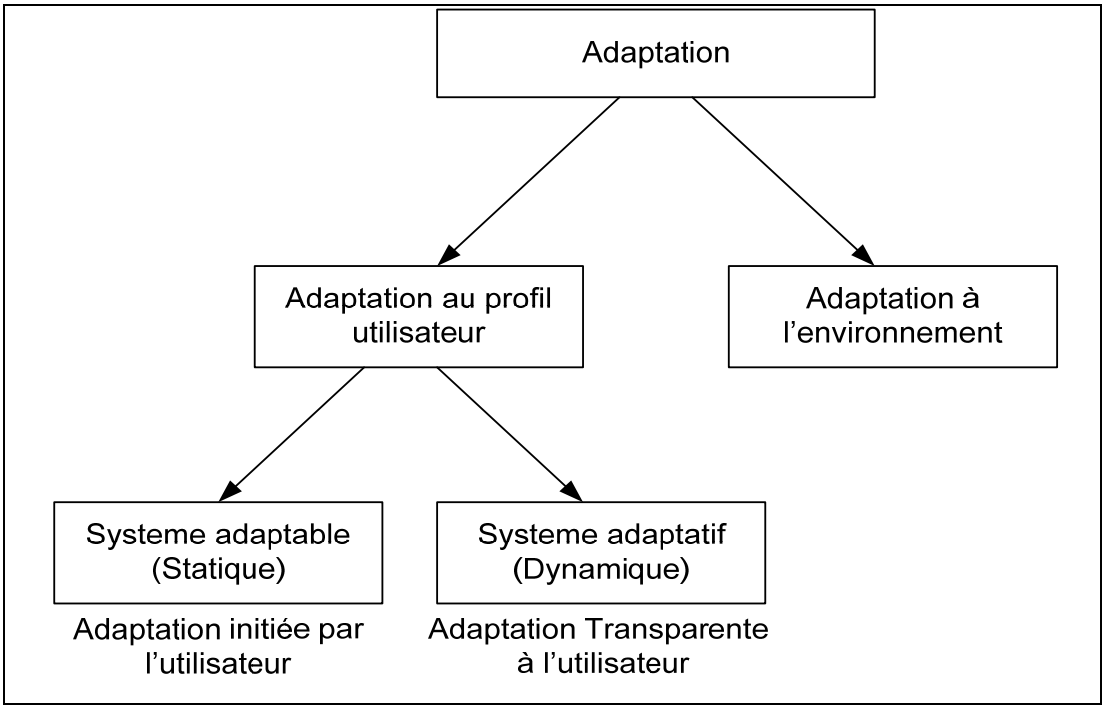

Figure 1.9 Adaptation dynamique vs. Statique

1.6 Les mesures de similarité en informatique diffuse

1.6.1 La similarité

La similarité est le fait d'évaluer les caractéristiques communes entre deux ou plusieurs objets/concepts, son équivalent quand il s'agit de comparer deux quantités est la distance.
La mesure de similarité ou de distance sont utilisées dans beaucoup de domaines aussi variés que la biologie, la chimie, la psychologie ou la théorie de l'information…..

Parmi leurs applications on peut citer la classification et le regroupement d'objets, la récupération d'informations, la recommandation de services, le Web sémantique (p.ex.: les interfaces web sensibles au contexte et la recherche intelligente de documents).

En informatique, la similarité est l'évaluation des ressemblances dans un ensemble de données à analyser permettant de quantifier ces ressemblances dans l'intervalle [0,1]. Ainsi il serait possible de les ordonner, de les hiérarchiser ou d'en extraire des invariants. Généralement, le calcul de la similarité intervient dans trois types de traitement des données à savoir : la classification, l'identification et la caractérisation (Gilles Bisson, 2000).

La classification vise à structurer les données dans un ensemble hétérogène en fonction de leur ressemblance. L'identification a pour but de connaître la classe à laquelle un objet inconnu est susceptible d'appartenir. Le processus de caractérisation permet la représentation explicite des informations communes à un ensemble de données. Plusieurs mesures de similarité/distance ont été proposées dépendamment du domaine d'application de la mesure ainsi que du type de représentation des objets. (Sung-Hyuk Cha, 2007).

En informatique diffuse, où la notion de contexte est très importante, la mesure de similarité/distance représente un outil d'évaluation de la ressemblance entre les instances d'un contexte permettant d'offrir les services les plus appropriés à l'utilisateur. Pour cela, plusieurs types de mesure de similarités sont proposés dans la littérature dépendamment du type du modèle du contexte et de la représentation des objets qui le décrivent. Pour les modèles en couches, où les données du monde réel sont exploitées par d'autres couches supérieures pour générer de nouvelles connaissances, tels que les modèles basés sur les ontologies, la mesure de similarité est du type sémantique. Si le modèle du contexte est du type objet-attribut, la mesure de la similarité est la distance entre ces deux objets/quantités.
La mesure de similarité entre objets quantifiables tels que la température ou les coordonnées géographiques, est évaluée par les mesures de similarité de type vectoriel. Quant aux objets du type catégorique (non-quantifiables) tels que l'activité de l'utilisateur ou son humeur, les mesures de similarité sémantique sont appliquées.

1.6.2 Les mesures de similarité sémantique

La mesure de la similarité sémantique a pour rôle de quantifier les caractéristiques *intrinsèques* communes entre deux objets. Une caractéristique est intrinsèque à un objet quand elle définit la nature de l'objet lui-même et qui ne peut en être séparée.

Les mesures de similarité sémantique sont très importantes dans des domaines comme le traitement du langage naturel (NLP), la bio-informatique, le Web services (Recherche web, interaction des interfaces utilisateurs) et la recommandation de services et dans beaucoup d'autres domaines. Parmi les applications les plus connues nous trouvons la recherche de l'information (Information Retrieval), l'extraction de l'information (Information Extraction) et autres systèmes de traitement des connaissances. Par exemple dans un système de recherche de l'information, la mesure de similarité sémantique sert à déterminer et ordonner les résultats d'une requête de document par rapport à un ensemble de termes d'une requête.

1.6.2.1 Les mesures de similarité sémantique appliquées aux ontologies

Les mesures de similarités sémantiques les plus développées ces dernières années, qui se basent sur la représentation ontologique des connaissances et spécialement sous sa forme taxonomique sont (Sánchez, D et al, 2012) (Thabet Slimani et al., 2007) (Figure 1.10):

1. Approches basées sur le comptage d'arcs, où plus la distance entre deux nœuds est courte et plus ils sont plus semblables;
2. Approches basées sur les nœuds (le contenu informationnel) où la similarité entre les concepts est mesurée par le degré de partage de l'information;
3. Approches hybrides qui combinent entre les deux approches;
4. Similarité sémantique basée sur les caractéristiques des concepts;
5. Approches basées sur l'espace vectoriel.

Les mesures de similarités sémantiques basées sur le comptage d'arc ont été introduites par (Rada et al., 1989). Elles s'appliquent aux ontologies ayant des relations entre concepts du type taxonomique (is-a). L'idée de base de ces mesures est « Plus le nombre d'arcs qui sépare

deux concepts est petit, plus ils sont similaires ». Les mesures de similarités basées sur le contenu informationnel du concept commun qui subsume les deux concepts à comparer, ont été introduites la première fois par (Resnik, 1995). Le contenu informationnel d'un concept est sa probabilité d'occurrence dans un corpus comme WordNet[1], où plus l'occurrence du concept est grande, moins il a de contenu informationnel.

Finalement, les mesures de similarité sémantiques fondées sur les caractéristiques des concepts reposent sur le modèle de similarité de (Tversky, A., 1977), où deux concepts sont plus similaires si : 1) ils ont plus de caractéristiques communes et 2) moins de caractéristiques non-communes.

Ces mesures seront détaillées dans les sections suivantes.

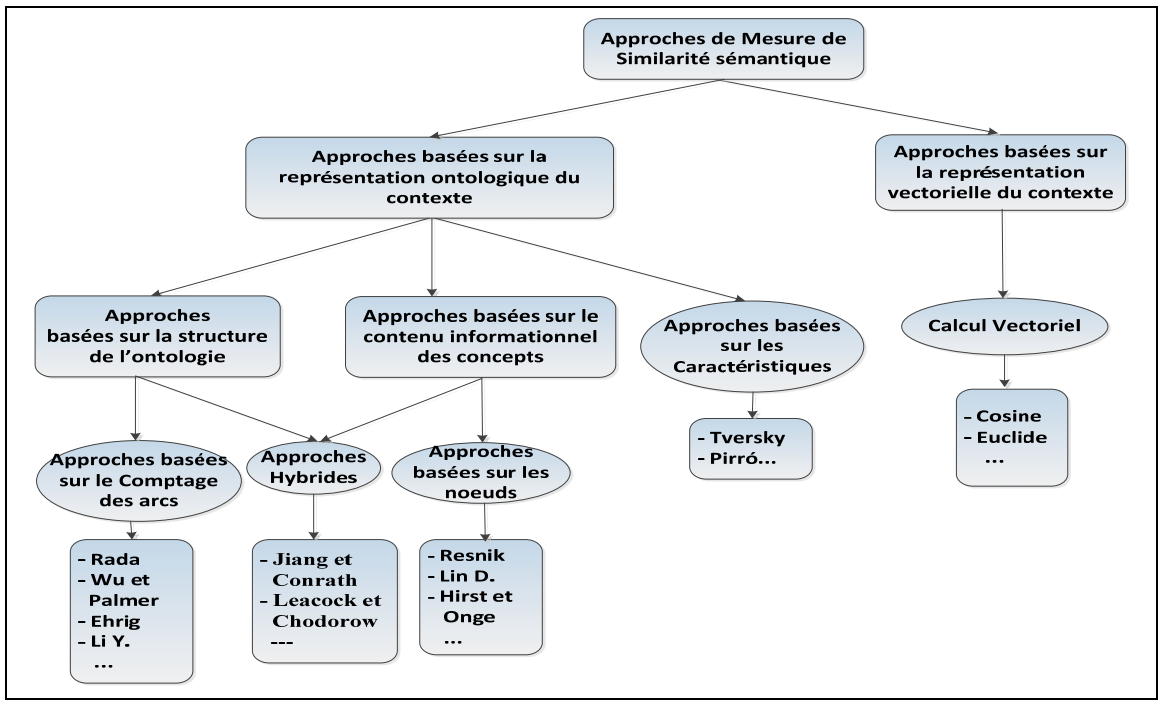

Figure 1.10 Approches de mesure de similarité sémantique

[1] WordNet: Une base de données qui organise plus de 100.000 mots anglais ou concepts, les groupe sous forme de synsets et fournit des définitions générales ainsi que les relations sémantiques entre ces concepts.

a) Approches basées sur le comptage des arcs

Avec cette approche, les mesures de similarité sémantique entre objets/concepts sont basées sur le comptage du nombre d'arcs qui sépare les deux objets/concepts d'une ontologie en se servant de sa structure hiérarchique.

L'application de ce type de mesure de similarité sémantique entre deux concepts C1 et C2, C1= Homme (nœuds 2) et C2=Femme (nœuds 3) de l'ontologie de la Figure 1.11, où les nœuds sont les concepts et les arcs sont les relations du type « is-a » entre les concepts, revient à compter le nombre d'arcs qui aboutissent au subsumant commun LCS=Humain, des deux concepts. Le subsumant commun est un concept d'un niveau supérieur qui relie les deux concepts C1 et C2.

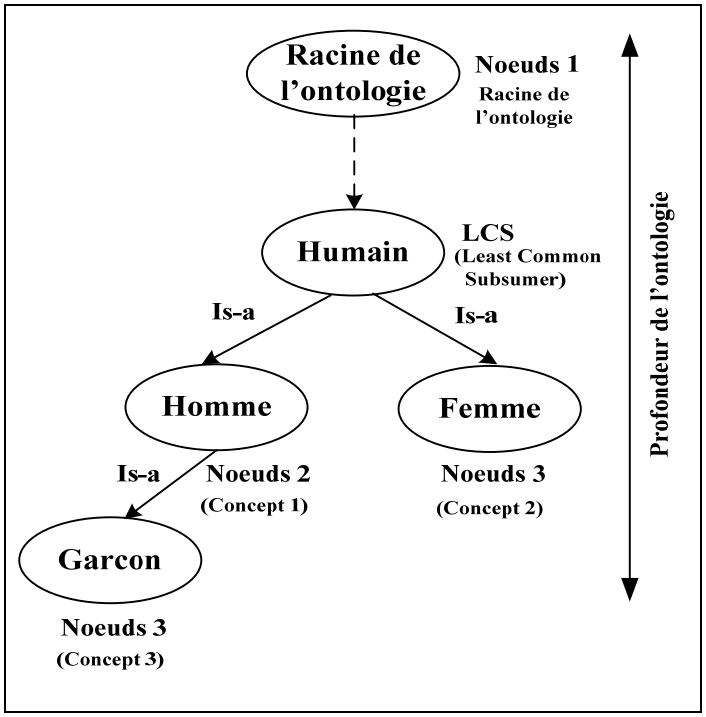

Figure 1.11 Exemple de structure taxonomique

La similarité sémantique entre C1 et C2 dans ce cas est donnée par :

$$dis(C1, C2) = \min(chemin(C1, C2))=2 \quad (1.1)$$

$$sim(C1, C2)= 1/\, dis(C1, C2) \quad (1.2)$$

Où : $sim(C_1, C_2)$ est la mesure de la similarité entre C_1 et C_2.
Il est clair que si nous voulons mesurer la similarité sémantique entre C2 et C3, nous aurons une similarité moindre, vu que le nombre d'arcs qui sépare C2 et C3 est plus grand.

Cette mesure sera acceptable si le contexte de la mesure est «genre de l'humain». Elle sera erronée si le contexte de la mesure est « caractéristiques de l'humain». En plus, nous pouvions représenter l'ontologie précédente sous une autre forme, ce qui aura donné de nouvelles valeurs de similarité sémantique. Donc cette mesure ne dépend pas du contexte et est très liée à la structure de l'ontologie qui peut varier d'un concepteur à un autre.

Parmi les travaux qui utilisent cette approche nous trouvons :

Mesure de Rada

Elle est fondée sur le fait qu'on peut calculer la similarité sémantique entre deux concepts dans une structure hiérarchique (ontologie) ayant des liens du type «is-a» par le calcul du chemin le plus court entre ces concepts.

Mesure de Wu et Palmer

Plusieurs variantes basées sur la mesure de Rada ont été proposées pour améliorer certains de ses aspects, comme celle de *Wu* et *Palmer* (Wu & Palmer, 1994), qui ont considéré la profondeur de l'ontologie dans la mesure, car plus deux concepts sont spécifiques (dans des niveaux plus bas de l'ontologie) plus ils seront plus similaires et vice versa. Cette mesure est donnée par :

$$sim(C1, C2) = \frac{2 \times P}{N_1 + N_2 + 2 \times P} \quad (1.3)$$

Où : N_1 est le nombre d'arcs (is-a) entre le concept C1 et le subsumant commun LCS
N_2 est le nombre d'arcs (is-a) entre le concept C2 et le subsumant commun LCS
P est le nombre d'arcs (is-a) entre le concept subsumant LCS et la racine de l'ontologie.

Cette mesure a aussi des lacunes (Thabet Slimani et al., 2007), où on peut trouver une similarité plus grande entre un concept et son voisinage plus qu'avec un concept fils et ont proposé une mesure qui pénalise des concepts éloignés et qui ne sont pas dans la même hiérarchie.

Plusieurs autres mesures ont ensuite été introduites comme celles de (Leacock & Chodorow, 1998), (Li, Y. et al., 2003), où chacune essaye de faire des ajustements sur un aspect particulier de la mesure de *Wu* et *Palmer*.

En conclusion, ce type de mesures est simple à implémenter, mais se limite aux ontologies avec des relations taxonomiques simples « is-a », ne prend pas le contexte en considération et peut donner des mesures de similarité sémantique erronées.
La mesure de *Wu* et *Palmer* appliquée à une ontologie *O* (Figure 1.12), constituée d'un ensemble de nœuds (*X, Y,….*) et un nœud racine *R,* définit la similarité entre les nœuds (concepts) *X* et *Y* de l'ontologie comme étant :

$$sim(X,Y) = \frac{2\times N}{N_1+N_2} \qquad (1.4)$$

Où N_1 et N_2 sont les distances qui séparent les éléments X et Y du nœud racine et *N* la distance du nœud (concept) commun au nœud racine *R*.

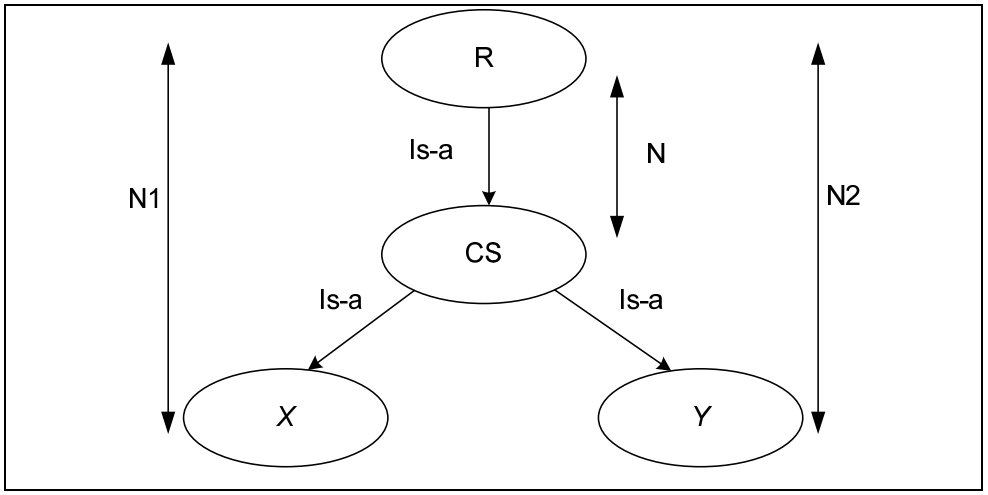

Figure 1.12 Exemple d'une ontologie

L'approche de la mesure de *Wu* et *Palmer* reste une mesure simple à calculer et donne de bonnes performances. Son inconvénient majeur est qu'on peut obtenir des résultats erronés où on peut obtenir une similarité supérieure entre un concept et son voisinage par rapport à ce même concept et son concept fils.

La mesure de *Wu* et *Palmer* a été appliquée par Ke Ning and David O'Sullivan (2012)*, à* une ontologie pour mesurer la similarité sémantique entre les concepts individuels de deux instances d'un contexte. La similarité globale est donnée par la somme pondérée de ces mesures individuelles (Eq. 1.5).

$$sim(C_1, C_2) = \sum_{i=1}^{n} w_i sim_{E_i}(E_i(C_1), E_i(C_2)) \quad (1.5)$$

Où : E_1, E_2,……., E_n , sont les concepts d'une ontologie *O*, C_1 et C_2 sont deux instances du contexte *C,* $sim(C_1, C_2)$ est la mesure de la similarité entre C_1 et C_2, w_i est le poids du concept *i* tel que : $(\sum_{i=1}^{n} w_i = 1)$, $sim_{E_i}(E_i(C_1), E_i(C_2))$ est la similarité individuelle entre deux concepts.

Les poids w_i sont définis par des experts du domaine d'application de l'ontologie ou par un algorithme d'apprentissage.
La mesure de *Wu* et *Palmer* est ensuite appliquée pour déterminer les similarités individuelles $sim_{E_i}(E_i(C_1), E_i(C_2))$.

Mesure d'Ehrig

Le travail d'*Ehrig* introduit trois couches : *Les données, l'ontologie et le contexte.*
La similarité est calculée au niveau des données de type simple ou complexe (entiers, caractères). Les relations sémantiques entres les concepts sont ensuite mesurées au niveau de la couche de l'ontologie. Finalement la couche du contexte spécifie comment les éléments de l'ontologie sont utilisés dans un certain contexte.

Plusieurs autres mesures ont ensuite été introduites comme celles de (Leacock & Chodorow, 1998), (Li, Y. et al., 2003), où chacune essaye de faire des ajustements sur un aspect particulier de la mesure de *Wu* et *Palmer*.

b) Approches basées sur les nœuds (contenu informationnel)

Les mesures précédentes se basent sur l'aspect graphique de l'ontologie et ne considèrent pas l'information contenue dans les concepts C1, C2, C3 et LCS. *Resnik* a proposé une mesure statistique qui mesure le contenu informationnel d'un concept (son entropie dans un ensemble d'autres concepts) par rapport à sa probabilité d'occurrence P(C) dans un corpus associé au domaine de l'ontologie. Ce contenu informationnel ou cette entropie est donné par :

$$IC(C) = -\log P(C) \tag{1.6}$$

Plus un concept a plus d'information, moins il est fréquent dans le corpus (p.ex.: WordNet). Selon *Resnik*, les deux concepts C1 = Homme et C2 = Femme (Figure 1.11) sont plus similaires quand leur subsumant commun LCS a un contenu informationnel plus grand parce que LCS représente les informations communes entre C1 et C2.

Plusieurs autres mesures inspirées de celle de *Resnik* ont été proposées ensuite, comme celle de (Lin D., 1998) et celle de (Jiang & Conrath, 1997) où les contenus informationnels de chacun des concepts C1 et C2 est considéré dans la mesure pour évaluer avec plus de précision les informations communes.

Ces méthodes ont plusieurs limitations dont la dépendance au corpus utilisé, où les concepts peuvent être parfois ambigus ou même ne pas être présents. Elles donnent aussi le même résultat pour n'importe quelles paires de concepts qui ont le même LCS (Sánchez, D et al, 2012).

Finalement, la dépendance de ces mesures de la conception de l'ontologie ainsi que la non-considération du contexte sont aussi deux autres importantes limitations de ces mesures.

Parmi les travaux employant cette approche citons :

Mesure de Resnik

Resnik (1995) a proposé une mesure qui combine les statistiques sur un corpus (ensemble de textes) avec *Wordnet*. Son approche se résume ainsi : Dès lors que le concept commun qui subsume (se situe à un niveau hiérarchique plus élevé) deux concepts représente ce qui est similaire entre eux, alors la similarité sémantique entre ces deux concepts est directement proportionnelle au degré de spécificité de ce concept commun. Si ce concept est plus général, la distance sémantique entre les concepts dont on mesure la similarité est plus grande. Cette spécificité est mesurée par la formule du contenu informationnel IC (Eq. 1.7) et la mesure de similarité selon *Resnik* est :

$$sim_{Res}(X,Y) = Max\left[E\left(CS(X,Y)\right)\right] = Max[-\log(p(CS(X,Y)))] \qquad (1.7)$$

Où $\mathrm{CS(X,Y)}$ est le concept commun qui subsume les concepts *X* et *Y*.
Nous pouvons remarquer sur la mesure de *Resnik* que plus le concept subsumant est plus bas dans la hiérarchie, plus la probabilité de son occurrence est basse avec les concepts qu'il subsume, et plus son contenu informationnel IC est plus grand et ainsi les concepts-fils sont plus similaires.

Mesure de Lin

Lin D. (1998) suppose que le concept subsumant est ce qui est commun entre les concepts fils et ainsi son contenu informationnel IC est l'information commune aux deux concepts fils.

Sa formule s'apparente au calcul du coefficient de *Dice* entre les informations des deux concepts fils. Elle est donnée par :

$$sim_l(X,Y) = \frac{2\times\log(P(AC(X,Y)))}{\log(P(X))+\log(P(Y))} \quad (1.8)$$

Mesure de Hirst et Onge

Pour mesurer la similarité sémantique entre deux concepts, Hirst et Onge (1998) se sont basés sur les concepts lexicalisés dans *Wordnet* et la définition de nouveaux liens entre eux (lien horizontal pour les antonymes, lien haut pour les superclasses et lien bas pour les sous-classes), ainsi la mesure de similarité entre objets/mots se fait par le poids du plus court chemin entre eux en prenant en compte les changements de direction. De cette façon, deux concepts sont similaires si leurs synonymes dans *Wordnet* sont reliés par un chemin qui n'est pas trop long et qui ne change pas souvent de direction. La similarité selon *Hirst* et *Onge* est donnée par :

$$Sim_{ht} = T - PCC - K \times nd \quad (1.9)$$

Où *T* et *K* sont des constantes, *PCC* est la distance du plus court chemin en nombre d'arcs et *nd* le nombre de changements de directions.

c) Approches hybrides

Ces approches s'appuient sur le calcul des distances (approches basées sur les arcs) en plus de l'utilisation du contenu informationnel IC, comme un facteur de décision, car il détermine l'information partagée entre deux concepts. La performance est améliorée par l'utilisation des deux approches (arcs et nœuds) simultanément.

Mesure de Jiang et Conrath

La mesure de Jiang et Conrath (1997), détermine la dissimilarité (distance) entre chaque nœud fils et son subsumant par la différence des deux contenus informationnels : IC *(IC(X)− IC(X, Y)) et (IC(Y)− IC((X, Y))*

La distance finale entre ces deux concepts est la somme de ces deux différences et la similarité est donnée par :

$$sim(X,Y) = \frac{1}{distance\ (X,Y)} \quad (1.10)$$

Où la distance entre *X* et *Y* est :

$$distance(X,Y) = E(X) + E(Y) - (2.E(CS(X,Y))) \quad (1.11)$$

Mesure de Leacock et Chodorow

Leacock and Chodorow (1998) ont combiné la mesure des arcs avec le contenu informationnel IC tout en incorporant le fait qu'un concept plus bas dans la hiérarchie d'une taxonomie correspond à une plus petite distance sémantique qu'un concept dans un niveau plus haut dans la hiérarchie. La formule de la similarité devient :

$$sim_{lc}(X,Y) = -\log(\frac{cd(X,Y)}{2.M}) \quad (1.12)$$

Où *M* est la profondeur de la hiérarchie (taxonomie) et $cd(X,Y)$ est la longueur du chemin le plus court qui sépare *X* et *Y*.

d) Mesures de similarité basées sur les caractéristiques

Ce type de mesure de similarité sémantique est intuitivement la mesure qui «a le plus de sens sémantique» que les autres, car elle dépend réellement des caractéristiques des concepts C1 et C2. Chaque concept est décrit par un ensemble de caractéristiques qui lui sont propres et la

similarité sémantique est évaluée par le nombre de caractéristiques communes ainsi que celles qui ne sont pas communes aux deux concepts.

Soient : $\emptyset(C1)$ et $\emptyset(C2)$ les caractéristiques de C1 et C2. $\emptyset(C1) \cap \emptyset(C2)$ sont les caractéristiques communes de C1 et C2. $\emptyset(C1) \mid \emptyset(C2)$, les caractéristiques que C1 possède et que C2 ne possède pas et vice versa pour $\emptyset(C2) \mid \emptyset(C1)$.
Alors, la similarité sémantique entre C1 et C2 est donnée par :

$$sim(C1, C2) = \alpha.F(\emptyset(a) \cap \emptyset(b) - \beta.F(\emptyset(C1)|\emptyset(C2)) - \gamma.F(\emptyset(C2)|\emptyset(C1))) \quad (1.13)$$

Où : F reflète les caractéristiques importantes de C1 et C2, α, β, γ sont des paramètres de pondération. Notons que les caractéristiques dépendent du contexte de leur définition.
Exemple : humain (homme, femme, garçon, fille,…) dans le contexte « genre de l'humain»
et Humain (raisonne, rit,…) dans le contexte « caractéristique de l'humain ».
(Rodríguez & Egenhofer, 2003), (Petrakis, et al., 2006) ont proposé des formules qui lient la mesure sémantique entre deux concepts aux : 1) Synonymes de ces concepts, 2) Glosses : qui sont les définitions de ces concepts et 3) les voisins sémantiques des concepts dans un corpus.

La détermination des paramètres de pondérations représente le désavantage majeur de ce type de mesure de similarité sémantique.

1.6.2.2 Mesures de similarité vectorielle

La similarité vectorielle est associée à toute information quantifiable ou mesurable au sens de la distance métrique qui décrit le contexte telle que la température, le bruit, l'heure, la position géographique etc. Si l'information est du type non quantifiable, telle que l'émotion, l'activité ou l'humeur de l'utilisateur, elle doit être numérisée (Lavirotte S. et al., 2005) sous forme vectorielle. Elle peut ainsi être projetée dans un espace multidimensionnel afin d'appliquer les calculs de similarité/distance proposées.

L'espace vectoriel est le support sur lequel est appliqué le calcul des distances métriques. Un contexte dont les entités (instances) s'écrivent sous forme vectorielle devient un ensemble de points dans l'espace vectoriel. La proximité entre ces entités définit la distance métrique ou la similarité sémantique entre les entités du contexte. Lorsque les attributs/informations du contexte sont du type scalaire, leur représentation est simple. Elle devient cependant plus complexe lorsqu'ils sont du type catégorique/symbolique. Dans ce cas, plusieurs approches sont disponibles comme la densité de probabilité (Sung-Hyuk Cha., 2007) ou les valeurs binaires (Manohar M. G, 2011).

Il y'a plusieurs types de distances qui s'appliquent sur l'espace vectoriel (exemple : Minkowsky, Mahalanobis, Canberra, Tchebychev, Quadratic, Corrélation et du Khi-carré) mais la plus utilisée reste la distance Euclidienne (D. Randall Wilson, 1997).

La quantification des informations communes entre deux entités du contexte quand celles-ci sont du type vectoriel est évaluée par la métrique de la similarité qui est défini ainsi :
Pour un ensemble de données *X,* la fonction $s : X \times X \rightarrow \mathbb{R}$ est une métrique de similarité si pour n'importe quel vecteur *x, y, z* $\in$ *X*, elle satisfait les conditions suivantes :

1. $s(x, y) \geq 0$ (non négativité);
2. $s(x, y) = s(y, x)$ (symétrie);
3. $s(x, x) > s(x, y) \quad \forall$ x,y $\in X$ *(si* $s(x, x) = s(x, y) \quad \rightarrow x = y$*).*

La première condition spécifie que la métrique de similarité ne doit pas être négative, la deuxième spécifie la symétrie de la métrique de similarité et la troisième spécifie que l'autosimilarité est plus grande que n'importe quelle autre similarité entre deux entités différentes.

1.6.2.3 Mesures de similarités et adaptation de services

Les mesures de similarité sémantiques sont utilisées dans beaucoup de domaines pour différents types d'applications. Dans ce qui suit, nous allons présenter quelques travaux représentatifs de l'application de ces mesures.

Par exemple (Liwei Liu et al, 2010) ont proposé un système de recommandation de services basé sur les préférences de l'utilisateur. Ils ont utilisé entre autres la distance Euclidienne pour mesurer la distance entre deux évaluations de deux utilisateurs différents. (Hartmann M. et al., 2008) ont aussi utilisé la mesure du *cosine* pour comparer une série de caractères dans leur système d'adaptation d'interface. (Ke Ning and David O'Sullivan, 2012) se sont appuyés sur la mesure de similarité de (G. Prasanna et al., 2003) entre concepts d'une ontologie pour inclure le contexte par l'allocation de poids aux relations entre concepts. Nous pouvons citer aussi d'autres applications des mesures de similarité dans des domaines qui peuvent être exploités dans le domaine de l'informatique diffuse telles que l'extraction de données proposée par (Ahmad El Sayed et al., 2008) ou le travail de (Thabet Slimani et al., 2007) qui ont amélioré la mesure de similarité sémantique de *Wu* et *Palmer* (Z. Wu et M. Palmer, 1994) pour tenir compte du contexte. Cette mesure a été appliquée sur les structures en arbres (ontologies, taxonomies). (Ahmed Alasoud, 2009) avait travaillé sur les mesures de similarité entre ontologies. Il a introduit une technique de mesure de correspondances ou similarités entre ontologies à plusieurs critères pour améliorer la précision des mesures de similarités individuelles. Il reste que ces travaux ont des applications générales dans des domaines telles que la recherche d'informations dans le web, le partage de fichiers dans des applications telles que le P2P (peer to peer) ou autres applications qui nécessitent des mesures de similarités sémantiques entre ontologies. Au meilleur de notre connaissance, cette technique n'a pas été appliquée dans un SID afin de fournir des services appropriés à un utilisateur ou à une application sensible au contexte. Finalement, citons le projet SimPack du groupe *Dynamic and Distributed Information Systems Group* de l'université de Zurich (Bernstein, A. et al, 2005). C'est un projet dont le but est l'étude et la mise en œuvre de plusieurs types de mesures (vectorielles, caractères, séquences, arbres, graphes, etc.) de similarités entre

concepts individuels dans des ontologies ou entre ontologies globalement et qui peuvent être exploités dans le travail de cette thèse.

Dans les cas où les mesures de similarité sont appliquées sur des systèmes informatiques diffus, la remarque générale est que chaque travail repose sur une définition particulière du contexte et que la mesure dans des cas particuliers n'a pas été évaluée ou testée. Nous pouvons citer le travail de (Kirsh-Pinheiro et al. 2008), qui ont proposé une adaptation dynamique de services pour résoudre le problème de l'incomplétude des informations dans le processus du choix du service adéquat à un contexte particulier. (Yazid Benazzouz, 2011) avait également utilisé ce type de mesures de similarité dans sa thèse pour regrouper des données du contexte afin de découvrir des situations particulières qui déclenchent un service particulier.

1.7 Conclusion

La revue de la littérature nous a permis de présenter les travaux ainsi que les tendances dans les domaines proches de notre axe de recherche. Elle nous a aussi permis d'adopter des approches qui serviront de base pour notre travail. Ainsi, parmi les définitions du contexte, nous adopterons celle où les informations contextuelles sont manipulées indépendamment des services fournis à l'utilisateur et où le contexte est un ensemble d'informations collectées pour un objectif précis à savoir l'adaptation des services. Nous avons également montré que les informations contextuelles n'ayant pas toutes la même valeur, ne peuvent pas être pondérées d'une manière identique. Aussi, la localisation d'un individu est l'information contextuelle la plus significative et la plus abordée dans la littérature. D'autre part, nous avons pu conclure que l'adaptation dans un système informatique diffus est soit une adaptation de contenu, de présentation ou de comportement et que les sources d'informations contextuelles sont de deux types : a) les informations qui proviennent de l'environnement utilisateur et b) celles qui proviennent de l'environnement système. Pour les mesures de similarité, nous avons pu noter qu'il n'y'a aucun travail dans lequel les mesures de similarités sont appliquées dans le but de permettre une adaptation dynamique de services. Nous avons

toutefois noté quelques travaux où les mesures de similarités sont appliquées indirectement comme dans la recommandation de services ou dans la recherche de l'information. Les chapitres suivants seront dédiés aux applications citées dans la partie des contributions et objectifs de notre travail ainsi qu'une enquête sur les mesures de similarité sémantiques dans le domaine de l'informatique ubiquitaire.

CHAPITRE 2

SURVEY OF SEMANTIC SIMILARITY MEASURES IN PERVASIVE COMPUTING

Djamel Guessoum [a] , Moeiz Miraoui [b], Chakib Tadj [a]

[a]Electrical Engineering Department, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[b]Al-Leith computer college, Umm Al-Qura university, Saudi Arabia
21421 Makkah, Saudi Arabia

Cet article a été publié dans le journal « International Journal on Smart Sensing and Intelligent Systems », Vol. 8, no. 1 (2015): 125-158.

Résumé

L'utilisation des mesures de similarité sémantique est très répandue dans l'informatique diffuse avec les objectifs suivants: 1) pour comparer les composants d'une application; 2) pour recommander et classer les services par degré de pertinence; 3) pour identifier les services en faisant correspondre la description d'une requête avec les services disponibles; 4) pour comparer le contexte actuel avec des contextes déjà connus. Les travaux existants qui appliquent des mesures de similarité sémantique dans le domaine de l'informatique diffuse se concentrent sur un seul problème. En outre, les enquêtes dans ce domaine sont limitées à la recommandation ou la découverte de services sensibles au contexte. Dans cet article, nous présentons une enquête sur les mesures de similarité sémantique sensibles au contexte utilisés dans divers domaines de l'informatique diffuse.

Mots clés*:* informatique diffuse, similarité sémantique, sensibilité au contexte, découverte de services, recommandation de services.

Abstract

Semantic similarity measures usage is prevalent in pervasive computing with the following aims: 1) to compare the components of an application; 2) to recommend and rank services by degree of relevance; 3) to identify services by matching the description of a query with the available services; 4) to compare the current context with already known contexts. The existing works that apply semantic similarity measures to pervasive computing focus on one particular issue. Furthermore, surveys in this domain are limited to the recommendation or discovery of context-aware services. In this article, we therefore present a survey of context-aware semantic similarity measures used in various areas of pervasive computing.

Keywords: pervasive computing, semantic similarity, context-aware, service discovery, service recommendation

2.1 Introduction

Similarity involves the assessment of intrinsic common characteristics between two or more concepts. A characteristic is intrinsic to an object when it defines the nature of the object itself and cannot be separated from it. In information systems in particular, similarity relates to the assessment of likeness in an analyzed date set in order to quantify these similarities in the interval [0, 1]. As a result, it is possible to order and prioritize them or extract invariants. Generally, the similarity evaluation involves three types of data processing, namely classification, identification, and characterization (Bisson 2000). Classification aims to structure data in a heterogeneous group according to similarity, while identification endeavors to recognize the class to which an unknown object is likely to belong. Finally, the characterization process allows the explicit representation of information that is common to a set of data.

Semantic similarity measures are referenced to the similarity measure based on human judgment. This latter notion was first introduced in the study of Ruinstein and Goodenough in 1965, in which two groups of 51 people evaluated the synonyms of 65 pairs of names. In

1991, Miller and Charles repeated the original experiments of Rubinstein and Goodenough using 30 pairs of names taken from the original list of 65: 10 pairs had a high level of synonymy, 10 an average level, and 10 a lower level (Saruladha et al. 2010).

In pervasive computing, semantic similarity measures were implemented as a mechanism to properly adapt the applications and services between the user and environment. Semantic similarities measures in a pervasive computing system (PCS) are thus applied in order to select the modules of a context-aware application that are appropriate to the user's current context, to choose the best advertised service by matching the user's query to the available service description and classifying the selected services according to relevance, and finally, to identify the current context by comparing information collected from the environment with a set of predefined situations.

2.2 Dynamic adaptation of services in pervasive computing

In pervasive computing, the adaptation of services is a dynamic process wherein services are offered reactively to a user in response to a change in context or proactively by predicting a change in context and reacting accordingly (Germán 2010). Several definitions are proposed in the literature, although the most generic is given by Efstratiou (2004) who generalizes the concept of adaptation for mobile equipment and context-aware applications in a PCS by assuming that an application or system is adaptive when it changes its behavior in response to a change in context (this change occurs in either the context or equipment resources). Zouari (2011) recently defined the dynamic adaptation of a context-aware application according to its ability to change its behavior during the execution phase in line with fluctuations in the environment or changes in user requirements.

Another approach has been adopted in other studies, such as that used by Simonin and Carbonell (2007), which categorizes the dynamic adaptation of services according to the purpose of adaptation, thus distinguishing two types of adaptation: adaptation to the user profile and to the environment. This approach requires the user context and environment to

be sources of information for an appropriate adaptation of services. The following works, however, are more comprehensive in terms of services. For example, Nicklas et al. (2008) categorize the adaptation of context-aware applications into four classes: 1) the selection of information and services; 2) the presentation of information and services; 3) the automatic execution of a service for a user; 4) the marking of context with information for later retrieval. Benazzouz (2012) classifies the adaptation into three classes, notably the personalization, recommendation, and reconfiguration of services. According to this classification, the personalization of services is linked directly to the user's preferences, deriving its contextual information from the user environment (e.g. ambient temperature, geographical location). Recommendation is a particular form of personalization that draws from user-stored preferences (history) to recommend the most adequate services. Finally, reconfiguration takes into account the system environment (e.g. releasing memory space for an application). Note that reconfiguration does not consider the user's environment.

In what follows, Section 2 of this survey discusses the concept of semantic similarity in general. Section 3 introduces the various applications of semantic similarities measures in the field of ubiquitous computing; semantic similarity measures are discussed between contexts, for the recommendation of services, in context-aware applications, and for the service discovery.

2.3 Notion of semantic similarity

Introduction

In pervasive computing, where the notion of context plays a very important role, the semantic similarity measure is a tool to evaluate the resemblance between instances of a context. It allows services to be chosen and classified according to their relevance to a given query, and a user's profile and preferences to be compared to those of other users in order to recommend similar services. Finally, semantic similarity aims to evaluate the similarity between application components in order to propose the most relevant one in a current context.

Harispe et al. (2013) classify semantic similarity measures according to the type of elements to be compared (i.e., words, sentences, paragraphs, and documents, concepts or groups of concepts, semantically related instances) and the semantic proxies used to extract the required semantics from the measure. In terms of the latter, the semantic proxies are of two types (Mihalcea et al. 2006): corpus-based proxies in which the similarity between two concepts is determined based on the information extracted from a large corpora, and knowledge-based proxies in which the similarity between two concepts is evaluated using information derived from the semantic networks (e.g. ontologies, WordNet).

2.3.1 Notion of distance and similarity

2.3.1.1 Distance

Distance is associated with all quantifiable (scalar or vector) or measurable information that describes a context, such as temperature, noise, time, and geographical position (Lavirotte et al. 2005). Thus, for a space E comprising contexts $E_1, E_2, \ldots . . E_n$ as described by m-dimensional vector entities, X = $(x_1, x_2, \ldots x_m)$, Y= $(y_1, y_2, \ldots y_m)$, ... , the function d: E x $E \rightarrow \mathbb{R}^+$ associated with X and Y has the following properties:

$$\begin{cases} d\,(X,Y) \geq 0 \\ d\,(X,Y) = 0 \Leftrightarrow X = Y \qquad (separation) \\ d\,(X,Y) = d(Y,X) \qquad (symetry) \\ d\,(X,Y) \leq d(X,Y) + d(Y,Z)\ (triangular\ inequality) \end{cases} \tag{2.1}$$

This is known as the *distance* or dissimilarity.

2.3.1.2 Similarity

Definition: Semantic similarity measures are mathematical tools used to quantitatively or qualitatively estimate the robustness of semantic relations between units of language, concepts, or instances of concepts through a numeric or symbolic description obtained from a

semantic support, such as a text or knowledge representation supporting its meaning or describing its nature (Harispe et al. 2013).

The function *s* that defines semantic similarity must have the following properties:
For a set of concepts in a domain X, the function *s*: X×X→ℝ is called "similarity" in X, if $\forall\, x, y \in X$:

$$\begin{cases} s(x,y) = s(y,x) & (symetry) \\ s(x,y) \geq 0 & (non\ negativity) \\ and \quad \forall\, y \in X\ and\ x \neq y : s(x,x) \geq s(x,y) \end{cases} \tag{2.2}$$

The transformations most frequently used to obtain the distance or dissimilarity d from similarity s bounded by 1 are as follows (Michel and Deza 2007):

$$d=1-s,\ = \frac{1-s}{s},\ d = \sqrt{1-s},\ d = \sqrt{2(1-s^2)},\ d = arccos\ (s),\ d = -\ln(s)$$

Semantic similarity measures applied to ontologies

With the advent of the internet and need for information and knowledge sharing on a semantic level, the use of ontologies has become necessary, and as a result, they have considerably developed. The advantages in adopting ontologies as a tool for knowledge representation in a PCS are summarized by Viterbo et al. (2008) as follows:

1. Ontologies are semantically richer than taxonomies or object-orientated models;
2. Knowledge is described through accurate representations;
3. Ontologies are formal; those in web ontology language (OWL-DL) map directly onto the DL (first-order logic);
4. Formal ontologies in OWL-DL can be verified or classified through inference mechanisms (e.g. RACER, FaCT): verification of the consistency, classification, and discovery of new information;
5. Ontologies in OWL use XML/RDF syntax, which allows them to be automatically manipulated and understood by most internet resources;

6. Ontologies capture and represent knowledge in detail;
7. Ontologies can be used to reduce ambiguity by providing a model for sharing information;
8. Ontologies are modular, reusable, and independent of the application's code;
9. Ontologies can be combined with the emerging rule-based languages like semantic web rule language.

The similarity measures between ontologies occur on two levels—lexical and conceptual—which include concepts with semantic relationships (Maedche and Staab 2002). In the case of PCS, the semantic similarity measure must take context into account so that the results are relevant and up-to-date.

Ehrig et al. (2005) classify the "contextual" semantic similarity measures between ontologies and intra-ontologies into three layers (Figure 2.1). First, in the data layer (representation layer), similarity measures are only simple measures between the values of the entities (i.e., integers or characters). Second, in the ontology layer (layer of meaning), the similarity between two concepts is based on the ontological structure and semantic relations represented by the ontology. Third, in the context layer, the external factor of the measure is considered, namely the context in which the ontology develops. Note that the semantic similarity measures between concepts made through the comparison of their common characteristics are also an integral part of the data layer. For example, the concept jaguar (car) and jaguar (animal) are syntactically similar, but very different when described according to their characteristics: vehicle or wheels versus animal or feline. For this reason, the data layer is divided into syntactic and semantic similarities.

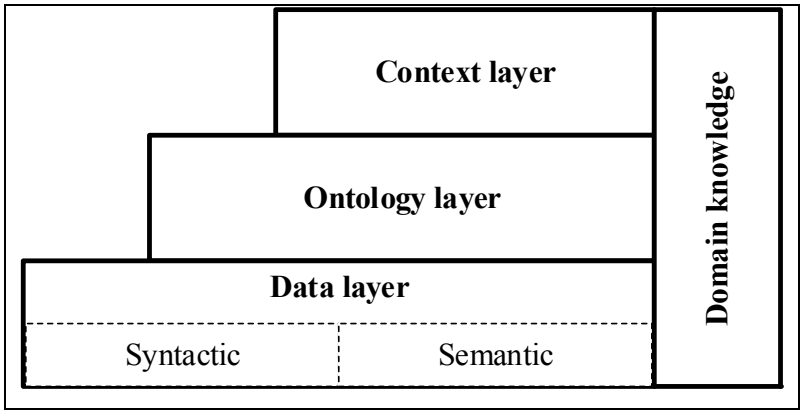

Figure 2.1 Layered model of semantic similarity measures between ontologies and intra-ontologies
Adapted from Ehrig et al. (2005)

Recently, Sanchez et al. (2012) and Saruladha (2011) made the most developed semantic similarities measures to date (Table 2.1) based on the ontological representation of knowledge (ontology layer), especially in its taxonomic form. The measures are as follows:

- edge counting-based measures;
- informational content-based measures;
- feature-based measures;
- hybrid measures.

Edge counting measures were first introduced by Rada et al. (1989). These apply to ontologies with relations between concepts of the taxonomic type (is-a). The basic idea of these measures is the fewer number of edges between two concepts, the more similar they are. The semantic similarity between two concepts, C1 and C2, in this case is given as:

$$dis(C1, C2) = \min(path(C1, C2)) \quad (2.3)$$

Wu and Palmer (1994) considered the depth of the ontology in the measure, because the more specific two concepts are (in the lower ontological levels), the more similar they will be, and vice versa.

This measure is given as:

$$sim(C1, C2) = \frac{2 \times P}{N_1 + N_2 + 2 \times P} \quad (2.4)$$

Where N_1 is the number of (is-a) edges between the concept C1 and the least common subsumer (LCS) of (C1,C2), N_2 is the number of (is-a) edges between the concept C2 and the LCS of (C1,C2), and P is the number of edges (is-a) between the LCS and ontology root.

Several other measures were subsequently introduced by Leacock and Chodorow (1998) and Li et al. (2003), as the authors attempted to make adjustments for a particular aspect of Wu and Palmer's measure. This type of measure is simple to implement, but it is limited to ontologies with taxonomic relations (is-a). Furthermore, it does not allow for the context and can give incorrect semantic similarity measures.

Semantic similarity measures based on the informational content of the common notion underlying two concepts were first introduced by Resnik (1995). The informational content of a concept is its probability of occurring in a corpus such as WordNet: the higher the occurrence of the concept, the less the informational content. The informational content is given as:

$$IC(C) = -\log P(C) \quad (2.5)$$

Several other measures inspired by Resnik were subsequently proposed. For Lin (1998) and Jiang and Conrath (1997), for example, the informational content of the concepts C1 and C2 is considered when evaluating the shared information more accurately.

Among the limitations of these measures is their dependence on the corpus, as the concepts may be sometimes ambiguous or even not present. They also give the same result for any pair of concepts with the same LCS (Sánchez et al. 2012). Their dependency on the design of the ontology and their lack of consideration for the context are also limitations.

Finally, the semantic similarity measures based on the features of the concepts are based on Tversky's model of similarity (1977), whereby two concepts are more similar if they have more common characteristics and less non-common characteristics.

Let $\emptyset(C1)$ and $\emptyset(C2)$ be the characteristics of C1 and C2. $\emptyset(C1) \cap \emptyset(C2)$ are the shared characteristics of C1 and C2. $\emptyset(C1) \mid \emptyset(C2)$, while the non-common characteristics of C1 and C2 are $\emptyset(C2) \mid \emptyset(C1)$. The semantic similarity between C1 and C2 is thus given as:

$$sim(C1, C2) = \alpha.F(\emptyset(a) \cap \emptyset(b) - \beta.F(\emptyset(C1)|\emptyset(C2)) - \gamma.F(\emptyset(C2)|\emptyset(C1))) \qquad (2.6)$$

Where F reflects the important characteristics of C1 and C2, and α, β, γ are the weighting parameters. Note that the characteristics depend on the context of their definition.

The determination of the weighting parameters represents the major disadvantage of this type of semantic similarity measure.

Table 2.1 Semantic similarity measures
Adapted from Saruladha (2011), Gomaa et al. (2013) and Meng et al. (2013)

1. Semantic similarity measures based on the ontological representation of knowledge		**Studies**	**Specificity**
Inter-ontologies	**Path-based**	Al Mubaid and Nguyen (2009)	- Concept specificity, shortest path, concept depth, is-a relation
	Feature-based	Rodriguez and Egenhofer (2003)	-Three independent similarity assessments: 1) similarity of synonym sets, 2) a feature similarity, 3) types of semantic relations
Intra-ontologies	**Path-based**	Rada et al. (1989)	- Simple, is-a relation, number of edges in a taxonomy - Two pairs with the shortest path of equal length will have the same similarity
		Hirst and St Onge (1998)	- Relatedness measure with different semantic relations, shortest path, automatic detection and correction of malapropisms
		Bulskov (2002)	- Is-a relation, path length, weighted paths, information retrieval
	Depth-based	Wu and Palmer (1994)	-Simple, is-a relation, number of edges, taxonomy depth
		Sussna (1993)	- Based on all possible links, weighted relations, measure between two adjacent concepts - Sensitive to: 1) shortest path between concepts, 2) density of concepts along this path, 3) shortest path from the root to the LCS
		Leacock and Chodorow (1998)	-Simple, Is-a relation,Similarity value using a logarithmic function, -Shortest path in taxonomy, -Maximum depth…..
Informational content-based measures (corpus-based)		Resnik (1995)	- Simple, is-a relation, information content in LCS - Coarse measure less likely to suffer from zero counts - Two pairs with the same LCS will have the same similarity
		Lin (1998)	- Same as Resnik's measure plus commonalities and distinct features of a concept considered
		Jiang and Conrath (1997)	- Is-a relation, shortest path and edges weighted by IC in a taxonomy
Hybrid measures		Li et al. (2003)	-Simple, shortest path, depth of LCS, local semantic density of

1. Semantic similarity measures based on the ontological representation of knowledge	Studies	Specificity
		concepts, multiple corpora used
Feature-based measures	Tversky (1977)	- Features common and distinct between concepts, aymmetrical measure, computational complexity
	Pirró Measure (2010)	- Common features among concepts and different features among concepts, defined in terms of information theoretic domain, corpus independent
2. Corpus-based semantic text similarities	HAL (Hyperspace Analogue to Language) (1995)	- Co-occurrence of words in a corpus (similarity by cosine of vectors) - Only information found in the corpus used - No human bias or influence
	LSA (Latent Semantic Analysis) (1997)	- Use of singular value decomposition (SVD), method for dimensionality reduction, information retrieval … - Solves polysemy, synonymy, and term dependence - Low efficiency and high data storage
	DISCO (Extracting DIStributionally similar words using CO-occurrences) (2009)	- Distributional similarity - Words with similar meaning occur in similar context - Use a context window of size ±3 words for counting co-occurrences
3. Logic-based representation of semantic measures	D'Amato (2007), D'Amato et al. (2009)	- Similarity is the result of the common and different features - Clustering and retrieval on DL knowledge bases - Weakness in cases involving individuals

2.4 Application of semantic similarity in pervasive computing

In a typical PCS, a context-aware application interacts with the physical environment and the user's system in order to provide appropriate services. This interaction may be a response to the user's request for a specific service or to the current context information with the aim of providing services that are relevant to the user. In such an environment, semantic similarity measures have been applied at several levels (Figure 2.2): the comparison of an application's components with respect to their appropriateness in a current context; the recommendation of services and collaborative filtering when comparing the preferences of multiple users with the ranking of services according to their relevance during the recommendation process; service discovery by the matching the description of a request with available services; lastly, the comparison of the current context with already known contexts or the detection of current situations. These applications are detailed in the forthcoming sections.

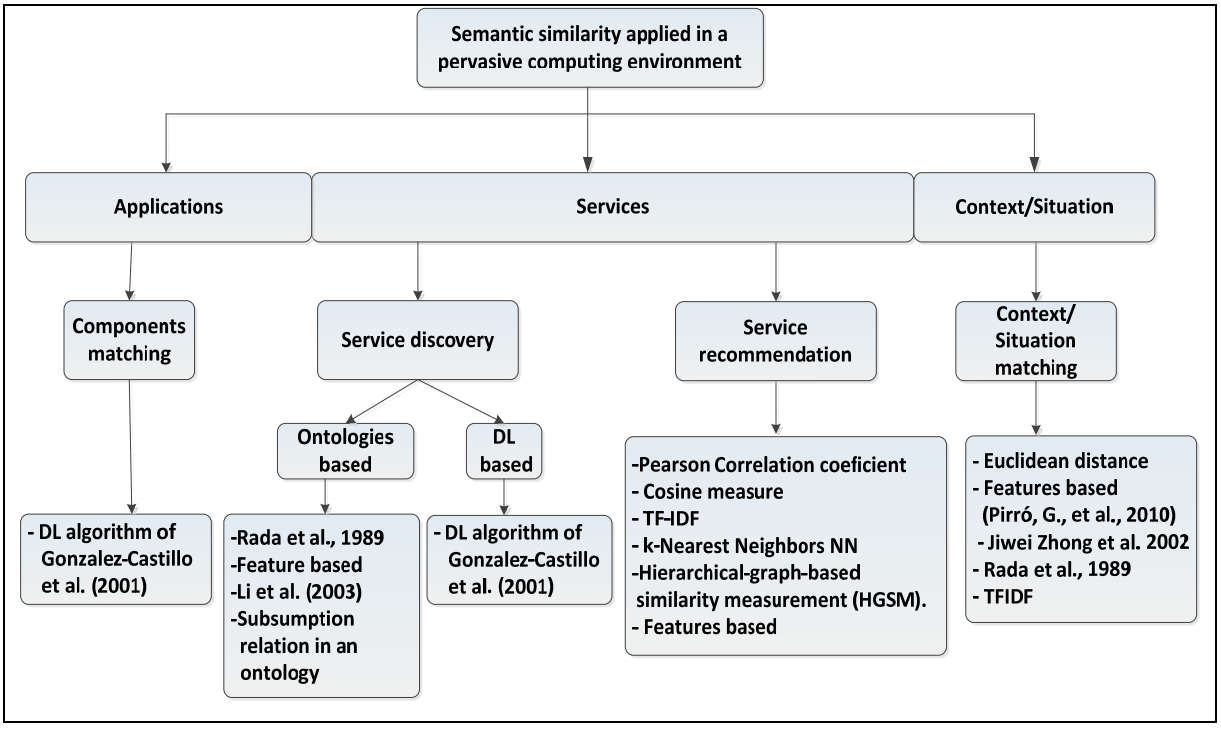

Figure 2.2 Application of semantic similarity measures in pervasive computing systems

2.4.1 Semantic similarity measures and context

The definition of context according to Petit (2005) along with the majority of researchers is based on the four following axes:

1. There is no context without context: the concept of context must be defined in terms of a purpose. For example, the aim may be to adapt the interactive capabilities of a system dynamically;
2. Context is an information space that serves the interpretation: context capture is not an end in itself, but captured data must serve an objective;
3. Context is an information space shared by several actors: the user and the system;
4. Context is an infinite and dynamic space of information: context is not permanently fixed, but is constructed over time.

The following definitions of context should be in accordance with the aforementioned axes. First, Brezillon et al. (1999) defined two concepts relating to context: 1) the set of contextual knowledge (e.g. time, location) to be used in a decision problem, which is latent and cannot be used without an emergent objective; 2) the context as the product of the emergent objective or intention that uses a large part of contextual knowledge.

In 1994, Schilit and Adams categorized context according to six areas. The first three relate to the human factor: user information (e.g. clothes, biophysical conditions), social environment (e.g. proximity to other people), and user tasks (e.g. active user tasks). The other three areas concern the physical environment: location, infrastructure (e.g. resources, communication), and physical conditions (e.g. noise, brightness, weather conditions).

The definition of Dey et al. (2001) is the most cited: “context is any information that can be used to characterize the situation of an entity. An entity is a person, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves” (p. 5). This definition is evidently similar to Schilit’s because context is defined as a set of information collected from the user environment (person), physical environment (physical object), or system environment, with the objective of collection being the characterization of these environments.

Given the preceding definitions, we may say that context is definitely a set of information characterizing an environment, whether the user, physical, or system environment, and that the collection of this information must serve for an objective.

2.4.1.1 Impact of context

Keßler (2007) defines context relative to the similarity measure in the following terms: “A similarity measurement’s context is any information that helps to specify the similarity of two entities more precisely concerning the current situation. This information must be represented in the same way as the knowledge base under consideration, and it must be

capturable at maintainable cost" (p. 4). This definition gives rise to the following questions regarding the choice of contextual information to be included in the similarity measure between two concepts:

1. Impact: does the chosen contextual information improve the accuracy of the semantic similarity?
2. Representation: can this contextual information be represented in the knowledge base?
3. Acquisition: can this contextual information be acquired at a reasonable cost?

Formally, for a contextual information c_n of a context C to be considered in a calculation of semantic similarity between contexts, its impact should be calculated by measuring the semantic similarity that includes and excludes this information. The impact must be greater than a minimum threshold $\underline{\delta}$:

$$Imp(c_n) = \frac{\sum|sim_{(C_n \in C)}(a,b) - sim_{(C_n \notin C)}(a,b)|}{|C|} \quad (2.7)$$

Where $C = \{c|imp(c) > \delta\}$ is the final context including all relevant contextual information. Most semantic similarity measures are between concepts without taking account of the context of the measure, which sometimes leads to implausible results. "Tablet" and "smart phone" are two similar concepts in terms of "information processing," but completely different in terms of "telephony." The limiting factor according to Janowicz (2008) in the collection of contextual information does not concern how much information can be collected, but rather whether this information can be incorporated into the similarity measure (e.g. through weights) and whether it plays a significant role (i.e., an impact on the result of the similarity measurement).

In pervasive computing, the introduction of context has improved existing semantic similarity measures by introducing weights to the characteristics and semantic links. Furthermore, it has facilitated the application of semantic similarity measures in the calculation of contextual similarities between situations, contexts, concepts, or instances of concepts.

2.4.1.2 Semantic similarity between contexts

In a PCS, the services provided to a user relate to the user context (environmental, system-based). The identification of context is thus an essential task. The question that arises is therefore, "What services must an intelligent device in a PCS provide to a user when the current context is identified?" The identification of the current context is defined by the contextual information related to the triggering of a service as well as a situation or "current context" in the set of current contextual information, similar to a known situation or context (Benazzouz 2012), with each identified situation being linked to one or more of the services to be provided. This identification forms the basis of the rule-based adaptation mechanism, which is a set of conditional rules with the form: if (contextual information I) then (service S).

A situation is "a snapshot of the environment at a given point in time" (Ramparany et al. 2011). Identifying a situation is based on data mining techniques. Once identified, semantic similarity measures are applied in order to compare it with situations with known services. In Dietze et al. (2008), semantic similarity is measured against the Euclidean distance between the contextual data vectored in mobile situation spaces. Gicquel (2012) modeled the spatio-temporal context of a museum visitor in an ontological form, with the semantic similarity measures being used to recommend artwork similar to the interests of the user by comparing the properties of two concepts in the knowledge base. The similarity measure is a modified version of the similarity proposed by Pirró and Euzenat (2010), which combines the similarity calculation based on Tversky's model with that of informational content.

Benazzouz (2012) and Ramparany et al. (2011) applied semantic similarity measures to group data and "pure" contexts in order to build relevant situations for the adaptation of services declared within the ontology of context. First, syntactical and conceptual (semantic) similarity measures between contextual data are applied based on the measures of Zhong et al. (2002) and Rada et al. (1989). Second, conceptual and relational similarity measures between "pure" contexts are used based on the quantification of information common to two

graphs and on the statistical technique known as TF-IDF (term frequency–inverse document frequency).

Ontological representation is also used to model a set of situations that occur frequently, such as the locations “at home” or “at work.” Semantic similarities are made between contextual variables representing the current situation and the “frequent” situations, while the services provided are a set of appropriate notifications (Meissen et al. 2005).

A similar approach was proposed by Kirsch-Pinheiro et al. (2006) for the adaptation of content found in an intelligent device with a PCS. The authors used semantic similarity measures to assess the degree of matching between the predefined profiles of situations and the current context of the user with the aim of prioritizing them. Using a graph-modeled context, it estimates the proportion of elements in the graph defined by the user’s current context, with the graphical elements defined by each user profile. This measure is determined as follows:

$$sim(C_u, C_p) = x, \quad x \in [0,1]$$

$$\text{With: } \begin{cases} x = 1 \ \ if\ each\ element\ of\ C_u \text{ has an equal element in } C_p \\ Else \\ x = \frac{|X|}{|C_u|}\ with \quad X = \{x | x\ equal\ to\ y, x \in C_u, y \in C_p\} \end{cases} \tag{2.8}$$

Semantic similarities between contexts in a PCS are thus based on the collection of one or several elements of contextual data that are relevant to one or several services. The description and semantic relations of these services are described in an ontological form, thus allowing the application of known semantic similarity measures.

2.4.2 Recommendation of services in a PCS

In a PCS, the recommendation of services must consider the context as well as the user’s preferences (Figure 2.3). The context and user preferences can be used to limit the number of

recommended services or rank them according to their relevance to the user (Van Setten et al. 2004), while the contextual information can also serve to reduce the issue of limited data (Liu et al. 2010).

Formally, if C is a set of users, S a set of products (services) to be recommended (e.g. books, movies), and u the utility function represented by the rating of how much a user c has appreciated the service s, then the measure of the relevancy of a product or service $s \in S$ to the user $c \in C$ is $u: C \times S \rightarrow R$, where R is a bounded set of integers or reals. For each user $c \in C$, we want to select the product or service $s' \in S$ that maximizes the utility function, where $\forall c \in C, s'_c = \arg\max_{s \in S} u(c, s)$.

The most popular types of service recommendations found in the literature are the following:

1. Collaborative filtering: The user's ratings of a product/service are collected, and services recommended to the user based on the ratings of other similar users. The two most popular approaches for measuring similarities between users are those of Adomavicius and Tuzhilin (2005) and Liu et al. (2010), notably the correlation and cosine approaches, defined as follows:

The *correlation approach* uses the Pearson correlation coefficient:

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{y,s} - \bar{r}_y)^2} \quad (2.9)$$

Where x, y are two users rating the same services, S_{xy} is the set of services rated by users x and y, ($r_{x,s}$, $r_{y,s}$) are the ratings of service s by the users x and y, and ($\bar{r}_x$, $\bar{r}_y$) are the average ratings of x and y.

The *cosine approach* considers users as a set of vectors in a space with the dimension of the set of services S_{xy}:

$$sim(x,y) = \cos(\vec{x},\vec{y}) = \frac{\vec{x}.\vec{y}}{\|\vec{x}\|_2 \times \|\vec{y}\|_2} = \frac{\sum_{s \in S_{xy}} r_{x,s} r_{y,s}}{\sqrt{\sum_{s \in S_{xy}} r^2{}_{x,s}} \sqrt{\sum_{s \in S_{xy}} r^2{}_{y,s}}} \quad (2.10)$$

Where $\vec{x}.\vec{y}$ is the dot product between vectors $\vec{x}$ and $\vec{y}$;

2. Content filtering: The services are recommended to a user based on their description and the user profile and preferences (Adomavicius and Tuzhilin 2005; Henricksen et al. 2006; Sharma and Gera 2013). The utility function is represented by:

$$u(c,s) = \cos(\overrightarrow{w_c}, \overrightarrow{w_s}) = \frac{\sum_{i}^{K} w_{i,c}.w_{i,s}}{\sqrt{\sum_{i=1}^{K} w^2{}_{i,c}} \cdot \sqrt{\sum_{i=1}^{K} w^2{}_{i,s}}} \quad (2.11)$$

Where $\overrightarrow{w_c}$ and $\overrightarrow{w_s}$ are the TF-IDF vectors of the keyword weights (keyword describing the content of an item), $u(c,s)$ is the utility function, and K is the total number of keywords in the system.

Case-based reasoning (CBR) is another technique used in context-aware systems for the recommendation of services (Lee and Lee 2007) in which the similarity function used to find in past cases similar to the current case (context) is based on the algorithm of the k-nearest neighbors:

$$sim(N,C) \frac{\sum_{i=1}^{n} f(N_i,C_i) \times W_i}{\sum_{i=1}^{n} W_i} \quad (2.12)$$

Where N_i is the value of characteristic i of the new case, C_i is the value of characteristic i of the old case, n is the number of characteristics, $f(N_i, C_i)$ is the distance function between N_i and C_i, and W_i is the weight of characteristic i.

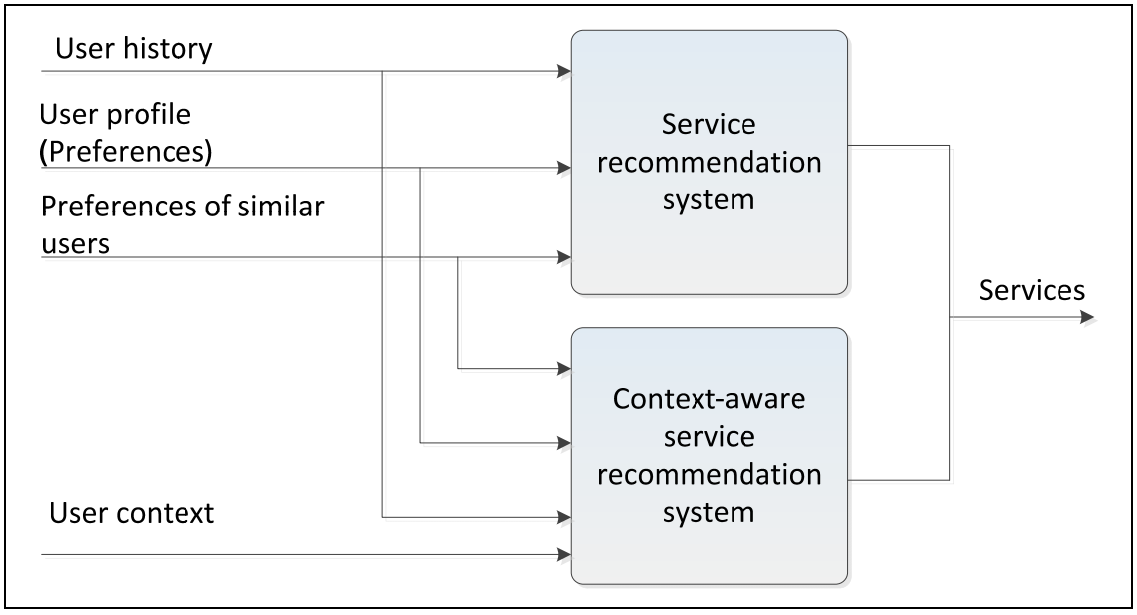

Figure 2.3 Service recommendation systems and context-aware service recommendation systems

2.4.2.1 Context-aware services

The use of contextual information in the service recommendation process in a PCS is achieved in two ways according to Adomavicius et al. (2011): recommendations through context-driven querying and searches (including the current user context) and through contextual preference elicitation and estimation (techniques that model and learn user preferences using collaborative filtering, content filtering, or various intelligent data analysis techniques). In recommendations through the incorporation of contextual information, the context for the selection of a service in the past is the central element on which the present recommendation in a current context is based.

With the context-aware collaborative filtering technique, several studies employ the Pearson correlation coefficient (Table 2.2, Section a) to introduce the contextual information relevant to the selection of services by multiple users in different contexts. Chen (2005) used this coefficient to measure the similarity between two sets of contextual information (Table 2.2, Section b) based on the assumption that if user preferences for a product do not differ in different contexts, then the ratings given in one particular context should apply to another

context. Thus, if the ratings of a product are similar in two different contexts, then these two values are relevant to one another.

A similar approach is adopted by Chang and Song (2012), where the spatio-temporal similarity of the user's service ratings is evaluated by the Pearson correlation coefficient (Table 2.2, Section c). The assumption is that two users are more similar when they choose the same co-located services at the same time. Furthermore, Li et al. (2008) assumed that the more two users have a common location history, the more they share common interests and preferences. The proposed similarity is thus a hierarchical-graph-based similarity measurement (HGSM).

The above approaches are a set of assumptions based on the spatio-temporal context in a user's history. In the best cases, this choice can be used as the final step to evaluate and choose between two or more selected services.

The ontological representation of services as well as contextual information is largely used for the measurement of semantic similarities in the recommendation of services. In García-Crespo et al. (2009), the services described by an ontology are recommended based on the user's preferences and history, with a defined threshold that decides the relevance of the recommended service; the context is represented by the actual location of the user. The semantic similarity algorithms used are thus based on characteristics (e.g. Paolucci et al. 2002). These ontologies are also found in McGovern (2013) to describe the occupation, interests, and so forth of user m, where the semantic similarity measures are calculated between attributes of the same type (e.g. food, occupation) and each attribute is described by a more specific taxonomy that facilitates the calculation of similarity. As a result, a developer can design a richer application for a number of similar users.

Table 2.2 Semantic similarity measures applied to service recommendations

Semantic similarity in recommendation systems	Semantic similarity type	Studies	
Similarity between contexts	- Concept abduction (Liu et al.) - Feature-based semantic similarity measures (García-Crespo et al.) - Semantic similarity and scalar distance (according to the context definition)	- Liu et al. (2010) - García-Crespo et al. (2009)	
Case-based reasoning (Similarity between cases/contexts)	k-nearest neighbors $sim(N,C) = \frac{\sum_{i=1}^{n} f(N_i,C_i)\times W_i}{\sum_{i=1}^{n} W_i}$	-Lee and Lee (2007)	
Collaborative filtering (similarity between users)	- Pearson coefficient of correlation: $sim(x,y) = \frac{\sum_{s\in S_{xy}}(r_{x,s}-\overline{r}_x)(r_{y,s}-\overline{r}_y)}{\sqrt{\sum_{s\in S_{xy}}(r_{x,s}-\overline{r}_x)^2 \sum_{s\in S_{xy}}(r_{y,s}-\overline{r}_y)^2}}$ - Cosine method: $sim(x,y) = \cos(\vec{x},\vec{y}) = \frac{\sum_{s\in S_{xy}} r_{x,s}r_{y,s}}{\sqrt{\sum_{s\in S_{xy}} r^2{}_{x,s}}\sqrt{\sum_{s\in S_{xy}} r^2{}_{y,s}}}$ - Euclidean distance between users who have rated the same product: $sim(u,u') = \frac{1}{1+\sqrt{\sum_{j=0}^{n}(r_j - r'_j)^2}}$	- Adomavicius and Tuzhilin (2005) - Liu et al. (2010)	Section a
Collaborative filtering (context-aware)	- Pearson coefficient of correlation :(contextual relevance): $Rel_t(x,y,i)^* = \frac{\sum_{u=1}^{n}(r_{u,i,x_t} - \overline{r}_i)(r_{u,i,y_t} - \overline{r}_i)}{\sigma_{x_t}.\sigma_{y_t}}$	- Chen (2005)	Section b
	- Pearson coefficient of correlation (spatio-temporal similarity): $sim(x,y)^{**} = \frac{\sum_{s\in S_{xy}}(T_{x,s} - \overline{T}_x)(T_{y,s} - \overline{T}_y)}{\sqrt{\sum_{s\in S_{xy}}(T_{x,s} - \overline{T}_x)^2 \sum_{s\in S_{xy}}(T_{y,s} - \overline{T}_y)^2}}$	- Chang and Song (2012)	Section c
Content-based measures	Cosine between keyword vectors : $u(c,s) = \cos(\vec{w}_c,\vec{w}_s) = \frac{\vec{w}_c,\vec{w}_s}{\|\vec{w}_c\|_2 \times \|\vec{w}_s\|_2})$	Adomavicius and Tuzhilin (2005)	

$\mathrm{Rel_t(x,y,i)^*}$ is the relevancy of ratings for a product i between two contextual variables x and y (of the same type) and is equivalent to their semantic similarity. r_{u,i,x_t} is the rating of a user u for a product i in a context x.

$sim(x, y)^{**}$, S_{xy} are the co-located services accessed by users x and y, $T_{x,s}$ and $T_{y,s}$ respectively denote users *x* and *y* accessing service s, and $\overline{T}_x$ and $\overline{T}_y$ respectively denote the mean value of time wehn users x and y access service s.

2.4.3 Semantic similarity measures and applications

In a PCS, applications must be sensitive to their execution context, which can be any element that influences the behavior of the application (Capra et al. 2001). To provide the functionalities expected by the user along with the desired quality, applications must therefore be able to reason about changes in context and reconfigure their behavior to meet well-defined objectives (Kakousis et al. 2010). Dalmau et al. (2009) categorize these adaptation objectives as the adaptation of data, services, and presentation. The first type of adaptation relates to the provision of complete and formatted information based on raw data. The adaptation of services concerns the architecture of the application (Figure 2.4), while that of presentation relates to the interfacing between the user and the equipment.

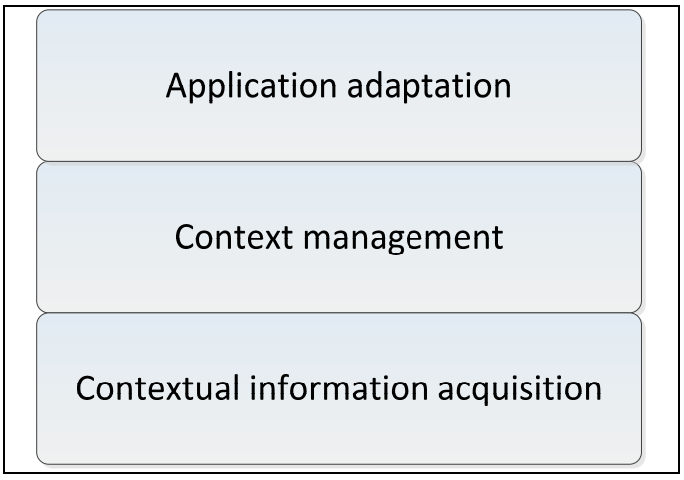

Figure 2.4 Context-aware applications architecture
Adapted from Dalmau et al. (2009)

The semantic similarity between components of an application is defined as follows: two components are similar if the substitution of one by the other allows the user to do the same task. For example, a PowerPoint slide is similar to an Acrobat slide because both allow the

presentation of texts and pictures. Ranganathan et al. (2005) applied this definition to measure the semantic similarity between components of an application by changing its architecture.

Ontologies are used to describe the semantic properties of an application's components (its function, applicable hardware, readable data formats, etc.). The semantic similarity measure uses the DL algorithm of Gonzalez-Castillo et al. (2001), and it is defined by the relative location of the components in the domain ontology, in which the two concepts C1 and C2 are similar to a certain level:

- C1 is equivalent to C2, with a similarity level of 0;
- C1 is a sub-concept of C2, with a similarity level of 1;
- C1 is a super-concept of C2 with a satisfiable intersection with C2, or C1 is a sub-concept of a super-concept of C2 with a satisfiable intersection with C2; the similarity level is 2+i, where i is the number of nodes on the path in the ontology hierarchy from C2 to the relevant super-concept of C2.

Preuveneers et al. (2009) adopted the same approach based on the modular architecture of a context-aware application for measuring the semantic similarity between components. The authors used the semantic similarity measure of Kirsch-Pinheiro et al. (2006) for content adaptation by having defined user profiles that contain information characterizing the user's context and a set of filtering rules when matching with user's current context to provide the proper content to the user.

2.4.4 Service discovery

Service discovery in a PCS is the process of locating the appropriate services to meet the needs of the entity making the request (person or device) (Huaglory Tianfield 2011; Yau et al. 2006). This process is characterized by the following phases: service query, matching, and delivery of the most appropriate service (Broens et al. 2004; Thompson 2006). The context in service discovery in a PCS is defined in Doulkeridis et al. (2006) by "the implicit information

related both to the requesting user and service provider that can affect the usefulness of the returned results" (p. 4). This information is used by Yau et al. (2006) in order to:

1. Expand the service requests to provide more relevant information that is not explicitly specified by users;
2. Describe users' preferences to different services;
3. Further categorize services to retrieve better results;
4. Define the policies to provide services among service providers;
5. Infer the service semantics based on service descriptions in the matchmaking phase in service discovery.

Finally, the information can improve the two evaluation factors "precision" and "recall" of the semantic similarity measures as well as the relevancy of services provided in a PCS (Klein and Bernstein 2004; Yau 2006). The two factors are defined as follows:

Recall = Number of relevant services retrieved in a service discovery / Total number of relevant services available

Precision = Number of relevant services retrieved in a service discovery /Total number of services identified

The semantic similarity measures involving the context used in the service discovery are applied in the phases shown in Figure 2.5 below:

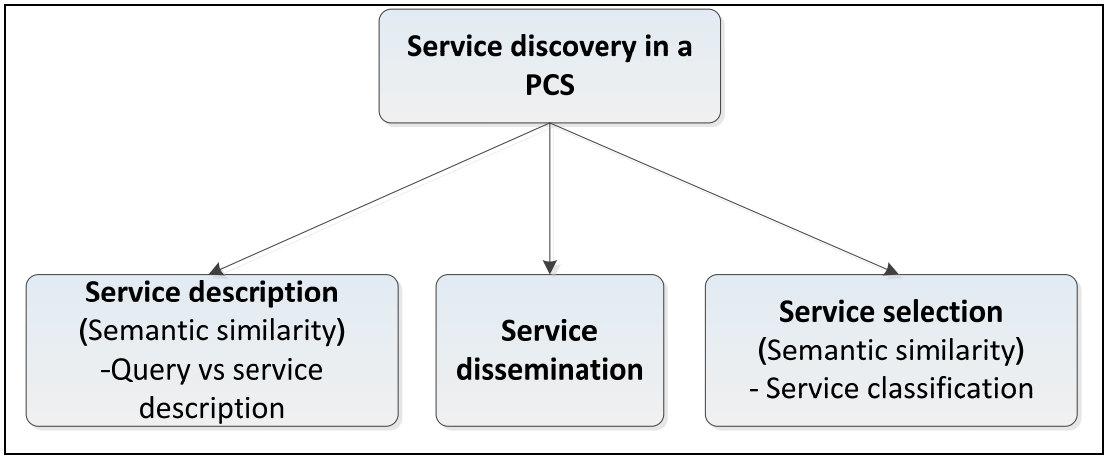

Figure 2.5 Service discovery in a PCS and semantic similarity measures

The semantic similarity between the query and service depends on their representation, which is either an ontological representation or expressed in DL language. For the ontological representation of queries and services, a common ontology to describe both the queries and services is required in order to implement measures such as edge counting (Aydoğan and Yolum 2007; Ge and Qui 2008; Rada et al. 1989). Moon et al. (2008) used WordNet as the ontology to find the synonym of a DTD expressed in XML. The subsumption relation in a common ontology (Bandara et al. 2007) is the tool used for measuring the semantic similarity between the symbolic attributes of the query and the available services. The semantics of each attribute of the query and services, as described by an ontology with "is-a" and "part-of" relations, is shared by all nodes of the PCS (Kang et al. 2007). The semantic similarity measure between attributes is thus given as follows:

$$sim(c_1, c_2) = \begin{cases} e^{-al.\frac{e^{\beta h}-e^{-\beta h}}{e^{\beta h}+e^{-\beta h}}} & si\ c_1 \neq c_2 \\ 1 & Else \end{cases} \quad (2.13)$$

Where l is the shortest path between the concepts c_1 and c_2, h is the level of LCS in the ontology, and $\alpha \geq 0$ and $\beta \geq 0$ are two scaling parameters for the contribution of l and h.

Finally, a graphical approach based on Tversky's semantic similarity measure is introduced in Ganter and Stumme (2002), where a service is more relevant if it has more contextual attributes (user preferences) in common with the query R. These ontology-based measures

always depend on the structure of the ontology, which may change from one designer to another, and as a result, they are not always consistent.

For the description of the query R and available services O as expressed in DL, the matchmaking in this case is categorized according to the five categories listed below (Gonzalez-Castillo et al. 2001; Ruta et al. 2012):

1. *Exact:* all features requested in R are exactly the same as those provided by O and vice versa;
2. *Full-subsumption*: All features requested in R are contained in O;
3. *Plug-in*: All features offered in O are contained in R;
4. *Potential-intersection:* An intersection exists between the features offered in O and those requested in R;
5. *Partial-disjoint*: Some features requested in R are in conflict with some of those offered in O.

For the classification of the identified services, semantic similarity measures are used to limit the number of services identified in accordance with their degree of relevance. The context is an element used in this classification. Ruta et al. (2012) represent context through the geographic proximity of the query to the service provider. This aims to classify services in terms of their functional and non-functional properties (e.g. context, quality of service) and according to the four levels of matching as defined by Paolucci et al. (2002). The current contextual information and services enriched with contextual information (e.g. age, location) both being described by ontologies, are compared node by node (Kirsch-Pinheiro et al. 2008). The semantic similarity measure thus depends on the shared proportion of nodes and arcs between the two graphs:

$$sim_l(E_i, E_j) = \frac{Sim_l(l_i, l_j) + \sum_1^p Sim_l(C_{E_i}, C_{E_j})}{(p+1)} \quad (2.14)$$

Where l_i, l_j are the edge labels and C_{E_i}, C_{E_j} are the edge extremities.

The contextual information (attribute-value) described by Broens et al. (2004) is the final phase in the matching process between a query R and service description S in order to classify the results of the previous phases. The process of matching is achieved by step-by-step filtering. During each step, a property of the service (service type, input, output, contextual attribute, etc.) that is present in the query but not present in service is used to eliminate the services not relevant to the query.

Table 2.3 Table summarizing the studies on semantic similarities in pervasive computing

	Studies	Measure support	Type of similarity	Specificities	Contextual elements
Semantic similarity between contexts	Gicquel (2012)	Ontology	Pirró and Euzenat (2010)	User interactions contextualized according to the user profile and physical location	- User Profile - Physical location
	Wen'an Zhou (2012)	Spatio-temporal data	Pearson coefficient	Increases the ratio of user satisfaction	- Space-time
	Benazzouz (2012)	Ontology	Jiwei Zhong (2002) Rada et al. (1989)	Method is able to detect recurring patterns and improve the efficiency of context-aware services	- Current context
	Hartmann et al. (2008)	Ontology	Wordnet, Wikipedia, Wiktionary,c-vector	String-based measures have higher performance than semantic ones	- Current context
	Dietze et al. (2008)	Vectorized data	Euclidean distance	Context-adaptation across distinct mobile situations	- Technical environment - User objectives - Current location
	Kirsch-Pinheiro et al. (2006)	Graph	Common elements	Analyzes the user's current context and selects from among the user's predefined profiles	- User profile
	Meissen et al. (2005)	Ontology /taxonomy	Subsumes path within the dimensions	Delivers relevant information at the right time to mobile users	- Space-time - Current context
Semantic similarity applied to the recommendation of services	McGovern (2013)	Ontology / taxonomy	Comparison between the taxonomies of same-type attributes	Determines if a given group of users have a quantitative similarity determinant	- User proximity - Occupation - Food - Interests
	Chang and Song (2012)	Spatio-temporal data	Pearson coefficient	Adaptation based on user-to-object, space-time interaction patterns	- Space-time
	Liu et al. (2010)	Ontology / DL	Concept abduction, scalar measure	Multi-context and multi-criteria service recommendations based on collaborative filtering	- Current context - User preferences
	García-Crespo et al. (2009)	Ontology	Features of Paolucci et al. (2002)	Fusion of context-aware pervasive systems, GIS systems, social networks, and semantics	- GIS - Social networks
	Li et al. (2008)	User location	Hierarchical-graph-based similarity measurement (HGSM).	Geographically mines the similarity between users based on their location histories	- Location

	Studies	Measure support	Type of similarity	Specificities	Contextual elements
	Lee and Lee (2007)	Features	k-nearest neighbors	Context-aware music recommendation system using case-based reasoning	- User profile (Listening history) - Current context
	Chen (2005)	Hierarchical structure within each context type	Pearson coefficient	System to predict a user's preference based on past experiences of like-minded users	- Current context
	Van Setten et al. (2004)	Context ontology and domain-specific rules	CBR Similarity functions	Recommendation system (COMPASS)	- Current context - Space-time - User interests
Semantic similarity applied to applications	Ranganathan et al. (2005)	Ontology / DL	Gonzalez-Castillo et al. (2001)	Allows mobile, ubiquitous applications to be adaptive, self-configuring, and self-repairing (built on top of GAIA)	- User location - Device location - Whether device already has applications running - Neighborhood - Current activity
	Preuveneers et al. (2009)	Graph	Kirsch-Pinheiro et al. (2006)	Addresses context in large-scale networks and context-aware redeployment of running applications in a distributed setting	- Current context - Location - Identifying attribute of the device
Semantic similarity applied to service discovery	Aydoğan and Yolum (2007)	Ontology	RP similarity (modified Rada et al. 1989)	Incremental learning architecture in which both consumers and producers use a shared ontology to negotiate a service	- User preferences
	Bandara et al. (2007)	Ontology	Subsumption relation, features (Tversky), scalar measure	Ranking mechanism to order available services according to their suitability	- User preferences and interests
	Kang et al. (2007)	Ontology	Li et al. (2003)	Service clustering supports scalable semantic queries with low communication overheads and balanced load distribution.	- User preferences
	Ruta et al. (2012)	Ontology / DL	Logic based	Ranks identified resources based on a combination of their semantic similarity with respect to the user request and their geographical distance from the user itself .	- Query and service provider geographical proximity
	Mokhtar et al. (2007)	Ontology / EASY-L	Paolucci	Supports efficient, semantic, context-aware service and quality-of-service aware identification in addition to the existing SDPs.	- Current context - User profile
	Kirsch-Pinheiro et al. (2008)	Ontology / graph	Local similarity measures between concepts and global measures between graphs.	A graph-based algorithm for matching contextual service descriptions using similarity measures. matches	- Current context - Space-time
	Broens et al. (2004)	Ontology	Li and Horrocks (2004), clustering	Uses ontologies to capture the semantics of the user's query, services, and the contextual information	- Current context - Space-time - User preferences

2.5 Conclusion

In this article, a survey of the semantic similarity measures applied in the field of pervasive computing was presented. The works related to the application of semantic similarity measures between contexts/situations, service recommendations, applications, and service discovery. Semantic similarity measures in the field of pervasive computing mainly relate to the notion of context and its representation. The most common representations of context are through ontologies given the qualities that they provide (possibility of reasoning, sharing, and reusing through digital media, etc.) despite their high costs. This representation allows the application of various measures of semantic similarity based on the structure of the ontology and the characteristics of the concepts. In most applications, context is represented by the spatio-temporal information of the user as well as his preferences and interests (recommendation and discovery of services), which is used as a service classification factor according to relevance. Pearson's correlation coefficient is the most frequent semantic similarity measure in the area of the service recommendations using the technique of collaborative filtering, which can be modified to include the contextual information.

CHAPITRE 3

A MEASURE OF SEMANTIC SIMILARITY BETWEEN A REFERENCE CONTEXT AND A CURRENT CONTEXT

Djamel Guessoum[a] , Moeiz Miraoui[b], Atef Zaguia[c], Chakib Tadj[a]

[a]Electrical Engineering Department, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[b]Higher Institute of Applied Science and Technology, University of Gafsa,
2112 Gafsa, Tunisia
[c]Taif University, College of Computers and Information Technology, Saudi Arabia
Al Huwaya, Taif, Saudi Arabia

Cet article a été publié dans le journal « Journal of Ambient Intelligence and Smart Environments » (IOS Press), Vol. 8, no. 6 (2016): 697-707.

Résumé

Les applications sensibles au contexte sont destinées à faciliter l'adaptation des services dans un système informatique diffus. La similarité sémantique entre les contextes et l'application d'une mesure de similarité sémantique comme un mécanisme pour l'adaptation des services sont des sujets qui doivent encore être explorés en profondeur. Dans ce travail nous mesurons les similarités sémantiques entre les variables quantitatives contextuelles et catégoriques dans le domaine de l'informatique diffuse, entre un contexte courant et des contextes de référence qui ont été prédéfinies sur la base d'un ensemble de données contextuelles. Les variables catégoriques sont pondérées à l'aide d'une pondération que nous avons proposé. Elle a la particularité d'être facile à mettre en œuvre et peut être utilisée pour évaluer le poids réel de chaque variable contextuelle.

Mots clés : informatique diffuse, contexte, similarité sémantique, contexte de référence

Abstract

Context-aware applications are intended to facilitate the adaptation of services in a pervasive computing system. The semantic similarity between contexts and the application of a semantic similarity measure as a mechanism for service adaptation are topics that have yet to be thoroughly explored in the literature. This study measured semantic similarities between quantitative contextual and categorical variables in the field of pervasive computing. The measure was applied to a current context and to several reference contexts, which were predefined based on a contextual data set. Built on the overlap measure because of its simplicity, the proposed weighted method is easy to implement and can be used to evaluate the actual weight of each contextual variable.

Keywords: pervasive computing, context, concept, similarity, reference context

3.1 Introduction

Pervasive computing, also known as ubiquitous computing, is based on an idea introduced by Mark Weiser ‘‘the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it,’’ (Weiser M., 1991). Through this concept, computing becomes part of a user’s general environment, extending beyond personal computing to exploit the full potential of computing and communication technologies, to classify complex human behavior, and to react to it in a context-specific way (Augusto J. C. et al., 2010). Thus pervasive computing refers to the full extent of accessibility supplied by an omnipresent computing power that is continuously available to a user, allowing him or her to access a certain application, service, or document, etc., at any time and at any moment (Lino J. A. et al., 2010).

The ultimate objective of such an environment consists in providing appropriate services to a user in a transparent fashion. This goal is achieved by using a dynamic adaptation process in which services are provided to a user on an individual basis, either reactively in response to a change in the current context or proactively by predicting a change and adapting accordingly

(G. Sancho, 2010), or by using explicit and implicit inputs in conjunction with the context in which these inputs are acquired (Lino J. A. et al., 2010). By using the environment and the user profile as a source of information, a pervasive computing system is able to adapt services dynamically in accordance with a specific purpose (J. Simonin and N. Carbonell, 2007).

Several types of mechanisms exist: 1) Rule-based adaptation is a dynamic adaptation mechanism for services that consists in writing a set of logical rules that determine the triggering of a service in a given context. 2) Machine learning adaptation is a dynamic adaptation process for services that detects contextual information. The user-driven selection of appropriate services is then based on some type of learning mechanism. 3) Data mining adaptation is a method that consists in exploring current context data to detect known situations or tendencies in order to provide appropriate services to the user. 4) Comparison-based adaptation consists in comparing conditions linked to the implementation of a service with current context data (Y. Benazzouz, 2011). Each mechanism has its own advantages and disadvantages as well as a particular application domain. Based on the available literature, adaptation mechanisms incorporating a comparison of semantic similarities have rarely been tested for the dynamic adaptation of services in a pervasive computing system.

In pervasive computing, in which the notion of context is very important, the similarity measure serves as a tool to evaluate similarities between instances of a context, which makes it possible to provide the user with the most appropriate services. Various types of similarity measure—depending on the context model and on the representation of the objects that describe this context —have been proposed in the literature (Guessoum, D. et al., 2015), (Saruladha K. et al., 2010). The similarity measure between quantifiable objects (simple objects, vectors, or probability distributions), such as temperature or geographic coordinates, is evaluated by measuring the distance; a vector space is used. Semantic similarity measures are applied for categorical variables (non-quantifiable), such as user activity or user mood.

Our method is based on the concept of case-based reasoning (CBR) (Kolodner, J., 2014), (Leake, D. B., 2003). In CBR, past experiences (reference contexts) are stored and subsequently compared with a currently experienced situation (current context) by using semantic similarity measures in order to select the most similar experience (context) and to provide the user with the corresponding services.

The paper is organized as follows. Section 2 presents related researches and highlights the novelty of our work. Section 3 introduces the notion of context in pervasive computing. Section 4 provides the definition of the reference context and discusses the selection criteria. Section 5 introduces the proposed weighting method for categorical variables. Section 6 provides a case study that serves to evaluate the proposed measure. Conclusions are presented in Section 7.

3.2 Related work

Semantic similarity measures are used in various fields with different types of applications. In pervasive computing, the application of these measures is connected to the concept of "context" and its impact on the adaptation of services provided to the user.

Several studies have applied these measures to service recommendation systems (L. Liu et al., 2010) in which context is represented by the user's profile-related preferences. By including context and allocating a weight to the relationship between concepts, Ning and O'Sullivan (K. Ning and D. O'Sullivan, 2012) developed a similarity measure for the ontological concepts described by Ganesan et al. (2003). Similarly, Miraoui et al. (2013) measured the similarity between a known context and a current context for the adaptation of mobile phone incoming call indications.

Mention should also be given to applications of similarity measures in other domains that may be of interest to the field of pervasive computing; such applications include data mining (El Sayed H. Hacid and D. Zighed, 2008) as well as research by Slimani et al. (2007), who improved on the previous semantic similarity measure of Wu and Palmer (Z. Wu and M. Palmer, 1994) by taking into account the context of the measure.

A pervasive computing system is designed to provide services to a user by minimizing the user's direct involvement. The few existing studies that have applied semantic similarity measures have each provided a particular definition of the context and its specific purpose. Examples include Kirsch-Pinheiro et al. (2008), who proposed a dynamic adaptation of services to solve the problem of incomplete information occurring for a selection process of adequate services in a particular context. Y. Benazzouz (2011) used the same type of similarity measure to cluster data in order to determine the specific situations that trigger a particular service.

A simple semantic similarity measure (overlap measure) was used for the present study. The proposed weighting method for contextual categorical variables aims to provide appropriate services to a user by defining a set of reference contexts (see Section 4) based on the user's location. We introduce the method for weighting contextual categorical variables with this similarity measure, followed by a comparison of our approach with other, better known methods, which we found to be more suitable in our case. The data collected in the current context were compared with data for each reference context, and the semantic similarity measure was then applied to categorical data types. This model allows the comparison of individual concepts (categorical-type variables) unrelated to a taxonomic or ontological structure, which in turn avoids design problems.

However, the model's main difficulty lies in determining the weighting parameters of the concepts to be compared. Although several authors have previously published work on weighting categorical variables—for example, E. S. Smirnov (1968), P. Gambaryan (1964), and others in (S. Boriah et al., 2008)—the weighting methods used in these studies fail to take into account the contribution of each contextual categorical variable concerning the calculation of the semantic similarity measure between two data sets (contexts).

To mitigate some of these shortcomings, we propose a weighting method that assumes that the weight of a characteristic is dependent on how frequently this characteristic occurs for the categorical variables that are compared for each reference context. A large number of

occurrences of a particular characteristic in different reference contexts indicated that it weakly characterized these reference contexts and was not specific to a single reference context, as shown in Section 5.

3.3 Context in pervasive computing

The dictionary defines context as “the set of circumstances or facts that surround an event or a particular situation.” It can nevertheless be shown that this seemingly generic definition contains the essence of the definition of context proposed in most professional domains. Before the term context can be defined, its characteristics should be stated precisely.
The general characteristics of “context” described in (M. Petit, 2005), (Dourish, P., 2004), (Greenberg, S., 2001) can be summarized as follows:

1. There is no context without “context”: the notion of context must be determined according to a purpose. For example, context dynamically adapts to the interactive capacity of a system;
2. A context is an information space used for interpretation: capturing the context is not an end unto itself, but rather the collected data must serve a purpose;
3. A context is an information space shared by several actors: in the proposed case, the user and the system;
4. A context is an infinite and scalable information space: context is not fixed permanently; it evolves over time;
5. It may be difficult to determine the information needed to infer a contextual state. The relevance of any information is highly dependent on the particular situation;
6. Context and activity are separable. The context as a set of features can be encoded and made available to a software system together with an encoding of the activity itself.

Two key features emerge from these common characteristics: 1) the dynamic nature of the context and 2) its purpose.

The following definitions of context focus on these two aspects. P. Brezillon and J.-Ch. Pomerol (1999) defined two concepts related to context: First, a set of contextual data (e.g. time, location) can be used in a decision-making problem; knowledge gained in this way is latent and cannot be used without an objective. Second, context is the product of an emerging objective or intention and requires a large amount of contextual knowledge.

Dalmau et al. (2009) showed that the objective of defining the context in pervasive computing is to enable a context-aware application to discover and react to situational changes. Schilit et al. (1994) considered context to have three important aspects in response to the following questions: Where are you? Who is with you? What resources are available nearby?

Schmidt, A. et al. (1999) categorized context according to six factors. The first three factors relate to the human component, namely information concerning the user (e.g. clothing, biophysical conditions), the social environment (e.g. proximity to other people), and the tasks of the user (e.g. smart tasks of the user). The remaining three factors relate to the physical environment, namely location, infrastructure (e.g. resources, communication), and environmental conditions (e.g. noise, light and climatic conditions).

A.K. Dey, (2001), whose definition is cited most frequently, defines context as “any information that can be used to characterize the situation of an entity (person, object, or physical computing”. This definition clearly resembles that of Schilit et al. because context is considered as a data set collected from a user environment (person), physical environment (physical object), or system environment. The characterization of these environments is the purpose of data collection.

P. Brezillon and J.-Ch. Pomerol (1999) subsequently provided the following definition: “Context is what does not intervene directly in the resolution of a problem but compels its resolution”. This rather generalized definition does not specify the nature of what may compel (constrain) the resolution of a problem.

M. Miraoui and C. Tadj (2007) proposed a service-oriented definition of context according to which relevant information is used to provide appropriate services to a user. This definition, however, includes only the expectations of the user and has many varied applications, because context is defined as any information for which a change in value triggers a service or alters the quality (form) of a service.

Y. Benazzouz (2011) categorized the sources of contextual information: user preferences, behavioral history, physical environment (i.e., ambient temperature, geographical location), as well as the system environment (i.e., applications and networks in which the system functions).
Contextual variables are selected according to the purpose of the application. This study therefore used contextual information that could be collected with mobile equipment (smartphones) and that could be used to determine the specific location of a user.

Table 3.1 provides a general classification of contextual information based on work by Lavirotte et al. (2005) and cited in various other sources (Schmidt, A. et al., 1999), (Perttunen M., 2009), (Topcu F., 2011), (Lee S. et al., 2011), (Zhang D., 2013). This classification was adopted throughout the present paper because of its simplicity and comprehensiveness. The selection was restricted to information relevant to the initial objective of providing appropriate services to a user in a pervasive computing system.

Table 3.1 Context types, variables, and attributes

Context type		**Variable**	**Attribute**
Spatio-Temporal	Space	Type (TP)	covered, open
		Coordinates (CD)	lt_m, lg_m, al_m (latitude, longitude, altitude)
	Time	Day (D)	weekday, public holiday, weekend, vacation
		Hour (H)	5 p.m. < H < 7 a.m. 7 a.m. < H < 5 p.m.

Context type	Variable	Attribute
User	Mobility (MB)	sleeping, walking, running, sitting, standing
Environmental	Temperature (T)	cold, warm, hot
	Surroundings (N)	alone, w/friend(s), w/family, unknown
	Noise level (NS)	silent, quiet, noisy
	Light level (Lm)	dark, dim, bright
System	Internet access (CI)	Cable, Wi-Fi, 3G

3.4 The reference context

A reference context is a context described by predefined contextual information (environmental, user, or system) including the services to be provided to the user or system. In this study, several reference contexts were defined for a particular device (smartphone) to provide a baseline for the similarity measure. These reference contexts were chosen based on the following criteria:

1. The probability (P) of a particular context occurring for a predetermined duration: the higher P, the more likely the context will be a candidate for a reference context;
2. Reference contexts must be sufficiently dissimilar so that the services provided to the user are of differing natures (it is necessary to determine a minimal threshold of semantic similarity (S);
3. Reference contexts should not be chosen in relation to the user's emotional or physiological state (e.g. happy, sad, thirsty, hungry) because these conditions cannot be captured accurately.

Studies on contextual models have shown that a user's location, identity, time, and activity are the most important parameters determining the type of service to provide (G. Chen and D. Kotz, 2000), (Yuan B. and Herbert J., 2014). According to this categorization and to meet the

criteria mentioned previously, user location-based reference contexts were chosen. A user's location can be determined accurately. Environmental, system, and user information is susceptible to change depending on the user's location; to provide the appropriate service; these location-induced changes must thus be taken into account. Furthermore, the user periodically occupies well-defined locations, such as "at home" (nighttime), "at work", or "at school" (daytime). As a first step, the following three locations were chosen as reference contexts:

1. at home;
2. at school;
3. in transit.

Note that this list is merely preliminary and may be incremented each time a new context meets the predefined selection criteria and is sufficiently dissimilar from other contexts.

3.4.1 Context variable

The data set that characterizes a context is collected from several sources of information, for example, physical sensors in the environment, intelligent devices, virtual sensors, Internet access, or even telecommunication service providers; this information is thus very heterogeneous.

In accordance with several previous studies that have addressed the classification of contextual information (L. Liu et al., 2010), (K.C. Gowda and E. Diday, 1992), the present study adopted the following three categories (Figure 3.1):

1. Quantitative variables are expressed in scalar or vector form (i.e., temperature, latitude, longitude, altitude);
2. Quantifiable variables are expressed in qualitative or ordinal form (i.e., large, small, first, second). Quantification is the interpretation of quality in terms of the quantity or projection of an ordinal variable in a metric space; for example, "hot" $\leftrightarrow T > 30℃$ and "first" are projected on a linear axis, in which case "second" directly follows "first.";

3. Categorical variables are not quantifiable. Variables of this type are described as a set of characteristics (e.g. standing, sitting).

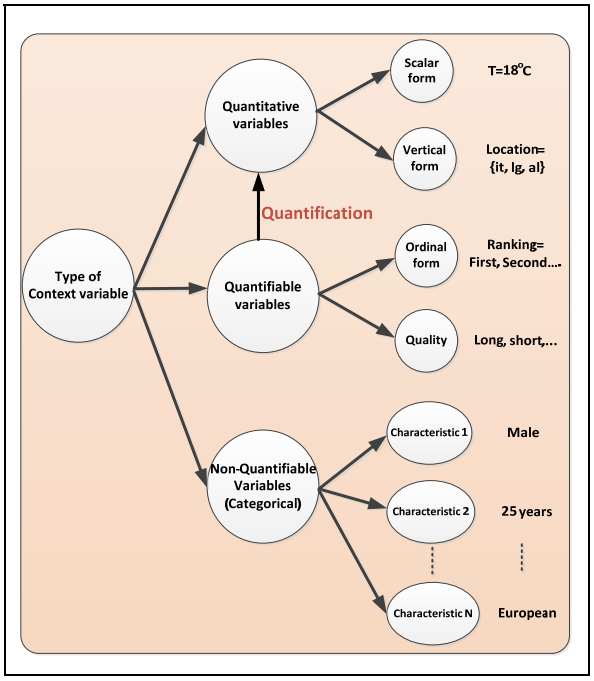

Figure 3.1 Classification of context variables

3.5 Semantic similarity measures

3.5.1 A measure of semantic similarity between a current context and a reference context

Because of the heterogeneous character of both the collected information and the context variables, a series of partial semantic similarity measures must be applied; these partial measures are applied to variables of the same type. Therefore, the measure of overall semantic similarity between the current context and the reference contexts is the weighted sum of these partial measures.

Let two contexts— C_r (reference context) and C_c (current context)—be defined by *n* scalar/vector variables and *m* categorical variables:

$$\begin{cases} C_r = \{Vrs_1, Vrs_2, \dots Vrs_n, Vrc_1, Vrc_2, \dots Vrc_m\} \\ C_c = \{Vs_1, Vs_2, \dots Vs_n, Vc_1, Vc_2, \dots Vc_m\} \end{cases} \quad (3.1)$$

Where Vrs_i is the scalar/vector variable and Vrc_i is the categorical variable of the reference context; $Vs_i$is the scalar/vector variable and Vc_i is the categorical variable of the current context.

Based on the assumption that "two contexts are similar if the context variables of the same type that characterize them are similar," defining the semantic similarity between these two contexts involves determining the similarity between variables of the same type—weighted according to their contribution to the characterization of the reference context—for both contexts.

3.5.2 Weighting of contextual variables

In the literature, the weight of a set of categorical variables is inversely proportional to the number of those variables (S. Boriah, 2008) (Overlap measure, Eskin measure, IOF, OF, etc.). The weighting, which is thus identical for all categorical variables, fails to identify each categorical variable's actual contribution to the semantic similarity measure. Existing measures that account for multiple values of a contextual variable, such as the methods proposed by E. S. Smirnov (1968) and P. Gambaryan (1964), suffer from the same problem.

Moreover, the total number of attributes for this variable affects these two methods. For example, the attributes "sleeping," "walking," "sitting," and "standing" of the categorical mobility variable (MB) characterize the reference context of "home" (n = 4), which can take the following attributes: "sleeping," "walking," "sitting," "standing," and "running" (nt = 5). The weight assigned to each context variable must indicate the contribution of this variable to the characterization of the reference context. Therefore, the more restrictive the value (fewer choices) characterizing the context, the more informative is the variable; it must therefore carry more weight. Consequently, this weighting must not be static and must vary according to context.

For a variable Vrs_i/Vrc_i of the reference context C_r—which can take nt values, but only n ($n \leq nt$) of these values characterize the particular reference context—the weight w_i of the context variable is given as follows:

$$w_i = \frac{(\frac{1}{n_o}.\frac{nt}{n})_i}{\sum_{i=1}^{N}.(\frac{1}{n_o}.\frac{nt}{n})_i} \tag{3.2}$$

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

Where N is the total number of contextual variables, and n_o is the number of occurrences of the attribute *i* of the current context in all reference contexts.

Eq. (3.2) and Figure 3.2 shows that the weight of a context variable is proportional to the ratio $(\frac{nt}{n})$ of the total number of attributes allowed for a contextual variable to the number of attributes characterizing a reference context; this weight is inversely proportional to the number of occurrences of the attribute *i* of the current context in all reference contexts n_o.

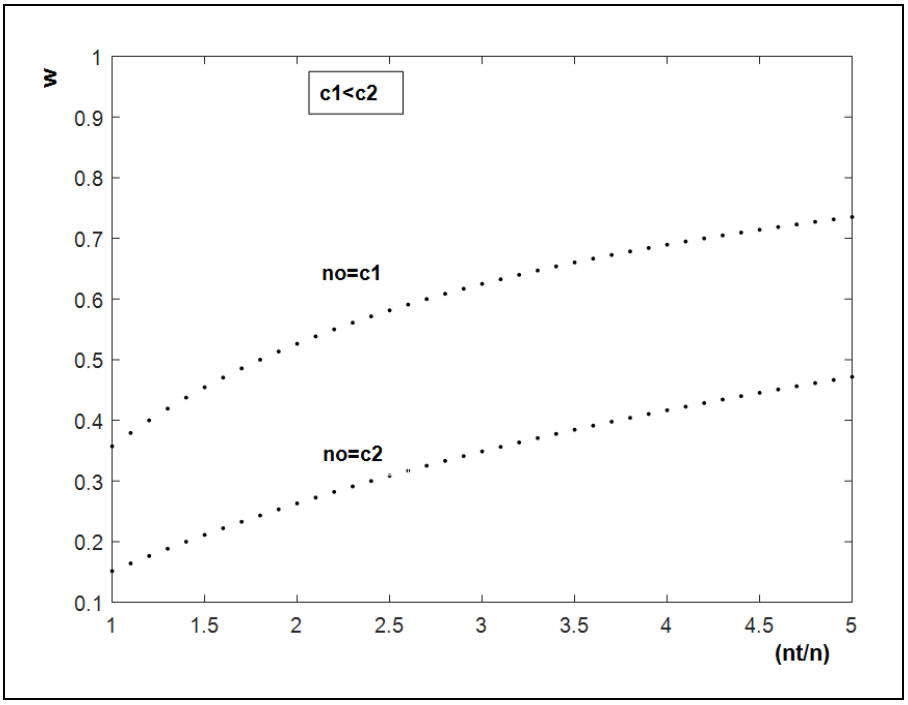

Figure 3.2 Weighting of the contextual variables

3.5.3 Measures of semantic similarity between quantitative and quantifiable variables

The measure of semantic similarity between two quantities is linked to the measurement of the distance between them in a projected space with identical dimensions. For the distance to have a semantic meaning, the quantities must represent a concept, generally specified as an interval (e.g. nighttime = $[7\ p.m., 6\ a.m.]$), low bandwidth = $[0\ kB/s, 128\ kB/s]$). Measuring the semantic similarity between a quantity in the current context and a quantity of the same type in the reference context involves measuring its distance to the interval.

Let Vs and Vrs—two quantities of the same type (except for their location coordinates)—belong to the current context and the reference context as defined in Eq. (3.1).
If the interval limited by $(\mathrm{Vrs_{min}}, \mathrm{Vrs_{max}})$ represents a concept ("warm," "hot," etc.) in the reference context, then the distance to the current context variables Vs and Vrs of the same type is as follows:

$$\begin{cases} \mathrm{d(Vs, Vrs)} = 0 \quad \text{if} \quad \mathrm{Vrs_{min}} \leq \mathrm{Vs} \leq \mathrm{Vrs_{max}} \\ \mathrm{d(Vs, Vrs)} = (\mathrm{Vrs_{min}} - \mathrm{Vs}) \ \text{if} \ \mathrm{Vs} < \mathrm{Vrs_{min}} \\ \mathrm{d(Vs, Vrs)} = (\mathrm{Vs} - \mathrm{Vrs_{max}}) \ \text{if} \ \mathrm{Vs} > \mathrm{Vrs_{max}} \end{cases} \tag{3.4}$$

Furthermore, the semantic similarity between Vs and Vrs is the following:

$$\begin{cases} \mathrm{Sim(Vs, Vrs)} = 1 \ \text{if} \ \mathrm{Vrs_{min}} \leq \mathrm{Vs} \leq \mathrm{Vrs_{max}} \\ \mathrm{Sim(Vs, Vrs)} = \frac{1}{1+\mathrm{d(Vs, Vrs)}} \quad \text{otherwise} \end{cases} \tag{3.5}$$

Example: If a "warm" temperature is defined as $\mathrm{T} \in [20\ ℃, 30\ ℃]$, then $\mathrm{T_1} = 20\ ℃$ is more similar to $\mathrm{T} = 25\ ℃$ than to $\mathrm{T_2} = 18\ ℃$, even though the distance between them would indicate the inverse (Figure 3.3).
Thus, from Eq. (3.4),

$\mathrm{T}_{min} = 20\ ℃ < \mathrm{Vs_1} = \mathrm{T_1} = 25\ ℃ < \mathrm{T}_{max} = 30\ ℃$

$$\mathrm{Vrs_{min}} \leq \mathrm{Vs_1} \leq \mathrm{Vrs_{max}} \Longrightarrow \begin{cases} \mathrm{d(Vs, Vrs)} = 0 \\ \mathrm{Sim(Vs, Vrs)} = 1 \end{cases}$$

$$\mathrm{Vs}_2 = \mathrm{T}_2 = 18\,℃ < \mathrm{T}_{min} = 20\,℃$$

$$\mathrm{Vs}_2 < \mathrm{Vrs}_{\min} \Longrightarrow \begin{cases} \mathrm{d}(\mathrm{Vs}, \mathrm{Vrs}) = (\mathrm{Vrs}_{\min} - \mathrm{Vs}) = 20 - 18 = 2 \\ \mathrm{Sim}(\mathrm{Vs}, \mathrm{Vrs}) = \frac{1}{1+\mathrm{d}(\mathrm{Vs}, \mathrm{Vrs})} = \frac{1}{1+2} = \frac{1}{3} \end{cases}$$

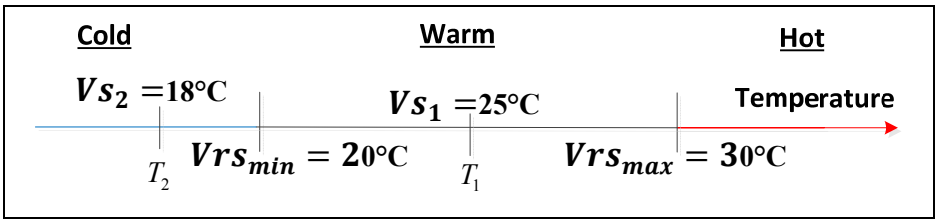

Figure 3.3 Scalar variable for temperature

If the variable Vrs of the reference context relates to geographical coordinates limited by $(\mathrm{Vrs}_{\min}, \mathrm{Vrs}_{\max})$ in three dimensions, then the semantic similarity between Vs and Vrs is

$$\begin{cases} \mathrm{Sim}(\mathrm{Vs}, \mathrm{Vrs}) = 1 & \text{if } \mathrm{Vrs}_{\min} \leq \mathrm{Vs} \leq \mathrm{Vrs}_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

For all quantitative variables, the overall similarity $\mathrm{Sim}(\mathrm{Vs}, \mathrm{Vrs})$ between the current context and the reference context is thus

$$\mathrm{Sim}(\mathrm{Vs}, \mathrm{Vrs}) = \sum_1^N w_i \mathrm{Sim}(\mathrm{Vs}_i, \mathrm{Vrs}_i) \tag{3.7}$$

Where N is the total number of quantitative variables.

3.5.4 Measures of semantic similarity between categorical variables

The measure of semantic similarity between categorical variables aims to quantify the intrinsic characteristics shared by these variables. A characteristic is intrinsic to an object when it defines the nature of the object and cannot be separated from it.

Each reference context $\mathrm{C_r}$ is characterized by a set of categorical variables.

Let: $\mathrm{Vc}_i = \{\mathrm{Car}\} = \{\mathrm{Car}_1, \mathrm{Car}_2 \ldots \mathrm{Car}_n\}$, be one such variable.

To determine the semantic similarity between the categorical variable of the current context Vc_i and the categorical variable of the reference context Vrc_i of the same type, the overlap measure (C. Stanfill and D. Waltz, 1986) was chosen because it is simple to implement for equipment with limited resources such as a smartphone, see Eq. (3.8).

$$\mathrm{Sim}(\mathrm{Vc}_i, \mathrm{Vrc}_i) = \begin{cases} 1 & k \neq 0 \\ 0 & k = 0 \end{cases} \tag{3.8}$$

where k is the number of common attributes shared by Vc_i and Vrc_i.
For all categorical variables, the overall similarity $\mathrm{Sim}(\mathrm{Vc}, \mathrm{Vrc})$ between the current context and the reference context is thus

$$\mathrm{sim}(\mathrm{Vc}, \mathrm{Vrc}) = \sum_1^N w_i \mathrm{Sim}(\mathrm{Vc}_i, \mathrm{Vrc}_i) \tag{3.9}$$

where N is the total number of categorical variables.

3.5.5 Overall semantic similarity

The overall semantic similarity between the current context C_c and the reference context $\mathrm{C_r}$ is as follows:

$$\mathrm{sim}(\mathrm{C_c}, \mathrm{C_r}) = (\mathrm{Sim}(\mathrm{Vs_i}, \mathrm{Vrs_i}) + \mathrm{Sim}(\mathrm{Vc_i}, \mathrm{Vrc_i}))/2 \tag{3.10}$$

where $\mathrm{sim}(\mathrm{Vs}_i, \mathrm{Vrs}_i)$ is the semantic similarity between the i^{th} variable (quantitative or quantifiable) in the reference context C_r and the i^{th} variable of the same type in the current context C_c, and $\mathrm{Sim}(\mathrm{Vc}_i, \mathrm{Vrc}_i)$ is the semantic similarity between the i^{th} variable (categorical) in the reference context C_r and the i^{th} variable of the same type in the current context C_c.

$$\mathrm{sim}(\mathrm{C_c}, \mathrm{C_r}) \in [0,1]$$

3.6 Case study

To provide an example of an application and to illustrate the weighting of the contextual variables, the measure of semantic similarity between a current context C_c and reference context C_r was applied to the data set[2].

This data set consists of feature files for 43 different recording sessions. In each recording session the same user—carrying a mobile phone, sensor box, and laptop computer—was commuting from home to the workplace or vice-versa. During the commute the user walked, used some kind of public transportation (bus or Metro), and sometimes drove a car. Occasionally, the user took either slightly different or considerably different routes/modes of transport. During the session, triaxial acceleration sensors recorded atmospheric pressure, temperature, humidity, etc. Ambient audio levels were recorded on a laptop computer equipped with a microphone and a sound card. On the mobile phone, changes in the user's location were recorded in the form of a Cell ID and a location area code obtained through the GSM network.

Five sessions were randomly selected, with each session characterized by the following contextual variables: categorical variables (Day Name (Saturday, Sunday…(1-7)), Day Period (night, morning, afternoon, evening (1–4)), User Activity (1–6)), quantitative variable (Day Date (1–31), Temperature (1–10), Relative Humidity (1–10), Pressure (1–10), Average Audio Level (1–5)).

These five selected sessions included seven different locations; several hundred values of contextual variables were recorded at each location.

To show the relevance of the weighted calculation, a current context was selected based on the recordings characterizing “Location 0.” Results from four locations (Loc. 0, Loc. 1, Loc. 2, and Loc. 3) were compared (Table 3.2), far right column, shows the total number of values

[1]The data set NokiaContextData, obtained from the Institut für Pervasive Computing (Johannes Kepler University Linz, Austria), is available at http://www.pervasive.jku.at/Research/Context_Database/

that a context variable can take (nt); the "Reference context by location" columns show the contextual variable values (n) characterizing each location (reference context). The column labeled "Current context" contains the current value of the contextual variable.
The values of the categorical context variables were then modified (Day Name, Day Period, and User Activity) to show the effect of weighting these variables (Table 3.4).
The following semantic similarity algorithm was applied:

Step 1: Determine the weight of the contextual variables (w_{DN},w_{DD}, w_T,w_H,w_A, w_P, ws_{DP},ws_{Act}), see Table 3.3.

$$w_i = \frac{(\frac{1}{n_o} \cdot \frac{nt}{n})_i}{\sum_{i=1}^{N} \cdot (\frac{1}{n_o} \cdot \frac{nt}{n})_i}$$

Step 2: For every reference context, determine the semantic similarity with the current context between the context variables of the same type.

For the reference context "Location 0," measure the semantic similarity between quantitative or quantifiable variables (Day Date, Temperature, Relative Humidity, Pressure, and Average Audio Level), see Eq. (3.7).

Similarly, measure the semantic similarity between categorical variables (Day Name, Day Period, User Activity), see Eq. (3.9).

Because the data set does not contain discrete values, the similarity between the quantitative and categorical variables is calculated by measuring the overlap, see Eq. (3.8).

Step 3: Calculate the overall semantic similarity, Eq. (3.10).

Table 3.3 shows that good agreement can be achieved with the proposed weighted method by using w = $\frac{1}{d}$ (Overlap, Eskin, IOF, OF, Burnaby, Goodall 1,2,3,4) (Desai, A. et al., 2011) to

measure the semantic similarity. To show the effect of the proposed weighting method, we modified the number of occurrences n0 and the ratio ($\frac{nt}{n}$) of the categorical context variables (Day Name, Day Period, and User Activity) as follows:

Day Name (no = 1, $\frac{nt}{n} = 2$)

Day Period (no = 4, $\frac{nt}{n} = 2$)

User Activity (no = 4, $\frac{nt}{n} = 1$ and $\frac{nt}{n} = 1.2$)

In accordance with these values, The "Day Name" variable must have a more significant weight than either the "Day Period" or the "User Activity" variable.

Table 3.4 shows that the most significant contextual variable (Day Name) has a weight that corresponds to its significance and that the other contextual variables are weighted lower.

Table 3.2 Attributes of contextual variables by location

Contextual variable	Current context	Reference context by location Number of attributes characterizing the reference context (n)				Total number of attributes (nt)
		Location 0	Location 1	Location 2	Location 3	
Day Name	2	(1.2.3.4)	(1.2)	(1.2.3.4)	(1.2)	7
Day Date	4	(3.4.5.6.10)	(3.4.10)	(4.5.6.10)	(4.10)	31
Day Period	4	(2.4)	(2.4)	(2)	(2)	4
Temperature	9	(4.5.6.7.8.9)	(4.5.6.7.8.6)	(4.5.6.7.8.9)	(8.9)	10
Relative Humidity	5	(1.2.3.4.5.6.7)	(1.2.3.4.5.6.7.8)	(1.2.3.4.5.6.7.8.9)	(6.9)	10
Pressure	10	(1.2.3.5.6.7.8.9.10)	(1.2.3.8.9.10)	(1.2.3.4.5.7.8.9.10)	(9.10)	10
User Activity	6	(1.3.4.5.6)	(1.2.3.4.5.6)	(1.2.3.4.5.6)	(3.4)	6
Average Audio Level	1	(1.2.3.4)	(1.2.3.4)	(1.2.3.4)	(3.4)	5

NOTE: Contextual variable names were taken from and are identical to the original data set.

Table 3.3 Overall semantic similarity

	Overall Similarity w =1/d	**Overall Similarity** $\mathbf{w}=\frac{1}{\sum_{k=1}^{d} n_k}$	**Overall Similarity** $\mathbf{w}=\frac{(\frac{1}{n_o}\cdot\frac{nt}{n})}{\sum_{i=1}^{N}\cdot(\frac{1}{n_o}\cdot\frac{nt}{n})}$
New location-Location 0	1	0.268	1
New location-Location 1	0.9	0.175	0.935
New location-Location 2	0.833	0.203	0.646
New location-Location 3	0.466	0.450	0.4

Table 3.4 Weight of contextual variables by location

Context	**Categorical variable**	$\mathbf{w}=\frac{1}{d}$ **(Overlap, Eskin, IOF,OF, Burnaby, Goodall1,2,3,4) [12]**	$\mathbf{w}=\frac{1}{\sum_{k=1}^{d} n_k}$ **(Smirnov, Gambaryan) [12]**	$\mathbf{w}=\frac{(\frac{1}{n_o}\cdot\frac{nt}{n})}{\sum_{i=1}^{N}\cdot(\frac{1}{n_o}\cdot\frac{nt}{n})}$ **(Proposed method)**
Loc. 0	Day Name	0.333	0.125	**0.714**
	Day Period	0.333	0.125	**0.178**
	User Activity	0.333	0.125	**0.107**
Loc. 1	Day Name	0.333	0.111	**0.727**
	Day Period	0.333	0.111	**0.181**
	User Activity	0.333	0.111	**0.090**
Loc. 2	Day Name	0.333	1	**0.727**
	Day Period	0.333	0.5	**0.181**
	User Activity	0.333	0.6	**0.090**
Loc. 3	Day Name	0.333	0.125	**0.714**
	Day Period	0.333	0.125	**0.178**
	User Activity	0.333	0.125	**0.107**

The number of contextual variables is denoted by d, and n_k is the number of attributes allowed for each contextual variable.

These results suggest that the proposed approach is applicable to measuring the importance of contextual variables and to providing a true assessment of the contribution of each individual variable while remaining simple to implement (thanks to the Overlap measure) with resource-limited equipment such as smartphones.

Figure 3.4 shows the dynamic character of the proposed approach to calculating the weight w in comparison with most approaches advocated in the literature (w = 1/d = 1/3 and $\mathbf{w=\frac{1}{\sum_{k=1}^{d} n_k} = 1/8}$).

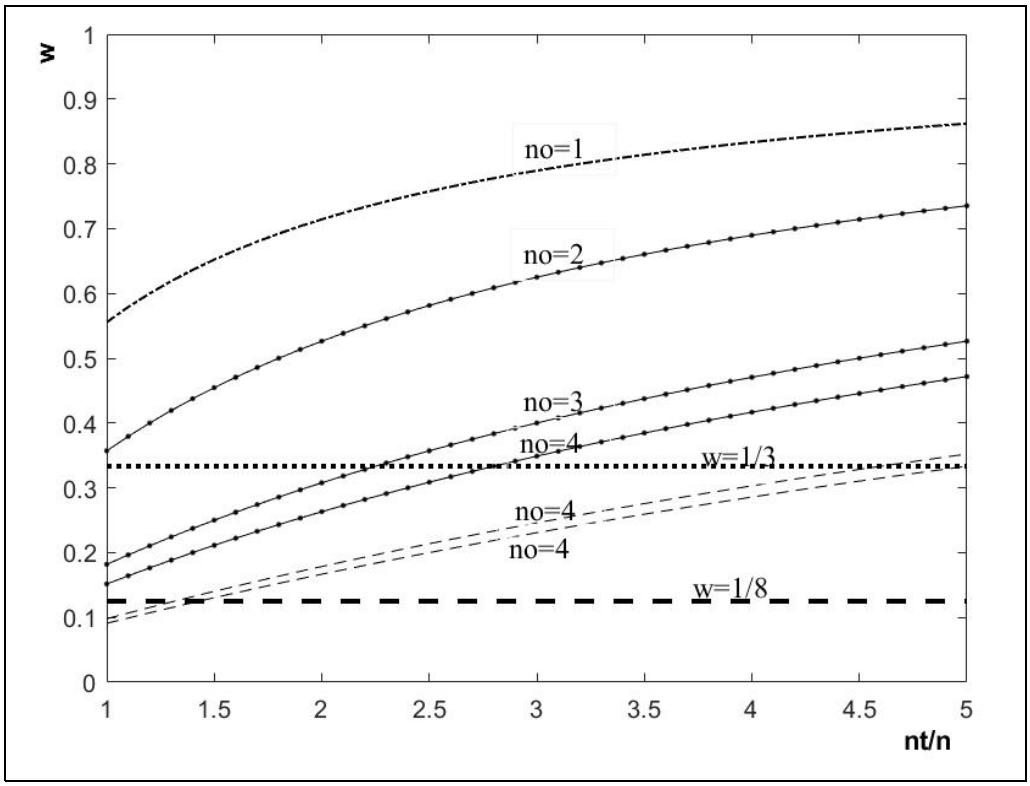

Figure 3.4 Weight changes according to no and nt/n

The proposed approach relies on the number of occurrences of the contextual variable in all reference contexts as well as on the ratio ($\frac{nt}{n}$). This ratio increases with a decreasing number (n) of attributes characterizing a categorical variable.

3.7 Conclusion

Based on a simple measure of semantic similarity, the proposed weighting method was used as a mechanism for service adaptation in a pervasive computing system by defining a type of context known as a "reference context" to serve as a basis of comparison with the current context.

The proposed methodology—despite its simplicity—is very intuitive and provides a realistic model for weighting the contextual variables of the semantic similarity measure in a pervasive computing system. However, the criteria for selecting the reference contexts may be improved to achieve more flexible and more dynamic guidelines and to include several types of context. It would therefore be useful to formulate the definition of partial reference contexts within the global reference context, (e.g. "school" as the global reference context and "library," "classroom," etc., as partial reference contexts).

The selection of variables that define a context in general should be revised, as should the selection of variables and attributes characterizing a reference context.

The proposed weighting method is applicable to our case only. Further research is needed to arrive at conclusions that generalize to other applications in other areas.

CHAPITRE 4

CONTEXTUAL CASE BASED REASONING APPLIED TO A MOBILE DEVICE

Djamel Guessoum[a], Moeiz Miraoui[b], Chakib Tadj[a]

[a]Electrical Engineering Department, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[b]Higher Institute of Applied Science and Technology, University of Gafsa,
2112 Gafsa, Tunisia

Cet article a été soumis dans « International Journal of Ambient Computing and Intelligence » (IGI Global) le 24 Mars 2016.

Résumé

La présente étude est une application du raisonnement par cas contextuel à un équipement intelligent mobile. Le raisonnement par cas a été choisi car il ne nécessite pas de phase d'apprentissage, exige des ressources de traitement minimes et s'intègre facilement avec la nature dynamique et incertaine de l'informatique diffuse. Selon la localisation et l'activité d'un utilisateur mobile, qui peuvent être déterminées par les capteurs inertiels du dispositif et les capacités GPS, il est possible de sélectionner et d'offrir des services appropriés à cet utilisateur. L'approche proposée comporte deux étapes : La première étape utilise des mesures de similarité sémantique simples pour extraire le cas qui correspond le mieux au contexte courant de la base des cas. Dans la deuxième étape, la sélection de services obtenue est ensuite filtrée par les informations contextuelles actuelles. Cette méthode en deux étapes ajoute un niveau plus élevé de pertinence des services proposés à l'utilisateur en restant facile à mettre en œuvre sur un appareil mobile.

Mots clés : Raisonnement par cas, sensibilité au contexte, informatique diffuse, GPS, équipement mobile, services

Abstract

The present study applied case-based reasoning (CBR), a problem solving process that can be used as a machine-learning method, to the context of mobile devices. The CBR method was chosen because it does not require training, demands minimal processing resources, and easily integrates with the dynamic and uncertain nature of pervasive computing. Based on a mobile user's location and activity, which can be determined through the device's inertial sensors and GPS capabilities, it is possible to select and offer appropriate services to this user. The proposed approach comprises two stages: The first stage uses simple semantic similarity measures to retrieve the case from the case base that best matches the current case. In the second stage, the obtained selection of services is then filtered based on current contextual information. This two-stage method adds a higher level of relevance to the services proposed to the user, yet it is easy to implement on a mobile device.

Keywords: Case-based reasoning, context-awareness, Pervasive computing, GPS, mobile device, services.

4.1 Introduction

Technological advances in the collection of contextual information (smartphone, social media, etc.) have resulted in the proliferation of context-aware applications aimed at offering relevant services to a user by adapting a service's behavior to changes in the environment. The origin of the term "context awareness" is attributed to Schilit & Theimer (1994) who asserted that context sensitivity is "the ability of a mobile application and/or of a user to discover and react to changing situations".

Before continuing this discussion, it is necessary to define the term "context" and to clarify how this concept differs from the terms "contextual information" and "situation." According to (Brézillon & Pomerol, 1999), context is a set of contextual data (e.g. time, location) that can be used in a decision-making process. Schilit et al. (1994) considered context to have three important aspects in response to the following questions: Where are you? Who is with

you? What resources are available nearby? The authors thus categorized context according to six factors. The first three factors relate to the human component, namely information concerning the user (e.g. clothing, biophysical conditions), the social environment (e.g. proximity to other people), and user tasks (e.g. smart tasks). The remaining three factors relate to the physical environment, namely location, infrastructure (e.g. resources, communication), and environmental conditions (e.g. noise, light, and climatic conditions).

Abowd, Dey, Brown, Davies, & Steggles (1999) whose definition is cited most frequently, defines context as "any information that can be used to characterize the situation of an entity (person, object, or physical computing). This definition clearly resembles that of Schilit et al. (1994) because context is considered a set of data collected from a user environment (person), physical environment (physical object), or system environment. The characterization of these environments is the purpose of data collection. This rather generalized definition does not specify the nature of what may compel (constrain) the resolution of a problem.

The present study is based on works by (Meissen et al., 2005; Kayes et al., 2014) in which context is considered a snapshot or an instantiation of all context variables (e.g. contextual information such as location, temperature…) at a specific moment in time. Context thus differs from situation, which is the result of introducing semantic relations between these parameters (e.g. being at home, on the phone) and which may be a series of contexts that do not change over time. Studies on contextual models have shown that a user's location, identity, time, and activity are the most important parameters that determine the type of service to provide to a user (Yuan, 2014; Benazzouz, 2011; Bolchini, 2007).

Case-based reasoning (CBR), which can be applied to machine learning, is a method that solves a common problem based on known solutions to similar problems encountered in the past. We chose to use CBR because it allowed us to save an initial number of cases containing the specific contexts and their corresponding services to a "seed" case base. These cases serve as a reference to be compared with a new case or a current context in order to

identify the most similar case encountered in the past and to subsequently provide appropriate services to the user. The proposed services are then updated in a second stage that filters these services based on information obtained for the current context (e.g. battery level, ambient light conditions…). This service adaptation method is the main contribution of the present paper.

The CBR method presents the following main advantages: It does not require prior knowledge of the new problem field, does not involve training, demands minimal processing resources (Kofod-Petersen & Aamodt, 2003), is capable of solving problems in poorly defined fields, and easily integrates with the dynamic and uncertain nature of pervasive computing (Öztürk & Aamodt, 1998). CBR originated in the cognitive sciences, specifically in research on human memory (Schmid & Richter, 2006). Unlike most problem solving methodologies in artificial intelligence (AI), CBR is memory-based; it thus relies on the human use of remembered problems and solutions as a starting point for solving new problems (Schank, 1983).

The literature proposes several definitions of the CBR cycle that are in agreement with the cycle described in (Kolodner, 1996), which can be summarized as follows:

- case retrieval **:** A new case consists of a request containing a problem that is similar to the cases stored in the case base.The case (or cases) most similar to the request is then retrieved from the case base;
- adapting (reusing) the solution **:** The retrieved cases and solutions are adapted to fit the solution;
- assessing the solution **:** The fit of the adapted solution is evaluated;
- updating the case base **:** If the retrieved/adapted solution is acceptable, it will be added to the case base.

CBR uses analogy to solve new problems based on the concept that *"*similar problems have similar solutions*"* (Leake & Jalali, 2014). Thus, if a case is a pair (*P*, *S*), where *P* is a problem and *S* is the solution to the problem, and if (*P*, *S1*) and (*P*, *S2*) then *S1 = S2* (Richter,

1995). Initially, cases containing the description of a problem (see 'problem space' in Figure 4.1) and the corresponding validated solution for a specific field (see 'solution space' in Figure 4.1) are carefully selected and then saved to a "seed" case base. The solution to a new problem in the same field is based on comparing the new problem description with the description of stored cases: The cases (or case) with the most similar descriptions are retrieved based on the minimum retrieval distance R (see Figure 4.1), so that their solutions can be applied to the new problem. Cases extracted during the retrieval phase that are considered unsuitable require adaptation. The adaptation stage uses the minimum adaptation distance A (see Figure 4.1) between the extracted case(s) and the new case.

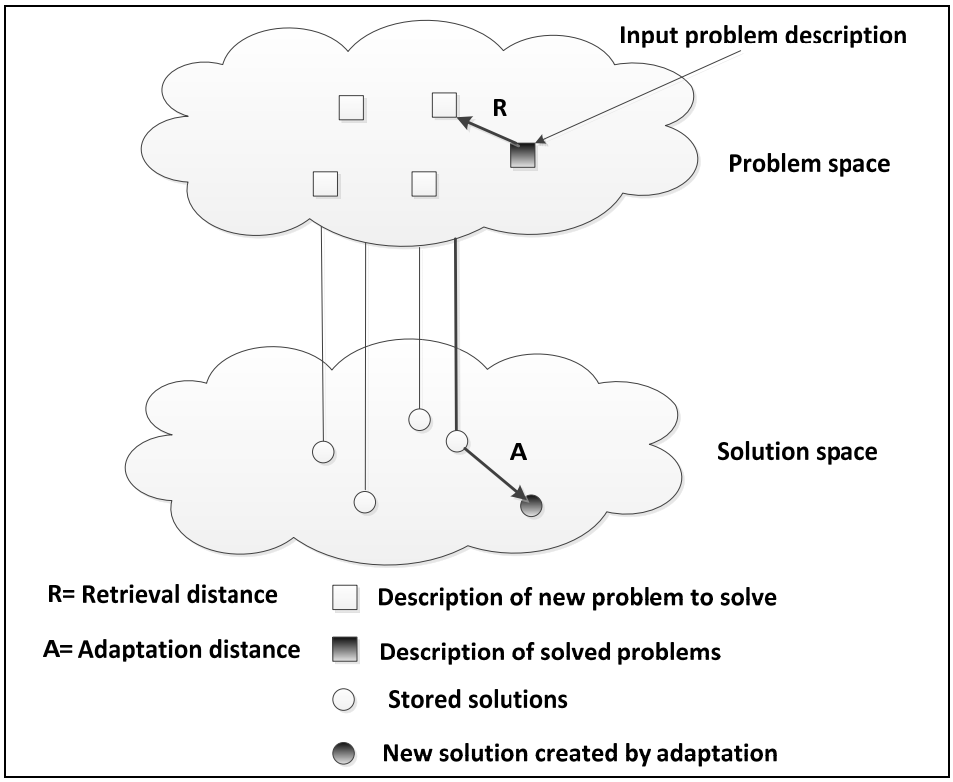

Figure 4.1 Relationship between problem and solution spaces in CBR
Adapted from De Mantaras et al. (2005)

The present paper is organized as follows: Section 2 presents related research and highlights the originality of our approach. Section 3 presents the proposed method for the adaptation of services based on CBR. Section 4 describes the experimental setup and the application of the two-stage service filtering process. Conclusions are presented in Section 5.

4.2 Related Work

In pervasive computing, the adaptation of services is a dynamic process wherein services are offered either reactively to a user in response to a change in context or proactively by predicting a change in context and acting accordingly (Guessoum, Miraoui & Tadj, 2015; Sancho, 2010). Several definitions are proposed in the literature; the most general definition (Efstratiou, 2004) generalizes the concept of adaptation for mobile equipment and context-aware applications in a pervasive computing system, based on the premise that an application or system is adaptive when it changes its behavior in response to a change in context (this change occurs in either the context or in the equipment resources).

Nicklas et al. (2008) categorized the adaptation of context-aware applications into four classes: 1) the selection of information and services, 2) the presentation of information and services, 3) the automatic execution of a service for a user, and 4) the labeling of contexts for later retrieval.

The main objective of the service adaptation process is to suggest the most appropriate service(s) to a user. The present study used the CBR method (for reasons stated above) to adapt services on a mobile device according to a user's current context. In general, CBR systems have incorporated contextual information as an adjunct to information previously acquired in a specific field 1) to improve performance of similarity calculations during the retrieval phase by reducing case search volume (Nwiabu, Allison, Holt, Lowit & Oyeneyin , 2011; Montani, 2011 ;Lee & Lee, 2007) and 2) to categorize the retrieved cases according to the relevance of the proposed solutions as well as adapt those solutions to a user's personal constraints (Montani, 2011; Abdrabou & Salem, 2010; Pla, Coll, Mordvaniuk & López, 2014; Kofod-Petersen & Aamodt, 2009). This contextual information varies depending on the field of application. Note that this contextual information (location, temperature, humidity, lighting conditions, user profile, etc.) has always been used separately (Song, Moon & Bae, 2013; Kwon & Sadeh, 2004). Furthermore, for rare cases in which contextual information has been used to describe an entire case, this use was limited by the field of

application (smart home (Ma, Kim, Ma, Tang & Zhou, 2005), museum visits - LISTEN project (Zimmermann, 2003).

Most studies using contextual information in conjunction with CBR have incorporated context into the cases to represent a past learning experience specific to the context for which this past experience can be used to determine the relevance of the proposed solution (Leake & Jalali, 2014; Pantic, 2005). Context is used as additional information for a given field of application in the various stages of CBR (Retrieve, Reuse, Revise, and Retain) (see Figure 4.2) (Kofod-Petersen & Aamodt, 2009; Leake & Jalali, 2014). In (Gouttaya & Begdouri , 2011) the authors used the ECA (event, condition, action) structure of active rules for adaptation but neglected the time factor of an event or of a change in context. In (Zimmermann, 2003) context was considered to represent a description of the problem, and the solution was regarded as the appropriate recommendation. The discretization of the continual acquisition of contextual attributes is transferred to a case in CBR.

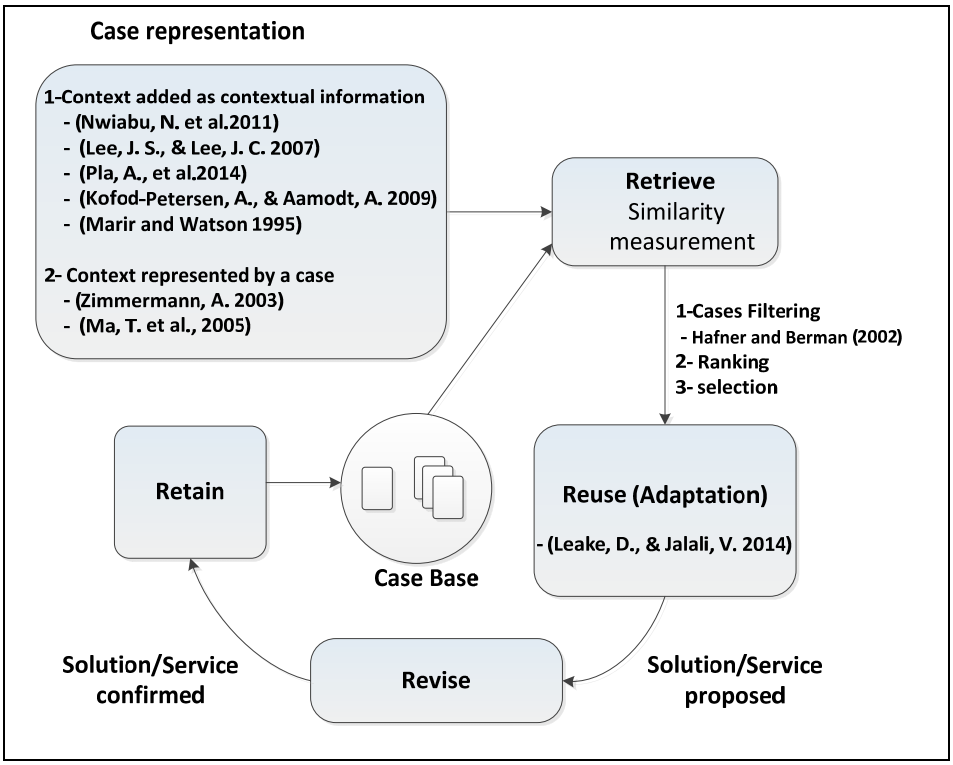

Figure 4.2 Context and the CBR cycle

Published research that considers context and uses CBR as a machine-learning method to select services (see ‘case representation’ in Figure 4.2) can be classified into two categories:

- studies in which context (contextual information) is used as additional information for the case structure;
- studies in which a case represents a specific context.

In both categories, the proposed services result from the retrieval/adaptation phase, and there is no guarantee that these services will be the most appropriate options. The present paper proposes a two-stage retrieval and service-filtering process to improve the relevance of the services offered to the user. Moreover, the study used methods that require minimal processing resources: CBR for the learning phase, the Wu and Palmer similarity measure to identify similarities between different user locations, the Euclidean distance to determine similarities between various user activities, and the overlap coefficient to measure similarities for the contextual information attribute “time.”

4.3 Our Approach

The proposed approach is based on the following hypothesis analogous to the assumption that “similar problems have similar solutions” (Leake & Jalali, 2014), which forms the basis of CBR: “In a context-aware pervasive computing system, similar services are provided in similar contexts/situations.” To apply the same methodology to a pervasive system, we thus sought to answer the following question: Based on our knowledge of the user’s current context/situation, what is the most appropriate service we can provide to the user?

As mentioned previously, we chose CBR over other reasoning and AI inference techniques because it offers advantages that fit our target application and it fulfills the following requirements: The CBR cycle 1) does not require training, 2) demands minimal processing resources (Kofod-Petersen & Aamodt, 2003), and 3) easily integrates with the dynamic and uncertain nature of pervasive computing (Knox, Coyle & Dobson, 2010).

Based on the CBR cycle, the proposed approach intends to provide appropriate customized services to smartphone users. This approach is summarized in Figure 4.3.

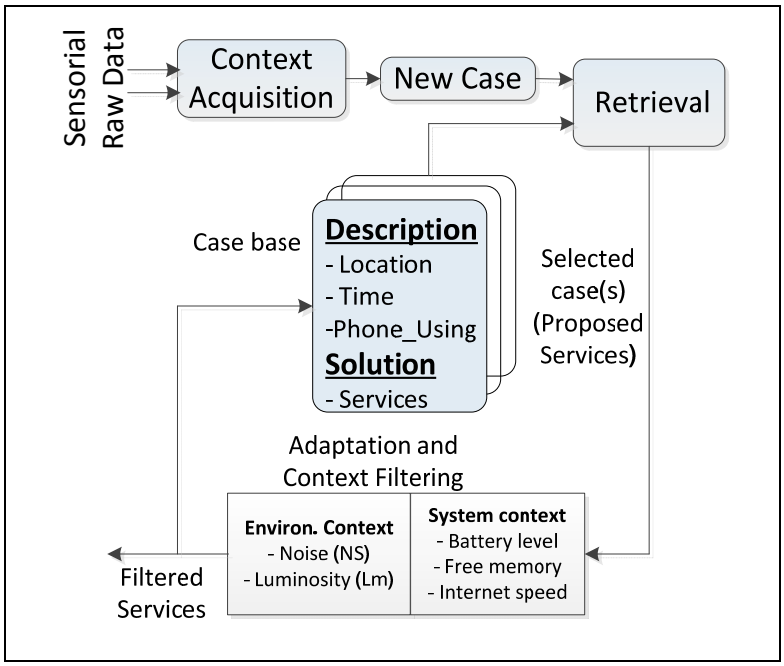

Figure 4.3 Context acquisition and service filtering

Before presenting the CBR cycle, we first describe the representation of cases employed in the present study.

4.3.1 Case Representation

A case is a description of a problem and its corresponding solution. In general, the three techniques for presenting cases most often found in the literature (Watson, 1999) are:

- flat organization **:** This simple organizational technique retrieves on a case-by-case basis and works well for small case bases;
- clustered organization **:** Cases are stored in clusters of similar cases, which allow easy cluster selection for matching; however, this technique requires a complex algorithm to add/delete cases;

- hierarchic organization : A hierarchic organization groups cases that share the same characteristics. Cases are organized in the form of a structured network of categories possessing semantic relationships. This type of organization permits a precise and rapid retrieval of cases; however, new cases are relatively difficult to add or delete.

The selection of contextual information used to represent a case is limited by several factors: the types of sensors current smartphones are equipped with, the processing power of current smartphone technology, the available storage capacity, and the potential constraints of processing parallel requests (Incel, Kose & Ersoy, 2013).

Studies on contextual models have shown that a user's location, identity, time, and activity are the most important parameters that determine the type of service to provide. In many cases, these parameters correlate strongly: An activity can, for example, be determined according to the user's location (e.g. if the location is L= restaurant, the activity is A = eating) (Phithakkitnukoon, Horanont, Di Lorenzo, Shibasaki & Ratti, 2010; Zhu & Sheng, 2011); it can be determined by using the smartphone's inertial or GPS sensors (e.g. the speed at which a user is moving can reveal whether the user is walking, running, or using another means of transportation) (Incel, Kose & Ersoy, 2013; Chon & Cha, 2011); and finally, an activity can be determined based on which functionality of the smartphone is being used (messaging, calling, browsing, playing games) (Chon & Cha, 2011). Accelerometers, gyroscopes, and GPS sensors constitute an effective means of inferring even complex human behavior and identifying significant locations in people's everyday lives (at work, at home, at the coffee shop). (The techniques used to acquire this information have been described elsewhere (Schmid & Richter, 2006; Cao, Cong & Jensen, 2010).

Table 4.1 provides an overview of the typical case structure (case = context, service) that follows from the preceding arguments.

Table 4.1 Case structure

<table>
<tr><td rowspan="11">Description</td><td colspan="3">1. Location</td><td>Collected Data</td></tr>
<tr><td>Location_eating</td><td colspan="2">restaurant, bakery, coffee shop</td><td rowspan="9">- location
- speed
- accel. x/y/z (mean, SD, VAR)</td></tr>
<tr><td>Location_education</td><td colspan="2">university, college, institute, school</td></tr>
<tr><td rowspan="2">Location_entertainment</td><td>Indoor</td><td>cinema, bowling theater</td></tr>
<tr><td>Outdoor</td><td>stadium</td></tr>
<tr><td rowspan="2">Location_variable</td><td>Motorized</td><td>car, bus, plane, train, subway</td></tr>
<tr><td>Non_motorized</td><td>walking, running, biking</td></tr>
<tr><td colspan="3">Location_home, Location_recreational, Location_shopping, Location_work, Location_ government, Location_worship</td></tr>
<tr><td colspan="3">2. Time</td><td rowspan="2">date
time</td></tr>
<tr><td colspan="3">daytime, nighttime, week day, week end</td></tr>
<tr><td colspan="3">3. Phone_Usage</td><td rowspan="2">phone usage</td></tr>
<tr><td colspan="3">messaging, calling, browsing, gaming, nil</td></tr>
<tr><td>Services</td><td colspan="4">Services =(rt, rv, pm, lm, apps)*</td></tr>
</table>

* rt = ring tone, rv = ringer volume, pm = phone mode, lm = luminosity, apps = recommended applications

The chosen case structure is hierarchic (see Figure 4.4), permitting rapid access to the various attributes (Ma, Kim, Ma, Tang & Zhou, 2005) as well as facilitating retrieval. The case description is composed of a user's location, time, and phone usage. The selection and classification of appropriate services is the subject of future work; here, we have focused our attention on services chosen specifically to explain the proposed approach.

4.3.2 Context acquisition

The following sensors can be used to obtain contextual data with a smartphone (Incel, Kose & Ersoy, 2013): accelerometer, gyroscope, camera, GPS, bluetooth, Wi-Fi,

microphone, digital compass, ambient light and proximity sensors. The user's location, the current time, and characteristics retrieved from the accelerometer and the gyroscope are determined during the acquisition phase. If one of these characteristics matches a closely defined condition, then a new case is detected and the CBR cycle is initiated (e.g. if the current location differs from the previous location, which was "home," then the newly detected case contains the current location, the acquisition time, and the characteristics obtained from the accelerometer and the gyroscope).

The context acquisition stage consists in retrieving the structural and statistical characteristics of the samples collected from the sensors (mean, variance, etc.). Time and frequency constitute the two main types of characteristics. Temporal characteristics are used most often because of their computational simplicity, in particular with respect to calculating the mean, the variance (VAR), and the standard deviation (SD) (Shoaib, Bosch, Ince, Scholten & Havinga, 2015).

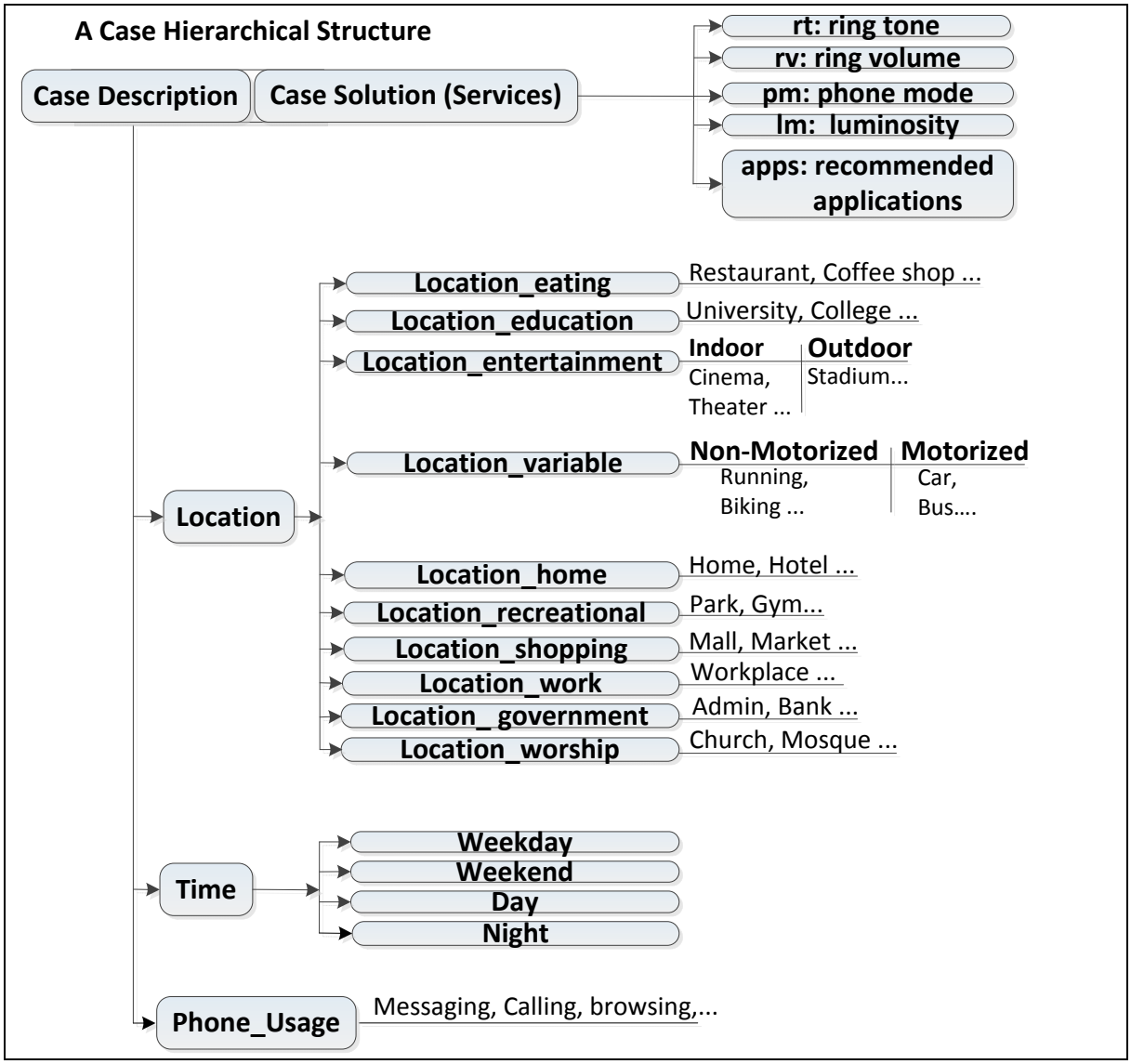

Figure 4.4 Hierarchic structure of a case

4.3.3 Service filtering

The retrieval phase of the CBR cycle yields a case (or multiple similar cases) with *n* services. An adaptation or contextual filtering module filters these services according to relevance based on current smartphone resources and the environmental context (such as lighting conditions, noise levels, etc.) and by using a rule-based system (*if* context *then* service).

4.3.3.1 Case Retrieval and Similarity Measure

It is important to choose similarity measures that meet the requirements of the CBR process because the nature of the selected measure affects the quality of the cases obtained during the retrieval phase (Gabel & Stahl, 2004). It has been shown that the efficiency of the retrieval process for cases that are similar to the new case depends on two factors (Zang, Gray, Hobbs & Pohl, 2008): 1) the type of similarity chosen and 2) the organizational structure of the cases.

In CBR, the global approach to measuring the similarity between cases is primarily based on calculating local similarities between attributes, which can be customized for each case base (Richter & Weber, 2013). The global similarity (Eq. 4.1) can then be calculated based on these local similarities by weighting each attribute:

$$Similarity(Case_{new}, Case_{old}) = \frac{\sum_{i=1}^{n} w_i \times Sim(a_i^{Case_{new}}, a_i^{Case_{old}})}{\sum_{i=1}^{n} w_i} \quad (4.1)$$

Where w_i is the weight of the attribute a_i, $a_i^{Case_{new}}$ is the attribute i of the new case, and $a_i^{Case_{old}}$ is the attribute i of the existing case in memory. Given the conditions of the present study, the limited resources of mobile devices were an important consideration for the selection of a suitable local similarity measure. We chose the semantic similarity measure proposed by Wu & Palmer (1994) (Eq. 4.2) because of its ease of calculation and its applicability to user location and phone usage taxonomies (see Figure 4.6). We nevertheless modified the Wu and Palmer measure to eliminate an inherent disadvantage, in which two concepts in the same hierarchy may show a lower similarity than two concepts belonging to different hierarchies (Richter & Weber, 2013; Shet & Acharya, 2012).

The modified approach resolves this problem while preserving the method's simplicity (see Eq. 4.3).

$$sim_{WP}(c1, c2) = \frac{2N}{N1+N2} \quad (4.2)$$

Where N is the number of edges between the taxonomy root and the LCS (least common subsumer) of the concepts *c1* and *c2*, *N1* is the number of edges between the root and *c1*, and *N2* is the number of edges between the root and *c2*.

4.3.3.2 Modified similarity measure

If two concepts are linked directly to the same LCS without any other intermediary concept $(N1 - N) = 1\ and\ (N2 - N) = 1$), such as in Figure 4.5a where the two concepts are "cinema" and "theater" and the LCS is the concept "indoor,"

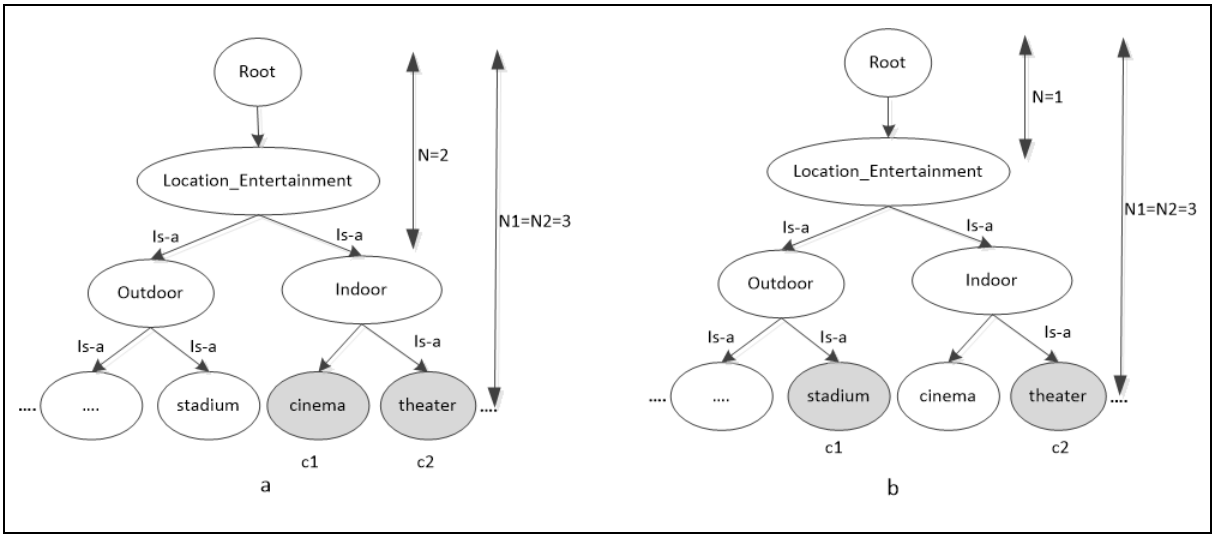

Figure 4.5 Wu and Palmer similarity measure

Then the Wu and Palmer measure is applied. However,

If the two concepts are linked to the LCS through other intermediary concepts $(\mathrm{N1} - \mathrm{N}) \neq 1\ \mathrm{and/or}\ (\mathrm{N2} - \mathrm{N}) \neq 1$, such as in Figure 4.5b where the two concepts are "stadium" and "theater" and the LCS is the concept "Location_entertainment,"

Then the modified measure decreases with the number of intermediary concepts $(\mathrm{N1} - \mathrm{N} - 1), (\mathrm{N2} - \mathrm{N} - 1)$:

$$Sim(c1,c2) = \begin{cases} Sim_{WP}(c1,c2) \ \ if \ (N1-N)=1 \ and \ (N2-N)=1 \\ \frac{N}{(N1+N2-N-1)} \ \ if \ \ (N1-N) \neq 1 \frac{and}{or} (N2-N) \neq 1 \end{cases} \quad (4.3)$$

$(\mathrm{N1}-\mathrm{N})=1$ and $(\mathrm{N2}-\mathrm{N})=1$ signifies that the concepts *c1* and *c2* belong to the same hierarchy and are linked directly to the same LCS.

If *c1* and *c2* are not in the same hierarchy $\left((\mathrm{N1}-\mathrm{N}) \neq 1 \frac{\mathrm{and}}{\mathrm{or}} (\mathrm{N2}-\mathrm{N}) \neq 1\right)$, **Then**

$$Sim(c1,c2) = \frac{2*N}{(N1+(N1-N-1)+(N2+(N2-N-1)))} = \frac{N}{(N1+N2-N-1)}$$

Where $(\mathrm{N1}-\mathrm{N}-1)$ and $(\mathrm{N2}-\mathrm{N}-1)$ represent the number of intermediary concepts in the taxonomy's hierarchy between the LCS and the two concepts (*c1* and *c2*) to be compared. The proposed modification meets the four criteria of similarity measures: non-negativity, identity, symmetry, and uniqueness.

4.4 Experimental Method

The CBR cycle was performed with the open-source software application myCBR, a similarity-based retrieval tool and software development kit (SDK). The application was developed at the CBR center of the German Resource Center for Artificial Intelligence (DFKI, www.dfki.de) in partnership with the School of Computing and Technology at the University of West London, UK (www.ewl.ac.uk). The standalone version of myCBR allowed us to acquire the necessary cases to construct an initial case base through csv (comma separated values) files. It was also used to store cases, calculate similarities, and retrieve the best matches. In myCBR, similarities are calculated at the attribute level (local similarity) as well as at a global level (global similarity) (Roth-Berghofer, Sauer, Garcia, Bach, Althoff & Agudo, 2008). The myCBR SDK 3.0.1 BETA version was used to acquire the case base, retrieve the best matches, and calculate the similarities.

A case description consists of raw data collected through the smartphone sensors (daily routine locations, current time, and mode of smartphone usage). The selection criteria for each characteristic are described in the following.

4.4.1 Daily Routine Locations

For a typical case, the description is composed of three groups of parameters: 1) The user's location, 2) the current time, and 3) the mode of smartphone usage. The kinds of locations and selected activities that make up a typical case (see Figure 4.4) were borrowed from (Incel, Kose & Ersoy, 2013), in which typical inferred activities were classified into six categories according to their objectives and in accordance with the state of the art in activity recognition: *cat. 1*: locomotion (walking, running); *cat. 2*: mode of transportation (bus, car); *cat. 3*: sporting activities (biking, etc.); *cat. 4*: health related activities (rehabilitation, etc.); *cat. 5*: daily/routine activities (shopping, eating); and *cat. 6*: smartphone usage (texting, browsing). In addition, we incorporated the work of Phithakkitnukoon et al. (2010) who assigned daily and routine activities linked to a person's location to nine categories: 1) eating (restaurant, bakery, coffee shop), 2) shopping (mall, store, market), 3) entertainment (theater, bowling alley, night club), 4) recreational (park, gym, fitness), 5) educational (university, college, school, institute), 6) work (workplace), 7) home (home, hotel), 8) variable (bus, car, subway, plane, train, running, walking, biking), and 9) worship (church, mosque, synagogue). Each group of parameters is associated with a set of services in the form Services = (rt, rv, pm, lm, apps), where each service is identified by an integer. Services = (1, 2, …10), (rt = ring tone = rt1, rt2, rt3), (rv = ringer volume = vibration only, low, medium, high), (pm = phone mode = normal, power saver), (lm = luminosity = low, medium, high), (app = online apps, offline apps, app update, AVscan, ebook, TV program, bus schedule, course schedule).

Table 4.2 shows the "seed" case base, which is composed of 10 initial cases corresponding to the 10 identified location categories (Location_Eating, Location_Entertainment, Location_Recreational, etc.).

Table 4.2 Initial Case Base

Case Name	Location	Time	Weekday	Phone Using	Services
case1	home	night	weekday	gaming	1
case2	workplace	day	weekday	nil	2
case3	cinema	night	weekend	nil	3
case4	church	day	weekend	nil	4
case5	school	day	weekday	browsing	5
case6	car	day	weekday	nil	6
case7	admin	day	weekday	calling	6
case8	park	day	weekend	gaming	8
case9	restaurant	night	weekend	nil	9
case10	mall	day	weekend	messaging	10

4.4.2 Activity Recognition Parameters

Among the multitude of available public data sets, we chose the simple yet comprehensive data set provided in (Shoaib, Scholten & Havinga, 2013). This data set was used to select the parameters that describe a user's physical activity based on data obtained through the inertial sensors of a smartphone. The data were collected by three smartphone users (male, 25–30 years old, equipped with a Samsung Galaxy S2) for six physical activities (walking, running, sitting, standing, walking upstairs, walking downstairs) with duration of 3–5 minutes per activity.

The following smartphone sensors were used to collect the data: the accelerometer (x/y/z, measured in m/sec2), the magnetometer (x/y/z, measured in micro Tesla units, μT), and the gyroscope (x/y/z, measured in rad/sec). An activity recognition algorithm running on MATLAB revealed that it is possible to identify the activities in the data set based on accelerometer characteristics—mean, standard deviation (SD), and variance (VAR) (see Table 4.3).

The similarity measure between two physical activities is given by:

$$sim\ (act1, act2) = \frac{1}{1 + dist\ (act1, act2)}$$

Where $dist(act1, act2)$ is the Euclidean distance between the activities *act1* and *act2*, given by:

$$dist\ (act1, act2) = \sqrt{(Acc_{x,y,z}(act1) - \ Acc_{x,y,z}(act2))^2}$$

Table 4.3 Similarity of Physical Activities

		Subject 1			
		Walking	Running	Standing	Sleeping
Subject 2	Walking	0.899	0.562	0.817	0.630
Subject 5	Running	0.680	0.722	0.604	0.436
Subject 10	Standing	0.756	0.484	0.912	0.747
Subject 14	Sleeping	0.629	0.375	0.775	0.782

4.4.3 Similarity Calculation

Local similarity attributes are of two types:

- **Similarities of the symbolic attributes "location" and "phone usage":**

The modified Wu and Palmer similarity measure was used according to (Eq. 4.3). Figure 4.6 (a,b,c,d) shows the taxonomies applicable to a user's location and phone usage. Location taxonomies (Location_shopping, Location_eating, Location_education, Location_home, Location_work, Location_government, Location_worship) are shown in Figure 4.6e.

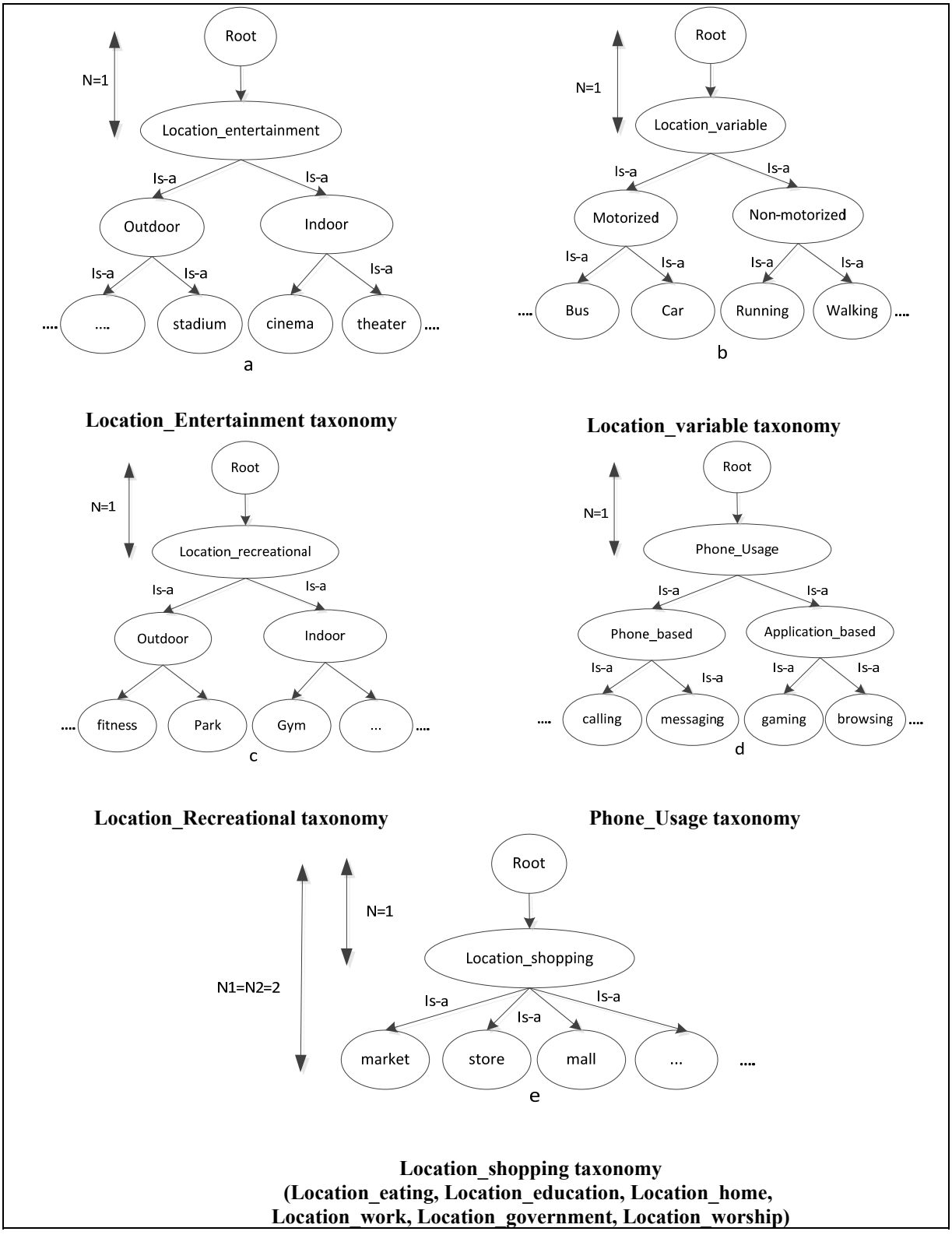

Figure 4.6 Taxonomies of the symbolic attributes “location” and “phone usage”

The following demonstrates how the modified similarity measure is typically applied to the taxonomy, using the location "Location_Entertainment" as an example (Table 4.4):

$sim(cinema, cinema) = 1$

$sim(cinema, stadium) = \frac{N}{(N1+N2-N-1)}$

Where N = 1, N1 = 3, N2 = 3 , (N1−N = 2 and N2−N = 2)

$sim(cinema, stadium) = \frac{1}{4} = 0.25$

$sim_{wp}(cinema, theater) = \frac{2N}{N1+N2} = \frac{4}{6} = 0.66$

Where N = 2, N1 = 3, N2 = 3

(N1−N = 1 and N2−N = 1)

Table 4.4 Similarities between Locations "Location_Entertainment"

	nil	stadium	cinema	bowling	theater
nil	0.0	0.0	0.0	0.0	0.0
stadium	0.0	1.0	0.25	0.25	0.25
cinema	0.0	0.25	1.0	0.66	0.66
bowling	0.0	0.25	0.66	1.0	0.66
theater	0.0	0.25	0.66	0.66	1.0

- **Similarities of the "time" attribute:**

Similarities between the symbolic attributes of "time" are calculated using the overlap similarity measure (Stanfill & Waltz, 1986):

$$sim(time_{new}, time_{old}) = \begin{cases} 1 & if \ time_{new} = time_{old} \\ 0 & Else \end{cases} \quad (4.4)$$

4.4.4 Application

We used the graphical interface of myCBR 3.0.1 for the initial phase (storing of the 10 seed cases in the case base), to calculate the similarities between (randomly generated) new cases

and cases in memory, and to retrieve the best match.

An example of the CBR cycle including, the retrieval of the best match cases, services retrieved, services filtered (recommended to the user) and database updating is given below:

- **Initial query**: Case 11 (Table 4.5)
- **Retrieval of best match case with Similarity Score**:

-[name=case10, location=mall, time=day, weekday=weekend, phoneUsing=messaging], Similarity=0.52

-[name=case3, location=cinema, time=night, weekday=weekend, phoneUsing=nil], Similarity= 0.58

-[name=case9, location=restaurant, time=night, weekday=weekend, phoneUsing=nil], Similarity= 0.82

Case 9 will be selected and the retrieved services are shown in figure 4.7.

Table 4.5 Example of a new case

Parameter Name	Value
Case Name	case11
Location	restaurant
Time	night
Weekday	weekend
PhoneUsing	messaging
Context Parameters	
Battery Level	High
Luminosity	Medium
Free Memory	Medium
Noise Level	Low
Internet Connection Speed	High

- **Services Filtering:**

A rule set—BatteryRule, FreeMemoryRule, InternetSpeedRule, LuminosityRule, and NoiseRule—was used to filter and adapt the services derived from the environmental context (noise level, NS; and Luminosity, LM) and the system context (free memory, FM; battery level, BL; internet connection speed, IS).

Table 4.6 Filtering rules

Filtering rule	Description
Noise Level based	**Rule1: IF** *NS* = high **THEN** *RINGTONE* = vibration
Luminosity based	**Rule2: IF** *LM* = low **THEN** *LUMINOSITY* = high,
Battery Level based	**Rule3: IF** *BL* = low **THEN** *MODE* = power saver,
Battery Level and Free Memory based	**Rule4: IF** *BL* = low **OR** *FM* = LOW **THEN** EXCLUDE (offline games, offline video, apps update, AVscan)
Internet Connection based.	**Rule5: IF** *IS* = low **THEN** EXCLUDE (online video, online games)

Each rule is implemented through the filterService(Collection<Service>,,UserContext) function. Services retrieved during the retrieval phase become the input to the function, and filtering is achieved by deleting (*exclude*) or changing (*add*) services based on the user's context (Figure 4.7)

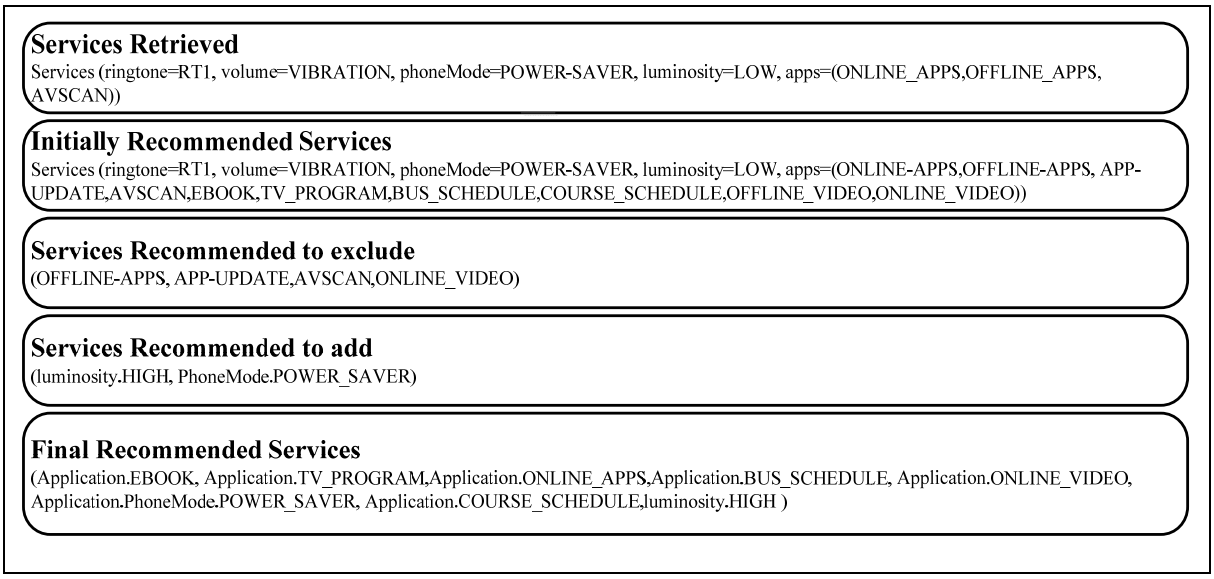

Figure 4.7 Retrieved and filtered services

The application was tested with 50+ cases and some of the results are shown in table 4.7. The services provided to the user correspond to the user's location, current time, smartphone usage, and current context.

Table 4.7 Retrieval and Filtering Results

New case	Best match case	Service retrieved	Context	Services filtered
cafeteria, night, weekday, messaging	Sim = 0.83 Smartphone9 = restaurant, night, weekend, nil	Services = 9 ring tone = rt2, ringer volume = vibration, phone mode = power saver, luminosity = medium, apps = apps update, AVscan	BL = low IS = high FM = low LM = high NS = low	*apps = apps update, AVscan* Removed by rule 4 BL = low
University, day, weekday, messaging	Sim = 0.85 Smartphone5 = school, day, weekday, browsing	Services = 4 ring tone = rt2, ringer volume = medium, phone mode = normal, luminosity = medium, apps = bus schedule, course schedule	BL = high IS = low FM = low LM = high NS = low	No filtered services
theater, day, weekend, nil	Sim = 0.88 Smartphone3 = cinema, night, weekend, nil	Services = 3 ring tone = rt3, ringer volume = vibration, phone mode = power saver, luminosity = medium, apps = agenda, games	BL = low IS = low FM = high LM = high NS = low	*App = games* Removed by rule 4 BL = low Is = low
market day, weekend, calling	Sim = 0.85 Smartphone10= mall, day, weekend, messaging	Services=5 ring tone = rt2, ringer volume = high, phone mode = normal, luminosity = medium, app = online apps, offline apps, agenda, games	BL = low IS = high FM = low LM = low NS = high	*App = games, Online apps, offline apps* Removed by rule 4,5 BL = low, FM = low *Ringvolume* changed to : vibration NS = high Removed by Rule 1
bus day, weekday, gaming	Sim = 0.85 Smartphone6 = car, day, weekday, nil	Services = 6 ring tone = rt1, ringer volume = high, phone mode = power saver, luminosity = medium, apps = agenda	BL = high IS = low FM = low LM = low NS = low	No filtered services

If the new case differs from all other cases in the case base ($\text{sim}(Case_{new}, Case_{old}) < 1$), then the case base is updated to include the new case $Case_{new}$ with the following structure: $Case_{new}$:{Description, $(\text{Location, time, phone\_using})_{new}$, Services=$(\text{services})_{filtred}$}. The two-stage service adaptation process (similarity measurement and contextual filtering)

consists in providing real services that are adapted to the user's current context. For example, for case 1, the user is sending a message in a cafeteria at night on a weekday. The best match found in the case base is case 9 containing the set of services = 9, which includes updating applications and scanning for viruses. However, neither of these two actions is initiated because of filtering rule 4 (BL = low). A classification of services according to available smartphone resources and current user context is critically important to create cases that are more relevant and to define filtering rules that provide a more relevant connection between the context and the proposed service.

4.5 Conclusion

We applied the case-based reasoning process to mobile devices (smartphones) using the devices' built-in sensors to adapt services provided to a user according to the user's location, current time, and smartphone usage. The similarity measure by Wu and Palmer was modified to correspond to the target application. Our approach provides a basis for the implementation of adaptive systems involving lightweight context-sensitive services in the field of resource-limited devices such as smartphones. Further research will be necessary to investigate topics such as choosing and classifying the services a smartphone can provide to a user, updating the case base, and defining adaptation and filtering rules.

CHAPITRE 5

CONTEXTUAL LOCATION PREDICTION USING SPATIO-TEMPORAL CLUSTERING

Djamel Guessoum[a], Moeiz Miraoui[b], Chakib Tadj[a]

[a]Electrical Engineering Department, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[b]Higher Institute of Applied Science and Technology, University of Gafsa,
2112 Gafsa, Tunisia

Cet article a été publié dans « International Journal of Pervasive Computing and Communications», Vol. 12, no. 3 (2016): 290-309.

Résumé

La prédiction d'un contexte et en particulier la localisation d'un utilisateur, est une tâche fondamentale dans le domaine de l'informatique diffuse. Elle ouvre un nouvel et riche domaine pour l'adaptation proactive des applications sensibles au contexte. La méthode proposée prédit la localisation d'un utilisateur sur la base de l'historique de sa mobilité. Le procédé utilise des informations contextuelles (position actuelle, le jour de la semaine, l'heure, la vitesse) pouvant être acquis facilement et avec précision à l'aide de capteurs communs tels que le GPS. Ces informations sont ensuite utilisées pour trouver les points d'intérêts qu'un utilisateur visite fréquemment et pour déterminer la séquence de ces visites à l'aide de la technique du clustering spatial, de la segmentation temporelle, et du filtrage de la vitesse. La méthode proposée a été testée avec des données réelles à l'aide de plusieurs algorithmes de classification supervisés et a donné des résultats très intéressants.

Mots clés : prédiction de la localisation, informatique diffuse, sensibilité au contexte, clustering, DBSCAN.

Abstract

The prediction of a context, especially of a user's location, is a fundamental task in the field of pervasive computing. Such predictions open up a new and rich field of proactive adaptation for context-aware applications. The proposed methodology predicts a user's location on the basis of a user's mobility history. The method uses contextual information (current position, day of the week, time and speed) that can be acquired easily and accurately with the help of common sensors such as GPS. This information is then used to find the points of interest that a user visits frequently and to determine the sequence of these visits with the aid of spatial clustering, temporal segmentation, and speed filtering. The proposed method was tested with a real dataset using several supervised classification algorithms, which yielded very interesting results.

Keywords: location prediction, pervasive computing, context-awareness, clustering, dbscan.

5.1 Introduction

Human behaviour is complex and usually context-dependent according to (Do and Gatica-Perez 2012). Because of its diverse applications, context prediction is a very important research topic in pervasive computing. It opens up a new and rich field of proactive adaptation for context-aware applications that adapt to the changing context in which they operate. This adaptation allows for more efficient interactions with the user, resulting in the proposal of relevant information and services (Lee et al. 2010) based on the existence of services related to future contexts.

Applications that predict future contexts are linked to the current context, which the application can acquire. An example of such services is the reconfiguration of a pervasive system that includes changing the system configuration according to changes in the environments for example, accident prevention, management of personal information such as

SMS user alerts and email notifications for appointments and actions, scheduling actions according to predictions made by a context-aware application (Mayrhofer 2005), and management of device resources. Studies on context models have shown that a user's location, time, and activity are the most important parameters that determine the type of service to be provided (Yuan and Herbert 2014; Bolchini et al. 2007).

Here, we are interested in predicting the location as contextual information, which is the most commonly used form of context (Ashbrook and Starner 2003). The importance of context (location) has been studied extensively (Voigtmann and David 2012), in particular concerning the richness of information that can be exploited in a pervasive system. For example, the location"at home" implies an individual's personal state, environment, and system as well as the type of service that this individual expects at that location. This information can be acquired easily and accurately using techniques such as GPS, WLAN, and Wi-Fi. Predicting the location has diverse applications, for example assistance and suggestions for taxi drivers to find the best routes (Do and Gatica-Perez 2012), dissemination of information related to points of interest such as advertising, recreation, or notifications of events and services available in the vicinity of these points of interest (Scellato et al. 2011).

In the present paper, context and contextual information are used interchangeably and designate identical concepts in accordance with (Meissen et al. 2004; and Kayes et al. 2014), who consider context to be an instantiation of all available contextual information (e.g. location, temperature) at a certain time. We also use the widely cited definition by (Dey and Abowd 1999), who defines context as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". Context is represented by a dataset collected from a user's physical or system environment.

We present an approach for the prediction of a user's outdoor location on the basis of current context information that we consider to be important for the prediction. This contextual information is comprised of the user's current location, the day of the week, the time, and the user's speed of locomotion. Most studies on the prediction of outdoor locations use additional contextual information such as traffic congestion, climate, and geographical data. Although this information can improve the prediction, its collection requires additional equipment, which may introduce additional uncertainty to the accuracy of the collected data and affect the actual implementation. In the present study, spatial data were temporally segregated according to the day of the week and the hour of the day. For each hour of the day, we used the spatial density clustering algorithm DBSCAN (density-based spatial clustering of applications with noise) to determine a set of close points that define a cluster and for which the user's speed of locomotion was below a certain threshold. This speed threshold defines the boundary between a user in transit (using a vehicle such as a car, bus, or train) and a user visiting a region of space and moving slowly. This threshold may be the subject of future research. For the present study, it was our goal to determine the points in observational space collected by GPS (including speed, time, and day) for places that a user visits frequently on particular days, at certain times, and in a specific order.

Section 2 presents a literature review as well as our motivations for the present study and its potential contributions. Section 3 describes the most commonly used prediction techniques, and in Section 4 we approach clustering in general and spatial clustering in particular. Section 5 explains our application in detail. Finally, our conclusions are given in Section 6.

5.2 Related work

In the literature, context (location) prediction is considered an important aspect of improving the performance of context-aware applications to provide appropriate services. Context prediction studies can be categorized according to the type of context to predict (an individual's future location, next activity, etc.). The contexts that can be exploited to

accomplish this prediction (such as traffic congestion or climate), if the context to predict is location, whether it is indoor or outdoor, and lastly, the type of prediction algorithm used (Bayesian networks, Markov chains, etc.).

Although the prediction algorithms are well known, the contexts to predict and the contexts that can be exploited for prediction vary and depend on the application of the prediction in each case.

In the field of context prediction, contextual information has been exploited to improve prediction performance. The types of information most used in the literature are the user's profile (e.g. senior, tourist, taxi driver), location, and time (Bar-David and Last 2016). The location of a user provides the most common and important contextual information and is the easiest to collect through several techniques (GPS, WLAN, Wi-Fi, etc.) (Voigtmann and David 2012).

Research on location predictions is abundant because it generally involves information that can be used to proactively provide relevant services (e.g. a user in a train station will need the train routes, time schedule, etc.).

Previous research has used traffic congestion, climate, day-of-the-week, user speed, and current location data in conjunction with k-means clustering to identify interesting points, which were labeled with an external web-based service (Bar-David and Last 2016). This contextual information was exploited to improve the prediction of future locations. In addition to predicting the location, other information was also predicted for example, the duration of a user's stay in an actual location or at a destination which was derived by extracting user mobility patterns in conjunction with contextual information (current location and time) (Do and Gatica-Perez 2012; Scellato et al. 2011). Other studies have simply investigated the user's destination or vehicle (Patterson et al. 2003; Krumm and Horvitz 2007; Yoon and Lee 2008).

Contextual information has also been used to improve prediction robustness and to make predictions less sensitive to sensor errors (Knig et al. 2013) by using the multi-context prediction approach proposed by (Sigg et al. 2010). Another study predicted the path of a hurricane by dividing its path into two trajectories: a spatiotemporal trajectory and a contextual trajectory derived from the geographic information of traversed regions modelled as a succession of labeled polygonal cells (Buchin et al. 2014).

Because the interior of a building cannot be accessed with a GPS, special equipment such as Wi-Fi or WLAN is required to predict indoor contexts and locations, for example an activity or movement occurring inside of a building. The CRAFFT project aims to predict activities occurring in a smart home (Nazerfard and Cook 2015), and the Smart Doorplate Project uses the Augsburg indoor location tracking benchmarks (Petzold et al. 2005) to predict the movements of an employee in an office or adjacent room (Petzold et al. 2004). In these examples of indoor location prediction, the prediction algorithms are essentially classifiers that use a user's mobility history as training data. Our work did not address the prediction of indoor locations.

The prediction of the context or location in outdoor spaces which is the focus of the present study is the type of prediction most often investigated because of the growing availability of GPS sensors with increased accuracy. A history of past observations serves as the basis for context prediction and localisation studies and for the prediction of general future behaviour of an individual in an outdoor space (Do and Gatica-Perez 2012). These methods are based on the deterministic assumption that future events are determined by past events and that whenever a situation is observed, the subsequent situation will be similar to previously observed behaviour. Daily and weekly routines are assumed to be well established, and the activities of individuals are characterised by a degree of regularity and predictability (Bar-David and Last 2016; Scellato et al. 2011). The behavioural history of an individual is modelled as a set of patterns of visited points in space and time (points of interest, activity

points, stay points, stay durations). These patterns are then exploited through various prediction techniques.

Clustering (such as DBSCAN and k-means) and spatial segmentation in the form of cells are common techniques used to determine spatial and temporal patterns. In (Schougaard 2007) author divided space into 200-m wide cells to predict which cell a driver will enter next; the study provided corresponding probabilities based on the vehicle's current direction and the road layout. (Eagle et al. 2009) used dynamic Bayesian networks with segmentation to predict the trajectory of signals emitted by telecommunication towers. (Krumm and Horvitz 2006) used Bayesian networks to predict the next destination of a taxi driver's trip in progress by using a history of the driver's habits and destinations. The probabilities were determined based on the number of times the destinations were visited in the past, and the map was divided into cells. By taking the time and the change of direction as constraints, the authors were able to segment the trajectories and identify stop points and activities with the CDBSCAN (DBSCAN clustering with constraints) algorithm as well as through SVM (support vector machines) with three attributes (stop duration, mean distance to the centroid of a cluster, and the shortest distance between the current location and the home or workplace) (Gong et al. 2015). The study did not consider points of interest at which the individual was in motion (in a park, in the city, etc.). Mobile phone network cells were used to represent the space (Anagnostopoulos et al. 2009). Algorithms from the field of machine learning and the transitions between the cells were applied to predict the future location. The MBB (minimum bounding box) was used to delimit areas according to time and related contextual information (Bar-David and Last 2016).

In all of these studies, the spatial patterns that form the mobility history were segments of space (such as telecommunication cells, polygonal cells and rectangular cells). These studies investigated areas that contained interesting features or additional information concerning user habits.

Because prediction algorithms are application-dependent, none of the prediction algorithms is superior to the others. Each algorithm has advantages and disadvantages. With respect to Bayesian networks, dynamic Bayesian networks are most frequently used because they are suitable for time series, which is the typical form of data representing a user's contextual history (Do and Gatica-Perez 2012; Patterson et al. 2003; van Kasteren and Krose 2007) . Ensemble classifiers, also known as a committee of classifiers, represent a combination of individual classifiers and are used to reduce prediction errors. They have demonstrated accurate performance (Lee and Cho 2010). Markov chains have proven relevant for cases in which the points of interest represent nodes and the transition between two nodes represents the probability of traveling between two points (Ashbrook and Starner 2003).

The prediction approach presented here uses a minimal amount of contextual information (location, day of the week, time and speed) (see Figure 5.1). Conventional sensors such as GPS, WLAN, and Wi-Fi can then be used to acquire the necessary data with considerable precision. These sensors are further able to determine locations considered to be noise, which are either transitory locations (with not enough density to create a cluster or a point of interest), or points at which the user is moving at high speed.

The proposed method is simple, and collected GPS points can be used to identify and predict the points of interest. In addition, we applied a temporal segmentation to the GPS data for which the day was divided into hours. Clusters and points of interest were searched for each hour of the day. This temporal segmentation is simpler than the usual spatial segmentation mentioned in the literature. Our study shows that filtering of high-speed and noise points can improve prediction accuracy.

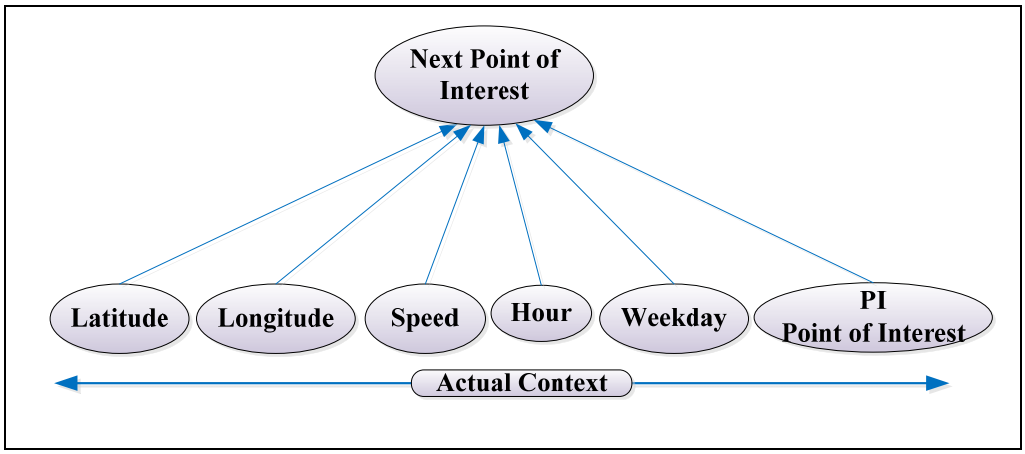

Figure 5.1 Contextual information for location prediction

5.3 Context prediction techniques

The best-known prediction techniques (Boytsov and Zaslavsky 2010) are classification algorithms from the field of machine learning, which are used to categorise unknown data (future context or location) into discrete classes (Bayesian networks, decision trees, KNN, SVM, neural networks, etc.). Other fields from which techniques have been borrowed include computational biology (alignment), data compression (Active LeZi), and branch prediction of microprocessors (state predictors).

- **Bayesian networks:**

Bayes nets, naive Bayes, and dynamic Bayesian networks are statistical classification algorithms based on Bayes' theorem. Naive Bayes networks assume that the effect of an attribute's value on the value of the class to predict is independent of the value of other attributes. Dynamic Bayesian networks are an extension of Bayesian networks. They model a dynamic system that changes over time and can represent the temporal properties of contextual information. Bayesian networks are suitable for the generation of predictive models in the real world because they take into account the uncertainty inherent in all facets of human activities (Zaguia et al. 2015); however, they require more data for learning;

- **Markov chains for context predictions:**

Markov chains are a variant of dynamic Bayesian networks. They are used to model a system with a finite number of non-overlapping states by calculating the probability of a transition from one state to another. A user's habits can be inferred from the user's past sequence of actions. Initially, the probabilities of transitions are not known. They are used primarily to address the problem of short-term location (Bar-David and Last 2016);

- **Expert systems and decision trees:**

These methods are based on expert systems and rule-based engines. The aim of this approach is to build rules for prediction, which provide a clear overview of the entire system. A decision tree (e.g. classifier C 4.5) is constructed using a function (information gain ratio) that determines the structure of the tree as well as the informative contribution of each branch. These systems are easy to understand and able to handle non-linear interactions between variables. They are not affected by outliers and can process large amounts of categorical and numerical data (Boytsov and Zaslavsky 2010);

- **Ensemble classifiers:**

Also known as a committee of classifiers, ensemble classifiers can improve context prediction performance by exploiting the advantages of individual classifiers for portions of the data (Lee and Cho 2010). There are several ways to combine individual classifiers, of which the most popular are (Anagnostopoulos et al. 2009):

1. Voting, in which each classifier votes for or predicts a class, and the selected final class is the one that received the most votes;

2. Bagging, in which the learning data are divided randomly into several parts, each individual classifier is applied to a portion of the data, and the final selected class is the one that receives the most votes;

3. Boosting, which uses an incremental classification involving the classification of instances that have not been classified in the previous iteration. The final selected class is determined by a weighted vote of individual classifiers for which the weights are determined based on the performance of these classifiers;

- **Alignment (sequence prediction):**

This method is used if the context can be decomposed into a sequence of events. Alignment is a context prediction algorithm applied to a time series and inspired by algorithms used in computational biology. This algorithm compares two sequences of contexts (Voigtmann 2014). A matrix, which contains penalties, uses columns to represent the context history and lines to represent the predicted contexts. This technique presents the values of past contextual information in the form of sequences that represent the context history. Each entry receives a timestamp (Knig et al. 2013). First, the algorithm combines the most recently obtained values of a context into a pattern that is aligned with the sequences of the history. This alignment then returns every match for which similar patterns are found in the history that share the quality of the match. Finally, the entry that conforms to the pattern of the match in the history is considered the prediction (Gopalratnam and Cook 2007) ;

- **Active LeZi:**

The active LeZi context prediction algorithm has been presented by (Gopalratnam and Cook 2004; Cook et al. 2003; Gopalratnam and Cook, 2007) . It is based on the LZ78 data compression algorithm by Jacob Ziv and Abraham Lempel. LeZi exploits information in a user's context history by using a sliding window to form a "LeZi" trie and to calculate the probability of each possible context transition. The maximum depth of the trie corresponds to the length of the context in the user's history (Voigtmann, 2014). To predict the context, the generated trie receives the current pattern Cp and calculates the probability of all possible contexts that may follow the given context. The context with the highest probability is predicted ;

- **State predictor:**

The state predictor approach was developed by Jan Petzold (Petzold et al. 2003a; Petzold et al. 2003b). It is based on the branch prediction techniques for microprocessors that have been applied to context prediction. The state predictor is a set of patterns of an individual's context or mobility history. It can assume two states: "strong" if the person performed the same sequence twice, and "weak" otherwise. A one-state predictor signifies that if a person repeats

an action once this habit is predicted and there is no weak or strong habit. The two-state predictor usually applies to repeating the same action twice, which introduces the concept of weak and strong habits.

These location prediction techniques clearly build on a user's mobility patterns inferred from the user's history. User habits are represented by a sequence of events or contexts that serve as patterns on which to base the prediction algorithms. Previously visited locations, or points that the user has shown interest in, represent an essential characteristic of these patterns and are used to predict the location. Clustering, which is an unsupervised classification technique for discovering similar structures in the data, is used to determine these points of interest.
This technique has been applied widely in the area of predictions (Ameyed et al. 2015) and has been used in several studies aimed at predicting location (Ying et al. 2011; Morzy 2007; Yavas et al. 2005).

In our work, clustering was used to find regions of space where the points collected are having a defined density.

5.4 Clustering

Clustering is an unsupervised classification algorithm. It is one of the most frequently used techniques in the fields of data mining and knowledge discovery. Clustering consists of a process that groups similar unlabeled data (such as binary, categorical, numerical, interval, ordinal, relational, textual, spatial, temporal, spatiotemporal, pictorial, or multimedia data) in the same cluster and assigns dissimilar data to other clusters. The main categories of clustering are partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods. Partitioning methods create k partitions of data from a set of n data ($k \leq n$). K-means and k-medoids are the most well-known algorithms in this category (Warren Liao 2005).

Hierarchical clustering groups the data into trees of clusters. The two methods in this category are the divisive and the agglomerative method, which differ in their tree

construction approach. The divisive method initially groups data into partial clusters, which are then grouped into a global cluster; the agglomerative method does the inverse (CHAMELEON and CURE).

Density-based clustering methods define clusters according to their data density, which depends on the proximity parameters and on the amount of data. These clustering methods are routinely used in spatial or spatiotemporal clustering. DBSCAN and OPTICS are two of the most well-known algorithms in this category. Grid-based clustering methods such as the STING approach divide the data space into cells in which clustering is performed. Examples of clustering applications include information extraction, text search, applications for spatial databases, web applications, and DNA analysis in computational biology (Berkhin 2006).

Because a location is defined by spatial data, we chose to use the spatial clustering (DBSCAN) method to determine the locations or points of interest visited in the mobility history of an individual.

5.4.1 DBSCAN and temporal filtering

The trajectories of an object recorded over a period of time can provide information on a user's mobility habits, such as points of interest visited, the duration of each visit, the succession of these visits, and their context dependence. A trajectory describes the behaviour of a moving object (Kisilevich et al. 2009). Therefore, clustering can be used to detect groups of objects that have the same behaviour or to detect interesting behaviours in these movements (e.g. a stop point or a frequently visited point).

DBSCAN is a data clustering algorithm proposed by (Ester et al. 1996). It uses a distance threshold (epsilon) between the data points and a minimum number of these points to discover dense regions in the data space. It is designed to discover clusters of arbitrary shapes

and can detect points considered noise as well as points that cannot be classified into any cluster (Figure 5.2).

In Figure 5.2, points 1 and 2 are density reachable; points 1 and 3 are density connected; points 1, 2, and 3 are core points; point 4 is a border point; and the N points are noise points (min. points = 4 and epsilon = e). To measure the distance between two points, DBSCAN uses a distance function (e.g. Manhattan or Euclidean). A point belongs to a cluster if there is at least one other point at a distance that is less than or equal to the threshold epsilon. A point is considered noise if its distance to all other points exceeds the threshold epsilon.

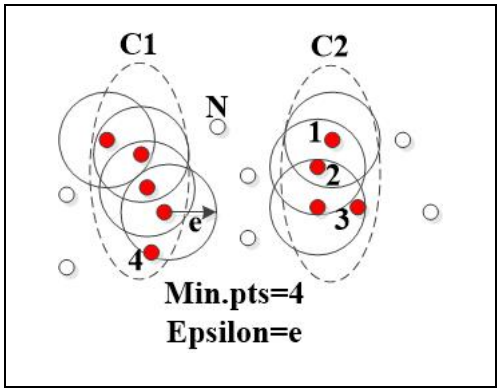

Figure 5.2 The DBSCAN algorithm

Spatial clustering has several weaknesses. It is possible that two points or observations belonging to the same cluster are observed at different times, such as when a user passes the same point in the morning and in the evening (Figure 5.3). Second, the same point may be passed several times at high speed.

To resolve these two issues, we opted for a temporal segmentation of the observed points: Clusters were searched at every hour of the day. The speed threshold was fixed (8km/h) to establish an intermediate value between low and high speed (users who walk and those using vehicles), thereby enabling the extraction of high-speed points from these clusters.

In Figure 5.3, points (X1; t1), (X2; t2), and (X3; t3) form the"candidate" cluster C because they belong to the same one-hour time window (afternoon). The point (Y1; t6) cannot be included in the cluster C because it belongs to a different time window (morning).

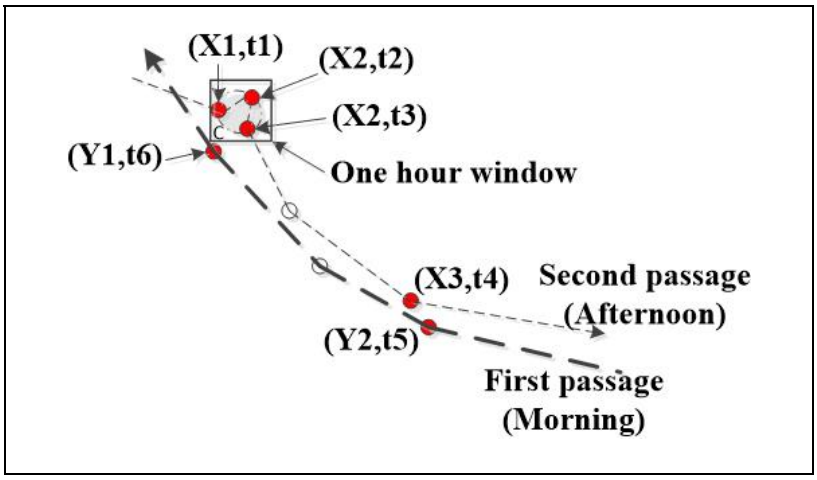

Figure 5.3 DBSCAN and temporal filtering

5.5 Experimentation

Our approach for the prediction of the location of an individual is based on the assumption that people do not move randomly and follow repetitive trajectories (Bar-David and Last 2016). Therefore, an individual's mobility history is shaped by contextual information (day of the week, time, current location, and speed). This fact is taken into account for the classification; it is represented by an (m+1)-dimensional vector v, where m are the visited locations ordered according to time, and the context represents the classifier training data. The localisation is the class of the next location to be visited (the next point of interest).

To create a user's mobility history, we used the MDC (Mobile Data Challenge) database made available by the Idiap Research Institute, Switzerland, and owned by Nokia (Kiukkonen et al. 2010; Laurila et al. 2012). This public dataset was released as part of the Nokia MDC in 2012. The dataset was collected in Switzerland from 2009 to 2011 using

Nokia N95 smartphones. Although the original dataset was collected with 200 participants, public data has only been released for 38 participants. The dataset contains continuously collected mobility data (GPS,Wi-Fi, GSM), social interactions (voice calls, SMS, Bluetooth), and phone usage (application/data usage) for all participants. The present analysis considered mobility data only.

GSM information was scanned every minute, Wi-Fi scanning was performed every 2 minutes, and GPS coordinates were sampled every 10 seconds. The dataset showed considerable spatial diversity because of the long collection duration and the large number of participants. The dataset contains about 122 days of GPS data, 191 days of GSM data, and 188 days of Wi-Fi data for nearly half of the participants. Figure 5.4 shows the overall user trajectories.

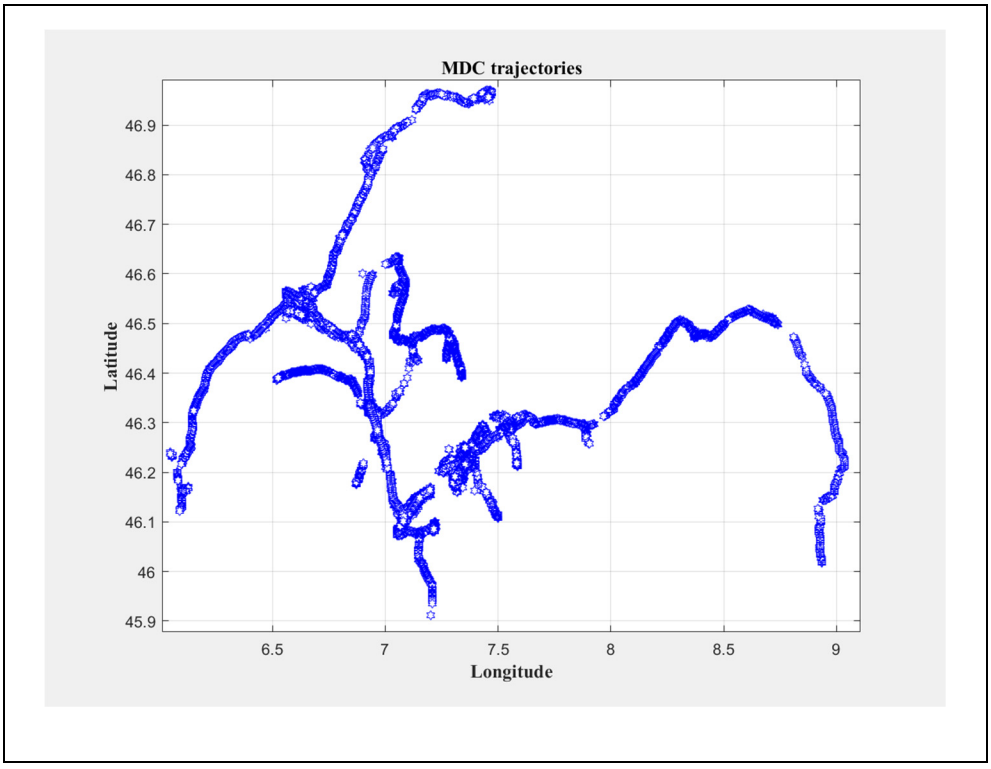

Figure 5.4 User trajectories

For the location prediction process, we selected the GPS data (latitude, longitude, day, time, speed) of user 5542 because this user had a rich history of visits that extended over a period of more than one year (from 2009-09-02 to 2011-02-23), as shown in Figure 5.5. The same process can be applied to all other users in the dataset.

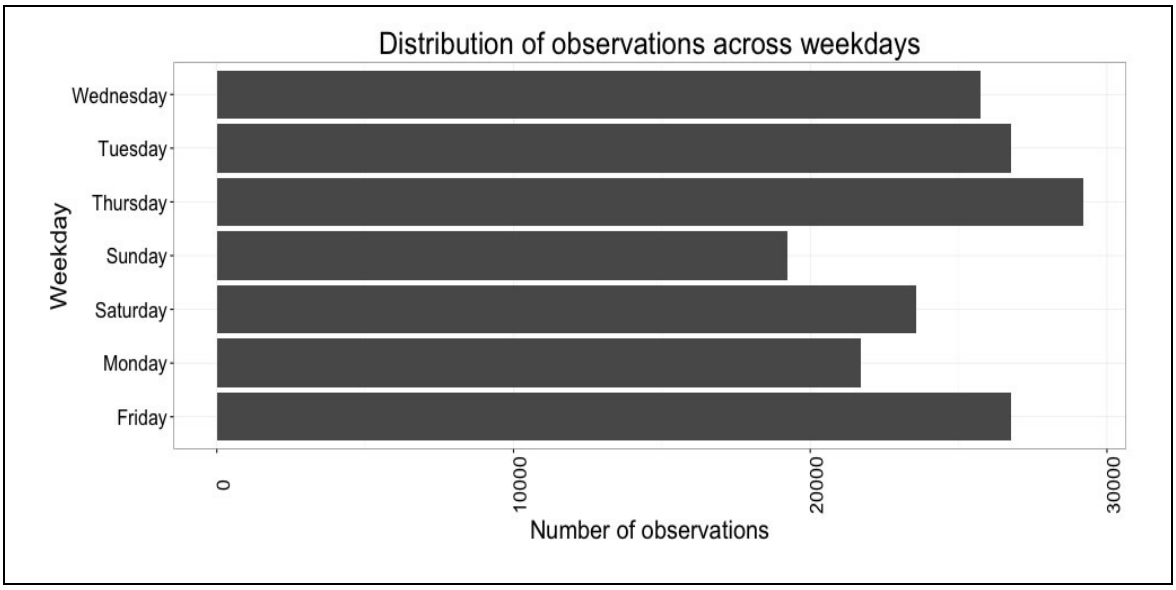

Figure 5.5 Collected points for user #5542

Definition of a point of interest. A region of space is considered a PI (point of interest) if a user spends a significant amount of time at this location on a regular basis. To match our PI criteria for the DBSCAN, a significant amount of time was defined as a number of user observations greater than the minimum sample parameter.

MinSample: Number of minimum core samples in a cluster.

Epsilon: Maximum distance between points in a core sample.

Distance metric: Haversine distance.

Definition of the transition point. A transition point is any point that does not belong to any clusters or PIs and is considered noise. There are three types of noise points:

Simple legal noise: A point that is too far away from any constructed clusters.

High-speed noise: A point that has high speed (> 8km/h) (Figure 5.6).

Noise with no clusters found: a point is located within a day of the week/hour, but no cluster exists for that combination.

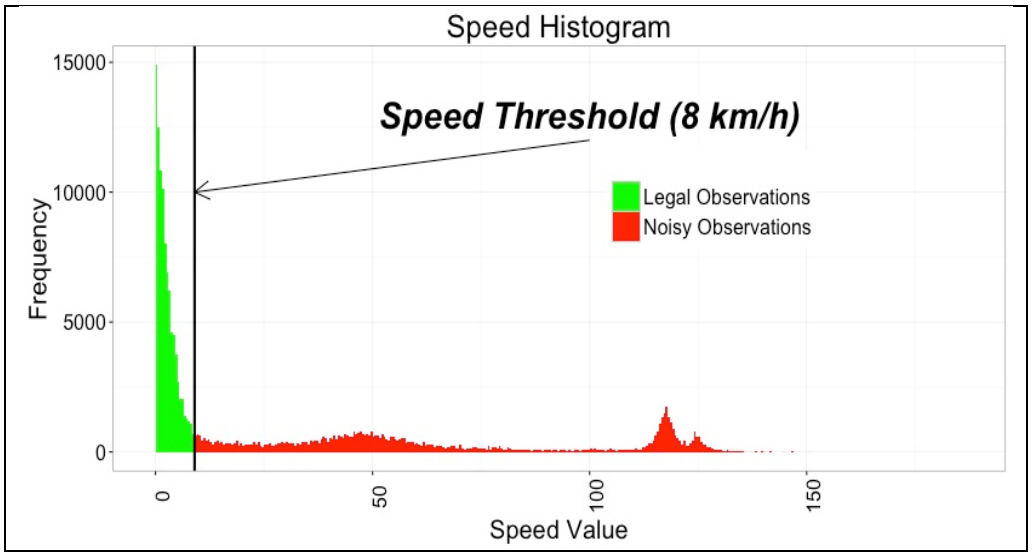

Figure 5.6 Speed histogram

A DBSCAN cluster or a point of interest is defined as follows (Figure 5.7):

1. Day-of-the-week cluster, cluster hour in a day (0-24);
2. Core samples (sets of points that define a cluster): the minimal number of points in a cluster is 200 points. The distance between two points in the cluster is 50 m (Haversine metric) (Skovsgaard et al. 2014). The critical maximal speed in a cluster $V_{crit.} = 8\ km/h$ (if $> V_{crit.}$, point is considered as high speed noise);
3. The cluster radius shift= 100 m, which is the distance to the nearest cluster (GPS data can vary arbitrarily by a maximum of 100 m (used during the prediction step);
4. The fill rate for high-speed noise = 0.2, which is the acceptable proportion of high-speed noise points in a cluster.

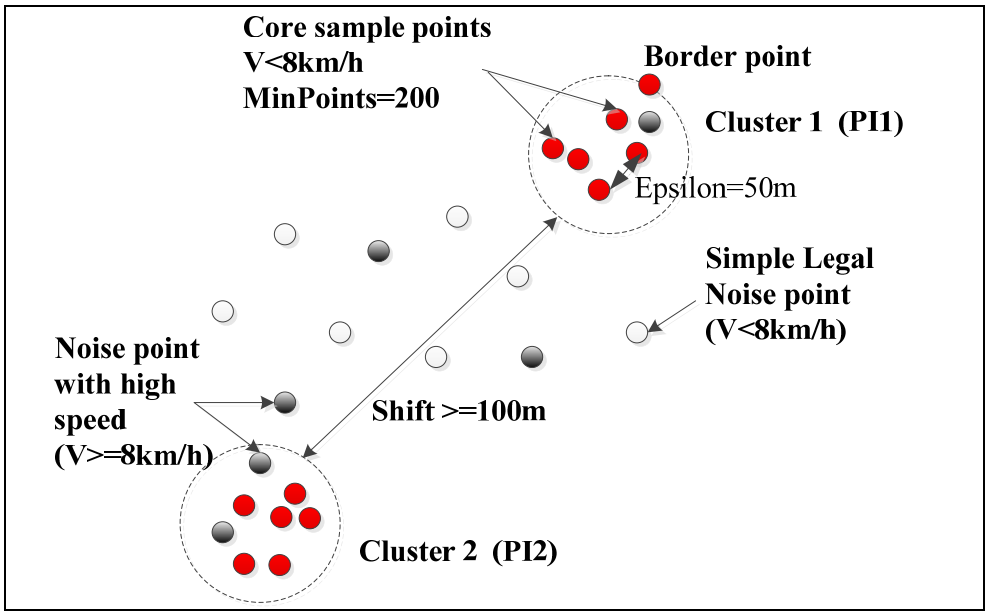

Figure 5.7 Cluster parameters

The GPS data preprocessing and DBSCAN clustering steps are shown in Figure 5.8.

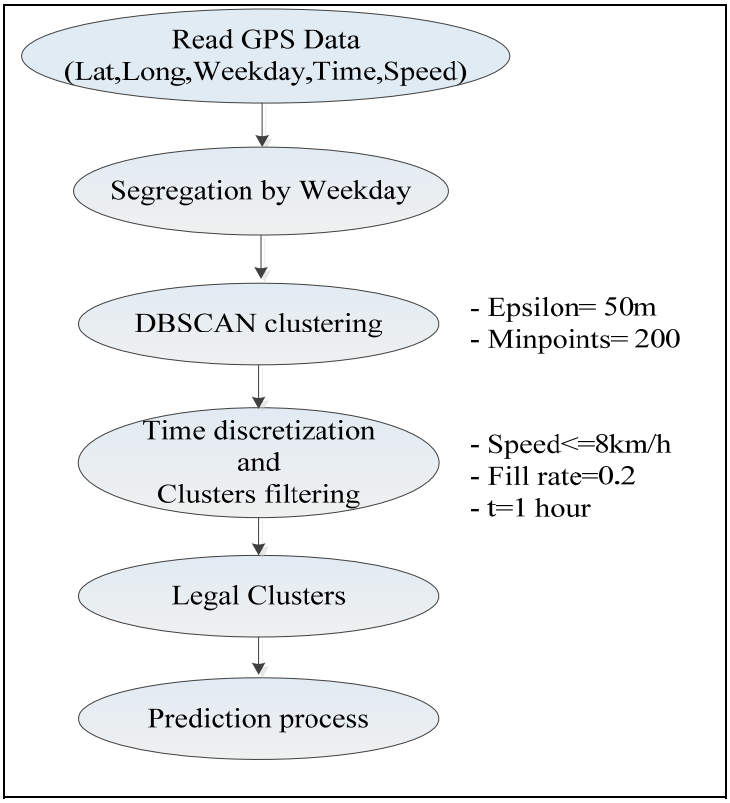

Figure 5.8 GPS data preprocessing and DBSCAN clustering

The steps in Figure 5.8 are performed for the data obtained for each day. The data are prepared as follows using the programming language R (a programming language and software environment for statistical computing).

1. Load data;

2. Fit the DBSCAN clustering algorithm for the data of an entire day using the metric, min samples, and epsilon of the configuration;

3. Construct a set of available clusters for the current day of the week using the defined discretization, which is one hour;

- examine each core sample of each cluster. If this cluster appears in the current hour, add it to this hour as an available cluster. For example, globally over all days of the week, Saturday hour = 0 can include clusters [PI0, PI6], and Saturday hour = 1 can include clusters [PI0, PI3]. Therefore, this is the set of available clusters for each hour (Figure 5.9).

4. Cluster Filtering:

- reduce a set of available clusters to a set of legal clusters for each core sample of each small cluster in each hour;
- compute its fill rate; if it is less than the critical fill rate, this cluster is legal;
- the fill rate is computed by the algorithm. First, we create a subset of core samples of a cluster associated with a given hour. Then, we compute the fraction of observations whose speed is higher than the critical speed. If this fraction is greater than the fill rate, it is ignored;
- following a filtering procedure, we can remove up to 50 of the clusters as noise clusters (the legal points of user 5542 were generated after filtering Npoints = 172928).

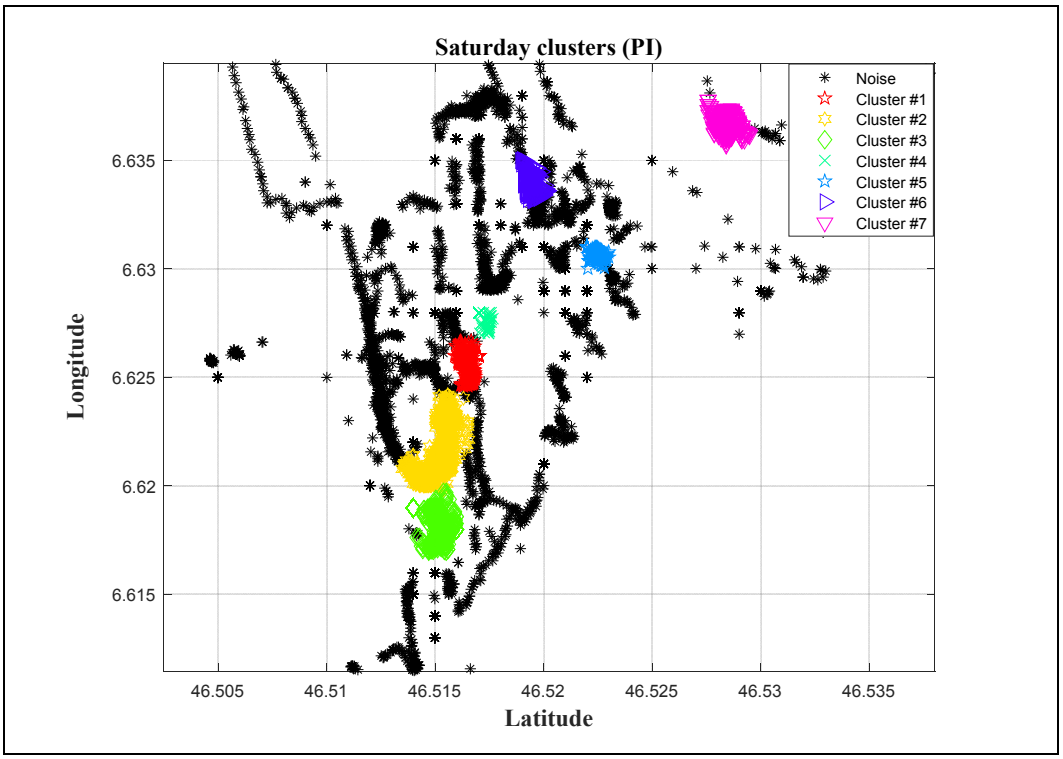

Figure 5.9 Cluster PI Saturday

After the DBSCAN we obtain 10 clusters (7 are shown).
After Filtering we obtain 9 clusters.

ClusterId = 0, hour = 0, Center = [46.51030707 6.64090675]
ClusterId = 6, hour = 0, Center = [46.50975226 6.65531007]
ClusterId = 0, hour = 1, Center = [46.51004731 6.64004524]
ClusterId = 3, hour = 1, Center = [46.45 6.89]

5. We obtain a set of clusters for (”day”, ”hour”).

5.5.1 Prediction process

The data obtained after preprocessing, clustering, and filtering constitute the displacement history of user 5542, modelled as a daily succession of PIs and noise. The construction process of the predicted class ”Next PI” (or the next point of interest) is given in Figure 5.10.

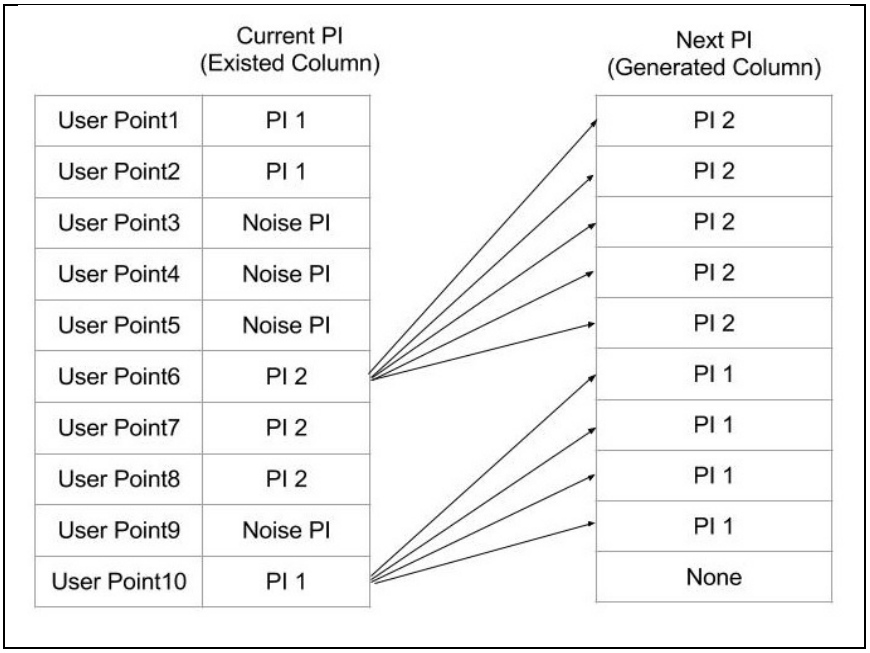

Figure 5.10 Construction of next cluster

In order to validate the generation of the Next PI (the next point of interest) derived from the current contextual information (current location (latitude, longitude), speed, hour, day, PI, noise (noise, high speed), five supervised machine learning algorithms were tested using a 10-fold cross validation (Naive Bayes, Bayes Nets, decision trees J48 (C4.5), KNN nearest neighbours (K = 1), and support vector machine (SVM). To benchmark the classification techniques, we implemented WEKA (Hall et al., 2009), which is a widely used collection of machine learning algorithms for data mining software. The results are shown in Table 5.1.

Table 5.1 Location prediction accuracy with noise (Bold) and without noise (user #5542).

Day	**Naïve Bayes %**		**Bayes Net %**		**Decision trees J48 = C4.5%**		**SVM %**		**Nearest neighbor Lbk (K=1) %**	
	with noise	without noise	**with noise**	Without noise	**with noise**	without noise	**with noise**	without noise	**with noise**	without noise
Saturday	**29.69**	66.62	**77.30**	76.43	**92.37**	92.11	**69.54**	73.19	**93.20**	90.36
Sunday	**31.64**	44.15	**74.25**	70.20	**91.22**	88.22	**59.56**	58.32	**92.46**	87.84
Monday	**52.94**	54.78	**69.73**	69.55	**88.06**	83.94	**65.43**	66.27	**89.47**	86.10
Tuesday	**40.36**	41.31	**65.18**	64.76	**87.58**	86.37	**55.69**	60.81	**89.74**	88.42
Wednesday	**54.80**	65.03	**75.60**	75.86	**92.11**	89.71	**65.04**	71.78	**93.98**	91.95
Thursday	**31.100**	35.28	**61.41**	60.42	**84.72**	83.63	**54.1**	60.17	**86.39**	83.75
Friday	**51.26**	50.80	**67.96**	69.63	**86.72**	87.77	**58.99**	58.99	**89.89**	89.89

The results reported in table 5.1 show that the location prediction methodology is effective, especially for the Bayes Net, decision trees (J48/C4.5), C 4.5, and KNN nearest neighbour classification algorithms. Additionally, the introduction of noise-prediction data improved the prediction (a mean improvement in prediction accuracy with average values of 0.654% for the Bayes Net, 1.575% for C4.5, and 2.102% for KNN).

Graphs of the receiver operating characteristics (ROC) can be used to organise classifiers and to visualise their performance. The graphs depict the relative trade-offs between benefits (true positives) and costs (false positives). Figure 5.11 presents an example of ROC curves with 5 classifiers for 1 PI (point of interest #6) on Saturday1. This is typical of other points of interest for the same day and for the other days of the week.

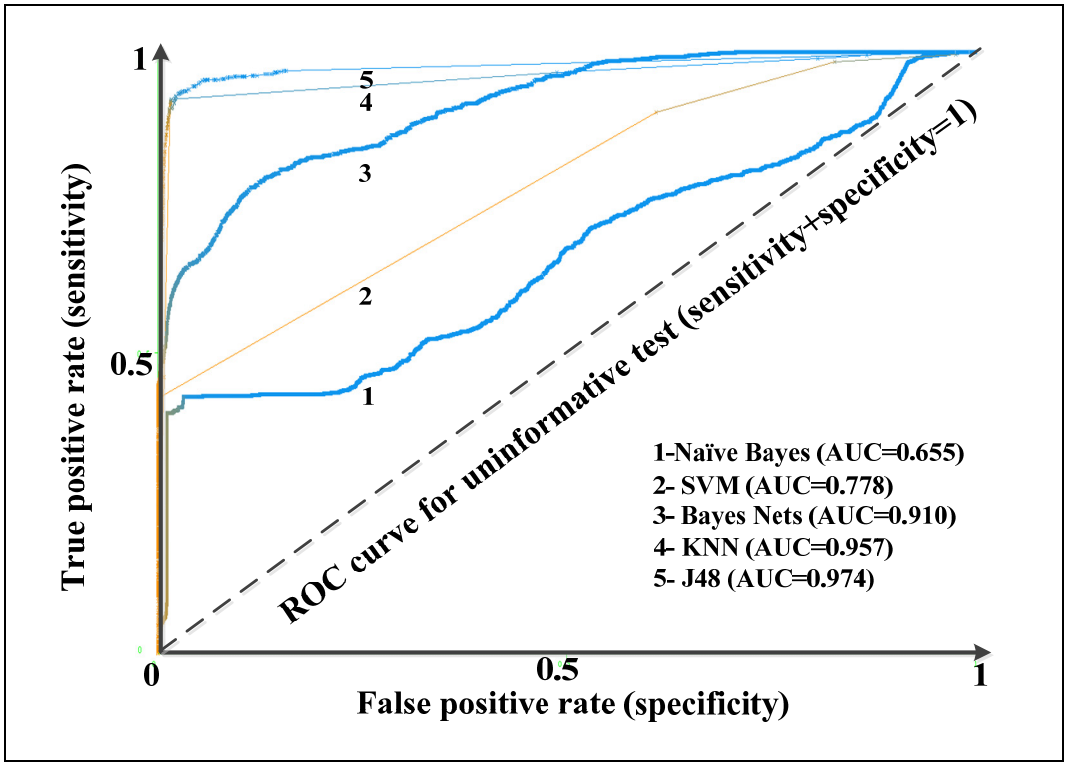

Figure 5.11 Saturday (point of interest #6) ROC (receiver operating characteristic)

Evidently, all AUC (area under the curve) values are higher than 0.5, which is the boundary between a random classification and a positive classification. The prediction algorithms for J48 (C4.5), KNN, and Bayes Nets provide better results than other algorithms.

ROC curves were evaluated with the AUC parameter, which indicates the probability of a classifier classifying a randomly chosen positive instance relative to a randomly chosen negative instance. In our case, the AUC indicates the probability of a selected algorithm positively classifying a point of interest (PI). It is well established that the performance of a classifier with an AUC of 0.9 to 0.99 is excellent. Figure 5.12 shows the mean AUC for the days of the week: obtained AUC values were 0.949 for Bayes Nets, 0.964 for J48 / C4.5, and 0.953 for KNN.

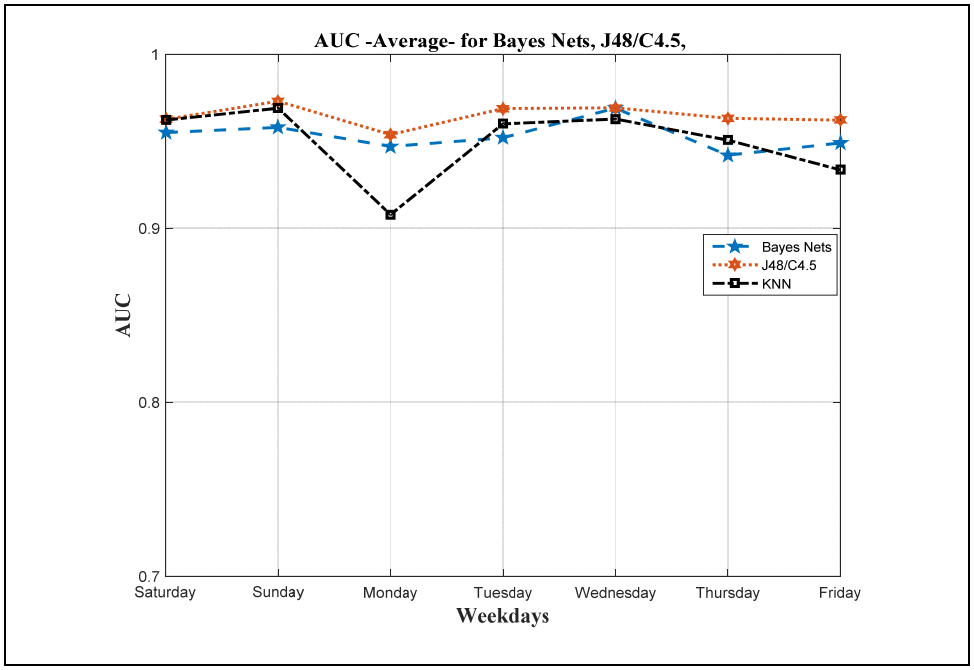

Figure 5.12 Average day-AUC for Bayes Net, J48/C4.5, and KNN

We tested the proposed prediction methodology with the data of a second user, user #5578. The results demonstrate that the prediction precision is similar (Table 5.2).

Table 5.2 Average location prediction accuracy (user #5578)

	Naïve Bayes %	**Bayes Net %**	**Decision trees J48=C4.5 %**	**SVM %**	**Nearest Neighbor Lbk (K=1) %**
Prediction accuracy (average)	67.56	86.54	97.09	77.48	98.09

Comparable accuracies were obtained for the prediction of the next point of interest of the two users (user #5542 and user #5578) and for the evaluation parameters of the classifiers (AUC), which shows that, despite the simplicity of our methodology, interesting results were obtained for the prediction of a user's future location, especially with Bayes Nets, J48 / C4.5, and KNN.

5.6 Conclusion

We propose a methodology for the prediction of a user's outdoor location derived from contextual data (current location, day of the week, time, and speed), which can be collected with a GPS device or with a smartphone This methodology is based on spatial clustering of data and on time segmentation to find points of interest that the user visits every day and every hour. With these points of interest, data considered noise were used to improve the prediction. It may be the subject of future work to determine whether points considered noise are relevant to improving the location prediction.

The results of this classification/prediction, which was based on two user profiles in the MDC dataset, demonstrate the relevance and simplicity of our methodology. We are planning a follow-up study to introduce the semantic identification of points of interest and improve the spatial clustering process to include interesting regions that have an insufficient number of points to create a cluster.

CONCLUSION

Ce travail est un point d'entrée au domaine de l'adaptation des services dans les systèmes informatiques diffus par l'utilisation des similarités contextuelles et aussi pour la prédiction du contexte à partir des informations contextuelles disponibles pour l'acquisition rapide et facile telle que la position d'un utilisateur, sa vitesse de déplacement, le temps et le jour de la semaine. Par similarités contextuelles, nous entendons faire des comparaisons avec des contextes déjà connus que nous appelons « contexte de référence » ou des cas sauvegardés de l'historique des activités de l'utilisateur. Les comparaisons se font respectivement par la mesure de la similarité sémantique entre les informations contextuelles du même type (catégoriques ou quantitatives), et finalement une similarité globale est calculée afin de confirmer qu'un contexte courant est semblable à un contexte connu et dont les services sont également connus. Cette approche de comparaison s'accommode avec l'environnement général des systèmes informatiques diffus où les ressources (traitement et stockage) sont relativement limitées.

Le premier travail a été initié par le besoin de satisfaire nos objectifs initiaux dont la mesure de similarité sémantique représente un concept essentiel à définir et aussi par le fait du manque de ce type d'enquêtes dans la littérature qui aborde les mesures des similarités sémantiques dans le domaine de l'informatique diffuse dans un seul travail. Nous avons réalisé une enquête sur les diverses applications de ces mesures que nous avons finalement catégorisé en quatre catégories d'applications : 1) Comparaison des composants d'une application, 2) Recommandation et classement des services par degré de pertinence, 3) Identification des services en comparant la description d'une requête avec les services disponibles et 4) Comparaison du contexte courant avec des contextes connus.

Le deuxième travail a été une contribution pour l'application de ce type d'adaptation. Nous avons proposé entre autres une pondération des informations contextuelles du type catégorique dont nous avons montré son aspect dynamique et que nous avons comparé avec d'autres approches de pondération. Ce travail peut être étendu et une application réelle peut

être testée avec un choix pertinent des informations contextuelles disponibles et qui représente suffisamment les contextes courant et de référence.

Le troisième travail est une application du même processus de comparaison contextuelle à l'aide des mesures de similarité sémantique en vue de l'adaptation des services dans un système informatique diffus. Cette fois-ci, ces mesures de similarité sémantique font partie du cycle du Raisonnement par cas et sont représentées par la phase de l'extraction des cas ou contextes les plus similaires au cas ou contexte actuel de l'utilisateur. Ce cycle comprend aussi une phase d'adaptation des services extraits ainsi que d'une phase de mise à jour de la base des cas.

Nous avons mis en œuvre ce cycle jusqu'à l'extraction des cas similaires sur lesquels nous avons ajouté une phase de filtrage à l'aide des informations contextuelles actuelles afin d'avoir les services les plus pertinents et qui tiennent compte du contexte de l'utilisateur.
Le quatrième travail est un pas vers la prédiction du contexte et l'adaptation proactive en exploitant les informations contextuelles disponibles. Les travaux dans cet axe sont aussi abondants, mais ceux dont la possibilité d'applications réelles sont rares de par la complexité des algorithmes utilisés et des processus de traitement avares en ressources. Nous avons abordé ce sujet et nous avons ajouté quelques contributions qui peuvent être à la base d'autres recherches afin d'être améliorées comme l'effet du bruit dans les données spatiales dans la prédiction du contexte en général et dans la localisation en particulier.

Nous estimons que nos objectifs sont atteints et que nos contributions sont pertinentes quant à l'application des mesures de similarité sémantique dans le domaine de l'informatique diffuse pour le but de l'adaptation des services. La pondération proposée est une pondération dynamique qui montre bien la contribution des variables contextuelles du type catégorique. La reconnaissance de l'activité physique d'un utilisateur à partir des capteurs d'un smartphone et des caractéristiques statistiques des données collectées est aussi intéressante et peut contribuer à améliorer la pertinence des services fournis. La modification de la mesure de Wu et Palmer ouvre de nouveaux horizons pour son application et enfin la prédiction de la

localisation à partir des données simples à collecter pourra être à la base de nouvelles applications sensible au contexte et proactive.

Les travaux futurs qui peuvent compléter ce travail ainsi que les axes qui restent à explorer peuvent être résumés dans les points suivants : 1) La sélection des informations contextuelles pertinentes à chaque application et qui peut être une phase d'analyse initiale afin de les prioriser selon leur impact sur les mesures de similarité sémantique 2) La définition et la catégorisation des services dans le domaine de l'informatique diffuse. Ce travail permettra de proposer des modèles de services qui améliorent le processus de l'adaptation 3) La sélection des contextes de références et les conditions de la mise à jour de la base des « contextes de référence» ou la base des cas afin de contrôler sa taille tout en restant une base représentative de la diversité des contextes de référence ou des cas auxquels un utilisateur peut faire face et 4) Proposer des travaux qui élaborent plus sur l'effet de données contextuelles considérées comme bruit et étudier leur valeur ajoutée dépendamment de l'application ciblée.

BIBLIOGRAPHIE

Aamodt, A. and Plaza, E.1994.« Case-based reasoning: Foundational issues, methodological variations, and system approaches ». AI communications, vol.7, n° 1, p. 39-59.

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P., 1999. « Towards a better understanding of context and context-awareness ». In Handheld and ubiquitous computing, p. 304-307.

Adomavicius, G., & Tuzhilin, A., 2011. « Context-aware recommender systems ». In Recommender Systems. Handbook. p. 217-253.

Adomavicius, G., & Tuzhilin, A. 2005. « Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions ». Knowledge and Data Engineering, IEEE Transactions, vol.17, (6), p.734-749.

Ahmad El Sayed, Hakim Hacid, and Djamel Zighed, 2008. « Using Semantic Distance in a Content-Based Heterogeneous Information Retrieval System ». Springer-Verlag Berlin Heidelberg, p. 224-237.

Ahmed Khalifa, Alasoud, 2009,« A multi-matching technique for combining similarity measures in ontology integration ». Doctoral thesis , Computer Science and Software Engineering , Montréal, Concordia University, 128 p.

Al-Mubaid, H., & Nguyen, H.A. 2006. « Using MEDLINE as standard corpus for measuring semantic similarity of concepts in the biomedical domain », In Proc. of the IEEE 6th Symposium on Bioinformatics and Bioengineering, p. 315-318.

Ameyed, D., Miraoui, M. and Tadj, C. 2015. « A survey of prediction approach in pervasive computing », International Journal of Scientific & Engineering Research, vol.6, n°5, , p. 306–316.

Anagnostopoulos, T., Anagnostopoulos, C., Hadjiefthymiades, S., Kyriakakos, M. and Kalousis, A. 2009. « Predicting the location of mobile users: a machine learning approach », Proceedings of the 2009 international conference on Pervasive services, ACM, p. 65–72.

Ashbrook, D. and Starner, T. 2003. « Using gps to learn significant locations and predict movement across multiple users » , Personal and Ubiquitous Computing, vol.7, n°5, p. 275–286.

Augusto, J. C., Nakashima, H., & Aghajan, H. 2010. « Ambient intelligence and smart environments: A state of the art ». In Handbook of ambient intelligence and smart environments .p. 3-31.

Aydoğan, R., & Yolum, P. 2007, May. « Learning consumer preferences using semantic similarity ». In Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems .p. 229.

Bandara, A., Payne, T., De Roure, D., & Lewis, T. 2007. « A semantic approach for service matching in pervasive environments ». Technical Report Number: ECSTR-IAM07-006, University of Southampton.

Bar-David, R. and Last, M. 2016. « Context-aware location prediction, Big Data Analytics ». in the Social and Ubiquitous Context: 5th International Workshop on Modeling Social Media, MSM 2014, 5th International Workshop on Mining Ubiquitous and Social Environments, MUSE 2014, p. 165–185.

Bernstein, A., Kaufmann, E., Kiefer, C., & Bürki, C. 2005, « Simpack: A generic java library for similarity measures in ontologies », University of Zurich.

Berkhin, P. 2006. « A survey of clustering data mining techniques, Grouping multidimensional data », Springer, p. 25–71.

Bettini, C.; Brdiczka, O.; Henricksen, K.; Indulska, J.; Nicklas, D.; Ranganathan, A.; Riboni. 2010. « A survey of context modelling and reasoning techniques ». Pervasive and Mobile Computing. vol.6, n°2, p. 161-180.

Bisson, G. 2000, « La similarite: Une notion symbolique/numerique ». IMAG-CNRS, Projet SHERPA, Unité de recherche INRIA Rhone-Alpes .p. 3.

Bolchini, C., Curino, C. A., Quintarelli, E., Schreiber, F. A., & Tanca, L. 2007. « A data-oriented survey of context models ». ACM Sigmod Record, vol.36, n°4, p.19-26.

Boytsov, A. and Zaslavsky, A. B. 2010. « Context prediction in pervasive computing systems: Achievements and challenges », in F. Burstein, P. Brzillon and A. B. Zaslavsky (Eds), Supporting Real Time Decision-Making, Vol. 13 of Annals of Information Systems, Springer, p. 35–63.

Brézillon, P., & Pomerol, J. C. 1999, « Contextual knowledge and proceduralized context ». In Proceedings AAAI Workshop on Reasoning in Context for AI Application. AAAI Technical Report WS-99, vol. 14, p. 16-20.

Broens, T., Pokraev, S., Van Sinderen, M., Koolwaaij, J., & Costa, P. D. 2004. « Context-aware, ontology-based service discovery ». In Ambient Intelligence .p. 72-83.

Buchin, M., Dodge, S. and Speckmann, B. 2014. « Similarity of trajectories taking into account geographic context. », J. Spatial Information Science, vol.9, n°1, p. 101–124.

Bulskov H., Knappe R., & Andreasen T. 2002. « On measuring similarity for conceptual querying ». In the Proc. of the 5th Int'l Conf. on Flexible Query Answering Systems , p. 100-111.

Cao, X., Cong, G., & Jensen, C. S. 2010. « Mining significant semantic locations from GPS data ». Proceedings of the VLDB Endowment, vol.3, n°1-2, p.1009-1020.

Capra, L., Emmerich, W., & Mascolo, C. 2001. « Reflective middleware solutions for context-aware applications ». In Metalevel Architectures and Separation of Crosscutting Concerns , p. 126-133.

Chang, J., & Song, J. 2012. « Research on context-awareness service adaptation mechanism in IMS under ubiquitous network ». In Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th , p. 1-5.

Chantal.Taconet , 2012. « Apport des modèles pour l'adaptation d'applications en environnement ubiquitaire », Chantal.Taconet@telecom-sudparis.eu,MOPS-RM.

Chen, A. 2005. « Context-aware collaborative filtering system: Predicting the user's preference in the ubiquitous computing environment ». In Location-and Context-Awareness , p. 244-253.

Chen, Harry, Tim Finin, and Amupam Joshi. 2005. « Semantic web in the context broker architecture ». Maryland Univ. Baltimore dept. of computer science and Electrical Engineering.

Chon, J., & Cha, H. 2011. « Lifemap: A smartphone-based context provider for location-based services ». IEEE Pervasive Computing, vol.2, p.58-67.

Constantin Schmidt, 2007, « Context-Aware Computing », Berlin Institute of Technology, Germany.

Cook, D. J., Youngblood, M., Heierman III, E. O., Gopalratnam, K., Rao, S., Litvin, A. and Khawaja, F. 2003. « Mavhome: An agent-based smart home », null, IEEE, p. 521.

d'Amato, Claudia 2007. « Similarity-based learning methods for the semantic web », PhD thesis, Universita Degli Studi di Bari Faculta di Scienze Dipartimento di Informatica, p 97 .

d'Amato, C., Fanizzi, N., & Esposito, F. 2009. « A semantic similarity measure for expressive description logics ». Universita Degli Studi di Bari Faculta di Scienze Dipartimento di Informatica arXiv preprint arXiv:0911.5043.

Daniela Nicklas, Karen Henricksen, 2008. « Context modeling and reasoning: Key concepts for Pervasive computing ». 5th IEEE Workshop on Context Modeling and Reasoning (CoMoRea'08) @PerCom Hong Kong, 17 or 21 March 2008.

De Mantaras, R. L., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S. & Keane, M. 2005. « Retrieval, reuse, revision and retention in case-based reasoning ». The Knowledge Engineering Review, vol.20, n°03.

Desai, A., Singh, H. and Pudi, V., 2011. « Disc: Data-intensive similarity measure for categorical data ». In Advances in Knowledge Discovery and Data Mining, p. 469-481.

Dey, A.K. 2001. « Understanding and using context ». College of Computing & GVU Center, Georgia Institute of Technology, Atlanta,Personal and Ubiquitous Computing, vol.5, p. 4-7.

Dietze, S., Gugliotta, A., & Domingue, J. 2008. « Bridging the gap between mobile application contexts and semantic web resources: Context-aware mobile and ubiquitous computing for enhanced usability: adaptive technologies and applications ». Information Science Publishing (IGI Global).

Do, T. M. T. and Gatica-Perez, D. 2012. « Contextual conditional models for smartphone-based human mobility prediction », Proceedings of the 2012 ACM conference on ubiquitous computing, ACM, p. 163–172.

Dourish, P. 2004. « What we talk about when we talk about context ». Personal and ubiquitous computing, vol.8, n°1, p.19-30.

Doulkeridis, C., Loutas, N., & Vazirgiannis, M. 2006. « A system architecture for context-aware service discovery ». Electr. Notes Theor. Comput. Sci., vol.146, n°1, p.101-116.

E. S. Smirnov. 1968. « On exact methods in systematics ». Systematic Zoology, vol.17, n°1, p. 1–13.

Eagle, N., Clauset, A. and Quinn, J. A. 2009. « Location segmentation, inference and prediction for anticipatory computing ». AAAI Spring Symposium: Technosocial Predictive Analytics, AAAI, p. 20–25.

Efstratiou, C. 2004. « Coordinated adaptation for adaptive context-aware applications ». Doctoral dissertation, Computing Department, Lancaster University, UK, 173 p.

Ehrig, M., Haase, P., Hefke, M., & Stojanovic, N. 2005. « Similarity for ontologies-a comprehensive framework ». ECIS 2005 Proceedings.

El Sayed, A., Hacid, H., & Zighed, D. 2007. « A new context-aware measure for semantic distance using a taxonomy and a text corpus ». In Information Reuse and Integration. IEEE International Conference, p. 279-284.

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. 1996. « A density-based algorithm for discovering clusters in large spatial databases with noise ». Kdd, vol. 96, p. 226–231.

Ferit Topcu. 2011. « Context Modeling and Reasoning Techniques » In SNET Seminar in the ST.

Frank, Andrew U. 2001. « Tiers of ontology and consistency constraints in geographical information systems ». International Journal of Geographical Information Science vol.15, n°7, p. 667-678.

G. Chen and D. Kotz. 2000 « A survey of context-aware mobile computing research », Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, p.1–16.

G. Prasanna, et al., 2003, « Exploiting hierarchical domain structure to compute similarity », ACM Trans. Inf. Syst., vol. 21, p. 64-93.

Gabel, T., & Stahl, A. 2004. « Exploiting background knowledge when learning similarity measures ». In Advances in Case-Based Reasoning , p. 169-183.

Ganter, B., & Stumme, G., 2002, « Formal concept analysis: Methods and applications in computer science ». TU Dresden, http://www.aifb.uni-karlsruhe.de/WBS/gst/FBA03.shtml.

García-Crespo, A., Chamizo, J., Rivera, I., Mencke, M., Colomo-Palacios, R., & Gómez-Berbís, J.M. 2009. « SPETA: Social pervasive e-tourism advisor. Telematics and Informatics », vol.26, n°3, p.306-315.

Ge, J., & Qiu, Y. 2008. « Concept similarity matching based on semantic distance ». In Semantics, Knowledge and Grid, 2008. SKG'08. Fourth International Conference, p. 380-383.

Germán Sancho, 2010. « Adaptation d'architectures logicielles collaboratives dans les environnements ubiquitaires.Contribution à l'interopérabilité par la sémantique.». Doctoral dissertation, Systèmes (EDSYS), France, 151 p.

Gicquel, P.Y 2012. « Similarités sémantiques et contextuelles pour l'apprentissage informel en mobilité ». RJC EIAH'2012, 45.

Gomaa, W.H., & Fahmy, A.A. 2013. « A survey of text similarity approaches ». International Journal of Computer Applications, vol.68, n°13, p.13-18.

Gonzalez-Castillo, J., Trastour, D., & Bartolini, C. 2001. « Description logics for matchmaking of services ». HP Laboratories technical report, 265.

Gong, L., Sato, H., Yamamoto, T., Miwa, T. and Morikawa, T. 2015. « Identification of activity stop locations in gps trajectories by density-based clustering method combined with support vector machines », Journal of Modern Transportation, vol.23, n°3, p. 202–213.

Gopalratnam, K. and Cook, D. J. 2004. « Active lezi: an incremental parsing algorithm for sequential prediction ». International Journal on Artificial Intelligence Tools, vol.13, n°4, p. 917–930.

Gopalratnam, K. and Cook, D. J. 2007. « Online sequential prediction via incremental parsing: The active lezi algorithm ». IEEE Intelligent Systems, vol.22, n°1, p. 52–58.

Gowda, K. C., & Diday, E., 1992, « Symbolic clustering using a new similarity measure: Systems, Man and Cybernetics », IEEE Transactions, vol.22, n°2, p.368-378.

Greenberg, S. 2001. « Context as a dynamic construct. Human-Computer Interaction », vol.16, n°2, p. 257-268.

Gu, Tao, Xiao Hang Wang, Hung Keng Pung, and Da Qing Zhang. 2004. « An ontology-based context model in intelligent environments ». In Proceedings of communication networks and distributed systems modeling and simulation conference, vol. 2004, p. 270-275.

Guarino, Nicola. 1998. « Formal ontology in information systems ». Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy. vol. 46. IOS press, 1998.

Guessoum, D., Miraoui, M., and Tadj, C. 2015. « Survey of semantic similarity measures in pervasive computing ». International journal on smart sensing and intelligent systems vol.8, n°1, p.125– 158.

Gouttaya, N., & Begdouri, A. 2011, May. « The quality integrating data mining with Case Based Reasoning for personalized adaptation of context-aware applications in pervasive environments ». In Information Science and Technology (CIST), 2011 Colloquium in , p. 13-14.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. 2009. « The weka data mining software: an update », ACM SIGKDD explorations newsletter, vol.11, n°1, p. 10–18.

Harispe, S., Ranwez, S., Janaqi, S., & Montmain, J. 2013. « Semantic measures for the comparison of units of language, concepts or instances from text and knowledge representation analysis, A Comprehensive Survey and a Technical Introduction to Knowledge-based Measures Using Semantic Graph Analysis », LGI2P/EMA Research Center, Parc scientifique, France.

Hartmann, M., Zesch, T., Muhlhauser, M., & Gurevych, I. 2008. « Using similarity measures for context-aware user interfaces ». In Semantic Computing, 2008 IEEE International Conference on , p. 190-197.

Henricksen, K., Indulska, J., & Rakotonirainy, A. 2006. « Using context and preferences to implement self-adapting pervasive computing applications ». Software: Practice and Experience, vol.36, n°11-12, p.1307-1330.

Hirst, G., & St Onge, D. 1998. « Lexical chains as representations of context for the detection and correction of malapropisms ». In C. Fellbaum (ed.), WordNet: An Electronic Lexical Database, Cambridge, MA: The MIT Press.

Incel, O. D., Kose, M., & Ersoy, C. 2013. « A review and taxonomy of activity recognition on mobile phones ». BioNanoScience, vol.3, n°2, p. 145-171.

J. Mylopoulos, C. Quix, C. Rolland, Y. Manolopoulos, H. Mouratidis and J. Horkoff (Eds), CAiSE, Vol. 8484 of « Lecture Notes in Computer Science », Springer, p. 58–74.

J. Ye, L. Coyle, S. Dobson, and P. Nixon, 2007. « Ontology-based models in pervasive computing systems », The Knowledge Engineering Review, vol. 22, p. 315–347.

Janowicz, K. 2008. « Kinds of contexts and their impact on semantic similarity measurement ». In Pervasive Computing and Communications. Sixth Annual IEEE International Conference, p. 441-446.

Jérôme Simonin, Carbonell, N., 2007, « Interfaces adaptatives, Adaptation dynamique à l'utilisateur courant ». arXiv preprint arXiv:0708.3742.

Jiang J.J., & Conrath D.W. 1997. « Semantic similarity based on corpus statistics and lexical taxonomy ». Proceedings of International Conference on Research in Computational Linguistics, August 22-24; Taipei, Taiwan.

Kakousis, K., Paspallis, N., & Papadopoulos, G.A. 2010. « A survey of software adaptation in mobile and ubiquitous computing ». Enterprise Information Systems, vol.4, n°4, p.355-389.

Kang, S., Kim, D., Lee, Y., Hyun, S.J., Lee, D., & Lee, B. 2007. « A semantic service discovery network for large-scale ubiquitous computing environments ». ETRI journal, vol.29, n°5, p. 545-558.

Kayes, A. S. M., Han, J., & Colman, A. 2014, January. « PO-SAAC: A Purpose-Oriented Situation-Aware Access Control Framework for Software Services ». InAdvanced Information Systems Engineering , p. 58-74.

Ke Ning and David O'Sullivan, 2012. « Context Modeling and Measuring for Context Aware Knowledge Management ». International Journal of Machine Learning and Computing, vol. 2, no. 3, p. 287-291.

Keßler, C. 2007. « Similarity measurement in context ». In Modeling and Using Context , p. 277-290.

Keßler, C., Raubal, M., & Janowicz, K. 2007. « The effect of context on semantic similarity measurement ». In On the Move to Meaningful Internet Systems: OTM 2007 Workshops , p. 1274-1284.

Kirsch-Pinheiro, M., Vanrompay, Y., & Berbers, Y. 2008. « Context-aware service selection using graph matching ». In 2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08), ECOWS. CEUR Workshop proceedings, vol. 411.

Kirsch-Pinheiro, M., Villanova-Oliver, M., Gensel, J., & Martin, H. 2006. « A personalized and context-aware adaptation process for web-based groupware systems ». In 4th International Workshop on Ubiquitous Mobile Information and Collaboration Systems, CAiSE'06 Workshop , p. 884-898.

Kisilevich, S., Mansmann, F., Nanni, M. and Rinzivillo, S. 2009. « Spatio-temporal clustering », Springer.

Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D. and Laurila, J. 2010. « Towards rich mobile phone datasets: Lausanne data collection campaign », Proc. ICPS, Berlin .

Klein, M., & Bernstein, A. 2004. « Towards high-precision service retrieval ». IEEE Internet Computing, January, 30-36.

Knig, I., Klein, B. N. and David, K. 2013. « On the stability of context prediction. », in F. Mattern, S. Santini, J. F. Canny,

Knox, S., Coyle, L., & Dobson, S. 2010. « Using ontologies in case-based activity recognition ».

Kofod-Petersen, A., & Aamodt, A. 2003, October. « Case-based situation assessment in a mobile context-aware system ». In Proceedings of AIMS2003, Workshop on Artificial Intgelligence for Mobil Systems, Seattle.

Kofod-Petersen, A., & Aamodt, A. 2009. « Case-based reasoning for situation-aware ambient intelligence: A hospital ward evaluation study ». In Case-Based Reasoning Research and Development , p. 450-464.

Kolodner, J. L. 1996. « Making the implicit explicit: Clarifying the principles of case-based reasoning. Case-based Reasoning: Experiences, Lessons and Future Directions », p. 349-370.

Kolodner, J. 2014. « Case-based reasoning ». Morgan Kaufmann

Korpipää, P. and Mäntyjärvi, J., 2003, « An ontology for mobile device sensor-based context awareness », Proceedings of CONTEXT, of Lecture Notes in Computer Science, vol. 2680, p.451–458.

Krumm, J. and Horvitz, E. 2006. « Predestination: Inferring destinations from partial trajectories. », in P. Dourish and A. Friday (Eds), Ubicomp, vol. 4206 of Lecture Notes in Computer Science, Springer, p. 243–260.

Krumm, J. and Horvitz, E. 2007. « Predestination: Where do you want to go today? », IEEE Computer, vol.40, n°4, p. 105–107.

Kwon, O. B., & Sadeh, N. 2004. « Applying case-based reasoning and multi-agent intelligent system to context-aware comparative shopping. Decision Support Systems », vol.37, n°2, p. 199-213.

Lara, O. D., & Labrador, M. A. 2013. « A survey on human activity recognition using wearable sensors ». Communications Surveys & Tutorials, IEEE, vol.15, n°3, p.1192-1209.

Laurila, Juha K., Daniel Gatica-Perez, Imad Aad, Olivier Bornet, Trinh-Minh-Tri Do, Olivier Dousse, Julien Eberle, and Markus Miettinen. 2012. « The mobile data challenge: Big data for mobile computing research. ». In Pervasive Computing, no. EPFL-CONF-192489. 2012.

Lavirotte, S., Lingrand, D., & Tigli, J.Y. 2005. « Définition du contexte: fonctions de coût et méthodes de sélection ». In Proceedings of the 2nd French-speaking Conference on Mobility and Ubiquity Computing , p. 9-12.

Leacock, C., & Chodorow, M., 1998, « Combining local context and WordNet similarity for word sense identification », In WordNet: An electronic lexical database, MIT Press, p. 265-283.

Leake, D. B. 2003. « Case-based reasoning ».

Leake, D., & Jalali, V. 2014. « Context and Case-Based Reasoning ». In Context in Computing , p. 473-490. Springer New York.

Lee, J.S., & Lee, J.C. 2007. « Context awareness by case-based reasoning in a music recommendation system ». In Ubiquitous Computing Systems , p. 45-58.

Lee, K. C. and Cho, H. 2010. « Performance of ensemble classifier for location prediction task: emphasis on markov blanket perspective », International Journal of u-and e-Service, Science and Technology, vol.3, n°3, p. 2010.

Lee, S., Lee, K. C. and Cho, H. 2010. « A dynamic bayesian network approach to location prediction in ubiquitous computing environments. », in B. Papasratorn,

Li, L., & Horrocks, I. 2004. « A software framework for matchmaking based on semantic web technology ». International Journal of Electronic Commerce, vol.8, n°4, p.39-60.

Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W., & Ma, W.Y. 2008. « Mining user similarity based on location history ». In Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems , p. 34.

Li, Y., Bandar, Z., & McLean, D., 2003, « An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources », IEEE Transactions on Knowledge and Data Engineering, vol.15, p.871-882.

Liao, T. Warren. « Clustering of time series data—a survey.2005. " Pattern recognition, vol.38, no. 11, p. 1857-1874.

Lin, D., 1998, « An Information-Theoretic Definition of Similarity », In J. Shavlik (Ed.), Fifteenth International Conference on Machine Learning, Madison, Wisconsin, USA: Morgan Kaufmann , ICML 1998, p. 296-304.

Liu, Q., Ma, H., Chen, E., & Xiong, H. 2013. « A survey of context-aware mobile recommendations ». International Journal of Information Technology & Decision Making, vol.12, n°1, p. 139-172.

Liwei Liu, Freddy Lecue, Nikolay Mehandjiev, Ling Xu, 2010, « Using context similarity for service recommendation », In Semantic Computing (ICSC), IEEE Fourth International Conference.p. 277-284.

Lino, J. A., Salem, B., & Rauterberg, M. 2010. « Responsive environments: User experiences for ambient intelligence ». Journal of ambient intelligence and smart environments, vol.2, n°4, p. 347-367.

Lotfy Abdrabou, E. A. M., & Salem, A. 2010. « A breast cancer classifier based on a combination of case-based reasoning and ontology approach ». In Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference on , p. 3-10.

Ma, T., Kim, Y. D., Ma, Q., Tang, M., & Zhou, W. 2005. « Context-aware implementation based on CBR for smart home ». In Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference, vol. 4, p. 112-115.

Maedche, A., & Staab, S. 2002. « Measuring similarity between ontologies ». In Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web , p. 251-263.

Manohar, M. G. 2011. « A study on similarity measure functions on engineering materials selection ».

Mayrhofer, R. 2005. « Context prediction based on context histories: Expected benefits, issues and current state-of-the-art », Cognitive science research paper-University of Sussex CSRP 577: 31.

Marc Dalmau, Philippe Roose, Sophie Laplace. 2009. « Context Aware Adaptable Applications - A global approach ». IJCSI International Journal of Computer Science vol.1, n°1, p. 1-13.

McGovern, J. 2013. « Context similarity evaluation: Inferring how users can collectively collaborate together in a pervasive environment ». In Cloud and Green Computing (CGC), 2013 Third International Conference on , p. 553-557.

Meissen, U., Pfennigschmidt, S., Voisard, A., & Wahnfried, T. 2005, January. « Context-and situation-awareness in information logistics ». In Current Trends in Database Technology-EDBT 2004 Workshops , p. 335-344.

Meng, L., Huang, R., & Gu, J. 2013. « A review of semantic similarity measures in wordnet ». International Journal of Hybrid Information Technology, vol.6, n°1, p.1-12.

Michel, M.D., & Deza, E. 2007. « Dictionnaire des distances. In Encyclopedia of Distances ».

Mihalcea, R., Corley, C., & Strapparava, C. 2006. « Corpus-based and knowledge-based measures of text semantic similarity ». In AAAI, vol. 6, p. 775-780.

Moeiz Miraoui, C. Tadj, and H. Belgacem. 2013. « Dynamic context-aware adaptation of mobile phone incoming call indication using context similarity », in: World Congress Computer and Information Technology (WCCIT), IEEE, p. 1–5.

Moeiz Miraoui, Chakib Tadj, Chokri ben Amar. 2009. « Dynamic Context-Aware Service Adaptation in a Pervasive Computing System ». Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, p. 77.

Moeiz Miraoui, Tadj, C. 2007, « A service oriented definition of context for pervasive computing ». In Proceedings of the 16th International Conference on Computing, IEEE computer society press, Mexico city, Mexico.

Mohamed Zouari, 2011. « Architecture logicielle pour l'adaptation distribuée: Application à la réplication de données », Doctoral dissertation, Université Rennes 1, France, 144 p.

Mokhtar, S.B., Preuveneers, D., Georgantas, N., Issarny, V., & Berbers, Y. 2008. « EASY: Efficient semAntic Service discoverY ». in pervasive computing environments with QoS and context support. Journal of Systems and Software, vol.81, n°5, p.785-808.

Montani, S. 2011. « How to use contextual knowledge in medical case-based reasoning systems: A survey on very recent trends ». Artificial intelligence in medicine, vol.51, n°2, p.125-131.

Moon, H. J., Kim, S., Moon, J., & Lee, E. S. 2008. « An Effective data processing method for fast clustering ». In Computational Science and its Applications–ICCSA 2008 , p. 335-347.

Morzy, M. 2007. « Mining frequent trajectories of moving objects for location prediction », in P. Perner (Ed.), MLDM, vol. 4571 of Lecture Notes in Computer Science, Springer, p. 667– 680.

Nazerfard, E. and Cook, D. J. 2015. « Crafft: an activity prediction model based on bayesian networks », J. Ambient Intelligence and Humanized Computing, vol.6, n°2, p.193–205.

Nwiabu, N., Allison, I., Holt, P., Lowit, P., & Oyeneyin, B. 2011, February. « Situation awareness in context-aware case-based decision support ». InCognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference, p. 9-16.

Öztürk, P., & Aamodt, A. 1998. « A context model for knowledge-intensive case-based reasoning ». International Journal of Human-Computer Studies, vol.48, n°3, p. 331-355.

P. Andritsos, P. Tsaparas, R.J. Miller, and K.C. Sevcik. 2004. « Limbo: Scalable clustering of categorical data », in: Advances in Database Technology-EDBT 2004. Springer, Berlin, p. 123–146.

P. Gambaryan. 1964. « A mathematical model of taxonomy ». Izvest. Akad. Nauk Armen. SSR, vol.17, n°12.

Pantic, M. 2005. « Introduction to Machine Learning & Case-Based Reasoning ». London: Imperial College.

Paolucci, M., Kawamura, T., Payne, T.R., & Sycara, K. 2002. « Semantic matching of web services capabilities ». Lecture Notes in Computer Science, 2342, p.333–347.

Patterson, D. J., Liao, L., Fox, D. and Kautz, H. 2003. « Inferring high-level behavior from low- level sensors », Ubi- Comp 2003.

Petit, M., 2005, « L'informatique contextuelle ». Technical Report, South Britany University (UBS), Vannes, 2005, p. 7.

Perttunen, M., Riekki, J., & Lassila, O. 2009. « Context representation and reasoning in pervasive computing: a review ». International Journal of Multimedia and Ubiquitous Engineering, vol.4, n°4.

Petrakis, E. G. M., Varelas, G., Hliaoutakis, A., & Raftopoulou, P., 2006, « X-Similarity:Computing Semantic Similarity between Concepts from Different Ontologies », Journal of Digital Information Management, 4, p.233-237.

Petzold, J., Bagci, F., Trumler, W. and Ungerer, T. 2003a. « Global and local state context prediction », Artificial Intelligence in Mobile Systems.

Petzold, J., Bagci, F., Trumler, W. and Ungerer, T. 2003b. « The state predictor method for context prediction », Adjunct Proceedings Fifth International Conference on Ubiquitous Computing, Citeseer, p. 135–147.

Petzold, J., Bagci, F., Trumler, W. and Ungerer, T. 2005. « Next location prediction within a smart office building », Cognitive Science Research Paper-University of Sussex CSRP 577: 69.

Petzold, J., Bagci, F., Trumler, W., Ungerer, T. and Vintan, L. 2004. « Global state context prediction techniques applied to a smart office building », The communication networks and distributed systems modeling and simulation conference.

Phithakkitnukoon, S., Horanont, T., Di Lorenzo, G., Shibasaki, R., & Ratti, C. 2010. « Activity-aware map: Identifying human daily activity pattern using mobile phone data ». In Human Behavior Understanding , p. 14-25.

Pirró, G., & Euzenat, J. 2010. « A feature and information theoretic framework for semantic similarity and relatedness ». In The Semantic Web–ISWC 2010 , p. 615-630.

Pla, A., Coll, J., Mordvaniuk, N., & López, B. 2014. « Context-Aware Case-Based Reasoning ». In Mining Intelligence and Knowledge Exploration , p. 229-238.

Preuveneers, D., Victor, K., Vanrompay, Y., Rigole, P., Pinheiro, M.K., & Berbers, Y. 2009. « Context-aware adaptation in an ecology of applications ». Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability: Adaptive Technologies and Applications, p.1-25.

Rada, R., Bicknell, H., Mili, E., & Blettner, M 1989. « Development and application of a metric on semantic nets ». IEEE Transaction on Systems, Man, and Cybernetics, vol.1, n°19, p.17-30.

Ramparany, F., Benazzouz, Y., Gadeyne, J., & Beaune, P. 2011. « Automated context learning in ubiquitous computing environments ». In SSN , p. 9-21.

Ranganathan, A., Shankar, C., & Campbell, R. 2005. « Application polymorphism for autonomic ubiquitous computing ». Multiagent and Grid Systems, vol.1, n°2, p.109-129.

Resnik, P., 1995, « Using Information Content to Evalutate Semantic Similarity in a Taxonomy », In C. S.Mellish (Ed.), 14th International Joint Conference on Artificial Intelligence, Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc.,IJCAI 1995, vol. 1, p.448-453.

Richter, M. M. 1995, January. « On the notion of similarity in case-based reasoning ». In Proceedings of the ISSEK94 Workshop on Mathematical and Statistical Methods in Artificial Intelligence , p. 171-183.

Richter, M. M., & Weber, R. O. 2013. « Case-based reasoning ». A Textbook, 546.

Roth-Berghofer, T., Sauer, C., Garcia, J. A. R., Bach, K., Althoff, K. D., & Agudo, B. D. 2008. « Building case-based reasoning applications with myCBR and COLIBRI studio ». In Proceedings of the UKCBR 2012 Workshop.

Rodríguez, M. A., & Egenhofer, M. J., 2003, « Determining semantic similarity among entity classes from different ontologies », IEEE Transactions on Knowledge and Data Engineering, 15, p.442–456.

Rubinstein, H. & Goodenough, J.B. 1965. « Contextual correlates of synonymy ». Communications of the ACM, vol.8, n°10.

Ruta, M., Scioscia, F., Di Sciascio, E., & Piscitelli, G. 2012. « Semantic matchmaking for location-aware ubiquitous resource discovery ». International Journal on Advances in Intelligent Systems, vol.4, n°3/4, p.113-127.

S. Boriah, V. Chandola, and V. Kumar. 2008.« Similarity measures for categorical data: A comparative evaluation », RED, vol.30, n°2, p.3.

S. Dobson, J. Ye, 2006, « Using fibrations for situation identification », Pervasive workshop proceedings, p.645–651.

Sánchez, D., Batet, M., Isern, D., & Valls, A., 2012, « Ontology-based semantic similarity: A new feature-based approach », Expert Systems with Applications, vol.39, n°9, p.7718-7728.

Sangkeun Lee, Juno Chang, Sang-goo Lee. 2011. « Survey and Trend Analysis of Context-Aware Systems ». Information-An International Interdisciplinary Journal, vol. 13, no Copyright 2010, The Institution of Engineering and Technology, vol. 14, n°2, p. 527-548

Saruladha, K., Aghila, G., & Raj, S. 2010, February. « A survey of semantic similarity methods for ontology based information retrieval ». In Machine Learning and Computing (ICMLC), 2010 Second International Conference on , p. 297-301.

Saruladha, K. 2011. « Semantic similarity measures for information retrieval systems using ontology ». Doctoral dissertation, Department of Computer Science, School of Engineering and Technology, Pondicherry University, 207 p.

Scellato, S., Musolesi, M., Mascolo, C., Latora, V. and Campbell, A. T. 2011. « Nextplace: a spatio-temporal prediction framework for pervasive systems », Pervasive computing, Springer, p. 152–169.

Schank, R. C. 1983. « Dynamic memory: A theory of reminding and learning in computers and people ». Cambridge University Press.

Schilit, B. N., & Theimer, M. M. 1994. « Disseminating active map information to mobile hosts ». Network, IEEE, vol.8, n°5, p.22-32.

Schilit, B., Adams, N., & Want, R. 1994, December. « Context-aware computing applications ». In Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on , p. 85-90.

Schmid, F., & Richter, K. F. 2006, September. « Extracting places from location data streams ». In International Workshop on Ubiquitous Geographical Information Services, Munster, Germany.

Schmidt, A., Beigl, M. and Gellersen, H.W., 1999. « There is more to context than location ». Computers & Graphics, vol.23, n°6, p.893-901

Sharma, L., & Gera, A. 2013. « A survey of recommendation system: Research challenges ». International Journal of Engineering Trends and Technology (IJETT), vol.4, n°5, p.1989-1992.

Shet, K. C., & Acharya, U. D. 2012. « A New Similarity Measure for Taxonomy Based on Edge Counting ». arXiv preprint arXiv:1211.4709.

Shoaib, M., Scholten, H., & Havinga, P. J. 2013, December. « Towards physical activity recognition using smartphone sensors ». In Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC), p. 80-87.

Shoaib, M., Bosch, S., Incel, O. D., Scholten, H., & Havinga, P. J. 2015. « A Survey of Online Activity Recognition Using Mobile Phones ». Sensors, vol.15, n°1.

Schougaard, K. 2007. « Vehicular mobility prediction by bayesian networks », DAIMI Report Series 36(582). Sigg, S., Haseloff, S. and David, K. 2010. An alignment approach for context prediction tasks in ubicomp environments., IEEE Pervasive Computing, vol.9, n°4, p. 90–97.

Skovsgaard, A., Sidlauskas, D. and Jensen, C. S. 2014. « A clustering approach to the discovery of points of interest from geo-tagged microblog posts », in A. B. Zaslavsky, P. K. Chrysanthis, C. Becker, J. Indulska, M. F. Mokbel, D. Nicklas and C.-Y. Chow (Eds), MDM (1), IEEE, p. 178–188.

Song, M. R., Moon, J. Y., & Bae, S. H. 2013. « A Design and Implement Contexts- Aware Case based u-Health System ». International Journal of Bio-Science & Bio-Technology, vol.5, n°5.

Stanfill, C., & Waltz, D. 1986. « Toward memory-based reasoning ». Communications of the ACM, vol.29, n°12, p.1213-1228.

Strang, T., & Linnhoff-Popien, C., 2004, « A context modeling survey », In Workshop Proceedings.

Sung-Hyuk Cha. 2007. « Comprehensive Survey on Distance Similarity Measures between Probability Density Functions ». International journal of mathematical models and methods in applied sciences , Issue 4, Volume 1, p. 300-307.

Sussna M. 1993. « Word sense disambiguation for free-text indexing using a massive semantic network ». In Proc. of Second Int'l Conf. Information Knowledge Management (CIKM '93).

Tarek C., 2007. « Adaptation d'applications pervasives dans des environnements multi-contextes ». These de Phd, Laboratoire d'Informatique en Image et Systèmes d'information, INSA de Lyon , p.38-44.

Thabet Slimani, Boutheina Ben Yaghlane & Khaled Mellouli. 2007, « Une extension de mesure de similarité entre les concepts d'une ontologie ». 4rth International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, Tunisia, p. 1-10.

Thompson, M.S. 2006. « Service discovery in pervasive computing environments ». Doctoral dissertation, Virginia Polytechnic Institute and State University, 135 p.

Tversky, A., 1977, « Features of Similarity », Psycological Review, vol.84, no 4, p.327-352.

Van Setten, M., Pokraev, S., & Koolwaaij, J. 2004. « Context-aware recommendations in the mobile tourist application COMPASS ». In Adaptive Hypermedia and Adaptive Web-based Systems , p. 235-244.

van Kasteren, T. and Krose, B. 2007. « Bayesian activity recognition in residence for elders », Intelligent Environments, 2007. IE 07. 3rd IET International Conference on, IET, p. 209– 212.

Viterbo, J., Mazuel, L., Charif, Y., Endler, M., Sabouret, N., Breitman, K., & Briot, J. 2008. « Ambient intelligence: Management of distributed and heterogeneous context knowledge ». In CRC Studies in Informatics Series , p. 1-44.

Voigtmann, C. 2014. « An algorithmic approach for collaborative-based prediction of user contexts in ubiquitous environments under consideration of legal implications ». URL: http://nbn-resolving.de/urn:nbn:de:hebis:34- 2014021945130

Voigtmann, C. and David, K. 2012. « A survey to location-based context prediction », in Springer (Ed.), AwareCast 2012 , pervasive.

Watson, I. 1999. « Case-based reasoning is a methodology not a technology ». Knowledge-based systems, vol.12, n°5, p.303-308.

Weiser, M. , 1991, « The computer for the 21st century », Scientific American, p.94–104.

Wu, Z., & Palmer, M. 1994, June. « Verbs semantics and lexical selection ». In Proceedings of the 32nd annual meeting on Association for Computational Linguistics , p. 133-138.

Yau, S.S., & Huang D. 2006. « Mobile middleware for situation-aware service discovery and coordination ». In P. Bellavista and A. Corradi (eds.), Handbook of Mobile Middleware, , p. 1059-1088.

Yavas, G., Katsaros, D., Ulusoy, . and Manolopoulos, Y. 2005. « A data mining approach for location prediction in mobile environments », Data Knowl. Eng., vol.54, n°2, p. 121–146.

Yazid Benazzouz, 2011, « Découverte de contexte pour une adaptation automatique de services en intelligence ambiante », Doctoral dissertation, École Nationale Supérieure des Mines de Saint-Étienne, France, 233 p.

Ying, J. J.-C., Lee, W.-C., Weng, T.-C. and Tseng, V. S. 2011. « Semantic trajectory mining for location prediction », in I. F. Cruz, D. Agrawal, C. S. Jensen, E. Ofek and E. Tanin (Eds), GIS, ACM, p. 34–43.

Yong Bin Kang, Yusuf Pisan. 2006. « A survey of major challenges and future directions for next generation pervasive computing ». Springer-Verlag Berlin heiderberg, , p. 755-764.

Yoon, T. B. and Lee, J.-H. 2008. « Goal and path prediction based on user's moving path data », in W. Kim and H.-J. Choi (Eds), ICUIMC, ACM, p. 475–480.

Yuan, B. and Herbert, J. 2014. « Context-aware hybrid reasoning framework for pervasive healthcare », Personal and Ubiquitous Computing, vol.18, n°4, p. 865–881.

Zaguia, A., Tadj, C. and Ramdane-Cherif, A. 2015. « Context-based method using bayesian network in multimodal fission system », Int. J. Computational Intelligence Systems, vol.8, n°6, p. 1076–1090.

Zang, M. A., Gray, A., Hobbs, M., & Pohl, J. G. 2008. « Similarity Assessment Techniques ». In Proceedings of InterSymp-2008: The 20th International Conference on Systems Research, Informatics and Cybernetics: Baden-Baden, Germany.

Zhang, D., Huang, H., Lai, C. F., Liang, X., Zou, Q., & Guo, M. 2013. « Survey on context-awareness in ubiquitous media ». Multimedia tools and applications, vol.67, n°1, p.179-211.

Zhang, F., Liu, W., & Bi, Y. « Review on Wordnet-based ontology construction in China », International Journal on Smart Sensing and Intelligent Systems, vol. 6, No. 2, April 2013.

Zhong, J., Zhu, H., Li, J., & Yu, Y. 2002. « Conceptual graph matching for semantic search ». In Proceedings of the 10th International Conference on Conceptual Structures (ICCS) , p. 92-196.

Zhu, C., & Sheng, W. 2011. « Motion-and location-based online human daily activity recognition ». Pervasive and Mobile Computing, vol.7, n°2, p. 256-269.

Zimmermann, A. 2003. « Context-awareness in user modelling: Requirements analysis for a case-based reasoning application ». In Case-Based Reasoning Research and Development , p. 718-732.