



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Larbi Tébessi–Tébessa-
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département :Mathématiques et Informatique



MEMOIRE DE MASTER

Domaine : Informatique

Filière : Informatique

Option : Systèmes d'Information

Thème :

*Adaptation des profils utilisateurs
selon le contexte*

Présenté par :

Oussama SOLTANI

Mohamed Abdelouaheb BOUAZDIA

Devant le jury :

A.A. Betouil	M.A.B	Université de Larbi Tébessi	Président
Louardi BRAIDJI	M.C.B	Université de Larbi Tébessi	Rapporteur
A. Zeggari	M.A.B	Université de Larbi Tébessi	Examineur

Date de soutenance :25-05-2017

Note :

Mention :

ملخص

العديد من التطبيقات " الحوسبة في كل مكان " تحتاج إلى تدفق مستمر من المعلومات لتكون حول بيئتهم لتكون قادرة على التكيف مع سياقها المتغير، وهذه التطبيقات تستخدم المعلومات السياقية للتكيف مع المعلومات الديناميكية لبيئات التشغيل و إحتياجات المستخدم يتم إكتشافها بشكل متزايد داخل كل مكان رؤية الحوسبة. و تستمر هذه الحالة إلى حد كبير بسبب عدم وجود خلاصات رفيعة المستوى مناسبة لوصف المعلومات السياقية و إستخداماتها كأساس للتكيف مع بيانات المستخدمين . في هذا العمل نذكر الأعمال ذات الصلة في هذا المجال ، و نحن نحاول إقتراح نموذج لتكيف الشخصية في الأنترنت مع المعلومات السياقية .

Abstract

Many “ubiquitous computing” applications need a constant flow of information about their environment to be able to adapt to their changing context, and this Applications that using the contextual information to adapt their dynamic parameters to operational environments and user needs are increasingly explored within the ubiquitous computing vision. This situation persists largely, because of lack of appropriate high-level abstractions to describe, reason and uses of contextual information as a basis for user profile adaptation. In this work we mention the related work in this field and we try to propose a model of adaptation of the online profile on the context.

Keyword :

Ubiquitous, computing, context, user profile

Résumé

De nombreuses applications « informatique ubiquitaire » ont besoin d'une Flux constant d'informations sur leur environnement, Pouvoir s'adapter à leur contexte changeant, et Les applications qui exploitent l'information contextuelle afin d'adapter leurs comportements à des environnements opérationnels dynamiques et aux besoins des utilisateurs sont de plus en plus explorées dans le cadre de la vision de l'informatique ubiquitaire. Cette situation persiste en grande partie en raison de l'absence d'abstractions appropriées de haut niveau pour décrire, raisonner et exploiter l'information contextuelle comme base d'adaptation de profile utilisateur. Dans ce travail, nous mentionnons les travaux connexes dans ce domaine et nous essayons de proposer un modèle d'adaptation du profil utilisateur basé sur le contexte.

Mot clé :

Informatique ubiquitaire, Adaptation, profile utilisateur, contexte.

Dédicace

Avec toute notre estime, nous dédions cet humble travail..

à nos familles, particulièrement nos parents

à nos enseignants

à Imane, Marwa, F.Tahar, T.Manel et spécialement Farouk pour tous les efforts et les sacrifices que vous n'avez jamais cessé de consentir pour notre instruction et notre bien-être

Nous tenons à travers cette modeste dédicace à vous exprimer toute notre affection et énorme respect.

Remerciement

Nous remercions toutes les personnes qui ont supporté nos humeurs dépressives au cours des années du cursus universitaire et grâce à qui nous avons pu aller jusqu'au bout de nos études.

Nous tenons de même à remercier nos parents qui nous ont soutenu dans notre vie.

Nos gratitudes vont à nos enseignants pour tout ce qu'ils nous ont fait acquérir, et surtout, à notre encadreur, Mr Louardi BRAIDJI pour avoir accepté de nous suivre dans ce projet

Merci 

Table des matières

LISTE DES TABLEAUX

LISTE DES FIGURES

ABREVIATIONS

INTRODUCTION GENERALE..... 1

Chapitre I : Informatique Ubiquitaire

INTRODUCTION	4
1.L'INFORMATIQUE UBIQUITAIRE	4
1.1.Définition.....	4
1.2.Caractéristiques de l'informatique ubiquitaire.....	6
1.3.Applications de l'informatique ubiquitaire.....	6
2.CONTEXTE	7
2.1.Définitions générales.....	7
2.2.Catégorisation du contexte.....	9
2.3.L'utilisation du contexte.....	10
2.4.Sensibilité au contexte.....	12
2.4.1. Définition.....	12
2.4.2. Système sensible au contexte.....	12
2.4.3. Architecture générale d'un système sensible au contexte.....	13
3.ADAPATATION	18
3.1.Définition.....	19
3.2.Les aspects de l'adaptation.....	20
3.2.1. Pourquoi (Why).....	20
3.2.2. Quoi (What).....	21
3.2.3. Qui (Who).....	21
3.2.4. Quand (When).....	22
3.2.5. Où (Where).....	22
3.2.6. A quoi (To what).....	23
3.3. Raisons d'adaptation.....	23
3.3.1. Adaptation corrective.....	23
3.3.2. Adaptation adaptative.....	23
3.3.3. Adaptation évolutive.....	23
3.3.4. Adaptation perfective.....	24
3.4. Les types d'adaptation.....	24
3.4.1. L'adaptation statique vs dynamique.....	24
3.4.2. L'adaptation externe vs interne.....	24
3.4.3. L'adaptation architecturale vs comportementale.....	24
3.4.4. L'adaptation basée sue les modèles vs sans modèles.....	25
3.4.5. L'adaptation réactive vs proactive.....	25
3.4.6. L'adaptation du contenu vs de la présentation.....	25
CONCLUSION	25

Chapitre II : Les profils utilisateurs

INTRODUCTION	27
1.DEFINITION DE PROFIL	27
2.DEFINITION DU PROFIL UTILISATEUR	27
3.UTILISATION DU PROFIL UTILISATEUR	28
4.REPRESENTATION DES PROFILES UTILISATEURS	29
4.1.Représentation ensembliste.....	29
4.2. Représentation par réseaux sémantiques.....	90
4.3.Représentation conceptuelle.....	32
5.CONSTRUCTION D’UN PROFIL UTILISATEUR	33
6. MODELISATION DU PROFIL UTILISATEUR	34
7. EXPLOITATION DU PROFIL UTILISATEUR	35
8. LES APPROCHES DE CONSTRUCTION D’UN PROFIL UTILISATEUR	36
CONCLUSION	38

Chapitre III : Contribution

INTRODUCTION	39
1.MODELE D’UN SYSTEME SENSIBLE AU CONTEXTE	39
2. MODELE DE L’ENVIRONNEMENT	40
3.MODELE D’ADAPTATEUR AU CONTEXTE	42
4.MANIPULATION DES ACTIONS	43
5.METHODOLOGIE D’ADAPTATION AU CONTEXTE	44
6.GENERATEUR DE PROFIL EN CONTEXTE AWARE	45
6.1. generateur du profil en contexte aware.....	45
7.TRAITEMENT DE L’INFORMATION CONTEXTUEL	48
7.1. Acquisition et stockage des informations du contexte.....	48
7.2. Control du degre d’abstraction du contexte.....	48
7.3. Utilisation de l’information du contexte pour des services ou des applications.....	49
8.NOTRE PROPOSITION	49
9.CAS D’ETUDE	50
10.L’IMPLEMENTATION	52
10.1.Les services web.....	52
10.1.1 architecture du service web.....	53
10.1.2 Langages et protocoles.....	53
10.2. Visual studio.....	54
10.3. Système d’adaptation.....	56
10.3.1.L’affichage de profile utilisateur.....	57
10.3.2. acquisition et stockage des informations du contexte.....	58
10.3.3. Control du degre d’abstraction du contexte.....	59
10.3.4. utilisation de l’information du contexte pour l’adaptation de profile.....	59
CONCLUSION	61
CONCLUSION GENERALE	62
PERSPECTIVE	63
BIBLIOGRAPHIE	64

Liste de figures

N°	Titre	Page
I.1	Architecture de systèmes sensibles au contexte	13
II.1	Un exemple de profil représenté par des mots clés	29
II.2	Exemple de profil sémantique de l'utilisateur	32
II.3	Exemple 1 profil vectoriel	37
II.4	Exemple 2 graphe de dépendance de termes	37
II.5	Exemple 4 d'une BD	38
III.1	Les composants d'un système sensible au contexte	39
III.2	Modèle de l'environnement de travail	41
III.3	Modèle de l'adaptateur au contexte	43
III.4	Architecture proposée par Ketaki	46
III.5	Modèle proposée	48
III.6	Exemple d'un profile utilisateur	49
III.7	Collection des informations contextuelles	49
III.8	Traitement de l'information contextuel	50
III.9	Adaptation de le profile.	50
III.10	Profile adapté	51
III.11	Architecture global de Service Web	53
III.12	Visual Studio, Microsoft .	54
III.13	Une autre table « affichage » de base de données contient les statuts d'affichage pour les champs de la table « profile »	55
III.13	Les champs de tableaux	56
III.14	Méthode afficher_profile()	56
III.15	Résultat de la méthode afficher_profile()	57
III.16	La méthode « context_info() »	57
III.17	La méthode « context_info() » sur l'interface publique de service web	58
III.18	Mis à jour de statut	59
III.19	Profile utilisateur après le mis à jour	59
III.20	Les champs de profile	59

Liste de tableaux

N°	Titre	Page
I.1	Exemples de paramètres de contexte.	10
II.1	Exemple 3 de profil	38

Introduction

Générale

Introduction Générale

Contexte

L'informatique ubiquitaire permet à l'utilisateur de gérer l'information depuis un terminal mobile. Un système mobile est en fait un système distribué doté de capacités de gestion de la mobilité. Pour assumer toutes ces fonctionnalités intelligentes de traitement autonome des informations ou de communication, les dispositifs mobiles doivent être dotés d'outils logiciels permettant d'exploiter les informations géo spatiales dans le cadre d'activités ayant trait aux déplacements.

Parallèlement aux systèmes informatiques, les SI doivent s'adapter et supporter cette nouvelle vision. L'objectif d'un SI est la production, la collection, le traitement, l'enregistrement et la diffusion de l'information.

Pour permettre l'amélioration de la qualité des services fournis par les SI, l'usage des technologies ubiquitaires permet la prise en compte du contexte d'exécution pour que les applications soient sensibles au contexte. La prise en compte du contexte est la perception de l'environnement pour interagir plus naturellement avec le profile utilisateur, et adapter cette profile selon l'information contextuelle fournit par des capteurs de l'environnement physique, des matériels autodescriptifs, la description des personnes ou encore les métadonnées sur les applications.

Problématique

Le contexte est n'importe quelle information qui peut être utilisée pour caractériser la situation d'une entité. Une entité est une personne, un endroit ou un objet qui est considérée pertinente dans l'interaction entre un utilisateur et une application, incluant l'utilisateur et les applications elles-mêmes. Les éléments du contexte qui caractérisent l'utilisation d'un système, pouvant être détectés et utilisés par le système pour offrir un résultat adapté telle que le profile utilisateur.

Mais le problème majeur est le suivant :

- Comment gérer et utiliser ces informations ?
- Comment s'assurer de la qualité de ces informations ?
- Comment construire un profil utilisateur de haute qualité basé sur ces informations contextuelles ?

Objectifs

L'objectif de notre travail consiste à proposer un modèle d'adaptation le profile utilisateur basé sur les l'information contextuelle dans l'environnement ubiquitaire.

Organisation du mémoire

Le document est organisé comme suit :

Chapitre 01 : décrit quelques concepts de l'informatique ubiquitaires.

Chapitre 02 : Vue générale sur le profile utilisateur.

Chapitre 03 : Étudier les travaux connexes et le modèle proposé de notre travail. Et enfin une conclusion générale qui résume notre travail.

Chapitre I :

Informatique Ubiquitaire

Chapitre I :

INTRODUCTION

1.L'INFORMATIQUE UBIQUITAIRE

1.1.DEFINITION

1.2.CARACTERISTIQUES DE L'INFORMATIQUE UBIQUITAIRE

1.3.APPLICATIONS DE L'INFORMATIQUE UBIQUITAIRE

2.CONTEXTE

2.1.DEFINITIONS GENERALES

2.2.CATEGORISATION DU CONTEXTE

2.3.L'UTILISATION DU CONTEXTE

2.4.SENSIBILITE AU CONTEXTE

2.4.1. DEFINITION

2.4.2. SYSTEME SENSIBLE AU CONTEXTE

2.4.3. ARCHITECTURE GENERALE D'UN SYSTEME SENSIBLE AU CONTEXTE

3.ADAPATATION

3.1.DEFINITION

3.2.LES ASPECTS DE L'ADAPTATION

3.2.1. POURQUOI (WHY)

3.2.2. QUOI (WHAT)

3.2.3. QUI (WHO)

3.2.4. QUAND (WHEN)

3.2.5. OU (WHERE)

3.2.6. A QUOI (TO WHAT)

3.3. RAISONS D'ADAPTATION

3.3.1. ADAPTATION CORRECTIVE

3.3.2. ADAPTATION ADAPTATIVE

3.3.3. ADAPTATION EVOLUTIVE

3.3.4. ADAPTATION PERFECTIVE

3.4. LES TYPES D'ADAPTATION

3.4.1. L'ADAPTATION STATIQUE VS DYNAMIQUE

3.4.2. L'ADAPTATION EXTERNE VS INTERNE

3.4.3. L'ADAPTATION ARCHITECTURALE VS COMPORTEMENTALE

3.4.4. L'ADAPTATION BASEE SUE LES MODELES VS SANS MODELES

3.4.5. L'ADAPTATION REACTIVE VS PROACTIVE

3.4.6. L'ADAPTATION DU CONTENU VS DE LA PRESENTATION

CONCLUSION

INTRODUCTION

Dans ce chapitre, nous introduisons les domaines de recherche dans lesquels s'inscrit notre travail. En premier lieu, nous présentons L'informatique ubiquitaire qui est un paradigme récent dont l'objectif est de permettre aux utilisateurs de consulter des données n'importe quand, n'importe où, notamment à l'aide de dispositifs mobiles (DM) distincts. Ainsi les caractéristiques et quelques exemples d'applications dans ce domaine.

Dans la suite de ce chapitre, on détaille Les notions de contexte et adaptation au moyen de définitions et de classifications, dont nous citons les plus importantes dans cette section. Nous nous intéressons également à la sensibilité au contexte, pour finalement présenter quelques solutions pour sa prise en compte dans les applications.

1. L'INFORMATIQUE UBIQUITAIRE

1.1.DÉFINITION

Le concept d'informatique ubiquitaire a été introduit par Marc Weiser [1] pour désigner sa vision futuriste de l'informatique du XXI^{ème} siècle. Il imaginait un monde abondamment peuplé d'objets informatiques et numériques qui seraient reliés en réseaux à très grande échelle et interagiraient de manière autonome et transparente afin d'accomplir diverses tâches de la vie quotidienne. L'idée novatrice de cette vision réside dans le fait de mettre les nouvelles technologies (e.g., les technologies de l'information et de la communication au service des utilisateurs, et non l'inverse. En particulier, l'objectif de l'informatique ubiquitaire est de permettre aux utilisateurs d'accéder aux différents services offerts par ces objets communicants, le plus naturellement possible, n'importe où, à tout instant et à partir de divers dispositifs [2].

Plus précisément, dans [3], M. Weiser place l'informatique ubiquitaire comme la dernière étape de l'évolution de l'informatique [2]:

- à l'époque du mainframe¹, plusieurs personnes devaient partager un même ordinateur.

Mainframe : Également connu sous le surnom de « BigIron », qui insiste sur la quantité de matériel - ou sous le nom d'ordinateur central - un système dit « mainframe » correspond à un ordinateur de hautes performances requérant une disponibilité et une sécurité supérieures à ce que peut offrir une machine de moindre envergure.

- à l'époque de l'ordinateur individuel, chacun pouvait disposer de son ou ses ordinateurs qu'il possédait personnellement.
- à l'époque de l'informatique ubiquitaire, nous aurons tous un large nombre d'ordinateurs que nous porterons sur nous ou qui seront intégrés à notre environnement quotidien[2].

Selon Weiser, les ordinateurs doivent être invisibles en les intégrant dans notre environnement. Il déclarait : "Les technologies les plus profondes sont celles qui disparaissent. Elles se fondent dans la vie de tous les jours jusqu'à ce qu'on ne puisse plus les percevoir". Cette vision permet aux utilisateurs de se concentrer surtout sur leurs tâches et leurs objectifs plutôt que sur la configuration de leurs machines.

En 2001, Satyanarayanan[4] a fait une étude sur les thématiques de recherche qui contribuent à l'émergence de l'informatique ubiquitaire. Il explique que l'évolution des systèmes ubiquitaires se base principalement sur les systèmes distribués et les systèmes mobiles. Dans son article *Pervasive Computing : Vision and Challenges* [4], il pose des questions liées à la sécurité telles que : "Comment on peut exprimer des contraintes de sécurité dans un environnement ubiquitaire ?" et "Quels sont les techniques d'authentification les plus adaptées à un environnement ubiquitaire ?". Dans le même article, Satyanarayanan identifie des axes de recherche liés à l'informatique ubiquitaire comme les espaces intelligents, l'invisibilité et le passage à l'échelle localisé [5].

Ce nouveau paradigme de l'informatique ubiquitaire a atteint une place excellente dans les dernières années ; il représente un nouveau type d'interaction entre les utilisateurs et leurs dispositifs. De nos jours, les technologies de communication sans fil telles que les réseaux locaux sans fil et Bluetooth assurent une meilleure interaction entre les dispositifs sans même avoir besoin d'une infrastructure centrale (les réseaux ad-hoc²). En plus, les utilisateurs disposent de différents types d'unités mobiles telles que les ordinateurs portables, les assistants numériques personnels (PDA) ou les téléphones intelligents (Smartphone). Le changement de type d'interaction ne se limite pas uniquement aux changements d'interface homme-machine ; l'utilisateur peut interagir avec plusieurs dispositifs simultanément, ce qui oblige les nouveaux systèmes informatiques à répondre à ces nouveaux besoins[5].

Les réseaux ad hoc : sont des réseaux sans fil capables de s'organiser sans infrastructure définie préalablement. ² Par exemple d'un équipement à un autre sans infrastructure (point d'accès).

1.2. CARACTERISTIQUE DE L'INFORMATIQUE UBIQUITAIRE

Selon [6] L'informatique ubiquitaire est caractérisé par :

- L'utilisation de dispositifs légers et sans fil ; Rahwan et al. [7] soulignent certaines de leurs caractéristiques et contraintes techniques telles que :

- Le stockage limité ;
- La puissance de traitement limitée (sur un DM, ne sont possibles que l'utilisation de petites bibliothèques et l'exécution de petites applications) ;
- Le besoin d'adjoindre des composants matériels embarqués (pour les dispositifs d'accès, par exemple, les capteurs de lumière, de son, de localisation, des lecteurs de code barres) pour la prise en compte des caractéristiques contextuelles de l'utilisateur (l'information sur les changements des caractéristiques contextuelles doit être fournie par l'utilisateur ou doit être détectée d'une manière automatique, par exemple, la localisation peut être détectée à travers un capteur GPS³).

- L'environnement ubiquitaire est caractérisé par la haute distribution, hétérogénéité, mobilité.

- L'informatique ubiquitaire a introduits plusieurs changements, mentionnent Koch et al. [8] :

- Un changement d'infrastructure qui nécessite la construction de technologies Robustes tant matérielles que logicielles facilitant la connectivité mobile, l'identification de la localisation, la découverte de services, etc.
- Le changement de services qui concerne la manière dont l'infrastructure disponible est utilisé afin de fournir de nouveaux services à l'utilisateur.

1.3. APPLICATIONS DE L'INFORMATIQUE UBIQUITAIRE

Selon [6] Parmi les applications de l'informatique ubiquitaire on peut citer les suivantes:

- MyCampus [7] est une infrastructure basée sur le Web Sémantique, sensible au contexte, en particulier aux préférences de confidentialité de l'utilisateur (« privacy preferences »). Mycampus est utilisée afin d'afficher les informations journalières

GPS (Global Positioning System) : Système de géo localisation par satellite, développé par l'armée américaine, ³ est mis à disposition des civils. Il permet de déterminer les coordonnées géographiques de n'importe quel point situé à la surface du globe. Sa précision peut atteindre 1 mètre. Le GPS s'utilise en association avec une carte pour se repérer et se positionner : randonnées, voile, trek...

d'un campus universitaire (par exemple, les activités pour des étudiants, le calendrier d'activités, l'emploi du temps, la localisation à l'intérieur du campus, etc.).

- Des applications telles que WAY (« Where Are You »)[10], Ad-Me («Advertising for mobile e-commerce user »)[11] et Gulliver's Genie[12] ont été développées afin d'adapter l'information à la localisation de l'utilisateur, compte tenu des caractéristiques de son DM (en général, des PDA et des téléphones portables).

- WAY[10] est un système pour la synchronisation et la prise de rendez vous entre utilisateurs nomades à l'aide de leurs PDA.

- Ad-Me[11] est un système qui fournit aux utilisateurs nomades des services de publicité (des magasins, des restaurants) à partir de leur localisation et de leurs préférences.

- Gulliver's Genie[12] est un guide touristique qui se configure selon la localisation et l'orientation de déplacement de l'utilisateur et personnalise l'information tenant compte de son profil (par exemple, Gulliver's Genie fait un plan de la ville en soulignant tous les musées si les sites touristiques les plus visités par l'utilisateur sont des musées).

Les applications de L'informatique ubiquitaire doivent satisfaire l'utilisateur en lui donnant une réponse pertinente de point de vue contenu et temps. Pour cela, elles doivent prendre en compte la situation de cet utilisateur dite : situation contextuelle. Ensuite, adapter tout le comportement à la situation en question. Dans ce domaine, les applications sont sensibles au contexte « context-aware applications ». La section suivante est consacrée à la notion de contexte.

2. CONTEXTE

Le contexte est considéré, généralement, comme un ensemble d'informations sur un environnement ; et comme il est déjà mentionné dans les sections précédentes, le concept de contexte se met en pole position pour déclencher le processus d'adaptation au profile de l'utilisateur. Il est aussi intéressant d'essayer d'aborder la sensibilité au contexte, pour finalement présenter quelques solutions pour sa prise en compte dans les applications.

2.1. DEFINITIONS GENERALES

Le contexte est, à la base, une notion complexe et abstraite, il est donc beaucoup plus facile de trouver une définition générale qu'en trouver une détaillée ; voici, donc, une liste des approches principales[13].

Parmi les premiers à essayer de définir le contexte, se trouvent Schilit et Theimer, pour eux, le contexte est constitué de la localisation de l'utilisateur, ainsi que des identités et des états des personnes et des objets qui l'entourent [14]. Brown et al.[15] amplifient cette définition par des données telles que l'identité de l'utilisateur, son orientation ou la température. Ryan et al.[16] ont, eux, ajouté la notion du temps.

Pascoe[17] introduit un élément important : l'intérêt. En effet, il définit le contexte comme un ensemble d'états physiques et conceptuels qui ont un certain intérêt pour une entité donnée. Cette notion d'intérêt ou de pertinence est reprise par Abowd, Dey et al.[18] dans leur définition, qui est communément acceptée:

«Le contexte couvre toutes les informations qui peuvent être utilisées pour caractériser la situation d'une entité. Une entité est une personne, un endroit ou un objet que l'on considère pertinent par rapport à l'interaction entre un utilisateur et une application, y compris l'utilisateur et l'application eux-mêmes. »

Chen et Kotz[19] ont eux proposé la définition suivante :

«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 ».

C'est à partir de cette dernière définition que Winograd[20] s'est appuyé pour apporter ces retouches en donnant plus de détails, car il considère que, malgré le fait qu'elle couvre tous les travaux existants, il s'avère qu'elle est trop générale et qu'elle nécessite une organisation pour optimiser le développement car, à la base, il a la réflexion que tout élément peut être considéré comme faisant partie du contexte. En premier lieu, il précise que le contexte est un ensemble d'informations. Cet ensemble est plus ou moins structuré et partagé, doté de capacités d'évoluer dans le temps. En deuxième lieu, selon lui, l'appartenance d'une information au contexte ne dépend guère de ses propriétés statiques et basiques, mais de la manière dont elle est utilisée. Une information fait partie du contexte seulement si le système dépend d'elle, d'une manière ou d'une autre.

Dey [21] à son tour proposé :

« 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 eux-Mêmes ».

Henricksen, Indulska et Rakotonirainy pensent que : « la circonstance ou la situation dans laquelle une tâche informatique se déroule »

2.2.CATEGORISATION DU CONTEXTE

Dans les travaux existants, nous avons trouvé plusieurs taxonomies du contexte, qui classifient les différents types de données contextuelles. Ces classifications permettent de lister les paramètres que l'on peut prendre en compte dans un système sensible au contexte.

La première catégorisation est celle de Schilit[22], qui propose de répondre aux questions où l'on se trouve, avec qui et quelles sont les ressources à proximité afin de caractériser le contexte d'une entité. Ryan et al.[16] proposent, eux, les catégories de position, identité, environnement et temps. La classification la plus acceptable est celle proposée par de Abowd, Dey et al.[18], qui est constituée de deux niveaux superposés, avec des paramètres primaires et secondaires. Les paramètres primaires sont les mêmes que ceux de Ryan et al., sauf que la notion d'activité (c'est-à-dire, ce qu'on est en train de faire) remplace celle de l'environnement, trop générale. Les paramètres secondaires sont des attributs des entités qui se trouvent dans le contexte primaire. Ces paramètres peuvent être retrouvés en utilisant les données primaires pour les indexer. Par exemple, le numéro de téléphone d'une personne peut être retrouvé en utilisant l'identité de la personne comme index dans une base de données qui contient des numéros de téléphone.

Chen et Kotz[19] ont traité une taxonomie alternative qui divise le contexte en trois catégories principales : le contexte informatique, le contexte utilisateur et le contexte physique.

- Le contexte informatique contient toutes les informations relatives aux ressources matérielles qui permettent l'exécution de l'application.

- Le contexte utilisateur, lui, contient des informations relatives à l'identité de l'utilisateur, ses préférences, ses relations sociales, etc.
- Finalement, les données du contexte physique caractérisent l'environnement physique dans lequel se trouvent les personnes et les applications. Cette dernière catégorie regroupe également les données temporelles.

Le Tableau I.1 présente quelques exemples de paramètres de contexte de ces trois catégories.

Tableau I.1 : Exemples de paramètres de contexte.

Type d'information	Catégorie	Exemples
Identité	utilisateur	nom, prénom
Spatiale	physique	position, orientation, vitesse
Temporelle	physique	date, heure, saison
Environnementale	physique	température, luminosité, bruit
Sociale	utilisateur	personnes aux alentours, activités, calendrier
Ressources d'une machine	informatique	CPU, RAM, batterie
Ressources de communication	informatique	bande passante, nombre de connexions TCP
Physiologique	utilisateur	pression artérielle, pouls

2.3. L'UTILISATION DU CONTEXTE

Le contexte est pris en compte dans différents domaines informatiques incluant le traitement du langage naturel, l'apprentissage automatique⁴, l'extraction de l'information et même la sécurité informatique. Naturellement, le but de la prise en compte du contexte est de renforcer l'adaptabilité et le support à la décision du système. Ces derniers aspects d'infrastructure informatique requièrent l'adaptabilité des services fournis et des médias des utilisateurs en fonction du lieu, de l'activité, c'est à dire dans le cadre du contexte. On couvre, ainsi, des systèmes informatiques diffus en disant qu'ils sont sensibles au contexte (en anglais contextaware) parce qu'ils sont capables d'utiliser le contexte d'une entité afin de modifier

⁴**L'apprentissage automatique** : ou apprentissage statistique (*machine learning* en anglais), champ d'étude de l'intelligence artificielle, concerne la conception, l'analyse, le développement et l'implémentation de méthodes permettant à une machine d'évoluer par un processus systématique, et ainsi de remplir des tâches difficiles ou problématiques à remplir par des moyens algorithmiques plus classiques.

leurs fonctionnements pour offrir des meilleurs services aux utilisateurs. Chalmers[23] communique cinq principales utilisations du contexte dans les systèmes informatiques comme suit :

- *Senseur du contexte* : le contexte y est capté et les informations décrivant le contexte courant (température, localisation, ...) sont présentées à l'utilisateur
- *Associer le contexte aux données appelées augmentation contextuelle*. Par exemple : les enregistrements sur les objets inspectés peuvent être associés à leur localisation ; les notes d'une réunion peuvent être associées aux personnes assistant à la réunion et le lieu où elle s'est déroulée
- *Permettre la découverte de ressources contextuelles*, par exemple, faire en sorte que l'impression d'un document ait lieu sur l'imprimante la plus proche
- Dans le cas des évènements déclenchés par le contexte pour déclencher les actions telles que le chargement de données cartographiques à l'entrée dans une région
- *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 al[21], les chercheurs revendiquent rarement l'exactitude du déclenchement de traitements contextuels, c'est à dire les premières apparitions de travaux concrets sur des systèmes informatiques contextuels. D'ailleurs, la quasi-totalité des spécialistes du domaine sont d'accord pour dire que les premiers travaux de recherche dans le domaine des systèmes informatiques sensibles au contexte concernent le système de badges actifs d'Olivetti. Le badge actif a été développé entre 1989 et 1992. Son but est de fournir des informations inhérentes aux positions avec précision, autrement dit, un moyen de localisation direct du personnel dans un immeuble pour acheminer automatiquement les appels téléphoniques là où il faut. Pour ce faire, chaque personne porte un badge qui transmet périodiquement des signaux vers un système centralisé de localisation. Le plus grand système de badges actifs actuellement utilisé se trouve au laboratoire d'informatique de l'université Cambridge, comptant plus de 200 badges et de 300 capteurs utilisés quotidiennement.

Dey[21] puis Chen et Kotz[19] ont recensés des travaux de recherche relatifs au contexte, en mettant l'accent sur les applications, les informations contextuelles utilisées, et la manière dont ils s'en servent. La plupart des systèmes informatiques sensibles au contexte font généralement appel surtout à la localisation de l'utilisateur comme information

contextuelle. Cela dit qu'il est plus ou moins rare qu'un index temporel est utilisé ainsi que la localisation des objets avoisinants. Cela pourrait s'expliquer par les difficultés de développements qui concernent les prélèvements des informations contextuelles ainsi que leur traitement.

2.4. SENSIBILITE AU CONTEXTE

Comme nous l'avons vu, l'idée de sensibilité au contexte émerge naturellement du concept d'informatique ubiquitaire. En effet, le fait de disposer d'informations contextuelles sert à adapter l'application au contexte perçu afin de mieux répondre aux attentes des utilisateurs. L'idée principale de l'informatique sensible au contexte est de sortir le plus possible les humains de la boucle de contrôle des machines, en réduisant les interactions entre l'humain et la machine. Par exemple : sur les réseaux sociaux, la sensibilité au contexte peut s'amplifier selon les interactions courantes de l'utilisateur, une fois les préférences du profil signalées, les suggestions des préférences s'appliqueront automatiquement au contexte perçu.

2.4.1. DEFINITION

La notion de sensibilité au contexte (context-awareness en anglais) a donc été produite dans le cadre des recherches sur l'informatique ubiquitaire. Schilit et Theimer[14] furent les premiers à proposer une définition de la sensibilité au contexte : il s'agit de la capacité d'un système à découvrir et à réagir à des changements dans l'environnement où il se trouve. Ils signalent également l'importance de l'adaptation du système à ces changements.

2.4.2. SYSTEME SENSIBLE AU CONTEXTE

Pour Abowd, Dey et al.[18], un système est sensible au contexte s'il utilise celui-ci pour fournir à l'utilisateur des informations et des services pertinents. Cette dernière (pertinence) dépend de l'activité que l'utilisateur est en train de réaliser. Ces auteurs proposent également une classification des systèmes sensibles au contexte selon leur réponse aux changements de contexte. Une réponse qui peut soit présenter des informations ou des services à l'utilisateur, soit exécuter automatiquement un service, soit stocker des données contextuelles.

De sa part, Miraoui[24] considère qu'un système est dit sensible au contexte s'il peut changer automatiquement ses formes de 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.

2.4.3. ARCHITECTURE GENERALE D'UN SYSTEME SENSIBLE AU CONTEXTE

L'architecture des applications sensibles au contexte est bien différente de celle des applications mobiles d'interactions classiques. Dey[21] confirme que les applications d'interactions classiques se différencient des applications sensibles au contexte au niveau des données explicites manipulées. Les données manipulées par les premières applications sont soit des variables internes de l'application soit des données explicites des utilisateurs. Pour les applications sensibles au contexte, en plus des variables internes de l'application et des données explicites des utilisateurs, elles traitent aussi les informations de contexte. Par conséquent, de supplémentaires traitements sont désignés par les systèmes sensibles au contexte. Ces traitements concernent la capture de ces informations de contexte, la gestion et l'utilisation de ces informations pour fournir une sortie convenable[25].

Selon[25]Plusieurs travaux ont proposé des architectures sensibles au contexte[24][27][27][28]. Un accord général, dans ces travaux, sur la séparation entre la capture des informations de contexte et l'utilisation de ces informations afin d'assurer l'extensibilité et la réutilisation du système. La figure I.1 montre l'architecture générale d'un système sensible au contexte. Elle est composée des couches suivantes : Capture de contexte, Interprétation de contexte, Gestion de contexte et Adaptation au contexte



Figure I.1 : Architecture de systèmes sensibles au contexte[25].

a. Capture de contexte

Selon[25]Les systèmes sensibles au contexte sont désignés pour être à l'écoute aux changements de l'environnement de système. La couche de capture est composée d'un ensemble de capteurs. Ces capteurs peuvent être classifiés selon qu'ils soient physiques ou logiques.

- *Capteurs physiques* : ces capteurs capturent des grandeurs physiques tels que la température, la position géographique, le son, la lumière, etc. Parmi ces capteurs on trouve : le GPS (Global Positioning System) pour déterminer les coordonnées d'un utilisateur, des caméras, des microphones, etc.
- *Capteurs logiques* : ces capteurs peuvent rassembler des données à partir des applications et des services logiciels. Par exemple, on consulte un calendrier électronique pour déterminer la position ou l'activité courante d'un utilisateur.

Dans le travail de Indulska[26], on donne une catégorisation des capteurs selon : physiques, virtuels et logiques. Les capteurs physiques sont pour les grandeurs physiques. Les capteurs virtuels sont les capteurs qui se basent sur les applications et les services et les capteurs logiques sont les combinaisons entre les capteurs physiques et virtuels [25].

b. Interprétation de contexte

Selon[25]Cette couche a pour but d'interpréter les données contextuelles fournies par les capteurs.

Elle sert à l'analyse et à la transformation des données brutes, fournies par la couche de capture, d'autres formats plus expressifs à l'application. Les transformations effectuées sur les données brutes concernent l'extraction, la quantification et l'agrégation.

Par exemple, les coordonnées GPS d'une personne peuvent être moins significatives qu'une adresse sous forme de numéro de rue et de ville. Cette couche peut aussi avoir le rôle de résolutions de conflits causés par l'utilisation de plusieurs sources de contexte. Ces sources peuvent donner des résultats contradictoires ou peuvent aboutir à des situations imprécises. Cette couche doit donc avoir des moyens pour résoudre ces conflits[27].

c. Gestion de contexte

A ce niveau, le contexte capturé et interprété doit être bien géré pour faciliter l'utilisation. La gestion de contexte contient le stockage et la représentation formelle des informations de contexte[25].

- Le stockage peut être centralisé ou distribué.

- Le stockage centralisé est le plus répandu et le plus utilisé puisqu'il facilite des mises à jour et des modifications des valeurs de contexte. Pourtant, cette solution se limite avec les contraintes de la capacité des équipements mobiles utilisés dans la sensibilité au contexte. Ces équipements sont caractérisés par leurs espaces de stockage restreints[25].

- Le stockage distribué remédie bien à la contrainte de l'espace de stockage des équipements utilisés. Cependant, il inflige des fonctions additionnelles relatives à la synchronisation et à l'actualisation des valeurs de contexte.

- La représentation de contexte est la structuration de ce contexte selon un modèle. Le choix du modèle dépend fortement de mécanismes choisis pour adapter les actions du système au contexte. Plusieurs méthodes de représentation ont été proposées dans la littérature, en invoquant les diverses technologies de modélisation et de représentation des données. Ces méthodes sont présentées selon[25] :

- *Représentations à base de paire (attribut, valeur)* : c'est la structure des données la plus simple pour la modélisation des informations contextuelles. L'*attribut* représente un élément de contexte tel que la localisation de l'utilisateur. La *valeur* est la valeur actuelle de cette information. Par exemple { Localisation="La gare", Terminal="PDA", Heure="t1"}. Cette représentation est caractérisée par sa facilité d'implémentation. Cependant, elle manque d'une force d'expressivité et ne permet pas de présenter les relations entre les éléments de contexte. La représentation la plus connue illustrant ce modèle est le modèle Context Toolkit de Dey[28].

- *Représentations basées sur XML* : ces modèles utilisent une structure de données hiérarchique. Plusieurs modèles de description des informations de contexte se

dérivent à partir de ce langage, à savoir UAProf⁵(User Agent Profile)[29] et CC/PP⁶ (Composite Capabilities/Preference Profile)[30], [25].

➤ *Représentations graphiques* : Ces modèles consistent à modéliser les informations contextuelles selon un graphe conceptuel. Quan et al.[31] ont utilisé une représentation graphique basée sur UML (Unified Modeling Language) pour la modélisation des informations contextuelles pour la gestion des services web[25].

➤ *Représentation orientée objet* : l'approche objet est utilisée pour pouvoir intégrer facilement la représentation du contexte au sein de l'application qui en dépend. Cette représentation du contexte s'appuie sur les propriétés de nommage, d'encapsulation, de réutilisation et d'héritage.

➤ *Représentations basées sur des ontologies* : Une ontologie est un ensemble structuré de concepts. Les concepts sont organisés dans un graphe dont les relations peuvent être des relations sémantiques ou des relations de composition et d'héritage. Elle fournit un vocabulaire représentatif pour un domaine donné, un ensemble de définitions et d'axiomes qui contraignent le sens des termes de ce vocabulaire de manière suffisante pour permettre une interprétation consistante des données représentées au moyen de ce vocabulaire[25].

d. Adaptation au contexte

L'adaptation au contexte est l'ensemble des mécanismes de réactions prévues suites aux changements de contexte. L'adaptation se base sur un ensemble de règles d'adaptation. Ces règles sont implémentées selon des langages de programmations traditionnels ou bien en utilisant la logique de prédicats. Pour remédier au caractère incertain de contexte, certains travaux adoptent la logique floue ou la logique probabiliste. Plusieurs formes d'adaptation peuvent être se présentées[32]:

⁵**UAProf (User Agent Profile)** :La spécification concerne les capacités de capture et les informations de préférence pour les périphériques sans fil. Ces informations peuvent être utilisées par les fournisseurs de contenu pour produire du contenu dans un format approprié pour le périphérique spécifique.

⁶**CC/PP (Composite Capability/PreferenceProfiles)** :est une spécification pour définir les capacités et les préférences des utilisateurs. Les informations fournies par CC/PP peuvent être utilisées pour guider le processus d'adaptation du contenu.

- *Adaptation architecturale* : désigne les changements faits, en temps réel, dans la structure des composants du système et/ou dans les interactions entre eux en utilisant un modèle architectural du système[25].
- *Adaptation compositionnelle* : concerne les modifications de la structure et du comportement d'une entité logiciel, en temps réel, en réponse aux changements arrivés à son environnement d'exécution[25].
- *Adaptation structurelle* : consiste en la mise à jour de sa structure en préservant son comportement et ses services. Une adaptation structurelle signifie le changement dynamique de type des composants de l'application tel qu'une signature d'une méthode[25].
- *Adaptation comportementale* : désigne les changements dynamiques dans la phase de l'exécution d'un composant logiciel d'une manière non intrusive (par exemple : en changeant sa configuration ou bien en interceptant ses requêtes et réponses)[25].
- *Adaptation de contenu* : concerne la transformation et la manipulation des contenus en se basant sur les caractéristiques de l'application et des terminaux utilisés [25].

2.5. Solutions pour la prise en compte du contexte

Comme l'a signalé Edwards[33], la signification et l'utilisation du contexte changent très souvent; c'est une notion fluide et peut-être même ambiguë. De ce fait, il est difficile de créer une infrastructure pour la prise en compte du contexte utile pour toutes les applications. Malgré cela, on peut trouver dans la littérature plusieurs travaux qui ont proposé des plateformes pour la gestion du contexte. Bien que nous soyons encore loin d'une plate-forme unifiée et automatique pour la gestion du contexte, ces travaux ont fait des progrès remarquables. Nous citons ici les exemples les plus pertinents par rapport à notre étude.

Dey et al.[34]ont proposé la boîte à outils Java Context Toolkit⁷, qui fournit notamment une architecture d'acquisition de données contextuelles basée sur des widgets. Les widgets sont des composants logiciels qui encapsulent les capteurs physiques ou logiciels qui produisent les données de contexte. Context Toolkit fournit à son tour des objets qui facilitent les communications réseau entre les widgets et les serveurs ou les applications qui veulent surveiller les données de contexte que les widgets produisent. Il fournit aussi des moyens pour stocker les données capturées. Dans Context Toolkit, les données de contexte se présentent sous la forme clé-valeur. Il est possible d'utiliser des unités pour les mesures, sachant que

⁷**ContextToolkit** : La boîte à outils contextuelle vise à faciliter le développement et le déploiement d'applications contextuelles.

leur sémantique est à définir par l'application. L'avantage de cette boîte à outils est sa simplicité, car tout en restant assez complète, la mise en œuvre de l'architecture d'acquisition de contexte dans une application requiert peu d'effort.

Chen[23] propose l'architecture Context Broker Architecture (CoBrA)⁸. Cette architecture utilise des agents pour la prise en compte du contexte dans des espaces intelligents. Dans chaque espace, on trouve un courtier central (broker en anglais), qui gère un modèle central du contexte qu'il construit avec les données acquises. Il partage ce modèle avec les dispositifs, les services et les agents qui sont dans le même espace. Le courtier peut aussi fournir des interprétations des données de contexte qu'il détient grâce à l'utilisation d'ontologies OWL⁹[35]. Le courtier fournit également des moyens pour garantir la confidentialité des données de contexte, ce qui est important.

Le système Service-Oriented Context-Aware Middleware (SOCAM), proposé par Gu et al.[36], est un intergiciel distribué qui permet l'implantation rapide de services mobiles sensibles au contexte. Comme dans le cas de CoBrA, cet intergiciel utilise des ontologies OWL pour représenter et pour effectuer des raisonnements sur le contexte. Dans l'architecture de SOCAM, on trouve des composants tels que les fournisseurs de contexte, qui capturent les données de contexte et les représentent en OWL, les interpréteurs de contexte, qui fournissent des services de raisonnement sur le contexte, les bases de données de contexte, qui stockent les données de contexte, les services sensibles au contexte, qui adaptent leur fonctionnement au contexte courant, et les services de découverte de services, qui permettent l'annonce et la découverte des fournisseurs et des interpréteurs de contexte.

3. ADAPTATION

L'adaptabilité est la capacité d'un système à s'adapter à des changements.

⁸**Context Broker Architecture (CoBrA)** : est une architecture basée sur un agent pour supporter des systèmes contextuels dans des espaces intelligents (par exemple, des salles de réunion intelligentes, des maisons intelligentes et des véhicules intelligents). Au centre de cette architecture est un agent intelligent appelé courtier de contexte qui maintient un modèle de contexte partagé au nom d'une communauté d'agents, de services et de dispositifs dans l'espace et fournit des protections de la vie privée aux utilisateurs dans l'espace en appliquant les règles de politique qui Ils définissent.

⁹**OWL (Web Ontology Language)** : est un langage de représentation des connaissances construit sur le modèle de données de RDF. Il fournit les moyens pour définir des ontologies web structurées. Il est basé sur les recherches effectuées dans le domaine de la logique de description.

Ces changements peuvent avoir lieu dans le système lui-même ou dans une partie de l'environnement qui peut influencer sur le comportement du système[37]. Ces systèmes sont souvent appelés auto-adaptatifs.

Selon McKinley et al.[38], l'adaptation est essentielle dans l'informatique ubiquitaire et ils signalent que les dispositifs mobiles doivent s'adapter aux contraintes des réseaux ad-hoc sans fil. Un système adaptatif doit s'adapter aux changements lors de son exécution[37].

Satyanarayanan[4]justifie le besoin d'adaptabilité comme réponse au compromis entre l'offre et la demande des ressources :

« L'adaptabilité est nécessaire quand il y a des disparités significatives entre l'approvisionnement et la demande d'une ressource.»

3.1. DEFINITION

Plusieurs auteurs ont essayé de définir l'adaptation. Nous citons ici les définitions les plus répandues[37].

En 1997, Laddaga[39]proposa une première définition:

« Un logiciel auto-adaptatif évalue son propre comportement et change ce comportement quand l'évaluation indique qu'il n'est pas en train de réaliser ce qu'il est censé réaliser, ou que de meilleures fonctionnalités ou performances sont possibles. »

Peu après, Oreizy et al.[40]ont donné une définition plus complète. Cette définition ajoute la notion d'environnement, que nous pouvons assimiler au contexte:

«Un logiciel auto-adaptatif modifie son propre comportement en réponse à des changements dans son environnement de fonctionnement. Par environnement de fonctionnement, nous nous référons à tout ce qui est observable par le système logiciel, comme les données fournies par l'utilisateur, les matériels et les capteurs externes ou des mesures. »

Ces premières définitions ne mentionnent pas explicitement l'informatique ubiquitaire. McKinley et al.[38]ont signalé que l'informatique ubiquitaire introduit un besoin d'adaptation en raison de la mobilité des utilisateurs et des interactions dynamiques avec l'environnement.

Donc L'adaptation, peut être définie comme la modification d'un système afin qu'il puisse assurer ses fonctions dans des conditions spécifiques, voire nouvelles.

L'adaptation est une tâche à la fois nécessaire et complexe. Nécessaire, pour répondre à de nouveaux besoins, aux changements de disponibilité des ressources, à la nécessité de corriger des dysfonctionnement, et à la prise en compte de la mobilité des utilisateurs[41]. Complexe, parce qu'elle peut avoir des conséquences sur le logiciel en de nombreux points, et nécessite à la fois une excellente compréhension du logiciel et une technicité importante.

3.2. LES ASPECTS DE L'ADAPTATION

Les principaux aspects de l'adaptation

3.2.1. Pourquoi (why)

« Pourquoi adapter ? » voilà la première question qui concerne l'adaptation. Fondamentalement, l'objectif de l'adaptation qui se trouve en pole-position consiste à accélérer et simplifier voir optimiser l'utilisation des systèmes informatiques, en présentant à l'utilisateur ce qu'il souhaite sur tous les plans ayant une possibilité d'adaptation et ainsi rendre des systèmes complexes plus utilisables adéquatement.

Selon Malinowski[42], l'objectif principal de l'adaptation consiste à présenter plus facilement et plus efficacement des interfaces.

Par contre, pour atteindre ces objectifs, il est absolument nécessaire pour Dietrich & al.[43] d'avoir à disposition une interface utilisateur qui s'adapte à un maximum de groupes hétérogènes d'utilisateurs et qui prend en considération l'expérience croissante de l'utilisateur.

Dans le domaine de documents numériques[44], principalement, le rôle de l'adaptabilité est de minimiser l'effort de l'utilisateur consacré à l'exploration des capacités du système afin d'optimiser celui nécessaire pour la résolution d'un problème.

Comme nous le savons, c'est clair qu'il est difficile de concevoir et de réaliser un logiciel qui conviendra avec perfection à des millions d'utilisateurs[45], par contre il est possible de développer une certaine maniabilité des systèmes pour qu'il y est une capacité d'ajuster leur interface conformément aux compétences, aux connaissances ou encore aux

préférences de chaque utilisateur. Pour y parvenir à la réalisation de cette adaptation, nous verrons que les modèles d'utilisateur qui existent sans être manifesté explicitement et de la tâche qui doit suivre équitablement sont essentiels, de même pour la séparation de l'interface utilisateur du reste de l'application[46].

Les systèmes adaptatifs sont utilisés dans de nombreux domaines pour résoudre d'innombrables tâches. L'ensemble de fonctions suivants, établie par[47] et adaptée par[48] présente quelques exemples d'usages : aider l'utilisateur à trouver de l'information ; personnaliser l'information dite « sur mesure » pour l'utilisateur ; recommander des produits (par exemple pour le commerce électronique) qu'on peut, d'ailleurs, la qualifier de publicité adaptée ou sur mesure ; fournir une aide appropriée, selon le contexte d'utilisation ; optimiser l'apprentissage ; encadrer un dialogue homme-machine ; favoriser la collaboration entre utilisateurs et/ou avec la machine...

3.2.2. Quoi (What)

Une autre interrogation importante réside dans la question « quoi adapter ? ». Parfois, seulement l'aide, les tutoriels ou les conseils sont adaptés. Dans d'autres cas, le dialogue entre le système et l'utilisateur est modifié par l'adaptation. Il peut également être question d'adaptation au contenu, à la navigation, à la présentation, aux fonctionnalités disponibles, etc. La plupart du temps, c'est uniquement la partie présentation de l'interface qui est adaptée et/ou adaptable ; Dans d'autres cas, la présentation et quelques fonctions du système sont adaptées conjointement. Ce dernier point est appelé « malléabilité » (malleability / tailorability) par certains auteurs[51], [52], et demeure assez proche de la notion de « flexibilité » proposée à son tour par[53] La malléabilité implique une adaptation par l'utilisateur lui-même : « Une application malléable est à la fois utilisable et modifiable par ses propres utilisateurs. L'activité de redéfinition correspond elle-même à une des facettes de son utilisation. »[54]. Le fait de pouvoir modifier (adapter ou manier) les fonctionnalités d'un système sans pour autant l'interrompre[55] est un sujet de recherche complexe, car cela suppose un maintien et une gestion fine de la cohérence globale du système.

3.2.3. Qui (who)

L'étape suivante nous amène à une réflexion en se demandant « Qui adapte ? ». Plus précisément, il s'agit là de définir qui contrôle l'adaptation et qui la réalise, concrètement. Elle peut être exécutée soit par l'utilisateur, soit par le système, soit encore par les deux, conjointement. Selon[56], lorsque le système contrôle l'adaptation, on parle de «

personnalisation » et plus généralement « d'adaptativité ». Lorsque l'utilisateur contrôle d'adaptation, on parle alors de « customisation », notamment lorsqu'on a le choix entre plusieurs options, et, plus généralement, « d'adaptabilité ». Il est toujours difficile de parler d'une adaptation semi-automatique car il n'existe pas de terme spécifique pour désigner le fait que l'adaptation soit à la fois réalisée par l'utilisateur et le système, certains bonhommes du domaine parlent de « contrôle mixte ». La personnalisation et la customisation visent à répondre aux besoins et aux caractéristiques particulières de chaque utilisateur[57]. En effet, ces mots viennent du domaine du commerce électronique et insistent sur une dimension individuelle de l'adaptation. Selon[58], la personnalisation est spécifique à l'utilisateur final et se base sur ses intérêts implicites durant la session en cours, et plus ou moins que les précédentes. Certains auteurs comme[59],[60] et[61] font remarquer que parfois, personnalisation, customisation et adaptation sont synonymes dans les travaux de[62],[63] ou.[64]. Pour Fischer, la différence fondamentale entre les systèmes adaptatifs et adaptable est que le premier terme désigne une adaptation par le système lui-même, à la tâche en cours et à l'utilisateur, alors que le second terme signifie que l'utilisateur change, avec parfois l'aide substantielle du système, les fonctionnalités du système[65]. Cependant, contrôler l'adaptation ne veut pas forcément dire pour autant exécuter l'adaptation. L'adaptation pourrait être effectuée par le système ou par les utilisateurs. Les utilisateurs peuvent être, par exemple, les concepteurs de systèmes ou les administrateurs, des experts ou des utilisateurs finaux[66].

3.2.4. Quand (when)

Essayons d'étudier à présent la question « Quand adapter ? ». On pourrait dire que l'adaptation peut être effectuée à différents moments. Le temps de la conception (design time) est souvent en opposition avec le temps de l'exécution (runtime). Certains auteurs opposent ainsi l'adaptation statique (qui s'effectue avant et entre les sessions) à l'adaptation dynamique, qui s'effectue pendant l'interaction[43]. Au cours d'une session, l'adaptation peut se faire de manière non alternative, autrement dit continue, avant ou après certaines fonctions prédéfinies, dans des situations particulières ou à la demande de l'utilisateur.

Dans la terminologie de[67], l'adaptativité fait directement allusion au fait que les adaptations se produisent en cours d'exécution du système, qu'elles soient réalisées par le système ou par l'utilisateur, tandis que l'adaptabilité, elle, désigne le fait que les adaptations interviennent avant l'exécution du système, par exemple, lors d'une première

installation[68]. Ici, c'est donc l'opposition « dynamique » versus « statique » qui entre en considération pour la distinction entre l'adaptativité et l'adaptabilité.

3.2.5. Où (where)

Il est inévitable de passer à la question « Où adapter ? ». L'adaptation peut être interne ou externe. Une adaptation est dite interne ou fermée lorsque les mécanismes d'adaptation sont intégrés au système lui-même alors qu'elle sera dite externe ou ouverte, et comme son nom l'indique, lorsque l'adaptation se fait en dehors du système, par exemple grâce à un service Web[69].

Les mécanismes d'adaptation externes semblent être préférés dans des projets aux technologies ambiantes, car ils donnent la possibilité de découvrir de nouveaux services au moment de l'exécution, et de faire l'adaptation à la volée, dans le respect de la philosophie des architectures orientées services (SOA).

3.2.6. A quoi (to what)

Passons à présent à la question « A quoi adapter ? ». En d'autres termes, que/quoi cible l'adaptation ? A quoi le système doit-il s'adapter ? Le système s'adapte, en premier temps, à l'utilisateur puis, à la plate-forme, à l'environnement ou à une combinaison des ces trois éléments.

3.3. Raisons d'adaptation

Les raisons d'adaptation peuvent être classées en quatre catégories :

3.3.1. Adaptation corrective

Dans certains cas, on remarque que le procédé en cours d'exécution ne se comporte pas correctement ou comme prévu.

3.3.2. Adaptation adaptative

Même si le procédé s'exécute correctement, parfois son environnement d'exécution change (Nouvelle technologie par exemple). Dans ce cas, le procédé est adapté en réponse aux changements affectant son environnement d'exécution.

3.3.3. Adaptation évolutive

Au moment du développement du procédé, certaines fonctionnalités ne sont pas prises en compte. Avec l'évolution des besoins de l'utilisateur, le procédé doit être étendu avec de nouvelles fonctionnalités. Cette extension peut être réalisée en ajoutant un ou plusieurs éléments pour assurer les nouvelles fonctionnalités ou simplement étendre les fonctionnalités des éléments existants.

3.3.4. Adaptation perfective

L'objectif de ce type d'adaptation est d'améliorer les performances du procédé. A titre d'exemple, on se rend compte que le déroulement d'une activité n'est pas optimisé. On décide alors de remplacer le comportement de cette activité.

3.4. Les types d'adaptation

Dans leur travail, Salehie et Tahvildari[37] proposent une taxonomie de l'adaptation qui essaie de classer les propriétés de l'adaptation. Notamment, on y trouve des éléments qui répondent à la question de comment réaliser l'adaptation. Parmi ces éléments, les considérations suivantes nous semblent importantes :

3.4.1. L'adaptation statique vs dynamique

Dans l'adaptation statique, le processus d'adaptation est codé en dur et sa modification nécessite une recompilation du système ou de certains de ses composants. Tandis que dans l'adaptation dynamique, les politiques d'adaptation sont définies et gérées à l'extérieur du système, et par conséquent, elles peuvent être configurées au cours de l'exécution du système.

3.4.2. L'adaptation externe vs interne

Les approches d'adaptation peuvent ainsi être classées dans deux catégories en fonction de la séparation entre le mécanisme d'adaptation et la logique de l'application. Dans l'adaptation externe, le système adaptatif est composé d'un moteur d'adaptation qui implémente le processus d'adaptation du système adaptable. Tandis que dans l'adaptation interne, le processus d'adaptation est intégré dans le code de l'application.

3.4.3. L'adaptation architecturale vs comportementale

Selon la façon de mettre en œuvre l'adaptation d'un système, nous pouvons trouver dans la littérature deux types principaux d'adaptation : l'adaptation comportementale et l'adaptation architecturale.

L'adaptation comportementale consiste à changer le comportement d'un système lors son exécution[70]. L'intérieur des composants est modifié, mais le nombre de composants ou la façon dont ils sont connectés restent les mêmes. Le composant adapte son fonctionnement interne en fonction des changements.

Garlan et al.[49] introduit la notion d'adaptation architecturale qui consiste à changer la structure du système lors de son exécution afin de s'adapter aux changements. Selon Salehie et Tahvildari[37], l'adaptation d'une architecture peut nécessiter le redéploiement de certains composants du système ainsi que la façon dont ils sont connectés.

3.4.4. L'adaptation basée sur les modèles vs sans modèles

Dans le second cas, le mécanisme d'adaptation ne dispose pas d'un modèle spécifique de l'application ni de son contexte ; le mécanisme contient toutes les informations nécessaires implicitement dans son code interne. Dans l'adaptation basée sur les modèles, le mécanisme d'adaptation s'appuie sur des modèles explicites de l'application et du contexte qui permettent de prendre en compte toutes les informations afin de réaliser l'adaptation.

3.4.5. L'adaptation réactive vs proactive

L'adaptation réactive agit une fois que les changements auxquels elle répond se sont produits. L'adaptation proactive intervient quand le système prédit que le changement va se produire.

3.4.6. L'adaptation du contenu vs la présentation

Dans les systèmes qui présentent des contenus aux utilisateurs, deux types d'adaptation sont possibles : l'adaptation du contenu et l'adaptation de la présentation. Dans le premier cas, le système change les contenus à présenter en fonction de l'utilisateur. Par exemple, un système vidéo peut afficher des dessins animés si l'utilisateur est un enfant, tandis que, pour les adultes, il affichera des actualités économiques. L'adaptation de la présentation, en revanche, change la façon de présenter un même contenu. Par exemple, un système qui

présente des informations en mode texte peut basculer vers l'audio lorsqu'une personne malvoyante l'utilise.

CONCLUSION

Nous avons présenté dans ce chapitre tout d'abord le domaine de l'informatique ubiquitaire ainsi que les caractéristiques et quelques exemples d'applications dans ce domaine. Ensuite, on a abordé la notion de contexte.

On a commencé par la définition du contexte, puis on a abordé taxonomie des données contextuelles, ensuite on a mis en évidence sensibilité au contexte, et enfin on a détaillé l'adaptation d'utilisateur.

Pour conclure, un système sensible au contexte peut utiliser un modèle de description du contexte qui lui offre une description abstraite de ces informations de contexte afin de pouvoir les utiliser. Cette abstraction peut être fournie par des modèles de description du contexte.

Le chapitre suivant présentera un état de l'art des profils utilisateurs.

Chapitre II :

Les profils Utilisateurs

★

Chapitre II:

INTRODUCTION

1.DEFINITION DE PROFIL

2.DEFINITION DU PROFIL UTILISATEUR

3.UTILISATION DU PROFIL UTILISATEUR

4.REPRESENTATION DES PROFILES UTILISATEURS

4.1.REPRESENTATION ENSEMBLISTE

4.2. REPRESENTATION PAR RESEAUX SEMANTIQUES

4.3.REPRESENTATION CONCEPTUELLE

5.CONSTRUCTION D'UN PROFIL UTILISATEUR

6. MODELISATION DU PROFIL UTILISATEUR

7. EXPLOITATION DU PROFIL UTILISATEUR

8. LES APPROCHES DE CONSTRUCTION D'UN PROFIL UTILISATEUR

CONCLUSION

Introduction

La modélisation de l'utilisateur fait partie du domaine des interactions homme machine et consiste à étudier le processus d'extraction et d'analyse des interactions entre l'utilisateur et le SI pour construire des profils utilisateurs. La modélisation de l'utilisateur peut alors être perçue comme un cas particulier de processus d'extraction de connaissances à partir de données (KDD en anglais), lorsque celles-ci sont issues des interactions entre les utilisateurs et les SI. D'un point de vue du domaine de gestion des connaissances, les profils utilisateurs construits par ce processus peuvent être perçus comme des données, des informations ou des connaissances en fonction des usages qui en sont faits

1. DEFINITION DE PROFIL

Un profil est un modèle utilisateur ou source de connaissance qui contient des acquisitions sur tous les aspects de l'utilisateur pouvant être utiles pour le comportement du système [72]

Un profil peut être relatif à une personne seule, ou à un groupe de personnes ayant des points communs, tels que par exemple les membres d'un groupe de travail.

2. DEFINITION DU PROFIL UTILISATEUR

Selon Guaussier[73], "Toutes les variations qui caractérisent un utilisateur ou un groupe d'utilisateurs peuvent se regrouper sous le terme de profil utilisateur". A partir de cette définition, on peut déduire que le profil utilisateur est une structure qui permet de modéliser et de stocker les données caractérisant l'utilisateur [74], en particulier ses données personnelles (nom, prénom, âge, adresse, contact, profession...); ses préférences (une préférence est une expression permettant de hiérarchiser les concepts aux quels l'utilisateur s'intéresse [75], ses centres d'intérêts; son but courant et l'historique de ses interactions avec le système.

Ainsi, un profil utilisateur peut être perçu comme une donnée ou une information lorsqu'il est utilisé pour les systèmes désirant répondre à ses besoins spécifiques (adaptation de l'information à l'utilisateur) :

- *Personnalisation* (en recherche d'information par exemple), ou recommandation (dans le e-commerce par exemple) par exemple. Par contre un profil utilisateur peut

également être perçu comme une connaissance dans les cas des systèmes d'analyses comportementales : détection de fraudes (banques, assurances, etc.),

➤ *Détection de réseaux terroristes* (bases de données dédiées, Web, etc.),
détection de pédophiles (Web), détection de leaders (Web, intranets d'entreprises, etc.), etc.

Outre les informations d'identification de base, le profil utilisateur peut regrouper des informations très diverses selon les besoins. Parmi celles-ci [76]:

➤ Des caractéristiques personnelles pouvant influencer fortement l'interaction (âge, sexe, etc.).

➤ Les intérêts et les préférences générales relatives à la tâche à accomplir, qui permettent une adaptation aux attentes de l'utilisateur.

➤ Les compétences ou le niveau d'expertise relative à la tâche (pour déterminer par exemple un degré d'autonomie et déceler un besoin d'aide ou de formation).

Il existe plusieurs formalismes pour exprimer ces informations, le plus souvent on utilise un couple (attribut, valeur) qui peut être un ensemble de mots clés pondérés ou de vecteurs de termes pondérés [77] [78]. Il est possible aussi d'utiliser des prédicats [79] [80] et des expressions à la clause de Horn [81] pour exprimer les préférences. Enfin beaucoup de travaux représentent les centres d'intérêts sous forme de concepts classés en utilisant des ontologies ou des réseaux sémantiques [82].

Les données du profil utilisateur sont représentées différemment selon les besoins. En général, on les stocke dans une table sous la forme de couples attribut-valeur où chaque couple représente une propriété du profil. Les propriétés peuvent éventuellement être regroupées par catégories.

Les valeurs peuvent être de tous types (numériques, alphanumériques) mais elles peuvent aussi stocker des distributions de probabilités (pour les services adaptatifs).

3. UTILISATION DES PROFILS UTILISATEURS

Les profils utilisateurs sont utilisés dans les logiciels collectifs, tels que les systèmes d'exploitation, les systèmes de gestion de bases de données, les moteurs de recherche ou les sites de vente en ligne.

4. REPRESENTATION DES PROFILS UTILISATEURS

Les profils sont généralement représentés de trois manières dans la littérature : représentation ensembliste, représentation par réseaux sémantiques, et représentation conceptuelle.

4.1. REPRESENTATION ENSEMBLISTE

L'approche ensembliste consiste à représenter le profil de l'utilisateur par des paquets de termes pondérés. On parle également de représentation vectorielle par analogie au modèle vectoriel de Salton[83] sur lequel elle se base. Ces paquets de termes, traduisant les centres d'intérêts de l'utilisateur peuvent être regroupés différemment selon l'approche suivie pour exploiter le profil de l'utilisateur. On distingue dans la littérature quatre grandes approches de représentation de profils utilisateurs basés sur la représentation ensembliste :

- Par liste de mots clés où chaque mot correspond à un centre d'intérêt spécifique [84].
- Par vecteur de termes pondérés pour chaque centre d'intérêts [85] [86].
- Par ensemble de vecteurs de termes pondérés (ou non) indépendants, pour prendre en compte des centres d'intérêts multiples[87] où chaque vecteur correspond à un domaine d'intérêt [88].

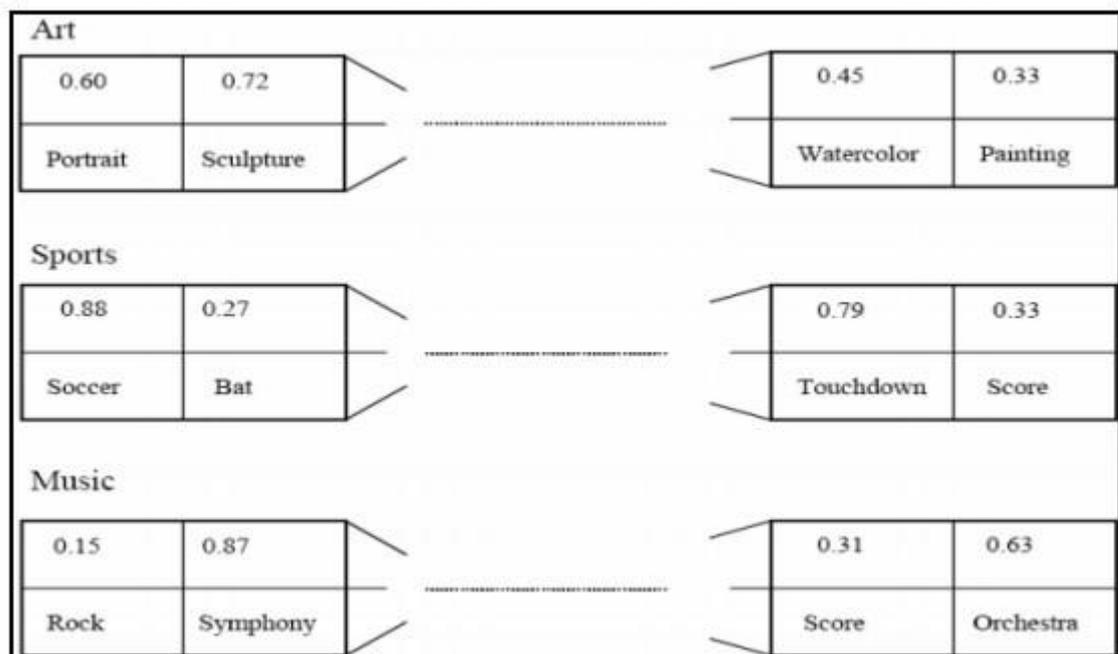


Figure II.1: Un exemple de profil représenté par des mots clés [88].

La représentation ensembliste fait partie des premières représentations de profils utilisateurs qui ont été utilisées. La pondération des termes est généralement basée sur le schéma tf .idf communément utilisé en recherche d'information [89]. Le poids associé à chaque terme permet de représenter son degré d'importance dans le profil de l'utilisateur. La figure II.1 donne un exemple de profil utilisateur représenté par des mots clés pondérés. Ce profil contient trois centres d'intérêts : Art, Sport, et Musique.

Chaque centre d'intérêt est représenté par un ensemble de termes pondérés. Music = <(Rock, 0.15), (Symphony, 0.87)...> est un extrait de l'ensemble de termes pondérés représentant le centre d'intérêt Music.

Plusieurs systèmes d'accès personnalisé utilisent ce type de représentation. On peut citer Anatonomy[90], un système personnalisé de consultation de nouvelles et de journaux en ligne, Fab[91] un système de recommandation de pages Web, Letizia[92], un système d'aide à la navigation, et Syskill&Weber[93], un système de recommandation. Tous ces systèmes proposent des profils utilisateurs représentés par une liste de mots clés.

L'efficacité des profils dans cette approche dépend fortement du degré de généralisation pour représenter les centres d'intérêts. Le problème est lié à l'application d'une analyse statistique de mots clés indépendamment de toute information contextuelle. Dans le cas d'une représentation en hiérarchies ou classes de concepts, le rapport généralisation/spécification existant naturellement dans ce genre de structure permet d'avoir une représentation plus réaliste du profil utilisateur. Ceci est par exemple utilisé dans la représentation par réseaux sémantiques.

4.2. REPRÉSENTATION PAR RESEAUX SEMANTIQUES

Afin d'adresser le problème de polysémie des termes inhérents à la représentation ensembliste, une première solution consiste à représenter le profil par un réseau de nœuds pondérés dans lequel chaque nœud représente un concept traduisant un centre d'intérêt utilisateur. Ce type de représentation offre le double avantage de la structuration et de la représentation associative (relations entre les termes) permettant de considérer l'ensemble des aspects représentatifs du profil. Les centres d'intérêts sont souvent représentés par des relations de paires de nœuds dans lesquelles chaque nœud contient un terme issu de données implicites utilisées pour construire le profil. Les arcs reliant les nœuds sont créés sur la base de cooccurrences entre ces termes.

Cependant, la représentation séparée de chaque mot par des nœuds dans le réseau sémantique n'est pas assez précise pour décliner les différentes significations des centres d'intérêts de l'utilisateur. Une alternative possible est d'exploiter les sources externes telles que les ontologies pour établir les liens entre les nœuds. Dans ce cadre, le système SiteIF propose d'utiliser les concepts inhérents à WordNet pour regrouper des termes semblables dans des concepts appelés des « ensembles de synonymes » ou des synsets. Le profil utilisateur est alors représenté comme un réseau sémantique dans lequel les nœuds sont les synsets et les arcs sont des cooccurrences des membres de synset avec le document intéressant l'utilisateur. Les nœuds et les poids des arcs représentent le niveau d'intérêt de l'utilisateur.

Une approche similaire a été étudiée par le système *InfoWeb*[93]. Initialement, chaque réseau sémantique contient une collection de nœuds unitaires dans laquelle chaque nœud représente un concept. Les nœuds du concept, appelés *planètes*, contiennent un vecteur unique de termes pondérés. Lorsque de nouvelles informations sur l'utilisateur sont collectées, le profil est enrichi en intégrant les termes pondérés dans les concepts correspondants. Ces termes sont stockés dans les nœuds auxiliaires (*satellites*) qui sont liés aux nœuds concepts (*planètes*) associés. La figure II.2 montre un exemple extrait d'un modèle d'utilisateur basé sur cette représentation. Sur cette figure, les cercles représentent les *planètes* (*Painting, Sculpture, Restauration, Environnement, Pollution*), et les carrés aux bords arrondis représentent les nœuds auxiliaires *satellites* liés aux *planètes* (pour la *Sculpture* par exemple, les *satellites* associés dans ce profil sont *Museum, Sculptor, Painter, et Chisel*). Les *planètes* peuvent également être reliées entre elles (cas de la relation entre *Pollution* et *Environnement* par exemple). Ces deux types de relations (entre *planètes* et entre *planètes* et *satellites*) permettent d'obtenir des profils sémantiques beaucoup plus structurés que de simples vecteurs de termes.

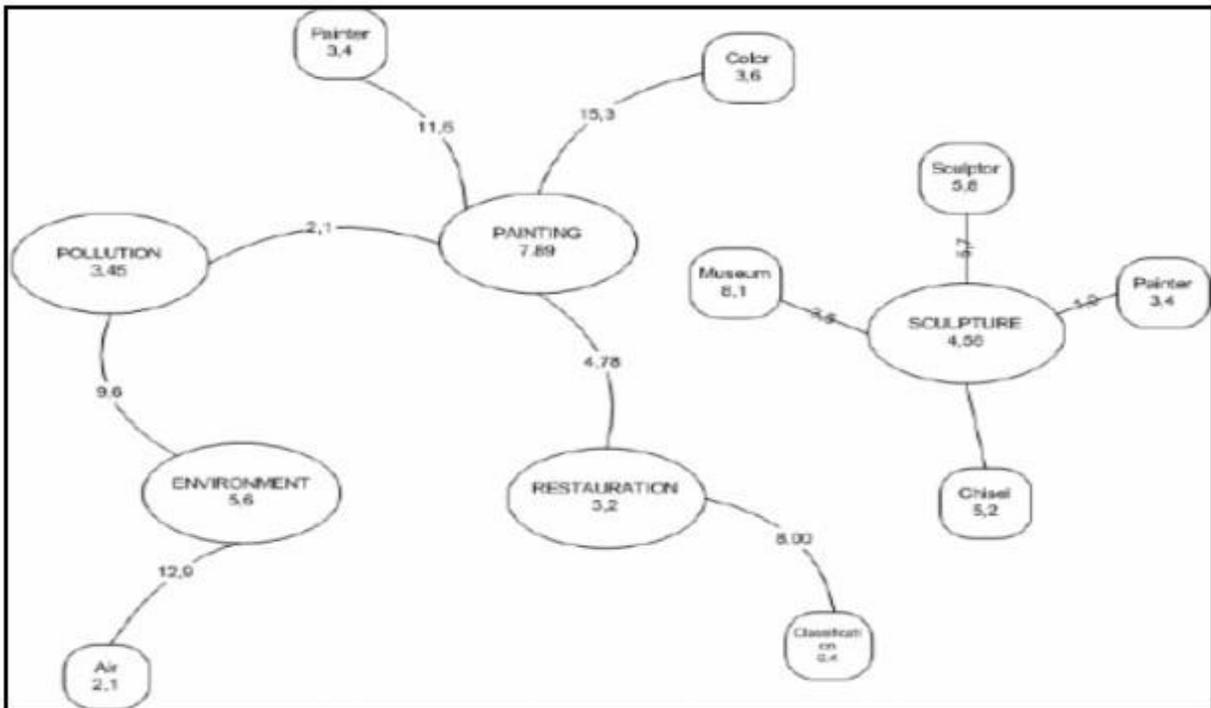


Figure II.2 : Exemple de profil sémantique de l'utilisateur

Cette représentation a été prolongée dans *WIFS*, une interface de filtrage d'information pour personnaliser les résultats du moteur de recherche d'*AltaVista23*. Dans ce système, le profil de l'utilisateur est représenté par trois composantes : *un entête*, intégrant les données personnelles de l'utilisateur, un ensemble de stéréotypes, et une liste de centres d'intérêts. Un stéréotype comporte un ensemble de centres d'intérêts, représentés par une classe d'information. Chaque classe contient trois champs : domaine (Domain), matière (Topic), poids (Weight), et deux autres informations supplémentaires : les liens sémantiques (Semantic Links) et la justification des liens (Justification Links).

4.3.REPRESENTATION CONCEPTUELLE

La représentation du profil met en évidence, dans cette approche, une ou plusieurs relations sémantiques entre les informations du profil. Suivant une direction proposée dans un contexte plus général [94], cette représentation offre une alternative intéressante à l'approche des réseaux sémantiques. En effet, les travaux actuels tendent à représenter le profil sous forme d'une ontologie de concepts personnels en se basant sur les connaissances contenues dans les ontologies plutôt que de construire les profils utilisateur seulement à partir des données implicites collectées de ses interactions [95].

La représentation est essentiellement basée sur l'utilisation d'ontologies ou de réseaux probabilistes

Dans ce type d'approche, les liens entre les concepts sont explicitement induits de l'ontologie et le profil résultant inclura des relations informationnelles plus diverses et spécifiques. La représentation conceptuelle est semblable à la représentation par réseaux sémantiques, dans le sens où elle représente les centres d'intérêts de l'utilisateur par un réseau de nœuds conceptuels. Cependant, dans l'approche conceptuelle, les nœuds correspondent à des domaines abstraits représentant les centres d'intérêt de l'utilisateur, contrairement à l'approche sémantique où les centres d'intérêts sont représentés par des mots spécifiques, ou ensemble de mots relatifs. La représentation conceptuelle peut également être assimilée à une approche ensembliste (vectorielle) du fait que les domaines sont souvent représentés comme les vecteurs de termes pondérés. Néanmoins, les termes de ces vecteurs sont regroupés pour former un domaine spécifique et non de simples mots clés.

De l'association des centres d'intérêts de l'utilisateur aux concepts des domaines de l'ontologie, on obtient un profil représenté sous forme d'une hiérarchie de concepts. Les données implicites issues des interactions de l'utilisateur sont classifiées dans ces concepts et l'intérêt de l'utilisateur pour de tels concepts est enregistré. Plusieurs mécanismes peuvent être appliqués pour exprimer le degré d'intérêt de l'utilisateur pour chaque domaine. Néanmoins la technique la plus utilisée est l'affectation d'un poids d'une valeur numérique pour chaque domaine[96].

5. CONSTRUCTION D'UN PROFIL UTILISATEUR

La construction du profil traduit un processus qui permet d'instancier sa représentation à partir de diverses sources d'information. Ce processus est généralement implicite et basé sur un procédé d'inférence du contexte et des préférences de l'utilisateur via son comportement lors de l'utilisation : [97]

- D'un système d'accès à l'information : requêtes et documents jugés explicitement ou implicitement pertinents (consultés et/ou imprimés et/ou sauvegardés etc.)
- D'un navigateur web : liens explorés, dernières pages visitées etc.
- D'autres applications : les applications de bureautique, les outils de messagerie électronique etc. [97]. Les informations extraites de ces sources sont organisées selon le modèle de représentation du profil à l'aide de différentes techniques.

La plus répandue est celle basée sur l'analyse statistique du texte selon l'algorithme de Rocchio. L'autre technique largement utilisée également dans les procédés de construction du profil, est celle de la classification appliquée aux informations collectées de l'utilisateur. On distingue plusieurs variantes dont la classification simple, la classification hiérarchique ou la classification Bayésienne [97].

6. MODELISATION DU PROFIL UTILISATEUR

L'introduction de la dimension utilisateur dans un processus d'accès à l'information, mérite, voire nécessite une réflexion sur la modélisation de l'entité utilisateur. La fiabilité ou qualité des profils est en effet d'une importance bien connue dans le domaine de la modélisation utilisateur. En effet, on constate que l'une des principales raisons du manque de performances des techniques de personnalisation est typiquement l'application d'un profil utilisateur hors contexte [98].

Les utilisateurs peuvent avoir des préférences générales, récurrentes et stables. Cependant, l'ensemble des informations contenues dans le profil ne sont pas forcément appropriées à toutes les situations de recherche. Le plus souvent, les systèmes n'utilisent seulement qu'un sous-ensemble de ces informations, qu'ils supposent pertinents pour la recherche en cours. Dès lors, le choix du profil adéquat constitue la principale réflexion lors de la mise en œuvre du SRIP. De ce fait, les questions fondamentales posées pour modéliser le profil utilisateur sont le «*Quoi* », le «*Comment* » et le «*Quand* » [99] :

- Quoi ?
 - Quelles propriétés informationnelles caractérisent l'utilisateur ?
 - Quelle structure informationnelle utiliser pour représenter l'utilisateur ?
- Comment ?
 - Comment collecter les informations du profil ?
 - Comment construire le profil de l'utilisateur ?
 - Comment détecter le contexte, le but de la recherche et les besoins à court/longtermes de l'utilisateur ?
 - Comment adapter le profil à l'évolution de l'utilisateur lui-même ?
 - Comment assurer la sécurité et la confidentialité des informations du profil ?
- Quand ?
 - Quand faut-il faire évoluer le profil de l'utilisateur ? [100].

7. EXPLOITATION DU PROFIL D'UTILISATEUR

Le domaine d'application conditionne fortement les techniques de personnalisation employées par le système. Cela a en effet un impact direct sur l'exploitation du profil dans la chaîne d'accès à l'information et par conséquent sur les mécanismes mis en œuvre. Les principales interrogations posées concerne le « Quoi » le « Comment » de la mise en œuvre : [100][99]

- Quoi ?
 - Quels services de personnalisation proposée : de la recommandation et/ou du filtrage, de l'aide à la navigation, un assistant personnel de recherche ?
 - Dans quelles étapes du cycle de vie de la requête faut-il intégrer le profil ?
 - Quelles informations du profil exploiter lors de l'accès à l'information ?
- Comment ?
 - Comment intégrer le profil de l'utilisateur dans le processus de personnalisation ?
 - Comment évaluer l'impact de la personnalisation sur le processus de recherche ? [101]

8. LES APPROCHES DE CONSTRUCTION D'UN PROFIL UTILISATEUR

Selon [75], plusieurs paramètres entrent en jeux lors de la construction du profil utilisateur, à savoir le moment de construction (le profil sera-t-il construit en amont de l'interaction avec le système ou appris au fur et à mesure?), l'implication de l'utilisateur (l'utilisateur renseigner a-t-il son profil explicitement ou le système le déduira par des algorithmes d'apprentissage?) et enfin, la nature des sources de données impliquées lors de la construction du profil (fichiers logs, données issues de l'utilisateur, information sur la structure et le contenu des documents consultés etc.).

Le choix de ces paramètres permet d'identifier trois approches de construction de profil cités par GOWAN, à savoir l'approche canonique, l'approche explicite et l'approche automatique. Nous les présentons dans ce qui suit.

L'approche canonique consiste à définir, lors de l'élaboration du système, un profil utilisateur. L'interaction permettra ensuite de cataloguer les utilisateurs à partir du modèle

canonique prédéfini. Cette approche est confrontée à un problème majeur du aux erreurs d'incertitudes pour cataloguer un utilisateur.

Dans l'approche explicite, le système dispose d'un panel de modèles canoniques d'utilisateurs. A la différence de l'approche canonique, ces modèles sont toutefois flexibles dans la mesure où l'utilisateur contrôle certaines parties lors de son interaction avec le système. Le principal inconvénient de cette approche est que cette tâche supplémentaire induit une surcharge cognitive chez l'utilisateur.

Quant à l'approche automatique, elle est la plus utilisée. Le profil utilisateur y est inféré automatiquement à partir d'informations collectées lors de sessions d'interaction avec le système. De ce fait, c'est le comportement de l'utilisateur lui-même qui permet implicitement de décrire son modèle. Pour mettre en œuvre cette dernière approche, il existe deux types de techniques qui sont les techniques collaboratives [102] et les techniques statistiques [103] : De complexité variable, implicites ou explicites, s'adaptant au cours d'une session d'interaction ou sur un ensemble de sessions, les approches de construction du profil restent intimement liées au domaine d'utilisation et à la nature de l'application.

9. QUELQUES EXEMPLES DE PROFILS

Le contenu du profil d'un utilisateur varie selon les approches et les applications. Dans le domaine IHM, le profil contient, par exemple, des informations permettant au système d'adapter l'affichage des résultats selon les préférences de l'utilisateur. Un exemple simple d'un tel profil est celui utilisé par les fournisseurs de services Web. Dans ces systèmes le profil d'un utilisateur représente un ensemble de données personnelles (nom, prénom, langue, genre, date naissance, code postal, email, profession, poste etc. dans le cas de Yahoo par exemple) et des catégories d'intérêts qui constituent sa page d'accueil (ex. météo, football, jeux, etc.). Le contenu de certaines catégories du centre d'intérêt peut être déduit à partir des données personnelles (par exemple si l'utilisateur est né au début du mois de Juin et s'il a choisi la catégorie 'Horoscope', le zodiaque affiché sera Gemmaux). Une autre approche, présentée dans [104], consiste à regrouper les transactions similaires d'un utilisateur dans des clusters. Le profil (noté PRc) d'un utilisateur est obtenu sur la base d'un cluster c et contient les identifiants d'un ensemble de pages dont le poids associé à chaque page dépasse un seuil μ : $PRc = \{(p, poids(p, PRc)), poids(p, PRc) > \mu\}$. Le poids d'une page p est calculé en fonction de la fréquence de son apparition dans le cluster.

Dans le domaine de la RI, le profil de l'utilisateur décrit le plus souvent son centre d'intérêt et, de ce fait, est souvent confondu avec la requête de l'utilisateur. Ce profil est généralement défini à l'aide d'un vecteur de mots clés avec éventuellement un poids associé à chaque mot [105], [106]]. Par exemple le profil d'un utilisateur intéressé par la personnalisation des données peut être présenté par le vecteur à trois termes suivant ; le poids de chaque terme correspond généralement à sa fréquence d'apparition dans les documents :

Exemple 1: profil vectoriel

$\{(personnalisation, 0.7), (profil, 0.9), (modèle, 0.5)\}.$

Figure II.3 : Exemple 1 profil vectoriel

Dans [107], le profil est représenté sous forme d'un graphe orienté (figure 1) où les sommets sont les mots et les arcs expriment les poids entre ces mots. Chaque poids correspond à une probabilité qu'un mot apparaisse après un autre dans le texte. Sur l'exemple de la figure 1, l'arc entre les nœuds « langage » et « naturel » signifie que dans 80% des cas le mot naturel est précédé par le mot langage.

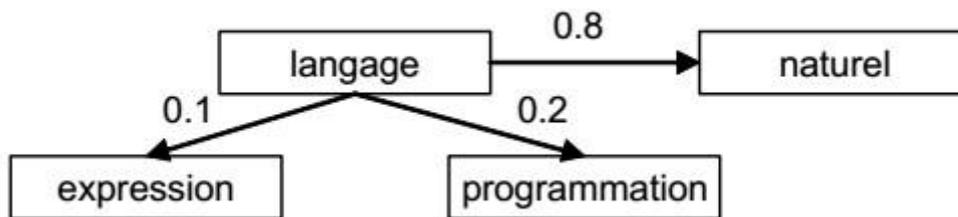


Figure II.4 : Exemple 2 graphe de dépendance de termes

Dans le projet CASPER [108] qui présente un moteur de recherche d'emploi, le profil d'un utilisateur est défini sous la forme de statistiques des actions effectuées par l'utilisateur sur les offres d'emplois. Dans ce cas, l'intérêt de l'utilisateur pour une annonce est déterminé en fonction du temps qu'il a passé à la lire et du type d'action qu'il a effectuée dessus. Prenons par exemple un utilisateur qui a lu une annonce de travail, a postulé pour une autre et a envoyé une troisième à un ami :

Tableau II.1 : Exemple 3 de profil

Profil de Paul

Annonce	Action	Nombre de Clicks	Temps de lecture
job5	lire	1	234
job56	candidater	2	186
job45	envoyer à un ami	1	54

En utilisant un ensemble de règles de décision, le système va décider quelle annonce est pertinente pour l'utilisateur. Par exemple il peut considérer que les annonces 'job56' et 'job45' sont pertinentes pour l'utilisateur en raison des actions effectuées (candidater et envoyer à un ami). Par contre il n'a fait que lire l'annonce 'job5' qui sera considérée comme inintéressante.

Dans le domaine des BD, le profil de l'utilisateur contient des données qui expriment ses habitudes, des prédicats fréquemment utilisés dans ses requêtes ou des définitions de l'ordre de préférences de ces prédicats [109]. L'intérêt de l'utilisateur pour chacun de ces éléments est exprimé par un *degré* qui est un nombre réel compris entre 0 et 1. Prenons par exemple une base de données dont le schéma est le suivant :

```

TRANSPORT (idT, moyen)
HOTEL (idH, nombre_étoiles, région)
VOYAGE (idV, prix, lieu_départ, lieu_arrivée, nombre_jours, idH, idT)
DEPART (idD, idV, date, heure)

```

Figure II.5 : Exemple 4d'une BD

CONCLUSION :

La modélisation de l'utilisateur a pour but la construction des profils utilisateurs qui contiennent des acquisitions pouvant être utile pour le comportement du système, le profil peut être relié à une seule ou groupe de personnes ayant des points communs. Le profil peut englober toute les variations qui caractérisent un groupe d'utilisateur ces profils sont utilisés dans des différents logiciels ils sont présentés par 3 manière : ensembliste, réseaux sémantique, conceptuelle. La construction du profil traduit un processus instancie sa représentation par plusieurs sources d'information ce processus est implicite, la modélisation du profil utilisateur nécessite la fiabilité et la qualité des profils car une des principales raisons du manque de performance est l'application d'un profil hors contexte et les informations ne sont pas appropriées à toutes les situations de recherche, le domaine de l'application conditionne les techniques de personnalisation employées par le système, la construction d'un profil selon plusieurs paramètres entrent en jeu l'identification de 3 approches : canonique explicite et automatique.

Chapitre III :

Contribution

Chapitre III :

INTRODUCTION

1. MODELE D'UN SYSTEME SENSIBLE AU CONTEXTE

2. MODELE DE L'ENVIRONNEMENT

3. MODELE D'ADAPTATEUR AU CONTEXTE

4. MANIPULATION DES ACTIONS

5. METHODOLOGIE D'ADAPTATION AU CONTEXTE

6. GENERATEUR DE PROFIL EN CONTEXTE AWARE

6.1. GENERATEUR DU PROFIL EN CONTEXTE AWARE

7. TRAITEMENT DE L'INFORMATION CONTEXTUEL

7.1. ACQUISITION ET STOCKAGE DES INFORMATIONS DU CONTEXTE

7.2. CONTROL DU DEGRE D'ABSTRACTION DU CONTEXTE

7.3. UTILISATION DE L'INFORMATION DU CONTEXTE POUR DES SERVICES OU DES APPLICATIONS

8. NOTRE PROPOSITION

9. CAS D'ETUDE

10. L'IMPLEMENTATION

10.1. LES SERVICES WEB

10.1.1 ARCHITECTURE DU SERVICE WEB

10.1.2 LANGAGES ET PROTOCOLES

10.2. VISUAL STUDIO

10.3. SYSTEME D'ADAPTATION

10.3.1. L'AFFICHAGE DE PROFILE UTILISATEUR

10.3.2. ACQUISITION ET STOCKAGE DES INFORMATIONS DU CONTEXTE

10.3.3. CONTROL DU DEGRE D'ABSTRACTION DU CONTEXTE

10.3.4. UTILISATION DE L'INFORMATION DU CONTEXTE POUR L'ADAPTATION DE PROFILE

CONCLUSION

INTRODUCTION :

Comme à maintes reprises, en planant au tour de chaque adaptation, on peut constater qu'il s'agit, à chaque fois, du fruit d'une contribution.

Même s'il y a des contextes paraissant similaires, il existe toujours des différences au niveau des éléments enchainés. Pour que la randomisation ne pèse pas sur la limitation de l'adaptation, il est inévitable de s'approfondir dans les détails qui font régulièrement ces différences et pour faire, il va falloir s'appuier sur des éléments de base ; le système, le contexte, l'environnement, le générateur de profile et même les petits détails au tour de la manipulation des actions, les traitements ou le tri d'informations voir aussi la méthodologie d'adaptation ce qui va être détaillé sur ce qui suit. Il y aura également des propositions de solutions ou des amplifications de conceptions déjà proposées

1. MODELE D'UN SYSTEME SENSIBLE AU CONTEXTE:

Un système sensible au contexte est présenté par les éléments suivants : un environnement de travail, des éléments de contexte, un ensemble des actions et un adaptateur au contexte⁴

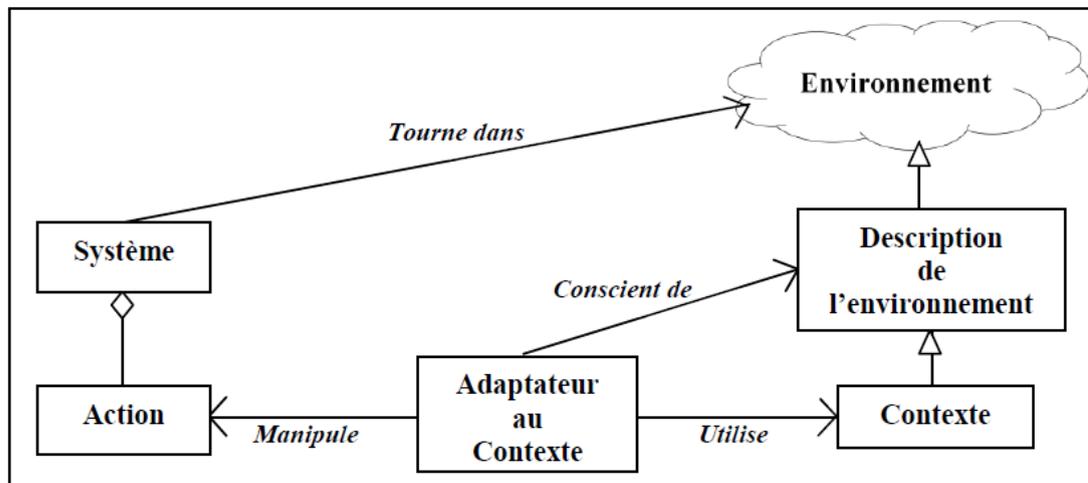


Figure III.1 Les composants d'un système sensible au contexte

Une action est une abstraction de différents types des réactions de système. Elle peut être :

- un service dans le domaine des services web ,

- une simple action d'une application telle que dés activer le son lors d'un appel téléphonique

-une information à fournir à un utilisateur ou à une autre action

Le contexte, selon la figure III.1, représente l'ensemble des informations décrivant l'environnement de travail à un moment donné. L'environnement de travail et l'adaptateur au contexte sont décrits dans les sections suivantes

2. MODELE DE L'ENVIRONNEMENT:

L'ensemble des entités telles que l'infrastructure, les services et les ressources constituant cet environnement. A l'environnement de travail, on associe aussi l'utilisateur, qui est représenté par des caractéristiques telles que son identité et ses préférences. La définition de l'environnement est donnée dans, DeLoach[111], par un environnement fournit les conditions sous lesquelles une entité (un objet, un agent) existe. Un environnement de travail est constitué d'une part d'un système social qui fournit les principes et les structures permettant de régulariser les interactions du système. D'autre part, un environnement de travail est composé d'un environnement physique fournissant les caractéristiques qui contraignent ces interactions⁴

Par conséquent, on peut en déduire que l'environnement est une partie active d'un système sensible au contexte et qui a la responsabilité de définir les propriétés et les conditions dans lesquelles un système peut fonctionner. Selon Wynes [112], l'environnement définit plusieurs responsabilités:

- a-** L'environnement impose une structure : l'environnement est un espace partagé qui structure des interactions entre des "agents", des ressources et des services. La structuration imposée par un environnement peut être distinguée selon :
 - Structure physique : réfère à la structure spatiale et topologique⁵
 - Structure de la communication : définit l'infrastructure d'échange de messages ⁶
 - Structure sociale : réfère à la structure organisationnelle de l'environnement en termes de rôles, groupes et sociétés, etc⁷
- b-** L'environnement impose des règles : les interactions dans un environnement donné sont soumises à des règles par un domaine particulier (par exemple, la mobilité dans un réseau), ou

bien respecter les lois imposées par le constructeur (par exemple, limitation d'accès aux noeuds voisins dans un réseau pour une raison de performance).

Ces règles peuvent aussi être décrites par les habitudes et la culture associées à des endroits particuliers tels qu'un hôpital ou un théâtre Agre[113]

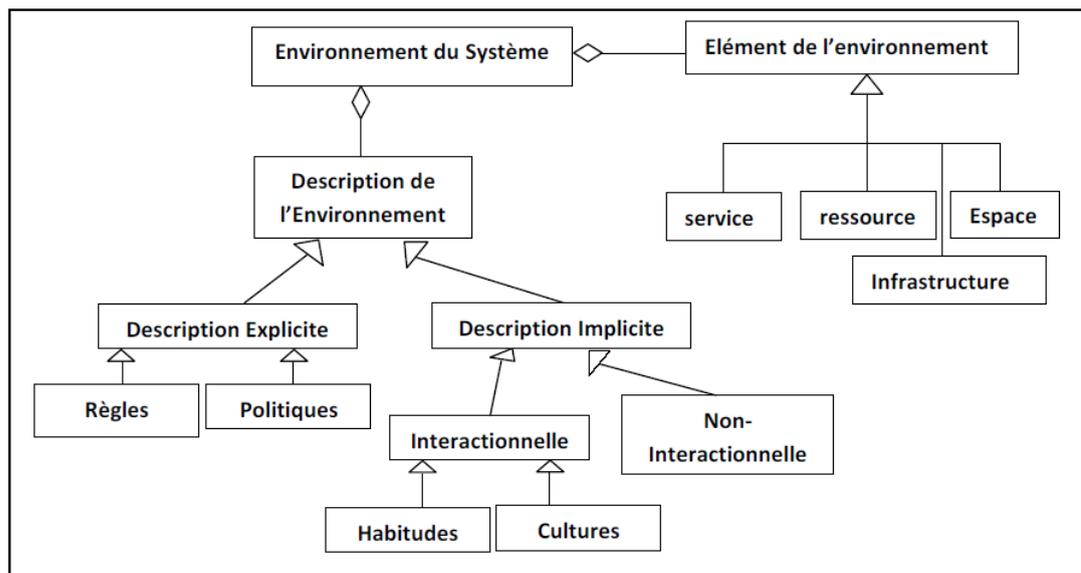


Figure III.2 : Modèle de l'environnement de travail Agre[113]

La description de l'environnement selon une description explicite et une description implicite:

- La description explicite définit les règles et les politiques régularisant le fonctionnement du système indépendamment de l'information de contexte
- La description implicite définit des éléments qui contraignent la description explicite. Elle est distinguée selon une description non-interactionnelle et interactionnelle :
 - La description non-interactionnelle contient la description des paramètres des infrastructures de réseaux tels que le débit et les paramètres des dispositifs utilisés dans cet environnement tels que la largeur de l'écran, la résolution, la puissance de CPU, etc
 - La description interactionnelle définit les habitudes, les pratiques et la culture d'un groupe de personnes dans un domaine particulier

Dans ce qui suit, nous expliquons comment utiliser la description de l'environnement de travail pour définir l'adaptation au contexte.

3. MODELE D'ADAPTATEUR AU CONTEXTE:

Un adaptateur au contexte est une entité d'un système sensible au contexte qui a pour rôle de GHAYAM [114].

- Etre conscient de l'environnement de travail.
- Utiliser le contexte.
- Manipuler les actions du système .

Premièrement, une conscience de l'environnement regroupe la conscience sur les connaissances qui régularisent l'environnement de travail. Ces connaissances apportent des exigences sur les interactions du système dans cet environnement, Agre[113], ou bien imposent des contraintes physiques sur le système, Jarir [115].

La conscience de l'environnement de travail englobe la description explicite et implicite de cet environnement. Cette conscience constitue le modèle interne de l'environnement de travail qu'un adaptateur au contexte peut détenir. Le modèle interne contient particulièrement des connaissances sûres:

Des régularités : l'ensemble des conditions, des normes, des valeurs et des pratiques d'un environnement de travail constituent des régularités de cet environnement. Ces régularités y limitent les interactions d'une application, GHAYAM [114]. Par exemple, dans un environnement de travail, tel qu'un hôpital. Une régularité définie par cet environnement est qu'un médecin dans une salle de chirurgie ne peut pas utiliser l'outil de message instantané pour communiquer avec les autres, vu sa situation qui limite ses mouvements et qui a besoin de plus de concentration sur son travail en cours ٢

Des références : la description des paramètres physiques des ressources composant un environnement de travail constitue les paramètres de références. Le niveau de batterie d'un terminal et le type d'images supporté peuvent être des exemples des références physiques. A noter que les paramètres de références d'un environnement de travail ne sont pas nécessairement fixes, ils peuvent changer continuellement de valeurs ٣

Des préférences : les préférences de l'utilisateur regroupent un ensemble des options imposées par l'utilisateur lui-même, telles que la langue, etc

4. MANIPULATION DES ACTIONS

Dans le travail présenté dans, Urbietta [116], les actions du système sont définies par des préconditions et un effet. Cette définition peut facilement automatiser l'intégration des actions dans un environnement de travail. Une précondition représente les conditions qui doivent être valides pour l'exécution de l'action. Ces conditions sont exprimées en fonction des paramètres de l'environnement. Un effet d'une action peut concerner des données de sortie telle qu'une image ou bien un changement d'état d'une entité de l'environnement⁴

La manipulation des actions désigne un ensemble des opérations qui peuvent être regroupées GHAYAM [114].

Sélection : l'adaptation d'un système peut être effectuée par la sélection d'une action parmi plusieurs autres formes d'une même action, Chen [116]. La sélection concerne la sélection de l'action qui a un effet qui répond le mieux à une exigence définie par une description donnée de l'environnement. Une description est définie par les paramètres de contexte⁴

Composition : l'adaptation peut être effectuée par la composition de plusieurs actions simples pour construire une action composée El-Khatib [117].

Redéfinition : la redéfinition concerne le changement de définition des préconditions d'une action⁴

Transformation : La transformation concerne la modification des types et des valeurs des informations délivrées à l'utilisateur⁴

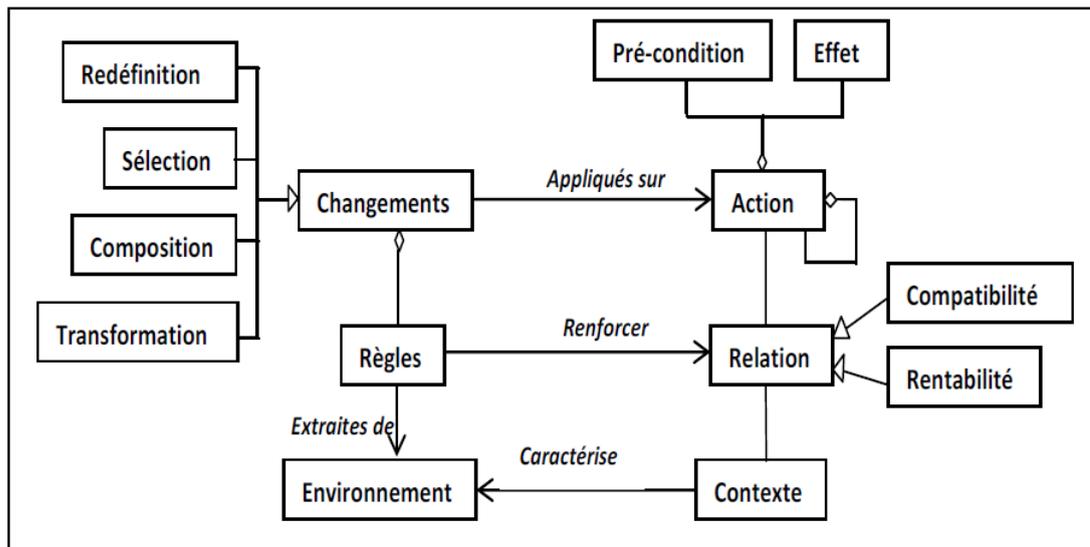


Figure III.3 : Modèle de l'adaptateur au contexte El-Khatib [117]

La figure III.3 montre qu'un adaptateur au contexte est un ensemble de changements appliqués sur les actions du système. Ces changements constituent un ensemble de règles extraites de l'environnement de travail. Les règles sont désignées pour renforcer la relation entre les actions du système et le contexte : assurer la compatibilité et la rentabilité. Les actions sont manipulées à partir de ses composants : les préconditions et l'effet GHAYAM [114].

5. METHODOLOGIE D'ADAPTATION AU CONTEXTE:

Il y a un grand accord qu'un environnement de travail, dans lequel un système sensible au contexte se situe, joue un rôle important dans l'analyse, la conception et la mise en opération de ce système. Cependant, ils ne sont pas nombreux les méthodologies d'adaptation au contexte qui considèrent l'environnement de travail comme point de départ pour l'adaptation du système⁴

Une première étape d'adaptation au contexte est de définir les éléments instituant cet environnement. Ces éléments, dites points d'adaptation, définissent les connaissances régularisant un environnement de travail (les régularités) et les points seuils de cet environnement (les paramètres de références et les préférences de l'utilisateur). La méthodologie de la réalisation d'un système sensible au contexte passe par les étapes suivantes GHAYAM [114] :

- ✓ Identifier toutes les actions du système. Ces actions représentent la fonctionnalité minimale du système. Elles sont supposées disponibles et indépendantes du contexte de l'environnement de travail^۴
- ✓ Définir les points d'adaptations de l'environnement de travail : les régularités, les préférences et les paramètres de références .
- ✓ A partir de ces points d'adaptation, définir les éléments pertinents de contexte .
- ✓ Trouver pour chaque ensemble de contexte, une configuration de contexte donnée, l'ensemble des actions qui répondent aux objectifs d'adaptation^۵
- ✓ Construire les règles d'adaptation. A base des configurations extraites et des actions déterminées, construire les règles d'adaptations^۶
- ✓ Manipuler les actions selon ces règles d'adaptation. La manipulation se fait au niveau de la construction ou bien au niveau de l'exécution^۷

Cette méthodologie réalise l'adaptation d'un système dans un environnement de travail. Ceci est fait en définissant les actions du système selon les caractéristiques et les particularités de l'environnement de travail GHAYAM [114].

6. GENERATEUR DE PROFIL EN CONTEXTE AWARE

Le profil de l'utilisateur est stocké aux prestataires de service. Car un utilisateur utilise divers types des services, son profil est stocké à différents endroits. Chaque fournisseur de services utilise leur propre architecture de personnalisation. Les qualifications dans le profil d'utilisateur continuent au changement fréquemment comme temps, endroit, et secteurs d'intérêt des changements d'utilisateur. Ainsi, L'utilisateur doit se souvenir de toutes les informations d'identification différentes qu'il a soumises à différents endroits. L'utilisateur doit mettre à jour son information avec tous les prestataires de service qu'il agit l'un sur l'autre avec. De cette façon, le stockage de la base de données de profils sur les fournisseurs de services soulève les problèmes concernant la vie privée et la cohérence de l'information. Cela peut être simplifié si le profil est sous contrôle de l'utilisateur Ketaki [119].

6.1. GENERATEUR DU PROFIL EN CONTEXTE AWARE:

Avec l'utilisation croissante des applications Web présentes dans les Smartphones, il est devenu un travail fastidieux pour les utilisateurs, Fournir le profil détaillé au prestataire de services. Dans le scénario actuel, le profil de l'utilisateur est maintenu par le service fournisseurs. Lorsque le site Web particulier requiert l'information de l'utilisateur, un profil complet est fourni par l'utilisateur. Mais ce n'est pas pertinent parce que toutes les informations ne sont pas nécessaires. En conséquence, Seuls les informations de l'utilisation devraient être fournies. Ketaki [119], Il a proposé un système basé sur Android et qui génère Profil spécifique pour un site spécifique. Comme si le site internet concernait les achats, le profil ne contiendrait que le nom, l'adresse postale et le contact ou la réservation en ligne nécessitera un nom, âge etc. Ainsi, selon les exigences, différents profils d'utilisateurs sont générés. Ce sera la vue abstraite du profil réel de l'utilisateur. Et ces vues abstraites sont fournies au fournisseur de services⁴

L'adaptation de cette information sera mise à la disposition de ce site de manière proactive. Le travail de Ketaki [119] comprend:

- ✓ pour faciliter la génération du profil utilisateur, il est nécessaire de concevoir un cadre qui comprend le stockage des informations d'identification utilisateurs, l'acceptation des requêtes, la récupération des données nécessaires à l'application avec ses besoins⁴
- ✓ Les conceptions structurelles qui donneront une idée claire et distincte de la façon dont chaque couche est interdépendante et contribuent également à tous⁴
- ✓ Les conditions requises pour Contexteaware dans les systèmes ubiquitaires⁴

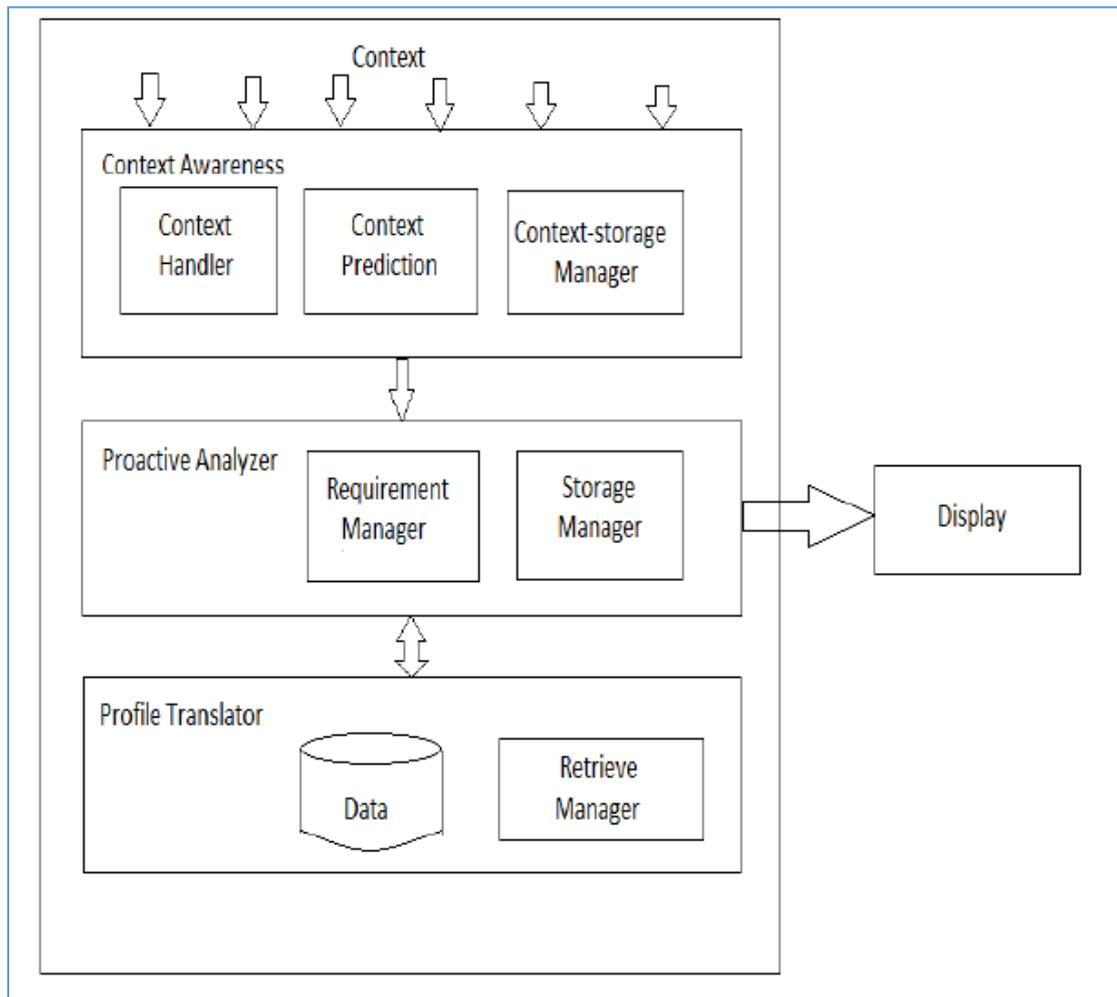


Figure III.4 : Architecture proposée par Ketaki [119]

La figure III.4 représente l'architecture en couches pour le système proposé qui intègre des fonctionnalités interdépendantes. La description détaillée de chaque couche est la suivante:

- a- **Couche de Contexte Awareness :** Le contexte est pris comme entrée principale du système. La tâche de base du gestionnaire de contexte est de reconnaître le contexte actuel dans lequel l'utilisateur fonctionne. La reconnaissance du contexte peut être effectuée en vérifiant le contexte stocké. Parfois, il se peut que l'utilisateur utilise cette application la première fois ou navigue sur le nouveau site Web. Maintenant, si l'utilisateur est dans un nouveau contexte, toutes les données sont transmises au Prédicteur de contexte. Le prédicteur de contexte vérifie si ce courant Le contexte nécessite le profil à générer. Si nécessaire, ce contexte est transmis à Contexte Storage Manager. Contexte Storage Manager stocke un nouveau contexte ou met à jour les données précédentes en fonction de la situation. De cette façon, d'abord Couche reconnaît le contexte qui n'est rien d'autre que la conscience de la situation et suggère si le contexte actuel est Approprié pour la traduction de profil.

- b- Couche d'analyse proactive :** Il faut que l'information contextuelle fournie par la couche supérieure. Le gestionnaire d'exigences dans cette couche rassemblera toutes les informations requises du profil. Cela se fait sans la demande de l'utilisateur qui n'est rien d'autre que d'extraire les exigences du contexte de manière proactive. Ici, les meilleurs résultats peuvent être obtenus en examinant correctement le contexte actuel et en recueillant les résultats en conséquence. Après cette accumulation, les données sont transmises au prochain calque.
- c- Générateur de profile :** La couche prend les exigences du deuxième niveau via le Retriever Manager. Il vérifiera si ces conditions d'utilisation sont remplies par la base de données. Si les exigences des utilisateurs sont satisfaites, cela crée la vue abstraite du profil. Le calque passera cette vue abstraite au Gestionnaire d'affichage de Couche d'analyse proactive. Il affichera la vue de manière proactive.

7. TRAITEMENT DE L'INFORMATION CONTEXTUEL

Dans la littérature il y'a plusieurs architectures de systèmes sensibles au contexte formées de composants responsables de la représentation, de la gestion, du raisonnement et de l'analyse de l'information du contexte⁴

Malgré la multitude de ces architectures, toutes utilisent le même processus de traitement de l'information illustré comme la suite Djamel [120].

7.1. ACQUISITION ET STOCKAGE DES INFORMATIONS DU CONTEXTE

Les types d'informations à acquérir du contexte sont diverses et peuvent être acquises de diverses manières telles que les capteurs physique eg(Contexte/Capteur: Lumiere/photodiode,Audio/Microphone,Mouvement/Accéléromètres,Position/GPS)Baldauf [121].

7.2. CONTROL DU DEGRE D'ABSTRACTION DU CONTEXTE

Le système sensible au contexte n'utilise pas directement les informations acquises du monde physique mais par contre il doit les transformer sous forme de données utilisables. La plupart de ces systèmes réalisent l'abstraction du contexte par deux méthodes ; l'interprétation et l'agrégationDjamel [120]⁴

L'interprétation du contexte : c'est une méthode qui interprète les données et les transforme sous forme de données sémantiques de niveau supérieur tel que : une donnée GPS est interprétée comme un nom de rue⁴

L'agrégation des données : c'est le fait de réduire beaucoup de données en quelques mots clés de haut niveau tel que convertir une série de lecture de températures par des mots clés comme « chaud » et « froid ».

7.3. UTILISATION DE L'INFORMATION DU CONTEXTE POUR DES SERVICES OU DES APPLICATIONS:

Les informations acquises du contexte peuvent être utilisées comme condition de déclenchement d'actions quand le contexte satisfait une situation spécifique ou comme information additionnelle pour améliorer la qualité d'un service fourni⁴

8. NOTRE PROPOSITION

Basé sur Les processus de traitement de l'information contextuel et l'architecture de translation de profile utilisateur basé sur le contexte Ketaki [119], Nous proposons un modèle d'adaptation de profile utilisateur basé sur les informations contextuel.

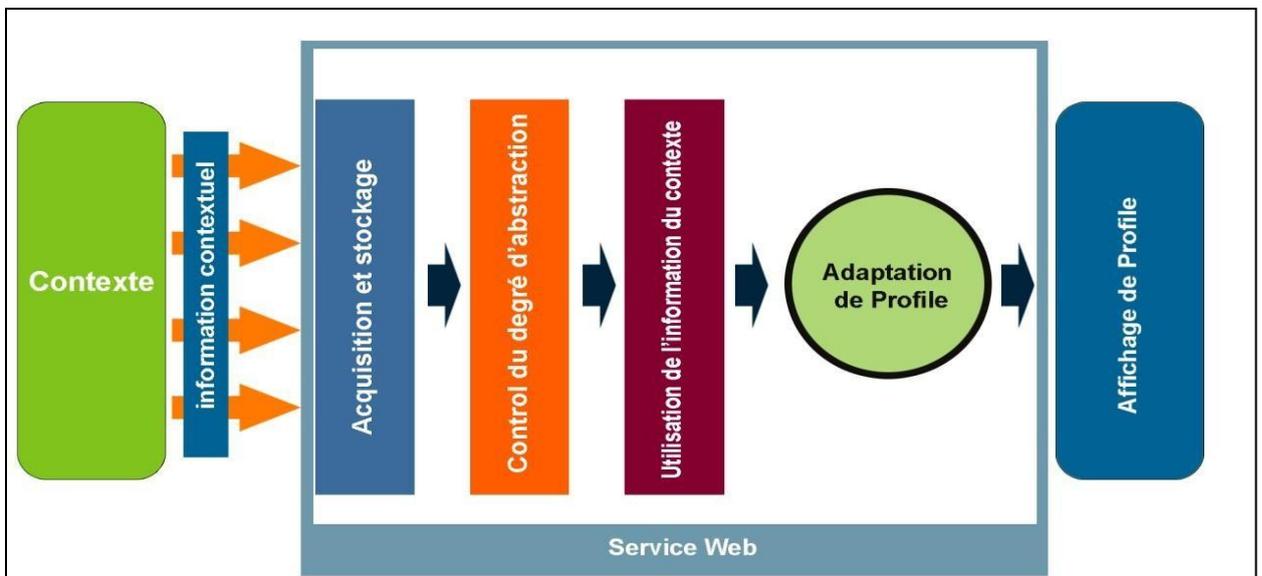


Figure III.5 :Modèle proposée

9. CAS D'ETUDE :

Dans notre travail, nous prenons un cas où l'utilisateur entre dans un espace commercial, habituellement l'utilisateur utilise son carte crédit pour acheter des trucs, mais dans le cas où la carte crédit d'utilisateur est intégré dans son profil. Pour des raisons de sécurité, la carte crédit est privée n'est pas une information publique pour tout le monde.

L'information contextuelle qui sera envoyée au service Web est l'emplacement et l'intention de l'utilisateur d'acheter des trucs.



Figure III.6 :Exemple d'un profil utilisateur.

Lorsque l'utilisateur dans l'espace commercial, l'information contextuelle pour son emplacement et la nécessité de son numéro de carte de crédit a été envoyée au service Web .

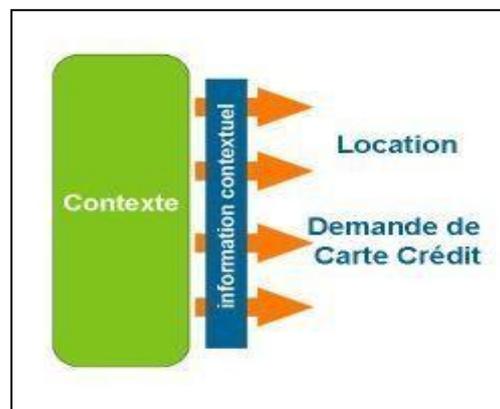


Figure III.7 :collection des informations contextuelles.

Le service web lance l'acquisition et le stockage des informations du contexte et contrôle ces informations pour l'utilisation.

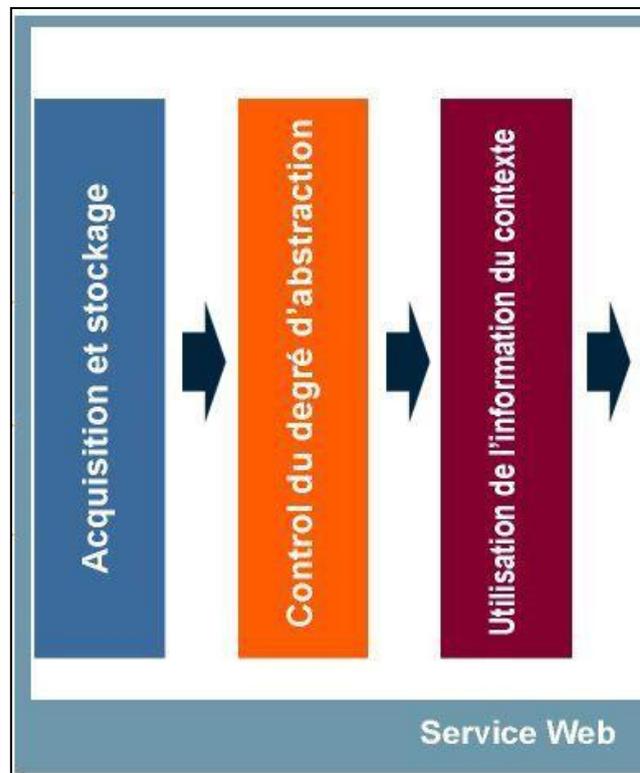


Figure III.8 : Traitement de l'information contextuelle.

Le service web adapte le profil d'utilisateur selon les informations contextuelles traitées.

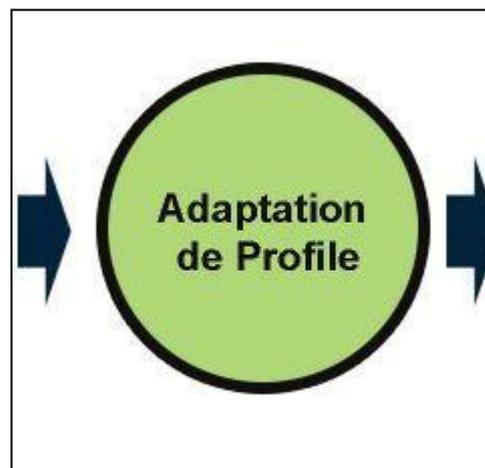


Figure III.9 : Adaptation de le profil.

L'adaptation au niveau de la localisation d'utilisateur et le statut de la carte crédit : Le statut de carte de crédit devient public.



Figure III.9 :Profile adapté.

10. L'IMPLEMENTATION

10.1.LES SERVICES WEB

Les services Web sont « des applications qui relient des programmes, des objets, des bases de données ou des processus d'affaires à l'aide de XML et des protocoles Internet standards. Les services Web sont des compléments aux programmes et applications existants, développés à l'aide de langages tel que Visual Basic, C, C++, C# (C sharp), Java ou autre, et servent de pont pour que ces programmes communiquent entre eux ».Babin[122]

Plusieurs autres définitions des services Web peuvent être retenues. Nous pouvons citer celle de Abouzaid qui semble la plus simple : « Un service Web est un système logiciel qui permet de soutenir l'interaction entre les machines sur un réseau. Il dispose d'une interface écrite en format exploitable par les machines, spécifiquement le WSDL (Web Services Description Language) »Abouzaid[123].

De ce qui précède, on peut conclure que le service Web est un service électronique, offert par des technologies universelles développées en fonction des protocoles réseautiques mondiaux (essentiellement d'Internet) qui agit comme infrastructure de communication⁴

10.1.1. ARCHITECTURE DU SERVICE WEB

Les services Web permettent, à travers un espace de communication, d'échanger différentes informations. Dans cette section, on tente de comprendre comment cet échange de données se fait en présentant un modèle d'architecture classique pour les services Web appelée SOA (Architecture orientée service).

Le fonctionnement des services Web se base sur un modèle composé de trois phases fondamentales : la publication, la découverte et l'interaction, Soukkarieh [124].

PUBLICATION :Après avoir structuré sémantiquement son service, le fournisseur passe ensuite à la publication de ce service afin d'être découvert puis utilisé par d'autres utilisateurs ou machines. La publication du service se fait par l'UDDI (Universal Description, Discovery, and Integration), un annuaire de services fondé sur XML. Toutefois, avant qu'il soit publié, le service doit être décrit, cette description se faisant dans le langage WSDL (Web Service Description Language) Soukkarieh [124].

DECOUVERTE :Une fois la publication du service faite, le client peut alors interroger l'annuaire (UDDI) en utilisant des mots clés permettant l'obtention d'un ensemble de descriptions (WSDL) qui contiennent toutes les informations nécessaires à l'invocation du service Abouzaid[123].

INTERACTION :Aussitôt que la description du service publiée dans l'annuaire correspond à la requête lancée par le client, ce dernier aura une interaction directe avec le service par l'intermédiaire de SOAP (Simple Object Access Protocol) qui invoque les descriptions du service Web Soukkarieh [124].

10.1.2.LANGAGES ET PROTOCOLES

Les Services Web se basent actuellement sur trois protocoles : SOAP, WSDL, UDDI, dans ce qui suit, nous allons détailler le fonctionnement de chaque protocole:

- ✓ **SOAP – Simple Object Access Protocol** :Au sein des protocoles liés aux Services Web, SOAP fait figure de pièce centrale. En effet, SOAP définit un protocole simple destiné à l'échange d'informations. Son objectif est de permettre la normalisation des échanges de données au sein d'architectures distribuées orientées objet Rada[125].
- ✓ **WSDL – Web Service Description Language:** Le WSDL est un langage dérivé d'XML permettant de fournir les spécifications nécessaires à l'utilisation d'un Service web en

décrivant les méthodes, les paramètres et les types de retour. Le WSDL est aussi l'équivalent de IDL (Interface DéfinitionLanguage) pour la programmation distribuée (CORBA). Ce langage permet de décrire de façon précise les services Web, en incluant des détails tels que les protocoles, les serveurs, les ports utilisés, les opérations pouvant être effectuées, et les formats des messages d'entrée et de sortie. Il y eut d'autres tentatives de langages pour résoudre le problème de la définition des services Web. Microsoft a d'abord proposé SDL (Service DefinitionLanguage) avec une implémentation fournie dans leur Toolkit SOAP, puis IBM a proposé NASSL (Network Accessible Service SpecificationLanguage), dont une implémentation est fournie dans SOAP4J. Microsoft modifia sa première spécification et proposa SCL (SOAP ContractLanguage), puis les différents acteurs s'accordèrent sur WSDL⁴

- ✓ **UDDI - UNIVERSAL DESCRIPTION DISCOVERY AND INTEGRATION:**UDDI est une spécification définissant la manière de publier et de découvrir les Services Web sur un réseau. Ainsi, lorsque l'on veut mettre à disposition un nouveau service, on crée un fichier appelé Business Registry qui décrit le service en utilisant un langage dérivé d'XML suivant les spécifications UDDI⁴

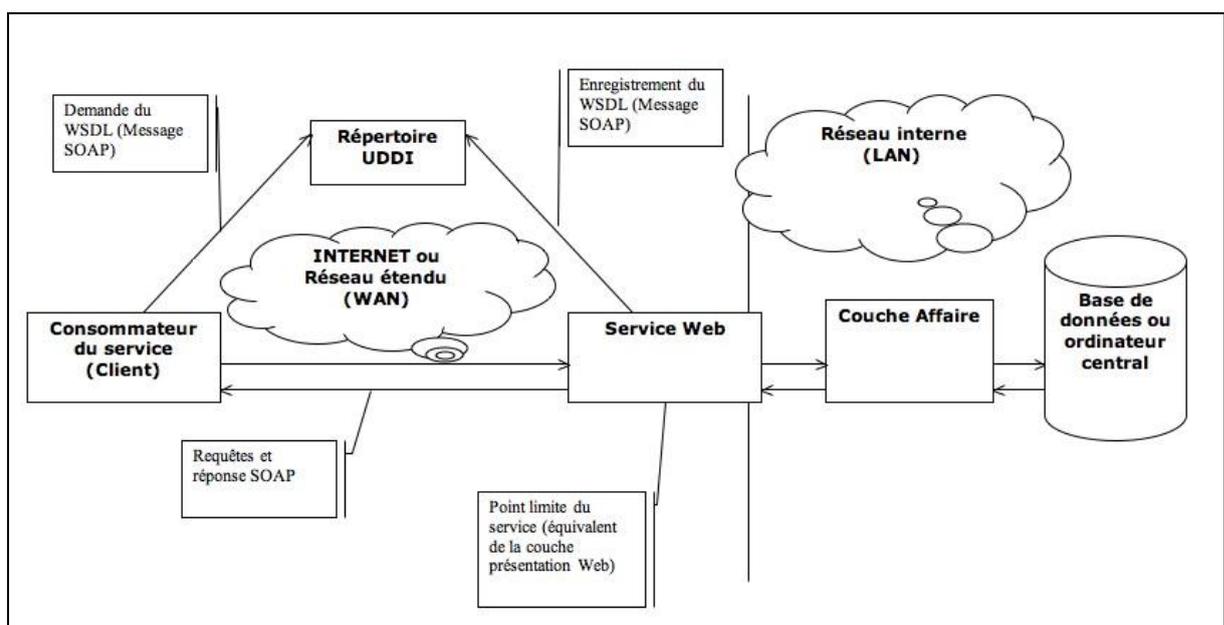


Figure III.10: Architecture globale de Service Web Soukkarieh [124].

10.2. VISUAL STUDIO

Visual Studio .NET est un jeu complet d'outils de développement permettant de générer des applications Web ASP, des services Web XML, des applications bureautiques et des applications mobiles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET et Visual J# .NET utilisent tous le même environnement de développement intégré (IDE,

integrateddevelopmentenvironment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du .NET Framework, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de services Web XML, Microsoft [126]



Figure III.11 : Visual Studio, Microsoft [126].

Cette rubrique comporte des informations sur les éléments suivants, Microsoft [126].

APPLICATIONS SMART DEVICE : L'environnement de développement intégré (IDE) de Visual Studio .NET comprend désormais des outils permettant de développer des applications pour les smart devices, comme le Pocket PC. À l'aide de ces outils et du .NET Compact Framework, un sous-ensemble du .NET Framework, vous pouvez créer, générer, déboguer et déployer des applications exécutées sur le .NET Compact Framework dans des assistants numériques personnels, des téléphones mobiles et d'autres périphériques limités en ressources⁴

CONCEPTEUR ASP.NET MOBILE : Le Concepteur ASP.NET Mobile étend ASP.NET et le .NET Framework afin de vous permettre de générer des applications Web pour les téléphones mobiles, les assistants numériques personnels et les pagers. Ce concepteur est intégré dans l'IDE de Visual Studio. Vous pouvez créer des applications Web mobiles, utiliser le Concepteur Mobile pour modifier un Web Form mobile, puis générer et exécuter l'application, tout cela depuis Visual Studio⁴

WEB FORMS : Les Web Forms sont une technologie ASP.NET que vous utilisez pour créer des pages Web programmables. Les Web Forms se présentent sous forme de langage HTML et de script compatibles avec les navigateurs, ce qui permet à n'importe quel navigateur d'afficher ces pages quelle que soit la plate-forme où il s'exécute. À l'aide de Web Forms, vous créez des pages Web en faisant glisser et en déplaçant des contrôles jusqu'au concepteur,

puis en ajoutant du code, selon une démarche analogue à celle utilisée pour créer des formulaires Visual C#.

SERVICES WEB XML : Les services Web XML sont des applications qui peuvent recevoir des demandes et des données au format XML via le protocole HTTP. Il est possible d'accéder aux services Web XML à partir de n'importe quel langage, modèle de composant ou système d'exploitation dans la mesure où ces services ne sont pas tributaires d'une technologie de composant ou d'une convention d'appel d'objet particulières. Dans Visual Studio .NET, vous pouvez rapidement créer et inclure des services Web XML en utilisant Visual Basic, Visual C#, JScript, les extensions managées pour C++ ou ATL Server.

10.3. SYSTEME D'ADAPTATION

Pour mettre en œuvre notre service web qui stocke l'information du profil utilisateur vers une cadre de implémenté et déployé un service web sur visual studio la publication, la découverte et la liaison du service ainsi que l'invocation et le exécution du service est un plateforme qui propose un architecture standard approprié à la gestion efficace des ces différent objets connecté. Le besoin essentiel à ce niveau est de disposer d'un système de gestion de base de données (SGBD) qui servirait à stocker les données de contexte et profile utilisateur, et à les récupérer à travers un langage de requête comme SQL.

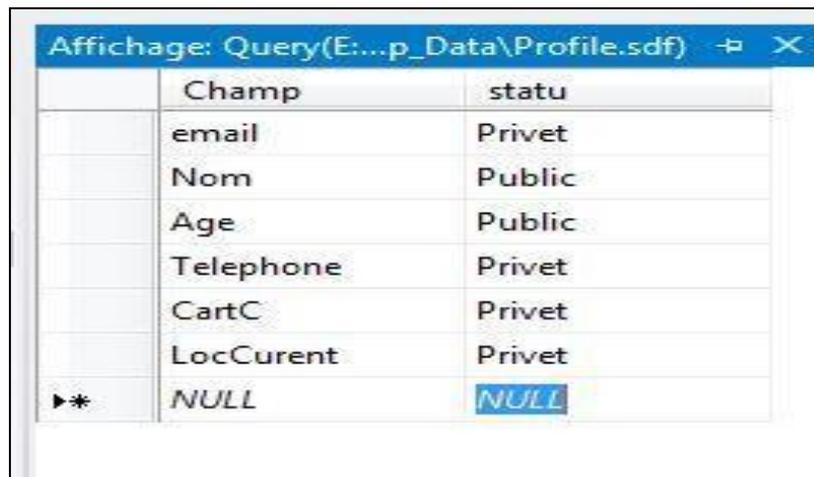
Nous avons créé une table « profile » de base de données contient les données standard d'utilisateur :

Nom ,age , email , telephone ,Num Carte Crédit et Localisation



	Nom	Aqe	email	Telephone	CartC	LocCurent
	SOLTANI Oussa...	28	ousamma@ma...	555556	0014 7772 3596 ...	Carfour
▶*	NULL	NULL	NULL	NULL	NULL	NULL

Figure III.12 Une autre table « affichage » de base de données contient les statuts d'affichage pour les champs de la table « profile »

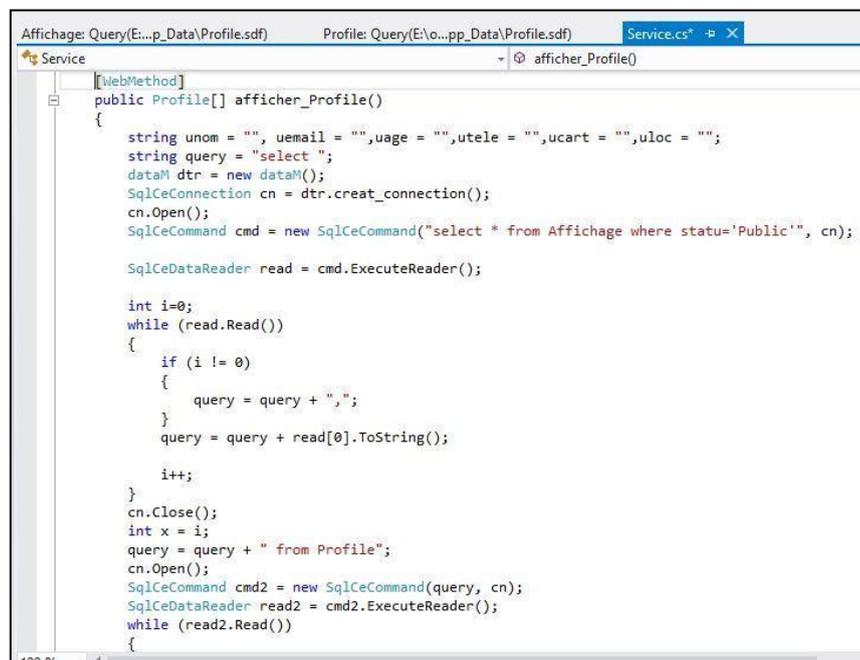


Champ	statu
email	Privet
Nom	Public
Age	Public
Telephone	Privet
CartC	Privet
LocCurent	Privet
▶*	NULL

Figure III.13 les champs de tableaux

- Les champs qui à Private statu n’pas affiché à tout le monde.
- Les champs qui à Private statu sont affichés à tout le monde comme (Carte Crédit et localisation)

10.3.1.L’AFFICHAGE DE PROFILE UTILISATEUR



```

[WebMethod]
public Profile[] afficher_Profile()
{
    string unom = "", uemail = "", uage = "", utele = "", ucart = "", uloc = "";
    string query = "select ";
    dataM dtr = new dataM();
    SqlConnection cn = dtr.creat_connection();
    cn.Open();
    SqlCommand cmd = new SqlCommand("select * from Affichage where statu='Public'", cn);

    SqlDataReader read = cmd.ExecuteReader();

    int i=0;
    while (read.Read())
    {
        if (i != 0)
        {
            query = query + ",";
        }
        query = query + read[0].ToString();

        i++;
    }
    cn.Close();
    int x = i;
    query = query + " from Profile";
    cn.Open();
    SqlCommand cmd2 = new SqlCommand(query, cn);
    SqlDataReader read2 = cmd2.ExecuteReader();
    while (read2.Read())
    {

```

Figure III.14 Méthode afficher_profile()

La méthode « `afficher_profile()` » fournit les informations de profil qui sont publiques.

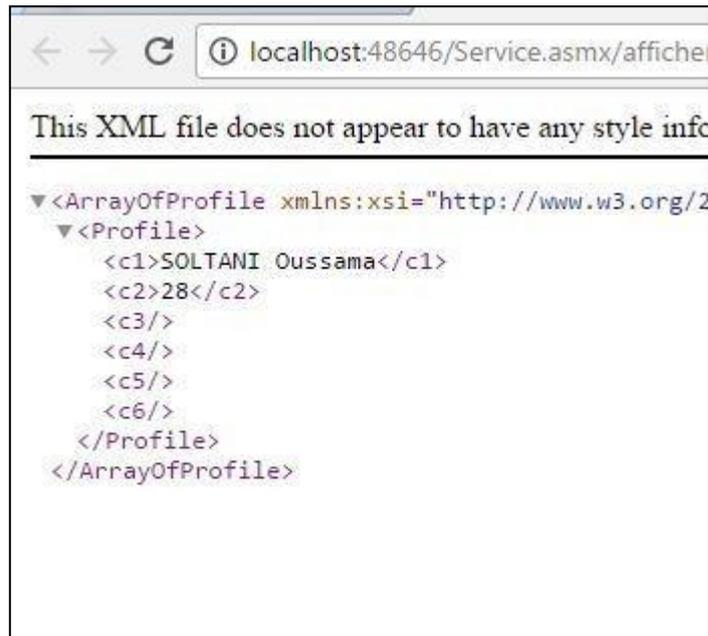


Figure III.15 résultat de la méthode `afficher_profile()`

Comme nous l'observons, la méthode affiche uniquement l'information qui est publique, telle que le nom et l'âge d'utilisateur.

10.3.2.ACQUISITION ET STOCKAGE DES INFORMATIONS DU CONTEXTE:

La méthode « `context_info()` » acquiert les informations contextuelles et les besoins d'utilisateur :

```
[WebMethod]
public void Context_Info(string Localisation, string Besoin)
{
    if (Localisation == "Carfour" && Besoin=="Carte Crédit")
    {
        dataM dm = new dataM();
        dm.update_loc(Localisation);
        dm.update_stat();
    }
}
```

Figure III.16 La méthode « `context_info()` »

Dans notre cas, nous avons traité l'information de l'entrée d'utilisateur à l'espace commercial et le besoin de la carte crédit.

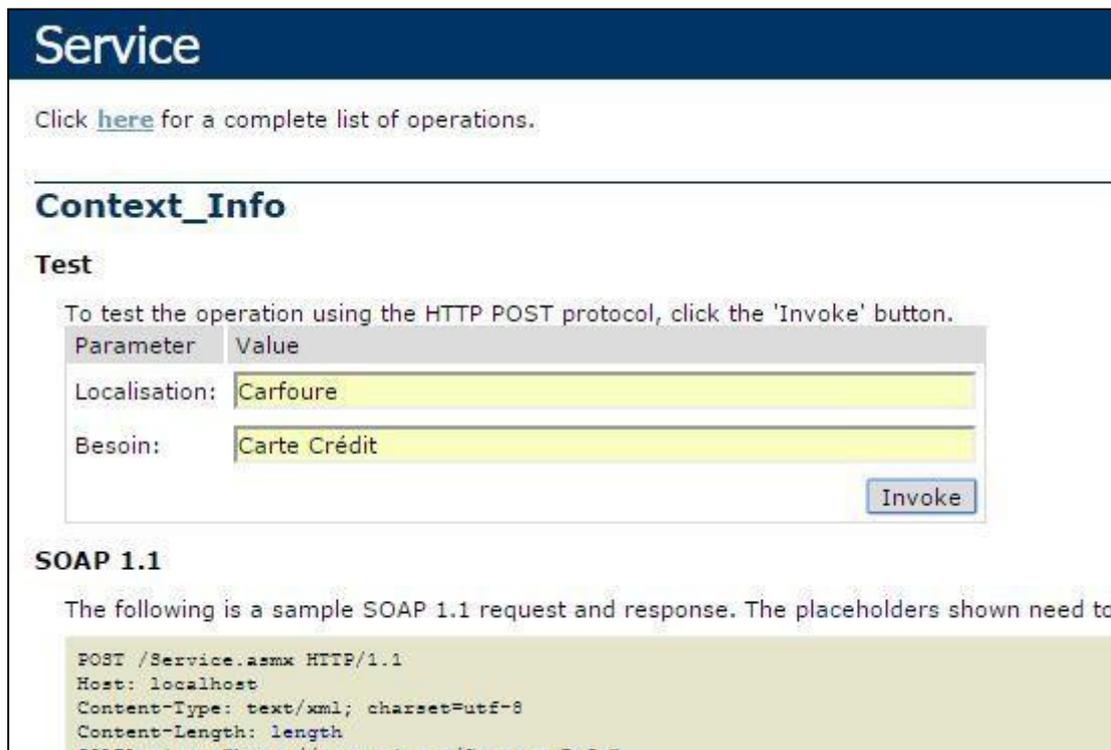


Figure III.17 la méthode « context_info() » sur l'interface publique de service web

10.3.3. CONTROL DU DEGRE D'ABSTRACTION DU CONTEXTE

Le system contrôler les information contextuel avec le premier procédure de la méthode context_info() :

```
if (Localisation == "Carfour" && Besoin=="Carte Crédit")
```

10.3.4.UTILISATION DE L'INFORMATION DU CONTEXTE POUR L'ADAPTATION DE PROFILE

Selon les informations et les besoins contextuel, Le système adapte le profil utilisateur en formation, Dans notre cas, le système a adapté la statue d'affichage pour l'emplacement et le numéro de carte de crédit en tant que public

```

public void update_stat()
{
    dataM dtr = new dataM();
    SqlConnection cn = dtr.creat_connection();
    cn.Open();
    SqlCommand cmd = new SqlCommand("update Affichage set statu='Public' where Champ='LocCurent' or Champ='CartC'", cn);
    cmd.ExecuteNonQuery();
    cn.Close();
}
public void update_loc(string loc)
{
    dataM dtr = new dataM ();
    SqlConnection cn = dtr.creat_connection();
    cn.Open();
    SqlCommand cmd = new SqlCommand("update Profile set LocCurent=@loc", cn);
    cmd.Parameters.AddWithValue("@loc", loc);
    cmd.ExecuteNonQuery();
    cn.Close();
}
}

```

Figure III.18 Mis à jour de statut

Après l'adaptation, le profil d'utilisateur sera affiché comme suit:

This XML file does not appear to have any style information associated with it.

```

<ArrayOfProfile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
  <Profile>
    <c1>SOLTANI Oussama</c1>
    <c2>28</c2>
    <c3>0014 7772 3596 1487</c3>
    <c4>Carfour</c4>
    <c5/>
    <c6/>
  </Profile>
</ArrayOfProfile>

```

Figure III.19 profile utilisateur après le mis a jour

Et les statuts d'affichage dans le profile utilisateur comme :

Champ	statu
email	Privet
Nom	Public
Age	Public
Telephone	Privet
CartC	Public
LocCurent	Public
* NULL	NULL

Figure III.20 Les champs de profile

CONCLUSION

Après avoir fouiné dans tous ces composants, qui ne céderont jamais de s'élargir, et à partir de toute information contextuelle, il est à présent important de prendre en considération la contribution concernant tous ces éléments et changer d'angle de vue envers ces derniers cars, comme mentionné dans NORTE PROPOSITION, les étapes du processus et le traitement des données au peigne fin assurerons la maîtrise méthodologique de chaque adaptation en partant du concept architectural de Ketaki.

Conclusion

générale

Conclusion générale

Les systèmes ubiquitaires représentent une nouvelle tendance adoptée dans différents domaines, L'apparition des Systèmes d'Information ubiquitaires est issue de l'émergence de nouvelles technologies fournissant au système une vision de son environnement, de l'environnement de ses utilisateurs ainsi que de leurs profils. Grâce à ces données formant le contexte d'utilisation.

Mais, le problème qui se pose à ce niveau concerne le degré d'adaptation, de prise de décision à la place de l'utilisateur et de l'identification des données contextuelles nécessaires et suffisantes pour ces services.

Notre objectif dans ce travail est proposé une model d'adaptation de profile d'utilisateur basé sur les informations contextuelles. Ce modèle est basé sur les processus de traitement de l'information contextuel Adopté par tous les chercheurs : (1) Acquisition et stockage des informations du contexte, (2) Control du degré d'abstraction du contexte et (3) Utilisation de l'information du contexte pour des services ou des applications pour Donner une haute qualité d'information pour l'adaptation du profil utilisateur.

Nous avons utilisé les services Web pour éviter l'hétérogénéité des périphériques et des plates-formes.

perspectives

Les perspectives des travaux présentés dans ce mémoire suivent différents axes de réflexion et se situent dans des contextes de recherche différents.

Bibliographie

Bibliographie

- [1] M Weiser, "The Computer for the 21st Century," *Scientific American*, no. 265(3), pp. 66–75, 1991.
- [2] Gandrille Etienne, "Adaptation autonome d'applications pervasives," *UNIVERS*.
- [3] Mark Weiser and John Seely Brown, "The Coming Age of Calm Technology," *Beyond Calculation*, pp. 75–85, 1997.
- [4] M Satyanarayanan, "Pervasive computing : vision and challenges," *Personal Communications*, no. 8(4), pp. 10–17, Aug 2001.
- [5] A Kamoun, "Adaptation d'architectures logicielles de contrôle d'accès dans les environnements collaboratifs ubiquitaires," 2014.
- [6] Cristina Angela and RAMOS CARRILLO, "Agents ubiquitaires pour un accès adapté aux systèmes d'information : Le Framework PUMAS," 2007.
- [7] T Rahwan, I Rahwan, and R Ashri, "Agent-Based Support for Mobile Users Using AgentSpeak (L)," in *Proceedings of the Workshop on Agent-Oriented Information Systems (AOIS 2003)*, vol. 3030, Berlin Heidelberg, 2004, pp. 45-60.
- [8] F Koch and I Rahwan, "Classification of Agent-based Mobile Assistant," in *Proceedings of the Workshop on Agents for Ubiquitous Computing (UbiAgents04)*, New York City, USA , 2004.
- [9] F Gandon and N Sadeh, "Semantic Web Technologies to Reconcile Privacy and Context Awareness," *Journal of Web Semantics*, vol. 1, no. 3, October 2004.
- [10] R.W Collier, G.M.P O'Hare, T Lowen, and C.F.B Rooney, "Beyond Prototyping in the Factory of the Agents," in *Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003)*, vol. 2691, Berlin

- Heidelberg, 2003, pp. 383-393.
- [11] N Hristova and G O'Hare, "Ad-me: wireless advertising adapted to the user location, device and emotions," in *Proceedings of 37th Annual Hawaii International Conference on System Sciences (HICSS37), Minitrack on Mobile Distributed Information Systems (MDIS) (Hawaii, Janvier 5–8, 2004)*, Hawaii, 2004, pp. 1-10.
- [12] G O'Hare and M O'Grady, "Addressing Mobile HCI Need through Agents," in *Proceedings of the 4th International Symposium on Human Computer Interaction with Mobile Devices and Services (MobileHCI'02)*, vol. 2411, Pisa, Italy, 2002, pp. 311-314.
- [13] T Chaari, *Adaptation d'applications pervasives dans des environnements multi-contextes.*: Thèse de doctorat, INSA de Lyon, septembre, 2007.
- [14] B Schilit and M Theimer, "Disseminating Active Map Information to Mobile Hosts," *IEEE Network*, no. 8(5), pp. 22–32, 1994.
- [15] P. J Brown, J. D Bovey, and X Chen, "Context-aware Applications : from the Laboratory to the Marketplace," *IEEE Personal Communications*, no. 4(5), pp. 58–64, October 1997.
- [16] N. S Ryan, J Pascoe, and D. R Morse, "Enhanced Reality Fieldwork : the Context-aware Archaeological Assistant," *British Archaeological Reports*, October 1998.
- [17] J Pascoe, "Adding generic contextual capabilities to wearable computers," *Wearable Computers, 1998. Digest of Papers.*, pp. 92–99, octobre 1998.
- [18] G. D Abowd et al., "Towards a Better Understanding of context and Context-Awareness," in *HUC '99 : Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, London, UK, 1999, pp. 304–307.
- [19] G Chen and D Kotz, "A survey of context-aware mobile computing research," Dept. of Computer Science, Rapport technique 2000.
- [20] T Winograd, *Architectures for context. Human-Computer Interaction.*, 2001.

- [21] A. K Dey and G.D Abowd , "Towards a Better Understanding of Context and Context-Awareness ," in *CHI 2000, Workshop on the What, Who, Where, When, and How of Context-Awareness*, The Hague, The Netherlands, 2000.
- [22] B Schilit, N Adams, and R Want, "Context-aware computing applications," in *In Mobile Computing Systems and Applications, WMCSA 1994. First Workshop on*, décembre 1994, pp. 85–90.
- [23] H Chen, *An Intelligent Broker Architecture for Pervasive Context-Aware Systems. Thèse de doctorat*. Baltimore County: University of Maryland, décembre 2004.
- [24] M Miraoui , *Architecture logiciel pour l'informatique diffuse : Modélisation du contexte et adaptation dynamique des services. Thèse de doctorat*. Montréal, Canada, 2009.
- [25] Yassine EL GHAYAM, *La Sensibilité au Contexte dans un Environnement Mobile*. RABAT: Université Mohammed V Souissi, 2011.
- [26] T Chaari, *Adaptation d'applications pervasives dans des environnements multicontextes, Thèse de doctorat*. Lyon, France: Institut National des Sciences Appliquées, 2007.
- [27] M Kirsch-Pinheiro, M Villanova-Oliver, J Gensel , and H Martin, *Context-Aware Filtering for Collaborative Web Systems: Adapting the Awareness Information to the User's Context*. Mexico, Mexique: ACM Symposium on Applied Computing, 2005.
- [28] M Baladauf, S Dustdar , and F Rosenberg, "A Survey on Context-Aware Systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, June 2007.
- [29] J Indulska and P Sutton, "Location Management in Pervasive Systems," in *Proceedings of the Australasian Information Security Workshop Conference on ACSW frontiers*, vol. 21, 2003, pp. 143–151.
- [30] Anind K. Dey and Gregory D Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *HumanComputer*

- Interaction (HCI) Journal*, vol. 16, pp. 2-4, 2001.
- [31] WAPFORUM. User Agent Profile (UAProf). [Online]. <http://www.wapforum.org>
- [32] W3C. Composite Capabilities / Preferences Profile (CC/PP). [Online]. <http://www.w3.org/Mobile/CCPP>
- [33] Z Quan and al., "ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services," in *International Conference on Mobile Business (ICMB'05)*, Sydney, Australia, 2005, pp. 206 – 212.
- [34] C Raibulet, "Facets of Adaptivity," in *Proceedings of the 2nd European Conference on Software Architecture*, vol. 5292, 2008, pp. 342-34.
- [35] W. K Edwards, "Putting computing in context : An infrastructure to support extensible context-enhanced collaborative applications," *ACM Trans. Comput.-Hum.Interact*, no. 12(4), pp. 446–474, 2005.
- [36] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Hum.-Comput. Interact*, no. 16(2), pp. 97–166, 2001.
- [37] M. K. Smith, C. Welty, and D. L. McGuinness. (2004, février) W3C Recommendation. [Online]. <http://www.w3.org/TR/owl-guide/>
- [38] T Gu, H. K. Pung , and D. Q. Zhang, "A Middleware for Building Context-Aware Mobile Services," in *Proceedings of IEEE Vehicular Technology Conference (VTC)*, 2004.
- [39] M Salehie and L. Tahvildari, "Self-adaptive software : Landscape and research challenges," *ACM Trans. Auton.Adapt. Syst*, no. 4(2), pp. 1–42, 2009.
- [40] P. K. McKinley, S. M. Sadjadi, E. P. Kasten , and B. H. C Cheng, *Composing Adaptive Software.*, 2004.
- [41] R. Laddaga, "Self-adaptive software," Rapport technique 1997.
- [42] P. Oreizy et al., "An architecture-based approach to selfadaptive software," *Intelligent*

- Systems and their Applications, IEEE*, no. 14(3), pp. 54–62, mai 1999.
- [43] A David Garlan, "10-year Perspective on Software Engineering Self-adaptive Systems," in *In Proceedings of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2013)*, Piscataway, NJ, USA, 2013, pp. 2–2.
- [44] U Malinowski, T Kuhme, H. Dietrich, and M Schneider-Hufschmidt, "A taxonomy of adaptive user interfaces," *People and Computers*, pp. 391-414, 1992.
- [45] H Dieterich, U Malinowski, T Kühme, and M Schneider-Hufschmidt, "State of the Art in Adaptive User Interfaces," *Adaptive User Interfaces, Principles and Practice, Included in series Human Factors in Information Technology*, no. 10, pp. 13-48, 1993.
- [46] J-Y Ramel, *Propositions pour la représentation et l'analyse de documents numériques*. France: HDR, Université François Rabelais De Tours, 2006.
- [47] S.A Horgen. (2002) A Domain Model for an Adaptive Hypertext System based on HTML, Master degree in Computer Science. [Online]. <http://www.aitel.hist.no/~svendah/ahs.html>
- [48] G Fischer, *User Modeling in Human Computer Interaction, User modeling and user-adapted interaction.*: Kluwer Academic Publishers, 2000.
- [49] S Weibelzahl, *Evaluation of Adaptive Systems, Graduate Programme Human and Machine Intelligence*. Germany: University of Education Freiburg, 2002.
- [50] A Jameson, *Systems That Adapt to Their Users: An Integrative Perspective.*: Saarland University, 2001.
- [51] A Morch, "Three levels of end-user tailoring: customization, integration, and extension," in *Third Decennial Aarhus Conference*, Aarhus, Denmark, 1995, pp. 41-51.
- [52] A Morch and N. D. Mehandjiev, "Tailoring as Collaboration: The Mediating Role of Multiple Representation and Application Units," in *CSCW'2000*, Philadelphia, Pennsylvania, USA, 2000, pp. 75-100.

- [53] D.L Scapin and J.M.C Bastien, "Ergonomic criteria for evaluating the ergonomic quality of interactive systems," in *Behaviour & Information Technology*., 1997, pp. 220-231.
- [54] G Bourguin, *Un support informatique à l'activité coopérative fondé sur la Théorie de l'Activité : Le projet DARE*, Thèse de doctorat d'informatique. Lille, France: Université de Lille 1 , Laboratoire Trigone, 2000.
- [55] M-L Betbeder-Matibet, *Symba : un environnement malléable support d'activités collectives en contexte d'apprentissage*. Thèse de doctorat d'informatique. Le Mans, France: Université du Maine, 2003.
- [56] D Browne, P Totterdell, and M Norman, Eds., *Adaptive User Interfaces Computer and People Series.*: Academic Press.
- [57] I Cingil , A Dogac , and A Azgin , *A broader approach to personalization.*: Communications of the ACM, 2000.
- [58] M Rosenberg. (2001, may) The personalization story, ITworld.com. [Online]. <http://www.itworld.com/Man/2676/ITW010511rosenberg/>
- [59] M Villanova-Oliver, *Adaptabilité dans les systèmes d'information sur le web : Modélisation et mise en oeuvre de l'accès progressif*. Thèse de doctorat d'informatique. France: Institut National Polytechnique de Grenoble, 2002.
- [60] H Habieb-Mammar, *EDPHA : un Environnement de Développement et de Présentation d'Hyperdocuments Adaptatifs*. Thèse de doctorat d'informatique. France: Institut National des Sciences Appliquées de Lyon, 2004.
- [61] K Samaan, *Prise en compte du modèle d'interaction dans le processus de construction et d'adaptation d'applications interactives*, Thèse de doctorat d'informatique. France: Ecole Centrale de Lyon, 2006.
- [62] B Mobasher, R Cooley, and J Srivastava , *Automatic Personalization Based on Web Usage Mining.*: Communications of the ACM, 2000.
- [63] G Kappel , W Retschitzegger, and W Schwinger , "Modeling Customizable Web

- Applications – A Requirement's Perspective," in *in Proceedings of the International Conference on Digital Libraries, (ICDL 2000)*, Kyoto, Japan, 2000.
- [64] G Rossi , D Schwabe, and R Guimarães , "Designing Personalized Web Applications," in *10th International World Wide Web Conference (WWW10)*, Hong Kong, 2001.
- [65] G. Fischer, *User Modeling in Human Computer Interaction.*: Kluwer Academic Publis, 2000.
- [66] E.A. Edmonds, *Adaptive man-computer interfaces*, Coombs and Alty, Eds.: CompuLing Skills and the User Interface, 1981.
- [67] C Stephanidis et al., *Adaptable and adaptive user interfaces for disabled users in AVANTI Project in Triglia*, Intelligence in Services and Networks: Technology for Ubiquitous Telecom Services ed., S Mullery et al., Eds.: Springer, 1998.
- [68] A Kobsa, J Koenemann, and W Pohl, "Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships," *The Knowledge Engineering Review*, no. 16(2), pp. 111-155, 2001.
- [69] P Oreizy et al., "An Architecture-Based Approach to SelfAdaptive Software," *IEEE Intelligent Systems*, vol. 14, pp. 54-62, 1999.
- [70] I. Gorton, Y. Liu, and N. Trivedi, *An extensible and lightweight architecture for adaptive server applications.*: Softw Pract. Exper, 2008.
- [71] D. Garlan, S. W. Cheng, A.-C. Huang, B. Schmerl, and P. Steenkiste, *Rainbow : architecture-based selfadaptation with reusable infrastructure.*, octobre 2004.
- [72] W. Wahlster and A. Kobsa, *Dialogue-based user models*, In *Proceedings of IEEE*, 747948960th ed., 1986.
- [73] E. Gaussier and MH. Stefanini, *Assistante Intelligente à la recherche d'informations*, 2746207265th ed.: Hermes Science, 2003.
- [74] W.N. Zemirli, " *Modèle d'accès personnalisé à l'information basé sur les diagrammes d'influence intégrant un profil utilisateur évolutif* ", PhD thesis L'universite Paul

Sabatie., Juin 2008.

- [75] D. Kostadinov, *Personnalisation de l'information : une approche de gestion de profils et de reformulation de requêtes*. PhD thesis, L'UNIVERSITE DE VERSAILLESAINT-QUENTIN-EN-YVELINES ed., Décembre 2007.
- [76] A. Jameson, *User Adaptive Systems An integrated Overview*, 7th ed., June 20-24, 1999.
- [77] J. Budzik and K. Hammoud, « *User interactions with every applications as context for just-in-time information access* », *Proceedings of the 5th international conference on intelligent user interfaces*, 4451st ed., Mars 2000.
- [78] S. Dumais, E. Cuttrel, J. Cadiz, G. Jancke, and R. Sarin, « *Stuff I've seen : A system for a personal information retrieval and re-use* », 26th ed., 2003.
- [79] G. Koutrika and Y.E. Ioannidis, *Personalization of Queries in Database Systems Systems*, In , *Proceedings of the 20th International Conference on Data Engineering*, 597597608th ed. Boston,Massachusetts, USA, 2004.
- [80] G. Koutrika and Y.E. Ioannidis, *Personalized Queries under a Generalized Preference Model Model*, In *Proceedings of the 21st International Conference on Data Engineering*, 2005841841852nd ed. Tokyo, Japan, 2005.
- [81] P. Dell'Acqua, P.L. Moniz, and A. Vitoria, *User Preference Information in Query Answering Answering*, In *Proc , Proceedings of the 5th International Conference on Flexible Query eedings Answering Systems*, 163173rd ed. Copenhagen, Denmark, 2003.
- [82] F.A. Asnicar, D. Fant, and C. Tasso, ""User Model Based Information Filtring" in M.Lenzerini ed.AI 97: Advances ArtificialIntelligence, Proc of the 5th Congress," *Springer*, no. LNAI 1321, pp. 242-253, 1997.
- [83] G. Salton, "The SMART Retrieval System—Experiments in Automatic Document Processing," *Prentice-Hall Inc*, no. NJ, 1971.
- [84] R. Armstrong, D. Freitag, D. Joachims, and T. Mitchell, "Webwatcher : A learning apprentice for the world wide web," *Spring symposium*, pp. pages 6–12, 1995.

- [85] L. Chen and K. Sycara, *Webmate : A personal agent for browsing and searching* In *Proceedings of the 2nd international conference on autonomous agents and multi*, 1013th ed. Minenapolis, 1998.
- [86] H. Tebri, M. Boughanem, and C. Chrisment, "Incremental profile learning based on a reinforcement method.," *ACM symposium*, pp. pages 1096–1101, 2005.
- [87] G. Somlo and A. Howe, "Using web helper agent profiles in query generation In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*," *ACM Press*, pp. pages 812–818, 2003.
- [88] M.J. Pazzani and J. Muramatsu, *Identifying interesting web sites*. In *Proceedings of the 30th National Conference on Artificial Intelligence*, 5461st ed. Portland, 1996.
- [89] G. Salton and C. Yang, *On the specification of term values in automatic indexing*. *Journal of documentation*, 351372nd ed., 1973.
- [90] H. Sakagami and T. Kamba, "Learning personal preferences on online newspaper articles from user behaviors," in *Proceedings of the 6th International WWW Conference*, Santa Clara, California, April, 1997, pp. 7-11.
- [91] M. Balabanovic and Y. Fab Shoham, "Content-based collaborative recommendation," *ACM*, no. 3(40), pp. 66–72, March 1997.
- [92] H. Lieberman, "An agent that assists web browsing.," in *In Proceedings of the 14th International Joint Conference On Artificial Intelligence*, Montreal, Canada, August 1995.
- [93] G. Gentili, A. Micarelli, and F. Sciarrone, "Infoweb : An adaptive information filteringsystem for the cultural heritage domain," *Applied Artificial Intelligence*, no. 17(8-9), pp. 715–744, 2003.
- [94] M.N. Huhn and L. Stephens, *Personal ontologies*. *Internet Computing* 3(5)., October 1999.
- [95] J. Chaffee and S. Gauch, "Personal ontologies for web navigation. In *Proceedings of the ninth international conference on Information and knowledge management*, , McLean,"

- ACM Press, pp. 227–234, 2000.
- [96] S. Gauch, J. Chaffe, and A. Pretschner., "Ontology-based personalized search and browsing.," in *Web Intelligence and Agent System*, 2003, pp. 219–234.
- [97] M.S. Bates, *Search techniques. In Annual Review of Information Science and Technology 16. M.E. William*, 1981139169th ed., 1981.
- [98] s. Gauch, M. Speretta, and A. Chandramouli, *User profiles for personalized information access*, 43215489th ed., 2007.
- [99] G. Amato and U. Staraccia, "User profile modelling and applications to digital libraries," in *In Proceedings of the 3rd European Conference on Research and advanced technology for digital libraries*, 1999, pp. 184–187.
- [100] R. Belew, "Adaptive information retrieval : Using a connectionist representation to retrieve and learn about documents," in *In Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, Cambridge, Massachusetts, United States, 1989, pp. 11–20.
- [101] N. Belkin et al., *Iterative exploration, design and evaluation of support of query reformulation in interactive information retrieval*, 373rd ed., 2001.
- [102] P. Melville, R.J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering," New Orleans, LA, In Proceedings of the 2001 Workshop on Recommender Systems September 2001.
- [103] M. Perkowitz and O. Etzioni, *Towards adaptive Web sites: conceptuel framework and case study*, 11812245275th ed., 2000.
- [104] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization," *Data Mining and Knowledge Discovery*, vol. 6(1), pp. 61-82, 2002.
- [105] J. Ferreira and A. Silva, "MySDI:A Generic Architecture to Develop SDI Personalised Services," in *Proceedings of the 3rd International Conference on Enterprise Information Systems*, Setubal, Portugal, July 7-10, 2001.

- [106] A. Pretschner and S. Gauch, "Ontology Based Personalized Search," in *Proceeding of the 11th IEEE Intl. On Tools with Artificial Intelligence*, Chicago, November 1999, pp. 391-398.
- [107] H. Sorensen and M. Mc Elligott, "PSUN : A Profiling System for Usenet News, Conference on Information and Knowledge Management (CIKM'95)," in *Intelligent Information Agents Workshop*, Baltimore, December 1995.
- [108] K. Bradley, R. Rafter, and B. Smyth, "Case-Based User Profiling for Content Personalisation," in *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based*, Trento, Italy, August 2000.
- [109] G.Y. Ioannidis, "Personalization of Queries in Database Systems," in *Proceedings of the*, Boston, Massachusetts, USA, April, 2004.
- [110] H Dieterich, U Malinowski, T Kühme, and M Schneider Hufschmidt, *State of the Art in Adaptive User Interfaces Principles and Practice.*: Included in series Human Factors in Information Technology, 1993.
- [111] DeLoach S.A., and Valenzuela Jorge. L., "An Agent-Environment Interaction Model". In: Padgham, L., and Zambonelli, F. (eds): Agent Oriented Software Engineering VII. Lecture Notes in Computer Science, Vol. 4405, 2007.
- [112] D., Omicini, A. and Odell, J., "Environment as a first class abstraction in multi-agent systems", *Journal of Autonomous Agents and Multi-Agent Systems*, Vol 14, N 1, 2007.
- [113] Agre, P. "Changing Places: Contexts of Awareness". *Computing. Human-Computer Interaction*, Vol 16, N 2, 2001.
- [114] D. Yassine EL GHAYAM La Sensibilité au Contexte dans un Environnement Mobile Université Mohammed V Souissi - RABAT École Nationale Supérieure d'Informatique et d'Analyse des Systèmes
- [115] Jarir Z. and Erradi M., "Personnalisation des applications Web pour une meilleure QoS", Villemur T., Saqui-Sannes P., and Drira K. Eds, NOTERE'06 : Nouvelles technologies de la répartition : 6^{ème} Conférence internationale sur les nouvelles technologies de la répartition, Toulouse, France, 2006.
- [116] Urbietta A., Azketa E., Gomez I., Parra J. and Arana N. "Bridging the Gap between Services and Context in Ubiquitous Computing Environments Using an Effect and Condition Based Model". 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008.

Springer Berlin / Heidelberg, 2008.

- [117] Chen H., Finin, T. and Joshi A., “An Ontology for Context Aware Pervasive Computing Environments”, Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, Vol 18, N 3, pp 197–207 2004.
- [118] El-Khatib K., Bochmann G. V. and Saddik A. E., “A QoS-Based Framework for Distributed Content Adaptation”, First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, QSHINE 2004, pp 308–312, 2004.
- [119] Ketaki Shah , Anuja Raundal, Gouri Bhandari, Santwana Rathi, Poonam Railkar ,Parikshit Mahalle Department of Computer Engineering , Smt. Kashibai Navale College of Engineering , Off Sinhgad Road , Vadgaon (Bk), Pune Ketaki Shah et.al / Indian Journal of Computer Science and Engineering (IJCSE)
- [120] djamel GUESSOUM thèse par articles présentée à l'école de technologie supérieure comme exigence partielle à l'obtention du doctorat en génie 24 novembre 2016.
- [121] M. Baldauf, S. Dustdar, and F. Rosenberg. A Survey On Context-Aware Systems. International Journal of Ad Hoc and Ubiquitous Computing Inderscience Publishers, June 2007.
- [122] G. Babin and M. Leblanc, "Les Web Services et leur impact sur le commerce B2B," CIRANO Centre interuniversitaire de recherche en analyse des organisations 2003.
- [123] F. Abouzaid, "Analyse formelle d'orchestrations de services Web," Doctora Génie informatique et génie logiciel, École Polytechnique de Montréal, 2010.
- [124] B. Soukkarieh, "Technique de l'internet et ses langages: vers un système d'information web restituant des services web sensibles au contexte," Doctora, Informatique Université de Toulouse, Université Toulouse III-Paul Sabatier, 2010
- [125] Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of ametric on semantic nets. In : IEEE Transactions on Systems , Man and Cybernetics(1998) .
- [126] Microsoft www.microsoft.com Visual Studio 2015. Avril 2016