

TABLE DES MATIÈRES

	Page
INTRODUCTION	5
CHAPTIRE 1 REVUE DE LA LITTÉRATURE	21
1.1 Introduction.....	21
1.2 Informatique diffuse.....	22
1.2.1 Présentation de l'informatique diffuse.....	22
1.2.2 Enjeux et défis.....	27
1.2.3 Avantage et inconvénient de l'informatique diffuse.....	29
1.3 Notion du contexte.....	30
1.3.1 Que disent les dictionnaires	31
1.3.2 Définition du context dans un système informatique	32
1.3.3 Synthèse et critiques	36
1.3.4 Utilisation du contexte	40
1.3.5 Catégorisation du contexte.....	42
1.4 Notion de la sensibilité au contexte	44
1.5 Modélisation du contexte.....	46
1.5.1 Attributs-Valeurs.....	47
1.5.2 Modèle de représentation par balises	48
1.5.3 Modèle graphique	50
1.5.4 Modèle orienté objet.....	51
1.5.5 Modèle logique	53
1.5.6 Les ontologies	54
1.6 Adaptation selon le contexte.....	57
1.6.1 Notion d'adaptation	57
1.6.2 Mécanismes d'adaptation de services.....	58
1.6.3 Les défis.....	61
1.7 La prédiction du contexte.....	61
1.7.1 La notion de prédiction	61
1.8 Conclusion	62
CHAPTIRE 2 A SURVEY OF PREDICTION APPROACH IN PERVASIVE COMPUTING	63
2.1 Introduction.....	64
2.2 Proactive Computing	65
2.2.1 Proactive Context-Aware Applications	65
2.2.2 Proactive Adaptation.....	66
2.2.3 Proactive System Challenge	67
2.3 Context Prediction	68
2.3.1 Context Prediction Methods	68
2.4 Comparative Analysis and Evaluation.....	68

2.4.1	Reasoning Evaluation Criteria	87
2.4.2	Data Evaluation Criteria	89
2.5	Challenges of Context Prediction	91
2.6	Conclusion	92
CHAPTIRE 3	A SPATIOTEMPORAL CONTEXT DEFINITION FOR SERVICE ADAPTATION PREDICTION IN A PERVASIVE COMPUTING ENVIRONMENT	93
3.1	Introduction	94
3.2	Related Work	96
3.2.1	First Trend : Physical Aspect	96
3.2.2	Second Trend: Towards Time Aspect	97
3.2.3	Third Trend: Adaptation-Driven Context Definition	97
3.2.4	Towards Adaptation and Prediction Trends	99
3.2.5	Synthesis	100
3.3	Proposed Definition	101
3.3.1	Link context and time	101
3.3.2	Link context and space	103
3.3.3	Relevance of context according to the prupose of use	104
3.3.4	Proposed context definition	105
3.3.5	Demonstratio scenario	107
3.4	Conclusion & Future Work	115
CHAPITRE 4	SPATIOTEMPORAL CONTEXT MODELLING IN PERVASIVE CONTEXT-AWARE COMPUTING ENVIRONMENT: A LOGIC PERSPECTIVE	117
4.1	Introduction	118
4.2	Related Work	120
4.2.1	Overviews of context modelling approaches	120
4.2.2	Synthesis	123
4.2.3	Related work on contexte modelling approaches logic based	123
4.2.4	Synthesis	124
4.3	Proposed Modelling Approach	127
4.3.1	Context formalism	127
4.3.2	Context interpretation	132
4.3.3	Scenario morning at work	135
4.4	Conclusion & Future Work	141
CHAPITRE 5	USING PROBABILISTIC TEMPORAL LOGIC PCTL AND MODEL CHECKING FOR CONTEXT PREDICTION	143
5.1	Introduction	144
5.2	Related work	144
5.2.1	Synthesis	152
5.3	Our approach	144
5.3.1	Temporal logic and context aware system	154
5.3.2	Probabilistic temporal logic	155

5.3.3	PCTL semantic.....	160
5.3.4	Labelled probabilistic transition model : Model-LPTM	161
5.3.5	Transition Probability	164
5.3.6	Action.....	166
5.3.7	Span time duration	168
5.3.6	Immediate contexte prediction processing.....	170
5.3.7	Use case and test.....	173
5.4	Conclusion and futur work	144
CONCLUSION		181

LISTE DES TABLEAUX

	Page
Tableau 1.1	Table décrivant des documents48
Tableau 2.1	Comparative overview of prediction approaches 90
Tableau 3.1	Contextual information scenario (1)..... 109
Tableau 3.2	Contextual information scenario (2) 113
Tableau 4.1	Comparison of context modelling approaches..... 122
Tableau 4.2	Conjunction, disjunction and negation operator 130
Tableau 4.3	Quantification operator 131
Tableau 4.4	Devices, states and services spaces..... 137
Tableau 4.5	Office context situation..... 138
Tableau 4.6	Meeting room context situation 139
Tableau 5.1	Comparative overview of context prediction research work 152
Tableau 5.2	Paths quantifiers..... 159
Tableau 5.3	The four aspectual categories..... 166

LISTE DES FIGURES

		Page
Figure 1.1	Évolution de l'utilisation des ordinateurs par personne.....	22
Figure 1.2	Ubiquité des équipements informatiques.....	24
Figure 1.3	Évolution vers l'informatique diffuse.....	25
Figure 1.4	Vue d'ensemble d'un système diffus.....	27
Figure 1.5	Composants d'un système diffus.....	36
Figure 1.6	Le modèle UML de l'approche Hydrogen.....	52
Figure 2.1	Sequence prediction approach.....	69
Figure 2.2	Architecture for proactivity via predicted user contexts.....	71
Figure 2.3	General architecture of an HMM.....	73
Figure 2.4	MDP control cycle.....	74
Figure 2.5	Example of Simple Bayesian Network.....	76
Figure 2.6	Example of Dynamic Bayesian Network.....	76
Figure 2.7	Simplified structure of an artificial neural network.....	77
Figure 2.8	Expert system structure.....	82
Figure 2.9	Context spaces theory.....	83
Figure 2.10	General structure of context prediction.....	87
Figure 2.11	Comparative accuracy diagram.....	88
Figure 3.1	Comparative graphical of context definitions.....	100
Figure 3.2	Conceptual diagram in the context of space-time.....	102
Figure 3.3	Example smart Luminosity control in an office according to user presence and time.....	102
Figure 3.4	Position of a context according to the 3 axes (time-space-purpose).....	105

Figure 3.5	Context awareness operations.....	106
Figure 3.6	State diagram	108
Figure 3.7	Model of luminosity adjustment service.....	110
Figure 3.8	User's overall environment.....	111
Figure 3.9	Model of heating adjustment service	114
Figure 4.1	Context awareness operations with different layers of abstraction	132
Figure 4.2	USS services spaces.....	135
Figure 5.1	Expressivity CTL vs LTL vs CTL* vs PCTL.....	155
Figure 5.2	Overview of our approach	157
Figure 5.3	Transition model.....	163
Figure 5.4	State transition probability.....	164
Figure 5.5	State transition	165
Figure 5.6	Action transition.....	168
Figure 5.7	Transition span time	169
Figure 5.8	Component diagram of LPTM system.....	173
Figure 5.9	General view of prediction model.....	174
Figure 5.10	Activities transitions information's.....	175
Figure 5.11	Action-time generators.....	176
Figure 5.12	Verifying results Next activity.....	177
Figure 5.13	The activities prediction accuracy	178

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

API	Interface de programmation d'applications
BdC	Base de connaissances
BN	Bayesian network
CC/PP	Composite Capabilities / Preferences Profiles
CCML	Centaurus Capability Markup Language
CMM	Context meta-model
CML	Context Modeling Language
CoBrA	Context Broker Architecture
CONON	CONtext ONtology
ConteXtML	Context Markup Language
CoOL	Context Ontology Language
CSCP	Comprehensive Structured Context Profiles
CSL	Continuous stochastic language
CTL	Computation Tree Logic
CTL*	Computation Tree Logic extended
DAG	Directed Acyclic Graph
DBN	Dynamic Bayesian Network
GPS	Global Position System
GPRS	General Packet Radio Service / service de transmission de données en mode paquet
GPM	General profile Model
HMM	Hidden Markov Model
KDD	Knowledge Discovery in Databases
LMS	Local Models Semantics
LTL	Linear Temporal Logic
LPTM	Labelled Probabilistic Transaction Model
MBS	Multi Background Systems

MCS	Multi Contexte Systèmes
MDP	Markov Decision Process
MM	Markov Model
MVC	Model-View-Controller
M2M	Machine to machine
OWL	Ontology Web Language
ORM	Object Role Modeling
PDA	Personal Digital Assistant
PC	Personal computer
PCS	Pervasive Computing System
PCTL	Probabilistic Computation Tree Logic
PPDL	Pervasive Profile Description Language
RDF	Resource Description Framework
SGML	Standard Generic Markup Language
SID	Système informatique diffus
SMC	Semi-Markov Chain
SOUPA	Standard Ontology for Ubiquitous and Pervasive Applications
TIC	Technologies de l'information et de la communication
TL	Temporal Logic
UAProf	User Agent Profile
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
URL	Universal Resource Locator
USS	User Service System
UV	Ultraviolet
WIFI	Wireless Fidelity, wireless internet
W3C	World Wide Web Consortium
XML	eXtended Markup Language

INTRODUCTION

1. Cadre générale

L'ubiquité ou l'omniprésence, désignant la capacité d'être présent partout ou en plusieurs lieux en même temps, sont des attributs de puissance qui ont jalonné l'imaginaire collectif humain depuis la nuit des temps. Aujourd'hui, ils représentent un concept technologique bien réel permettant une ubiquité numérique. Les pionniers de l'informatique ambiante imaginaient la possibilité d'une diffusion omniprésente, invisible, partagée où une gamme d'appareils, interconnectés discrètement à travers un réseau hétérogène, est intégrée au milieu ambiant. Ce modèle représente la vision de Mark Weiser pour qui « les technologies les plus profondes sont celles devenues invisibles. Celles qui, nouées ensemble, forment le tissu de notre vie quotidienne au point d'en devenir indissociables ».

L'évolution technologique des systèmes embarqués, des moyens de communication et la miniaturisation des diapositives physiques ont favorisé l'intégration des terminaux mobiles et divers dispositifs communicants dans les applications informatiques donnant naissance à des systèmes d'informations diffus. L'informatique diffuse, telle qu'introduite par Mark Weiser, décrit l'ordinateur du XXI^e siècle comme un compagnon actif plus intelligent qu'un assistant de bureau, et qui dépasse la forme d'une infrastructure physique omniprésente pour une mise en place d'une intelligence ambiante assistant d'une manière active et intelligente ses usagers. En leur apportant aide et commodité dans l'accomplissement des activités quotidiennes, et ce dans différents domaines tels que : la communication, la santé, l'assistance à domicile, etc.

Combinant ces aspects distribués, mobilité et intelligence, l'informatique diffuse présente progressivement une nouvelle vision où la technologie devient invisible et permet une adaptation proactive aux changements du contexte des utilisateurs et des applications. Malgré leur évolution croissante, ces systèmes diffus soulèvent encore beaucoup de défis pour

assurer cette proactivité sensible au contexte et adéquate au besoin de l'utilisateur. Parmi leurs défis, ces systèmes doivent bien cerner le contexte de leur environnement de fonctionnement, en l'occurrence leur espace service et aussi pouvoir adopter un raisonnement complexe multidimensionnel respectant la nature complexe, dynamique et imprévisible de leur contexte et la nature stochastique et imprévisible du sujet humain.

Critiquant les services offerts par les systèmes ubiquitaires Gille Willet affirme que ces systèmes fabriquent un espace-temps artificiel perturbant l'espace-temps biologique de l'humain et de la réalité. Ce qui engendre des interactions de plus en plus rythmées par la technologie et non plus l'action humaine ou son contexte réel. Ces systèmes informatiques diffus ont développé l'illusion de l'ubiquité et du partage de la même réalité de leurs usagers qui n'est pas, réellement, du même principe significatif. Montrant ainsi les limites des systèmes existants à cerner le contexte réel et offrir des services selon les besoins utilisateurs.

Les critiques de Gille Willet ont bâti, d'une manière générale, les fondements de notre réflexion et tracé les défis de cette thèse. Elles seront de facto démontrées dans la littérature de notre domaine. Déterminant ainsi notre cadre générale de recherche. En effet, celui-ci discute la capacité des systèmes informatiques ubiquitaires à cerner le contexte, à raisonner sensiblement selon le contexte dans un cadre multidimensionnel pour assurer une adaptation proactive dans un espace qui incarne une complexité spatiotemporelle et comportementale dynamique et imprévisible.

Traitant principalement le contexte de ces systèmes omniprésents pour assurer la sensibilité au contexte; ce projet de thèse discute la notion du contexte. Plus spécifiquement, il porte sur le formalisme du contexte et la prédiction formelle du contexte futur dans les systèmes ambiants, et ce dans le cadre d'une approche générale et logique respectant un aspect complexe écoulant de la nature plurielle et dynamique de la réalité de ces systèmes et de leurs environnements ; l'espace, le temps et l'humain.

2. Motivation

Les systèmes omniprésents sensibles à leur contexte tentent d'offrir des services invisibles et adéquats au besoin de l'utilisateur afin d'améliorer sa qualité de vie, gérer son environnement intelligemment et lui procurer le confort et l'assistance selon ses besoins. Pour y arriver, ses systèmes doivent alléger leurs bases de raisonnement respectant l'aspect dynamique et stochastique de la nature humaine de ses usagers. Pour atteindre une adaptation proactive sensible au contexte, ces systèmes doivent être capables de comprendre leur contexte et être sensibles aux changements dynamiques et imprévisibles de ce dernier. Mettre en œuvre cette adaptation est loin d'être une tâche facile. Plusieurs questions de recherche doivent être résolues avant de déboucher vers des systèmes qui assurent une adaptation proactive sensible au contexte. Commenant par la définition du contexte, le formalisme et la modélisation du contexte et les méthodes de raisonnement sur le contexte.

L'adaptation et l'accès aux services dans les systèmes ubiquitaires sensibles au contexte sont directement liés ou même déduits à partir de l'information contextuelle. C'est pour cela que le contexte est une notion fondamentale dans l'informatique omniprésente et la base pour assurer une adaptation proactive et efficace. La majorité des travaux sur l'adaptation de services selon le contexte se sont concentrés sur le contexte actuel (présent). Par contre, une nouvelle tendance dans le domaine de recherche sur les approches d'adaptation de services appuie l'idée de l'importance de prendre en considération l'évolution du contexte pour prévoir un contexte futur. Être conscient du contexte futur permet à un système ubiquitaire de choisir les stratégies les plus efficaces pour atteindre ses objectifs et permet donc de fournir une adaptation active et rapide aux situations à venir. À partir de là découle notre intérêt au problème de prédiction du contexte. Cependant, les approches qui offrent des prédictions de contexte générales ainsi qu'une expressivité formelle sont manquantes dans la littérature. En conséquence, le plein potentiel des technologies sensibles au contexte, offrant des services proactifs et anticipatoires, n'a pas encore été complètement réalisé.

3. Problématique de recherche

La prédiction de contexte est un problème relativement nouveau dans le domaine de l'informatique omniprésente. Les travaux portants sur la prédiction du contexte sont récemment en cours d'élaboration. Ils soulèvent encore de nombreux défis qui restent à résoudre. Dans notre projet on s'intéresse principalement à la notion du contexte et la prédiction du contexte futur dans un système ubiquitaire.

Cerner la notion de contexte reste encore un défi dans le domaine des systèmes sensibles au contexte surtout si l'on parle des systèmes de nature proactive et anticipatoire. Avec l'évolution de la technologie, les techniques de captation et de traitement de l'information, les techniques de raisonnement et d'apprentissage, cette notion de contexte a dû être revisitée. Malgré le développement des autres notions en lien, comme : l'adaptation, l'adaptation proactive, l'anticipation et la sensibilité au contexte, la notion du contexte elle-même est restée vague et non actualisée. Généralement traitée dans un raisonnement unidimensionnel, physique et d'une nature systématique, la notion de contexte s'est donc figée dans le vague ou l'élémentaire cerné selon le cas d'utilisation précis. Du fait que le contexte représente généralement la description d'un espace de service qui est enfin l'espace de la vie des usagers et que réellement cet espace est complexe et ne peut pas être traité loin de la nature dynamique complexe et imprévisible de la réalité. Partant de ce constat, se dessine notre premier défi : il s'agit de définir le contexte ou plutôt de le redéfinir essayant d'offrir une définition à jour adéquate aux nouvelles tendances du domaine des systèmes ubiquitaires proactifs sensibles au contexte.

Un autre défi principal est le manque d'approches formelles au problème de la prédiction de contexte général. La plupart des solutions actuelles prédisent le contexte dans des situations particulières et dans une approche unidimensionnelle. En conséquence, leur expressivité et leur efficacité d'adaptation proactive ou anticipatoire restent limitées. Puisqu'ils ne reflètent pas la réalité de l'environnement des systèmes ambiants; étant donné que l'informatique ubiquitaire incorpore nécessairement un projet humain intégrant son ambiant quotidien. Ces

systèmes fonctionnent dans un espace d'interaction directe avec ses usagers qui est naturellement multidimensionnels spatiotemporel et comportemental. Le changement de contexte doit donc nécessairement être considéré en respectant un dynamisme spatiotemporel étendu. Offrant la possibilité de reconnaître le temps prévu de ces changements de contexte et le comportement sous-jacent du système et de ses usagers.

Pour répondre à cette problématique, nous proposons donc une approche formelle de prédiction de contexte général basée sur la logique temporelle. Commençant par redéfinir le contexte dans un aspect dynamique et avec une vision adéquate à la nouvelle nature proactive des systèmes ubiquitaires. Et le modéliser sous un formalisme logique intégrant l'aspect spatiotemporel.

Pour fixer nos objectifs plus tard, nous allons commencer par énumérer quelques questions qui doivent être abordées à fin de développer notre modèle de prédiction :

- comment peut-on cerner le contexte dans un cadre complexe multidimensionnel?
- Quels sont les éléments de contexte nécessaires dans une adaptation proactive et/ou anticipatoire?
- Quel est le contexte à définir pour une stratégie de prédiction?
- Comment faire usage d'un contexte?
- Comment peut-on modéliser d'une manière formelle le contexte?
- Comment peut-on intégrer l'attribut temps dans une expressivité formelle d'un modèle contextuel logique?
- Comment peut-on formaliser une prédiction assurant le respect de la nature complexe, stochastique et probabiliste de l'environnant et des usagers des systèmes informatiques ubiquitaires ?
- Comment peut-on intégrer la contrainte temporelle dans notre modèle prédictif?
- Comment peut-on générer une prédiction qui peut supporter plusieurs types de requêtes dans un cadre formel?
- Comment peut-on offrir une prédiction d'information contextuelle générale, multidimensionnelle?

- Comment le modèle gère-t-il l'aspect stochastique et probabiliste de la prédiction?

4. Objectifs de la recherche

Le cadre général de nos recherches est le domaine de l'informatique ubiquitaire. Nous nous intéressons particulièrement à la sensibilité au contexte et la prédiction du contexte future. Notre intérêt porte sur la proposition d'une nouvelle approche de prédiction du contexte général sous un formalisme temporel logique pour appuyer le développement de systèmes sensibles proactifs et sensibles à leur contexte. Nos objectifs seront de :

- proposer une spécification claire des informations contextuelles tenant compte des caractéristiques spatiotemporelles;
- proposer une modélisation formelle de l'information contextuelle intégrant l'aspect spatiotemporel;
- proposer une approche pour la mise en place d'un processus de prédiction du contexte basé sur la logique temporelle;
- proposer une approche de prédiction du contexte général;
- proposer une approche de prédiction qui sera capable de supporter une contrainte temporelle;
- proposer un cadre formel de prédiction du contexte général.

Nous présentons, dans la prochaine section, notre démarche pour résoudre les problématiques principales de notre recherche, détaillant la méthodologie suivie pour atteindre nos objectifs cités ci-dessus.

5. Méthodologie

Notre démarche globale est représentée selon les phases suivantes :

5.1 La phase 1 : exploration

Cette phase d'exploration a contribué à cerner notre problématique, à identifier les besoins dans notre domaine de recherche et à fixer nos objectifs de recherche. Nous avons pu ainsi

situer nos travaux par rapport à l'état actuel du développement des systèmes diffus particulièrement les systèmes sensibles au contexte. Nous avons pu aussi analyser les limites et les défis qui restent à surmonter (Chapitre.1).

Cette étape d'études et d'analyses a pu mettre le doigt sur les défis qui demeurent à jour dans les systèmes omniprésents et qui bloquent le passage vers des systèmes omniprésents proactifs, anticipatoires, sensibles à leur contexte qui peuvent offrir des services proactifs efficaces et adéquats avec les besoins de leurs utilisateurs.

Traitant en profondeur les concepts de l'adaptation proactive et la sensibilité au contexte en se centrant sur la notion et les techniques de la prédiction du contexte futur pour assurer ce passage vers l'anticipatoire et le proactif sensible au contexte. Ce qui a fait le sujet d'une étude analytique soulignant les défis du domaine des systèmes proactifs et étudiant les différentes techniques de prédiction et leurs limites résultant une étude comparative de ces méthodes. Contribuant ainsi à souligner les principaux challenges à surmonter dans notre domaine pour assurer une proactivité sensible à son contexte et au besoin utilisateur. (Chapitre .2).

5.2 La phase 2 : Spécification et Modélisation

Spécification et modélisation du contexte

La première étape vers le développement d'un système diffus sensible au contexte est de comprendre le contexte et établir ses composants. Cela exige une description claire de ce qu'est un contexte. Nous allons de fait commencer par définir et établir les principaux éléments du contexte dans un système informatique diffus et proposer une définition respectant les aspects multidimensionnels et spatiotemporels.

Nous proposons par la suite une approche de modélisation formelle des informations contextuelles qui favorise l'aspect dynamique des systèmes informatiques diffus et supporte le raisonnement logique qui servira plus tard pour la prédiction. Et ce, à travers la

spécification et la modélisation d'un nouveau modèle logique, clair et intégrant l'aspect spatiotemporel afin de demeurer fidèle à notre réflexion globale sur le contexte qui définit ce dernier autour d'un dynamisme et une évolution spatiotemporelle étendue.

Spécification et modélisation de la prédiction du contexte futur

Nous fournirons une approche de prédiction formelle basée sur un raisonnement logique respectant la nature dynamique et non-déterministe de l'environnement diffus. On propose une approche qui:

- supporte un formalisme logique plus expressive de ce qui existe dans la littérature qui est temporelle et probabiliste à savoir le PCTL *Probabilistic Computation Tree Logic*;
- supporte la prédiction d'un contexte général;
- supporte les contraintes temporelles;
- supporte un raisonnement logique temporel, stochastique et non déterministe respectant la nature dynamique et imprévisible de l'environnement diffus proactif.

5.3 La phase 3 : Validation

Toute au long de nos étapes de spécifications et de modélisations nous démontrons nos raisonnements et nous validerons nos approches et nos modèles, à savoir le modèle logique des données contextuelles et le modèle prédictif du contexte, via des scénarios décrivant des cas d'utilisation dans des systèmes omniprésents proactifs, pour faire preuve de la validé de nos approches et leurs apports expressif et formel. Et ce en les incluant dans les sections validation, résultats et test dans les chapitres à suivre pour chacune des étapes évoquées durant cette sous-section; section 3.3.5 du chapitre 3, section 4.33 du chapitre 4 et section 5.3.9 du chapitre 5.

Pour notre modèle prédictif, nous nous baserons sur la méthode symbolique qui est la méthode la plus pertinente. Parcourir toute la structure de Kripke peut être coûteux, de fait les méthodes symboliques, introduites par Ken McMillan et Ed Clarke seront plus

pertinentes. Cette méthode symbolique une approche répandue pour la vérification de propriétés exprimées dans la logique temporelle arborescente. Elle est fondée sur la représentation symbolique du modèle de trace de fonctionnement du système à vérifier. Nous vérifierons et nous validerons notre raisonnement et notre modèle via des preuves et des exemples explicatifs ainsi qu'un cas d'étude présentant des traces des informations contextuelles dans un système omniprésent.

6. Contribution

Dans un système omniprésent, nous devons fournir un service d'adaptation, en fonction de l'environnement du système et le profil de l'utilisateur afin d'offrir une adaptation de service adéquate. La plupart des travaux sur le service d'adaptation sensible au contexte se concentrent sur le contexte actuel. Cependant, tenir compte de l'évolution temporelle et logique du contexte afin de prédire le contexte futur permet au système de choisir les stratégies d'adaptation de service les plus efficaces pour atteindre ses objectifs, et ainsi fournir une adaptation active et rapide à des situations futures et concevoir des systèmes dynamiques et anticipatoires.

Or, le déploiement efficace d'une prédiction sensible au contexte, à son évolution dynamique et imprévisible, est encore limité en raison d'un fossé sémantique entre les données fournies par les dispositifs de détection physiques et les informations nécessaires pour prédire le comportement futur du système et de ses utilisateurs.

À cet effet notre thèse est axée sur la définition et la modélisation contexte, ainsi que la prédiction du contexte futur; leurs expressivité, spécification et modélisation formelle et à base d'une approche logique temporelle. Cette thèse propose une approche formelle pour la prédiction du contexte afin d'améliorer le comportement proactif des applications ambiantes sensibles au contexte. L'approche proposée surpasse les faiblesses qu'on a repérées dans la littérature en offrant :

1. Une meilleure expressivité du contexte dans une approche multidimensionnelle intégrant une vision de stratégie d'utilisation des informations contextuelles à court ou à long terme dans l'adaptation proactive.
2. Une spécification multidimensionnelle du contexte tenant compte de son dynamisme spatiotemporel, et ce via l'évolution de l'information dans son cadre spatial et temporel.
3. Une modélisation logique du contexte intégrant l'aspect spatiotemporel.
4. Un modèle prédictif offrant avec une forte expressivité, grâce au formalisme logique du processus de prédiction.
5. Une approche générale:
 - pour une utilisation sur un large spectre de données contextuelles et de diverses applications;
 - pour faire face à la nature hétérogène des systèmes informatiques omniprésents (e.g : la nature et le type des données, la diversité des modalités, etc.) ;
 - pour une prédiction du contexte général à bas et/ou au haut niveau.
6. Un modèle prédictif offrant une forte capacité de raisonnement logique temporel et stochastique ainsi qu'une prédiction du contexte futur dans un espace multidimensionnel adéquat avec la nature des systèmes ambiants.
7. L'adoption d'un formalisme PCTL permet une meilleure expressivité pour décrire la nature non déterministe du comportement humain qui peut fournir une prédiction efficace et offrir par conséquent une proactivité sensible au changement du contexte et adéquate aux besoins de l'utilisateur.

Pour conclure, nos contributions consistent en :

a. Une étude analytique

Via notre étude, nous avons analysé l'évolution des systèmes proactifs et souligné les challenges à surmonter pour parvenir à mettre en place des systèmes proactifs sensibles au contexte et plus efficaces. Nous avons identifié les questions en suspens pour une prédiction de contexte plus efficace, et les nombreux défis existants dans le domaine via une analyse approfondie des techniques de prédiction existantes, et ce en dressant une liste de critères comparatifs. Cette étude a permis de mettre au clair les réels défis et de mettre en lumière les questions de recherche à résoudre pour une adaptation proactive et anticipatoire efficace (détaillée dans le chapitre 2 (article 1)).

b. Une nouvelle définition du contexte

Malgré l'évolution de la technologie, les techniques de captation et de traitement de l'information, les techniques de raisonnement et d'apprentissage, la notion de contexte n'a pas été revisitée. La notion du contexte est restés vague, non à jour, traitée dans un raisonnement unidimensionnel, physique et d'une nature systématique, de telles sortes qu'on se trouve devant un concept figé dans le vague ou l'élémentaire et cerné selon un cas d'utilisation précis.

Delà, se dessine notre premier défi; qui était de définir le contexte ou plutôt de le redéfinir d'autant plus qu'il y avait des aspects manquants dans les définitions présentes dans la littérature et qui ne sont plus à jour avec les nouvelles tendances du domaine des systèmes proactifs sensibles au contexte. Comprendre le contexte et établir ses composants étaient une étape de base vers le développement d'un modèle prédictif sensible au contexte. Cela a exigé une description claire de ce qu'est un contexte. On a donc proposé une nouvelle définition du contexte qui permet de mieux cerner l'information contextuelle, respectant un cadre multidimensionnel du contexte tenant en compte l'aspect spatiotemporel et la stratégie de l'utilisation dans le fonctionnement du système et l'adaptation des services (détaillée dans le chapitre 3 (article 2)).

c. Un nouveau modèle spatiotemporel logique du contexte

Une fois défini, le contexte doit être bien représenté et modélisé sous une forme appropriée qui surpasse l'aspect hétérogène des systèmes omniprésents et qui permet une utilisation plus efficace, fournissant un niveau d'abstraction plus élevé pour faciliter le raisonnement sur les données contextuelles. En nous intéressant à l'utilisation du contexte pour développer un raisonnement logique, nous avons présenté un modèle de contexte formel tenant compte du cadre spatiotemporel. Nous avons fourni aussi un modèle spatiotemporel dynamique, facilement compréhensible, qui soutient le langage naturel et le raisonnement logique, tout en offrant la possibilité de l'abstraction vers la déduction d'un contexte de haut niveau basé sur la théorie de la situation (détaillée dans le chapitre 4 (article 3)).

d. Une nouvelle approche formelle de prédiction du contexte futur basée sur un raisonnement logique, temporel, probabiliste et stochastique

Les approches actuelles de prédiction de contexte sont unidimensionnelles. En conséquence, leur expression et leur efficacité sont limitées. Nous proposons une nouvelle approche générale, logique, présentant la prédiction dans un aspect multidimensionnel respectant la nature dynamique et non déterministe de l'évolution du contexte dans un système ambiant. Nous avons donc fourni une approche surpassant les limites dans la littérature qui est capable de :

- supporter une prédiction générale du contexte, offrant une prédiction sur un large spectre de données contextuelles;
- supporter une contrainte temporelle;
- fournir une logique de prédiction expressive avec un formalisme sémantique clair;
- supporter un raisonnement logique temporel probabiliste (PCTL).

Ce modèle est donc une autre étape pour fournir une expressivité du contexte et un raisonnement logique basé sur le dynamisme temporel du contexte ambiant, et ce par un

formalisme logique temporel supportant l'aspect stochastique probabiliste de l'évolution de l'information contextuelle et le comportement utilisateur des systèmes ambiants diffus.

L'adoption du PCTL comme un formalisme fournit une meilleure expressivité pour décrire la nature stochastique de l'évolution étendue du contexte dans un système ambiant ce qui permet de fournir une prédiction efficace et résultante ainsi une proactivité en raccord avec les besoins de l'utilisateur (détaillé dans le chapitre 5 (article 4)).

7. Structure de document

L'organisation de ce manuscrit est la suivante :

Le premier chapitre fera l'objet d'une présentation de l'état de l'art qui aborde les notions et les concepts du domaine de l'informatique omniprésente et proactive à savoir le contexte, l'adaptation et la prédiction. Les chapitres qui suivent sont les travaux publiés et soumis.

Le deuxième chapitre, est un article publié dans le journal « The International Journal of Scientific & Engineering Research»

Darine Ameyed, Moeiz Miraoui, Chakib Tadj, «A Survey Of Prediction Approach In Pervasive Computing», Journal of Scientific & Engineering Research, Volume 6, Issue 4, April-2015 ISSN 2229-5518.

Dans cet article, nous avons élaboré une étude sur les systèmes omniprésents sensibles au contexte. Nous avons présenté et analysé le développement de ces systèmes mettant en évidence les problématiques à surmonter pour parvenir à mettre en place des systèmes proactifs et anticipatoires plus efficaces. Nous avons aussi survolé les techniques de prédiction existantes, examinant en profondeur les projets et les travaux de recherche liés à la prédiction du contexte. Nous avons établi un ensemble de critères de comparaison que nous avons utilisé pour la comparaison et l'analyse de ces approches. Nous avons présenté également plusieurs concepts qui sont nécessaires pour la prévision du contexte. Et nous

avons identifié les questions en suspens et les nombreux défis existants pour assurer une prédiction efficace dans les systèmes ambiants.

Le troisième chapitre est un article publié dans le journal «The International Journal of advanced studies in Computer Science and Engineering IJASCSE»

Darine Ameyed, Moeiz Miraoui, Chakib Tadj, «A Spatiotemporal Context Definition For Service Adaptation Prediction In A Pervasive Computing Environment» the International Journal of advanced studies in Computer Science and Engineering IJASCSE Volume 4, Issue 4, 2015.

Dans cet article, nous avons proposé une nouvelle définition qui permet de mieux cerner le contexte dans un cadre multidimensionnel en prenant en considération la stratégie d'utilisation. Présentant ainsi le contexte dans une spécification mettant en évidence la dynamique spatio-temporelle des éléments de contexte dans l'adaptation proactive sensible au contexte. Nous avons présenté une nouvelle définition du contexte en tenant compte : 1) des informations spatio-temporelles, 2) du but de l'utilisation, et 3) et toute autre information qui peut être utile et qui peut conduire à un changement actuel ou futur dans l'adaptation proactive.

Le quatrième chapitre est un article publié dans le journal « The International Journal of Advanced Computer Science and Applications »

Darine Ameyed, Moeiz Miraoui, Chakib Tadj, «Spatiotemporal Context Modelling In Pervasive Context-Aware Computing Environment: A Logic Perspective» the (IJACSA) International Journal of Advanced Computer Science and Applications Vol. 7, No. 4, Avril 2016.

Dans cet article, nous avons présenté un modèle de contexte formel tenant en compte le cadre spatiotemporel. Notre modèle suit notre réflexion sur l'information contextuelle spatiotemporelle, et fournit une méthode formelle pour intégrer ces aspects dans la modélisation du contexte. Comparée à l'autre modèle formel, notre proposition assure des

propriétés notables pour le modèle de contexte: contexte dynamique facilement compréhensible; soutien de langue naturelle, de soutien du raisonnement logique sur le contexte en demeurant fidèle au cadre spatiotemporel et la nature évolutive et dynamique du contexte dans un système omniprésent et ambiant.

Le cinquième chapitre est un article soumis dans le journal «International Journal of Intelligent Systems Technologies and Applications ».

«Using Probabilistic Temporal Logic Pctl And Model Checking For Context Prediction»

Darine Ameyed, Moeiz Miraoui, Fahmi Jaafer, Atef Zaguia, Chakib Tadj

Dans cet article, nous présentons une nouvelle approche formelle pour la prédiction de contexte dans les systèmes proactifs sensibles au contexte. Nous exprimons le contexte et les transitions entre les états dans un système diffus sous une présentation formelle, en utilisant la logique temporelle probabiliste PCTL (une extension probabiliste de la logique temporelle). Nous proposons un modèle de transition probabiliste pour coder et tracer le comportement du système dans le temps. La combinaison de PCTL avec le modèle stochastique nous permet de tracer, analyser et de prédire le contexte futur en utilisant le modèle checking pour vérifier les propriétés des états futurs retournant ainsi des résultats quantitatifs. Dans cet article, nous avons démontré comment les méthodes formelles pourraient être adoptées pour offrir un cadre formel et général de prédiction. Et ce pour réduire le fossé sémantique et fournir une meilleure expressivité via la logique. L'intégration des contraintes de temps et d'action ajoute de la puissance à l'expressivité logique et fournit un modèle de traçage et d'apprentissage plus clair et plus expressif et augmente ainsi l'efficacité des mesures probabilistes utilisées pour la prédiction du contexte futur. Cette expressivité formelle de prédiction est utile dans un système informatique omniprésent pour faire face à leur nature hétérogène, aussi pour sa forte capacité d'exprimer les aspects dynamiques temporels et découvrir le contexte futur dans un espace multidimensionnel. Ceci offre un modèle qui peut supporter une prédiction du contexte général de bas ou haut niveau respectant la nature non déterministe du comportement humain et ainsi fournir une prédiction

efficace et offrir par conséquent une proactivité adéquate en accord avec les besoins de l'utilisateur.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

1.1 Introduction

Les systèmes informatiques connaissent un considérable développement ces dernières années. Ce développement est marqué par l'évolution technologique des terminaux mobiles (utilisation des téléphones cellulaires intelligents, les assistants numériques personnels (PDA), etc.). L'évolution de la technologie mobile se manifeste aussi par le développement des réseaux de communications sans fil (Wi-Fi, Wimax, Bluetooth, etc.). L'ensemble de ces technologies changent complètement le mode d'utilisation de l'Informatique. Avec l'avènement de ces technologies mobiles, on parle de l'Informatique diffuse, omniprésente ou les systèmes ubiquitaires. L'Informatique ubiquitaire rend l'utilisation de l'Informatique possible en tout moment et indépendamment du positionnement de l'utilisateur. Cette nouvelle forme de l'Informatique ouvre de nouvelles opportunités aux utilisateurs d'avoir accès à des services partagés et omniprésents centrés sur leurs besoins.

Dans ce chapitre, nous commençons par présenter quelques concepts de base dans ce domaine et des notions que nous croisons tout au long de ce document. Nous nous attardons sur de la notion de contexte et le principe de la sensibilité au contexte (ou context-awareness) qui constituent les notions centrales de notre travail de recherche. Nous introduisons la notion de la prédiction du contexte vers la fin pour laisser place au chapitre suivant l'approfondissement des techniques de la prédiction et les défis principaux du domaine de l'informatique diffuse.

1.2 Informatique diffuse

1.2.1 Présentation de l'informatique diffuse

La notion de l'informatique diffuse a été présentée dans un article fondateur par Mark Weiser en 1991. Il décrivait à un horizon de vingt ans l'avenir de l'informatique. Suite à une analyse de marché mondial des ordinateurs et de l'équipement informatique, il remarquait la croissance du nombre des ordinateurs par utilisateur, ainsi que l'intégration des processeurs dans les objets de la vie quotidienne (Figure 1.1). Il nomma cette nouvelle forme d'informatique, informatique ambiante en anglais « *ubiquitous computing* ». L'objectif de ces systèmes est d'assister implicitement et discrètement l'utilisateur dans les tâches qu'il effectue quotidiennement, d'où la base des systèmes diffus.

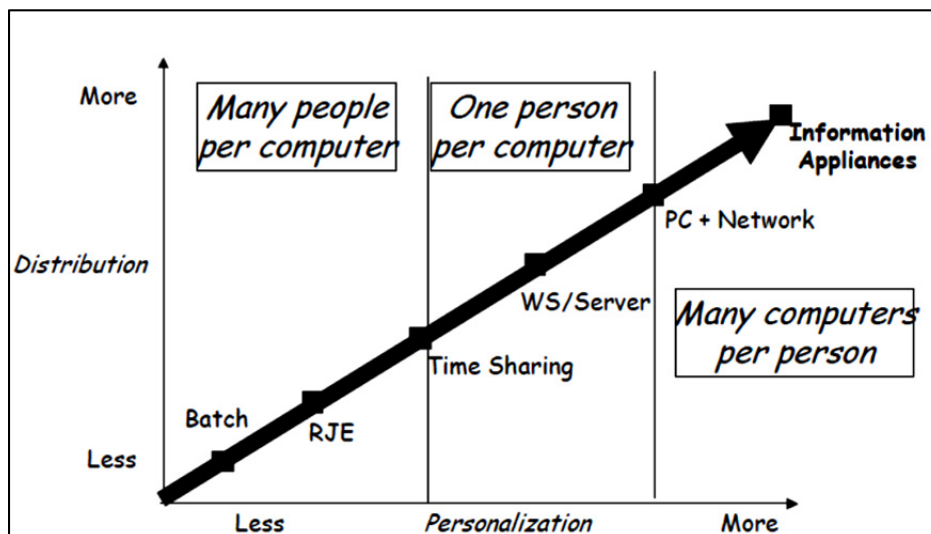


Figure 1.1 Évolution de l'utilisation des ordinateurs par personne

Tirée de Franklin (2001)

L'évolution technologique actuelle favorise la réalisation de la vision de Mark Weiser. Depuis quelques années, la miniaturisation des dispositifs électroniques aide à leur intégration dans des objets que nous utilisons quotidiennement dans notre environnement. Par exemple les téléphones mobiles sont quasiment tous équipés d'un appareil photo, d'une

connexion à divers réseaux tels que WiFi, GRPS, UMTS, un certain nombre d'assistants personnels (PDA) disposent d'une fonctionnalité GPS qui permet de se positionner et d'être guidé vocalement et visuellement vers un lieu précis.

Le développement d'outils de communication a aussi favorisé l'échange entre différents équipements informatiques pour coopérer d'une manière fluide et rapide. Cette tendance à l'informatisation systématique des ressources favorisant l'accès et l'échange des informations partout et à tout moment est qualifiée par l'informatique omniprésente (Krupitzer, Roth et al. 2015)

Ce paradigme est décrit dans la littérature par diverses expressions. On parle d'informatique ambiante, informatique ubiquitaire ou pervasive, faisant référence à l'utilisation d'équipement intégrant des processeurs de petite taille communiquant spontanément les uns avec les autres et de capteurs intégrés dans les objets de la vie quotidienne qui deviennent presque invisible pour les utilisateurs, tels que décrit par Mark Weiser « *the most profound technologie are those that disppear* ». L'informatique diffuse (pervasive) fait référence à l'accès à un service selon les besoins et des contraintes de l'utilisateur sans son intervention explicite et lui offre la possibilité de se concentrer sur sa tâche principale sans se soucier de la configuration et la gestion de l'ensemble des équipements informatiques mis à sa disposition. En d'autres termes, la configuration doit se faire d'une manière transparente et automatique (Figure 1.2).



Figure 1.2 Ubiquité des équipements informatiques

Adam Greenfield utilise le mot « everyware » en français informatique omniprésente. Un mot formé de everywhere et hard/software est un néologisme qui réunit les termes d'informatique ubiquitaire, informatique pervasive, informatique ambiante et média physique désignant les systèmes mobiles ou distribués.

Il explique « quand je parle de rendre visible l'information qui a jusqu'ici toujours existé à l'état latent dans nos vies, j'entends l'attribution de valeurs numériques précises à notre localisation, à la tâche que nous accomplissons en ce moment » (Abbar, Bouzeghoub et al. 2009).

L'informatique diffuse favorise ainsi la création des environnements intelligents et augmente les capacités de leurs systèmes informatiques à interagir avec le réel, tels que le concept de la maison intelligente capable de gérer automatiquement les différents équipements au domicile d'un utilisateur.

1.2.1.1 Évolution vers l'informatique diffuse

L'informatique diffuse est le résultat de la convergence du domaine de l'informatique mobile et des systèmes distribués. Un système informatique distribué est une collection de postes ou calculateurs autonomes qui sont connectés à l'aide d'un réseau de communication. Ils permettent le partage des capacités et des ressources à travers un réseau et une infrastructure de communication. Avec l'introduction de l'omniprésence de l'information, cette convergence constitue la première étape de l'informatique diffuse. Une deuxième étape vers l'informatique diffuse sera franchie suite à l'intégration de l'informatique cellulaire et du web qui donnera forme à l'informatique mobile. Et c'est grâce à des équipements de petite taille et de faibles coûts que l'informatique mobile a permis aux utilisateurs d'accéder à l'information à n'importe quel endroit « n'importe où, n'importe quand » (Satyanarayanan 2001).

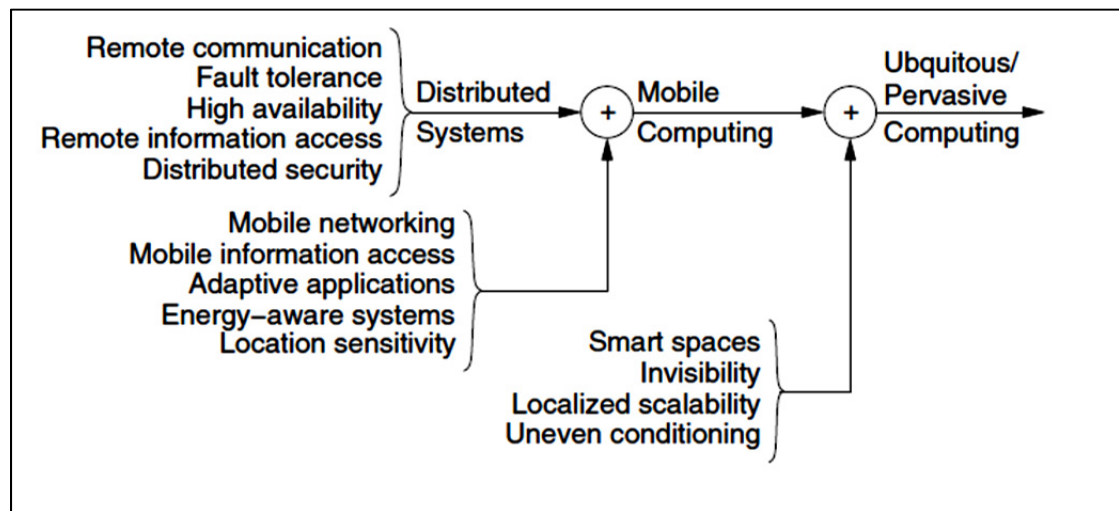


Figure 1.3 Évolution vers l'informatique diffuse.
Tirée de Satyanarayanan (2001)

L'informatique diffuse se base sur la connaissance du contexte de l'utilisateur pour lui fournir un service adapté à son contexte.

1.2.1.2 Exemple de système informatique diffuse

Le système de badge actif (activebadge) (Want, Hopper et al. 1992) est l'une des premières applications de l'informatique diffuse. L'application était conçue pour permettre l'accès par identifiant aux bureaux du centre de recherche Xeros et à localiser un usager à fin de lui transmettre ses appels téléphoniques selon le bureau où il se trouve.

Les systèmes de la localisation géographique se sont multipliés par la suite. Ils peuvent servir par exemple aux services de guidage routier par GPS (Global Position System/ système de guidage par satellites). Plus tard, Dey et Abowd ont proposé une définition plus globale du contexte. Ils l'ont défini comme « l'ensemble des informations qui peuvent être utilisées pour caractériser la situation courante d'une entité, tel qu'un individu, un lieu ou un objet considéré comme ayant un rapport avec le service délivré par le système » (Dey and Abowd 2000).

En se basant sur cette dernière définition du contexte, Sayeh et al ont créé « My compus » un système qui exploite différentes données issues de l'environnement pour définir le contexte courant d'un usager. Cette application servie sur le campus de Carnegie Mellon assiste les usagers dans diverses tâches, fonctionnant à base d'agents : par exemple un agent intelligent indique dans quel restaurant l'étudiant peut prendre ses repas en fonction de ses préférences gastronomiques, de l'heure, de sa localisation, et des conditions météorologiques. L'agent *réunion* qui présente un autre aspect de l'informatique ambiante utilisant l'échange direct de machine à machine pour parcourir les agendas des utilisateurs et les aide à choisir une date commune de réunion. Ces applications ambiantes se basent sur l'assistance discrète et permanente des usagers dans leurs tâches courantes.

Pour éviter la confusion en terminologie, nous présentons les définitions des termes qui caractérisent le domaine de l'informatique diffuse (Miraoui 2009); (Le Mouël and Laforest 2006) :

- **Ubiquitaire** : accessible de n'importe où;
- **Mobile** : Qui intègre les terminaux mobiles;
- **Sensible au contexte** : qui prend en compte le contexte d'exécution;

- **diffus** : qui associe ubiquité, mobilité et sensibilité au contexte;
- **Ambiante** : qui est intégré dans un objet quotidien.

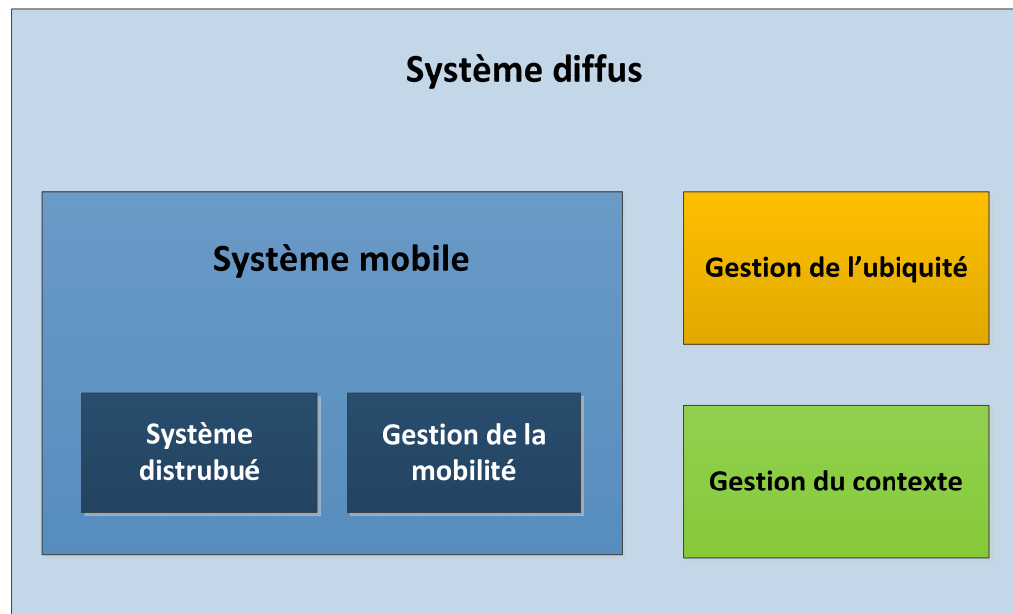


Figure 1.4 vue d'ensemble d'un système diffus

1.2.2 Enjeux et défis

Tel que présenté dans la figure 1.4, un SID (système d'informatique diffuse) englobe les systèmes distribués, l'informatique mobile, l'ubiquité et la sensibilité au contexte. Un SID doit donc assurer plus que la mobilité. Il doit assurer certaines fonctionnalités, celles d'un système distribué à savoir (Perera, Zaslavsky et al. 2014); (Pejovic and Musolesi 2015) :

- persistance des données;
- échange des données entre applications hétérogènes;
- gestion de la cohérence permanente des données;
- portabilité des applications;
- gestion des accès concurrents;

- ouverture et sécurité : avoir des architectures d'adaptation ouvertes et flexibles;
- sensibilité au contexte;
- proactivité;
- anticipation de service.

Certaines propriétés communes sont requises telles que:

- **L'évolutivité** : l'environnement informatique omniprésent est un environnement évolutif et dynamique soit en nombre d'utilisateurs et/ou en équipement. Ce point est en lui-même un défi à surmonter pour chaque SID en sa capacité d'adaptation et de l'intégration de nouveaux équipements et de nouveaux utilisateurs;
- **L'hétérogénéité** : un SID doit réussir à masquer l'hétérogénéité des équipements ou des modes de communication utilisés;
- **L'intégration** : bien que les composants informatiques omniprésents soient déjà déployés dans de nombreux environnements, l'intégration en une seule plateforme est encore un problème de recherche. L'intégration influence la qualité de service, l'invisibilité et la sécurité du système;
- **L'invisibilité** : un SID doit offrir une technologie invisible et minimise l'intervention humaine. Il doit être capable d'adapter son environnement et de répondre aux attentes des utilisateurs automatiquement;
- **la perception** (contexte-awareness) : la plupart des systèmes informatiques actuels ne disposent pas d'une sensibilité aux contextes. Cependant un système diffus doit être réactif au contexte répondant discrètement à un événement. La perception ou la sensibilité au contexte est une valeur caractéristique des systèmes diffus;

- **l'intelligence** : (contexte management) : une fois que le système diffus perçoit un contexte actuel, il doit avoir les moyens d'utiliser ces informations efficacement pour assurer une interaction automatique et adéquate;
- **proaction** : un système diffus doit être en mesure de suggérer et proposer des actions correctives à l'utilisateur en fonction du contexte présent ou prédit, à savoir par exemple prévoir un événement ou une situation, évaluer une situation courante ou possible, comparer deux situations et choisir la meilleure.

1.2.3 Avantage et inconvénient de l'informatique diffuse

L'intérêt croissant à ce domaine de l'informatique diffuse et son invasion dans notre vie motive plusieurs chercheurs à évaluer les impacts, les avantages et les inconvénients de ce dernier.

L'informatique diffuse fait référence à l'application des technologies de l'information et de la communication (TIC) se basant sur l'évolution de la technologie en miniaturisation et l'intégration de la microélectronique dans divers objets de la vie quotidienne en mesure de communiquer les uns avec les autres. Contrairement à la plupart des produits TIC actuels, cette nouvelle technologie est pourvue de capteurs capables de saisir, analyser et comprendre automatiquement leur environnement sans que l'utilisateur n'ait à intervenir directement, pour collecter des informations. Ces informations seront traitées par la suite pour offrir un service approprié aux usagers.

Les avantages indéniables de ces nouveaux développements technologiques, comme l'optimisation des processus de communication, la réduction de la charge environnementale concernant la consommation de matières et de l'énergie et leur potentiel économique ainsi que les améliorations sur le plan des applications destinées aux services médicaux (mise en place de SID pour les soins médicaux), risquent cependant d'être contrebalancés par des effets indésirables. Citons par exemple : La connexion totale des machines risque notamment de nuire à la santé des individus, en raison principalement d'une exposition accrue aux

champs électromagnétiques et, peut-être aussi, au développement durable, en raison notamment d'une augmentation de la consommation d'énergie et de la production de déchets ainsi qu'à la préservation de la vie privée.

1.3 Notion du contexte

Le contexte est un concept complexe. Il a été évoqué dans plusieurs disciplines souvent d'une manière complètement indépendante, par exemple la linguistique, l'intelligence artificielle et l'intelligence ambiante.

En linguistique, le contexte est décrit comme étant le texte qui entoure un élément de la langue et qui peut aider à déterminer son sens. Tandis qu'en intelligence artificielle, la définition du contexte découle de deux approches : la première est une approche dite logique propositionnelle proposée par McCarthy (McCarthy 1987), Guha (Guha 1991), McCarthy et Buvac (McCarthy 1993, McCarthy and Buvac 1997). Le contexte est défini, dans cette approche, comme étant les circonstances dans lesquelles est déterminée la valeur de vérité d'un terme. Cette conception est traduite en un ensemble de prédicats qui permettent de traiter un contexte dans le but de faire du raisonnement et des déductions logiques.

La deuxième approche est appelée raisonnement local (Local Models Semantics (LMS) / Multi Contexte Systèmes (MCS)) de Ghidini et Giunchiglia (Ghidini and Giunchiglia 2001). Dans l'approche LMS/MCS, le contexte est déterminé par un ensemble connu de faits sur lesquels est réalisé un raisonnement. L'approche donne une description incomplète de l'environnement. Le contexte est donc partiel et approximatif.

En intelligence ambiante, la notion du contexte a émergé suite à l'avancement technologique des capteurs. C'est ainsi que s'est développé, la tendance à mettre en place des solutions pour l'acquisition, le traitement et l'utilisation du contexte dans diverses applications. Ceci nécessite des descriptions claires de ce que constitue le contexte, aussi bien au niveau du développement des représentations logiques que celui au niveau des traitements, pour exploiter le contexte efficacement et aisément dans les applications.

1.3.1 Que disent les dictionnaires ?

Selon les dictionnaires de référence nous mentionnons les définitions suivantes de la notion du contexte :

- (Robert, 1991) : « ensemble du texte qui entoure un élément de la langue (mot, phrase, fragment d'énoncé) et dont dépend son sens, sa valeur » ou aussi « ensemble des circonstances dans lesquelles s'insère un fait ».
- (Encyclopédie Larousse) « ensemble des conditions naturelles, sociales, culturelles dans lesquelles se situe un énoncé, un discours ». Ou encore : « ensemble des circonstances dans lesquelles se produit un événement, se situe une action ».
- (Hachette Multimédia) « ensemble des éléments qui entourent un fait et permettent de le comprendre ».
- (Grand Dictionnaire : Office québécois de la langue française) « Énoncé dans lequel figure le terme étudié » ou encore « ensemble d'un texte précédant ou, suivant un mot, une phrase, un passage qui éclaire particulièrement la pensée d'un auteur ». Et, si l'on parle d'informatique : « Ensemble d'informations concernant l'action du stylet en rapport principalement avec sa localisation à l'écran, qui permet au système d'exploitation de l'ordinateur à stylet de différencier les commandes et l'entrée des données, et de fonctionner en conséquence ».

Ces définitions présentent le contexte comme étant un ensemble d'éléments ou de situations particulières qu'on peut assimiler à des informations. Ces éléments ou ces informations dépendent d'une action et permettent de concevoir la nature de « quelque chose » (fait, discours, événement, action, etc.) dont dépend précisément l'utilité du contexte. Cette utilité permet de « comprendre » de « fonctionner en conséquence », ou d'accorder un « sens », « une valeur » ou d'une façon générique de « servir l'interprétation ».

Dans leur étude sur la sensibilité au contexte dans l'informatique mobile, G.Chen et al. (Chen and Kotz 2000) ont montré que les définitions du contexte demeurent générales, vagues et inadaptées à un environnement informatique.

Dans la suite de ce document, nous nous intéresserons au contexte dans le domaine de l'informatique diffuse.

1.3.2 Définition du contexte dans un système informatique

Définir le contexte comme étant une information sur l'environnement d'un système informatique, ou bien comme étant des conditions qui déterminent un événement reste une définition sans limites apparente. Que peut-on obtenir si l'on envisage de décrire en détail les composants d'un système informatique ? Décrire l'ensemble de ces conditions et ces informations semble ambitieux. C'est pour ces raisons et bien d'autres que des chercheurs ont essayé d'analyser, de définir et de redéfinir le terme contexte pour l'utiliser dans leurs recherches.

Nous présenterons dans ce qui suit, les principales définitions du contexte dans le domaine de l'informatique.

Schilit, Adams et Want (Schilit, Adams et al. 1994) ont considéré que le contexte détient trois aspects importants qui donnent des réponses aux questions suivantes : où es-tu ? Avec qui? De quelles ressources disposes-tu à proximité ?

Schilit et Theimer (Schilit and Theimer 1994) ont défini le contexte comme étant les changements de l'environnement physique selon : la localisation, la description de personnes et d'objets dans l'entourage et les changements de ces objets.

Brown, (Brown 1995) restreint le contexte aux éléments de l'environnement de l'utilisateur : « les éléments de l'environnement d'un utilisateur dont l'ordinateur à connaissance ».

Plus tard, Brown introduit l'heure, la saison, la température, l'identité et la localisation de l'utilisateur à sa définition du contexte. Brown, Bovey et Chen, (Brown, Bovey et al. 1997) ont présenté un ensemble d'éléments extensibles pour caractériser le contexte dont les éléments de base sont : la localisation, l'ensemble des objets dont l'utilisateur a besoin, le temps et l'orientation spatiale (direction).

Simultanément aux travaux de Brown, des définitions émergent avec l'introduction explicite du temps et la notion d'état. Citons, Ryan, Pascoe et Morse, (Pascoe, Morse et al. 1997): « les éléments du contexte sont : la localisation de l'utilisateur, l'environnement, l'identité et le temps ».

Ward perçoit le contexte comme étant les états des environnements probables de l'application Ward et Hopper, (Ward, Jones et al. 1997) : « les états des environnements possibles de l'application ».

Pascoe décrit le contexte comme un sous-ensemble d'états physiques et conceptuels ayant un intérêt pour une entité particulière (Pascoe 1998).

Schmidt (Schmidt, Aidoo et al. 1999): « connaissances à propos de l'utilisateur et les états des équipements, l'entourage, la situation et la localisation ». Brézillon et Pomerol, (Brézillon and Pomerol 1999): « tout ce que n'intervient pas explicitement dans la résolution d'un problème, mais le contraint ».

Chen et Kotz, (Chen and Kotz 2000): « ensemble des états environnementaux et paramètres qui déterminent le comportement d'une application ou dans lequel un événement de l'application se déroule et ayant un intérêt pour l'utilisateur ».

Dey insiste sur la notion de pertinence de l'information en fournissant une définition où il tente de préciser la nature des entités relatives au contexte (Dey 2001): « toute information qui peut être utilisée pour caractériser la situation d'une entité. Toute entité est une personne, ou un objet qui est considéré significatif à l'interaction entre l'utilisateur et l'application, incluant l'utilisateur et l'application lui-même ».

Cette définition encapsule toutes les définitions précédentes puisqu'elle est d'ordre très générique. Dey précise que les paramètres du contexte peuvent être implicites ou explicites. En effet, les paramètres du contexte peuvent être fournis par l'utilisateur ou par des capteurs situés dans l'environnement de l'utilisateur et de l'application. Ils peuvent également parvenir d'une interprétation plus au moins complexe de ces paramètres. Dey a fourni une

définition qui englobe toute donnée implicite ou explicite qui peut être essentielle à l'application.

En 2001, Winograd a agréé la définition donnée par Dey et défend le fait qu'elle couvre tous les travaux existants sur le contexte. Cependant, il considère que les expressions utilisées par Dey comme « toute information » et « caractériser une entité » restent d'ordre très général et ne marquent aucune limite à la notion de contexte (tout peut être contexte). Pour apporter plus de précision à la définition de Dey, Winograd présente le contexte comme étant un ensemble d'informations structurées et partagées. Il détaille cette définition en disant : « la considération d'une information comme contexte est due à la manière dont elle est utilisée et à ses propriétés inhérentes ». Il soutient son idée par l'exemple : « le voltage des lignes d'électricité fait partie du contexte si le système en dépend; sinon, il ne peut être qu'un paramètre quelconque de l'environnement ».

En 2002, Henricksen définit le contexte comme étant « la circonstance ou la situation dans laquelle une tâche informatique se déroule » (Henricksen, Indulska et al. 2002, Henricksen, Indulska et al. 2002).

Le concept contexte a toujours été défini d'une manière générale, malgré le nombre élevé des définitions existantes. La plupart de ces définitions font référence à la localisation et l'environnement.

Une analyse de Brezilloon (Mostefaoui, Pasquier-Rocha et al. 2004) faite sur les définitions du terme contexte mène à conclure que la plupart des définitions sont des réponses aux questions suivantes:

- qui ? Pour identifier l'utilisateur courant et d'autres personnes présentes dans l'environnement;
- quoi ? Pour percevoir et interpréter l'activité de l'utilisateur ou des utilisateurs;
- où ? Pour localiser l'usager ou un événement du système;
- quand ? Pour repérer temporellement une activité, indexer un événement dans le temps ou calculer le temps écoulé de la présence d'un sujet à un point donné;
- pourquoi ? Pour comprendre la raison d'être de l'activité;

- comment ? Pour décrire la manière dont se déroule l'activité.

Répondant à ces questions peut générer un grand ensemble d'informations dont une grande partie est superflue ainsi que des problèmes de gestion de ces informations et de stockage dans le cadre des systèmes mobiles.

En 2005, Truong donne une nouvelle définition du contexte inspirée de celle de Dey, mais qui met plus l'accent sur le rôle du contexte dans l'interaction. Il considère le contexte comme un ensemble d'indications et d'informations qui définissent l'interaction entre l'utilisateur et l'application (Truong, Lee et al. 2005).

Dans une version plus abstraite et sans détailler les éléments composant le contexte, Miraoui décrit le contexte comme étant « Toute information dont le changement de valeur déclenche un service ou change la qualité (forme) d'un service », Il explique : « Cette définition est plus abstraite dans le sens où elle n'énumère pas les exemples du contexte et limite l'ensemble des informations contextuelles à celles qui sont reliées aux services (objectif d'un système diffus). Elle ne prend donc pas en compte les autres informations qui peuvent caractériser le contexte, mais ne jouent pas un rôle crucial pour l'adaptation des services. » (Miraoui 2009). (Explication schématisée Figure 2.5).

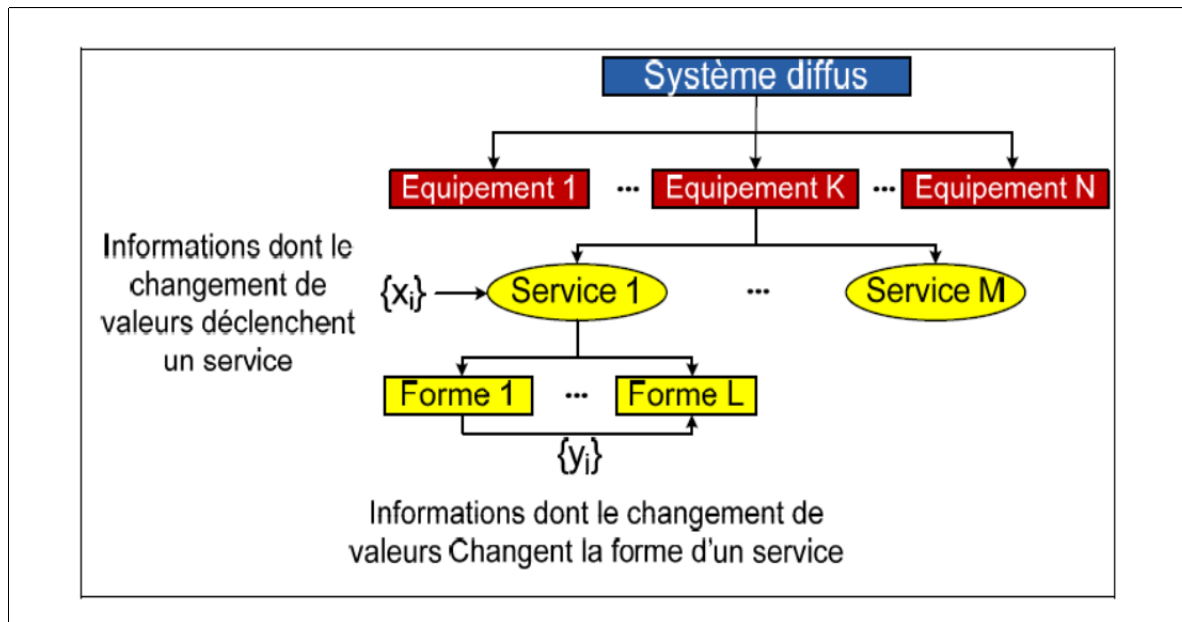


Figure 1.5 Composants d'un système diffus
Tirée de Miraoui (2009)

1.3.3 Synthèse et critique

Dans ce qui suit, nous ferons une brève synthèse des travaux sur le contexte pour cerner quelques grandes lignes essentielles qui entrent dans la définition du contexte qui n'était pas forcément pris en considération dans les définitions déjà utilisées dans la littérature. Ces grandes lignes procèdent de notre propre vision du contexte et son utilisation future dans nos travaux. Nous les exposons dans les sous-sections suivantes. Nous revenons plus tard pour discuter et analyser notre propre vision du contexte proposant une nouvelle définition (Chapitre.3).

1.3.3.1 Le contexte et le temps

Le contexte ne peut pas être pris indépendamment de l'espace-temps. Ce constat rejoint l'avis de Ryan et al (Baldauf, Dustdar et al. 2007) sur l'importance du temps dans la caractérisation du contexte. En effet, toute situation, évènement ou interaction s'accomplit

obligatoirement dans un intervalle temporel. La granularité de cet intervalle est cruciale puisqu'elle précise l'étendue temporelle des informations décrivant le contexte. Ce fait influence aussi la quantité d'information : on peut passer d'un nombre réduit d'informations à une base de données, selon qu'on s'intéresse à un évènement isolé (évènement de courte durée) ou à une suite d'évènement (activité).

1.3.3.2 Contexte et espace (contexte spatial)

Dans la littérature, le contexte spatial est réduit à la variable localisation (localisation de l'utilisateur) dans le meilleur des cas. Nous pouvons identifier les places se trouvant dans l'environnement relatif à l'utilisateur (Kapitsaki, Prezerakos et al. 2009). On peut retrouver plutôt la notion de « contexte d'usage » qui définit et englobe toute information caractérisant l'utilisateur, l'environnement et la plateforme (Najar, Pinheiro et al. 2014).

Li (Yuan, Li et al. 2007), présente une définition du contexte spatial (ou contexte géographique) en trois dimensions : (1) le contexte spatial statique (2), le contexte spatial dynamique et (3) le contexte spatial interne. Il les définit comme suit :

1. Le contexte spatial statique : représente des informations géospaciales qui peuvent avoir un impact sur l'environnement de l'utilisateur. Par exemple les informations concernant le réseau routier, les bâtiments, les magasins, les structures intérieures (salles, bureaux, appartements ...).
2. Le contexte spatial dynamique réunit les informations géoréférencées collectées par différents types de capteurs. Par exemple les données de vitesse captées sur les routes, les données sur le taux d'occupation des lots de stationnements, etc.
3. Le contexte spatial interne : rassemble les informations accessibles à partir des appareils locaux (ex. GPS). Par exemple, un automobiliste peut avoir des informations concernant sa position, sa vitesse, son orientation, sa destination avec son GPS de voiture. Il peut

également obtenir de l'information à partir d'un capteur intégré tel qu'une jauge d'essence d'un véhicule.

1.3.3.3 Contexte et Interaction

L'interaction dans un système diffus doit être définie selon deux axes : 1) d'interaction homme-machine exposée par Dey dans sa définition du contexte, 2) d'interaction machine-machine. Selon notre étude, ce dernier axe ne semble pas avoir été pris en considération. Or cet aspect devient de plus en plus important vu le développement croissant d'infrastructures M2M (machine to machine) et services web où les informations contextuelles relatives à l'interaction entre applications peuvent soutenir la réactivité et l'efficacité de ces systèmes à l'égard de l'utilisateur. Cet aspect est défendu par Brézillon et Coutaz qui considèrent « le contexte comme un problème interactionnel dans lequel l'interaction elle-même peut être importante » (Gensel, Villanova-Oliver et al. 2008).

1.3.3.4 Contexte et pertinence

Un SID possède un environnement très dynamique intégrant des équipements mobiles hétérogènes. Ceci peut multiplier les erreurs dans les informations contextuelles soit au niveau de la collection, ou au niveau de l'interprétation et de la présentation de ces informations. Ainsi, pour assurer une adaptation fiable selon le contexte, une évaluation de la qualité de l'information contextuelle est nécessaire.

Krause et Hochstatter, (Krause and Hochstatter 2005) listent les causes des erreurs dans les informations contextuelles:

- l'information ou la source fondamentale n'est pas valide;
- l'information n'est pas adéquate à la situation actuelle;
- Imprécision des capteurs à cause des contraintes physiques et temporelles;
- l'information sur le profil par défaut n'est pas applicable à la situation actuelle;

- les règles de raisonnement ne sont pas appropriées à toutes les situations (exemple : le degré de luminosité dans une chambre ne signifie pas obligatoirement qu'il fait jour ou nuit).

La pertinence du contexte permet de cerner les informations contextuelles utiles et directement impliquées dans la conception d'un système sensible au contexte. Dey évoque ce besoin dans sa définition sans détailler comment déterminer quelles sont les informations pertinentes et comment parvenir à ces informations pertinentes.

Pour Henricksen, Indulska et Rakotonirainy, (Henricksen, Indulska et al. 2002, Truong, Lee et al. 2005) l'imperfection de l'information contextuelle est considérée comme une des caractéristiques du contexte dans un SID. Elle peut causer :

- des informations incorrectes qui échouent à refléter l'état réel du monde qu'elles modélisent;
- des informations inconsistantes qui contiennent des données contradictoires;
- des informations incomplètes dont certains aspects du contexte restent inconnus.

Bucur et al (Bucur, Beaune et al. 2005) ont approfondi la caractérisation et la formulation de la pertinence des informations du contexte en intégrant le concept de finalité au contexte (définir les éléments contextuels selon l'objectif de leur utilisation). La finalité renvoie à l'objectif pour lequel le contexte est utilisé à un moment donné. Cet objectif peut être à court ou à long terme.

Si l'objectif est fixé à long terme, le contexte est archivé au fil du temps tout en entretenant la pertinence. Par exemple si un système vise à adapter l'interface utilisateur selon les habitudes d'utilisation, alors les données des utilisations passées restent pertinentes parce qu'elles peuvent déterminer les prochaines adaptations et permettent de prédire les actions futures de l'utilisateur. Si l'objectif est à court terme, disant instantané, le contexte actuel sera utilisé immédiatement dans le processus d'adaptation. Citons comme exemple, le passage au mode silencieux du téléphone à l'entrée de l'utilisateur à la bibliothèque.

Certains chercheurs ont proposé des approches pour la modélisation de la qualité et la pertinence de l'information contextuelle. Par exemple, Razzaque, Dobson et Nixon, (Razzaque, Dobson et al. 2006) ont développé une approche générique qui consiste en un processus par étapes. Tandis que Krause et Hochstatter, (Fuchs, Hochstatter et al. 2005) ont proposé une méthode pour la représentation des aspects de la qualité du contexte intégré dans leur modèle de représentation du contexte CMM (context meta-model).

La plupart des travaux sur la pertinence et la qualité de l'information contextuelle ne fournissent pas de méthodes formelles pour la modélisation de la qualité, mais se réduisent à des méthodes incomplètes, imparfaites et limitées à l'application.

En générale, ces quatre questions sur: le temps, le contexte spatial, l'interaction et la pertinence présentent encore un grand défi dans la conception des systèmes, des applications et des services sensibles au contexte.

1.3.4 Utilisation du contexte

Le contexte est un concept qui est pris en compte dans différents domaines de l'informatique tels que l'apprentissage automatique, le traitement du langage naturel, l'extraction de l'information, l'informatique diffuse ou aussi la sécurité informatique. Le but de la prise en compte du contexte est de consolider l'adaptabilité et l'aide à la décision du système. L'hétérogénéité et l'ubiquité des entités communicantes dans un environnement diffus rendent le contexte, un concept crucial dans l'informatique diffuse. Ces deux aspects de l'infrastructure d'un système diffus nécessitent l'adaptabilité des services fournis et des médias des utilisateurs en fonction du lieu, de l'activité, etc.; c'est à dire du contexte. On parle donc de systèmes informatiques diffus sensibles au contexte (en anglais *contexte-aware*), puisqu'ils sont capables d'utiliser le contexte d'un élément pour changer leurs fonctionnements; l'objectif ultime étant d'offrir de meilleurs services aux usagers.

Chalmers (Chalmers 2004) identifie cinq principales utilisations du contexte dans les systèmes informatiques:

1. Senseur du contexte : où le contexte est saisi et les informations décrivant le contexte courant sont présentées à l'utilisateur.
2. L'association du contexte aux données ou encore appelé l'augmentation contextuelle par exemple, le fait d'associer des notes d'une réunion aux personnes assistant à la réunion et le lieu où elle s'est déroulée.
3. Le pouvoir de découvrir des ressources contextuelles, par exemple, pouvoir imprimer un document sur l'imprimante la plus proche.
4. Utilisation dans des événements déclenchés par le contexte pour produire les actions telles que par exemple le chargement de données cartographiques à l'entrée dans une région.
5. Médiation contextuelle : elle consiste à utiliser le contexte pour modifier un service. Par exemple pour décrire les limites et les préférences dans une large variété de données offertes, et ainsi pouvoir afficher les plus appropriées.

Selon Dey et Abowd,(Dey and Abowd 2000) les premiers travaux de recherche dans le domaine des systèmes informatiques sensibles au contexte étaient réalisés dans le cadre du système de badges actifs d'Olivetti (Ward, Jones et al. 1997). Le système était développé par la suite entre 1989 et 1992. Il servait comme un moyen de localisation du personnel dans un immeuble pour acheminer automatiquement les appels téléphoniques. Il se base sur l'idée que chaque personne porte un badge qui émet périodiquement des signaux vers un système centralisé de localisation. Actuellement le plus grand système de badges actifs utilisé se trouve au laboratoire d'informatique de l'université Cambridge, comptant plus de 200 badges et 300 capteurs utilisés quotidiennement.

Ces systèmes ne sont qu'un exemple d'une large variété de systèmes informatiques sensibles au contexte. Ces systèmes font un usage intensif des informations contextuelles pour progresser et améliorer le quotidien des utilisateurs.

Plusieurs études (Pejovic and Musolesi 2015), (da Rosa, Barbosa et al. 2016) (VanSyckel and Becker 2014) ont analysé et comparé des travaux de recherches relatifs au contexte, en essayant de mettre l'accent sur les applications, les informations contextuelles utilisées, et la manière dont ils s'en servent. Ils concluent qu'en majorité, les systèmes informatiques diffus

font appel principalement à la localisation de l'utilisateur comme information contextuelle, rarement un index temporel ou la localisation des objets à proximité ne sont pas utilisés. Cela pourrait s'expliquer par les difficultés liées à la saisie et à la collecte de l'information contextuelle ainsi qu'à son traitement.

1.3.5 Catégorisation du contexte

Vue la diversité et l'hétérogénéité des informations contextuelles, il est utile de faire une classification par catégorie pour faciliter leur utilisation. Dans cette section, nous présentons les catégorisations existantes qui synthétisent les informations contextuelles.

Schilit, Adams et Want,(Schilit, Adams et al. 1994) et Dey (Dey 2001) ont présenté le contexte sous deux classes : un contexte primaire qui englobe les informations sur la localisation, l'identité, le temps et l'activité; et un contexte secondaire qui peut être déduit du contexte primaire (exemple : de la localisation, on peut déduire les équipements disponibles).

Tandis que Chen et Kotz, (Chen and Kotz 2000) ont présenté deux catégories : un contexte actif qui influence le comportement ou l'aspect d'une application et un autre passif qui est essentiel, mais pas crucial pour l'application.

Petrelli a proposé une autre catégorisation qui distingue un contexte matériel et un autre social. Le contexte matériel peut englober par exemple : la localisation, les machines, la plateforme existante ; et le contexte social rassemble les aspects sociaux comme la relation entre les individus (Razzaque, Dobson et al. 2006).

Quant à Gwizdka, (Gwizdka 2000), il a présenté deux catégories : le contexte interne contenant l'état de l'utilisateur et le contexte externe pour l'état de l'environnement.

Hofer et al., (Baldauf, Dustdar et al. 2007) divisent les contextes en contexte logique et physique : le contexte physique est mesuré par les capteurs physiques alors que le contexte logique contient les informations sur l'interaction.

Une catégorisation en six classes a été proposée par Razzaque, Dobson et Nixon, (Razzaque, Dobson et al. 2006). Ils ont défini les catégories comme suit :

1. Contexte utilisateur : englobe les informations sur les utilisateurs du système informatique, par exemple : son identification, la liste de ses tâches, etc.
2. Contexte physique : présente les informations liées à l'environnement physique, telles que la localisation, la température, le niveau de bruit, etc.
3. Contexte du réseau : il fournit des informations concernant principalement le réseau informatique. Exemple : connectivité, bande passante, protocole, etc.
4. Contexte d'activité : liste les événements qui se sont produits dans l'environnement ainsi que leur estampille temporelle. Exemple : la lumière s'allume, il fait nuit, une personne entre, il pleut, etc.
5. Contexte matériel : il identifie les équipements et les appareils utiles de l'environnement. Il décrit le profil et les activités des dispositifs de l'environnement (identification, localisation, etc.)
6. Contexte de service : il informe sur ce qui peut être obtenu par exemple les informations relatives aux fonctionnalités que le système peut offrir.

L'avantage de cette catégorisation est qu'elle englobe les éléments considérés par les catégorisations précédentes.

En adoptant une approche fondée sur le service (Miraoui 2009), (Ameyed, Miraoui et al. 2015) proposent une catégorisation des informations contextuelles en deux classes: (1) : les informations de déclenchement d'un service qui rassemble les informations dont le changement de valeurs provoque le déclenchement automatique des services fournis par un équipement et (2) : les informations de changement de forme d'un service qui regroupent les informations dont le changement de valeurs provoque le changement de forme d'un service (qualité).

L'avantage de cette catégorisation est sa simplicité puis qu'elle catégorise les informations contextuelles en seulement deux classes. Elle semble complète et couvre tous les aspects du contexte.

Notons que les catégorisations que nous avons citées ne présentent qu'une liste non exhaustive. De nouvelles classifications peuvent être révélées à mesure que de nouvelles caractéristiques des informations contextuelles seront découvertes.

La classification est une étape importante dans la mesure où elle permet aux développeurs des SID de manipuler plus efficacement les informations contextuelles. Ceci influencera l'efficacité de l'adaptation et la qualité de services offerts.

1.4 Notion de sensibilité au contexte (context-awareness)

La notion de sensibilité en anglais "context-awareness" définit la capacité d'un système à s'adapter aux changements du contexte. Dans cette partie, nous faisons un bref survol des définitions existantes de la sensibilité au contexte.

Schilit, Adams et Want, (Schilit, Adams et al. 1994) c'était ces auteurs qui ont évoqué le terme *sensibilité au contexte* pour la première fois, dans leurs travaux sur un système de localisation. Ils ont défini la sensibilité au contexte comme étant l'aptitude d'une application à s'ajuster et s'adapter au contexte de son exécution selon par exemple: la localisation, les usagers à proximité, les équipements accessibles, ainsi que les changements de ses objets dans le temps.

Brown, (Brown 1995) a défini la sensibilité au contexte dans son travail sur un guide touristique comme toute application qui prend en compte le contexte de l'utilisateur.

Chen et Kotz, (Chen and Kotz 2000) ont classé la sensibilité au contexte en sensibilités active et passive : (1) la sensibilité active au contexte renvoie aux applications qui s'adaptent automatiquement au contexte découvert par le changement du comportement de l'application. Tandis que (2) la sensibilité passive au contexte englobe les applications qui exposent un nouveau contexte ou celui mis à jour à un utilisateur intéressé ou produit, un contexte persistant pour une utilisation ultérieure.

Dey, (Dey 2001) l'a défini comme un système qui utilise le contexte pour fournir des informations et/ou des services pertinents à l'utilisateur. La pertinence dépend de la tâche sollicitée par l'utilisateur.

Cette définition de Dey a été adoptée par la majorité des chercheurs dans ce domaine. Elle a mis en évidence trois catégories de fonctions liées à la présentation et au stockage d'information, ainsi qu'à l'exécution de services selon le contexte. Ces catégories sont présentées comme suit :

1. La première catégorie

Cette catégorie englobe la présentation d'information et de services. On y classe les applications qui présentent des informations du contexte. Ces applications peuvent offrir un éventail d'opérations appropriées à l'utilisateur.

2. La deuxième catégorie

La deuxième catégorisation concerne l'exécution automatique de services. Elle décrit les applications qui déclenchent une commande, ou reconfigurent automatiquement le système selon les changements de contexte et son recours à l'utilisateur.

3. La troisième catégorie

Cette catégorie s'intéresse au stockage d'informations selon le contexte. Les applications associent des données au contexte de leur utilisation.

En se basant sur l'approche orientée service (Miraoui 2009) ont défini un système sensible au contexte comme suit: «Un système est dit sensible au contexte s'il peut changer automatiquement les formes des services ou déclencher un service comme réponse au changement de la valeur d'une information ou d'un ensemble d'informations qui caractérisent le service. »

Cette définition présente davantage un système sensible au contexte puisqu'elle explique la sensibilité en termes d'interaction du système aux transformations du contexte par le

déclenchement d'un service ou le changement des formes de services (Ameyed, Miraoui et al. 2015).

Malgré les multiples travaux dans le domaine de la sensibilité au contexte, ce domaine est loin d'être au point. En effet, plusieurs sujets sont encore à approfondir et à étudier.

On identifie encore des lacunes concernant les définitions et la pertinence des informations contextuelles : e.g (i) la notion de contexte n'est pas encore bien définie. (ii) les travaux existants manquent de modèles et de méthodes conceptuels.

1.5 Modélisation du contexte

Pour concevoir un système sensible au contexte efficace, le contexte doit être bien saisi et modélisé sous une forme appropriée qui favorise le partage de l'information entre les différents équipements SID. Utilisant ainsi un modèle apte à fournir un niveau supérieur d'abstraction pour faciliter la tâche d'adaptation.

Comme indiqué par (Magherini, Fantechi et al. 2013) il y a généralement un écart significatif entre l'information saisie et l'information qui est utile pour les applications. Cet écart peut être surmonté par différents types de traitement d'informations de contexte. La modélisation est ainsi une étape cruciale dans le traitement du contexte. En effet, la modélisation consiste à faire l'analyse et la conception de l'information contextuelle comprise dans le système sous forme d'une représentation abstraite au niveau de la structure de données et au niveau de la sémantique.

Plusieurs approches de modélisation ont été proposées, étudiées, et analysées dans la littérature. Dans cette section nous allons exposer, analyser et classifier certaines de ces approches. Il y a plusieurs classifications pour la modélisation du contexte. Nous nous référons à la classification de Strang et Linnho-Popien où ils ont classifié les approches de modélisation par le modèle de structures de données utilisé pour échanger des informations contextuelles dans leur système respectif.

1.5.1 Attributs-Valeurs

Cette approche était proposée par Schilit, Adams et Want, (Schilit, Adams et al. 1994) pour la gestion des informations contextuelles d'un environnement. La représentation attribut/valeur est la structure de données la plus simple pour la modélisation des informations contextuelles. Les données de contexte décrites sous la forme d'attribut-valeur sont organisées en couple (nom de l'attribut - valeur), paramètres attributs et sous forme de balises champs. L'attribut est une métadonnée qui définit la nature (type) de la valeur à laquelle il est associé.

Par exemple, pour une application de localisation, une information sur l'occupation d'une pièce ou un bâtiment est représentée comme suit :

OCCUPANTS = adams:schilit:theimer:weiser:Welch

Cette approche est fréquemment utilisée dans les services distribués. Les services sont présentés sous une liste d'attributs sous la forme d'attribut/valeur. Ces paires attribut/valeur sont utilisées par un algorithme pour la découverte de services (Held, Buchholz et al. 2002, Indulska, Robinson et al. 2003). Selon le travail de Kammanahalli (Kammanahalli, Gopalan et al. 2004, Strang and Linnhoff-Popien 2004) les informations peuvent être sauvegardées dans une base de données sous forme de tables dont les colonnes correspondent à une information représentée par une même métadonnée. Chaque ligne (tuple) représente une suite d'informations pour caractériser les différents documents suivant le terminal de l'utilisateur tel que présenté dans le (Tableau 1.1).

Tableau 1.1 Table décrivant des documents
Tiré de Kammanahalli, Gopalan et al. (2004)

Activité	Equipement	Type de document présenté
Courriel	Ordinateur	Entier
Courriel	Ordinateur	Entier
Courriel	Ordinateur	Résumé
Courriel	Ordinateur	Résumé
Lecture	Ordinateur	Entier
Lecture	Ordinateur	Entier
Réunion	PDA	Résumé
Réunion	Mobile	Résumé vocal
Réunion	PDA	Résumé

Le modèle attribue-valeur est simple et facile à gérer, mais reste non convenable dans le cas de structures complexes, ne permet pas un raisonnement efficace sur le contexte et ne favorise pas la réutilisabilité.

1.5.2 Modèle de représentation par balises

Cette représentation est présentée sous forme d'une structure de donnée hiérarchique composée de balises avec des attributs et des contenus qui peuvent être définis eux-mêmes par d'autres balises. Fondé sur les balises, ce modèle utilise donc des langages dérivés du SGML (Standard Generic Markup Language) essentiellement, le XML (eXtended Markup Language). Les entités du contexte sont présentées sous forme de pages web auxquelles on peut accéder par une URL (adresse web). Il y a plus qu'un langage qu'on peut utiliser sous cette approche. Nous citons :

- **ConteXtML**

Le ConteXtML (Contexte Markup Language) est un protocole basé sur XML pour échanger des informations contextuelles entre un client mobile et un serveur. Les messages ConteXtML sont regroupés dans des balises <context> ou des éléments.

- **CC/PP**

Le langage CC/PP (Composite Capabilities / Preferences Profiles), est une recommandation de W3C (World Wide Web Consortium) dans le cadre des travaux sur « device dependance » pour supporter la négociation de contenu entre un navigateur web et un serveur. Il est basé sur RDF (Resource Description Framework) qui permet de décrire le profil d'un terminal (PC, PDA, laptop, etc), suivant plusieurs caractéristiques matérielles, logicielles, préférences utilisateurs. Le CC/PP est utilisé pour personnaliser le contenu sur la base de ses capacités et ses préférences.

Held, Buchholz et Schill, (Held, Buchholz et al. 2002) ont proposé une extension de ce langage, le CSCP (Comprehensive Structured Context Profiles). Ce dernier résout les lacunes de CC/PP au niveau de la structuration. Le CSCP appuie la flexibilité de RDF pour formuler les structures naturelles d'informations d'un profil utile pour l'information contextuelle. Le CSCP est un métalangage basé sur RDF et en hérite l'interchangeabilité, la décomposabilité et l'extensibilité.

Indulska et al., (Indulska, Robinson et al. 2003) ont dévoilé une approche semblable en présentant le CC/PP Context Extension qui se base sur le vocabulaire de CC/PP et UAProf (User Agent Profile) avec un certain nombre d'arbres composant/attribut correspondant à certains aspects du contexte dans l'informatique diffuse.

Ils existent d'autres approches de modélisation par le langage de balises, telles que le PDDL (Pervasive Profile Description Language) ou le CCML (Centaurus Capability Markup Language). Cependant, ces langages restent spécifiques à un domaine, limités à un ensemble d'aspects du contexte (localisation, environnement, etc.) et incapables de représenter des contextes complexes.

1.5.3 Modèle graphique

Cette approche très connue permet la modélisation d'informations contextuelles selon un graphe conceptuel. L'UML (Unified Modeling Language), grâce à sa structure générique, est bien appropriée pour modéliser le contexte. Utilisant l'UML (Strang and Linnhoff-Popien 2004, Henriksen and Indulska 2006) a fourni une modélisation graphique des informations contextuelles d'un système de contrôle du trafic aérien. Un autre exemple est le modèle graphique de contexte qui se base sur le formalisme « entité/association » qui était introduit par Henriksen (Strang and Linnhoff-Popien 2004, Henriksen and Indulska 2006).

Plus tard Henriksen et Indulska, ont développé une approche de modélisation graphique basée sur la méthode ORM (Object Role Modeling). Cette méthode est orientée « fait » pour l'analyse de l'information au niveau conceptuel. Elle consiste à identifier les types de faits appropriés et les rôles des types d'entités (De Virgilio and Torlone 2006, Serral, Valderas et al. 2010).

Selon la littérature cette modélisation est plus formelle. Elle permet de saisir les différents types d'informations contextuelles et appuyer le raisonnement sur le contexte. Cette approche permet de résoudre l'ambiguïté dans l'information contextuelle.

Suite à des améliorations apportées à l'approche ORM (Bouzy and Cazenave 1997, Henriksen and Indulska 2006) présente le CML (Context Modeling Language) qui servira de base pour une extension vers une représentation basée sur XML, XCML.

Virgilio et Torlone, (De Virgilio and Torlone 2006, Niforatos, Karapanos et al. 2012) quant à eux, ils ont présenté le GPM (General profile Model) pour la description de représentations hétérogènes de données web d'une manière uniforme. Le GPM peut être utilisé pour décrire de nombreux contextes d'une manière homogène et uniforme et fournit un outil robuste pour la conception et l'analyse des applications sensibles au contexte. L'avantage de cette approche est sa simplicité au niveau de la représentation du contexte, mais demeure la moins formelle parmi les autres méthodes et n'amène à aucune approche empirique.

1.5.4 Modèle orienté objet

Une approche orientée objet favorise la réutilisabilité, l'encapsulation et l'héritage. Ceci permet de résoudre des problèmes liés à l'aspect dynamique du contexte dans l'environnement ubiquitaire. Au niveau du traitement par exemple, les détails de traitement du contexte sont encapsulés au niveau de l'objet et donc masqués aux autres composants.

Bouzy et Cazenave, (Bouzy and Cazenave 1997, Hofer, Schwinger et al. 2003) ont présenté un modèle orienté objet pour le contexte pour simplifier la représentation des connaissances dans des systèmes complexes. Ils ont montré que le contexte peut se présenter sous plusieurs types en donnant des exemples : temporel, but, spécial et global. En s'appuyant sur les avantages de l'approche de modélisation orientée objet tels que l'héritage et la réutilisation, ils ont présenté un modèle qui permet de définir un nombre restreint de propriétés, fonctions et règles dans le but de simplifier la représentation des connaissances dans des systèmes complexes.

Dans le cadre du projet Guide (un guide touristique sensible au contexte), Cheverest et al. (Attardi and Simi 1994, Niforatos, Karapanos et al. 2012) ont proposé une approche basée sur l'intégration d'un modèle orienté objet et un modèle d'information hypertexte. Le modèle proposé était formé de deux objets distincts : l'objet point de navigation et l'objet localisation. Le modèle fourni reste limité à la localisation et ne couvre pas l'aspect général du contexte.

Plus tard Hofer et al. (Hofer, Schwinger et al. 2003, Bao, Smart et al. 2010) ont développé l'approche « HYDROGEN ». Intégrant le domaine de l'informatique mobile et sous une architecture à 3 niveaux, leur modèle de contexte est modélisé sous forme de diagramme de classes UML (figure 1.6). Ils ont décomposé le contexte en contexte local et en contexte distant. Où chaque type de contexte est formé de plusieurs objets contextuels qui composent la superclasse de plusieurs éléments du contexte, tel que : le temps, le réseau, la localisation, l'utilisateur, la machine, etc. qui peuvent être ajoutés par héritage de la superclasse (Figure 1.6).

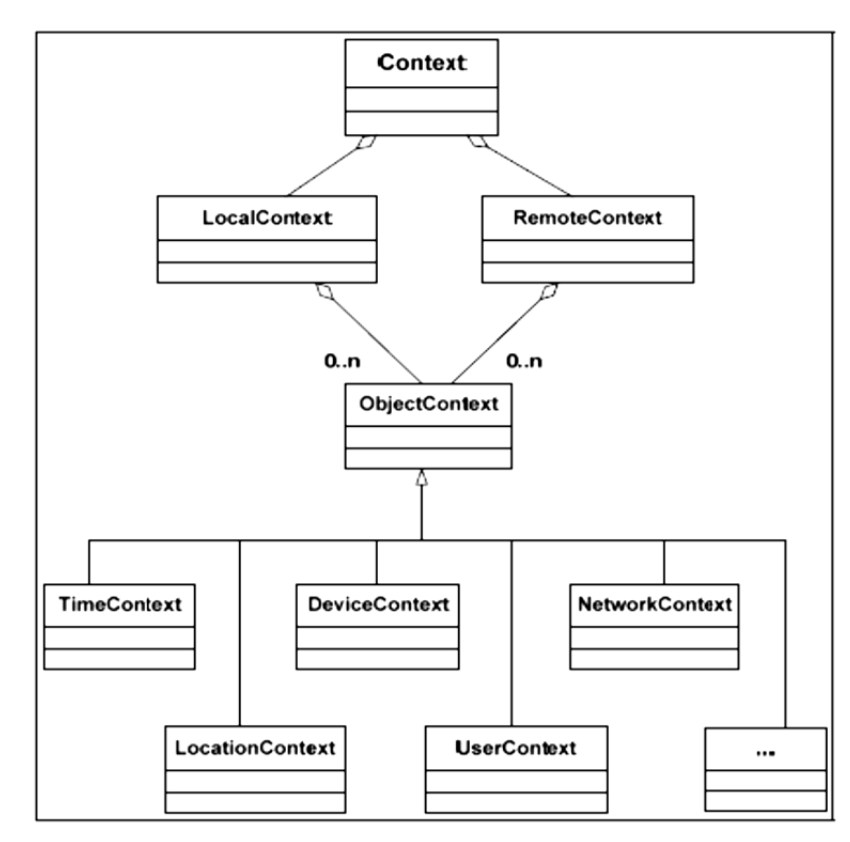


Figure 1.6 Le modèle UML de l'approche Hydrogen
Tirée de Bao, Smart et al. (2010)

Cette approche favorise la réutilisation des éléments du contexte puisqu'elle offre la possibilité de représenter le contexte comme une hiérarchie et de décrire chaque élément indépendamment des autres en utilisant l'encapsulation. Mais elle ne donne pas la possibilité de représenter les relations entre les objets. Elle est bien efficace en termes de distribution et d'abstraction. Cette modélisation ne favorise pas le partage du contexte entre applications et reste propre à une application spécifique.

La représentation orientée objet du contexte s'appuie sur les propriétés de nommage, d'encapsulation, de réutilisation et d'héritage, ce qui permet une intégration facile de la représentation du contexte dans l'application.

Dans ce modèle, les termes sont présentés par les classes et les informations par les attributs de la classe. Grâce à la technique d'encapsulation, les détails du contexte sont invisibles aux autres objets.

1.5.5 Modèle logique

Un modèle logique offre une représentation formelle des informations contextuelles. En utilisant un processus de raisonnement ou un moteur d'inférence, un modèle logique peut déduire de nouveaux faits basés sur des règles existantes dans le système. Parmi les premiers travaux utilisant cette approche, ceux de Carthy et Buvac, (Attardi and Simi 1994, Akman and Surav 1997) qui ont introduit le contexte comme objet formel. Ils ont défini de simples axiomes pour les événements ou phénomènes ayant un sens commun et ont traité le contexte associé à une situation particulière. Ils ont fourni une relation de base *ist (c,p)*, qui signifie que la proposition *p* est vraie dans le contexte *c*, définie par des formules telles que :

$C_0: ist (\text{contexte} - of (Sherlock Holmes stories) , Holmes is detective).$

Le module de McCarthy supporte le concept d'héritage (Bao, Smart et al. 2010, Fenstad, Halvorsen et al. 2012).

Un autre représentant précoce de ce type d'approche est la théorie des situations introduite par Akman & Surav, (Barwise and Perry 1981, Akman and Surav 1997), (Gray and Salber 2001, Fenstad, Halvorsen et al. 2012) Cette approche est inspirée de la théorie proposée par (Barwise et Perry, (Barwise and Perry 1981, Bacon, Bates et al. 1997). Ceux-ci ont tenté de donner un modèle sémantique théorique du langage naturel dans un système de logique formelle. Akram et Surav ont fourni par la suite une extension pour ce modèle. Ils représentent les faits liés à un contexte particulier avec des expressions non paramétrées et pris en charge par le type de la situation qui correspond au contexte.

Une approche similaire est proposée par Gray & Salber, (Gray and Salber 2001, Ranganathan, Campbell et al. 2002). Ils utilisaient la logique des prédicats du premier ordre

comme une représentation formelle des informations contextuelles et leurs relations. Une autre approche de cette même catégorie est le système multimédia développé par Bacon et al. (Bacon, Bates et al. 1997, Öztürk and Aamodt 1997). Dans ce système, l'emplacement pris comme un aspect du contexte est exprimé en tant que fait dans un système fondé sur des règles. Le système lui-même est mis en œuvre en Prolog.

Ranganathan et al. (Ranganathan, Campbell et al. 2002, De Bruijn, Lausen et al. 2006) ont proposé une modélisation du contexte basée sur la logique des prédicats dans le cadre du projet ConChat. Le modèle permet l'utilisation des opérations sur le contexte par exemple, la conjonction, la disjonction, la négation et la quantification. Cette approche offre la possibilité de créer des expressions complexes en logique du premier ordre et de déduire un contexte de haut niveau à partir du contexte de base (capturé) en utilisant une approche basée sur des règles. Le modèle définit une structure de base pour présenter de façon atomique chaque objet du contexte.

Les approches de déduction des solutions basées sur une modélisation logique offrent les mécanismes les mieux adaptés pour réaliser l'abstraction des informations en concepts. Toutefois, l'inférence sur un ensemble de prédicats sollicite un nombre important de ressources système, en matière de mémoire et capacité de calcul.

1.5.6 Les ontologies

Une ontologie est le niveau de base d'un schéma de représentation des connaissances. Elle présente un ensemble de blocs à partir duquel des modèles du champ d'informations ou d'une certaine partie de ce champ d'informations sont conçus.

L'une des premières approches de la modélisation du contexte avec ontologies a été proposée par Ozturk et Aamodt, (Öztürk and Aamodt 1997, Chen, Finin et al. 2005). Ils l'ont utilisé pour analyser des études psychologiques sur la différence entre le rappel et la reconnaissance de plusieurs questions, en combinaison avec des informations contextuelles. Les travaux

utilisant cette approche se sont multipliés par la suite dans la modélisation du contexte. Parmi les plus intéressants, nous citons :

De Bruijn, (Chen, Perich et al. 2004, De Bruijn, Lausen et al. 2006) a proposé un moyen uniforme pour spécifier le modèle de concepts de base ainsi qu'un ensemble de sous-concepts et de faits, permettant le partage des connaissances contextuelles et sa réutilisation dans un système informatique ubiquitaire.

Chen, Finin et Joshi (Wang, Zhang et al. 2004, Chen, Finin et al. 2005), ont proposé une approche fondée sur l'idée d'un courtier de contexte (Context Broker Architecture ou CoBrA). Le contexte est géré et maintenu à travers un serveur central qui servira à maintenir l'état du contexte vis-à-vis d'un ensemble de terminaux et d'agents. Il fonctionne à la base d'une collection d'ontologies nommée COBRA-ONT exprimée dans le langage OWL pour la modélisation de contexte dans un environnement d'une salle de rencontre intelligente. Le modèle ontologique assiste le courtier de contexte dans le partage des informations contextuelles avec d'autres agents et lui permet de raisonner sur le contexte. Pour appuyer le raisonnement sur les ontologies dans CoBrA, les auteurs ont conçu un moteur d'inférence OWL appelé F-OWL. Par ce modèle les auteurs ont fourni un système doté : (i) de la capacité de raisonner avec le modèle d'ontologie défini par la dernière recommandation du standard du langage OWL du W3C, (ii) d'aptitude pour le support de la vérification de consistance de connaissances en utilisant des règles axiomatiques définies dans Flora-2 (qui est un langage de base de connaissance orientée objet et une plateforme de développement des applications qui traduit un langage unifié de F-logic,) et enfin (iii) d'une interface de programmation d'applications (API) ouverte pour l'intégration des applications Java.

Chen et al., (Chen, Perich et al. 2004, Preuveneers, Van den Bergh et al. 2004), quant à eux, ils ont présenté une ontologie partageable appelée SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications). Cette ontologie favorise le partage de connaissances, le raisonnement sur le contexte et l'interopérabilité dans un système ubiquitaire diffus. Présentement, l'extension de SOUPA possède des ontologies expérimentales adaptées aux applications diffuses sensibles au contexte dans un espace

intelligent et supporte la communication des données d'égal-à-égal dans un environnement diffus.

H.Wang et al., (Lemlouma and Layaïda 2003, Wang, Zhang et al. 2004) ont fourni une ontologie du contexte nommée CONON (CONtext ONtology) codée en OWL pour la modélisation du contexte dans un environnement diffus et qui supporte le raisonnement logique sur le contexte.

CONON présente une première ontologie du contexte supérieur qui saisit les concepts généraux du contexte de base. Elle offre un modèle extensible sur lequel on peut ajouter des ontologies particulières d'un domaine spécifique d'une manière hiérarchique. L'approche CONON offre la possibilité d'ajouter des règles personnalisées en plus des règles de déductions de bases utilisées pour les ontologies. Ces règles sont présentées par des prédicats associés à la logique de premier ordre et qui sont introduites pour préciser la situation de l'utilisateur.

Preuveneers et al., (Preuveneers, Van den Bergh et al. 2004, Cremene, Riveill et al. 2006) ont proposé une ontologie du contexte adaptable et extensible pour les systèmes sensibles au contexte qui sont soit des micros systèmes embarqués, soit des plateformes de services. Le modèle ontologique est décrit par le langage OWL, mais peut être formulé avec un autre langage. Il est formé de quatre ontologies de base : l'utilisateur, l'environnement, la plateforme et enfin le service. Cette approche reste extensible puisqu'elle offre une description abstraite de ces concepts. Elle permet aussi d'intégrer les services comme élément de contexte grâce à la description des services basée sur OWL-S.

Plusieurs autres approches ontologiques ont été aussi proposées telles que le système CoBrA ou l'approche CoOL (Context Ontology Language).

Plusieurs études effectuées concernant les méthodes de modélisation du contexte présentent une comparaison intéressante tel que: (Strang and Linnhoff-Popien 2004, Henricksen and Indulska 2006), (Perera, Zaslavsky et al. 2014), (Strang and Linnhoff-Popien 2004, Henricksen and Indulska 2006).

1.6 Adaptation selon le contexte

1.6.1 Notion de l'adaptation

Selon le grand dictionnaire, l'adaptation est définie comme étant : « l'opération qui consiste à apporter des modifications à un logiciel ou à un système informatique dans le but d'assurer ses fonctions et, si possible, d'améliorer ses performances, dans un environnement d'utilisation » [grand-dictionnaire].

Dans le domaine de la sensibilité au contexte, l'adaptation est présentée généralement par la capacité d'un système à changer son comportement lui-même pour améliorer ses performances ou de poursuivre son fonctionnement dans des environnements différents.

Layaïda, donne la définition suivante : « Par adaptabilité, on entend les moyens automatiques ou semi-automatiques qui permettent aux contenus [...] d'être utilisables sur des terminaux ayant des caractéristiques et des ressources très variées. » (Lemlouma and Layaïda 2003, Miraoui 2014).

Tandis que selon M.Satyanarayanan, «L'adaptation est nécessaire quand il y a une disparité significative entre l'offre et la demande d'une ressource» (Satyanarayanan 2001).

Essentiellement, il y a quatre genres d'adaptation à savoir : (i) l'adaptation de contenu, (ii) l'adaptation du comportement (service), (iii) l'adaptation de présentation (ou interface) et (iv) l'adaptation de composants logiciels. Nous nous sommes limités à présenter les approches d'adaptation de services selon le contexte.

La notion d'adaptation de services a été évoquée et a sollicité l'intérêt des chercheurs depuis longtemps (Pejovic and Musolesi 2015). L'adaptation reste encore un intérêt d'actualité dans le domaine de l'informatique diffuse. Plusieurs approches d'adaptation étaient présentées, soit de façon manuelle, semi-automatique ou automatique.

On parle d'adaptation de service lorsqu'il y a déclenchement d'un service ou changement de sa forme suite à la production d'un événement dans l'environnement de l'équipement informatique.

Les méthodes d'adaptation de services se basent principalement sur les structures conditionnelles, c'est-à-dire une fois qu'une condition devient vraie, le système doit changer de comportement de la manière convenable. L'inconvénient de ces structures est leur aspect statique, qui ne prend pas en considération la nature dynamique d'un SID dû à la mobilité des utilisateurs et des équipements. Les approches d'adaptation de service existantes dans les systèmes sensibles au contexte demeurent superficielles et ne traitent pas le problème en profondeur. Elles n'arrivent pas encore à établir une forte conjonction entre l'adaptation de service et la sensibilité au contexte.

1.6.2 Les mécanismes d'adaptation de services

1.6.2.1 Adaptation à base de règles

L'adaptation à base de règle consiste à définir les différentes conditions qui régissent et produisent les services sous forme logique en utilisant un langage de description approprié. Dans une approche d'adaptation à base de règles, les développeurs sont contraints de prévoir toutes les situations et les scénarios possibles durant la programmation des règles et avant que le système ne soit fonctionnel, sachant que le service ne peut pas fonctionner correctement en dehors de ces prévisions (Cremene, Riveill et al. 2006, Zaidenberg, Boulay et al. 2012). Si un service doit s'adapter à de nouvelles situations qui n'ont pas été identifiées durant la conception et l'implémentation, il est obligatoire de définir des nouvelles règles et de reproduire le service, ce qui n'est pas évident dans tous les cas (Cremene, Riveill et al. 2006, Zaidenberg, Boulay et al. 2012).

Les systèmes à base de règles ont sollicité beaucoup d'intérêt, mais leur utilisation comme modèles de prévision était limitée à cause du manque des techniques statistiques permettant de dériver ces règles à partir des données des activités (Perera, Zaslavsky et al. 2014).

L'enjeu pour les systèmes à base de règles est à la fois dans la description du contexte : les situations de l'utilisateur et de son environnement, et dans leur aptitude à maintenir plus de flexibilité et d'ouverture pour le développement futur des systèmes.

1.6.2.2 Adaptation à base d'apprentissage automatique

L'adaptation à base d'apprentissage automatique est fondée sur la prise de décision à partir de l'expérience. L'apprentissage par renforcement est une des approches d'apprentissage automatique qui a été souvent utilisée. Cette approche a pour but de concevoir des systèmes autonomes et adaptatifs.

O. Bridiczka et al. ont adopté l'apprentissage supervisé dans leurs travaux. Le modèle des conditions est appris et classifié par l'expert, puis l'utilisateur donne un retour en indiquant le service correspondant durant chaque situation (Dietze, Gugliotta et al. 2009, Miraoui 2014).

Plus tard S. Zaidenberg, (Preuveneers, Victor et al. 2009, Zaidenberg, Boulay et al. 2012) ont utilisé un algorithme d'apprentissage par renforcement pour développer un système qui détermine le modèle d'adaptation à partir des interactions du système avec l'utilisateur et de l'observation de ses réactions aux réponses du système.

Cette approche soutient l'amélioration dans l'interaction de l'utilisateur avec son environnement. Mais ce mécanisme reste limité puisqu'il n'assure pas la continuité des services si le modèle d'apprentissage affronte une situation pour laquelle il n'a pas été entraîné. Le temps réel et l'entraînement du modèle restent des défis primordiaux pour ces systèmes.

1.6.2.3 Adaptation à base de comparaison

L'adaptation à base de comparaison est moins utilisée que l'adaptation à base de règles ou à base d'apprentissage. La méthode consiste à comparer les conditions liées au déclenchement d'un service avec les données de la situation en cours. Ces conditions sont définies antérieurement durant la conception et la description des services. Le service qui correspond le mieux à la situation courante sera alors sélectionné et exécuté.

S. Dietze et al., (Dietze, Gugliotta et al. 2009, Deployment 2012) ont utilisé des mesures de similarité, pour développer un algorithme basé sur une mise en correspondance des descriptions des services avec les besoins utilisateurs pour fournir l'adaptation nécessaire.

Une approche similaire, utilisée par D.Preuveneers et al, (Preuveneers and Berbers 2008, Preuveneers, Victor et al. 2009) se base sur une adaptation par comparaison des profils définis dans l'application avec le profil de l'utilisateur. Une fois le profil sélectionné, les règles d'adaptation déjà déterminées dans le profil seront exécutées.

Adoptant cette approche, la description sémantique des conditions et des services est fondamentale pour le bon fonctionnement du système. Vu que ces descriptions sont prédéfinies par le développeur, la continuation des services n'est alors assurée que pour les cas prédéfinis.

1.6.2.4 Adaptation à base de fouille de données

L'approche consiste à explorer les données d'utilisation des services en utilisant les techniques de fouille de données pour reconnaître des situations éventuellement utiles ou des prédispositions intéressantes dans les données pour la réclamation future des services.

Cette approche a été utilisée par Edwin et al, (Benazzouz 2011, Deployment 2012) pour améliorer l'usage d'un appareil domotique. Ils proposent d'automatiser certaines opérations en se basant sur les manipulations régulières dans l'historique des données enregistrées de la situation d'utilisation d'un appareil.

L'inconvénient de ce mécanisme est qu'il ne s'applique qu'à un seul dispositif et il n'est pas capable de tenir compte d'autres données de l'environnement et les états des autres appareils ou dispositifs.

1.6.3 Les défis

La génération de services adaptatifs soutenant des environnements dynamiques et des besoins utilisateurs variables, sollicite la résolution de différents défis.

Un des défis primordiaux est la prise en compte du contexte. L'ensemble des situations de l'utilisateur et de son environnement sont décrites par le "contexte". Ceci rend obligatoire sa prise en compte dans l'interaction de l'utilisateur avec le système. Une des questions de recherche qui reste ouverte est de trouver le moyen de percevoir ce contexte. Les solutions actuelles se basent sur l'utilisation de divers capteurs pour saisir les données sur l'environnement et l'utilisateur ce qui génère des données hétérogènes provenant de sources variées. Il en résulte que les situations nécessaires engendrant une adaptation des services peuvent être complexes à cerner et à modéliser.

Un autre défi est l'extraction de connaissances comme étant l'aptitude de reconnaître, d'identifier, de concevoir les connaissances utiles pour le processus d'adaptation et de présenter des solutions efficaces pour que ces connaissances soient exploitées en temps réel, spécifiquement dans un environnement très dynamique tel que les environnements diffus.

Les études (Pejovic and Musolesi 2015) (Gwizdka 2000, da Rosa, Barbosa et al. 2016), présentent ce défi comme étant une création d'intelligence : analyse des informations du contexte, déduction de la signification et l'intégration de cette connaissance dans un mécanisme actif pour l'adaptation des services. Trouver des solutions fonctionnant en temps réel est aussi un autre point qui sollicite l'intérêt des chercheurs dans ce domaine.

Finalement, un autre défi est de pouvoir offrir des modèles d'architecture ouverts et flexibles.

1.7 La prédiction du contexte

1.7.1 La notion de prédiction

"Un système est sensible au contexte s'il utilise le contexte pour fournir des informations pertinentes et/ou des services à l'utilisateur, où la pertinence dépend de la tâche de

l'utilisateur " (Dey 2001, Held, Buchholz et al. 2002). En plus, on peut dire qu'un système est sensible au contexte s'il peut utiliser le contexte pour son profit. La sensibilité au contexte est l'un des principes fondamentaux de l'informatique omniprésente. Raisonner sur le contexte est le processus d'obtention de nouvelles informations à partir du contexte. La prédiction du contexte, par conséquent, signifie simplement la possibilité de prédire l'avenir des informations de contexte pour pouvoir fournir un service proactif et des systèmes autonomes et dynamiques. La sensibilité au contexte et la prévision de contexte représentent relativement de nouveaux domaines.

Cependant, il existe déjà des méthodes développées pour raisonner sur le contexte et le prédire. Dans le chapitre suivant (Chapitre.2) nous présenterons les différentes méthodes déjà utilisées pour la prédiction du contexte avec des études analytiques des différentes techniques et problématiques de recherches liées à ce concept.

1.8 Conclusion

Dans ce chapitre nous avons présenté les concepts de base de notre domaine de recherche, on s'est attardé sur la notion du contexte : les définitions du contexte existant dans la littérature ainsi que les techniques de modélisation du contexte. On rediscutera plus tard (Chapitre.3) la notion de contexte pour approfondir notre vision déjà soulignée dans ce chapitre.

Nous avons aussi fait un survol des approches antérieures de l'adaptation des services dans les systèmes diffus et leurs limites. Nous avons aussi détaillé les défis pour mettre en place une adaptation proactive et on a souligné l'intérêt de l'adaptation de la prédiction du contexte pour offrir une adaptation proactive.

La revue de littérature était essentielle pour comprendre les notions de base : le contexte, la sensibilité au contexte, l'adaptation et la proactivité.

CHAPTIRE 2

A SURVEY OF PREDICTION APPROACH IN PERVASIVE COMPUTING

Darine Ameyed¹, Moeiz Miraoui², Chakib Tadj¹

¹MMS Laboratory, Université du Québec, École de technologie supérieure
1100, rue Notre-Dame Ouest, Montréal, Quebec, Canada, H3C 1K3

²CMMomputer Science Department in Umm Al-Qura University, Saudi Arabia

This paper is published in the International Journal of Scientific & Engineering Research, Volume 6, Issue 4, April-2015 ISSN 2229-5518.

Résumé

La sensibilité au contexte est l'un des principes fondamentaux qui sous-tendent l'informatique ubiquitaire. La prédiction de contexte, une nouvelle tendance dans l'informatique ubiquitaire, est un sujet de la recherche avec beaucoup de défis et d'opportunités pour l'innovation. Ce travail présente et analyse le développement dans ce domaine et compare les différentes techniques et approches de prédiction de contexte.

Mots clés : prédiction; adaptation proactive; système sensible au contexte ; système diffus.

Abstract

Context awareness is one of the fundamental principles underpinning pervasive computing. Context prediction, a new trend in pervasive computing, is an open-ended research topic with a lot of challenges and opportunities for innovation. This work presents and analyses the development in this area and compares different context prediction techniques and approaches.

Keywords: prediction, proactive adaptation, context-aware system, pervasive computing, ubiquitous system, adaptative service.

2.1 Introduction

In a pervasive system, we must provide service adaptation, depending on the system's environment and the user's profile in order to offer an adequate service. In the literature, adaptation is also called plasticity (Calvary, Coutaz et al. 2002). It determines the adaptation degree of a given application to new requirements and new situations. While using ubiquitous systems, users might perform tasks that are part of their daily schedule. They might use various systems (smart phone, tablet, personal digital assistant (PDA), personal computer (PC), mobile or otherwise) depending on the nature of their activities, their preferences and other considerations. Users might make use of several modalities (keyboard, mouse, screen ...) to input data, depending on their profiles and the nature of the physical environment.

Furthermore, a ubiquitous system must provide the user with services well adapted to the overall context. Indeed services will be triggered dynamically and without an explicit user intervention in a proactive way. Making use of the context in applications, is a current area of research known as "context-awareness"(Abowd, Dey et al. 1999). A sensitive-context application must perceive the context of the users and their environment and adapt its behavior accordingly.

Most of the work on service adaptation in context-awareness, is focused on the current context. However, a new trend in this research field, takes into account the evolution of the context through time, in order to predict a future context (Mayrhofer 2005); (Nurmi, Martin et al. 2005) (Boytssov and Zaslavsky 2011). Therefore, the future context allows the pervasive system to choose the most effective strategies to achieve its goals, and makes it possible to provide an active and fast adaptation to future situations. This raises our interest in context prediction and its integration in proactive service adaptation. In this paper, we present a survey of prediction methods, used in proactive adaptation for pervasive systems. We give an overview of the available research in the field; we analyze, and discuss these approaches against a set of criteria.

The remainder of the paper is structured as follows:

- first, we give a short background and introduce some basic concepts;
- second, we present the various methods illustrated in some projects and cited in the literature, before analyzing them against our evaluation criteria matrix;
- finally we present a summary with the different challenges yet to overcome in the field.

2.2 Proactive computing

2.2.1 Proactive Context-Aware Applications

So far, most of the context-aware systems are merely reactive, adopting decisions based only on the current context. Research in anticipatory and proactive systems, including prediction of future situations is still in its beginnings. In these systems, a user is linked to a profile with a set of defining characteristics. Leveraging that profile, allows us to improve the systems' efficiency by anticipating the user's needs and adapting the services accordingly. Several approaches have been proposed to achieve this anticipatory aspect (Nurmi, Martin et al. 2005). These approaches can be classified under two categories: passive or active :

a. Passive context-awareness

The passive context-awareness system is a reactive system that requires the user to be reactive. The system presents the new or updated context, as a recommendation, to an interested user. The adaptation is completed upon the user's reaction. System decisions are up to the user, who agrees or not on implementing the recommended adaptation (Dey and Abowd 2000); (Miraoui 2009); (Le Mouël and Laforest 2006).

b. Active context-awareness

An active context-awareness system is a proactive system. It automatically adapts to discovered contexts and changes the application behavior accordingly. Adaptation is fully automated and rely on the system itself to learn or acquire information allowing the improvement of its performance with experience (Meiners, Zaplata et al. 2010).

2.2.2 Proactive Adaptation

In a system that supports context-awareness, adaptation is generally presented as the ability of the system to alter its behavior on its own, in order to improve its performance or pursue ongoing tasks, in different environments. Various types of adaptation can be differentiated: (i) content-based adaptation, (ii) behavior-based adaptation, (iii) presentation-based adaptation (interface), (iv) software components adaptation.

Adaptation has been mentioned in several occurrences of research work, and garnered interest among researchers for some time: (Fayad and Cline 1996); (Yarvis, Reiher et al. 1999); (Narayanan, Flinn et al. 2000); (South, Lenaghan et al. 2000); (Efstratiou, Cheverst et al. 2001);(Aksit and Choukair 2003); (Keeney and Cahill 2003). Adaptation is still of interest in the pervasive computing field.

Service adaptation occurs when a change in the system's environment triggers a special event. Control of adaptation can be, manual from the user, automatic or generic automatic.

a. Manual: the user decides the system's behavior based on given recommendations.

b. Automatic: the adaptation routine is implemented within the system. This routine, is mainly based on a static set of conditional structures. This means, when a given input hits one of the expected states implemented within the conditional structures, the system must change its behavior appropriately. This also means that an unexpected input will go unnoticed by the system. And no change in behavior is triggered.

c. **Generic automatic:** the adaptation routine is generic. Which means the system can respond to unexpected events and refine and improve its behavior over time.

2.2.3 Proactive system challenge

Adaptive services, supporting dynamic environments, raise different challenges. The environment and the set of situations, a user might evolve through are described in the context. This context specifies the interaction between user and system. A question in this respect, still open, is how to find means of perceiving this context. The currently proposed solutions rely on the use of various sensors to capture environmental and user-related data. This generates heterogeneous data from a variety of sources. And makes services adaptation more and more complex to identify and model.

Another challenge is the extraction of knowledge such as the ability to recognize, identify, design knowledge relevant to the adaptation process. This challenge can be presented as a creation of intelligence: analysis of context information, deduction of meaning and construction of knowledge.

System adaptability entails fault tolerance, recovery from failure and also the ability to discover new resources and devise new strategies to respond to unexpected input.

The availability of user data through all the processes underlying context awareness raises some concerns about the user privacy and brings another challenge in this respect.

Finding operating solutions in real time are also another point which biases the interest of researchers in the field. Finally, another is to offer architectural models open and flexible.

These challenges are among the main research topics in ubiquitous computing. And this is where prediction comes into focus since it addresses some of them. In this paper we will survey relevant work in this respect.

"A system is context-aware, if it uses context to provide pertinent information and / or services to the user, where relevance depends on the user's task" (Abowd, Dey et al. 1999). Also, we might define context-awareness of a system as its ability to make use of the context to improve its overall functioning. Context-awareness is one of the fundamentals of ubiquitous computing. Reasoning about the context is the process of acquiring new information from the context. Therefore, prediction of the context means the ability to predict the future context information in order to provide proactive service. Later in this section, we shall present the most frequently used methods to predict the context.

2.3 Context prediction

2.3.1 Context prediction methods

▪ Sequence Prediction Approach

This method stems from the prediction task, as defined in theoretical computer science. The general idea is, a problem could be described as sequence of events or states received in the time interval from 1 to t. The goal reduces then, to predicting the next set of events or the next state (Figure.1).

Davison and Hirsh were amongst the first researchers to investigate the usefulness of sequence-based prediction. They used it to predict commands in a UNIX shell (Davison and Hirsh 1997); (Davison and Hirsh 1998). This work inspired research in activity prediction, including research in pervasive computing and smart environment (Bhattacharya and Das 1999); (Das, Cook et al. 2002); (Gopalratnam and Cook 2004); (Mayrhofer 2005). Starting from the LZ87 algorithm (Ziv and Lempel 1978) Bhattacharya and Das developed different context prediction algorithms.

They modified the original LZ87 algorithm to make it predict the next users' location in a cellular network, yielding the LeZi-Update approach. Later Das et al used this updated algorithm for the smart home environment (Das, Cook et al. 2002). LeZi-Updated inspired Cook et al. to develop the Active LeZi-Updated algorithm to predict the user's actions in a

smart home. In their Project MavHome, Cook et al built a smart home management system acting on users' behalf and support them in different tasks. The system evaluates the next users' action and triggers a set of actions to fulfill the underlying tasks (e.g opening store, starting the lawn sprinklers) (Gopalratnam and Cook 2004).

Mayrhofer used Active LeZi to develop a general context prediction approach, in order to proactively assist users. The prediction process comes in 5 steps: (i) collect context information from a heterogeneous sensor network. (ii) extract features. (iii) classify. (iv) label, and (v) predict. As shown in Figure.2.1 The process takes as input a times series (observation sets, each recorded at a specific time) and provides as output the user's current context and the expected future context.

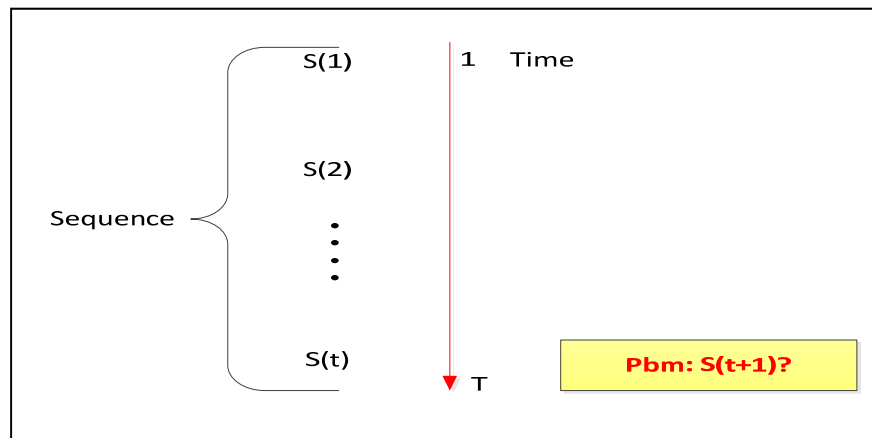


Figure 2. 1 Sequence prediction approach

The context acquisition step concerns itself with gathering contextual information from physical and logical sensors. The captured data is then transformed into more meaningful features during the extraction step for better interpretation. The classification step then, recognizes recurring patterns, called clusters in the feature space, built in the previous step. Classification uses a feature vector which can be assigned to multiple clusters with different degrees of belonging. This degree represents the probability that a feature vector belongs to a cluster. Labeling allocates the descriptive names for individual clusters or combinations of

several clusters. Based on the observed history, the future context cluster can be predicted. The prediction step uses the vector of clusters generated by the classification step. The final goal is to generate a cluster of vectors for the future, which matches the current cluster provided by the classification step. This classification is used for supplying the cluster vectors predicted in the labeling step to provide context label, intended for use in dynamic applications.

The sequence prediction proved to be an effective approach for context prediction. As shown previously, several researchers have contributed theoretical improvements to adjust it to the context prediction problem. It should also be noted, that sequence prediction has several shortcomings. Among those mention:

- the reliability of prediction can be verified but the reliability of the observed data cannot be taken into consideration;
- abstracting the whole context into a simple sequence might result in the loss of information;
- this approach does not deal with time: it cannot detect time-dependent patterns;
- this approach can only be applied if the context is described as a flow of events.

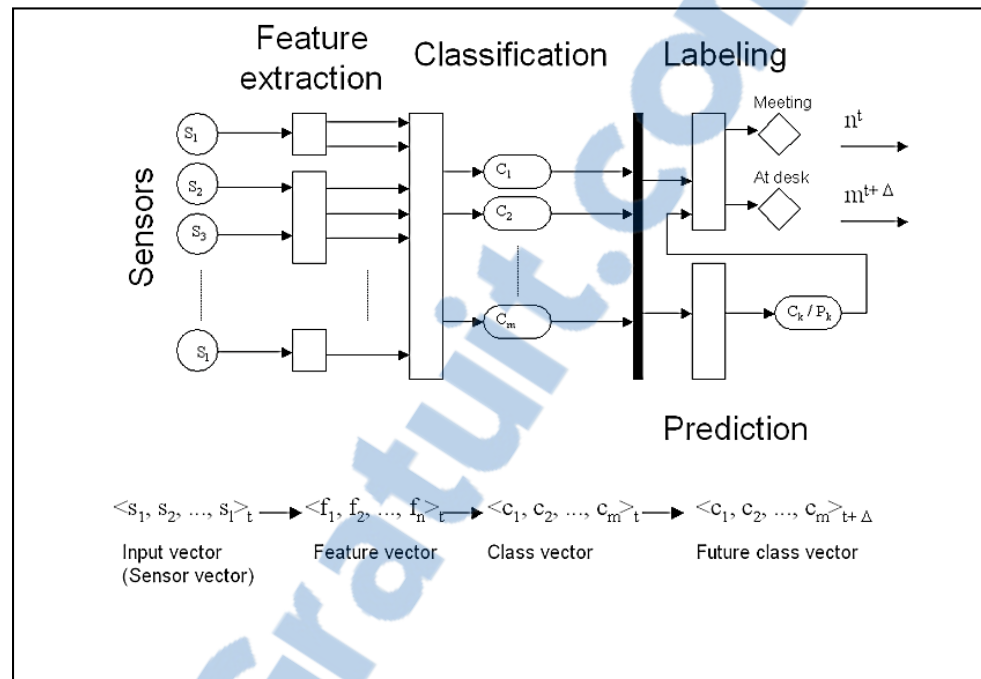


Figure 2.2 Architecture for proactivity via predicted user contexts
Taken from Mayrhofer (2005)

▪ Markov Chains

A Markov chain is a set of states. The process starts in one of these states and moves incrementally from one state to another. Each move is called a step. If the chain is currently in state X_i , then it moves to state X_j at the next step with a probability denoted by p_{ij} , and this probability does not depend on the previous states.

$\{X_n, n > 0\}$ X_n is the state being processed at time n

$p_{ij}(n)$ is called transition probability from state i to state j at time n .

$$P[X_{n+1} = j \mid X_n = i_n, \dots, X_0 = i_0] = P[X_{n+1} = j \mid X_n = i_n] = p_{ij}(n)$$

Prediction based on Markov chains is fairly widespread. Some projects used this technique for resolving context prediction problems. For example, Chen (Liu and Chen 1998) used Markov chain to predict a person's movement. Zukerman (Zukerman, Albrecht et al. 1999) used them to predict the next user's request in a web context.

In the work of Kaowthumrong et al (Kaowthumrong, Lebsack et al. 2002), the authors tackled the problem of active devices determination. The proposed solution was to provide the user with the next device to be used, when there are several devices in the vicinity of user. Based on a Markov property, the next device depended on the current device only. The context prediction process was built on a Markov chain representing the order of devices access. Krumm used discrete time Markov chain for route prediction during driving (Krumm 2008).

Despite some limitations of Markov chains, (limited number of states over a discrete timeline), they are a viable solution, easy to implement. But more complicated use cases need extension of the Markov chain. One of the most famous of these, is the Hidden Markov Chain (HMM) introduced by Baum and Petrie in 1966. This model can be used with unobserved (hidden) states.

Figure.3, outlines the basic layout an HMM. A circle maps to a random variable that might contain any sequence of values. $x(t)$ represents the hidden state as a function of the time t . $y(t)$ represents the observation as a function of the time t . The arrows in the diagram (often called a trellis diagram) denote conditional dependencies.

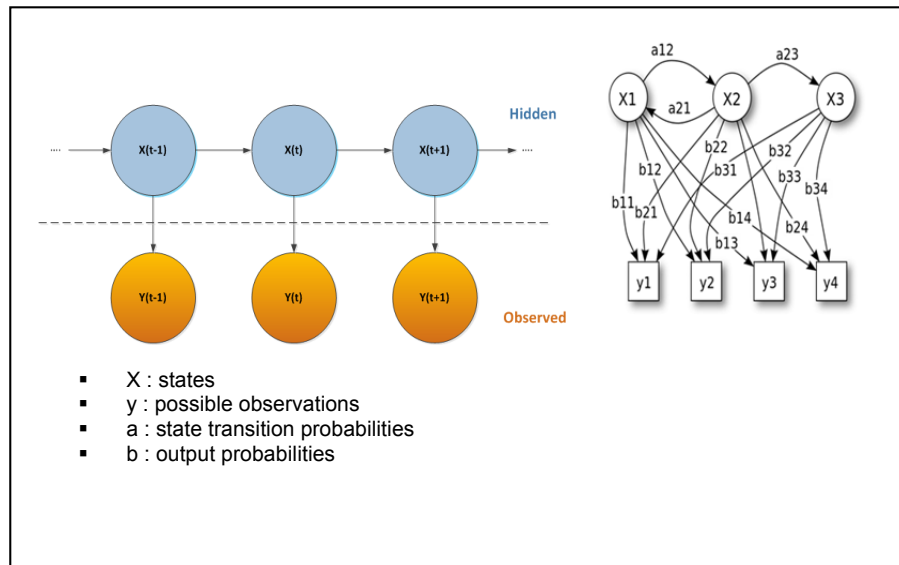


Figure 2.3 General architecture of an HMM

From the diagram, the conditional probability distribution of the hidden variable $x(t)$ at time t , given the values of the hidden variable x at all times, depends only on the value of the hidden variable $x(t - 1)$: the values at time $t - 2$ and before have no influence. This is called the Markov property. Similarly, the value of the observed variable $y(t)$ only depends on the value of the hidden variable $x(t)$ (both at time t).

The HMM was used in some projects for context prediction. We mention Gellert and Vintan, (Gellert and Vintan 2006), who used the HMM to predict the next room that user want to visit. Simmons et al, used the HMM for itinerary prediction (Simmons, Browning et al. 2006).

One more extension of Markov chain is the Markov Decision Process (MDP), with the addition of actions (allowing choice) and rewards (giving motivation).

A MDP is a formal model that consists of the following (Figure 2.4).

S is a finite set of states

A is a finite set of actions (alternatively, A_s is the finite set of actions available from state S)

- **Bayesian Network**

A Bayesian network (BN) is a probabilistic graphical model (a type of statistical models) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). The structure of a DAG is defined by two sets: the set of nodes (vertices) and the set of directed edges. The nodes represent random variables and are drawn as circles labeled by the variable names. The edges represent direct dependence among the variables and are drawn by arrows between nodes. Every node is associated with a fact and every directed edge represents the influence on one fact by another.

A Bayesian network specifies a joint distribution in a structured form. It represents dependence/independence via a directed graph composed with (i) Nodes (random variables) and (ii) Edges (direct dependence). There two components in a Bayesian network, the graph structure (conditional independence assumptions) and the numerical probabilities (for each variable given its parents), which in general is: $p(X_1, X_2, \dots, X_N) = \prod p(X_i | \text{parents}(X_i))$.

A Dynamic Bayesian Network (DBN) is a special Bayesian Network which relates variables to each other over adjacent time steps. A DBN represents all variables at two (or more) points in time. Edges are drawn from the variables at the earlier time to those at the later time. For example, in the DBN on the above figure (Figure.6), we see that A at time t influences C, at time $(t+1)$, B influences for A and C.

Several researchers work on how to make use of DBN in context prediction, like Albrecht's (Albrecht, Zukerman et al. 1998). The system, they developed, used DBN to predict further user actions in an adventure game. With a similar method, Horvits et al, built a system to predict the goal of software users and run services proactively (Horvitz, Breese et al. 1998). Another example is how Petzold used the dynamic Bayesian network to predict a person's movement inside some defined space (Petzold, Pietzowski et al. 2005). Petzold et al, represented the context as a DBA, where the current location depends on several locations previously visited and also the time spent at each location. This model has provided high prediction accuracy (around 70% to 90 %).

The BN approach can be seen as a generalization over the Markov models. A BN cannot model a feedback loop because by definition it cannot have loops, or cycles, in its simple forms. But this limit can be overcome with DBN. Indeed, the DBN offers more flexibility but requires more training data.

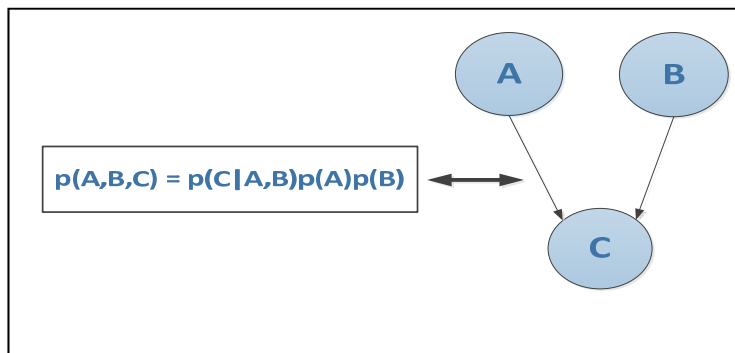


Figure 2.5 Example of Simple Bayesian Network

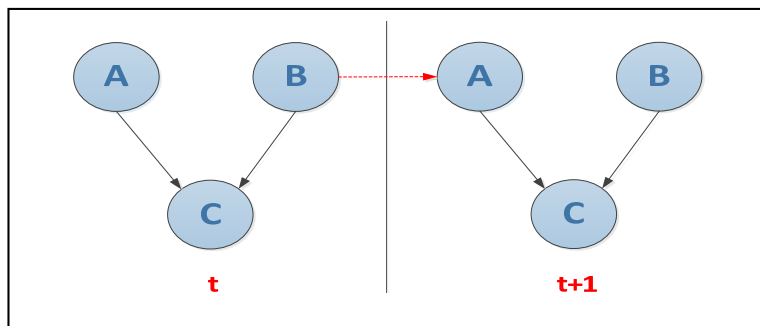


Figure 2.6 Example of Dynamic Bayesian Network

- **Neural Networks**

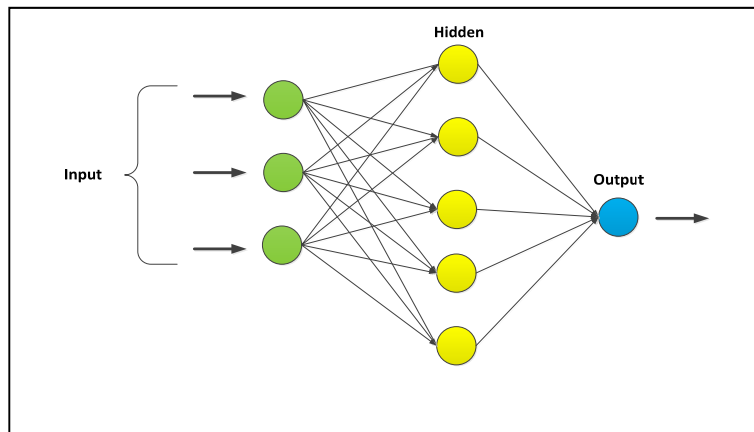


Figure 2.7 Simplified structure of an artificial neural network

Neural networks are inspired from their biological counterparts. They are very popular in machine learning. And they implement formal mathematical models emulating the structure of biological neurons. Each node represents an artificial neuron and each arrow represents a connection from the output of one neuron to the input of another (Figure 2.7).

This approach has become one of the very popular ones, to solve several problems pertaining to artificial intelligence such as: pattern recognition, association models, approximation theory, etc.

Neural networks have been used, first, to predict context through the work of Mozer, (Mozer 1998). This work described a smart home with various services, such as space occupation models, estimates of hot water consumption rates. Vintan et al used this same approach to predict the location of an individual, the places he's most likely to visit, in a given environment (Vintan, Gellert et al. 2004).

The work of Al-Masri (Al-Masri and Mahmoud 2006) also resulted in a model of neural networks to provide mobile services. Recently, Lin (Petzold, Bagci et al. 2003) used neural networks for a smart intracellular handoff of work.

There are several types of neural networks. Approaches based on them are flexible and efficient.

However these approaches act like a black box, have no tolerance for data missing, are slow in their training phase (require long periods of time).



▪ **Branch Prediction Method**

This approach, predicts the flow of instructions in a microprocessor, after a branching command is received. The purpose of the branch predictor is to improve the flow in the instruction pipeline. Branch predictors play a critical role in achieving high effective performance in many modern pipelined microprocessor architectures. To predict whether a branch is taken or not-taken, the branch prediction unit implements a hardware algorithm. There is a variety of branch prediction algorithms:

Static prediction: the idea is to make a distinction between the different types of branches with greater probability of being taken and those which will never or almost never be taken. The simplest static prediction algorithm makes a distinction between conditional branching and unconditional branches: an unconditional branch is always taken; conditional branches are never taken.

Saturating counter: it consists in keeping track of the number of times that branch has been taken, and then comparing this number with the total number of branch executions. Thus an average can be estimated to see if the branch is usually taken or not taken. This storage is based on counters, integrated into the branch prediction unit. There is a counter per branch stored in the branch target buffer. This counter is changed each time a branch is taken or not-

taken: if it is taken, the counter is incremented (unless it is already at its maximum value); if it is not, its value is decremented.

Two-level adaptive predictor: if there are three ‘if’ statements in a code, the third ‘if’ statement might be taken depending upon whether the previous two were taken/not-taken. In such scenarios, a two-level adaptive predictor works more efficiently than saturation counters. Conditional jumps that are taken every second time or have some other regularly recurring patterns, are not predicted well by the saturating counter. A two-level adaptive predictor remembers the history of the last n occurrences of the branch and uses one saturating counter for each of the possible 2^n history patterns. The general rule for a two-level adaptive predictor with an n -bit history is that it can predict any repetitive sequence with any period if all n -bit sub-sequences are different.

Global branch prediction: Global branch predictors make use of the fact that the behavior of many branches is strongly correlated with the history of other recently taken branches. Those ways they can keep a single shift register updated with the recent history of every branch executed, and use this value as an index in a table of bimodal counters.

Neural branch prediction: The conditional branch prediction is a Machine learning problem, where machine learns to predict the outcome of conditional branches. Artificial neural network is very efficient to learn, to recognize and classify patterns. Similarly the branch predictor also uses the classification of branches and tries to keep correlated branches together.

Some context prediction systems used Branch Prediction Algorithms. Starting from this approach Petzold (Petzold, Bagci et al. 2003) provided a context prediction algorithm for a ubiquitous information system, in order to discover the daily movement pattern of an individual in a finite environment such as a house or an office. This system used several kinds of predictors; a counter predictor, a two-level-adaptive predictor and a global predictor.

This approach is simple and fast but it can detect only a very simple range of behaviour patterns.

▪ **Trajectory Prolongation Approach**

Some context prediction approaches, handle the context data vector, from a multidimensional space perspective. Then the underlying algorithm, computes an approximation or an interpolation with the points given through the data vector, with some related function in order to predict the next set of events.

This approach has been used in the work of Karbassi and Barth (Karbassi and Barth 2003), which attempted to estimate the arrival time of a vehicle going through a traffic jam. The system built a linear regression between the traffic jam factor and the arrival time. Then, it computed the parameters from the vehicle's history. Also the work of Mayrhofer (Mayrhofer 2005); Sigg (Sigg, Haseloff et al. 2006) suggested the use of an autoregressive model to predict the context. In Sigg's system the prediction model was based on an alignment method for predicting the most probable continuance of a time series from the observed sequence of suffix. This alignment prediction approach seeks to align two series using their differences so that the number of positions corresponding to the two series is maximized. Finally, Sigg (Sigg, Haseloff et al. 2010) offered a continuous learning module in order to adapt to changing environments or user habits, through continuous monitoring of a time series recorded in the context's history and updates of the registered patterns.

The trajectory prolongation approach is an efficient approach for location prediction. But it has significant shortcomings: it can't handle non numerical data, and has no tolerance for missing data.

▪ **Expert system**

An expert system is a tool, able to reproduce the cognitive processes of an expert in a particular field. Expert systems are designed to solve complex problems by reasoning about knowledge, represented primarily as if-then rules. An expert system is divided into two sub-systems: (1) the inference engine and (2) the knowledge base. The knowledge base represents

facts and rules. The inference engine applies the rules to the known facts to deduce new facts (Figure 2.8).

This approach was based on expert systems and rule engines. It has been used in some implementation of context prediction. The end goal is, to build the prediction rules. This method allows a clear view of the system. An example is the work of Vainio, which presented a proactive context adaptation system for smart home environments. The system controls the state of an adaptable context (e.g: lighting, temperature...) according to the user's routine. The system adapts the context under the user's supervision. The system is based on fuzzy logic techniques, allowing it to control the environment in uncertain situations.

Another example is the work of (Hong, Suh et al. 2009). They developed an expert system that infers rules in the execution's environment to find user preferences and provide related proactive services, based on their respective context history. The main challenge in this work is to deduce the user preferences for the current state from past states, and then figure out the corresponding service.

Context prediction based on expert systems is a promising approach allowing quick and efficient integration of previous knowledge. It also allows an easy integration of the adaptation action. This approach can ensure learning and self-correction.

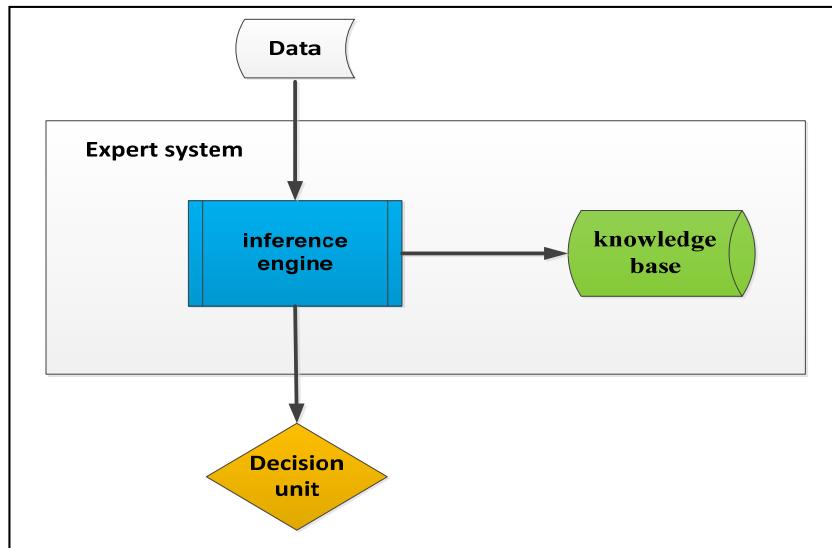


Figure 2.8 Expert system structure

- **Context space theory**

Contextual spaces theory is an approach developed, in order to best define context-awareness and to deal with sensor problems that create uncertainty and incur a lack of reliability. This theory uses spatial metaphors to represent the context as a multidimensional space (Figure 2.9).

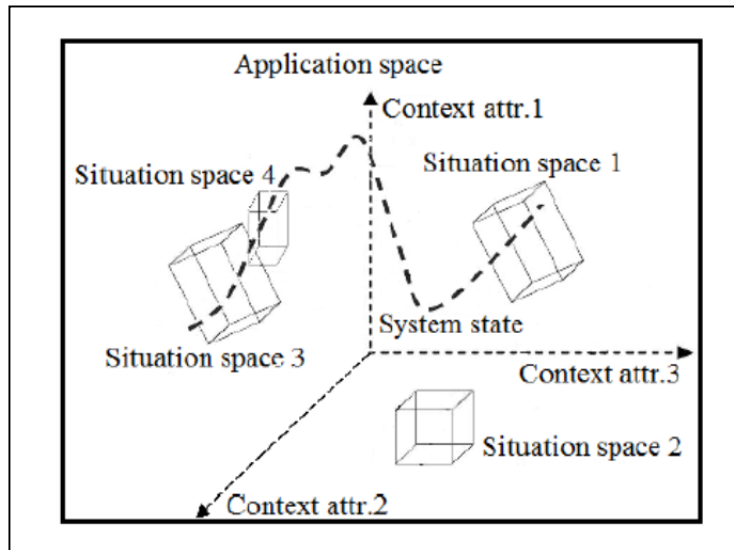


Figure 2.9 Context spaces theory
Taken from Boytsov (2011)

It is designed to make context-awareness clearer. In the space theory, any kind of data is used to reason about the context, is called a context attribute. Attributes are either directly measured by sensors or calculated from other context attributes. They may have a numeric value or a non-numeric value. A context state is represented by the entire relevant context at a specified point in time. All possible context states, define the application space. This allows seeing the application space as a multidimensional space, where the number of dimensions is equivalent to the number of context attributes in the context state.

The theory of context space, initially submitted by Padovitz and Zaslavsky (Padovitz, Loke et al. 2004). The authors attempt to provide a general model to help thinking and describing the context and develop context-aware applications. This work will be later, the basis for several researches of Zaslavzky and Boytsov (Boytsov, Zaslavsky et al. 2009); (Boytsov and Zaslavsky 2010); (Boytsov 2011). Boytsov and Zaslavsky present the CALCHAS system, which offers context prediction, used extension to the context space theory to provide proactive adaptation. The authors see proactive adaptation as a reinforcement to learning

tasks that aim at improving both the prediction as well as the adaptation decision (Boytsov 2011).

This approach addresses the context prediction problem in a general sense.

▪ **Data-mining algorithm**

Data mining or the Knowledge Discovery in Databases (KDD), an interdisciplinary subfield of computer science, is the computational process of discovering patterns in large data sets, involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining is to extract information from a data set and transform it into an understandable structure for further use. Data mining involves six common classes of tasks:

- anomaly detection: the identification of unusual data records,
- association rule learning: searches for relationships between variables,
- clustering: is the task of discovering structures in the data that are similar, without using known structures in the data,
- classification: is the task of generalizing known structures to apply to new data,
- regression: attempts to find a function which models the data with the least error,
- summarization: provides a more compact representation of the data set, including visualization and report generation.

Several methods are used for knowledge discovery in a large amount of data. Clustering is one of the most used methods. It uses unsupervised learning algorithms by similarity measures to group data elements with similar patterns (in the sense of these similarities) in clusters. In a context discovery work, Y. Benazzouz, (Benazzouz 2011) presented a method based on clustering to measure common situations, according to a context clustering approach in the MIDAS project. He adopted a bottom-up discovery of common situations

and identification of unusual situations to provide a proactive adaptation service according to user behavior. In this approach the situation to discover, is defined beforehand.

This approach requires a database archiving user history. Data mining requires a first phase of collection to get enough data which will be processed later. In fact, that generates a significant amount of data to store and manage.

- **Similarity and semantic similarity**

This approach uses existing relations between context values in order to deduce the emerging context. The relationships in a context are of two types: relationships between context values and relation between multiple context values. The relationship between context values, identify two entities belonging to different ontologies based on the similarity between them. This similarity can be between two atomic properties (if they have the same name and the same value of property), or between two classes or two non-atomic properties (if they have the same name or any property defined within one, is similar to a property in the other). Based on this similarity, four relationships can be identified between two context values, namely: intersection, complement, equivalence and independence. The relationships between multiple context values are described using an Entity-Relationship (ER) model. For each connection between two values, the values are converted into units of the model and the types of relationships relating nodes in the model.

As for the generation of search criteria, this approach defines generic rules in order to deduce the user's requests from the ER model. Then, it extracts the search criteria (key words) describing the user's requests, to find services to recommend. Xiao (Xiao, Zou et al. 2010) developed a method to dynamically determine a context model in order to recommend services. This model supported various contextual types and values, and then recommends services using this information. The authors made use of relations of similarity between the context values to find potential services the user might need.

In her work on services prediction, Salma Najar, offered a mechanism of discovery and prediction guided both by the context and user intent [48]. She used semantic similarity techniques. The system is based on the implementation of a matching algorithm, which computes the matching degree between the intention and the current context of the user and the set of semantic services described accordingly. OWL- SIC (OWL-S Intentional & Contextual) is an extension of OWL-S (Web Ontology Language-Semantic, is ontology, within the OWL-based framework of the Semantic).

The similarity approach requires historical data, to select and recommend services that are not always available. In fact, it requires a first phase of collection to get enough data which will be processed thereafter. Intentional approach provided by Najar (Najar, Pinheiro et al. 2014) is a user-centered approach, but can generate conflict: technically, a problem of interoperability between services. Indeed, two compatible intentions do not necessarily make two technically compatible services.

A number of prediction algorithms have been developed for many projects and in different domains in recent years. In this section we have reviewed and detailed approaches and research projects related to context prediction. In the next section, we summarize in a comparative matrix, the different approaches we studied.

2.4 Comparative analysis and evaluation

Based on the formal presentation of the context prediction task (Boytsov 2011), we can formalize a prediction task as a follow:

$$P_r = G (S_1, S_2, \dots, S_t)$$

- S_t : the context data at a certain moment in time.
- P_r : a prediction result
- G : operator of context prediction operation

A context predictor in the most general sense can be viewed as follows: (Figure 2.10).

To evaluate a prediction approach, we established criteria related to data (e.g: type) and criteria related to reasoning (e.g: runtime, accuracy, ...), based on our analysis of the approaches developed in the previous section. These criteria seem to be the most relevant to evaluate and compare these methods.

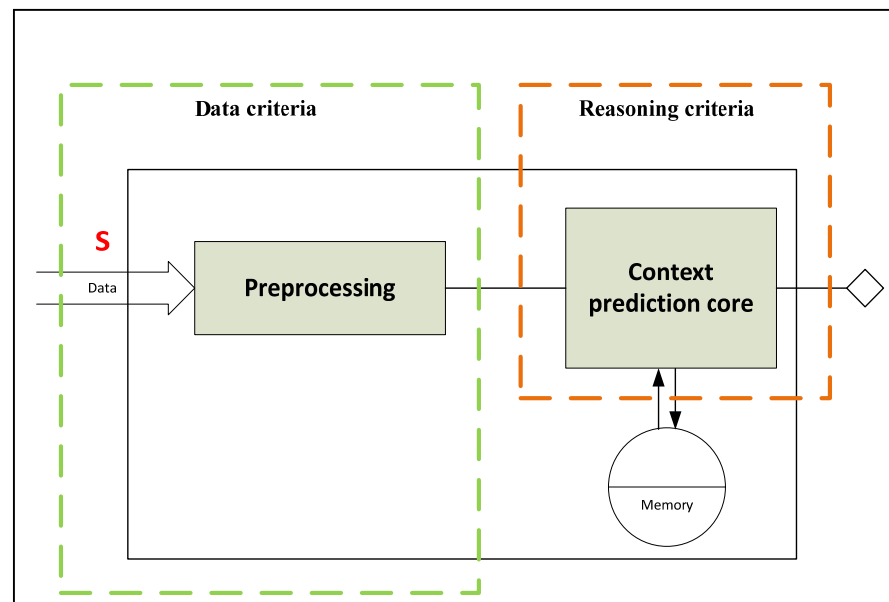


Figure 2.10 General structure of context prediction
Inspired from Boytsov (2011)

2.4.1 Reasoning evaluation criteria

We start with the reasoning criteria: we established some criteria addressing efficiency, accuracy, time reasoning and scalability:

Accuracy: to assess the overall reliability and precision of the approach. Some projects were tested against real data to evaluate the accuracy of their approach, and are summarized in (Figure 2.11). Using the Lezi algorithm (Gopalratnam and Cook 2004), testing with real data. They obtained prediction rates nearing 47%. Using MM, (Kaowthumrong, Lebsack et al. 2002) the prediction accuracy was around 70-80%. Using HMM [29] prediction accuracy reached 98%. Using DBN, Petzold (Petzold, Pietzowski et al. 2005) had a prediction rating floating between 70% and 90%. In S.Najar’s work, the system was based on the implementation of a matching algorithm. The prediction algorithm had a good quality result that neared 60% (Najar, Pinheiro et al. 2014).

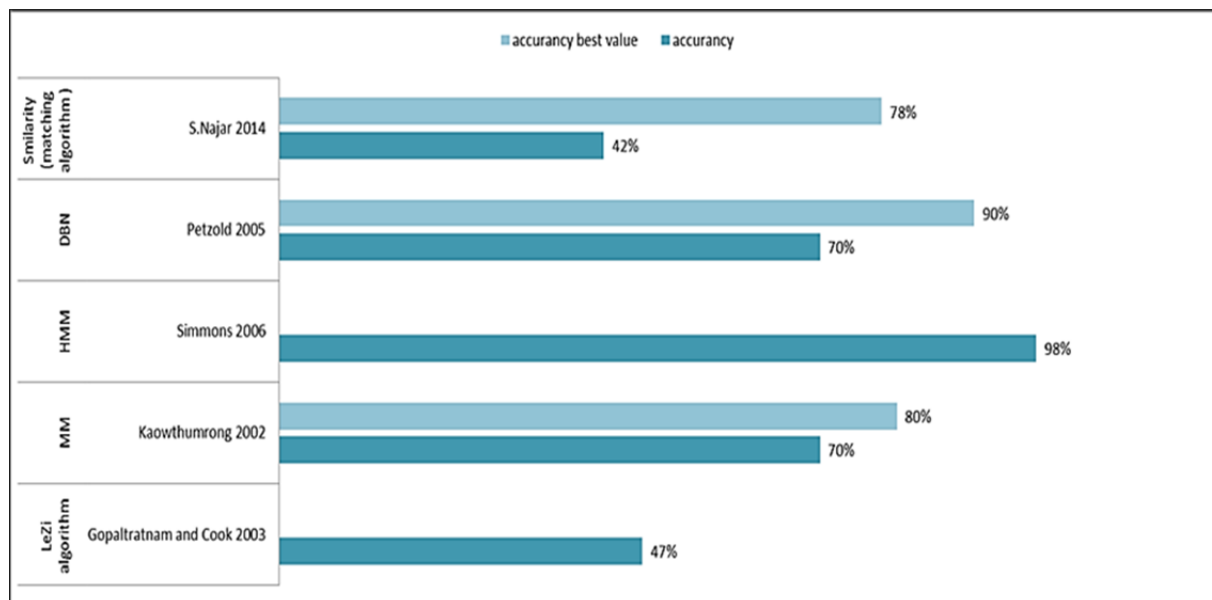


Figure 2.11 Comparative accuracy diagram

In our general evaluation of different approaches, we estimate the general accuracy as ranging from low/worst performance to high/best performance, depending on the capacity of the approach to be effective in a ubiquitous environment, taking into account some other criteria: e.g the loss of data, running time, etc.

Tolerance of missing values: dealing with missing data is a typical problem of preprocessing. Several machine learning methods implemented their own strategies for

dealing with missing data. Even if some approaches are more sensitive to data loss. For example in similarity approach the similarity measure can be easily distorted by error in data value.

Evolvability: defined as the capacity of an approach for adaptive evolution and providing a scalable system.

Speed of runtime: indicates the processing time. These criteria evolved the overall time of the procedure; training time and real-time reasoning. Some methods were faster in reasoning time but needed a long training time e.g NN method.

Observability: if the method is a white box, its parameters show in a rather perspicacious way, how the system works. Looking at the system parameters, we can know exactly what the system has learned. And we can have some preliminary estimation; thus it becomes possible to configure the system accordingly. Otherwise if the method is a black box, it remains capable of prediction, but an expert cannot see the underlying reasons for the current decision, even if the expert knows all system parameters.

Prior knowledge inference: evaluates whether the approach can incorporate prior knowledge inference. If the method cannot do this, it could lead to low efficiency at the start. To overcome the problem we can pre-train the predictor and use generated training data before the system starts.

2.4.2 Data evaluation criteria

Data loss in preprocessing: the loss of information is dependent on both the prediction process and the data capturing the system. Preprocessing contextual information can causes a data loss that could influence the prediction results.

Type: numeric/ no numeric: if some type of data will not be accepted by the prediction approach a preprocessing will be required.

Tableau 2.1 Comparative overview of prediction approaches

Approach	Reasoning criteria							Data criteria		
	Accuracy	Tolerance to missing value	Evolvability	Speed of run time	Observability	Prior knowledge inference	Data loss in preprocessing	Numeric data	No numeric data	
Sequence	*	*	*	*	<input type="checkbox"/>	✓	✓	×	✓	
Markov	**	**	***	**	<input checked="" type="checkbox"/>	✓	✓	×	✓	
Bayesian networks	***	****	****	****	<input type="checkbox"/>	✓	✓	✓	✓	
Neuronal networks	***	*	**	**	<input checked="" type="checkbox"/>	×	×	✓	×	
Branch method	**	***	***	**	<input type="checkbox"/>	✓	✓	×	✓	
Trajectory prolongation	**	*	*	****	<input checked="" type="checkbox"/>	×	×	✓	×	
Interpolation										
Approximation	**	*	*	**	<input type="checkbox"/>	✓	×	✓	×	
Expert system	***	**	****	***	<input type="checkbox"/>	✓	✓	×	✓	
Space theory	***	**	***		<input type="checkbox"/>	✓	✓	✓	✓	
Data mining algorithm	***	**	***	**	<input type="checkbox"/>	✓	✓	✓	✓	
Similarity and semantic similarity	***	**	****	***	<input type="checkbox"/>	✓	✓	✓	✓	

Performance	Observability	Confirmation
**** Best performance	<input type="checkbox"/> White box	No ×
* Worst performance	<input checked="" type="checkbox"/> Black box	Yes ✓

Legend

2.5 Challenges of context prediction

Context prediction is a relatively new problem in computer science. Ground work on finding solutions and approaches for context prediction is still in progress. This work still faces many challenges including:

1. Lack of general approaches, in much of what has been attempted so far, to tackle the context prediction. There have been only a few attempts to define and solve context prediction tasks with general approaches.
2. Lack of generic automatic adaptation: most of the work is focused on the prediction, while neglecting the effective integration of results in services adaptation. With the notable exception of a few solutions implementing Markov Decision Processes, most of the remaining propositions, rely on expert systems to process the outcome of a context-prediction, thus making these system, less adaptive or autonomous.
3. Interdependence between prediction and service adaptation: prediction systems process the context sequentially in a simple fashion. They predict, then pipe the result into the adaptation routine to act on it. This might be enough for simple problems, but will fall short for more complex problems. And as far as we could find, there is almost no attempt to explore this area.
4. Optimisation: using pervasive system in health care, as in as other high-risk endeavours. Prediction algorithm must be able to deliver prediction with High accuracy and minimal delays for computing.

We believe that once these challenges are resolved, we will be able to develop proactive and dynamic awareness-context systems that can handle more sophisticated use cases and situations, and improve their efficiency. Thus we might facilitate the overall use of such computer systems, and promote their integration within our daily activities.

2.6 Conclusion

In this paper we presented an overview of the literature on prediction approaches. We reviewed projects and research work related to context prediction. We established a set of comparative criteria we used for comparing and analyzing these approaches. We presented also several concepts that are required for context prediction. And we identified open issues for context prediction, such as many existing challenges.

CHAPTIRE 3

A SPATIOTEMPORAL CONTEXT DEFINITION FOR SERVICE ADAPTATION PREDICTION IN A PERVASIVE COMPUTING ENVIRONMENT

Darine Ameyed¹, Moeiz Miraoui², Chakib Tadj¹

¹MMS Laboratory, Université du Québec, École de technologie supérieure
1100, rue Notre-Dame Ouest, Montréal, Quebec, Canada, H3C 1K3

²CMMomputer Science Department in Umm Al-Qura University, Saudi Arabia

This paper is published in the International Journal of advanced studies in Computer Science and Engineering IJASCSE Volume 4, Issue 4, 2015

Résumé

Les systèmes pervasifs réfèrent à des systèmes sensibles au contexte qui peuvent détecter les changements dans leur environnements, et adapter leur comportement en conséquence pour fournir des services appropriés. L'adaptation dynamique de ces systèmes permet de modifier le service et le contexte en concordance avec la prédiction. Toutefois, les définitions existantes du contexte sont encore vagues et ne conviennent pas à une intégration de la prédiction. Dans cet article, nous discutons et procédons à la classification des définitions de contexte contribuées jusqu'à date. Ensuite, nous présentons une nouvelle définition qui permet aux systèmes pervasifs de comprendre et de prédire leurs contextes. Nous analysons les éléments essentiels qui entrent dans la définition du contexte, et proposons des scénarios à préciser notre approche.

Mots clés: Prédiction; sensible au contexte ; Systèmes diffus; Définition du contexte.

Abstract

Pervasive systems refers to context-aware systems that can sense their context, and adapt their behaviour accordingly to provide adaptable services. Proactive adaptation of such systems allows changing the service and the context based on prediction. However, the definition of the context is still vague and not suitable to prediction. In this paper we discuss and classify previous definitions of context. Then, we present a new definition which allows pervasive systems to understand and predict their contexts. We analyze the essential lines that fall within the context definition, and propose some scenarios to make it clear our approach.

Keywords: Prediction; Context-aware; Pervasive Computing; Context Definition.

3.1 Introduction

Since its introduction by Mark Weiser in 1991, the ubiquitous computing presented progressively a new vision of computer systems, where technology becomes invisible to the user, and allows for the proactive adaptation to changes in context of users and applications. From this vision evolved the current interest in context-awareness, supporting the approach of adapting context-aware services, which should be directly related or even deduced from the contextual information. Hence the notion of context-aware computing.

Implicitly, a context was defined as the set of information pertaining to an event or a particular situation in any environment or supervising communication between humans.

In computer science, the context is specified in a mobile, smart, and scalable environment. This environment provides the ubiquitous systems (Bouzeghoub, Caron et al. 2008), whose objective is to allow a transparent human-machine and / or machine-machine interaction.

In order to ensure this interaction, the pervasive system must opt for a context-aware adaptation. Defining and collecting of effective contextual information become crucial, and entails challenges that constantly increase with technological development emphasizing mobility for users, and equipment.

Several basic elements of research, in context-aware computing have been proposed in order to answer some key issues, including the context and context awareness (Dey 2001), (Abowd, Ebling et al. 2002). This might be summarized by questions such as: what's the context? How to make use of a context? And how to develop context-aware applications?

There are also many other researches that used the specific context information and the development of context aware applications. Such research had already led to a large number of prototypes and even end-user products (Chen and Kotz 2000), (Abdenneji, Lavirotte et al. 2012), (Benazzouz 2011), and (Madkour, El Ghanami et al. 2013).

Most of work on the adaptive services based on context has focused on the current context (i.e. the present context). However, a new trend in research on context-aware adaptive services approaches supports taking into account the temporal evolution of context to predict a future context (Mayrhofer 2005), (Benazzouz 2011), and (VanSyckel and Becker 2014).

Being aware of the future context allows pervasive computing system (PCS) to choose the most effective strategies to achieve its objectives and therefore to provide an active and fast adaptation to future situations.

The first step towards the development of a context-aware pervasive system is to understand the context and establish its components. This requires a clear description of what a context is. Both in the development of logical representations and in the processing, in order to use the context efficiently and easily in applications.

Many attempts have been made to provide a concise definition of context. However, the current definitions are still vague and general. Also all the proposed definitions do not take into account new adaptations based on the prediction strategies.

In this paper, we propose a new definition of context, taking into account the goals such as: predicting the context and context-aware adaptive services. We try to introduce the context in a model highlighting the spatiotemporal dynamics of context elements in the context-aware proactive adaptation. Understanding how to use the general context while designing adaptive application can help application designers to determine how this context must be implemented in their applications to meet the use requirement.

The rest of the paper is organized as follows: Section 3.2 provides an overview of context definitions in the literature. Section 3.3 presents our proposed definition of context. We will detail our approach highlighting the link between the definition and usefulness of contextual information. We will introduce our context definition and we will show how contextual elements are constructed based on the proposed definition. At the end of section 3.3, we will propose some scenarios to demonstrate the consistency of our definition. Section 3.4 concludes the paper with a discussion, our contributions and presents our future work.

3.2 Related work

So far, the context has been defined as the set of circumstances or facts that surround a particular event or situation, which is the essence of the definitions proposed in several researches (Chen and Kotz 2000), (Dey 2001), (Henricksen, Indulska et al. 2002), and (Najar, Saidani et al. 2009).

The next subsection enumerates the most significant trends we noticed in context definitions.

3.2.1 First trend: physical aspect

Most the first researches on the context, focused on a physical enumeration of contextual factors (element/ data). In their work Schilit et al (Schilit, Adams et al. 1994), have considered that the context has three important aspects related to the following questions: Where are you? With whom? What resources do you have nearby? They defined context as changes in the physical environment using: location, description of people and objects in the immediate environment and changes in these objects. Brown restricted the context to elements of the user environment, "the elements of the environment that the user's computer knows about" (Brown 1995). Later, Brown, Bovey and Chen, introduced the season, the temperature, the identity and location of the user to the context definition (Brown, Bovey et al. 1997). They presented a set of extensible components to characterize the context in which the basic elements are: the location, all items, the user needs, time, and spatial orientation.

While we agree with most of these definitions (in particular the physical aspect of context elements), we find they did not take into account logical and abstract data (e.g. time), resulting in changes in the system state and services.

3.2.2 Second trend: towards time aspect

Rayan defined the context element: "the elements of context are: user location, environment, identity and time." (Pascoe, Morse et al. 1997). Ward saw context as statements of probable application environments (Ward, Jones et al. 1997), "states the possible application environments". Pascoe described the context as a subset of physical and conceptual states of interest to a particular entity (Pascoe 1998). Schmidt defined the context as "knowledge about the user and the statements of equipment, environment and location." (Schmidt, Aidoo et al. 1999). While Brézillon "perceived it as everything not explicitly involved in solving a problem, but constrains it" (Brézillon and Pomerol 1999).

In these definitions, time is implicitly or vaguely defined. However, time is a crucial element of context. It is also an element that should be defined according to the purpose of the application. Based in purpose, time information may change. As we will show in the subsection: 3.1. and 3.3.

3.2.3 Third trend: adaptation-driven context definition

In their study of context awareness in mobile, Chen and Kotz showed that general context definitions remain vague and inadequate in a computing environment (Chen and Kotz 2000). They defined the context as the set of environmental states and parameters that determine the behavior of an application or in which an event of the application runs and is of interest to the user. Thus research on context definition was oriented towards adaptation.

Dey emphasized the relevance of information by providing a definition where he attempts to clarify the nature of the entities of the context: "Context is any information that can be used to characterise 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." (Bouzeghoub, Caron et al. 2008). This definition encapsulates all the above definitions since it is very generic. Dey reported that the context parameters can be implicit or explicit. Indeed, the context parameters may be provided by the user or by sensors located in the environment of the user and the application. They can also be reached by a more

or less complex interpretation of these parameters. Dey gave a definition covering any implicit or explicit data that can be critical to the application. In 2001, Winograd approved the definition given by Dey and argued it covers all existing work on context. However, he considers that the expressions used by Dey such as "any information" and "characterize an entity" very general and not marking any limit to the concept of context (anything could be a context). In order to bring more precision to the definition of Dey, Winograd presented context as a set of structured and shared information. He explained this definition by stating "the consideration of information as context is due to how it is used and its inherent properties." He supported his idea by the following example "The voltage on the power lines is context if there is some action by the user and/or computer whose interpretation is dependent on it, but otherwise is just part of the environment." (Winograd 2001). In 2002, Henricksen defined context as "the circumstance or situation in which a computing task takes place." (Henricksen, Indulska et al. 2002).

An analysis by Brezillon (Brézillon, Borges et al. 2004) conducted on the definitions of context led to the conclusion that most definitions are answers to the following questions: Who? To identify the current user and others in the environment; what? Where? When? Why? How? To describe how the activity takes place. Answering these questions can generate a large set of information, most of it is unnecessary in addition to management problems of this information and storage in the context of mobile systems. In 2005, Truong gave a new definition of context based on Dey's, but put more emphasis on the role of context in interaction. He considered the context as a set of directions and information that define the interaction between the user and the application (Schilit, Adams et al. 1994).

In a more abstract version without detailing the components of the context, Miraoui described the context as "Any information that trigger a service or change the quality of a service if its value changes" (Miraoui 2009). He explained, "This definition is a compromise between the abstraction of the term context and the limitation of the set of contextual information related to services provided (goal of pervasive computing). It does not enumerate examples of context which makes it more generic and independent of the application. It does not take into account

information that might characterize context but do not play a relevant part in service adaptation (Miraoui 2009).

According to Gensel the context is the set of characteristics of the physical or virtual environment that affects the behavior of an application and whose representation and acquisition are essential for the adaptation of information and services (Viana, Miron et al. 2011). For Najar, the context acts as an external element to the computer system that affects its internal variability, and as a parameter guiding the variant and the process of adaptation (Najar, Saidani et al. 2009).

Some of those definitions remain vague and general such as Dey's definition, or neglected the time information, which is important in the prediction, or provided information that is not necessary for the application.

3.2.4 Towards adaptation and prediction trends

Up to now, most of the work on context-aware adaption services focused on the current context. Recently, a new trend supported the idea of taking into account the temporal evolution of context to predict a future context. A noteworthy example is the work proposed by (Mayrhofer 2005); (Najar, Pinheiro et al. 2014), (Kirsch-Pinheiro and Rychkova 2013).

While these researchers have demonstrated the importance of the use of prediction and future context, they did not elaborate the context definition or the purpose of its use. Although, they all had to revisit the old definitions, trying to identify the contextual information they would use in their modeling during the service adaptation or prediction according to user behavior and as part of a specific application. As a result we did not find a specific definition adapted to prediction, especially in a general and reusable form.

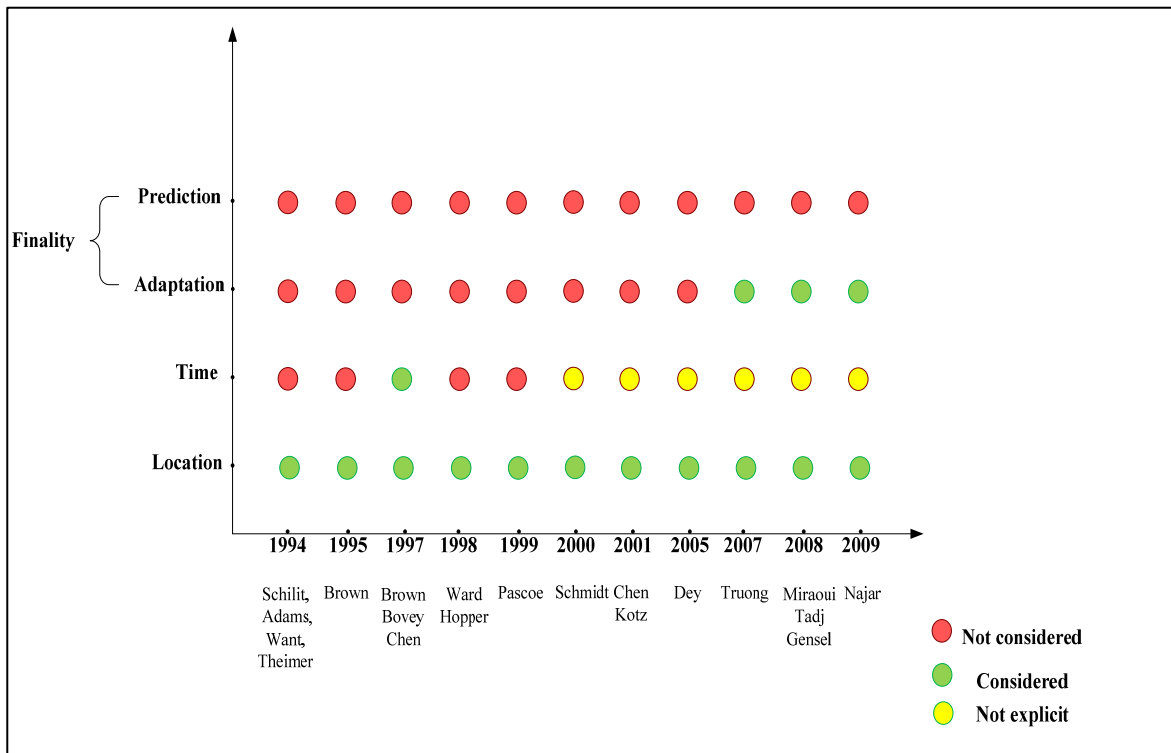


Figure 3.1 Comparative graphical of context definitions

3.2.5 Synthesis

Definitions of context remain general, vague and do not reflect the new trend in service adaptation based on prediction or context discovery.

All the recent work showed the importance of being aware of the future context, which allows a pervasive system to choose the most effective strategies to achieve its objectives and therefore makes it possible to provide an active and fast adaptation to future situations.

We conclude with a comparative diagram shown in Figure 3.1, summarizing the overall trends of the context definitions while putting forward the presence of the criteria that we consider crucial in defining a context: location, time and purpose. We are therefore interested in modeling the context in such a way that it takes into account the prediction or/and adaptation.

3.3 Proposed definition

In the following, we identify three essential lines that fall within the definition of context. These lines stem from our own approach to define the context.

3.3.1 Link context and time

The context must be defined in accordance with the space-time domain. This observation is consistent with Ryan's point of view with respect to the importance of time in the characterization of the context (Pascoe, Morse et al. 1997). In fact, any situation, event or interaction should be performed within a mandatory interval of time. The granularity of this interval is crucial since it presents the temporal extension of the information describing the context. This fact has also an impact on the quantity of information: one can pass from a small number of information entities to a database, depending on whether one focuses on an event of short duration or a sequence of events (activity).

We also strengthen our approach with the affirmation of Jensen (Jensen and Snodgrass 1999). He distinguishes two aspects of time that are very specific in the description of temporal data:

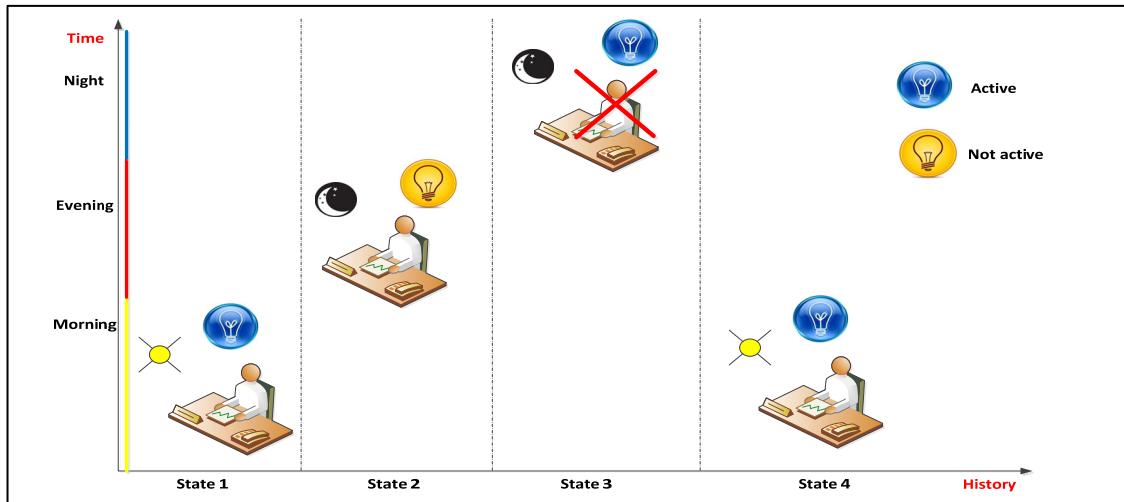


Figure 3.2 Conceptual diagram in the context of space-time

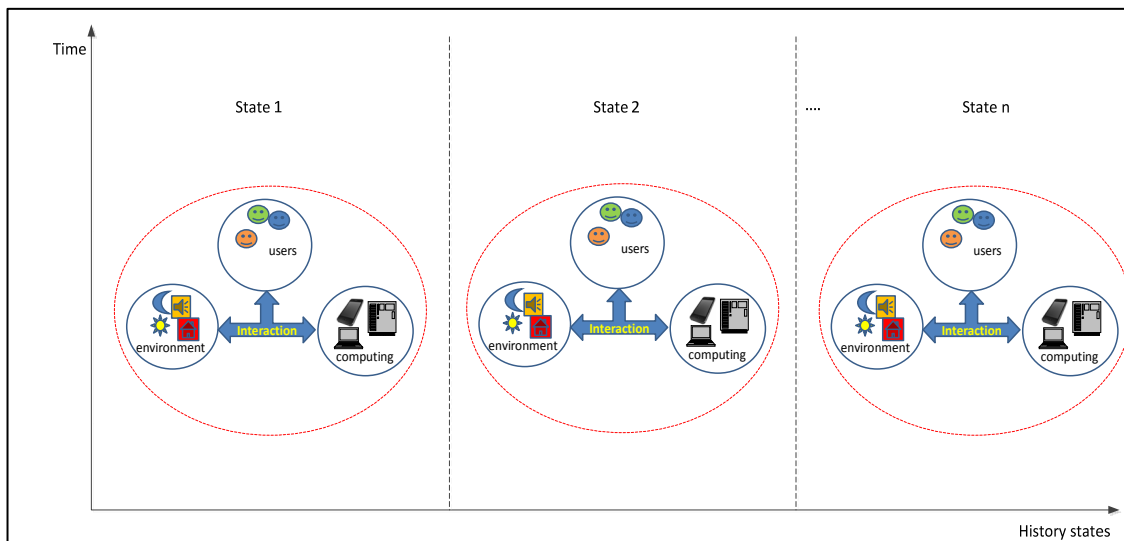


Figure 3.3 Example smart Luminosity control in an office according to user presence and time.

- the valid time (the time, the phenomenon modeled in reality was produced) and
- the transaction time (the time, data on this phenomenon was introduced in the database).

We conclude then that if the time isn't defined during the production of the event or introduced as an explicit element of the context, we would have estimation failures in time during prediction, which affect the service quality.

Therefore, any entity (user, physical or logical environment component, computer component) used in the interaction around an event in a pervasive system must be considered in relation to a time axis describing its dynamics. This helps us recognize a prior state (past), a current state (present) or a future state, to be estimated following a temporal logic, as shown in Figure 3.2 and in Figure 3.3.

3.3.2 Link context and space

In the literature, the spatial context is reduced, in the best case, to a variable location (user location). As stated by M. Georgia et al "We can identify the places located in the relative user environment." (Kapitsaki, Prezerakos et al. 2009). We can rather find the concept of "context usage" that defines and includes any information characterizing the user, the environment and the platform (Petit 2010).

Li (Li 2007), presented a definition of spatial context (or geographical context) in three dimensions: (1) the static spatial context, (2) the dynamic spatial context and (3) the internal spatial context.

He defined them as follows:

- The static spatial context: is geo-spatial information that may have an impact on the environment of the user. For example information about the road network, buildings, shops, interior structures (rooms, offices, flats ...).

- Dynamic spatial context: gathers geo-referenced information collected by different types of sensors. For example, the speed data collected on the roads, the data on the occupancy rate of parking lots, etc.
- The internal spatial context: gathers information available from local devices (e.g. GPS). For example, a motorist may have information about his position, speed, direction, destination with his GPS. He may also obtain information from an integrated sensor such as a vehicle's fuel gauge.

3.3.3 Relevance of context according to the purpose of use

O.Bucur (Bucur, Beaune et al. 2005) deepened the characterization and formulation of the relevance of context information by integrating the concept of finality context (define contextual elements according to the purpose of their use). The finality refers to the purpose for which the context is used at a given time. This purpose can be short or long term.

If the purpose is set to long term, the context is archived over time while maintaining relevance. For example, if a system is designed to adapt the user interface according to usage patterns, then the data of past uses continue to be relevant because they can determine the next adaptations and predict the future actions of the user. If purpose is short-term (instantaneous), the current context will be used immediately in the process of adaptation. As an example, the transition to the silent mode of the user's mobile phone upon entering the library.

We conclude, these questions regarding: time, spatial context, and purpose, still present a major challenge in the design of systems, applications and context-aware services. We discuss this in the following section with more details.

3.3.4 Proposed context definition

Our definition promotes following axes: (1) presenting the time as a most important feature of context, (2) redefining the spatial dimension in the environmental integrity of the system and services in a usage context, (3) and incorporating the finality of the context in the definition of contextual information. This means that, the elements of context should be considered according to the objective for which the context is used. We notice then, that the prediction oriented definition is mandatory. Knowing the adaptation scenario based on the current context does not require the identification of the spatiotemporal or temporal information, neither the finality of the use of context. However, the attributes are crucial for prediction.

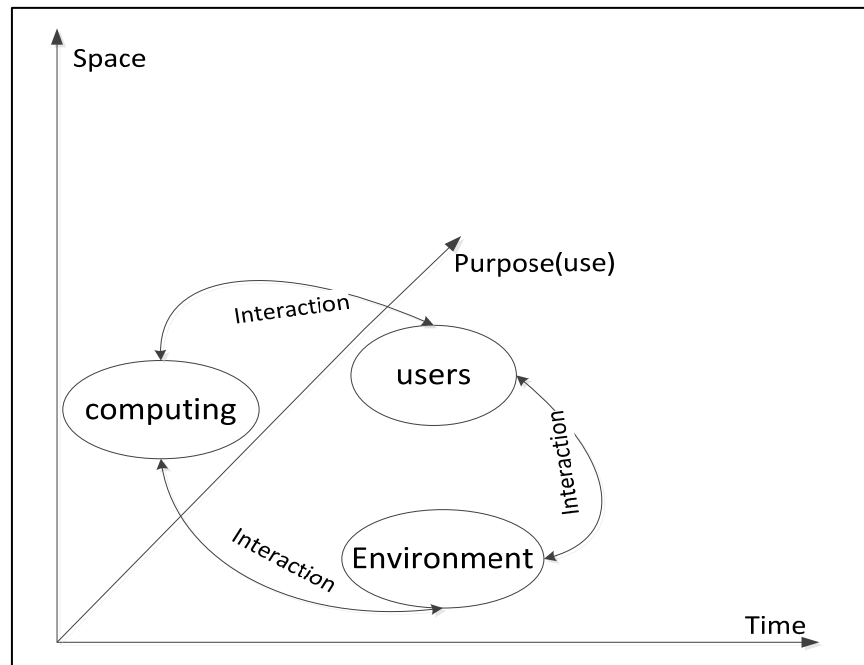


Figure 3.4 Position of a context according to the 3 axes (Time-space-purpose)

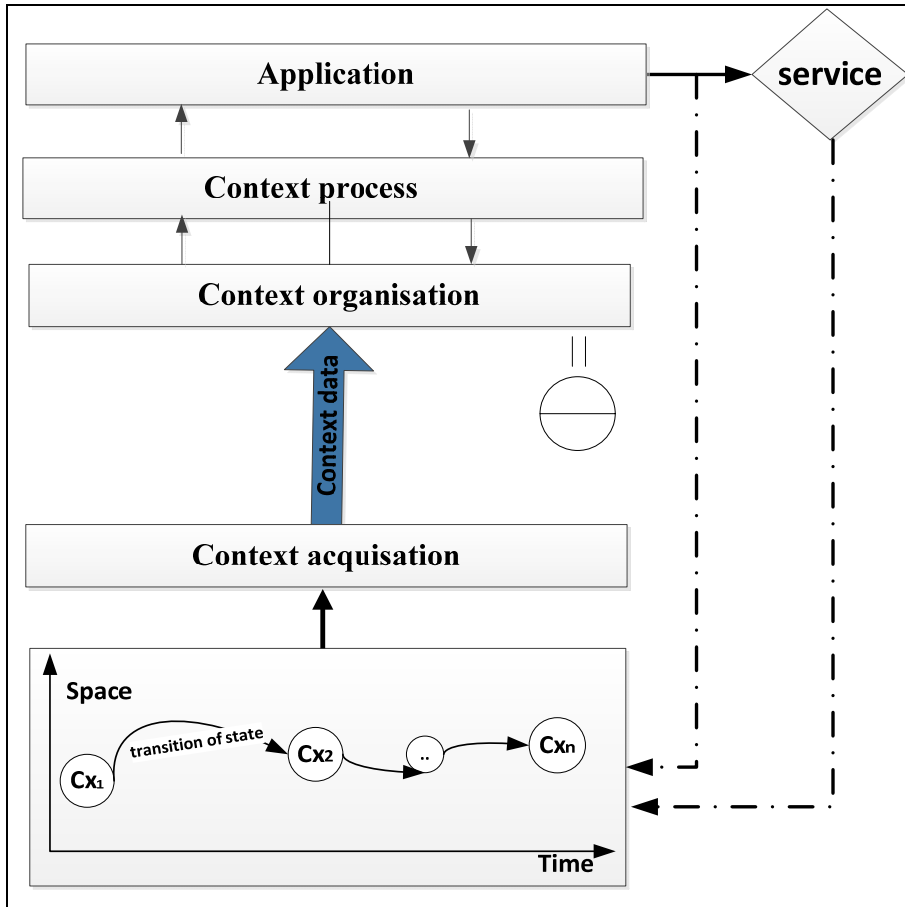


Figure 3.5 Context awareness operations

From our point of view, the context in a pervasive system is located around a spatiotemporal variation and is defined by the three axes mentioned above in Figure 3.4: (1) Time (2) Space, and (3) Purpose (finality) of its use. Therefore, we summarize our context as: **"Any entity undergoing a spatiotemporal variation and that may lead to a change in the service or the quality of service in the short or long term"**.

This definition provides a description of a context able to answer: Where? When? What for? And any other information for which the change of the value triggers a service or change the quality of a current or future service. Thereby, it can be useful in an habitual or exceptional system behavior (e.g. failure sensor, user habit change).

1. Where? As well as the geographical location, this also describes the spatial location of the use in a dynamic, static and internal dimension.
2. When? The most real time temporal data a real-temporal data with consideration of possible granularity of time measurement in a dynamic and proactive system such as in pervasive systems.
3. What for? To understand the finality behind the information as defined as an element of context and determine the purpose of the use of the contextual information for the short and long term.

As shown previously, the context is following a spatiotemporal dynamic, that needs to be taken into account, when acquiring contextual data (as mentioned in our definition). Once the data acquired, we organize it, according to its projected use to make it easily available to the relevant processes.

The context generated at the "process" step, which might be either the current context or the predicted context, will influence the application's behavior. Conversely, once the application updates itself after new adaptations triggered by underlying services, the context will in turn be updated.

We believe that this definition identifies the most relevant contextual information and is directly involved in the design of a context-aware system. It could be oriented adaptation and prediction (Figure 3.5).

3.3.5 Demonstration scenarios

The following scenarios are presented to show the consistency of the proposed definition.

Mobile Scenarios: Screen Luminosity Control System

A screen of a cell phone increases its luminosity when it is in a dark place and reduces when it is in lighted area, as shown in Figure 3.6.

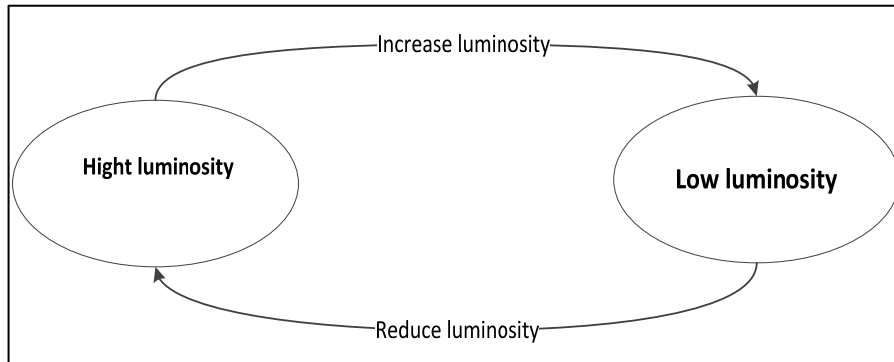



Figure 3.6 State diagram

In traditional scenario, the service will change depending on the information defined by a brightness sensor.

Applying previous definition oriented adaptation that gives the following description of service, service form, and context information.

- **Service:** luminosity adjustment of the screen.
- **Service form:** increased or reduce luminosity
- **Context information:** - Lighting

Tableau 3.1 Contextual information scenario (1)

Equipment	service	Form	Contextual information	Contextual information for prediction	Contextual information for adaptation
 Smart-phone	Luminosity Adjustment	Increase luminosity	Lighting	Discover context (prediction)	Selection context (adaptation)
			Location	Location	Lighting
		Reduce luminosity	Time	Time	
			Previous state	Previous state	

If brightness sensor fails, the context prediction can provide a solution to ensure the continuity of service.

Based on previous definitions we cannot have solution to make decision about service, and we cannot specify the information, according to the adaptation module or prediction.

Based on our definition, detecting contextual information will be easier, more accurate, and specific for each module (prediction, adaptation) as shown in Table.1 and following description (Figure 3.7).

- **Service:** luminosity adjustment of the screen.
- **Service form:** increased or reduce luminosity
- **Context information:** - Lighting - Location - Time - Previous state.

Home Automation Scenarios: Home Heating Control System

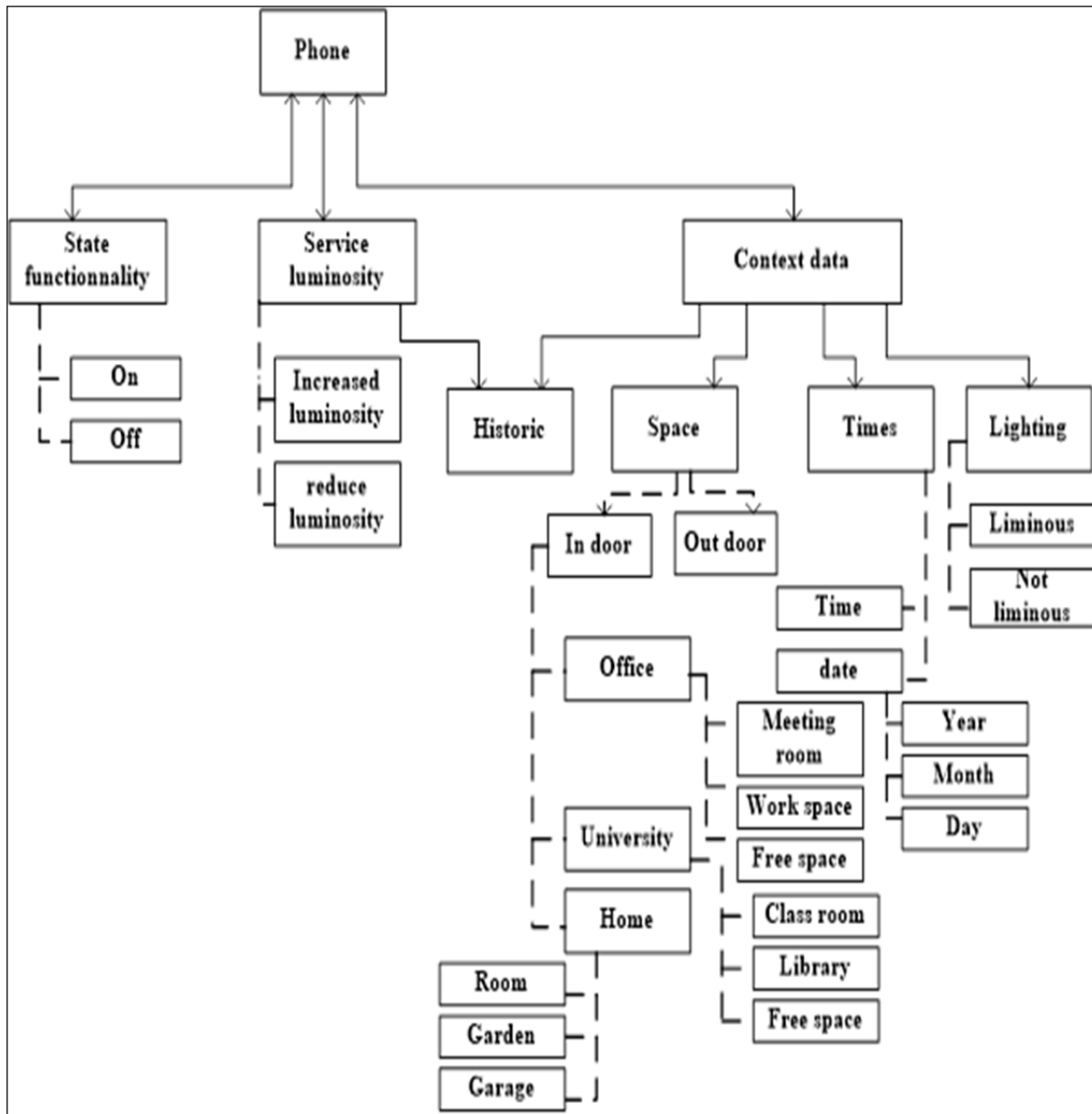


Figure 3.7 Model of luminosity adjustment service

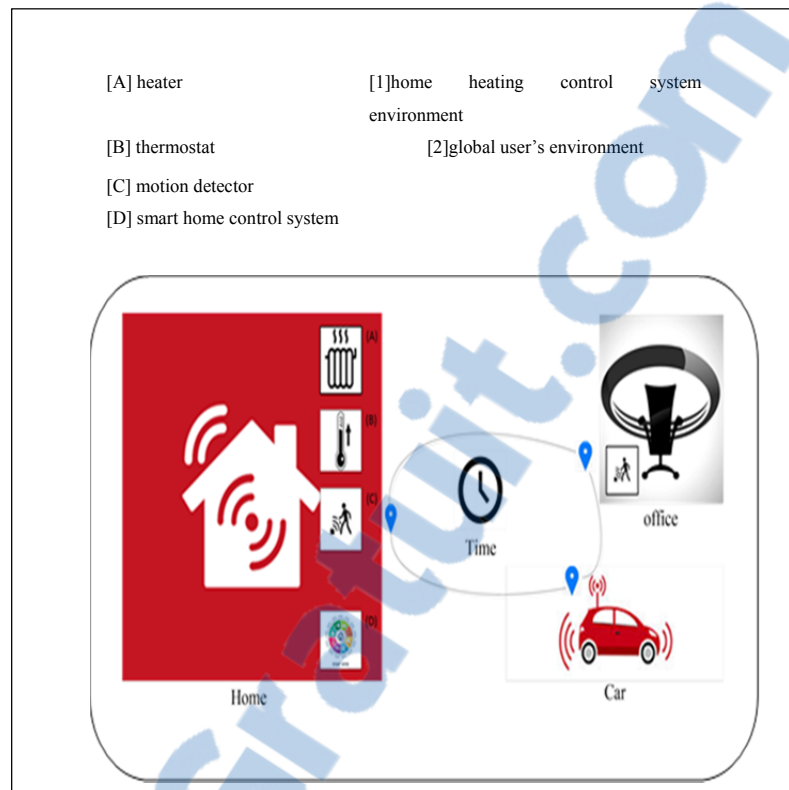


Figure 3.8 User's overall environment

For energy-saving heating should be turned off when the user leaves home in the morning for work. The user is primed to leave work and go home, and he wants to have his home in an ambient temperature when he returns there.

In the traditional adaptation process, the heating system is activated when it detects the user presence in the house.

Achieving the desired temperature takes a little time, (Figure 3.8: (1) home heating control system environment).

Whereas in an adaptation process based on prediction, the system starts preparing the home when the user is still on the way, to get the desired temperature by the time the user will just reach home. By estimating the time according to itinerary indicated by the user's car GPS, the

service system can trigger heating, sometime before the arrival of the user at home (Figure 3.8: (2) home heating control system environment).

Applying previous definition oriented adaptation that gives the following description of service, service form, and context information.

- **service:** heating temperature adjustment in house.
- **service form:** on or off heating .
- **context information:** User intern location.

Actual temperature.

Providing the service before physical presence of user in home, autonomously, proactively and invisibly remains impossible. Service will depend on a manual user command.

Based on our definition, detecting contextual information will be easier, and specific for each module (prediction, adaptation) as shown in table 3.2 and following description (Figure 3.9). It makes possible the autonomous heating service without manual user command or physical user presence.

- **service:** heating temperature adjustment in house.
- **service form:** on or off heating.
- **context information:** User intern location,

Actual temperature,

User external location,

Time.

The following table shows all the contextual information necessary in scenario (2) according to the purpose.

Tableau 3.2 Contextual information scenario (2)

Service	Contextual information	Contextual information for prediction	Contextual information for adaptation
Heating temperature adjustment	User intern location	Actual temperature	User intern location
	Actual temperature	User extern location	Actual temperature
	User extern location	Time	
	Time		

The overall conclusion, our definition offers an efficient approach to define contextual information in a pervasive system. It establishes clear parameters that define efficiently the context. It takes into account the dynamism of a pervasive system. It encompasses adaptation and prediction takes into account the purpose giving a more efficient design. And allows exceeding the previous definitions limits, to define useful contextual information for prediction and proactive adaptation.

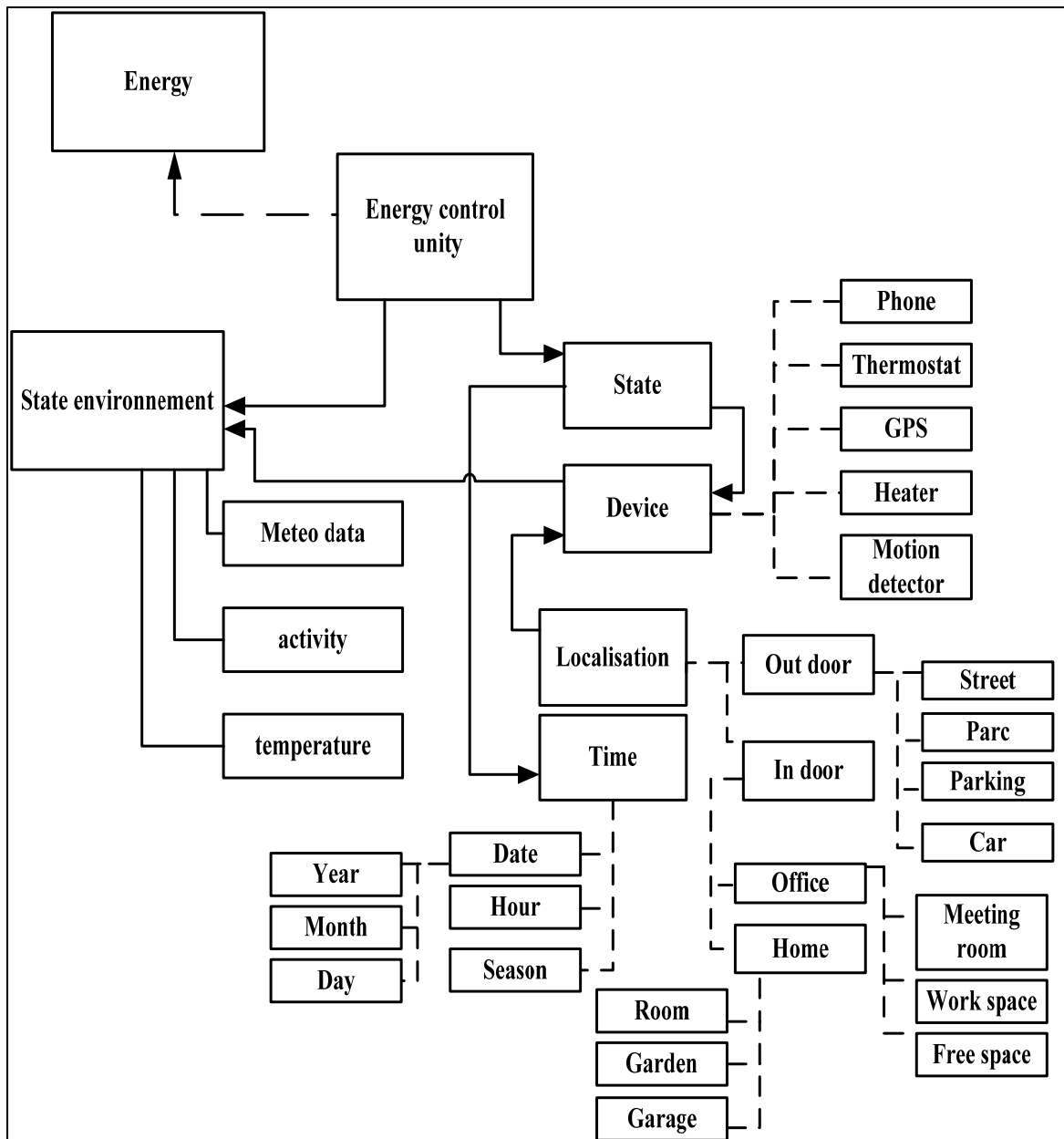


Figure 3.9 Model of heating adjustment service

3.4 Conclusion & future work

In this paper, we presented a new definition of the context, taking into account 1) the spatiotemporal information, 2) the purpose of use, and 3) any other information that may be useful and leads to a current or future change in the adaptation of service.

We presented an overview of works that have discussed the definition of context and its adaptation to the service, and we proposed a new vision of the definition of context. In the scenarios proposed in the section 3.3, we showed the difference in the contextual information oriented adaption or oriented prediction, which proves our vision.

Future work will concentrate mainly, on context modeling and prediction method. Taking into account the complexity of identifying the elements of the context in a pervasive system, in our future work, we will propose a method to model contextual information. We are going to develop architectural model of context-aware systems incorporating the prediction approach. Currently we focus to create our prediction approach based on our spatiotemporal context vision.

CHAPITRE 4

SPATIOTEMPORAL CONTEXT MODELLING IN PERVASIVE CONTEXT-AWARE COMPUTING ENVIRONMENT: A LOGIC PERSPECTIVE

Darine Ameyed¹, Moeiz Miraoui², Chakib Tadj¹

¹MMS Laboratory, Université du Québec, École de technologie supérieure
1100, rue Notre-Dame Ouest, Montréal, Québec, Canada, H3C 1K3

²Institut Supérieure des sciences appliquées et technologies de Gafsa

This paper is published the (IJACSA) International Journal of Advanced Computer Science and Applications Vol. 7, No. 4, Avril 2016.

Résumé

L'informatique pervasives et sensible au contexte est l'un des sujets qui ont reçu une attention particulière de la part des chercheurs. Le contexte, est une notion importante explorée dans de nombreux travaux. Parmi les propriétés / fonctions discutées: l'acquisition, la définition, la modélisation, le raisonnement, etc. Compte tenu de l'évolution continue des systèmes sensibles au contexte, la modélisation de contexte est toujours une tâche complexe en raison de l'absence d'une représentation adéquate de contexte, à la fois dynamique, formelle et pertinente.

Dans cet article, nous discutons différentes approches de modélisation de contexte ainsi que des travaux précédents ayant fait usage de la logique comme élément de modélisation. Nous proposons également une modélisation formelle préliminaire de contexte spatio-temporel basée sur la logique du premier ordre, dérivée de la structure des langues naturelles.

Mots clés : modèle de contexte; logique; formelle; système diffus; système sensible au contexte.

Abstract

Pervasive context-aware computing is one of the topics that received particular attention from researchers. The context, itself is an important notion explored in many works discussing its: acquisition, definition, modelling, reasoning and more. In view of the permanent evolution of context-aware systems, today context modeling, still a complex task due to the lack of; an adequate, dynamic, formal and relevant context representation.

In this paper we discuss various context modeling approaches and previous logic based works. We also propose a preliminary formal spatiotemporal context modelling based on first order logic, derived from the structure of natural languages.

Keywords: context modelling; logic; formal; pervasive system; context-aware system.

4.1 Introduction

Context-aware computing has had significant attention in diverse areas such as human computer interaction and mobile computing. The context awareness is also an important concept explored in pervasive systems and ambient intelligence. There are still questions around context: definition, modeling, and reasoning.

The notion of context itself is not new and has been explored in areas such as linguistics. Indeed, the context was described as the surrounding text element of the language and can help to determine its meaning. While in artificial intelligence, the context definition result of two main approaches: the first is a so-called propositional logical approach offered by McCarthy (McCarthy 1987); (McCarthy 1993), and Guha (Guha 1991). In this approach the context is defined as the circumstances that could determine the truth value of a term. This was translated to a set of predicates which can process a context in order to make reasoning and logic deductions.

The second approach is called local reasoning (Local Models Semantics (LMS) / Multi Background Systems (MCS)) (Giunchiglia and Ghidini 1998); (Ghidini and Giunchiglia 2001). In the LMS / MCS approach, the context is determined by a known set of facts which is performs a line of reasoning. The approach provides an incomplete environment description. Thus, the context is partial, inexact and approximate.

However, to design an effective context-aware system, the context must be well represented and modeled in an appropriate form that allows sharing between different devices in a pervasive system. Using a model more efficient provides a higher level of abstraction to facilitate adaptation.

As indicated by Henricksen (Henricksen, Indulska et al. 2002), there is generally a significant difference between the input information and the one which is useful for applications. This difference may be overcome by various types of context information processing. Therefore, modelling is a crucial step in the context treatment. Indeed, modelling includes the analysis and design of contextual information comprised within the system, as an abstract representation at the data structure level and at the semantics level.

Context-aware applications in pervasive environments can tailor their operation to the current context and thus can be more user-friendly and effective. A lot of work has gone into developing infrastructures that enable applications to be context-aware, (Nalepa and Bobek 2014); (Hu, Wang et al. 2012); (Dey 2001); (Ranganathan, Campbell et al. 2002). One of the challenges in promoting context-awareness is developing a flexible and expressive model for context. On one hand, we need a uniform representation that can cover the wide variety of possible contexts. On the other hand, the model should allow performing complex operations on context.

In this paper, we show that using a logic model based on spatiotemporal axes is a very effective and powerful way of dealing with contexts. In this model, contexts are represented as first order predicates. This representation of context is very expressive and can be used to represent various kinds of contexts. The model supports operations on context like conjunction, disjunction, negation and quantification. A key advantage of using a formal model for context is that one can clearly specify the power and expressiveness of the model.

A lot of work has been done in the formal methods area of first order logic strength, expressiveness and decidability. However, most researchers worked on current context and adaptation but not on prediction and anticipate adaptation. This paper develops our vision around context after having redefined in previous work: ``A Spatiotemporal Context Definition for Service Adaptation Prediction in a Pervasive Computing Environment`` (Ameyed, Miraoui et al. 2015), we propose actually our method to model contextual information. This paper offers a new model based on first order logic and spatiotemporal axes.

The rest of the paper is organized as follows: Section 4.2 provides an overview of context modelling in the literature. Section 4.3 presents our proposed context model based on logic. We will introduce our modelling methodology and we will show how our model help to get a context model with a high level abstraction. At the end of section 4.3, we will propose a scenario to demonstrate how we can model a context based in our logic model. Section 4.4 concludes the paper with a discussion, our contributions and presents our future work.

4.2 Related Work

Context modelling is a fundamental step for the development of context-aware systems. The existence of well-designed context models will ease the construction of such systems. Context modelling consists of analysis and design contextual information contained in the system as an abstract form at the data structure level as well as the semantic level. Several modelling approaches have been proposed, studied and analyzed in the literature. In this section, at first, we will begin by presenting a survey and comparison (table.1) about different context modelling approaches in general, followed by another overview focused more on the work using a logic based model. In this part, we discuss this model, showing its weakness on which we based our motivation to propose our new model.

4.2.1 Overviews of context modelling approaches

Strang et al(Strang and Linnhoff-Popien 2004) surveyed the most relevant approaches for context modelling and compared them to some requirements of ubiquitous computing such as: distributed composition, partial validation, information quality richness, incompleteness and ambiguity, formality level and applicability. They concluded that ontology makes the best context description compared to the surveyed methods because it provides a good information sharing with common semantics. However, this does not mean that the other approaches are unsuitable for ubiquitous computing environment.

Bettini et al. (Bettini, Brdiczka et al. 2010) discussed the requirements that context modelling and reasoning techniques should meet. They have selected a set requirement for context models: heterogeneity and mobility, relationships and dependencies, timeliness, imperfection, reasoning, modelling formalisms usability and efficient context provisioning. They did not mention logic based context model: instead, they introduced hybrid approaches as an attempt to combine different formalisms and techniques to improve the identified requirements. Perera et al. (Perera, Zaslavsky et al. 2014) surveyed context awareness from an Internet of Things perspective. They discussed high-level context modelling techniques.

Their focus was on conceptual perspective of each modelling technique not on specific implementation. Their discussion was based on the six most popular context modelling methods: key-value, markup schemes, graphical, object based, logic based, and ontology based modelling.

In their conclusion, they mentioned that logic based modelling provides much more expressive fullness compared to the other models. However, lack of standardization reduces their reusability and applicability.

Most importantly, they concluded that incorporating multiple modelling techniques is the best way to produce efficient and effective results, which will lessen each other's weaknesses. Therefore, no single modelling technique is ideal to be used in a standalone manner.

Tableau 4.1 comparison of context modelling approaches

Approach	Strength	weakness
Key-value	<ul style="list-style-type: none"> ▪ Simple ▪ Flexible ▪ Easy to manage in a small system 	<ul style="list-style-type: none"> ▪ Model limited amount of data ▪ Depend in application ▪ No structure ▪ Not adaptive ▪ No standard processing tool available ▪ No validation support ▪ No relationship modelling ▪ Hard to extract information
Markup schemes	<ul style="list-style-type: none"> ▪ Flexible ▪ Structured ▪ Available processing tools ▪ Useful as intermediate data organization format like network data transfer mode. 	<ul style="list-style-type: none"> ▪ Depends on application ▪ No standard ▪ Start be complexes in evolving ▪ Hard to extract information
Graphical	<ul style="list-style-type: none"> ▪ Provide relationships modelling ▪ Easy to extract information ▪ Flexible implementation ▪ Useful for data archival and historic context store 	<ul style="list-style-type: none"> ▪ Complex to retrieve information ▪ Configuration is obligatory ▪ No standard ▪ Complex implementation ▪ Hard interoperability between different implementation
Object based	<ul style="list-style-type: none"> ▪ Provide relationship modelling ▪ Available processing tools ▪ Easy integration ▪ Support data transformation over network 	<ul style="list-style-type: none"> ▪ Complex to retrieve information ▪ No standard
Ontology based	<ul style="list-style-type: none"> ▪ Support semantic reasoning ▪ Provide an easier 	<ul style="list-style-type: none"> ▪ Complex representation ▪ Complex to retrieve information

Approach	Strength	weakness
	representation of context <ul style="list-style-type: none"> ▪ Advanced tools available ▪ Provide sharing model ▪ Supported by standardization 	
Logic based	<ul style="list-style-type: none"> ▪ Generate high-level context based on low-level context ▪ Simple to use ▪ Simple to model ▪ Supports logical reasoning ▪ Processing tools available ▪ Can generate new knowledge ▪ Model event and action ▪ Define constrains and restrictions ▪ High level of formality 	<ul style="list-style-type: none"> ▪ Partial validation difficult to maintain ▪ Applicability can be complicated. ▪ No standard

4.2.2 Synthesis

To synthesize, we propose the following comparison table (Table 4.1).

Most of the previous work has focused on ontology based context modelling and less effort has been spent on logic based context modelling. In the following, we are going to focus in depth on related work based on context modelling approaches using logic.

4.2.3 Related work on context modelling approaches logic based

A logic model provides a formal representation of contextual information. Using a reasoning process or an inference engine, a logic model can deduce new information based on existing rules in the system.

Among the first works using this approach, those by Carthy and Buvac (Carthy 1993); (Carthy and Buvac 1997), introduced the context as a formal object. They defined simple axioms for events or phenomena with common sense and treated the context associated with a particular situation. They provided basic relationship $ist(c, p)$, which means that the proposition p is true in the context c , defined by formulas such as:

$C0: ist(\text{contexte} - \text{of}(\text{Sherlock Holmes stories}), \text{Holmes is detective})$.

This model also uses the notion of inheritance (Carthy and Buvac 1997).

Another early representative of this type of approach is the theory of situations introduced by Akram et al. (Akman and Surav 1997). This approach is inspired by the theory proposed by Barwise et al (Barwise and Perry 1981). They have tried to give theoretical semantics model of natural language in a formal logic system. Akram et al, have subsequently provided an extension for this model. They represented the facts related to a particular context with non-parametric expressions supported by the type of situation that matches the context.

A similar approach proposed by Gray and Salber (Gray and Salber 2001) used the first order logic as a formal representation of context information and their relationships.

Another approach in this same category was used to develop multimedia system (Bacon, Bates et al. 1997). In this system, location taken as a context aspect is expressed as a fact in a rules-based system. The system itself is implemented in Prolog.

Ranganathan et al (Ranganathan and Campbell 2003) proposed a context model based on first order predicate, in the ConChat project. Their context model describes context information properties and structure and the kinds of operations that can be performed on context, e.g. conjunction, disjunction, negation and quantification. The predicate name is the type of context being described.

It is also possible to have relational operators inside predicates. The predicate form is not general and the meaning and number of parameter depends on the described context element.

The context model didn't make restriction on the type of values that different arguments in the context predicate can take. So, predicate arguments can be arbitrarily complex structures. One or more arguments of a context predicates can be functions that return some values. In second time, the authors used rules to deduce new contexts based on existing contexts.

Roman et al. (Henricksen, Indulska et al. 2002) presented an experimental middleware infrastructure called Gaia (an Active Space System Software Infrastructure) where they used a model for context that is based on first order logic and Boolean algebra, which allows them to easily write various rules to describe context information. They represented context through a 4-ary predicate, whose structure is borrowed from a simple clause in the English language of the form <subject><verb><object>. An atomic context predicate is defined as follows:

Context (<ContextType>, <Subject>, <Relater>, <Object>)

e.g. Context (location, chris, entering, room 3231).

In some cases, one or more elements of a predicate may be empty. It is possible to construct more complex contexts by performing first order logic operations such as context predicates quantification, implication, conjunction, disjunction, and negation.

Gu et al. (Gu, Pung et al. 2005) proposed a Service-Oriented Context-Aware Middleware (SOCAM) architecture for the building and rapid prototyping of context-aware services. In their model, contexts are represented as first-order predicate calculus. The basic model had the form of Predicate (subject, value). The context predicates structures are described in ontology. The ontology is written in OWL as a collection of RDF triples, each statement being in the form (subject, predicate, object).

Other works followed the same approach (Hu, Wang et al. 2012), (Kadouche, Mokhtari et al. 2008) Nalepa & Bobek (Nalepa and Bobek 2014) proposed a new rule-based context reasoning platform tailored to the needs of intelligent distributed mobile computing devices. They made a comparison of existent context modelling approaches and they took into consideration the following aspect of context modelling methods: formalization, simplicity,

expressiveness, support for inference, uncertainty handling and existing tools that support design. They also proposed an inference service that uses HeaRT inference engine to provide on-line efficient reasoning for mobile devices.

4.2.4 Synthesis

A logic based model provides the ability to create complex expressions in first order logic, and deduct a high-level context from the basic context (captured) using an approach based on rules. The model defines a base structure to present atomically each object context. Deduction approaches based on a logical modelling offer the most appropriate mechanisms to achieve abstraction information; it will be more specified later in section (4.3).

In spite of the high formal level of logic, less effort has been spent on logic based context modelling and most previous work on this topic has been centered on the ontology model. The previous proposed works on logic based context modelling suffer from two main weakness points:

- the context predicates are not generic enough and their components are not fixed and vary according to the predicate usage;
- predicates components do not cover all aspects of context because they are not based on a clear and concise context definition which limit their usage to some specific applications and negatively affect the expressiveness.

Therefore, we propose in our approach to follow a logic model that solves these weaknesses. Based on natural language and our context definition (Ameyed, Miraoui et al. 2015), focusing on spatiotemporal parameters and the contextual information usage which promotes proactive adaptation: current or anticipatory based on future context prediction. We have demonstrated that space and time are an important context information in many context-aware applications (Ameyed, Miraoui et al. 2015). Most definition mentions a space as vital factor e.g., the most frequently used by Dey (Dey 2001).

As described in section (4.2), many research works used the logical approach for its high level of formality, its abstraction benefits, effectiveness and its support for logical reasoning, except these works neglected the time aspect. Knowing that, based on the spatiotemporal axes, contextual information was better defined (Ameyed, Miraoui et al. 2015), we integrate these parameters in our model. This will allow a better description of the service space context and thus more expressive context reasoning and so, more efficient adaptation.

4.3 Proposed modelling approach

The logic based models are usually used in context-aware systems for their strong formalism, allowing verification and validation of context models and their ability to automate inductive and deductive reasoning on contextual information. We choose to use the first order logic to model our context. The first order logic provides an expressive description on contextual information close to the real environment and natural language.

Firstly, we describe our context formalism first order logic based and how we can describe more complicate context information using Boolean operators and existential and universal quantifiers (Section 4.3.1). Secondly, we try to provide simple reasoning logic model that provide high level representation of context which can be used as a basis for more advanced reasoning on the context, such as the context discovery or prediction. We believe that logic based models are very efficient tools for context reasoning and are adequate for general pervasive context-aware systems (Section 4.3.2).

4.3.1 Context formalism

- **The basic structure - The context predicate**

The required context is the one in which, the service is more likely to be offered. If the current context satisfies this requirement, then the service will be offered.

Definition.1: The context is the set of entities with a spatiotemporal variation that affects the quality of the service, in a short or long term (current service vs anticipatory service) (Ameyed, Miraoui et al. 2015).

Definition.2: The state of a service space, is the combination of the all the states of the entities existing in this space (including active services and contexts linked to those services).

A context can be reduced to an atomic form, derived from the structure of natural languages. For example, in a natural language we describe information with simple-clause sentence containing a subject and a verb:

Simple clause (<subject> <verb>)

exp : Adam enters

We can also use the natural form sentence: subject-verb-object

Sentence (<subject> <verb> <object>)

exp : Adam enters in room

This sentence might be an observation in the context, which might influence the behaviour of a system and trigger an adaptation in order to offer a service:

Context (<user> <action> <localisation>)

However, the contextual information available is less useful unless, we have a complement of information about the spatiotemporal qualifications. In a natural language sentence, time is implicit and given off by the tense of the sentence.

In a systematic description, we use parameters. Knowing that spatiotemporal information in the service space, might lead to a more efficient adaptation (Ameyed, Miraoui et al. 2015), we make it a requirement to add two parameters to the description: a time parameter and a location parameter.

This may take the following predicate form:

Context (<element> <state> <value> <times> <location>)

Example: if we attempt to describe the following information “Alex enters the room” in a service space we should emphasize the time and location parameters in that information

Context (<Adam> <presence> <active> <21 :00> <room 1>)

- **element**: indicates the type of object (i.e. temperature, individual, printer, etc.);
- **state**: indicates the state of an element, an action, a functionality and is linked to the type of element it describes;
- **value**: observation qualifying a state a functionality or an action (i.e. on, off);
- **time**: observation time, instant when the element’s state was observed;
- **location**: place where the observation happened.

To describe a complex context expression, we might use Boolean, quantitative and existential operators, as will be detailed below.

- **Operation on context**

Our goal is to have an accurate description of the physical world. With a pervasive system, we would also like to describe the service space. To ensure context-awareness with an accurate description yields an efficient system. As described above, the atomic form can be extended to describe all the elements of a context in a service space. It is possible to scale in complexity adapting the description accordingly by integrating Boolean operators and logical quantifiers to the predicates.

Tableau 4.2 conjunction, disjunction and negation operator

Conjunction	And	\wedge
Disjunction	Or	\vee
Negation	Not	\neg

We can perform conjunctions, disjunction and negation for a complex description, as described in table 4.2.

▪ Example :

Context (<lamp> <lighting> <on> <22:00> <room1>) \wedge Context (<User1> presence> <active> <22 :00> <room1>)

Describes user 1 as being in room 1 while the light is on, at 22h00.

\neg Context (<User2> <presence> <active> <22:00> <room1>)

Describes user 2 as not being in room 1 at 22h00.

Context (<lamp> <lighting> <on> <22:00> <room1>) \wedge Context (<User1> <presence> <active> <22:00> <room1>) \vee Context (<User2> <presence> <active> <22:00> <room1>)

Describes light as being on in room 1 at 22h00 and either user 2 or user 1 are registering their presence there.

- **Quantification**

An existential or universal quantification model allows us to represent even richer sets, table 4.3. A context might be quantified with respect to one of its parameters.

The existential quantifier indicates that the context is true, at least for one mentioned variable.

Exemple

$$\exists \text{location } Y \text{ Context } (\langle \text{user1} \rangle \langle \text{presence} \rangle \langle \text{active} \rangle \langle 22 :00 \rangle \langle Y \rangle)$$

The user is present at least in one location

The universal quantifier shows that the context is true for all the occurrences of the mentioned variable.

Example

$$\forall \text{user } X \text{ Context } (\langle X \rangle \langle \text{presence} \rangle \langle \text{active} \rangle \langle 22 :00 \rangle \langle \text{room1} \rangle)$$

To describe any user in the location designated by 'room 1'

Tableau 4.3 Quantification operator

Existential	Exists	\exists
	Exists and is unique	$\exists!$
Universal	For all	\forall

4.3.2 Context Interpretation

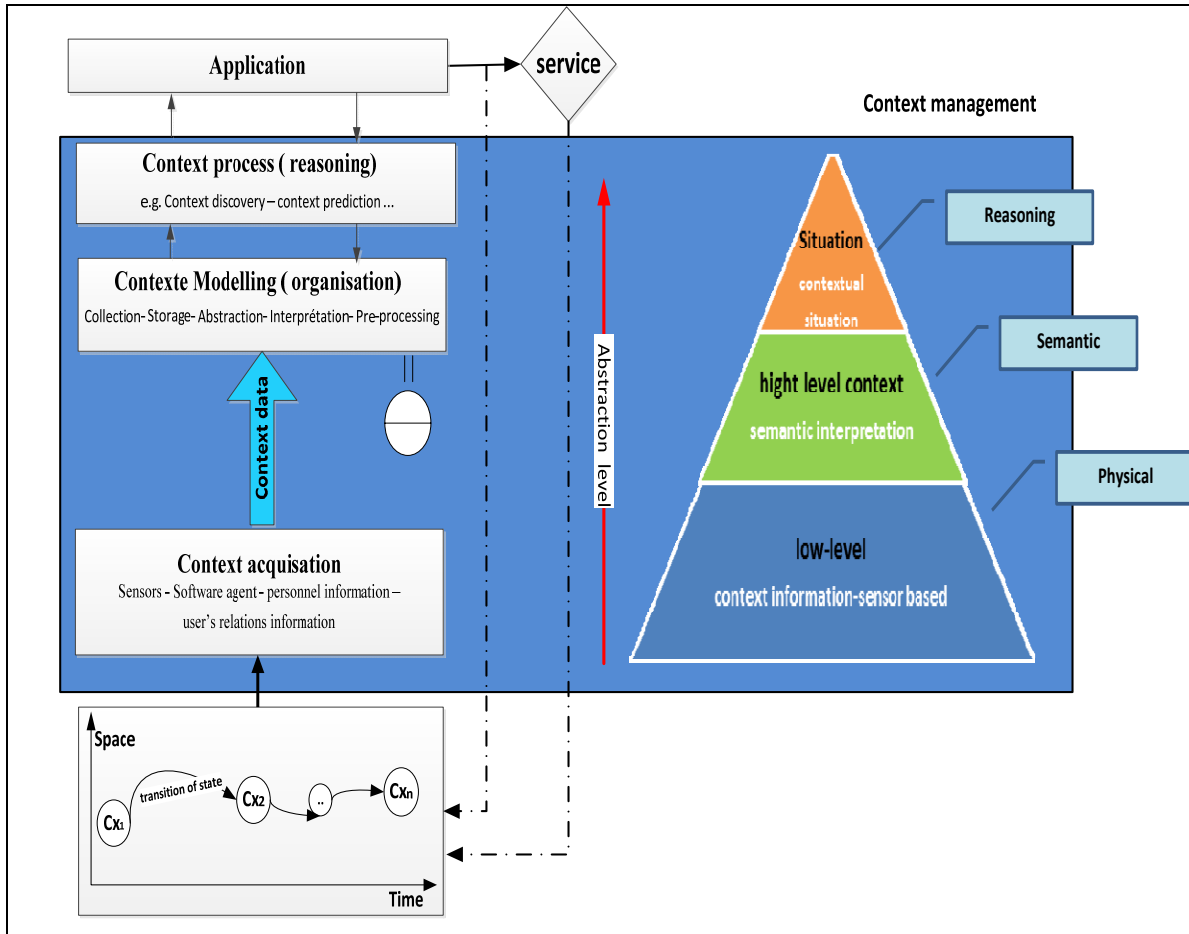


Figure 4.1 Context awareness operations with different layers of abstraction

Information from physical sensor, called low-level context and acquired without any extra interpretation or analysis, can be meaningless, susceptible, superfluous, or uncertain, the limitation of low-level contextual cues: when modelling different service space interaction risks reducing the usefulness of context-aware applications. A way to mitigate this problem is the elicitation of higher-level context from raw and unrefined sensor values, named context reasoning and interpretation as show in Figure 4.1.

Using a formal approach to model context, context can be processed with logical reasoning methods (e.g.: rule based, temporal logic ...). The context reasoning was useful to: (i) checking the consistency of context, and (ii) deducing high level, implicit context from low-level explicit context.

The idea is to abstract from low-level context by creating a new model layer that generates a higher-level. We refer to a different work in literature discussed a contextual situation (Dey 2001); (Dobson and Ye 2006).

In context-aware system, situations are the semantic interpretation of low-level context situations, which give meaning into the application, being more stable, easier to define and maintain than basic contextual (Dobson and Ye 2006). Adaptations in context-aware applications are then caused by the change of situations. Operating at a high level of context abstraction to define a contextual situation make easier application implementation.

In literature several approaches have been proposed to get abstraction and define situation. For example, (Bettini, Brdiczka et al. 2010) describes six different ways to specify the situation ``*in_meeting_now*`` based on:

- co-location of people and information in agenda;
- co-location of filled coffee cups in a room;
- weight sensors on the floor;
- devices in the room;
- sounds and noises;
- cameras "watch" meeting room for activity.

In this paper, we use formal logic approach to model our context and acquire high level contextual model concerning situation.

Early approaches use formal logic to describe and represent these states. A first work, using this kind is the Situation Theory was proposed by Barwise and Perry (Barwise and Perry 1983). Situation Theory tries to cover model-theoretic semantics of natural language in a

formal logic system (Reiter 2001). The situation deduction provides a logical language for reasoning about action and adaptation.

Our approach based on formal logic provides a high level of abstraction and formality for specifying the context and contextual situation. Our approach builds a logic link between context and situation. It puts under the causal connection. This joined our vision of context and its use on adaptation or prediction (Ameyed, Miraoui et al. 2015).

Based on McCarthy's definition of situation (McCarthy 1993), who described a situation as a complete state of the universe at an instant of time. We deduce, in order to describe a service space situation we do not need to get the whole state of universe but rather system environment at this time; which in reality is the context, as a snapshot or instantiation of all context variables at some point of time in space service as mention in definition 2 (subsection 4.3.1).

The value of context entity parameters changes from situation to situation. To be able to deduce a situation and abstract a context into situation, we use the characteristic features of a context to get properties that are more stable over one situation. Situation encompass a complex context witch can be represented by a predicate and link structure. Situations are a complex context limited by time. We can form a situation as:

$$S = (T_i, T_e, C_s)$$

Where: (i) T_i is the starting time it is the first time context parameter associated to the specific situation; (ii) T_e is the end time, it is the last time context parameter associated to the same situation; and (iii) C_s is the conjunction of all context entity associated to the situation.

This may take the following predicate form, that we can use it as a deduction rule:

$$\forall \text{time } t \in [T_i, T_e] \text{ Context } (\langle \text{element1} \rangle \langle \text{state} \rangle \langle \text{value} \rangle \langle t \rangle \langle \text{location} \rangle) \wedge \text{Context } (\langle \text{element2} \rangle \langle \text{state} \rangle \langle \text{value} \rangle \langle t \rangle \langle \text{location} \rangle) \wedge \text{Context } (\langle \text{element } n \rangle \langle \text{state} \rangle \langle \text{value} \rangle \langle t \rangle \langle \text{location} \rangle) \rightarrow C_s.$$

In high level:

$Cs \rightarrow S$

4.3.3 Senario Morning at work

Adam starts his day, it is a work day. He leaves to work and issues a vocal command to his car indicating his destination: the office. The computed commuting time is 30 minutes. Adam should be at the office at 09h00. On his today's schedule, he has a meeting planned for 10h00, where he is supposed to make a presentation for his team.

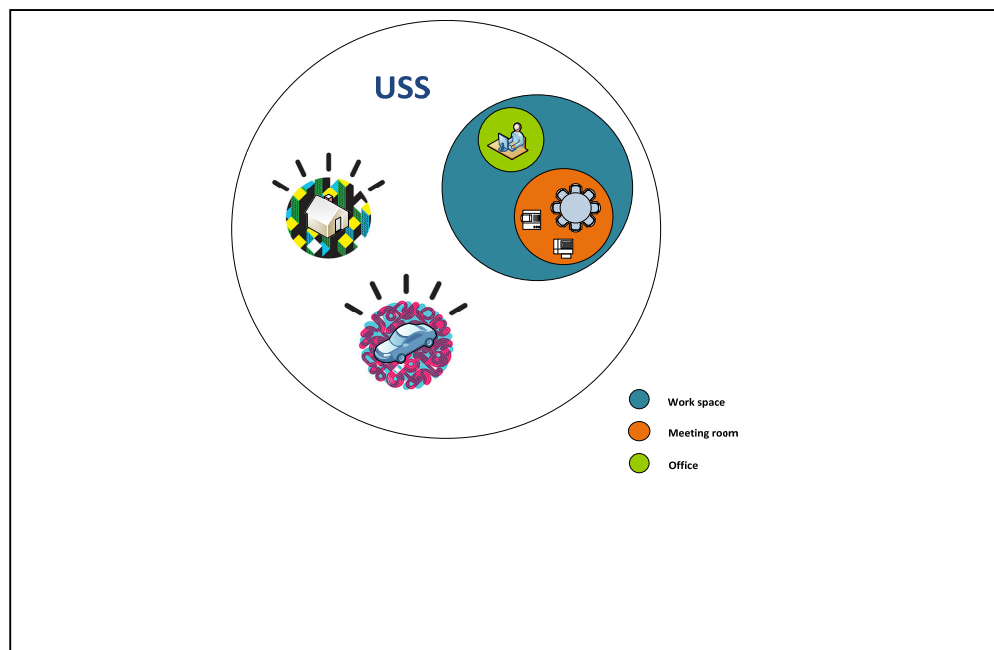


Figure 4.2 USS services spaces

Reaching office by 09h00, Adam swipes his access card; the system authenticates him and opens the door. The blinds were already retracted, the temperature adjusted to ambient and the office computer, started. Adam starts working on his presentation. At 09h45, an audiovisual notification appears on the computer screen reminding Adam of his upcoming meeting in 15

minutes. According to the meeting location, he needs 10 minutes to reach the meeting room. The desktop computer saves Adam's work and synchronizes with his laptop.

On the way to the meeting, the system issues commands to upgrade the service spaces:

- USS prepares the meeting room: launch the projector and the display screen, activate the audio system, adjust luminosity, temperature and deploys the blinds;
- USS manages Adam's office while he's away: turns off the desk lamp, locks his computer session, launches system upgrades on the computer and locks access to the office.

Adam and his team are in the meeting room, the presentation is about to start. The system switches all phones to silent mode and locks access to the meeting room. The meeting ends at 11h00. Adam goes back to his office. The system updates the office's context:

- unlock the computer session;
- activate the UV blind;
- switch the phone back to normal mode.

USS updates the state of the meeting room, according to its scheduled uses. On this paper we focus on the context modelling we use this scenario to define the context and the context situation. On our future work we will use the same scenario for our context prediction approach and adaptation.

Tableau 4.4 Devices, states and services spaces

Service space	Devices	Devices states
Office	Door	Locked/Unlocked
	Blinds	Open/Close
	Light	Swish on /Swish off
	Pc	Run / shut off/ standby
	Air conditioner	Shut of/ cool mode / warming mode
	Smart phone	Outdoors mode/indoors mode/ meeting mode
Meeting room	Door	Locked/Unlocked
	Blinds	Open/Close
	Lights	Swish on /Swish off
	Pc	Run / shut off/ standby
	Lap-top	Run / shut off/ standby
	Air conditioner	Shut of/ cool mode / warming mode
	Screen	Open/close
	Projector	Swish on/ Swish off
	Audio-system	Swish on/ Swish off
	Smart phone	Outdoors mode/indoors mode/ meeting mode
	Video conferencing system	Swish on/ Swish off

To describe the situation, one of our spaces service scenario; the meeting room or the office we will be based on the ambiance (light, sound ...), the time, location of users (present, absent, co-present ...) and applications (type of application, run, off).

Our scenario's time: a morning in a working day

- T_i : initial time to context situation.
- T_e : end time to context situation.

Based on ambiance – location – time (sample contextual-situation- office), we form various rules written in first-order predicate based in our context-model in order to deduce the space situation.

A few of these rules are as following :

1. Office Context Modelling

Tableau 4.5 Office context situation

Situation	Ambient Cs information
Cs-office→Work-time	<ul style="list-style-type: none"> ▪ Lighting (bright) ▪ Occupation (busy) ▪ Sound (noisy)
Cs-office→At-rest	<ul style="list-style-type: none"> ▪ Lighting (gloomy) ▪ Occupation (empty) ▪ Sound (silent)

$\forall \text{time } t \text{ Contexte } (\langle \text{lighting} \rangle \langle \text{bright} \rangle \langle 1 \rangle \langle t \rangle \langle \text{office} \rangle) \wedge \text{Contexte } (\langle \text{sound} \rangle \langle \text{noisy} \rangle \langle 1 \rangle \langle t \rangle \langle \text{office} \rangle) \wedge \text{Contexte } (\langle \text{occupation} \rangle \langle \text{busy} \rangle \langle \text{user} = 1 \rangle \langle t \rangle \langle \text{office} \rangle) \rightarrow \text{Contexte } (\langle \text{adam-office-situation} \rangle \langle \text{work time} \rangle \langle \text{true} \rangle \langle t \rangle \langle \text{workspace} \rangle)$

$$\forall \text{time } t \text{ Contexte } (\langle \text{lighting} \rangle \langle \text{gloomy} \rangle \langle 1 \rangle \langle t \rangle \langle \text{office} \rangle) \wedge \text{Contexte } (\langle \text{sound} \rangle \langle \text{silent} \rangle \langle \text{true} \rangle \langle t \rangle \langle \text{office} \rangle) \wedge \text{Contexte } (\langle \text{occupation} \rangle \langle \text{empty} \rangle \langle \text{user}=0 \rangle \langle t \rangle \langle \text{office} \rangle) \rightarrow \text{Contexte } (\langle \text{adam-office} \rangle \langle \text{at rest} \rangle \langle \text{true} \rangle \langle t \rangle \langle \text{workspace} \rangle)$$

2. Office Context Modeling

Tableau 4.6 Meeting room context situation

Situation	Ambient Cs information
Cs-room→Meeting	<ul style="list-style-type: none"> ▪ Lighting (bright-level2) ▪ Occupation (busy) ▪ Sound (noisy) ▪ Phone (meeting-mood)
Cs-room→Presentation	<ul style="list-style-type: none"> ▪ Lighting (bright-bright-level1) ▪ Occupation (busy) ▪ Sound (noisy) ▪ Phone (meeting-mood) ▪ PowerPoint (run)
Cs-room→Video-conference	<ul style="list-style-type: none"> ▪ Lighting (bright-bright-level1) ▪ Occupation (busy) ▪ Sound (noisy) ▪ Phone (meeting-mood) ▪ Video-conferencing-system (run)
Cs-room→At-rest	<ul style="list-style-type: none"> ▪ Lighting (gloomy) ▪ Occupation (empty) ▪ Sound (silent)

$\forall \text{time } t \in [T_i, T_e] \text{Contexte } (\langle \text{lighting} \rangle \langle \text{bright} \rangle \langle \text{level1} \rangle \langle t \rangle \langle \text{meeting-room} \rangle)$
 $\wedge \text{Contexte } (\langle \text{software-app} \rangle \langle \text{powerpoint} \rangle \langle \text{on} \rangle \langle t \rangle \langle \text{meeting-room} \rangle) \wedge \text{Contexte}$
 $(\langle \text{occupation} \rangle \langle \text{busy} \rangle \langle \text{user} \rangle 1 \rangle \langle t \rangle \langle \text{meeting-room} \rangle)$
 $\rightarrow \text{Contexte } (\langle \text{room-situation} \rangle \langle \text{presentation} \rangle \langle \text{true} \rangle \langle t \rangle \langle \text{meeting-room1} \rangle)$

$\forall \text{time } t \in [T_i, T_e] \text{Contexte } (\langle \text{lighting} \rangle \langle \text{bright} \rangle \langle \text{level2} \rangle \langle t \rangle \langle \text{meeting-room} \rangle)$
 $\wedge \text{Contexte } (\langle \text{software-app} \rangle \langle \text{powerpoint} \rangle \langle \text{of} \rangle \langle t \rangle \langle \text{meeting-room} \rangle) \wedge \text{Contexte}$
 $(\langle \text{occupation} \rangle \langle \text{busy} \rangle \langle \text{user} \rangle 1 \rangle \langle t \rangle \langle \text{meeting-room} \rangle)$
 $\rightarrow \text{Contexte } (\langle \text{room-situation} \rangle \langle \text{meeting} \rangle \langle \text{true} \rangle \langle t \rangle \langle \text{meeting-room1} \rangle)$

$\forall \text{time } t \in [T_i, T_e] \text{Contexte } (\langle \text{lighting} \rangle \langle \text{bright} \rangle \langle \text{level1} \rangle \langle t \rangle \langle \text{meeting-room} \rangle)$
 $\wedge \text{Contexte } (\langle \text{video-conf-system} \rangle \langle 1 \rangle \langle \text{on} \rangle \langle t \rangle \langle \text{meeting-room} \rangle) \wedge \text{Contexte}$
 $(\langle \text{occupation} \rangle \langle \text{busy} \rangle \langle \text{user} \rangle 1 \rangle \langle t \rangle \langle \text{meeting-room} \rangle)$
 $\rightarrow \text{Contexte } (\langle \text{room-situation} \rangle \langle \text{video-conference} \rangle \langle \text{true} \rangle \langle t \rangle \langle \text{meeting-room1} \rangle)$

$\forall \text{time } t \text{Contexte } (\langle \text{lighting} \rangle \langle \text{gloomy} \rangle \langle \text{level} \rangle \langle t \rangle \langle \text{meeting-room} \rangle) \wedge$
 $\text{Contexte } (\langle \text{video-conf-system} \rangle \langle 1 \rangle \langle \text{on} \rangle \langle t \rangle \langle \text{meeting-room} \rangle) \wedge \text{Contexte}$
 $(\langle \text{occupation} \rangle \langle \text{empty} \rangle \langle \text{user} \rangle 0 \rangle \langle t \rangle \langle \text{meeting-room} \rangle)$
 $\rightarrow \text{Contexte } (\langle \text{room-situation} \rangle \langle \text{at-rest} \rangle \langle \text{true} \rangle \langle t \rangle \langle \text{meeting-room1} \rangle)$

4.4 Conclusion & Future Work

In this paper we have presented a formal context model taking into account the spatiotemporal frame. In our model context are presented as first-order predicate calculus. We showed how we can extend the basic model form to an extended context and how we can use it to deduce various situations to provide high-level context information.

Our model follows our reflection on the spatiotemporal contextual information, and provides a formal method to introduce it in the context modelling. Compare to the other formal model, our proposal provide notables properties for context model: dynamic context easily understandable; natural language support, logic reasoning support, remaining faithful to spatiotemporal framework.

Future work will concentrate mainly, on context reasoning and prediction method. We are exploring how to better design a context discovery engine, and make it in a generic model reusable.

CHAPITRE 5

USING PROBABILISTIC TEMPORAL LOGIC PCTL AND MODEL CHECKING FOR CONTEXT PREDICTION

Darine Ameyed¹, Moeiz Miraoui², Atef Zguia^{1,3}, Fahmi Jaafer⁴, Chakib Tadj¹

¹MMS Laboratory, Université du Québec, École de technologie supérieure
1100, rue Notre-Dame Ouest, Montréal, Quebec, Canada, H3C 1K3

²CMMomputer Science Department ISSAT, Gafsa, Tunisia

³Department of Computer science, Taif University, Taif, Saudi Arabia

⁴Concordia university of Edmonton, Edmonton, Canada

Submit to the International Journal of Intelligent Systems Technologies and Applications.

Résumé

La prédiction de contexte est un problème relativement nouveau dans le domaine informatique. Les travaux portants sur la recherche de solutions et d'approche pour la prédiction du contexte sont justes en cours d'élaboration. Ils soulèvent encore des nombreux défis qui restent à résoudre.

Dans cet article, nous présentons une nouvelle approche formelle pour la prédiction de contexte dans les systèmes proactifs sensibles au contexte. Nous exprimons le contexte et la transition dans un système diffus sous une présentation formelle, en utilisant la logique temporelle probabiliste PCTL (une extension probabiliste de la logique temporelle). Nous proposons un modèle de transition probabiliste pour coder le comportement du système dans le temps. La combinaison de PCTL avec le modèle stochastique nous permet de tracer, analyser et prédire le contexte futur en utilisant le modèle checking pour vérifier les propriétés des états futurs retournant ainsi des résultats quantitatifs.

Mots clés: Context prediction; logic; PCTL; pervasive system; context-aware system; stochastic transition model.

Abstract

Context-awareness is one of the fundamental principal underpinning pervasive computing. Context prediction, a new trend in pervasive computing, is an open-ended research topic with a lot of challenges and opportunities of innovation. With the constant evolution of context-aware systems, up to now context prediction still a complex task due to the lack of formal approach, an efficient expressivity and a reasoning based on temporal logic fitting with the

dynamic and nondeterministic nature of the system's environment, such as the user's behavior. Probabilistic model checking is a method for automatically verifying that a probabilistic system satisfies a property with a given likelihood. The probabilistic temporal logic PCTL is however, a better choice for the property specification language. In this paper we present a new approach using those techniques; a probabilistic temporal logic and model checking for context prediction in context-aware system.

Keywords: Context prediction; logic; PCTL; pervasive system; context-aware system; stochastic transition model.

5.1 Introduction

Prediction was a research topic in different fields: meteorology, economy, trends of prices and stocks as well as in computer science and software engineering such as predicting failure in software (Salfner, Lenk et al. 2010). Predictive mechanisms help to anticipate actions and to implement the appropriate preventive measures. Ubiquitous computing systems are no exception in this respect; they do actually follow this trend. To be more proactive, Ubiquitous systems have to provide service adaptation, according to the dynamic evolution of their context, in order to offer an adequate service fitting the user's needs.

One significant challenge in particular, is to proactively assess the user's needs in the real world without requiring explicit input. Furthermore, a ubiquitous system must provide the user with services well adapted to the overall context. Indeed services will be triggered dynamically and without an explicit user intervention in a proactive way. Making use of the context in applications is a current area of research known as "context-awareness"(Boytsov 2011). A sensitive-context application must perceive the context of the users and their environment and adapt its behaviour accordingly. Most of the work on service adaptation in context-awareness is focused on the current context.

Therefore, the future context allows the pervasive system to choose the most effective strategies to achieve its goals, and to provide an active and fast adaptation to future situations. This raises our interest in context prediction. In ubiquitous computing several studies and research have been conducted too, under the prediction topic (Boytsov 2011)

(Boytsov and Zaslavsky 2010); (Boytsov, Zaslavsky et al. 2009); (da Rosa, Barbosa et al. 2016). These works aim to introduce new prediction techniques to increase the dynamic nature and the proactivity of those pervasive systems.



▪ **Problem and motivation**

Current approaches to context prediction only deduce uni-dimensional information for the future context (e.g. future location). As a consequence, their expressiveness and effectiveness, is limited. Even more so, if the system is unable to recognize the expected time of such context changes and the underlying behaviour.

Moreover, these approaches face a common challenge: the lack of formal and general approaches for dealing with context prediction and more specifically, allowing proactivity and service anticipation using context prediction. They assert the lack of a common development framework for context prediction as well as formal representation for the context and a formal approach for the prediction.

Over the past few years a more general research trend emerged, focusing on context prediction such as the work described in (Sigg, Haseloff et al. 2010); (da Rosa, Barbosa et al. 2016); (Sigg, Haseloff et al. 2010), which discussed directions for research on this issue. They pointed out that the work in this area is mostly limited to location information and a challenge they face is: (i) to consider more general context information, (ii) to be able to support a temporal constraint (iii), and to provide a logic based expressive prediction with a clear semantic and formalism.

▪ **Proposition**

Thus we think that pervasive proactive systems need therefore, the ability to reason with time dependencies and even more complex than that: spatiotemporal dimensions and the overall context. To be able to recognise a future contextual information (e.g where is the location of the user X in the next 5 minutes?) and to provide an answer and anticipate a service

associated with a future context, must be possible (e.g activity X can be executed on location Y in the next Z minutes). A system that can include this kind of knowledge provides more flexibility and allows the ability to act in a more efficient manner.

In our previous research work, we emphasized on context prediction context in pervasive context-aware systems. We proposed a new definition that supports prediction in the same multi-dimension reasoning (Ameyed, Miraoui et al. 2015). In another step towards our goal for providing formal prediction approach to context modeling, we proposed a logic-based model including a temporal constraint (Ameyed, Miraoui et al.). This paper is therefore another step to provide a new spatiotemporal expressive prediction based on a formal semantic of a probabilistic temporal logic and stochastic transition model.

▪ **Contribution**

In this paper we are formalizing a new approach to express context prediction in context-aware systems. We express context and the transition in a pervasive system with a formal semantic, using a probabilistic temporal logic PCTL (a probabilistic extension of temporal logic). We propose a probabilistic transition model to encode the system's behaviour over the time. Combining PCTL with a stochastic model, we can trace, analyse and predict the future context, using the model checking verification to verify the future states property with a quantitative result and return the future state that has the maximum probability.

▪ **Paper's Structure**

The paper is organized as follows. First, we give an overview of the available prediction methods (section 5.2) with a synthesis and an evaluation. After that, we present our approach (section 5.3) starting by a presentation of temporal logic and an explanation of our choice of probabilistic temporal logic. We then present our model detailing each included component.

And we finish this section by explaining the prediction process. Before conclude the paper, we present the evaluation of our approach (section 5.3.9) and our future work (section 5.4).

5.2 Related work

In this section we give an overview of the available research within the context prediction topic, specifically those including proactive adaptation for pervasive systems; we analyze, discuss those various works and later we present an evaluation / synthesis according to a selected set of criteria. As we have discussed and analyzed the prediction research work in our previous survey (Ameyed, Miraoui et al. 2015), according to the technical prediction approaches, we tried in this overview to discuss other related work, mostly from recent research in chronological order.

Also, we circumscribed our survey to research proposing generic models to support context prediction. Hence, the chosen works should support generic context information: works specifically devoted to the location prediction were not considered relevant. As discussed in recent surveys (Ameyed, Miraoui et al. 2015), (David, Kusber et al. 2014) the development of generic approaches is a challenge in this research area.

One of the first contributions in context prediction was proposed by **Mayrhofer** (Mayrhofer 2005). Mayrhofer proposed an architecture and a framework for context prediction that are based on an unsupervised classification, attempting to find context clusters, previously unknown from the input data. These context clusters represented recurring patterns in the input data. This approach modeled the context as a finite sequence of states where a user or a device triggers the change of the current state from one state to another. This modelling helped predicting the next states of the context based on the current state. He suggested a five-step process, taking sets of observations, each recorded at a specific time, as input and providing as output the current context of the user as well as predicting the future states of the context. The proposed stages are: sensor data acquisition, feature extraction, classification, labeling, and prediction.

Mayrhofer proposed a prediction module based on the sequence prediction technique. This technique is based on the prediction task of a theoretical computer sequence and can only be applied if the context is broken down into some form of event flow. The context prediction in this work is based only on high level context and the framework doesn't have any mechanism to support an adaptive strategy.

Like Mayrhofer, **Sigg et al.**(Sigg, Haseloff et al. 2006), (Sigg, Haseloff et al. 2010) provided a formal definition for the context prediction task relevant to the issues raised on the quality of the context and on how to handle the ambiguity of incomplete data.

This method is also based on patterns of context, the learning algorithm builds, to enable the prediction module. The context prediction module is based on an alignment method that attempts to predict the most likely continuation of a time series starting from the suffix of the observed sequence. Finally Sigg et al. (Sigg, Haseloff et al. 2010) also offers a continuous learning module in order to adapt to the change in the environment or in user habits. It continuously monitors the recorded time series stored in context history and updates the relevant patterns.

However, we didn't find in this work any specific implementation for this learning module. Only its constraints were given, including the interface specified by the context history and language description of the rules, representing patterns. Sigg does not describe any adaptive mechanism for prediction neither consider any specification for context information.

Meiners et al. (Meiners, Zaplata et al. 2010) suggested a context prediction approach called SCP (Structured Context Prediction). This approach is based on two key principles. The first is making use of knowledge of the application domain that developers can integrate when designing the application. This knowledge is described as a prediction model that specifies how the predictions are to be executed and which configures the prediction system.

The second principle sets out the application of several prediction methods, which are interchangeable. These methods are proposed to ensure the accuracy and effectiveness of predictions relevant to a given domain. They can be selected and combined by the

application developers. According to (Meiners, Zaplata et al. 2010) the prediction model assigns a method for each variable to predict its value. The method uses as input the values of other variables that are either already predicted by their own methods, or simply measured by sensors. In addition, the authors proposed an architecture for a prediction system which can be used as a reusable component by context-aware applications.

In this work the proposed Contexts Prediction architecture supports an adaptive mechanism for contexts prediction. However, this mechanism is manual, that is, the designer needs to choose at design time the most suitable algorithms for predictions. Furthermore, the architecture also has a learning component and supports only low-level context data and doesn't have a formal context representation.

Contextual spaces theory is an approach developed by **Andrey Boytsov (Boytsov 2011)**, in order to best define context-awareness and to deal with sensor problems that create uncertainty and incur a lack of reliability. This theory used spatial metaphors to represent the context as a multidimensional space. It was designed to make context-awareness clearer.

The theory of context space, was initially submitted by Padovitz and Zaslavsky (Padovitz, Loke et al. 2004). The authors attempted to provide a general model to help thinking about and describing the context and develop context-aware applications. This work will be later, the basis for several researches of Zaslavzky and Boytsov (Boytsov, Zaslavsky et al. 2009);(Boytsov and Zaslavsky 2010);(Boytsov and Zaslavsky 2011). Boytsov and Zaslavsky presented the CALCHAS system, which offered context prediction, and used an extension to the context space theory to provide proactive adaptation.

This approach addressed the context prediction problem in a general sense. In context spaces theory several methods were tested and used for reasoning about the context. The authors judged sequence technique as the most prospective prediction approach.

For adaptation mechanisms, algebraic operations on situations and some logic-based methods were developed for reasoning in terms of situations(Abowd, Dey et al. 1999).

This work had presented a general framework model, included an adaptation approach based on prediction but did not propose a new formal or a generic prediction method.

In her work on services prediction, **Salma Najar**, offered a mechanism of discovery and prediction guided both by context and user intent (Najar, Pinheiro et al. 2014). She used semantic similarity techniques. The system is based on the implementation of a matching algorithm, which computes the matching degree between the intention and the current context of the user and the set of semantic services described accordingly. OWL-SIC (OWL-S Intentional & Contextual) is an extension of OWL-S (Web Ontology Language-Semantic, is ontology, within the OWL-based framework of the Semantic).

The similarity approach required historical data, to select and recommend services that are not always available. In fact, it requires a first phase of collection to get enough data which will be processed thereafter. Intentional approach provided by Najar (Najar, Pinheiro et al. 2014) was a user-centered approach, but can generate conflict: for instance a problem of interoperability between services. Indeed, two compatible intentions do not necessarily map to two technically compatible services. This work also proposed, a conceptual framework focused on services prediction.

Joao.H et al (da Rosa, Barbosa et al. 2016) proposed new framework including a prediction-algorithms library. They named the proposed model ORACON. The architecture of this model is based on the Model-View-Controller (MVC) design pattern. It has three layers, two agents, one library of prediction algorithms, External Histories, External Ontologies, and External Applications. ORACON proposed prediction of entities. An entity, in this sense, can be a living being, an object or even a location. Each entity can have many applications, modeled as External Applications, which can interact with the model in order to obtain predictions.

This work focused more on the framework; it didn't propose a specific prediction approach. There prediction algorithm library contains four prediction approaches: alignment, enhanced alignment, semi-markov and collaboration (da Rosa, Barbosa et al. 2016).

This proposed model was an interesting work which can be enhanced with many extensions to improve the performance, increase the accuracy of classification and optimize the processing time.

Stefan Föll et al (Föll, Herrmann et al. 2011), proposed a PreCon as a multi-dimensional context predicting method, composed of three parts: a stochastic model to represent context changes, an expressive temporal-logic query language using CSL (continuous stochastic language) and stochastic algorithms to predict the context. The model based on user behaviour was presented as a SMC (Semi-Markov Chain).

This work was the unique formal work using the CSL as a query language of the system, and a Semi-Markov Chain. There is also another work that had tried to automate the recognition of activities using the LTL formalism with a model checking (Magherini, Fantechi et al. 2013).

They concluded their work, noting that a probabilistic extension using a PCTL can increase the expressive power of the formal core.

We found this to be the most relevant work and we based our approach off it, specifically in a model checking verification. We use PCTL formalism and include action in our model to get a more descriptive model.

5.2.1 Synthesis

Table 5.1 Comparative overview of context prediction research work

	Adaptive strategy	Context formal presentation	Low and high context level	Learning capability	Prediction technique	Framework proposed
Mayrhofer (2004)	No	No	No	Yes	Sequence prediction approach	Yes
Sigg 2008-2010	No	No	Yes	Yes	Trajectory prolongation	Yes
Meiners 2010	Yes)	No	Yes	Yes	Bayesian	Yes
Boytsov 2011	Yes	Yes	No	Yes	Sequence predictors (the most perspective approach)	Yes
Stefan Foll-PreCon 2014	No	No	Yes	Yes	Temporal query prediction	No
Salma Najar 2014	Yes	No	No	Yes	semantic similarity (discovering)	Yes
Joao.H. 2016	Yes	Yes	Yes	Yes	alignment semi-markov	Yes

Table 5.1 summarizes a comparison of the related works. As we can see the majority of works don't support formal representation of the context, low and high context level. They

focused more on providing a framework including a predictive module, rather than on the prediction module itself. The essential part of a prediction model being the approach used in the prediction process itself.

Ubiquitous environments are highly dynamic, that is, applications can interact with a great number of different and unknown applications all the time (Wagner, Barbosa et al. 2014), (Miraoui 2014) (Bohn, Coroamă et al. 2005). Hence, it is essential to define a formal representation for the context, so that different systems can easily communicate. Thus, specifying a context representation is considered a key feature for model prediction. This is why, we choose a formal context representation based on a logic perspective (Ameyed, Miraoui et al. 2016). Also, we build our model in a temporal logic formalism providing clear formal semantics by using a probabilistic temporal logic (PCTL) and we propose a new probabilistic-labelled transaction model Model-LPTM.

One might also conclude that the prediction approaches supported by previous works compute the most probable future context, based on simple uni-dimensional context information. Existing systems do not allow a formal context prediction through temporal-semantics and multidimensional processes.

Our work proposes to investigate the application of probabilistic temporal logic as a powerful formal presentation for context prediction. It also proposes a formal prediction approach based on temporal logic in a multidimensional context space and on a new formalism that integrates probability and labeling; which provide a new probabilistic labelled transaction model thus helping effective context-aware prediction.

5.3 Our approach

5.3.1 Temporal logic in the context aware system

Time is a very exciting subject. We are moving through time continuously and in order to survive and manage ourselves we constantly have to make temporal-logic-based decisions. In our daily lives we manage our day involving time-dependent information e.g. when to go to the dentist? When a meeting is to be held? Etc. With the rise of ubiquitous systems (which ideally aim to provide a smart user-focused service; like reminder services, assisted-living services and more), temporal analysis and reasoning appear best-suited to ensure the proper functioning for this kind of system. Temporal logic can also be used as a programming language. The basic paradigm is to review the past and then take action in the future. Abstractly we have an initial state, and certain actions that can be performed on a given state if it satisfies a certain set of conditions. Performing an action on a state produces a new state.

We have defined a variant of TL (temporal logic) as a language for the specification of to describe each situation and its related context. In general, TL has been developed and applied as a formalism for reasoning about the ordering and quantitative timing of events.

Several formulations have been proposed to satisfy the needs of different contexts. TL may be classified according to the underlying nature of time: linear temporal logic LTL and computational tree logic CTL.

LTL, CTL and CTL* can express qualitative properties of a system. Real systems such as a pervasive system, however, are quite often characterized by non-deterministic behaviour and this because of the human presence. In order to provide efficient services, to be user-centric and more realistic, those systems should be attuned to the unpredictable behaviour of humans. Taking probabilities into account, in addition to non-deterministic behaviour, would expand this aspect of the system allowing the quantification of unpredictable behaviour, if the specification holds with an arbitrary probability value and within a given time limit.

We propose to use PCTL, which had the expressive power of probabilistic temporal logic (it introduces probability to extend CTL which is inadequate in dealing with a real-life system like a ubiquitous computing system) (Figure 5.1).

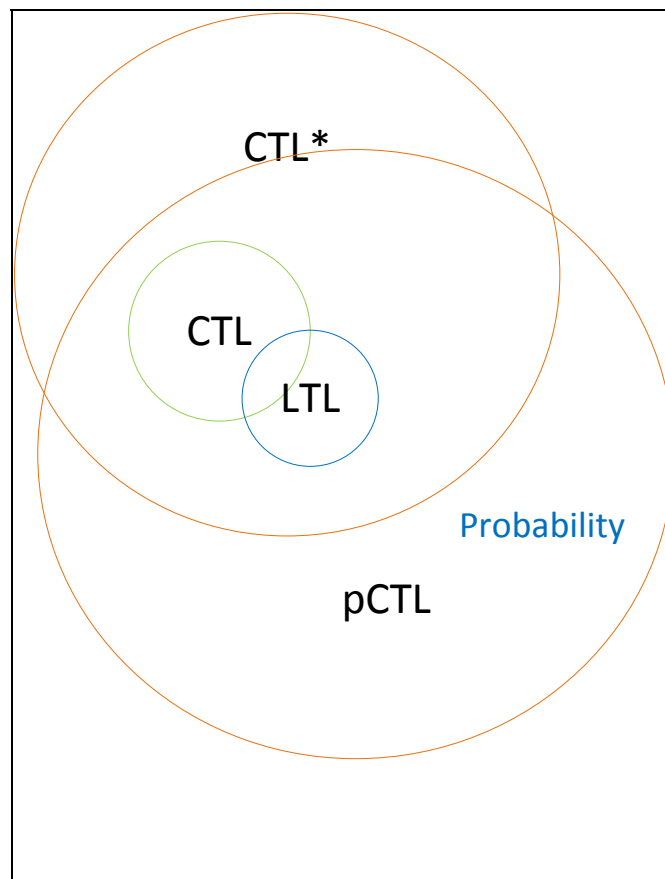


Figure 5.1 Expressivity CTL vs LTL vs CTL* vs PCTL

5.3.2 Probabilistic temporal logic specification

Temporal logic extends traditional modal logic to allow the description of when a formula is true. That is, rather than just “necessity” or “possibility”, a formula may be true at the next point in time or at some other point in the future.

Branching time logic, such as Computation Tree Logic (CTL) (Clarke 1999), enables the choice of a path among multiple possible paths in a tree structure describing probable future events. So that, each choice has to mirror the possible set of behaviours starting from the current state. As opposed to a linear-time temporal logic, for which, there is only one possible future path, we can express whether a property holds for all possible paths (*A formula*), or if there exists at least one path for which it is true (*E formula*). The values of these formulas are determined with a Kripke structure: a graph with a set of states, transitions between states, and labels indicating which propositions are true within the states.

We will use a probabilistic extension of CTL, Probabilistic Computation Tree Logic (PCTL) (Kwiatkowska, Norman et al. 2007), (C. Baier 2016) as it allows probabilistic state transitions, as well as explicit deadlines for when a formula must hold.

Our PCTL syntax is based on the syntax and semantics proposed in (Kwiatkowska, Norman et al. 2007), (C. Baier 2016). For the sake of clarity, some specific notations as well as the underlying probabilistic model have been slightly modified from the original syntax presented by those papers, in order to adapt them to our work context.

In this section we present our proposed model for the context prediction problem based on real-world situation and the related features; which represent the contextual information of each situation e.g. (location, time, occupation, ambient information, sound, temperature ...) etc.

The following Figure.5.2 summarizes our approach. It is based on PCTL formalism, a probabilistic labelled transition model which will be detailed later (section 5.3.4). The context prediction is based on model checking, which will return the future situation and its probability.

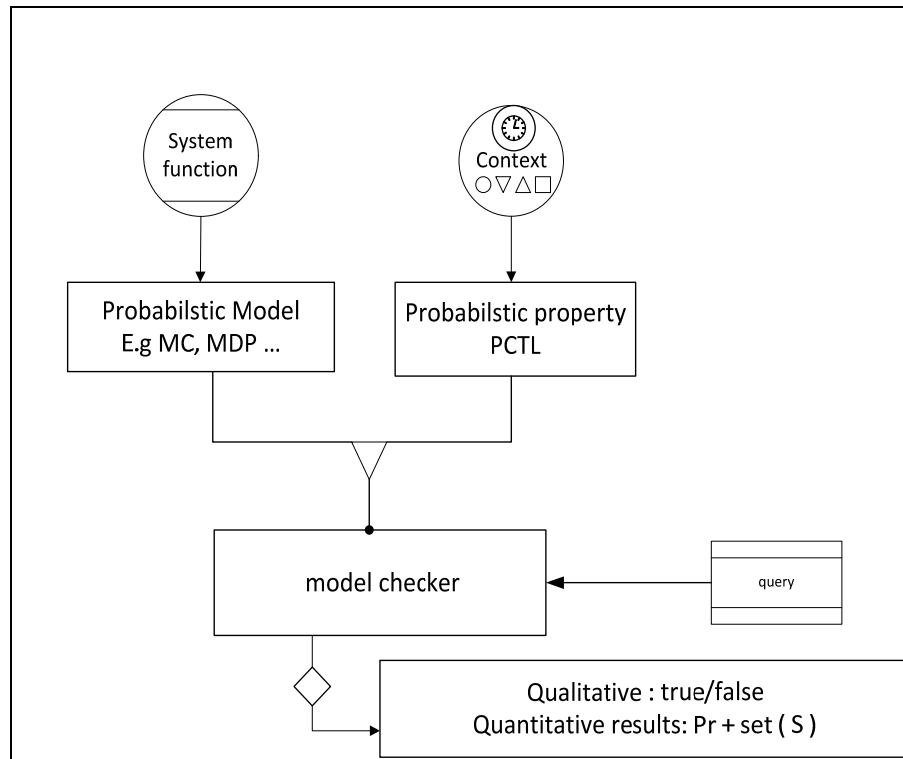


Figure 5.2 Overview of our approach

5.3.2.1 Formalism

a. Context

Definition 1

In order to specify this situation-context, let $s = (c_1, c_2, \dots, c_n) \in S$, s being an n -dimension vector of context information described by a preposition or a combination of prepositions, where each component c_i of s is of a specific context type C_i (e.g. <location>, <occupation> ...). A state s can be multidimensional and expresses composite contextual data describing the features of a specific situation; e.g. $s = ((\text{meeting-room}) \times (\text{power-point}) \times (\text{occupation} = 5))$ designates the presentation situation on a meeting room.

For each new combination of context information (c_1, \dots, c_n) that has not been observed before, is detected, a new state s will be inserted into the model and labeled with (c_1, \dots, c_n) .

For more details about our context logic-based modelling we refer to our previous work on this topic (Ameyed, Miraoui et al. 2016).

b. Path and state

The prediction semantics is based on PCTL syntax. For this let $p \in [0,1]$ be a probability, let $t \in \mathbb{R}^+$ be a time bound, and let (C_i, c_i) be a contextual value c_i of type C_i as defined earlier.

Definition 2

Path formulas express the properties and behaviour allocated to paths.

$$\varphi = X^{\leq t} \Phi \mid \Phi 1 \mid U^{\leq t} \Phi 2$$

State formulas express the properties and behaviour allocated to states

$$\Phi: = tt \mid ff \mid (C_i, c_i) \mid (A, a) \mid \neg \Phi \mid \Phi 1 \vee \Phi 2 \mid \Phi 1 \wedge \Phi 2 \mid P \sim p(\varphi)$$

Where $c_i \in 2^{AP} \setminus AP$ a set of atomic proposition describing situation context (e.g. location: <meeting room>, light: <bright>, occupation <3>, application-running: <power-point>), $a \in A$ is a finite set of actions, and \sim is a comparison operator $\sim \in \{<, >, \leq, \geq\}$ and p is a probability threshold $p \in [0, 1]$.

Path quantifiers as in PCTL are built from one of the temporal modalities: X (next) or U (until), (Table.2). t , is a time constraint defining an upper bound on a time interval to describe the duration of a situation, the subsequent transition and when an action will be active.

Tableau 5.2 Paths quantifiers

Quantifiers over paths	
A ϕ – All	ϕ has to hold on all paths starting from the current state.
E ϕ – Exists	There exists at least one path starting from the current state where Φ holds.
Path-specific quantifiers	
G Φ – Globally	Φ has to hold on the entire subsequent path
F Φ – finally	Φ eventually has to hold
X Φ – Next	Φ has to hold at the next state
Φ U ψ – Until	Φ has to hold at least until at some position ψ holds. ψ will be verified in the future

Considering Φ a state formula expressed as a pair (C_i, c) , which describe the type of context and the specific context value in this state (e.g: location, meeting-room).

We leverage these operators to analyse the future context behaviour;

- **F** is the Eventually operator used to verify if a condition ϕ eventually has to hold in any state from s somewhere on an subsequent path in the model.
- **G** is the Globally operator and it can be used to check if the condition ϕ holds in every state on all subsequent paths starting in s .
- **X** is the Next operator: It evaluates a condition Φ on all immediate successor states to the current state s . It has to hold at the next state (this operator is sometimes noted N instead of X). Since we focus in immediate prediction, we will build our prediction model on this operator in this paper.
- **U** is the Until operator and expresses that Φ_2 will be verified in the future. And Φ_1 has to hold starting at the current state at least until at some further position Φ_2 holds.

The PCTL state formula $P \sim p(\varphi)$ asserts that, under all schedulers, the probability for the event expressed by the path formula φ meets the bound specified by $\sim p$. The probability bounds “ $\sim p$ ” can be understood as quantitative counterparts to the CTL path quantifiers \exists and \forall .

5.3.3 PCTL semantic

Definition 3

Let $M=(S,AP,L)$ be a PCTL model, s is a state $\in M$, AP a set of atomic prepositions, and Φ is a PCTL formula. The satisfaction relation is noted as

$$M, s \models \Phi$$

Let s be a state, $s \in S$ we can define the satisfaction relation for state formulas as follows

$$M, s \models \text{true} \quad \forall s \in S$$

$$M, s \models ci \Leftrightarrow ci \in L(s)$$

$$M, s \models \neg \Phi \Leftrightarrow M, s \not\models \Phi$$

$$M, s \models \Phi_1 \wedge \Phi_2 \Leftrightarrow M, s \models \Phi_1 \text{ and } s \models \Phi_2$$

$$M, s \models \Phi_1 \vee \Phi_2 \Leftrightarrow M, s \models \Phi_1 \text{ or } s \models \Phi_2$$

$$M, s \models P \sim p(\varphi) \Leftrightarrow P\{\pi \in \text{Paths}(s) \mid M, \pi \models \varphi\} \sim p$$

The satisfaction relation for path formula is defined inductively as follows:

$$M, \pi \models X \Phi \Leftrightarrow \pi = s_0 \xrightarrow{a_0, t_0} s_1 \xrightarrow{a_1, t_1} \dots s_{n-1} \xrightarrow{a_{n-1}, t_{n-1}} s_n \text{ and } M, s_1 \models \Phi$$

$$M, \pi \models \Phi_1 U \Phi_2 \Leftrightarrow \pi = s_0 \xrightarrow{a_0, t_0} s_1 \xrightarrow{a_1, t_1} \dots s_{n-1} \xrightarrow{a_{n-1}, t_{n-1}} s_n \text{ and } \exists k. M, s_k \models \Phi_2 \text{ and } \forall j < k. M, s_j \not\models \Phi_2$$

5.3.4 Labelled Probabilistic transition model : Model-LPTM

A pervasive system follows various behavioural patterns depending on user's behaviour. Those patterns cannot be described in a deterministic way. Hence, our choice of a probabilistic non-deterministic model. In the following, we give a description of this model and our approach to predicting the next situation using this formalism.

We represent our model LPTM as a transition system which combines probabilistic choice as in Markov chains with non-deterministic choice. We define the model with a timed probabilistic transition based on models defined in (Kwiatkowska, Norman et al. 2007), (C. Baier 2016). The model integrates time and action and will be presented as follow:

Definition 4

Let LPTM be a Kripke (S, A, P, L) : a labelled transition probabilistic model defined as follow:

S : finite set of states where $s \in S$ and $s_{init} \in S$

Act: finite set of actions where $a \in A$ and $A \subseteq \text{Act}$

$L: S \rightarrow 2^{AP}$ state labelling function assigning to each state one or several atomic prepositions $\in AP$

$P \subseteq S \times A \times \mathbb{R}^+ \times \text{Dist}(S)$ is the function assigning a probabilistic transition distribution, such that if $(s, \delta t, a, \rho) \in \text{Dist}(S)$ and $\delta t > 0$ after a span time Δt in a situation s was spent and a is an active $\in A(s)$ then ρ is a point distribution.

As for probabilistic systems, we can introduce paths for timed probabilistic systems except that transitions are now labelled by a (duration, action, distribution) tuple. Each transition is labelled by a tuple $(\delta t, a, \rho) \in \text{Dist}(S)$, where:

- δt is the time span between s_i and s_j (Section 5.3.7);
- $P(s_i, s_j)$ is the probability assigned to the path transition between s_i and s_j (Section 5.3.5);

- $a \in A(s_i)$ is an action active between two states s_i and s_j . (Section 5.3.6).

Our contribution using this model consist in considering every $s_i \in S$ as described by a set of context parameters ($c_i \in C_i$) such that $L(s_i) = c_i$ and an action for a transition path with a temporal duration constraint δt .

To avoid transient states, we choose to integrate them as proposals in paths. Thus, the path describes a transient context (as an accomplishment action or activity action. (See section 5.3.7) that can be part of the next state. This makes the modeling more context-aware and proactive.

Using this LPTM we can describe a pervasive environment according to the user's behaviour; at each spatiotemporal interval, we have an active state describing a specific context $s_i \in S$. While the user (e.g: walking, driving, be...) or the environment and the system environment (running process, etc.) act, the context changes and the LPTM moves to the new state $s_j \in S$ expressing the property of new context. This successor state s_j is visited with a probability $p(s_i, s_j)$. Before leaving the current state s_i , the context doesn't change and stay active for a limited duration of time Δt spent in s_i .

Example: model (Figure 5.3)

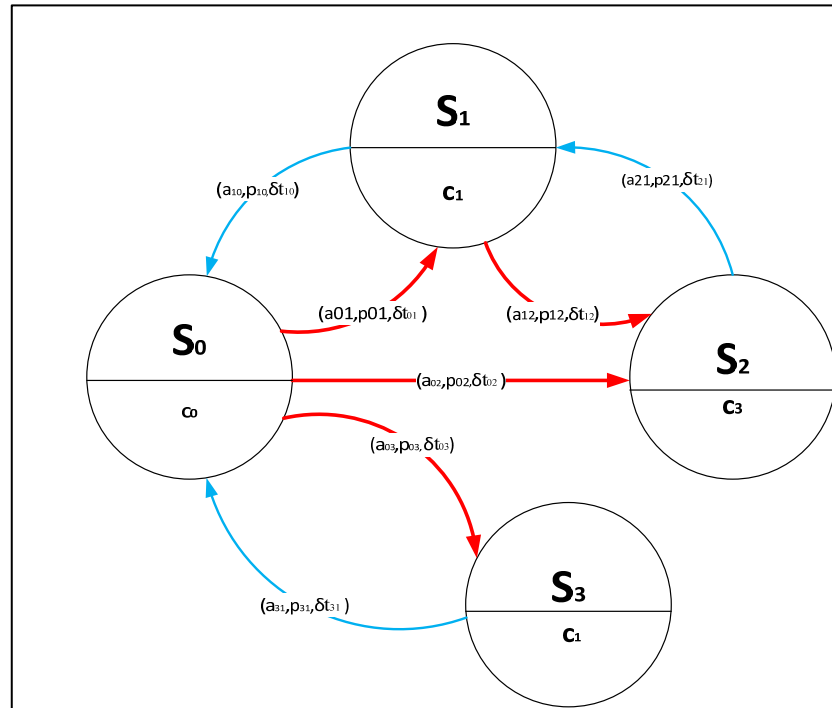


Figure 5.3 Transition model

Explanation:

To lead the next situation from the current situation i to the next one j we counts:

- a_{sn} represent an action active for a given state (e.g a_{01} describe the active action from S_0 to S_1);
- δt_{ij} represents the time span between s_i and s_j (e.g δt_{01} ; describes transition duration from S_0 to S_1),
- P_{ij} refers to the transition probability from the situation s_i to the situation s_j such that $\sum_j p_{ij} = 1$.

5.3.5 Transition Probability

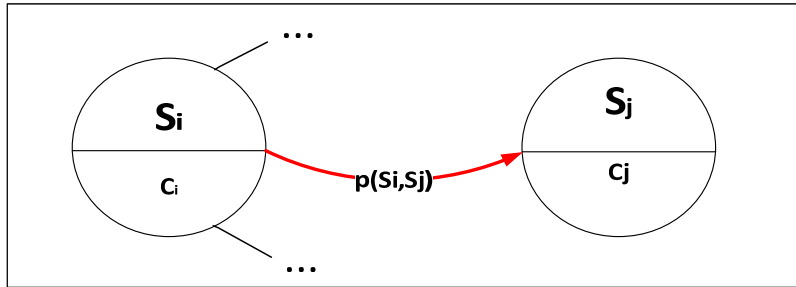


Figure 5.4 state transition probability

For each transition (S_i, S_j) , the transition probability will be:

$$p(s_i, s_j) = P(X_{n+1} = s_j \mid X_n = s_i)$$

Where X_n is the random variable that models the stochastic behaviour at the current state and X_{n+1} model the stochastic behaviour at the next state.

Recall that the formulas are defined with regard to a probabilistic structure, as described earlier. While our structures consist of labeled states and path, they only imply that it is possible to transition from the state at the tail to the state at the head with some non-zero probability.

We express our model as a causal model. In this paper we assume a dependent relation between current state and the next one. The probabilistic transition depends only on the current state s_i and is independent of all previous state changes.

The transition probability and the set of prepositions describing contextual feature situations can be estimated and deduced from the history of past trace of state transitions and their linked contextual features.

As in statistic computation, let the transition weight be ω_{ij} , which define the number of transitions observed from s_i to s_j . The transition probability is calculated as follows:

$$p(s_i, s_j) = P(X_{n+1} = s_j | X_n = s_i) = \frac{\omega_{s_i, s_j}}{\sum_{s_n \in S} \omega_{s_i, s_n}}$$

The probability of transition between two states is the ratio of the number of observed state transitions from s_i to s_j to the number of all observed transitions from s_i .

Example:

We have a current state s_0 that can lead to any of the immediate next states as in Figure.5.5 as a distributed probability

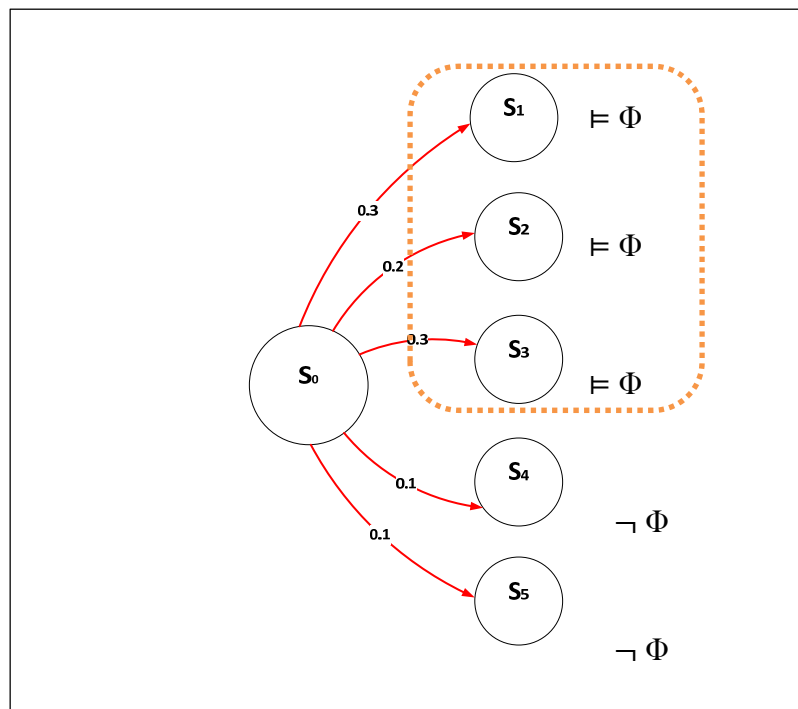


Figure 5.5 State transition

The probability without any constraint of time or action to lead to any next state when Φ will be verified as a Next (optimal) in S_1 , S_2 and as a Next (all) in S_3 be S_3 or S_1 .

5.3.6 Action

Observing the system's and user's behaviour, we also noted information describing actions which influences a service process and make change in a situation in the feature context. Based on reference work discussing the linguistics of time and the semantic verbs and time (Vendler 2005), (Ratté and Ratté 1994), these actions can use the aspectual verbs according to the categories in Table.3.

Tableau 5.3 The four aspectual categories

	Expressivity	Dynamic	Durative	Telic (bound)
accomplishment	Describing durative action ending by a culmination point	Yes	Yes	Yes
activity	Describing durative action	Yes	Yes	No
state	Often durative	No	Yes (temporary state) No (permanent state)	Yes
achievement	Change of state near punctual duration	Yes	No	Yes

In our model we can use accomplishment and activity to describe a transition over a path and a state and achievement in a situation (node).

In our computation over the proposed model we use the accomplishment-action on the path because we are reasoning in a dynamic system with a time bound and we count the durative actions in a bound time during a transition. We can label our graph with state-action and achievement to clearly describe a scenario or an example.

In our model, actions depend on transition and describe a transition over a special path. The set of actions available at $s \in S$ is denoted by $A(s)$. For each action $a \in A(s_i)$, the probabilities can be estimated as other observations from the history of past trace. We count the probability of transitioning from s_i to s_j under the action a , and we denote this probability by $\alpha_a^{s_i}(s_j)$. We refer to (Lahijanian, Andersson et al. 2011) for more details about computation in mapping and learning steps.

Example:

We have a set $A(s_0) = \{a_1, a_2\}$ and a transition and s_0 can lead to any of the immediate states as in Figure.5.5.

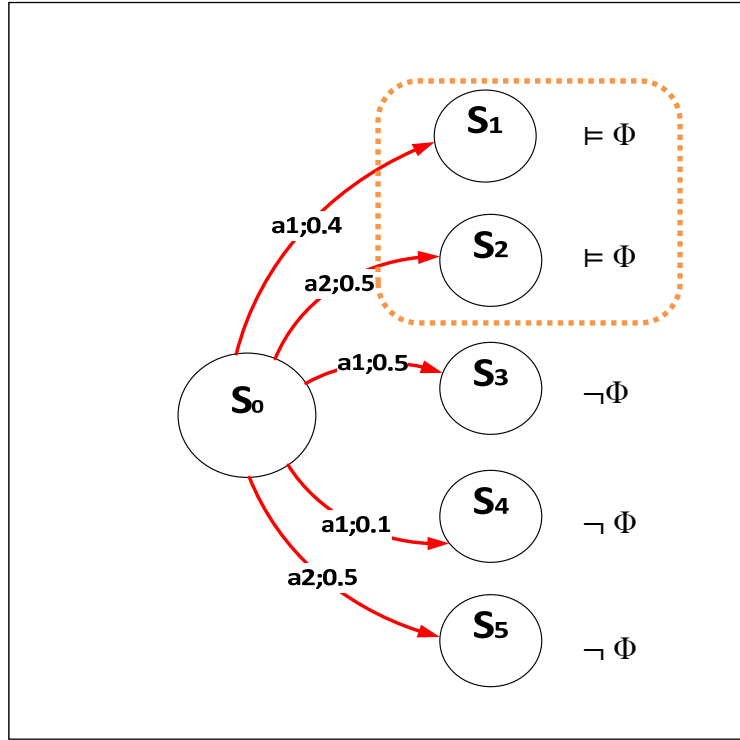


Figure 5.6 Action transition

In this example the probability next Φ to occur with any action $a \in A(s_0)$ is

$$\frac{\sum_{s_j \in S \wedge s_j \models \Phi} \alpha_a^{s_i}(s_j)}{\sum_{s_j \in S} P(s_j, s_i) \cdot \sum_{s_j \in S \wedge s_j} \alpha_a^{s_i}(s_j)} = 0.45$$

the optimal next will be the path with a strategy probability ≥ 0.45 in this case that will be the transition (S_0, S_2) under the action a_2 .

5.3.7 Span Time duration

In this section we discussed the predicted starting time for the next state. We will show how we can estimate the time span between s_i and the next s_j . The time was considered in our model as the constraint parameter for states as well as transitions (path), as described in our previous contextual definition and model. Every situation has a time interval describing its start time and end time which can be useful as a learning data base (Ameyed, Miraoui et al. 2016), (Ameyed, Miraoui et al. 2015).

We express the time span as a probability function where μ and σ are the mean and standard deviation values, calculated from the time span. In order to limit the computation, we consider in our work, only the observation falling with standard deviation.

$$f(\delta t_{ij}, (\mu, \sigma)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\delta t_{ij} - \mu}{\sigma}\right)^2}$$

The Figure.7 gives an example of transition time span: the typical time span falls in the following range

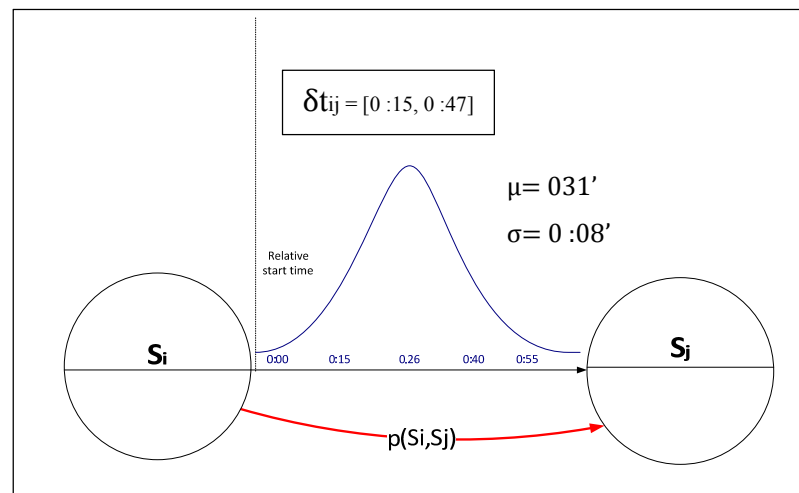


Figure 5.7 Transition span time

We model the time span as a random variable D_n expressing the time spent between s_i and s_j . To figure out D_n , we observe the time periods spent between consecutive states transitions

and we associate an individual distribution to every transition between s_i and s_j . Formally the distribution can be presented as:

$$\Delta_{ij}(\delta t) = P(D_n = \delta t \mid X_{n+1} = S_j, X_n = S_i).$$

The cumulative distribution is given by Δ_{ij} which is given as $\int_0^b \Delta_{ij}(\delta t) d\delta t$ and can be computed as the sum of probabilities associated with consecutive intervals up to a desired upper time bound b . The probability of a time span to lie within the interval $[a, b]$ can be derived from the cumulative distribution as $\int_a^b \Delta_{ij}(\delta t) d\delta t$.

5.3.8 Immediate-context Prediction processing

The state space can be traversed by going from one state to the next as allowed by transitions among states. The resulting series of visited states (path) models one possible spatiotemporal behaviour of context. For context prediction we start at the state $s_i \in S$ occupied in the real world and we evaluate the possible path starting at s_i and leading to the next state s_j . Our state and path follows the PCTL semantic as explained in (section 3.3).

In our model we can evaluate a satisfaction relation for the path formula as follows:

$$X_{n+1} \leftarrow \operatorname{argmax}_{X_{n+1}} P(X_{n+1} = s_j \mid X_n = s_i, a \in A(s_i))$$

The path formula φ is satisfied after Δt unit of time elapsed in a situation s and under an action a if and only if the probability $P((s, a, \Delta t) \models \varphi)$ satisfies the threshold $\sim p$

In our case, we need to be able to verify that a given state satisfies the context's state preposition $\Phi = (C_i, c)$ (as describe in section 3.3). We need also to consider the temporal operator Next $P \sim p [X\Phi]$ and to define its probability computation.

Using a PCTL we can investigate the reachability properties using the Next operator, evaluating a condition state formula Φ , expressed over the contextual information (C_i, c_i) , on all immediate successor states s_j of the current situation s_i . Using this reasoning, the system is

able to predict a variety of information about the context (e.g. the next location, the next activity, what time the user finishes work, at what time the next meeting starts, what is the optimal strategy to lead the next situation, etc.). The high threshold probabilities according to a special action describing a transition reduces the number of false prediction prepositions and make the prediction more efficient and more context-aware.

We can derive the prediction based on next operator in PCTL as explained in the next subsection using the verification algorithm in a model checking based on symbolic method (Kwiatkowska, Norman et al. 2007) (Kwiatkowska, Norman et al. 2003).

5.3.8.1 Computation for PCTL next operator

In this paper we focus on immediate prediction. Thus, we will only use the next operator. In future work we might extend our approach with the two more temporal operators: (i) Until: $P \sim p[\Phi_1 U \Phi_2]$ and (ii) Bounded Until : $P \sim p[\Phi_1 U^{\leq k} \Phi_2]$ which can be useful for a long term prediction.

The Next operator restricts the space of satisfaction property of path formula ϕ to the immediate successor the next state s_j of the current state s_i .

We need to determine the Next (optimal) $\Phi = P_{\max=?}(\text{Next } \Phi)$ which is the maximum probability satisfying Next Φ .

$$X_{n+1} \leftarrow \operatorname{argmax}_{X_{n+1}} P(X_{n+1} = s_j | X_n = s_i)$$

Or the all Next (all) $\Phi = P_{\forall p}(\text{Next } \Phi)$; here we can find all the policies that satisfy the next state with Φ property, where:

$$P (X_{n+1} = s_j | X_n = s_i, a \in A(s_i))$$

$$= P_{\langle a \rangle} \left(X_{\Delta t}^{\leq \delta t_{ij}}(\Phi) \right)$$

$$= \frac{\sum_{s_j \in S \wedge s_j \neq \Phi} P(s_j | s_i) \cdot \sum_{s_j \in S \wedge s_j \neq \Phi} \alpha_a^{s_i}(s_j) \cdot \int_{\Delta t}^{\Delta t + \delta t_{ij}} T_{ij}(\delta t) d\delta t}{\sum_{s_j \in S} P(s_j | s_i) \cdot \sum_{s_j \in S \wedge s_j} \alpha_a^{s_i}(s_j) \cdot \int_{\Delta t}^{\infty} T_{ij}(\delta t) d\delta t}$$

The optimization function $\log (P (\Phi | \lambda))$ is proposed to avoid data overflow in the computation of feedforward probability.

The prediction approach is based on the traces contained in the stochastic user model. The traces are used as a search space of possible context changes. Information about the recent sensed context changes (current state's context) is used to condition the prediction on what the optimal Next might be expected in the immediate future. Using a model based on statistical knowledge, the predictions in our approach, work as a scanning process in a stochastic transition system to find the Next verifying the property expressed in the formula. A component diagram the prediction model can be represented as shown in Figure 5.8.

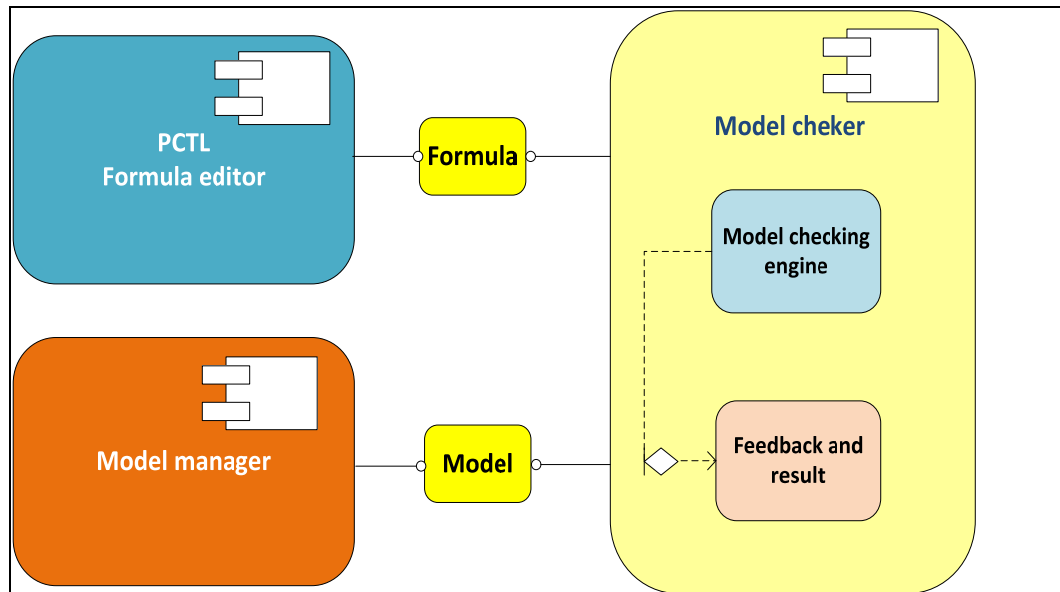


Figure 5.8 Component diagram of LPTM system.

5.3.9 Use case and test

In this section, we present the experimental results for our model. Before getting into evaluation of the prediction model we describe the data set we used (Kadouche, Pigot et al. 2010), (Chikhaoui, Wang et al. 2010).

We use a real-world context traces from Domus smart home case study. The Domus smart home is one bedroom apartment mounted inside the University of Sherbrooke. The apartment is equipped with different types of sensors. During the experiments, users have participated to evaluate the early morning routines, which correspond to the basic everyday tasks during the morning. The routine describe morning activities as follow: wake up, toileting, preparing breakfast, having breakfast and other activities. We use this study case to predict the Next activity. The activities we consider in our simulation are as follow: wake up, use toilet, preparing breakfast, having breakfast.



As a simulator tools we use Petri nets, it is a formal models of information flow which support timing specifications and a non-deterministic behavior for more details about a tools we refers to (Liu, Miao et al. 2013).

We first model the prediction model as shown in the Figure 5.9.

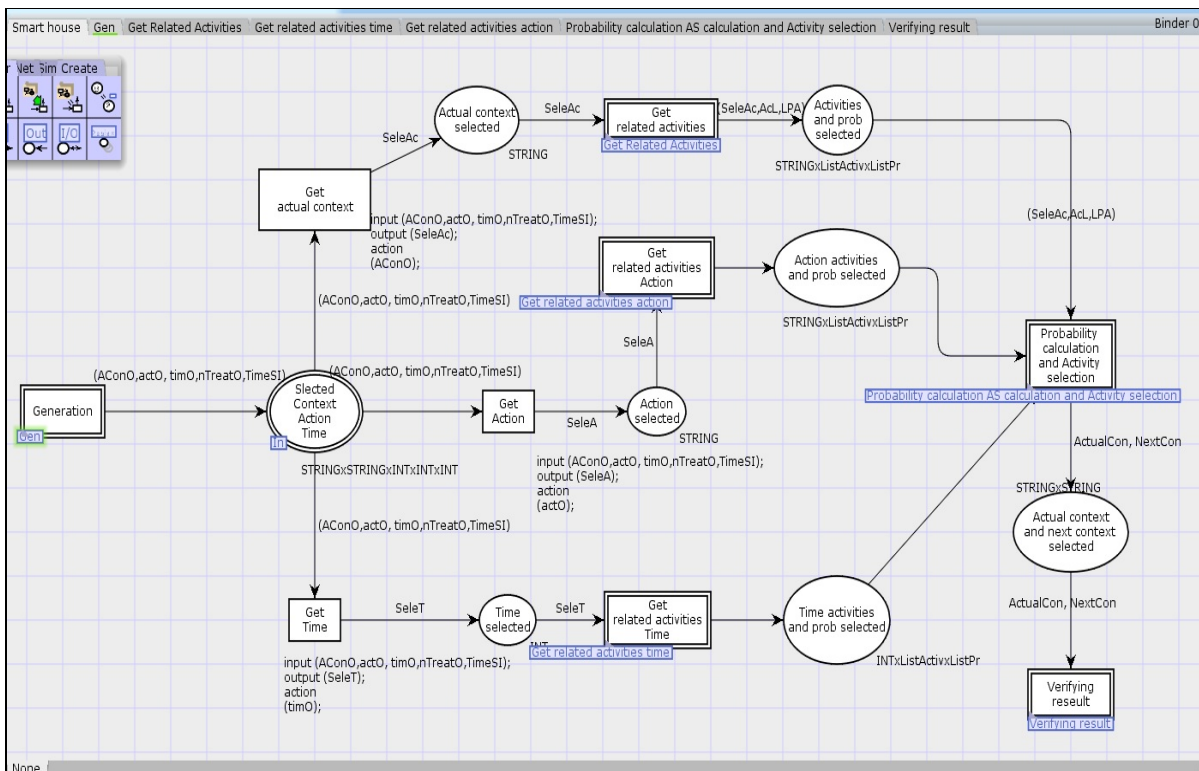


Figure 5.9 General View of Prediction Model

The model is composed mainly by:

Generation: this module generates the current context and constraints as random choice.

Get related activity: the module gives the activity probability.

Get related activity Action: the module determines the action probability.

Get related activity time: the module defines the time span probability.

Probability calculation: the module gives the probability to the most probable Next activity

The transitions between different activities are learn based on the LPTM trace Model as shown in Figure 5.10.

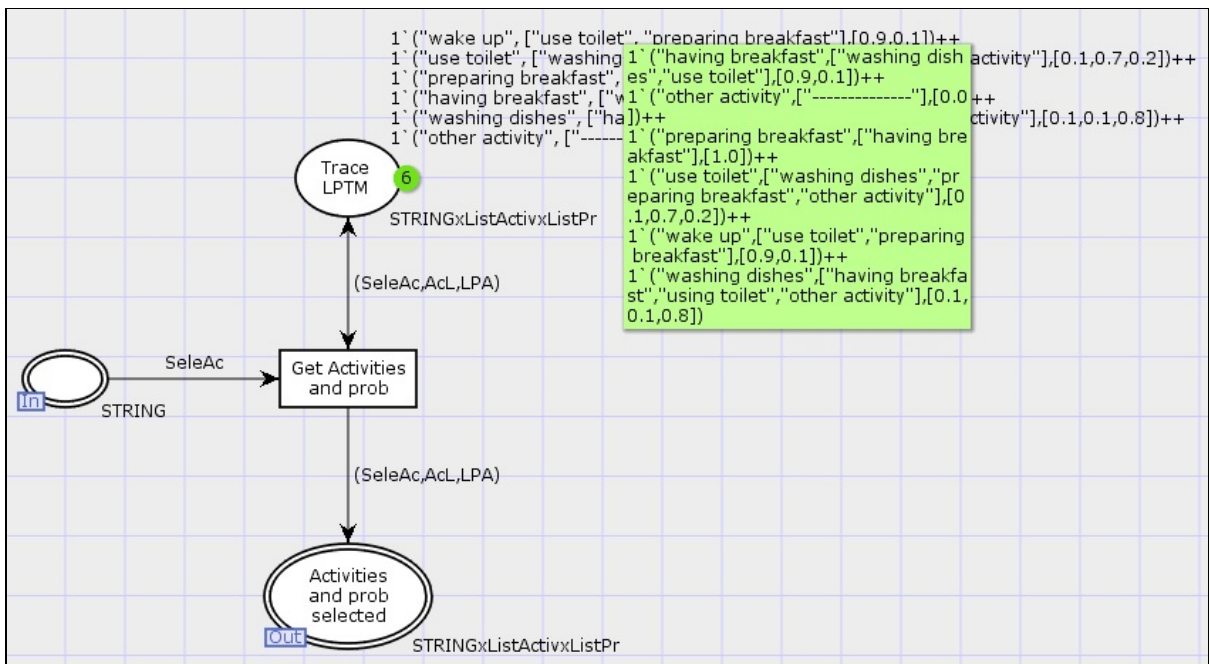


Figure 5.10 Activities transitions information's

To recognize the next activity we generate a random for variety of: activity (context value) time and action as shown in the Figure 5.11.

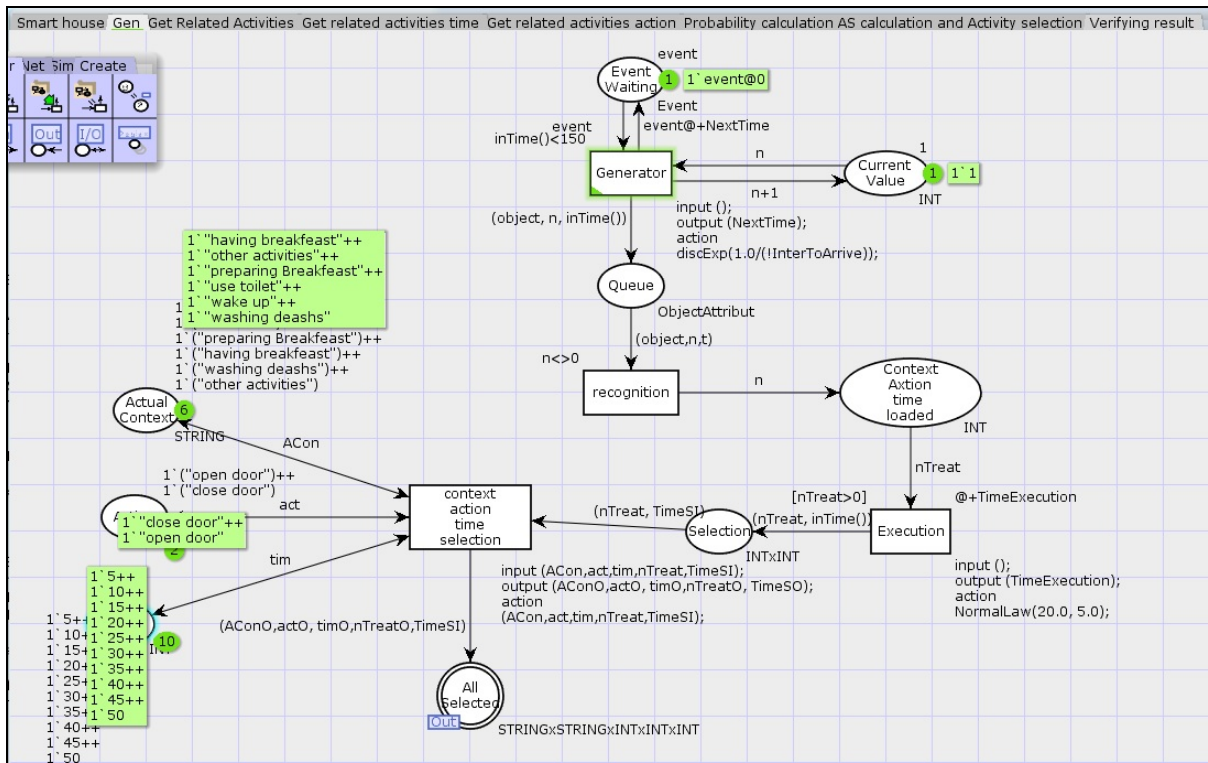


Figure 5.11 Action-time generators

Finally after getting the Next activity identified, we evaluate our result based on real flow evidence as shown in Figure 5.12.

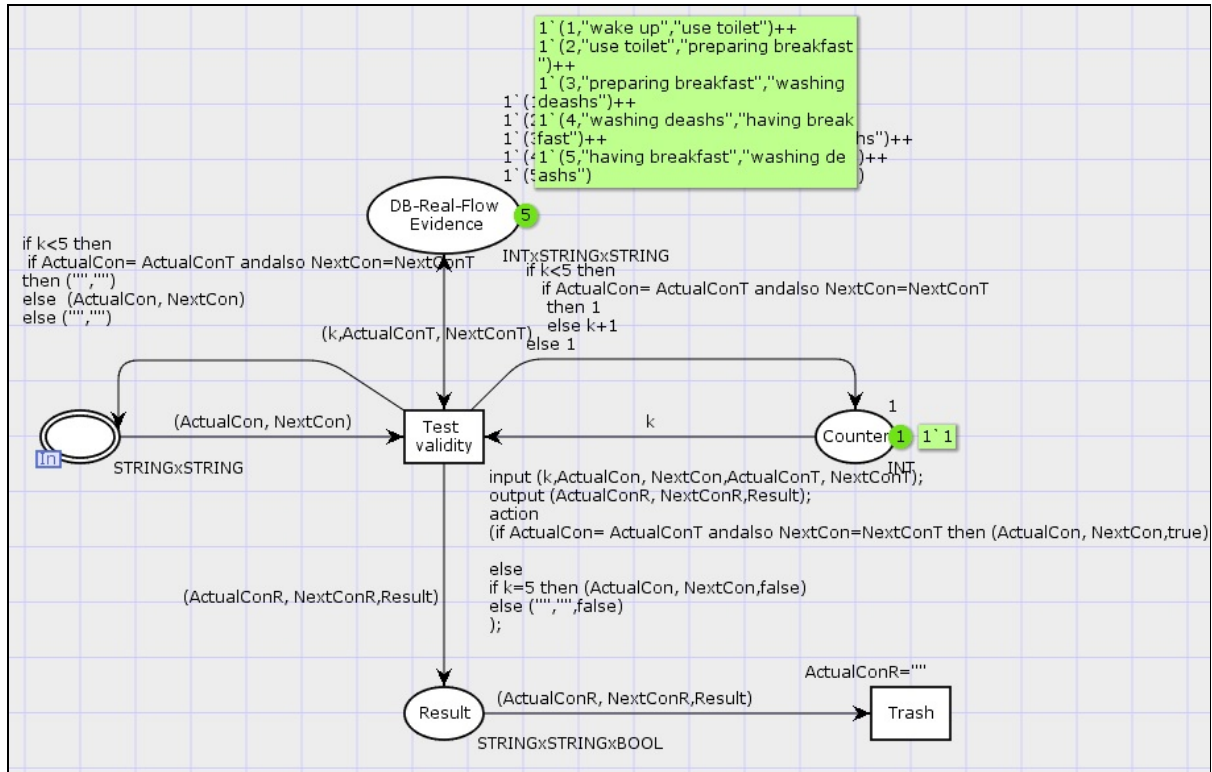


Figure 5.12 Verifying results Next activity

The diagram, in Figure 5.13, resumes the prediction results for each activity. The average of our prediction model was 65%, we get also 78% in some activity as shown in follow diagram.

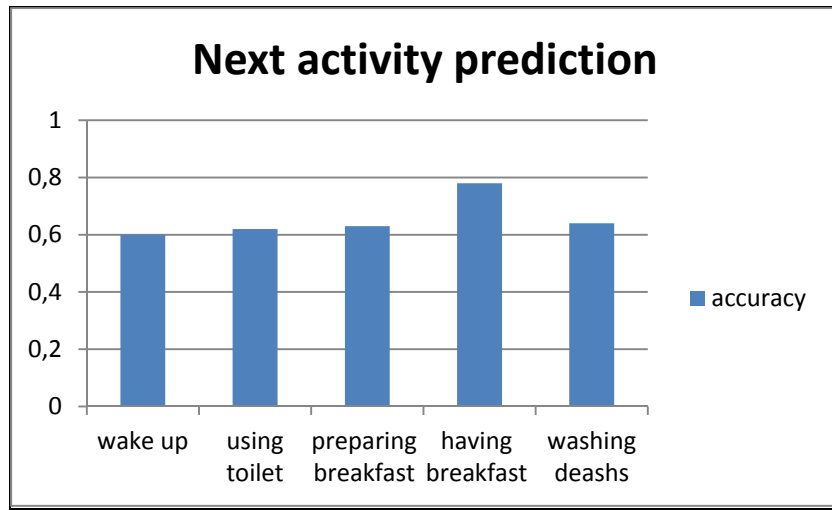


Figure 5.13 The activities prediction accuracy

5.4 Conclusion and future work

The prediction of future context has become a central element in pervasive systems to provide proactive context-awareness adaptation. However, the effective deployment of a context-aware prediction is still limited due to a semantic gap between the data provided by the physical sensing devices and the necessary information to predict future behavior of the system and its users. In this paper we have demonstrated how formal methods could be adopted to offer a formal ground to reduce this semantic gap and provide improved expressiveness via the PCTL logic. And therefore verify reachability a next-state in the future. Introducing the constraints of time and action adds logic-based expressiveness and provides a clear tracing and learning model. Thus increasing the effectiveness of probabilistic measures.

In this paper we present a new formal approach using probabilistic temporal logic and model checking to provide an immediate prediction. Our approach allows a formal expressivity of prediction. This is useful in pervasive computing systems to deal with their inherently heterogeneous nature. The model offers a real time ability to discover a future context on multidimensional space and can handle a general context in low or high level. Adopting a PCTL as formalism, provides better expressivity to describe the nondeterministic nature of

human behavior which can provide an efficient prediction and consequently offer an adequate proactivity, fitting with the user's needs.

In future work, we will extend our research to include the long-term prediction and possibly discuss a generic framework that can support our prediction model to automating proactive adaptation based on predicted context. We will try to investigate more, the issue of semantic in action in order to be able to provide a more expressive model, inducing cognitive and linguistic support.

CONCLUSION

Les systèmes omniprésents proactifs visent à offrir un espace intelligent accessible à divers services permettant une utilisation adaptée, naturelle et conviviale. Un espace technologique qui sera capable de s'adapter contextuellement aux changements de son environnement et réagir de façon appropriée et adéquate au besoin de ses usagers. La prédiction de contexte devient un élément central dans la nouvelle tendance de ces systèmes qui évoluent vers une adaptation proactive et aussi anticipatoire. La prédiction de contexte dans ce domaine est un sujet stimulant qui porte plusieurs défis et beaucoup plus d'opportunités pour l'innovation dans les technologies ambiantes. Cerner le contexte et développer une prédiction de ce dernier, appropriée dans les systèmes diffus, est toutefois un défi. La raison est que ces systèmes fonctionnent dans des environnements fortement dynamiques, complexes et imprévisibles.

Ce travail s'inscrit sous cette question, il vise à spécifier et modéliser le contexte et la prédiction du contexte futur dans un cadre formel tenant compte de l'évolution temporelle logique de l'espace de service et le comportement de ses usagers. De nombreux travaux existants traitent la prédiction du contexte dans les systèmes diffus proactifs et montrent l'importance de développer un cadre formel et général pour le contexte et la prédiction sauf que jusque-là les approches formelles de la prédiction du contexte général sont manquantes. Suite à notre lecture de l'état de l'art, nous déduisons que le contexte dans les systèmes diffus reste vaguement défini et généralement traité dans une perspective unidimensionnelle qui ne reflète pas la réalité complexe de l'environnement des systèmes ambiants. Ainsi les approches de prédictions proposées sont unidimensionnelles et spécifiques aux applications ciblées, par conséquent une adaptation intelligente proactive et anticipatoire reste loin d'être atteignable. Et comme l'essence de l'informatique ambiante est d'offrir assistance et confort à ses usagers d'une manière invisible et conviviale, comprendre la routine et le besoin des usagers de l'espace de services est un point central pour développer des systèmes ambiants efficaces.

Dans nos travaux de recherche, nous avons défini le contexte sous une vision multidimensionnelle respectant une logique d'évolution spatiotemporelle. Nous avons aussi spécifié et modélisé le contexte dans un cadre formel logique intégrant la dimension spatiotemporelle. À la fin et pour mener à terme notre démarche nous avons spécifié et modélisé la prédiction de contexte futur dans un cadre formel et logique et sous une vision multidimensionnelle, dynamique et non déterministe. Adaptant le PCTL comme formalisme permet de fournir une plus forte expressivité de la prédiction contexte, et ce via une logique temporelle supportant l'aspect stochastique probabiliste. Ce qui permet une prédiction efficace et offre par conséquent une proactivité adéquate en raccord avec les besoins de l'utilisateur. Le modèle proposé est apte à ; (i) Supporter une prédiction générale du contexte, (ii) supporter une contrainte temporelle, (iii) fournir une logique de prédiction expressive avec un formalisme sémantique clair, et (iv) supporter un raisonnement logique temporel non déterministe, respectant ainsi la nature complexe, plurielle et dynamique de la réalité de l'espace ambiant.

Nous pensons que nos travaux peuvent contribuer à l'avancement de la recherche dans le domaine des systèmes diffus proactifs, et ce via une meilleure compréhension du contexte. L'aspect formel et logique de nos modèles de contexte et de prédiction de contexte futur, offrent la possibilité d'une réutilisation indépendamment de l'application ciblée, et ce sur un large spectre de données contextuelles à bas et/ou à haut. Notre modèle prédictif, basé sur le raisonnement logique, permet une intégration des connaissances profondes cognitives, temporelle et linguistique. Il peut, de fait, être utilisé dans les applications ambiantes d'assistance à risque comme e.g : l'assistance des personnes âgées, les handicapées, l'assistance santé à distance «e-health» et bien d'autres.

▪ **Perspectives**

Des perspectives intéressantes s'ouvrent et se construisent sur la base des résultats de notre travail, et qui dépassent l'étendue de cette présente thèse. Dans cette sous-section nous

discutons des directions pour la recherche future dans le domaine du contexte et la prédiction du contexte futur dans les systèmes diffus proactifs et anticipatoires.

Dans la recherche actuelle, seules des bases de données de contexte de durées limitées sont disponibles. Les données disponibles sont très restreintes pour pouvoir mettre en place un apprentissage statistique approfondi, multidimensionnel avec le potentiel d'étudier le comportement des utilisateurs et de l'environnement sur des périodes de temps plus longues.

L'information sur les actions est presque non existante ou définie d'une manière triviale. Mettre en place un champ d'expérimentation adéquat qui permettra une collecte d'ensembles diversifiés de données à grande échelle pour offrir une base d'apprentissage approfondie et de longues durées reste en lui-même un défi à réaliser. Ce qui permettra en premier lieu des études d'évaluation réalistes pour pouvoir comparer les performances des méthodes proposées.

Avec notre système de prédiction de contexte, nous avons exploré de nouvelles façons de prévoir le comportement de l'environnement diffus concentré sur l'évolution dynamique de comportement de l'utilisateur. Cette approche peut avoir une extension possible vers la prédiction des comportements de foule, utile, par exemple, dans une application directe dans la gestion stratégique des bâtiments intelligents et des villes intelligentes.

Toutefois, nous pensons que l'ouverture vers des modélisations hybrides du contexte et de la prédiction de contexte augmentera le potentiel des systèmes diffus proactifs du futur pour des adaptations génériques plus intelligentes.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUE

- Abbar, S., M. Bouzeghoub and S. Lopez (2009). Context-aware recommender systems: A service-oriented approach. VLDB PersDB workshop.
- Abdenneji, S. F. B., S. Lavirotte, J.-Y. Tigli, G. Rey and M. Riveill (2012). A symmetric compositional approach for adaptive ubiquitous systems. Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on, IEEE.
- Abowd, G. D., A. K. Dey, P. J. Brown, N. Davies, M. Smith and P. Steggles (1999). Towards a better understanding of context and context-awareness. Handheld and ubiquitous computing, Springer Berlin Heidelberg: 304-307.
- Abowd, G. D., M. Ebling, G. Hung, H. Lei and H.-W. Gellersen (2002). "Context-aware computing [Guest Editors' Intro.]" IEEE Pervasive Computing 1(3): 22-23.
- Akman, V. and M. Surav (1997). "The use of situation theory in context modeling." Computational Intelligence: An International Journal 13(3): 427-438.
- Aksit, M. and Z. Choukair (2003). Dynamic, adaptive and reconfigurable systems overview and prospective vision. Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on, IEEE.
- Al-Masri, E. and Q. H. Mahmoud (2006). A context-aware mobile service discovery and selection mechanism using artificial neural networks. Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet, ACM.
- Albrecht, D. W., I. Zukerman and A. E. Nicholson (1998). "Bayesian models for keyhole plan recognition in an adventure game." User modeling and user-adapted interaction 8(1-2): 5-47.
- Ameyed, D., M. Miraoui and C. Tadj (2015). "A Spatiotemporal Context Definition for Service Adaptation Prediction in a Pervasive Computing Environment." arXiv preprint arXiv:1505.01071.
- Ameyed, D., M. Miraoui and C. Tadj (2015). "A Survey of Prediction Approach in Pervasive Computing." International Journal of Scientific & Engineering Research 6(5): 306-316.
- Ameyed, D., M. Miraoui and C. Tadj (2016). "Spatiotemporal Context Modelling in Pervasive Context-Aware Computing Environment: A Logic Perspective." (IJACSA) International Journal of Advanced Computer Science and Applications Vol. 7, (No. 4, Avril 2016).

- Attardi, G. and M. Simi (1994). "Proofs in Context. KR". Principale of knowledge representation and reasoning: proceeding of the froujt international conference KR'94, San Francisco, California.
- Bacon, J., J. Bates and D. Halls (1997). "Location-oriented multimedia." *Personal Communications, IEEE* 4(5): 48-57.
- Bacon, J., J. Bates and D. Halls (1997). "Location-oriented multimedia." *IEEE Personal Communications* 4(5): 48-57.
- Baldauf, M., S. Dustdar and F. Rosenberg (2007). "A survey on context-aware systems." *International Journal of Ad Hoc and Ubiquitous Computing* 2(4): 263-277.
- Bao, J., P. R. Smart, D. Mott and D. Braines (2010). "A formal context representation framework for network-enabled cognition."
- Barwise, J. and J. Perry (1981). "Semantic innocence and uncompromising situations." *Midwest studies in philosophy* 6(1): 387-404.
- Barwise, J. and J. Perry (1981). "Situations and attitudes." *The Journal of Philosophy* 78(11): 668-691.
- Benazzouz, Y. (2011). *Découverte de contexte pour une adaptation automatique de services en intelligence ambiante*, Ecole Nationale Supérieure des Mines de Saint-Etienne.
- Bettini, C., O. Brdiczka, K. Henriksen, J. Indulska, D. Nicklas, A. Ranganathan and D. Riboni (2010). "A survey of context modelling and reasoning techniques." *Pervasive and Mobile Computing* 6(2): 161-180.
- Bhattacharya, A. and S. K. Das (1999). *LeZi-update: an information-theoretic approach to track mobile users in PCS networks*. Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, ACM.
- Bohn, J., V. Coroamă, M. Langheinrich, F. Mattern and M. Rohs (2005). *Social, economic, and ethical implications of ambient intelligence and ubiquitous computing*. Ambient intelligence, Springer: 5-29.
- Bouzeghoub, A., P.-A. Caron, S. Kaddouci, C. Lecocq, X. Le Pallec, F.-J. Ritaine and J. Rouillard (2008). *Experiments in ubiquitous computing for communities of practice using learning resources*. 2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies, IEEE.

- Bouzy, B. and T. Cazenave (1997). Using the object oriented paradigm to model context in computer go. Proceedings of the first international and interdisciplinary conference on modeling and using context.
- Boytsov, A. (2011). Context reasoning, context prediction and proactive adaptation in pervasive computing systems, Luleå tekniska universitet.
- Boytsov, A. and A. Zaslavsky (2010). Extending context spaces theory by proactive adaptation. Smart Spaces and Next Generation Wired/Wireless Networking, Springer: 1-12.
- Boytsov, A. and A. Zaslavsky (2011). Context prediction in pervasive computing systems: Achievements and challenges. Supporting Real Time Decision-Making, Springer: 35-63.
- Boytsov, A., A. Zaslavsky and K. Synnes (2009). Extending context spaces theory by predicting run-time context. Smart spaces and next generation wired/wireless networking, Springer: 8-21.
- Brézillon, P., M. Borges, J. Pino and J.-C. Pomerol (2004). Context-awareness in group work: Three case studies. Proc. of.
- Brézillon, P. and J.-C. Pomerol (1999). Contextual knowledge and proceduralized context. Proceedings of the AAAI-99 Workshop on Modeling Context in AI Applications, Orlando, Florida, USA, July. AAAI Technical Report.
- Brown, P. J. (1995). "The stick-e document: a framework for creating context-aware applications." ELECTRONIC PUBLISHING-CHICHESTER- 8: 259-272.
- Brown, P. J., J. D. Bovey and X. Chen (1997). "Context-aware applications: from the laboratory to the marketplace." IEEE personal communications 4(5): 58-64.
- Bucur, O., P. Beaune and O. Boissier (2005). Définition et représentation du contexte pour des agents sensibles au contexte. Proceedings of the 2nd French-speaking conference on Mobility and ubiquity computing, ACM.
- C. Baier, L. d. A., V. Forejt, and M. Kwiatkowska (2016). "Probabilistic Model Checking." In Handbook of Model Checking, Springer.
- Calvary, G., J. Coutaz, D. Thevenin, Q. Limbourg, N. Souchon, L. Bouillon, M. Florins and J. Vanderdonckt (2002). Plasticity of user interfaces: A revisited reference framework. In Task Models and Diagrams for User Interface Design, Citeseer.
- Chalmers, M. (2004). "A historical view of context." Computer Supported Cooperative Work (CSCW) 13(3-4): 223-247.

- Chen, G. and D. Kotz (2000). A survey of context-aware mobile computing research, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.
- Chen, H., T. Finin and A. Joshi (2005). Semantic web in the context broker architecture, DTIC Document.
- Chen, H., F. Perich, T. Finin and A. Joshi (2004). Soupa: Standard ontology for ubiquitous and pervasive applications. Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on, IEEE.
- Chikhaoui, B., S. Wang and H. Pigot (2010). A new algorithm based on sequential pattern mining for person identification in ubiquitous environments. KDD workshop on knowledge discovery from sensor data.
- Clarke, E., Grumberg, O., & Peled, D. A (1999). "model checking." MIT Press.
- Cremene, M., M. Riveill and C. Martel (2006). "Towards unanticipated dynamic service adaptation." Steffen Becker: 25.
- Da Rosa, J. H., J. L. Barbosa and G. D. Ribeiro (2016). "ORACON: An adaptive model for context prediction." Expert Systems with Applications 45: 56-70.
- Das, S. K., D. J. Cook, A. Battacharya, E. O. Heierman III and T.-Y. Lin (2002). "The role of prediction algorithms in the MavHome smart home architecture." Wireless Communications, IEEE 9(6): 77-84.
- David, K., R. Kusber, S. L. Lau, S. Sigg and B. Ziebart (2014). 3 rd workshop on recent advances in behavior prediction and pro-active pervasive computing. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, ACM.
- Davison, B. and H. Hirsh (1998). Probabilistic online action prediction. Proceedings of the AAAI Spring Symposium on Intelligent Environments.
- Davison, B. D. and H. Hirsh (1997). Toward an adaptive command line interface. HCI (2).
- De Bruijn, J., H. Lausen, A. Polleres and D. Fensel (2006). The web service modeling language WSML: an overview. European Semantic Web Conference, Springer.
- De Virgilio, R. and R. Torlone (2006). Modeling heterogeneous context information in adaptive web based applications. Proceedings of the 6th international conference on Web engineering, ACM.

- Deployment, A. L. S. S. H. (2012). Smart home in a box: A large scale smart home deployment. Workshop Proceedings of the 8th International Conference on Intelligent Environments, IOS Press.
- Dey, A. K. (2001). "Understanding and using context." *Personal and ubiquitous computing* 5(1): 4-7.
- Dey, A. K. and G. D. Abowd (2000). *CybreMinder: A context-aware system for supporting reminders*. International Symposium on Handheld and Ubiquitous Computing, Springer.
- Dietze, S., A. Gugliotta and J. Domingue (2009). Exploiting metrics for similarity-based semantic web service discovery. *Web Services, 2009. ICWS 2009. IEEE International Conference on*, IEEE.
- Dobson, S. and J. Ye (2006). Using fibrations for situation identification. *Pervasive 2006 workshop proceedings*.
- Efstratiou, C., K. Cheverst, N. Davies and A. Friday (2001). An architecture for the effective support of adaptive context-aware applications. *Mobile Data Management*, Springer.
- Fayad, M. and M. P. Cline (1996). "Aspects of software adaptability." *Communications of the ACM* 39(10): 58-59.
- Fenstad, J. E., P.-K. Halvorsen, T. Langholm and J. van Benthem (2012). *Situations, language and logic*, Springer Science & Business Media.
- Föll, S., K. Herrmann and K. Rothermel (2011). *PreCon–Expressive Context Prediction Using Stochastic Model Checking*. International Conference on Ubiquitous Intelligence and Computing, Springer.
- Franklin, M. J. (2001). *Challenges in ubiquitous data management*. Informatics, Springer.
- Fuchs, F., I. Hochstatter, M. Krause and M. Berger (2005). *A Metamodel Approach to Context Information*. PerCom Workshops.
- Gellert, A. and L. Vintan (2006). "Person movement prediction using hidden Markov models." *Studies in Informatics and control* 15(1): 17.
- Gensel, J., M. Villanova-Oliver and M. Kirsch-Pinheiro (2008). *Modèles de contexte pour l'adaptation à l'utilisateur dans des Systèmes d'Information Web collaboratifs*. Workshop from " 8èmes journées francophones". Sophia-Antipolis, France.
- Ghidini, C. and F. Giunchiglia (2001). "Local models semantics, or contextual reasoning= locality+ compatibility." *Artificial intelligence* 127(2): 221-259.

- Giunchiglia, F. and C. Ghidini (1998). "Local models semantics, or contextual reasoning= locality+ compatibility." KR 98: 282-289.
- Gopalratnam, K. and D. J. Cook (2004). "Active LeZi: An incremental parsing algorithm for sequential prediction." International Journal on Artificial Intelligence Tools 13(04): 917-929.
- Gray, P. and D. Salber (2001). Modelling and using sensed context information in the design of interactive applications. Engineering for Human-Computer Interaction, Springer: 317-335.
- Gu, T., H. K. Pung and D. Q. Zhang (2005). "A service-oriented middleware for building context-aware services." Journal of Network and computer applications 28(1): 1-18.
- Guha, R. V. (1991). Contexts: a formalization and some applications, Stanford University Stanford, CA.
- Gwizdka, J. (2000). "What's in the context. Computer Human Interaction.", CHI Workshop 11. The What, Who, Where, When, Why and How of Context-Awareness.
- Held, A., S. Buchholz and A. Schill (2002). "Modeling of context information for pervasive computing applications." Proceedings of SCI: 167-180.
- Henricksen, K. and J. Indulska (2006). "Developing context-aware pervasive computing applications: Models and approach." Pervasive and mobile computing 2(1): 37-64.
- Henricksen, K., J. Indulska and A. Rakotonirainy (2002). Modeling context information in pervasive computing systems. Pervasive Computing, Springer: 167-180.
- Henricksen, K., J. Indulska and A. Rakotonirainy (2002). Modeling context information in pervasive computing systems. International Conference on Pervasive Computing, Springer.
- Hofer, T., W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann and W. Retschitzegger (2003). Context-awareness on mobile devices-the hydrogen approach. System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on, IEEE.
- Hong, J., E.-H. Suh, J. Kim and S. Kim (2009). "Context-aware system for proactive personalized service based on context history." Expert Systems with Applications 36(4): 7448-7457.
- Horvitz, E., J. Breese, D. Heckerman, D. Hovel and K. Rommelse (1998). The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users.

Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc.

Hu, B., Z. Wang and Q. Dong (2012). A modeling and reasoning approach using description logic for context-aware pervasive computing. *Emerging Research in Artificial Intelligence and Computational Intelligence*, Springer: 155-165.

Indulska, J., R. Robinson, A. Rakotonirainy and K. Henriksen (2003). Experiences in using cc/pp in context-aware systems. *International Conference on Mobile Data Management*, Springer.

Jensen, C. S. and R. T. Snodgrass (1999). "Temporal data management." *IEEE Transactions on Knowledge and Data Engineering* 11(1): 36-44.

Kadouche, R., M. Mokhtari, S. Giroux and B. Abdulrazak (2008). Semantic approach for modelling an assistive environment using description logic. *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, ACM.

Kadouche, R., H. Pigot, B. Abdulrazaka and S. Giroux (2010). Support vector machines for inhabitant identification in smart houses. *International Conference on Ubiquitous Intelligence and Computing*, Springer.

Kammanahalli, H., S. Gopalan, V. Sridhar and K. Ramamritham (2004). Context aware retrieval in web-based collaborations. *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, IEEE.

Kaowthumrong, K., J. Lebsack and R. Han (2002). Automated selection of the active device in interactive multi-device smart spaces. *Workshop at UbiComp*, Citeseer.

Kapitsaki, G. M., G. N. Prezerakos, N. D. Tselikas and I. S. Venieris (2009). "Context-aware service engineering: A survey." *Journal of Systems and Software* 82(8): 1285-1297.

Karbassi, A. and M. Barth (2003). Vehicle route prediction and time of arrival estimation techniques for improved transportation system management. *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, IEEE.

Keeney, J. and V. Cahill (2003). Chisel: A policy-driven, context-aware, dynamic adaptation framework. *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*, IEEE.

Kirsch-Pinheiro, M. and I. Rychkova (2013). Dynamic Context Modeling for Agile Case Management. *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, Springer.

- Krause, M. and I. Hochstatter (2005). Challenges in modelling and using quality of context (qoc). International Workshop on Mobile Agents for Telecommunication Applications, Springer.
- Krumm, J. (2008). A markov model for driver turn prediction, SAE Technical Paper.
- Krupitzer, C., F. M. Roth, S. VanSyckel, G. Schiele and C. Becker (2015). "A survey on engineering approaches for self-adaptive systems." *Pervasive and Mobile Computing* 17: 184-206.
- Kwiatkowska, M., G. Norman and D. Parker (2007). Stochastic model checking. International School on Formal Methods for the Design of Computer, Communication and Software Systems, Springer.
- Kwiatkowska, M., G. Norman and J. Sproston (2003). Pctl model checking of symbolic probabilistic systems, University of Birmingham, School of Computer Science.
- Kwiatkowska, M., G. Norman, J. Sproston and F. Wang (2007). "Symbolic model checking for probabilistic timed automata." *Information and Computation* 205(7): 1027-1077.
- Lahijanian, M., S. Andersson and C. Belta (2011). Control of Markov decision processes from PCTL specifications. Proceedings of the 2011 American Control Conference, IEEE.
- Le Mouël, F. and F. Laforest (2006). "Adaptation et gestion du contexte.", Hermès, pp.166, *Ingénierie des Systèmes d'Information (ISI)*, 2-7462-1672-8.
- Lemlouma, T. and N. Layaïda (2003). Adapted content delivery for different contexts. Applications and the Internet, 2003. Proceedings. 2003 Symposium on, IEEE.
- Li, K.-J. (2007). "Ubiquitous GIS, Part I: Basic Concepts of Ubiquitous GIS." Lecture Slides, Pusan National University.
- Liu, J. S. and R. Chen (1998). "Sequential Monte Carlo methods for dynamic systems." *Journal of the American statistical association* 93(443): 1032-1044.
- Liu, Y., H.-K. Miao, H.-W. Zeng, Y. Ma and P. Liu (2013). "Nondeterministic Probabilistic Petri Net—A New Method to Study Qualitative and Quantitative Behaviors of System." *Journal of Computer Science and Technology* 28(1): 203-216.
- Madkour, M., D. El Ghanami, A. Maach and A. Hasbi (2013). "Context-aware service adaptation: an approach based on fuzzy sets and service composition." *Journal of Information Science and Engineering* 29(1): 1-16.

- Magherini, T., A. Fantechi, C. D. Nugent and E. Vicario (2013). "Using temporal logic and model checking in automated recognition of human activities for ambient-assisted living." *IEEE Transactions on Human-Machine Systems* 43(6): 509-521.
- 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 CSR* 577: 31.
- McCarthy, J. (1987). "Circumscription—a form of nonmonotonic reasoning.", Computer science department, Stanford University, Standford, CA 94305, U.S.A.
- McCarthy, J. (1993). "Notes on formalizing context.", Computer science department, Stanford University, Standford, CA 94305, U.S.A.
- McCarthy, J. and S. Buvac (1997). "Formalizing context (expanded notes).", computing natural language, Stanford University, U.S.A.
- Meiners, M., S. Zaplata and W. Lamersdorf (2010). *Structured context prediction: A generic approach. Distributed Applications and Interoperable Systems*, Springer.
- Miraoui, M. (2009). *Architecture logicielle pour l'informatique diffuse: Modélisation du contexte et adaptation dynamique des services*, École de technologie supérieure.
- Miraoui, M. (2014). "Dynamic Adaptation in Ubiquitous Services: A Conceptual Architecture." *Handbook of Research on Architectural Trends in Service-Driven Computing*: 160.
- Mostefaoui, G. K., J. Pasquier-Rocha and P. Brezillon (2004). *Context-aware computing: a guide for the pervasive computing community. Pervasive Services, 2004. ICPS 2004. IEEE/ACS International Conference on*, IEEE.
- Mozer, M. C. (1998). *The neural network house: An environment hat adapts to its inhabitants. Proc. AAAI Spring Symp. Intelligent Environments*.
- Najar, S., M. K. Pinheiro and C. Souveyet (2014). "A new approach for service discovery and prediction on Pervasive Information System." *procedia computer science* 32: 421-428.
- Najar, S., O. Saidani, M. Kirsch-Pinheiro, C. Souveyet and S. Nurcan (2009). *Semantic representation of context models: a framework for analyzing and understanding. Proceedings of the 1st Workshop on Context, Information and Ontologies*, ACM.
- Nalepa, G. J. and S. Bobek (2014). "Rule-based solution for context-aware reasoning on mobile devices." *Computer Science and Information Systems* 11(1): 171-193.

- Narayanan, D., J. Flinn and M. Satyanarayanan (2000). Using history to improve mobile application adaptation. *Mobile Computing Systems and Applications*, 2000 Third IEEE Workshop on., IEEE.
- Niforatos, E., E. Karapanos and S. Sioutas (2012). "PLBSD: a platform for proactive location-based service discovery." *Journal of Location based services* 6(4): 234-249.
- Nurmi, P., M. Martin and J. A. Flanagan (2005). Enabling proactiveness through context prediction. *Proceedings of the Workshop on Context Awareness for Proactive Systems*, Helsinki, Citeseer.
- Öztürk, P. and A. Aamodt (1997). Towards a model of context for case-based diagnostic problem solving. *Context-97; Proceedings of the interdisciplinary conference on modeling and using context*. Rio de Janeiro, Citeseer.
- Padovitz, A., S. W. Loke and A. Zaslavsky (2004). Towards a theory of context spaces. *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, IEEE.
- Pascoe, J. (1998). Adding generic contextual capabilities to wearable computers. *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, IEEE.
- Pascoe, J., D. Morse and N. Ryan (1997). Enhanced reality fieldwork: the context-aware archaeological assistant. *Computer Applications in Archaeology*.
- Pejovic, V. and M. Musolesi (2015). "Anticipatory mobile computing: A survey of the state of the art and research challenges." *ACM Computing Surveys (CSUR)* 47(3): 47.
- Perera, C., A. Zaslavsky, P. Christen and D. Georgakopoulos (2014). "Context aware computing for the internet of things: A survey." *Communications Surveys & Tutorials*, IEEE 16(1): 414-454.
- Petit, M. (2010). "A spatial approach to execution-context modeling in ubiquitous information's systems." Unpublished PhD report, Arts et Metiers ParisTech.
- Petzold, J., F. Bagci, W. Trumler and T. Ungerer (2003). Context prediction based on branch prediction methods, Citeseer.
- Petzold, J., A. Pietzowski, F. Bagci, W. Trumler and T. Ungerer (2005). Prediction of indoor movements using bayesian networks. *Location-and Context-Awareness*, Springer: 211-222.
- Preuveneers, D. and Y. Berbers (2008). "Encoding semantic awareness in resource-constrained devices." *IEEE Intelligent Systems* 23(2): 26-33.

- Preuveneers, D., J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers and K. De Bosschere (2004). Towards an extensible context ontology for ambient intelligence. European Symposium on Ambient Intelligence, Springer.
- Preuveneers, D., K. Victor, Y. Vanrompay, P. Rigole, M. K. Pinheiro and Y. Berbers (2009). "Context-aware adaptation in an ecology of applications." Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability: Adaptive Technologies and Applications: 1-25.
- Ranganathan, A. and R. H. Campbell (2003). "An infrastructure for context-awareness based on first order logic." Personal and Ubiquitous Computing 7(6): 353-364.
- Ranganathan, A., R. H. Campbell, A. Ravi and A. Mahajan (2002). "Conchat: A context-aware chat program." IEEE Pervasive computing 1(3): 51-57.
- Ranganathan, A., R. H. Campbell, A. Ravi and A. Mahajan (2002). "Conchat: A context-aware chat program." Pervasive Computing, IEEE 1(3): 51-57.
- Ratté, S. and S. Ratté (1994). "Computational Event Structures Part I.", BCS Memo, MIT Brain and cognitive sciences department.
- Razzaque, M. A., S. Dobson and P. Nixon (2006). "Categorization and modelling of quality in context information."
- Reiter, R. (2001). Knowledge in action: logical foundations for specifying and implementing dynamical systems, MIT press.
- Salfner, F., M. Lenk and M. Malek (2010). "A survey of online failure prediction methods." ACM Computing Surveys (CSUR) 42(3): 10.
- Satyanarayanan, M. (2001). "Pervasive computing: Vision and challenges." IEEE Personal communications 8(4): 10-17.
- Schilit, B., N. Adams and R. Want (1994). Context-aware computing applications. Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on, IEEE.
- Schilit, B. N. and M. M. Theimer (1994). "Disseminating active map information to mobile hosts." IEEE network 8(5): 22-32.
- Schmidt, A., K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven and W. Van de Velde (1999). Advanced interaction in context. International Symposium on Handheld and Ubiquitous Computing, Springer.

- Serral, E., P. Valderas and V. Pelechano (2010). Supporting runtime system evolution to adapt to user behaviour. International Conference on Advanced Information Systems Engineering, Springer.
- Sigg, S., S. Haseloff and K. David (2006). A novel approach to context prediction in ubicomp environments. Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on, IEEE.
- Sigg, S., S. Haseloff and K. David (2010). "An alignment approach for context prediction tasks in ubicomp environments." IEEE Pervasive Computing 9(4): 90-97.
- Simmons, R., B. Browning, Y. Zhang and V. Sadekar (2006). Learning to predict driver route and destination intent. Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE, IEEE.
- South, G., A. Lenaghan and R. Malyan (2000). Using reflection for service adaptation in mobile clients. XVII World Telecommunications Congress, Birmingham, UK.
- Strang, T. and C. Linnhoff-Popien (2004). A context modeling survey. Workshop Proceedings.
- Truong, B. A., Y.-K. Lee and S.-Y. Lee (2005). Modeling and reasoning about uncertainty in context-aware systems. IEEE International Conference on e-Business Engineering (ICEBE'05), IEEE.
- VanSyckel, S. and C. Becker (2014). A survey of proactive pervasive computing. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, ACM.
- Vendler, Z. (2005). "Verbs and Times1." The Language of Time: a reader 66: 21.
- Viana, W., A. D. Miron, B. Moiscuc, J. Gensel, M. Villanova-Oliver and H. Martin (2011). "Towards the semantic and context-aware management of mobile multimedia." Multimedia Tools and Applications 53(2): 391-429.
- Vintan, L., A. Gellert, J. Petzold and T. Ungerer (2004). Person movement prediction using neural networks. First Workshop on Modeling and Retrieval of Context.
- Wagner, A., J. L. V. Barbosa and D. N. F. Barbosa (2014). "A model for profile management applied to ubiquitous learning environments." Expert Systems with Applications 41(4): 2023-2034.
- Wang, X. H., D. Q. Zhang, T. Gu and H. K. Pung (2004). Ontology based context modeling and reasoning using OWL. Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, Ieee.

- Want, R., A. Hopper, V. Falcao and J. Gibbons (1992). "The active badge location system." *ACM Transactions on Information Systems (TOIS)* 10(1): 91-102.
- Ward, A., A. Jones and A. Hopper (1997). "A new location technique for the active office." *IEEE Personal Communications* 4(5): 42-47.
- Winograd, T. (2001). "Architectures for context." *Human-Computer Interaction* 16(2): 401-419.
- Xiao, H., Y. Zou, J. Ng and L. Nigul (2010). An approach for context-aware service discovery and recommendation. *Web Services (ICWS), 2010 IEEE International Conference on*, IEEE.
- Yarvis, M., P. Reiher and G. J. Popek (1999). Conductor: A framework for distributed adaptation. *Hot Topics in Operating Systems, 1999. Proceedings of the Seventh Workshop on*, IEEE.
- Yuan, J., J. Li and B. Zhang (2007). Exploiting spatial context constraints for automatic image region annotation. *Proceedings of the 15th ACM international conference on Multimedia*, ACM.
- Zaidenberg, S., B. Boulay and F. Brémond (2012). A generic framework for video understanding applied to group behavior recognition. *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, IEEE.
- Ziebart, B. D., A. L. Maas, A. K. Dey and J. A. Bagnell (2008). Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. *Proceedings of the 10th international conference on Ubiquitous computing*, ACM.
- Ziv, J. and A. Lempel (1978). "Compression of individual sequences via variable-rate coding." *Information Theory, IEEE Transactions on* 24(5): 530-536.
- Zukerman, I., D. W. Albrecht and A. E. Nicholson (1999). *Predicting users' requests on the WWW*, Springer.