

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



THESE

Présentée pour l'obtention du **grade de DOCTORAT 3^{ème} Cycle**

En : Télécommunication

Spécialité : Système et Réseaux Informatiques des Télécommunication

Par : BELHADI Siham

Sujet

Gestion du contexte pour des applications mobiles dédiées aux services de santé (au domaine médical)

Soutenue publiquement, le 02 / 07 / 2020, devant le jury composé de :

Mr KAMECHE Samir	Professeur	Univ. Tlemcen	Président
Mr MERZOUGUI Rachid	Professeur	Univ. Tlemcen	Directeur de thèse
Mr BENAÏSSA Mohammed	MCA	Centre Univ Ain Temouchent	Examineur
Mr HADJILA Mourad	MCA	Univ. Tlemcen	Examineur

**Gestion du contexte pour des applications mobiles
Dédiées aux services de santé (au domaine médical)**

Auteur : BELHADI Siham
Prof. Responsable : MERZOUGUI Rachid
Sujet proposé au sein du labo *STIC*

À la mémoire de ma Grande mère.

A mes chers parents à qui je souhaite bonheur et santé.

A mon très cher mari.

Aux prunelles de mes yeux mes filles « Allae Ghizlan et Loudjayn Chourouk ».

A ma chère tante « Fatna ».

A mes frères et mes sœurs.

A mon neveu « Ayoub ».

A mes très chères amies « Souad et Asma ».

A chacune des familles BELHADI et EL HELALI.

« Tout est possible à qui rêve, ose, travaille et n'abandonne jamais ».

Xavier Dolan.

« Chaque difficulté rencontrée doit être l'occasion d'un nouveau progrès ».

Pierre de Coubertin.

Remerciement

Nous remercions ALLAH le Tout-puissant de nous avoir donné le courage, la volonté et la patience de mener à terme le présent travail.

Ce thème de recherche a été effectué au Laboratoire de *Systèmes et Technologies de l'information et de la Communication (STIC)* au Département de Télécommunication de la Faculté de Technologie à l'Université Abou-bekr Belkaïd Tlemcen, au sein du groupe, *SID: Systèmes Intelligents et Domotique*, dirigé par Mr. **MERZOUGUI Rachid**, Professeur à l'Université Abou-bekr Belkaïd Tlemcen, je lui exprime particulièrement toutes mes reconnaissances pour m'avoir fait bénéficier de ces compétences scientifiques, ses qualités humaines, sa clairvoyance, son charisme, son dynamisme et sa constante disponibilité.

Je tiens à exprimer mes plus vifs remerciements à Monsieur **KAMECHE Samir**, Professeur à l'université Abou-bekr Belkaïd Tlemcen, qui m'a fait l'honneur de présider ce jury.

Je remercie sincèrement Mr. **BENAISSA Mohammed**, Maître de Conférences au centre universitaire Ain Temouchent et Mr. **HADJILA Mourad**, Maître de Conférences à l'université Abou-bekr Belkaïd de Tlemcen, d'avoir accepté de juger les travaux de cette Thèse.

J'exprime également, mes remerciements au Mr. **GOUALI Samir** pour l'intérêt qu'il a porté à nos travaux ainsi que pour ses conseils scientifiques et ses qualités humaines.

Je remercie en particulier Madame **KADI Ouassila** pour son assistance de compléter ce travail.

Merci également à toutes les personnes dont l'amitié nous a apporté des moments de réconfort et distraction nécessaires lors de déroulement d'un tel projet, notamment, le groupe de recherche du laboratoire STIC, pour sa disponibilité à toutes heures, leur aide et leur soutien.

Je dédie cette thèse à ma famille. J'y puise une ouverture d'esprit, un soutien, une confiance et un amour indéfectibles que j'espère rendre pareillement et transmettre à mon tour. Par ailleurs, puisque l'occasion se présente ici, Je remercie mon très cher mari pour son soutien continu et sa patience durant toutes ses années.

Enfin j'adresse mes remerciements les plus sincères à tous ceux qui ont contribué de près ou de loin à la concrétisation de ce travail.

ملخص

يعرف إنشاء تطبيقات وخدمات الهاتف المحمول تطوراً واسعاً في الآونة الأخيرة، من خلال الاعتماد على مفهوم السياق وحالة المستخدم.

معظم التطبيقات المتاحة تواجه ثغرات تعيق عملها. ويعود ذلك لأنشائها خارج السياق، لهذا تصبح غير مناسبة لشروط التنفيذ. هذه الإشكالية هي التي دفعتنا إلى إنشاء نموذج موحد يركز على إدارة الخدمات في وقت واحد بهدف تشكيل تطبيق قوي.

في بداية عملنا نقوم بدراسة متعمقة للتطبيقات المحمولة وسلوكها، اعتماداً على سياق التنفيذ، مع الأخذ في الاعتبار مختلف التطبيقات المخصصة للصحة، التي تهدف لدعم وللمساعدة المريض في حياته اليومية (المراقبة الطبية، الأمن الدائم، تطبيقات التحليلات والعلاج عن بُعد، إلخ).

أما في خطوة ثانية، نقترح نموذجاً ديناميكياً وحساساً للسياق، يسمح بإدارة السياق والتكيف من أجل موازنة إمكانيات الخدمة التي تم تطويرها مع بيئتها وللاستخدام الأفضل للموارد في بيئات شديدة التغير. نحن نركز على الجانب التفاعلي، مما يسمح لنا بتكييف سلوك التطبيقات مع ظروف استخدامها.

في إطار هذه الأطروحة، نقوم بتنفيذ النموذج الأولي لمنصة CAHS (خدمات الصحة الحساسة لسياق) المقترح ضمن نظام Java IPOJO واستناداً إلى التكيف مع خدمة SOA. هذه المنصة تعمل على توفير حساسية التطبيقات للسياق و ضمان أداء أفضل لجميع التطبيقات المخصصة للصحة. يهدف النهج الذي نتبعه إلى تصميم جيل جديد من المنصات المحمولة المفتوحة والجامعة لمختلف أنواع التطبيقات المحددة لصحة المحمول.

Mot-clé— Contexte ; Système sensible au contexte ; Architecture orientée service ; Santé mobile ; iPOJO.

Résumé

Actuellement, la création et le développement des applications et services mobiles se propagent de plus en plus en ouvrant de nouvelles pistes dépendantes aux particularités de contexte et à la situation de l'utilisateur. La majorité des applications disponibles rencontrent des lacunes qui entravent leur fonctionnement. Etablies en dehors de leur contexte, elles deviennent inadaptées aux conditions d'exécution. C'est cette constatation qui nous a suscité à créer un modèle unificateur qui tend à gérer les services de manière simultanée pour former une application solide.

Les travaux portent, dans un premier temps, sur une étude approfondie des applications mobiles et de leur comportement, en fonction de contexte d'exécution. Nous considérons le cas des applications dédiées à la santé, conçues pour accompagner et assister un patient dans sa vie quotidienne (suivi médicale, sécurité permanente, applications des analyses et traitement à distance, etc.). Dans un second temps, nous proposons un modèle dynamique et sensible au contexte, permettant la gestion de contexte ainsi que l'adaptation afin d'harmoniser les potentialités du service développé avec son environnement et pour une meilleure utilisation des ressources dans des milieux très évolutifs. Nous nous focalisons sur le côté réactif, permettant d'adapter le comportement des applications à leurs conditions d'utilisation.

Le prototype de notre plateforme CAHS (Context-Aware Health Services) proposée dans le cadre de cette thèse est implémenté sous la plateforme Java IPOJO et basé sur l'adaptation orientée services SOA, elle offre la sensibilité des applications au contexte et permet d'assurer un meilleur fonctionnement de l'ensemble des applications dédiées à la santé. Notre démarche tend vers une nouvelle génération de plateformes mobiles intégralement ouvertes et unanimes en toutes les catégories d'applications dimensionnant la santé mobile.

Mot-clé— Contexte ; Système sensible au contexte ; Architecture orientée service ; Santé mobile ; iPOJO.

Abstract

Currently, the conception and the development of mobiles applications and services are still growing by the fact of more strong new horizons and services related to the context and the user situation.

The majority of available applications face the set of constraints which obstruct their execution. This is because designed applications are unsuitable for the conditions of execution. This makes it interesting to use a unified conception and execution model that can manage these applications.

Our work in a first step, presents an in-depth study of mobiles applications and their behavior, depending on the execution context. We consider the case of applications dedicated to health, designed to accompany and to assist a patient in his daily life (medical monitoring, permanent security, applications of remote analysis and treatment, etc.). In a second step, we propose a dynamic and context-aware model, allowing the context manager and an adaptation in order to harmonize the potentialities of the developed service with its environment and to assure a better use of resources in much evolved environments. We focus on the reactive side, allowing adapting the behavior of applications to their use conditions.

In this Thesis, the prototype of our Context-Aware Health Services (CAHS) platform is implemented under the Java IPOJO platform and based on the SOA. This platform provides the sensitivity of applications to context and ensures the proper functioning of all applications dedicated to health. The goal of our approach is to move towards a new generation of fully open and unified mobile platforms in all categories of applications that scale mobile health.

Keywords— Context; Context-aware system; Service oriented architecture; Mobile health; iPOJO.

Table des matières

Remerciement.....	vii
Résumé en arabe.....	ix
Résumé.....	xi
Abstract.....	xiii
Table des matières.....	xv
Table des figures.....	xviii
Liste des tableaux.....	xx
Glossaire.....	xxii

Introduction

Contexte.....	xxvi
Motivation et les objectifs de la thèse.....	xxvi
Organisation de la thèse.....	xxvii

Premier Chapitre: Etat de l'art et Problématique

I.1 Concepts généraux.....	31
I.1.1 Informatique Ubiquitaire.....	31
I.1.2 Informatique mobile.....	32
I.1.3 Santé mobile.....	33
I.2 Système sensible au contexte.....	34
I.2.1 Sensibilité du contexte.....	34
I.2.2 Aperçu général de l'architecture.....	35
I.2.3 Adaptation au contexte.....	36
I.2.3.1 Adaptation du contenu.....	36
I.2.3.2 Adaptation de service.....	36
I.2.3.3 Adaptation de la présentation.....	37
I.3 Travaux connexes.....	37
I.3.1 Systèmes sensibles aux contextes.....	37
I.3.1.1 Projet CAMPH (Context Aware Middleware for Pervasive Elderly Homecare).....	37
I.3.1.2 Projet CAMPUS (Context-Aware Middleware for Pervasive and Ubiquitous Service).....	38
I.3.1.3 Framework des services sensible au contexte.....	39
I.3.1.4 Plateforme CoCaMAAL (Cloud-oriented Context-Aware Middleware in Ambient Assisted Living).....	40
I.3.1.5 Projet BDCaM (Big Data for Context Aware Monitoring).....	41

I.3.1.6 Framework de la composition de services Web adaptatifs sensibles au contexte.....	42
I.3.1.7 Plateforme (Context-Aware Transport Service).....	43
I.3.2 Comparaison entre les systèmes sensibles au contexte.....	43
I.4 Problématique de la conception d'une plateforme sensible au contexte pour les applications de la santé mobile.....	45
I.4.1 Exécution de plusieurs applications sur Smartphones.....	45
I.4.2 Applications de la santé mobile.....	46
I.4.3 Présentation de notre problématique.....	47
I.4.3.1 Contexte – Identification et classification.....	47
I.4.3.2 Approche de la gestion de contexte.....	47
I.4.3.3 Mécanisme de l'adaptation au contexte.....	48
I.5 Notre contribution.....	48
I.5.1 Implémentation sous Android.....	50
I.5.2 Utilisation OSGi et iPOJO sur les applications Android.....	50
I.6 Conclusion.....	51

Deuxième Chapitre:
Contexte lié au domaine de la santé mobile

II.1 Contexte en système informatique.....	54
II.2 Contexte en santé mobile.....	55
II.2.1 Identification des éléments de contexte	55
II.2.1.1 Contexte de service	56
II.2.1.2 Localisation	56
II.2.1.3 Réseaux de communication.....	57
II.2.1.4 Profil d'utilisateur	57
II.2.1.5 Préférence d'utilisateur.....	57
II.2.1.6 Signes vitaux.....	57
II.2.1.7 Médicaments prescrits.....	57
II.2.1.8 Activités	57
II.2.1.9 Contexte d'appareil	57
II.2.2 Classification des éléments.....	57
II.2.2.1 Proposition d'une classification.....	58
II.3 Modélisation du contexte	59
II.3.1 Modèles de contexte.....	59
II.3.1.1 Modèle clé-valeur.....	60
II.3.1.2 Modèle à balise.....	60
II.3.1.3 Modèle graphique.....	61
II.3.1.4 Modèle orienté objet.....	61
II.3.1.5 Modèle basé sur la logique.....	62
II.3.1.6 Ontologie.....	62
II.3.2 Notre approche de la modélisation	64
II.3.2.1 Modélisation de la classification	64
II.3.2.2 Modélisation des éléments du contexte	65
II.4 Insertion du contexte dans l'application	67
II.5 Conclusion	68

Troisième Chapitre:

Méthodologie et modélisation: implémentation du modèle CAHS

III.1 Application à base des services sensibles au contexte.....	72
III.1.1 Concept de Service	72
III.1.2 Services dépendants de la situation de contexte (ScDS)	73
III.1.3 Description des versions de service.....	74
III.2 Mécanisme d'adaptation des services sensible au contexte.....	75
III.3 Description de la plateforme des Services de la Santé Sensibles au Contexte (CAHS).....	77
III.3.1 Architecture de la plateforme proposée.....	77
III.3.2 Implémentation de CAHS.....	78
III.3.3 Modules de la gestion de CAHS.....	78
III.3.3.1 Gestion de contexte.....	78
III.3.3.2 Implémentation de la gestion de contexte	80
III.3.3.3 Contrôleur-SDC (Contrôleur de service dépendant de contexte).....	82
III.3.3.4 Implémentation de contrôleur-SDC.....	83
III.3.3.5 Adaptateur.....	84
III.3.3.6 Implémentation d'adaptateur	85
III.3.4 Application à base des services de la santé mobile.....	86
III.3.5 Implémentation des applications à base des services.....	89
III.4 Fonctionnement de La plateforme CAHS.....	91
III.5 Conclusion.....	91

Quatrième Chapitre:

Résultats : Evaluation de la plateforme proposé CAHS

IV.1 Notre plateforme CAHS conçu.....	94
IV.2 Evaluation des performances de la gestion de contexte.....	94
IV.2.1 Test d'expérimentation.....	94
IV.2.2 Mesures de performances.....	97
IV.2.3 Conclusion.....	100
IV.3 Evaluation de mécanisme d'adaptation au contexte.....	100
IV.3.1 Tests de performance	101
IV.3.2 Déroulement de l'expérience.....	103
IV.3.3 Estimation de performances.....	104
IV.3.4 Conclusion.....	109
IV.4 Validation des résultats.....	109
IV.5 Evaluation.....	110
IV.6 Conclusion.....	111

Conclusion générale et perspectives.....	113
Références.....	117

Table des figures

FIG. I.1 – Services d'améliorer la qualité de la vie des personnes âgées	32
FIG. I.2 – Principe de la santé mobile.....	33
FIG. I.3 – Couches de système sensible au contexte.....	35
FIG. I.4 – Principe de l'architecture orientée service.....	36
FIG. I.5 – Architecture de CAMPH.....	38
FIG. I.6 – Architecture de CAMPUS.....	39
FIG. I.7 – Architecture de Framework des services sensible au contexte.....	40
FIG. I.8 – Architecture de CoCaMAA.....	41
FIG. I.9 – Architecture de la composition de services Web.....	42
FIG. I.10 – Architecture de CATS.....	43
FIG. I.11 – Applications exploitées sur Smartphone.....	46
FIG. I.12 – Mécanisme d'adaptation sensible au contexte.....	48
FIG. I.13 – Modèle de l'application à base des services.....	49
FIG. I.14 – Technologies OSGi / iPOJO sur android.....	50
FIG. II.1 - Contexte en système informatique.....	54
FIG. II.2 - Contexte en santé mobile.....	55
FIG. II.3 – Eléments de contexte.....	56
FIG. II.4 – Arbre des éléments de contexte.....	59
FIG. II.5 – Objectif de la modélisation de contexte.....	60
FIG. II.6 – Structure des données clé-valeur.....	60
FIG. II.7 – Structure hiérarchique.....	61
FIG. II.8 – Illustration du principe d'encapsulation.....	62
FIG. II.9 – Modélisation d'arbre des éléments de contexte.....	65
FIG. II.10 – Gestion de contexte externe.....	67
FIG. II.11 – Code non fonctionnel dépendant aux capteurs.....	67
FIG. II.12 – Serveur de contexte.....	68
FIG. III.1 – Service S avec leurs méthodes de calculs.....	72
FIG. III.2 – Services dépendants de la situation de contexte ScDS	73
FIG. III.3 – Description des versions de service.....	75
FIG. III.4 – Méthodologie de l'adaptation au contexte.....	76
FIG. III.5 – Architecture de la plateforme CAHS.....	77
FIG. III.6 – Compatibilité d'OSGi avec Android.....	78
FIG. III.7 – Gestion du Contexte.....	79
FIG. III.8 – Création du moniteur d'élément de contexte (M_{CEij}).....	80
FIG. III.9 – Diagramme du module de la gestion de contexte.....	81
FIG. III.10 - Contrôleur-SDC.....	82
FIG. III.11 - Algorithme de la création la matrice de classe HEC.....	83
FIG. III.12 - Algorithme de la détection des versions de service « affichage ».....	84
FIG. III.13 – Adaptateur.....	85
FIG. III.14 - Algorithme d'adaptation d'un service selon la situation de contexte.....	85
FIG. III.15 – Signal ECG	87

FIG. III.16 – Services d’affichage dépendant de la situation de contexte.....	88
FIG. III.17 – Manifest.mf d’une version 1 d’un service d’affichage.....	89
FIG. III.18 – les versions des services.....	90
FIG. IV.1 – Eléments appartiennent à la class1.....	95
FIG. IV.2 – Les éléments appartiennent à la class2.....	96
FIG. IV.3 – Structure de test.....	96
FIG. IV.4 – Méthode de calcul $T_{\text{retard}}(\text{HEC})$	97
FIG. IV.5 – Temps d’installation des bundles (Moniteurs).....	98
FIG. IV.6 – Temps de retard entre le démarrage M_{CEij} de class1.....	98
FIG. IV.7 – Temps de retard entre le démarrage M_{CEij} de class2.....	99
FIG. IV.8 – Mémoire utilisé pour gérer les Moniteurs de CE	100
FIG. IV.9 – Versions de service avec leur situation contextuelle.....	102
FIG. IV.10 – Configuration de l’expérimentation.....	103
FIG. IV.11 – Déroulement de l’expérience.....	104
FIG. IV.12 – Moyenne de Temps A, Temps B et Time_adapt.....	105
FIG. IV.13 – Time_adapt selon la taille de version.....	106
FIG. IV.14 – Moyenne de β dans tous les cas.....	107
FIG. IV.15 – Moyenne de β selon les différents nombres d’échantillons.....	108
FIG. IV.16 – Mémoire exploité pour le processus d’adaptation.....	109

Liste des tableaux

TAB. I.1 – <i>Définitions de la santé mobile</i>	34
TAB. I.2 – <i>Comparaison des systèmes sensibles au contexte</i>	44
TAB. II.1 – <i>Comparaison entre les modèles de contexte</i>	62
TAB. II.2 – <i>Paramètres d'un élément de contexte</i>	65
TAB. II.3 – <i>Éléments de contexte de chaque type</i>	66
TAB. III.1 – <i>La description d'une version de service</i>	74
TAB. III.2 – <i>Paramètres normaux du signal ECG</i>	87
TAB. IV.1 – <i>Cas étudiés</i>	95
TAB. IV.2 – <i>Cas étudiés</i>	101
TAB. IV.3 – <i>Moyenne de Time_adapt avec et sans α</i>	105
TAB. IV.4 – <i>Caractéristiques de chaque cas</i>	106

Glossaire

A

AAL : *Ambient Assisted Living.*
API : *Application Programming Interface.*

B

BDCaM : *Big Data for Context Aware Monitoring.*
BPEL : *Business Process Execution Language.*
BSN : *Body Sensor Network.*

C

CA : *Context Aggregator.*
CE : *Context Element.*
CDS : *Contexte Dependent Service*
CP : *Context Providers.*
CAHS : *Context-Aware Health Services Framework.*
CaM : *Context aware middleware.*
CAMPH : *Context Aware Middleware for Pervasive Elderly Homecare.*
CAMPUS : *Context-Aware Middleware for Pervasive and Ubiquitous Service.*
CAP : *Context aggregator and providers.*
CASF : *Context Aware Service Framework*
CATS : *Context-Aware Transport Service.*
CBSE : *Component-based software engineering.*
CDS : *Contexte Dependent Service.*
CMS : *Context Management System.*
CoCaMAAL : *Cloud-oriented Context-Aware Middleware in Ambient Assisted Living.*

D

DCF : *Data Collector and Forwarder.*

E

ECG : *Électrocardiogramme.*
EEG : *Électroencéphalogramme.*

ERM : *Entity-Relation Model*.

G

GNSS : *Géolocalisation et Navigation par un Système de Satellites*.

GPS : *Global Positioning System*

I

ipojo : *injected Plain Old Java Object*.

J

Jar : *Java Archive*.

M

MC : *Matrice de Classe*

MARCH : *Mobile Aware Server Architecture*.

MIT : *Massachussetts Institute of Technology*.

MOO : *modèle orienté objet*.

O

OSGi : *Open Service Gateway Initiative*.

ORM : *Objet-Relation Model*.

OWL-S : *Web Ontology Language for Web Services*.

P

PCS : *Serveurs de cloud personnel*.

R

RMS : *Systèmes de surveillance à distance*.

S

SCA : *Service Component Architecture*.

ScDS (*Situation of context Dependent Service*)

SECAS : *Component-based software engineering*.

SDK: *Software Development Kit*.

SP Fournisseurs de services

SPL : *Software Product Line*

SOA : *Service Oriented Architecture*.

T

TIC : *Technologie de l'Information et de la Communication.*

U

UDDI : *Universal Description Discovery and Integration.*

UML : *Unified Modeling Language.*

W

XML : *Extensible Markup Language.*

WSDL : *Web services description language.*

WSIG : *Westren States Investment Group.*

Introduction générale

Contexte général

Dans un passé récent, de nombreux terminaux informatiques, notamment les tablettes tactiles et les Smartphones deviennent une partie intégrante de la vie quotidienne d'utilisateur. Plusieurs technologies sont exploitées par ces appareils portables, telles que le réseau téléphonique 3G ou 4G, Wi-Fi, Bluetooth, technologie GNSS2, écran tactile, appareil photo et processeurs de grande puissance, etc. Ces technologies fournissent aux développeurs une atmosphère propice à la conception des applications intelligentes et développées. Le marché technologique a connu une énorme propagation des applications prenant comme fondement les systèmes d'exploitation mobiles (IOS avec AppStore, Android avec Google Play, . . .).

Actuellement, les grandes sociétés de programmation optent pour l'installation et le téléchargement des différentes applications à travers les terminaux mobiles.

Selon les statistiques, le marché mondial des applications a connu son apogée en matière de la productivité durant le deuxième trimestre de 2019 par un chiffre qui dépasse 30,3 milliards d'applications téléchargées [1].

Étant donné ces évolutions transitoires qui marquent le domaine de la technologie informatique, grâce aux chercheurs révolutionnant les Technologies de l'Information et de la Communication (TIC) qui comprennent: cloud computing, l'informatique ubiquitaire, Internet of Things, les réseaux ad hoc mobiles, etc.

Motivation et objectifs de la thèse

Partant de l'idée que le système informatique est l'objet d'une évolution permanente notamment les systèmes qui intègrent la notion de contexte dans l'exécution des applications mobiles.

Au fil des années ce sujet était largement traité dans l'état de l'art [2]. Nous pouvons mentionner à ce niveau quelques exemples de travaux actuels liés à ce thème. Les auteurs de travail [3] exploitent le contexte pour adapter les services de transport. Selon [4], le contexte est inséré dans les techniques d'auto-adaptation automatisées. Ces méthodes facilitent le développement d'applications mobiles flexibles, évolutives et adaptables au contexte. De plus, Le contexte exprime alors toutes les informations qui influencent l'exécution d'une application ou d'un service [5]. Il occupe une place prépondérante dans la création et l'exploitation des applications sensibles au contexte. Notre attention est orientée vers les applications de la santé mobile, avec ses problématiques particulières dues à des diversités et de masse de données médicales exploitées.

Dans le cadre de notre travail, nous ciblons de nombreuses applications qui peuvent fonctionner sur un terminal mobile et qui assistent et suivent l'état d'utilisateur en termes de leur vie de la santé. Ces applications fournissent une multitude de services dont l'utilisateur a besoin : la détection des situations médicales, l'affichage des informations médicales, la précision de la localisation des cabinets médicaux et le contact entre les patients et leurs médecins. L'utilisateur peut se trouver dans des différentes situations: le médecin, le patient, l'infirmière, etc. En outre, chacune d'entre elles se distingue par des caractéristiques

spécifiques, ce qui oblige de gérer les applications utilisées selon ses caractéristiques. On peut prendre l'exemple d'un utilisateur qui souffre simultanément de deux pathologies chroniques. Il nécessite un suivi médical minutieux en exploitant plusieurs applications afin que chaque application surveille une maladie spécifique. Pour atteindre cet objectif, le terminal devrait gérer les fonctionnalités pour une meilleure exploitation de ses ressources (mémoire et batterie). La gestion des fonctionnalités peut être réalisée par le démarrage et l'arrêt de son fonctionnement. L'étape clef de notre démarche compte plus spécifiquement sur l'identification des informations contextuelles, ainsi que sur la nécessité de leur intégration au niveau de chaque fonctionnalité. Ce qui exige le besoin d'une approche permettant de concevoir une situation contextuelle particulière adéquate à chaque fonctionnalité.

Les applications que nos utilisateurs pourraient exploiter, nécessitent une grande quantité d'informations pour détecter correctement leur état médical. L'environnement de l'utilisateur et la situation dans laquelle il emploie ses applications influencent leur exécution. Dans cette dimension la prise en compte de la totalité des aspects liés au contexte devient une exigence afin d'approfondir l'étude de leur influence sur le fonctionnement des applications de la santé mobile.

Compte tenu de nombreux travaux liés aux applications sensibles au contexte, ce domaine de recherche est jugé très vaste pour l'exploiter dans une optique multidimensionnelle. Alors, La proposition d'un modèle global permettant de couvrir tous les contextes des applications informatiques semble une mission impossible. C'est la raison pour laquelle que notre recherche sera restreinte. Nous prenons donc, l'initiative pour étudier ce sujet selon un angle bien défini afin de résoudre les lacunes entravant les systèmes informatiques, par l'intermédiaire d'une mise en valeur des informations contextuelles liées à l'utilisateur, les paramètres de son terminal mobile et l'environnement dans lequel il est impliqué.

L'intérêt de notre travail est axé sur la conception d'un modèle performant qui exploite la notion de contexte dans la gestion des applications dédiées à la santé, l'objectif en est d'assurer une bonne exécution de ces applications qui gèrent une immense et diverse quantité des données sur le terminal portable.

De prime abord, nous tentons d'analyser ce concept en considérant son impact sur le fonctionnement des applications et l'approche ayant l'ambition de l'exploiter dans le but de l'optimisation de leur performance.

Malgré les évolutions remarquables qui ont touché les applications mobiles dans le domaine de la santé et de même que la généralisation de l'usage de ces applications sur les différentes catégories de la population, nous allons envisager la création d'une plateforme sensible au contexte dotée de la capacité de gérer le comportement de ces applications pour fournir des divers services suivant plusieurs maladies au moyen de la même plateforme.

Organisation de cette thèse

L'architecture de notre travail comprend quatre chapitres : (I) État de l'art et Problématique, (II) Contexte lié au domaine de la santé, (III) Méthodologie et modélisation : implémentation du modèle CAHS, (IV) Résultats: Evaluation de la plateforme proposée CAHS.

La première partie de ce manuscrit, met en évidence les concepts généraux constitutifs de notre problématique, le système sensible au contexte et les travaux antérieurs s'inscrivant dans les différents domaines ayant une relation étroite avec ce système. Elle présente aussi les problématiques considérées afin de créer une plateforme de la gestion des applications mobiles pour le domaine de la santé.

Le deuxième chapitre a pour objectif de définir la notion de contexte selon notre étude. Nous identifions les éléments contextuels, qui influencent l'exécution des applications concernées et nous classifions ces composants en fonction des relations qui existent entre eux. Le sujet

abordé dans cette partie est lié à l'étude des modèles de contexte (clé valeur, à balise, graphique, orienté objet, basé sur la logique et ontologie) et ses comparaisons. Puis nous suggérons la modélisation de la catégorisation proposée et des éléments contextuels identifiés, exploitant le modèle à balise et orienté d'objet.

Dans le troisième chapitre de cette thèse, nous visons premièrement à présenter l'approche d'adaptation basée sur l'architecture orientée de services. De plus, nous analysons en détail la méthodologie et les mécanismes suivis pour la création et le développement de notre modèle CAHS sur les terminaux mobiles, avec la construction générale de CAHS. Nous créons les applications modulaires pour le domaine de la santé pour l'exécuter spécifiquement sur notre modèle CAHS.

La quatrième partie concerne l'évaluation la performance de plateforme CAHS pour la gestion des applications de la santé mobile. Nous nous focalisons sur l'approche d'adaptation en nous bénéficiant des avantages d'OSGi et iPOJO. Nous confirmons par la suite l'efficacité de nos propositions, qui se composent de deux parties : la gestion de contexte et le mécanisme d'adaptation. Enfin, nous essayons de valider notre méthodologie en comparant nos résultats obtenus avec ceux des autres travaux antérieurs.

Enfin, la conclusion générale et les perspectives de cette thèse sont exposées, en récapitulant les principales contributions et en citant nos futurs travaux de recherche.

Chapitre 1

Etat de l'art et Problématique

Ce chapitre décrit les problèmes liés aux systèmes sensibles au contexte. Il est divisé en cinq sections : La section 1 présente la diversité des concepts généraux dans le cadre de notre travail. La deuxième section expose un aperçu général de système sensible au contexte en définissant leur architecture et les type d'adaptation selon les informations contextuelles. La troisième section sera consacrée à introduire ces types de systèmes existants en comparant leurs techniques et leurs mécanismes exploités. Dans la section 4 nous présentons les problématiques abordées dans ce contexte afin de proposer une plateforme mobile d'exécution dédiée à la santé. Enfin, la dernière partie présente le cadre de notre proposition pour l'implémentation des applications et des services.

I.1 Concepts généraux

I.1.1 Informatique Ubiquitaire

L'exploitation de l'informatique et la technologie de l'information et de la communication (TIC) dans de nombreux domaines conduit à l'apparition d'un nouveau concept de l'informatique, nommé *l'informatique ubiquitaire (omniprésente)*. Le premier utilisateur de terme «*informatique omniprésente*» (*UbiComp*) est Mark Weiser [6].

Aujourd'hui, l'informatique *omniprésente* devient une technologie importante pour développer les systèmes d'informations. *L'informatique ubiquitaire* permet de faciliter l'intégration des relations entre les objets de l'informatique et l'environnement d'utilisateur ou entre les utilisateurs eux-mêmes à proximité ou à distance, tout en exploitant toute la gamme des technologies modernes de l'information et de la communication (TIC).

Selon Friedewald et Raabe [7], l'utilisation et l'omniprésence des technologies de l'informatique dans les différents domaines de la vie quotidienne permet d'offrir plusieurs concepts («informatique omniprésente», «intelligence ambiante» et «Internet des objets»). Ces concepts ciblent un objectif commun qui est l'assistance des gens dans leur vie quotidienne à travers l'optimisation et le développement continu des applications dédiées aux différents domaines tel que la formation, le transport, la médecine et le tourisme,....

Les auteurs [7] résumant les caractéristiques de l'informatique ubiquitaire dans les points suivants :

- L'utilisation du matériel informatique et des logiciels dans la vie quotidienne,
- La mobilité permet de fournir des services n'importe où, à n'importe quel moment !
- Le système a la capacité de la sensibilité au contexte et de s'adapter selon des nouvelles informations courantes,
- La reconnaissance automatique et le traitement autonome de tâches répétitives sans l'intervention de l'utilisateur.

Les petits appareils d'informatique (le Smartphone ou l'assistant personnel) sont la partie intégrante dans la vie quotidienne. Dans cette optique, plusieurs applications sont apparues dans l'informatique ubiquitaire. Friedewald et Raabe [7] présentent certains domaines d'application, notamment la vente au détail, la production industrielle et la gestion du matériel, l'identification personnelle et l'authentification, le soin de la santé ainsi que la logistique du transport. De plus, L'informatique ubiquitaire liée au domaine de la santé a été abordée depuis quelques années sous le terme de la télématique de la santé, mais récemment, elle est connue par AAL (Ambient Assisted Living) qui permet d'améliorer la qualité de la vie des personnes âgées et de suivre les malades à la maison [8]. La figure (FIG. I.1) explique le parcours du suivi à distance.

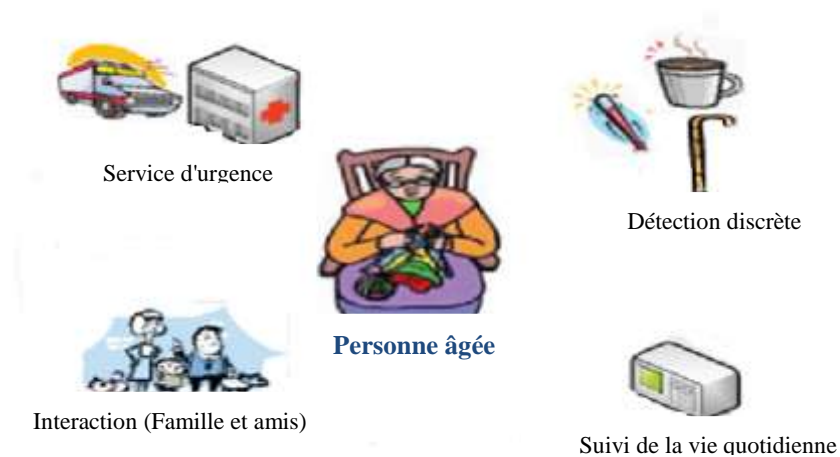


FIG. I.1 – Services d'améliorer la qualité de la vie des personnes âgées

L'exploitation des méthodes de l'acquisition et des informations de manière autonome et continue imposent des exigences beaucoup plus élevées sur la performance technologique. Pour cela, il est nécessaire de fournir des guides et des outils qui permettent de gérer facilement les informations personnelles ainsi que l'adaptation des services sans avoir l'obstruction. L'état de l'art défini en [9] montre les outils et les techniques d'AAL pour les personnes âgées ; Peter et all [10] proposent une plate-forme *openAAL* qui permet de faciliter la mise en œuvre, la configuration et la mise à disposition des services flexibles, contextuels et personnalisés. Les auteurs [11] développent un système de surveillance des soins de santé qui offre un service d'alarme pour la mesure et la notification en cas de niveaux inhabituels d'indicateurs de patients, sur la base des données de bio-signal de l'utilisateur et du contexte. L'informatique ubiquitaire a ajouté des privilèges au développement de systèmes des informations. Elle permet de fournir des nouvelles exigences et de meilleurs services tels que le suivi à domicile des maladies, la gestion des circulations de la route, l'assistance des touristes et la gestion de la vente. Bien que l'informatique ubiquitaire possède plusieurs avantages qui servent les différents domaines (transport, santé, éducation, touriste, ...), mais il existe des défis qui doivent être traités.

I.1.2 Informatique mobile

Les appareils mobiles deviennent des outils de communication les plus efficaces et les plus pratiques, à tout moment et en tout lieu. Le taux d'usage des terminaux mobile connaît l'augmentation continue ces dernières années. Le progrès rapide de l'informatique mobile [12] devient une tendance puissante dans le développement de la technologie informatique ainsi que dans les domaines du commerce et de l'industrie. Cependant, les dispositifs mobiles sont confrontés à de nombreux défis dans leurs ressources (la durée de vie de la batterie, le stockage et la bande passante) et les communications (la mobilité et la sécurité) [13]. Pour illustrer le potentiel de l'informatique mobile sur Smartphone, nous présentons trois exemples d'applications :

Assistant personnel Technologie. Un téléphone mobile est capable d'accès aux informations personnelles tel que l'historique de navigation Web, les événements d'agenda et les contacts de réseaux sociaux en ligne. Les développeurs d'applications peuvent concevoir les applications qui utilisent ces données permettant d'aider et de prédire les intentions des utilisateurs. *MindMeld* permet d'améliorer le service de la vidéoconférence en ligne avec des informations pertinentes aux utilisateurs. Google Now adopte une approche plus générale et vise à fournir à l'utilisateur de téléphone portable toutes les informations ou les

fonctionnalités dont il peut avoir besoin, sans que l'utilisateur ne le demande explicitement [14].

Soins de santé. La détection mobile a été proposée comme un moyen de diagnostic [15]. De plus, les téléphones mobiles sont utilisés pour offrir des thérapies personnalisées [16-18]. Actuellement, ces thérapies sont téléchargeables comme les applications sur Player store. Dans l'informatique mobile, Le calcul anticipatoire peut construire et développer un modèle de comportement humain et concevoir des thérapies automatiquement, visant à amener l'utilisateur vers un certain objectif de bien-être. Par exemple, le téléphone peut détecter le niveau d'activité physique de l'utilisateur, grâce à un accéléromètre intégré, un capteur Bluetooth ainsi que le comportement d'appel.

Villes intelligentes. Le pourcentage de la population urbaine connaît une croissance remarquable et de plus la moitié de populations résident dans des zones urbaines [19]. Ceci Conduit à l'émergence de plusieurs problèmes par exemple le trafic, la pollution et la criminalité. Ces problèmes affectent les villes modernes et la vie des populations en particulier.

Il existe des applications sur des terminaux mobiles qui permettent de résoudre ces problèmes. Ces applications comptent sur la détection mobile participative, où les citoyens sont activement impliqués dans la collecte de données, ainsi que la détection mobile opportuniste, où les utilisateurs se chargent ces applications autonomes sur leurs appareils. Le projet *CarTel* du MIT utilise la détection mobile pour l'atténuation du trafic, la surveillance de la circulation des routes et la détection des dangers [20]. Le système *ParkNet* [21] recueille des informations sur l'occupation des espaces de stationnement grâce à la détection distribuée par les véhicules de passage. Dutta et al. [22] implémentent une architecture de détection participative pour surveiller la qualité de l'air.

I.1.3 La santé mobile

Aujourd'hui, le développement des applications de l'informatique mobile et des technologies de la communication prend une importante place dans le domaine de la santé. Ce concept définit la santé mobile (M_health) comme un sous-segment important du domaine de la santé électronique (E_health) [23]. Plus largement, la santé mobile est capable de créer, collecter, analyser, stocker et transmettre des informations tout en assurant la sauvegarde des vies humaines.

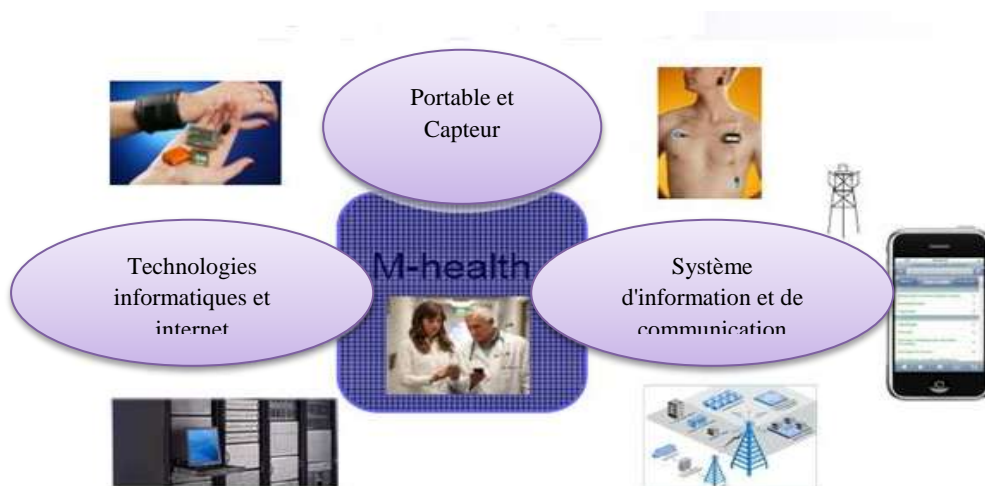


FIG. I.2 – Principe de la santé mobile

Le but du M-health est l'amélioration de la qualité de soins de la santé [24]. Le tableau (TAB. I.1) récapitule plusieurs définitions proposées pour la santé mobile :

Travail	Définition
[24]	Il est défini comme l'utilisation des communications mobiles - telles que les PDA et les téléphones mobiles - pour les services de santé et l'information.
[25]	Il utilise les nouvelles technologies de communication mobile et de réseau pour les soins de santé, comme illustré dans la figure (FIG. I.2).
[26]	Il s'agit d'un sous-ensemble d'e_santé utilisant des appareils mobiles pour fournir des services de santé aux patients.
[27]	Il se réfère à l'application de périphériques sans fil intégrés pour suivre les paramètres liés à la santé.
[28]	Il s'agit d'un service personnalisé et interactif qui offre un accès omniprésent et universel aux conseils médicaux et informations à tous les utilisateurs à tout moment sur une plate-forme mobile.

TAB. I.1 – Définitions de la santé mobile

D'après le tableau (TAB. I.1), nous remarquons que le principe de la santé mobile est le même malgré les différentes définitions proposées par les chercheurs. Mais à part les auteurs [24] qui donnent une définition plus générale, tous les auteurs proposent des définitions selon les axes de recherche ciblés.

I.2 Système sensible au contexte

Ces dernières années, les dispositifs, les applications et les services informatiques ont rapidement développé. Ceci rend plus complexes, les utilisateurs quotidiens sont passent plus de temps et d'efforts à configurer et à instruire ces appareils et ces services. Le système sensible au contexte est un paradigme émergé pour surmonter ces problèmes.

I.2.1 Sensibilité du contexte

Des nombreuse travaux et projets exploitent la sensibilité du contexte dans leur domaine de recherche. Ils visent par exemple à définir ce terme dans le système informatique, à détecter, interpréter et répondre aux aspects d'un l'environnement local de l'utilisateur et les dispositifs informatiques eux-mêmes [29], à offrir la plus grande souplesse de service possible en fonction du contexte en temps réel [30] ou encore à fournir automatiquement des informations ainsi que à prendre des mesures selon le contexte actuel de l'utilisateur et le besoin [31]. Selon N. Ryan [32], si une application permet d'exploiter les dispositifs de la détection et de choisir le contexte approprié aux besoins ou aux intérêts de l'utilisateur, elle peut être décrite comme une application sensible au contexte.

La définition en générale est proposée par Anind et al. [33] «Un système est sensible au contexte s'il utilise le contexte pour fournir des informations et / ou services pertinents à l'utilisateur, dont la pertinence dépend de la tâche de l'utilisateur». Cette définition nous permet de bien expliquer le système sensible au contexte en termes de leurs caractéristiques et leurs objectifs. Nous détaillerons par la suite l'architecture de ce système.

I.2.2 Aperçu général de l'architecture

Les auteurs des travaux [34-35] montrent que le système sensible au contexte est un cadre stratifié composé de bas en haut par : capteurs, récupération de données brutes, prétraitement, stockage / gestion et couche d'application (**FIG. I.3**). Par ailleurs, Il est chargé de récupérer les données brutes des capteurs, d'extraire et de combiner les données détectées dans un contexte de haut niveau, puis de les rendre disponibles pour les applications sensibles au contexte.

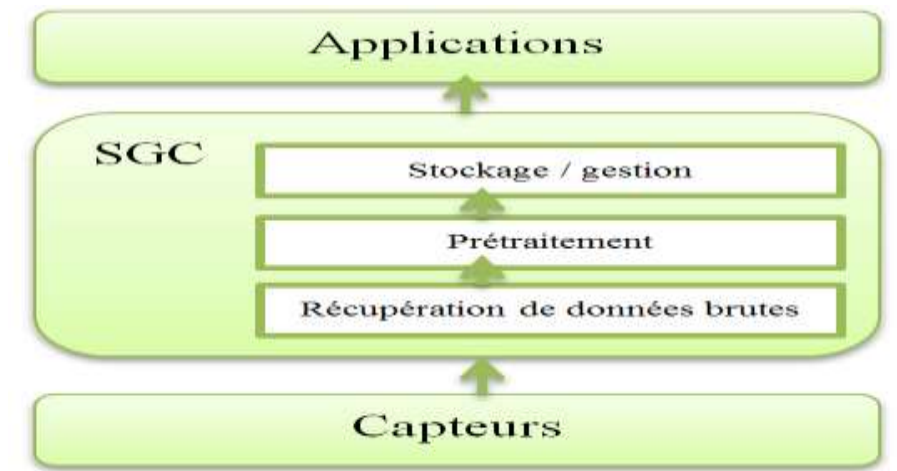


FIG. I.3 – Couches de système sensible au contexte

La première couche (capteurs) est un ensemble de capteurs chargé de collecter des données brutes provenant de l'environnement utilisateur (par exemple, dispositif utilisateur, activités, localisation, etc.). Les capteurs peuvent être classés en trois groupes [34] :

Capteurs physiques. Les capteurs les plus fréquemment utilisés sont les capteurs physiques. De nombreux capteurs matériels sont disponibles de nos jours, capables de capturer presque toutes les données physiques.

Capteurs virtuels. Il s'agit de données capturées à partir d'applications et des services logiciels. Par exemple, en utilisant un calendrier électronique.

Capteurs logiques. Ces capteurs utilisent plusieurs sources d'informations et combinent des capteurs physiques et virtuels avec des informations supplémentaires provenant de bases de données ou de diverses autres sources afin de résoudre des tâches plus complexes.

La deuxième couche (récupération de données brutes) utilise une API ou les protocoles spécifiques pour demander des données à la première couche. Ces interrogations doivent être mises en œuvre, rendant possible le remplacement des capteurs.

La troisième couche interprète les informations contextuelles. Cette couche transforme les informations renvoyées par la couche inférieure (récupération de données brutes) au niveau d'abstraction supérieur. Ces données détectées ou déduites doivent être modélisées pour une description claire.

La quatrième couche (stockage et gestion) permet de stocker, organiser et transmettre les données collectées via une liaison synchrone ou asynchrone. Dans le premier mode, le client interroge périodiquement le serveur pour connaître les modifications effectuées via des appels

de méthode distants. Le deuxième mode fonctionne via des abonnements. Chaque client s'inscrit aux événements spécifiques qui l'intéressent. Lors de l'apparition de l'un de ces événements, le client est simplement averti ou sa méthode est directement impliquée par un appel précédent.

La cinquième couche (Application) réagit aux changements de contexte mises en œuvre, par exemple, l'affichage du texte dans un contraste de couleurs plus élevé pour un mauvais éclairage.

I. 2.3 Adaptation au contexte

Le mécanisme d'adaptation est une étape primordiale dans le cycle du développement des systèmes sensibles aux contextes. Nous avons remarqué qu'il y a trois types principaux d'adaptation. Chaque type permet de résoudre certaines contraintes qui constituent un obstacle pour l'exécution des applications.

I. 2.3.1 Adaptation du contenu

L'adaptation de contenu peut être illustrée par la transformation du type ou du format des données afin de répondre aux exigences du contexte prédéfini (contraintes des terminaux et les réseaux mobiles).

Plusieurs techniques d'adaptation du contenu multimédia ont été présentées dans l'état de l'art [36]. J. Nam et al [37] exposent les méthodes pour adapter le contenu visuel aux préférences de l'utilisateur en termes de niveau de couleur. Un autre projet nommé Mobile Aware Server Architecture (MARCH) [38] cible une architecture distribuée. Il s'agit de l'adaptation de contenu multimédia échangé dans les environnements client-serveur pour l'exploitation potentielle des capacités des terminaux mobiles et ces réseaux d'accès.

I. 2.3.2 Adaptation de service

Ce type d'adaptation permet de changer l'état (active ou désactive) ou la forme de service pour un certain événement produit. Parmi les architectures adaptables exploitées, il y a l'architecture orientée service (SOA) qui est basée sur la composition des services. Cette composition comporte trois acteurs : fournisseur, consommateur et annuaire de services (FIG. I.4). Selon Papazoglou et al. [39], SOA est défini comme une source de fourniture de services issue d'un système. Ces services sont disponibles au niveau de fournisseurs et exploités par des consommateurs qui peuvent partager le même service. Enfin, un annuaire comporte des informations décrivant ces services implémentés.

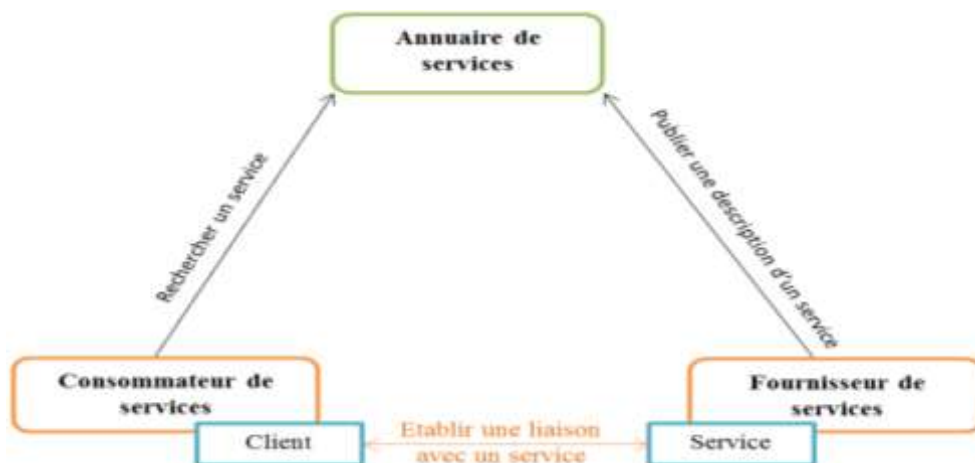


FIG. I.4 – Principe de l'architecture orientée service

L'avantage de SOA est de réduire la dépendance entre les clients et les fournisseurs. Ceci permet d'évaluer chaque élément à part, ce qui rend l'application plus flexible [40]. L'autre propriété fondamentale est le couplage lâche entre les services [41]. Le désavantage de SOA est la disponibilité des services et l'infrastructure de communication entre le fournisseur et le consommateur.

De nombreux travaux visent à réaliser ce genre d'adaptation, à titre d'exemple on peut mentionner la plateforme CATS [42] basée sur SOA qui propose un mécanisme d'adaptation, ou encore le projet de M. Miraoui [43] qu'introduit l'approche d'adaptation des services dans un système informatique omniprésent.

I. 2.3.3 Adaptation de la présentation

Elle garantit une présentation des données appropriées à la préférence d'utilisateur (la langue...) et aux caractéristiques des terminaux mobiles (taille d'écran, résolution...) [44].

Les travaux de T. Lemlouma et al. [45] implémentent un modèle permettant de faciliter les processus d'adaptation pour assurer une présentation adéquate sur les petits périphériques, en exploitant les services Web, la gestion de contexte et la recherche de profils. Le projet SECAS [46] offre un mécanisme d'adaptation qui assure l'adaptation des données, des services et des interfaces des utilisateurs aux différentes situations contextuelles.

Après avoir analysé les différents types d'adaptation au contexte, nous proposons dans le cadre des travaux de cette thèse un modèle de gestion des applications de la santé qui s'articule autour d'une architecture orientée services SOA. Il s'agit d'implémenter des méthodes d'adaptation au comportement des applications à leurs nouvelles conditions d'utilisation permettant non seulement d'harmoniser les potentialités du service développé dans son environnement mais aussi d'assurer une meilleure utilisation des ressources dans des milieux très évolutifs.

I.3 Travaux connexes

Des travaux et projets sont réalisés à travers le monde dans les concepts et les objectifs de la gestion des applications au contexte. Ils visent généralement, à définir une architecture générique des systèmes sensibles au contexte (collection des données, adaptation...).

I.3.1 Systèmes sensibles aux contextes

I.3.1.1 Projet CAMPH (Context Aware Middleware for Pervasive Elderly Homecare)

Le CAMPH [47] est un intergiciel entre une infrastructure matérielle et les diverses applications sensibles au contexte dans le domaine de santé. CAMPH regroupe plusieurs tâches, notamment l'acquisition de données, le stockage et le raisonnement contextuel ainsi que l'organisation et la découverte des services. La figure (**FIG. I.5**) illustre l'architecture de CAMPH composée de quatre couches logiques.

Couche d'espace physique : Chaque espace contient des entités physiques telles que des capteurs, des actionneurs et des dispositifs informatiques. Au niveau de cette couche, les données collectées sont modélisées par un modèle attributs-valeur. Les espaces physiques ayant les mêmes attributs sont triés dans la même classe, c'est ce qu'on appelle *le domaine de contexte* ou *l'espace de contexte*.

B- Couche de gestion des données contextuelle : Les principes composants situés dans cette couche sont : la base de données (données/événements) de contexte, les traitements des requêtes, la gestion de l'espace de contexte et les raisonnements de contexte.

C- Couche de gestion des services : les données de contexte sont utilisées pour l'organisation et la découverte des services sensibles au contexte.

D- Couche d'application : Les différentes applications de la santé peuvent demander des services et des données sensibles au contexte.

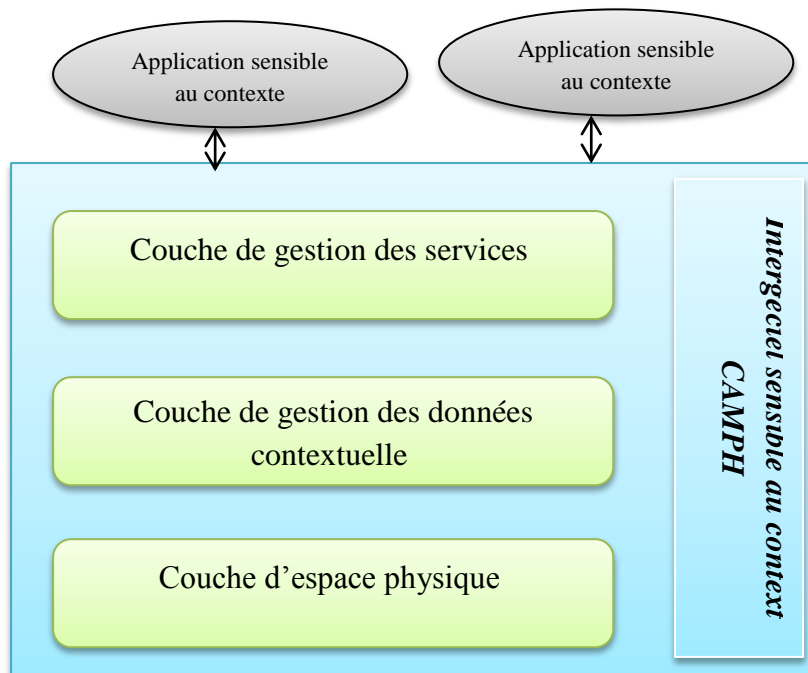


FIG. I.5 – Architecture de CAMPH

Les problématiques de ce système sont : l'interchangeabilité et l'incompatibilité des données contextuelles entre les différents domaines de contexte, l'inexistence d'un mécanisme de la gestion de grande quantité des données contextuelles issues de plusieurs espaces, la sécurité et la confidentialité.

I.3.1.2 Projet CAMPUS (Context-Aware Middleware for Pervasive and Ubiquitous Service)

CAMPUS [48] est l'abréviation de middleware sensible au contexte pour les services pervasive et ubiquitaire. Il est proposé afin d'automatiser les détections d'adaptation sensible au contexte basé sur trois technologies clés : l'adaptation compositionnelle, l'ontologie et la logique de description/la logique de premier ordre. Ce projet permet de requêter les décisions d'adaptation automatiquement au moment d'exécution sans exploiter les politiques d'adaptation prédéfinies car elles ne prennent que les changements contextuels limités dans une situation dynamique. L'architecture de CAMPUS (FIG. I.6) comprend trois couches typiques :

A- Couche de programmation : Elle permet de construire et de reconfigurer les applications sensibles au contexte en adoptant les instructions de la couche de décision.

B- Couche de connaissance : Il existe trois ontologies : *le modèle de contexte*, *le modèle de « tasklet »* et *le modèle de service*. Ces ontologies sont proposées pour représenter la sémantique de la connaissance qui est nécessairement requise par CAMPUS pour prendre les décisions d'adaptation.

C- Couche de décision : Le décideur exploite un modèle décisionnel normatif. Ce modèle implémente plusieurs étapes : le prétraitement, le criblage et le choix.

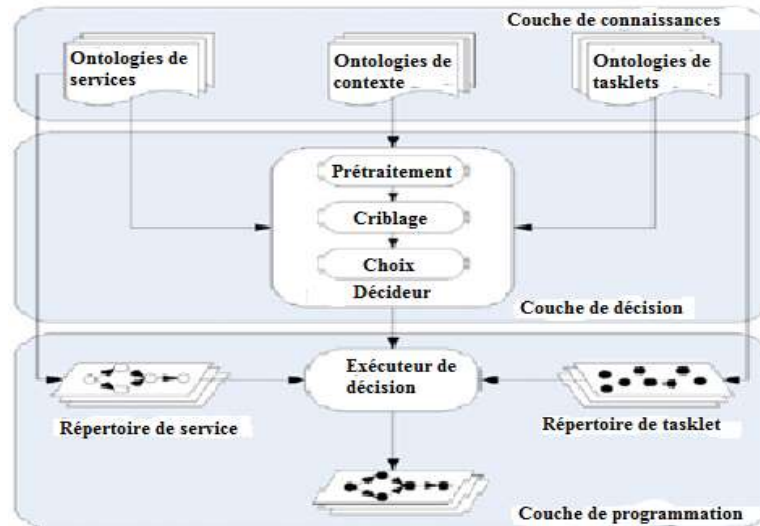


FIG. I.6 – Architecture de CAMPUS

CAMPUS fournit une solution efficace pour intégrer la sensibilité de contexte dans le développement des applications. Il pourrait automatiquement prendre les décisions au moment de l'exécution des applications grâce à une prise de décision adéquate via la sémantique. Cependant, l'aspect de la sécurité n'a pas été mentionné dans cette proposition. La prise de décision en collaboration avec plusieurs instances de middleware CAMPUS reste comme une perspective.

I.3.1.3 Framework des services sensible au contexte

CASF (Context Aware Service Framework) [49] permet de fournir la diversité des services sensible au contexte. Juyoung et al ont mis au point cette nouvelle architecture qui résout le problème de la capacité de composition et de découverte de services. Cette infrastructure est basée sur les services Web sémantiques tenant en compte la capacité de prendre en charge la découverte et l'intégration automatiques de services. L'architecture de ce Framework contient trois couches comme illustré dans la figure (FIG. I.7):

A- Couche de capteur physique: Cette couche est consacrée aux données issues du capteur.

B- Couche de contexte public: Elle inclue deux types de fournisseurs de contexte: le premier est un fournisseur de contexte de base qui traite uniquement les données collectées via les capteurs physiques. Le second est un fournisseur de contexte combiné qui peut exploiter les informations générées, soit par les capteurs ou les autres fournisseurs de contexte.

C- Couche de service de contexte: Les informations contextuelles sont exploitées pour fournir les services sensibles au contexte.

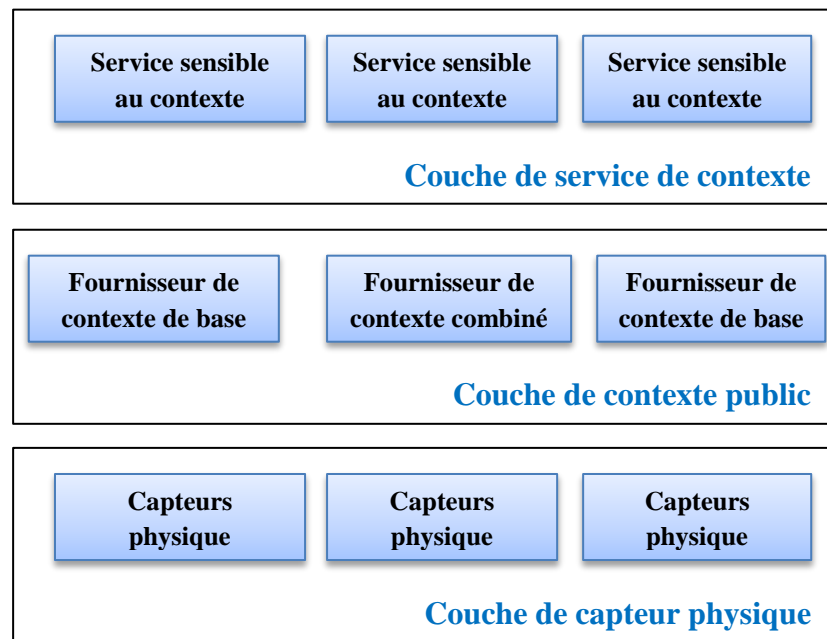


FIG. I.7 – Architecture de Framework des services sensible au contexte

La particularité de cette proposition est l'utilisation de services Web sémantiques, ce qui permet de réaliser une découverte et une intégration automatiques pour les informations de contexte. Cependant, de nombreuses études devraient être menées pour compléter cette proposition. Tout d'abord, les protocoles et les ontologies doivent être spécifiés en détail pour traduire les informations contextuelles en des services Web.

I.3.1.4 Plateforme CoCaMAAL (Cloud-oriented Context-Aware Middleware in Ambient Assisted Living)

Forkan et al. [50] proposent un cadre de travail (middleware) CoCaMAAL (Cloud-oriented Context-Aware Middleware in Ambient Assisted Living). L'objectif de ce projet est de faciliter le flux entre la collecte et le traitement de données dans des scénarios AAL. Tout d'abord, la conception de CoCaMAAL est basée sur l'architecture SOA. Cette architecture effectue la modélisation de contexte pour les données brutes, la gestion et l'adaptation de données contextuelles, le mappage de services sensibles au contexte, la distribution et la découverte de services. La figure (FIG. I.8) présente ce middleware, qui comprend cinq principaux composants orienté cloud :

A- Système AAL : L'architecture matérielle de ce système comprend le BSN (Body Sensor Network) et les systèmes de surveillance.

B- Agrégateur et les fournisseurs de contexte (CAP): Les données brutes des systèmes AAL sont converties et modélisées en contexte de haut niveau.

C- Fournisseurs de services : Ils sont les producteurs de services sensibles au contexte, tels que des applications.

D- Middleware sensible au contexte (CaM) : Selon les connaissances existantes et le contexte entrant, CaM est capable d'identifier les services d'assistance pour le contexte donné et de déclencher des actions associées.

E- Visualisation des données contextuelles: les interfaces appropriées (par exemple une interface graphique) sont disponibles pour les utilisateurs afin de visualiser les données du contexte.



FIG. I.8 – Architecture de CoCaMAA

Les résultats d'évaluation prouvent que CoCaMAAL est une solution efficace pour collecter, modéliser et exploiter le contexte des environnements AAL afin de fournir des services appropriés. La particularité de ce travail réside dans l'adoption du cloud computing, qui offre une capacité puissante en informatique pour le traitement et la gestion.

Cependant, plusieurs problèmes préoccupants ne peuvent être ignorés. Par exemple, les conflits en contexte ne sont pas pris en compte et l'analyse de fiabilité n'est pas effectuée dans ce modèle. Bien que Forkan et al. [50] indiquent que le contrôle d'accès basé sur le rôle sensible au contexte et le protocole de service garantissent la confidentialité dans cet intergiciel. Mais ces deux approches mentionnées ne sont pas incluses dans le test.

I.3.1.5 Projet BDCaM (Big Data for Context Aware Monitoring)

BDCaM [51] c'est l'extension de CoCaMAAL. Il permet de traiter une préoccupation supplémentaire qui est la découverte de connaissances personnalisée. L'approche principale de la découverte de connaissances personnalisées consiste à dériver / apprendre des anomalies spécifiques au patient à partir de quantités de données.

Une méthodologie d'apprentissage en deux étapes est exploitée dans ce système pour dériver des informations plus utiles. Ces informations sont exploitées pour la décision sensible au contexte. Les procédures spécifiques de cette approche d'apprentissage sont: les corrélations entre les attributs de contexte, les valeurs de seuil et l'apprentissage supervisé.

La première adapte les règles d'association au patient tout en exploitant l'algorithme MapReduce Apriori [52]. La seconde définit des données de contexte selon des règles générées lors de la première étape.

Comme l'architecture de BDCaM est inspirée du CoCaMAAL, le modèle BDCaM implémente plusieurs composants distribués et basés sur le nuage. Ces composants sont: systèmes AAL, Serveurs de cloud personnel (PCS), Collecteur de données et redirecteur

(DCF), Agrégateur de contexte (CA), Fournisseurs de contexte (CP), Système de gestion de contexte (CMS), Fournisseurs de services (SP) et Systèmes de surveillance à distance (RMS). Les résultats d'implémentation montrent que ce système est fiable et efficace. Il introduit une solution pour détecter les anomalies liées à la santé du patient. Cependant, la sécurité et la confidentialité des données personnelles ne sont pas résolus dans ce système. De plus, la généralisation de ce système pour couvrir plusieurs domaines d'applications constitue une perspective très lointaine.

I.3.1.6 Framework de la composition de services Web adaptatifs sensibles au contexte

Cette plateforme [53] proposée par Z. Cao et al comporte la composition de service Web adaptatif sensible au contexte. Ce système est implémenté sous BPEL (*Business Process Execution Language*) pour décrire les processus de la composition de service Web. Il exploite aussi un type de service web spécifique appelé *service de contexte* pour supporter la sensibilité au contexte. Cette plateforme est basée sur la technologie des agents permettant au service de contexte de percevoir et de traiter les contextes et la composition du service Web. L'architecture est décrite sur la figure (FIG. I.9) :

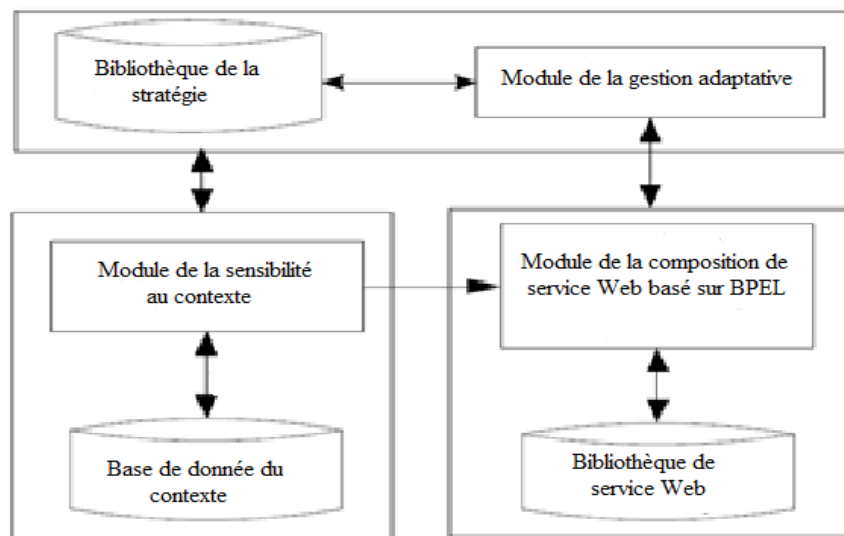


FIG. I.9 – Architecture de la composition de services Web

A- Module de la composition de service web : Ce module effectue deux tâches: la première consiste à rassembler les services Web pertinents et enregistrer ces services dans une base de données. La deuxième est de décrire les processus de la composition de service Web selon la norme BPEL.

B- Module de la sensibilité au contexte : Lorsque le processus BPEL est exécuté, ce module implémente la collection, la vérification des informations contextuelles et la modification de la valeur du contexte en informant le module de gestion adaptative afin qu'il s'adapte aux nouveaux changements.

C- Module de la gestion adaptative : Ce module contient Update Agent pour exploiter la stratégie de la composition adaptative en fonction de la valeur de contexte. Il envoie la stratégie choisie au moteur d'exécution BPEL qui modifie le processus de composition selon cette stratégie. Les stratégies dans la bibliothèque sont mises en place en fonction des valeurs de variation des contextes associés.

Cette plateforme peut améliorer la précision et l'efficacité de la composition du service Web et le rendre plus personnalisé. Mais l'utilisation de l'intergiciel WSIG pour la transformation des messages entre les services Web et les agents influence dans une certaine mesure sur les

performances des agents et des services Web. Pour cela, il est nécessaire de résoudre ce problème dans le futur afin d'améliorer l'efficacité de la composition.

I.3.1.7 Plateforme (Context-Aware Transport Service)

CATS (Context-Aware Transport Service) [3] est une plateforme qui gère l'exécution des applications mobiles issue du domaine du transport. CATS est basé sur les principes SOA pour concevoir les modules de la gestion et les applications. La SOA assure une architecture flexible et dynamique. Donc, cette plateforme offre un moyen simple et efficace pour gérer le contexte sur un simple terminal mobile. L'architecture de CATS est schématisée sur la figure (FIG. I.10):

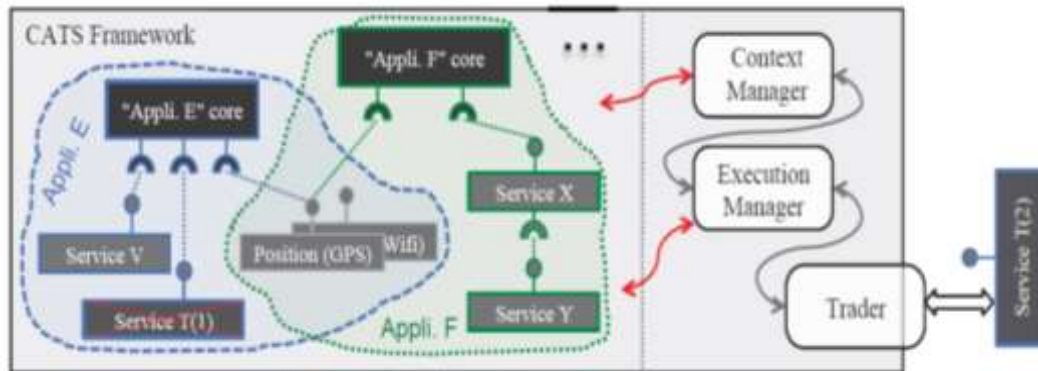


FIG. I.10 – Architecture de CATS

A- Gestion du contexte : Il évalue l'état du contexte via le moniteur de contexte. Cette évaluation peut être effectuée, soit par l'appel de démarrage qui est exigé par la gestion de contexte à un moment donné, soit automatiquement selon une fréquence spécifiée.

B- Gestionnaire d'exécution : Ce module gère l'état de service (installer, démarrer, arrêter et désinstaller) selon les informations contextuelles. Il peut assurer un meilleur assemblage des applications (l'adaptation de comportement) en fonction de contexte.

C- Courtier : Il résout le problème des services indisponibles, en permettant de chercher et télécharger des services à partir d'autres terminaux.

Les résultats d'évaluation du CATS montrent que la gestion de contexte peut être effectuée juste sur le terminal mobile et les applications sont capables de se reconfigurer automatiquement au cours d'exécution. Cette reconfiguration est réalisée par le changement d'état des services, qui peuvent être téléchargés et installés à la volée. L'aspect de la sécurité et de la confidentialité est ignoré dans le cadre de ce travail. Cette plateforme ne cible que les applications dédiées au domaine de transport. Ainsi, la génération de cette plateforme vers d'autres domaines constitue une grande perspective. À partir de cette raison, nous envisageons cette perspective comme une motivation dans le cadre de notre travail.

I. 3.2 Comparaison entre les systèmes sensibles au contexte

Après avoir analysé et évalué les différents projets liés à la sensibilité du contexte, nous présentons dans cette section une étude comparative entre ces projets en termes de nombreux paramètres exploités dans la conception de leurs systèmes afin de définir notre problématique et par conséquent notre contribution. Le tableau (TAB. I.2) récapitule les résultats comparatifs des modèles étudiés auparavant :

CATS	[53]	CoCaMAAL	CASF 2013	CAMPUS 2013	CAMPM 2009	Architecture		la relation entre
						gestion de contexte	Composant de l'adaptation	
Gestion de contexte	Module de la sensibilité au contexte	Agrégateur de contexte et les fournisseurs	Couche de contexte public	Couche de la connaissance	Gestion des données contextuelles	gestion de contexte	Modèle de contexte	de
Gestionnaire d'exécution	Module de la gestion adaptative	Middleware sensible au contexte	Couche de service de contexte	Couche de la décision	Gestion des services	Composant de l'adaptation	Modèle de contexte	de
XML et orienté d'objet	Non	Ontologie	Ontologie	Ontologie	Key-value	Modèle de contexte	Modèle de contexte	de
Non	Utilisateur Physique service	Personne Localisation Devise	Non	Computational Utilisateur	Localisation User Température Vital signes	Composant de l'adaptation	Modèle de contexte	de
Service	Service	Service	Service	Tacklets	Service	Composant de l'adaptation	Modèle de contexte	de
SCA et SPL	Service web	Cloud computing & SOA	Services web	Non	SOA	Composant de l'adaptation	Modèle de contexte	de
xml	Règles	Structure XML similaire	OWL-S mechanism	Tacklets & Service ontologie	Ontologie	Composant de l'adaptation	Modèle de contexte	de
non	oui	Non	Non	Non	Non	Composant de l'adaptation	Modèle de contexte	de

TAB. I.2 – Comparaison des systèmes sensibles au contexte

Discussions :

Structure de l'architecture – On définit deux modes de communication entre les couches : architecture en couche et distribuée. Dans le premier type la communication entre les couches est effectuée dans l'ordre (niveau par niveau). Par contre la seconde, la communication est aléatoire. La plupart des travaux entrepris dans ce domaine implémente leurs systèmes selon le premier type en raison de la facilité des fonctionnalités supplémentaires, la flexibilité et la fiabilité, c'est pour cette raison que notre proposition est basée sur ce type de modèle de communication.

Modélisation du contexte – L'ontologie a dominé l'aspect de la modélisation du contexte dans la plupart des propositions examinées. Les fonctionnalités supplémentaires sont renforcées par l'adoption d'ontologies telles que l'abstraction du contexte de haut niveau, le raisonnement puissant, l'interopérabilité sémantique (probablement) et la prise de la sensibilité du contexte avancée. L'ontologie est utilisée par les applications s'exécutant sur un PC de bureau ou un serveur à grande capacité. Par contre le modèle orienté objet utilisé par la plateforme CATS, est développé sur les applications mobiles. L'avantage de ce modèle réside dans la réaction plus vite aux changements du contexte. Dans le cadre de notre travail nous exploitons le modèle orienté objet (MOO) pour modéliser les éléments du contexte destiné à la santé mobile.

Relation entre les éléments du contexte – Elle interprète les informations contextuelles permettant l'amélioration de la prise de conscience et de l'intelligence de la plateforme. Comme illustré dans le tableau (**TAB. I.2**), la plupart des travaux cités auparavant ne tiennent pas en compte cette relation dans leurs conceptions de leurs systèmes. Dans le cadre de cette thèse, nous avons exploité cette relation pour combiner les différents paramètres caractérisant la situation en cours afin de démarrer le service en question dans les meilleures conditions.

Relation entre les services et le contexte – Elle intègre des éléments du contexte au niveau des services. L'implémentation de cette relation est une étape primaire dans l'adaptation des services et nécessite les différentes règles et plusieurs techniques tels que : Ontologie, XML, OWL-S. Notre proposition est basée sur cette relation pour la gestion des différents services développés dans la santé mobile.

I.4 Problématique de la conception d'une plateforme sensible au contexte pour les applications de la santé mobile

Avant d'entamer notre problématique, nous allons parcourir la présentation des différents éléments et aspects nécessaires à la conception d'une plateforme sensible au contexte, qui a pour rôle d'exécuter et de gérer les applications de la santé mobile.

I.4.1 Exécution de plusieurs applications sur Smartphones

Selon [54], il y a 44,7% des personnes qui utilisent 3 à 5 applications par jour, 25,3% exploitent deux applications et seulement 8,8% utilisent 12 applications voir plus.

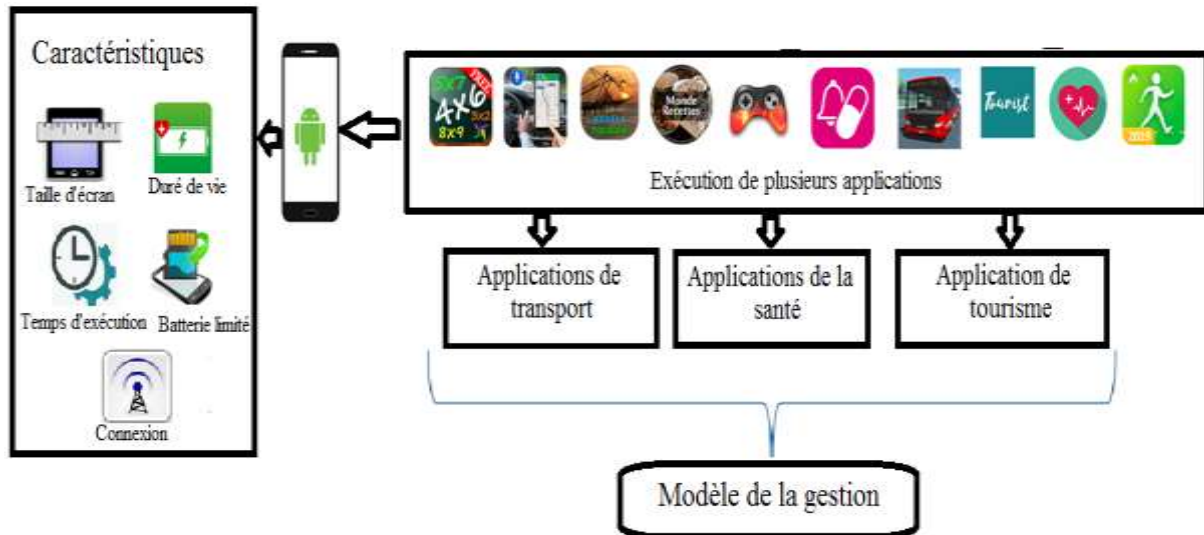


FIG. I.11 – Applications exploitées sur Smartphone

Cette utilisation énorme des Smartphones (FIG. I.11) lors de l'exécution des applications nécessite d'envisager des stratégies et méthodes pour la gestion de ces applications (objectif de notre travail).

I.4.2 Applications de la santé mobile

La santé mobile est un moyen de maximiser les informations liées aux patients afin d'aider le médecin dans son travail.

L'état de l'art présenté en [55] expose plusieurs applications de la santé mobile, notamment le suivi des personnes diabétiques, les patients cardiaques, le soin des personnes âgées et l'assistance des agents (le médecin, l'infirmière...).

Notre objectif est d'améliorer l'exploitation des applications mobiles de la santé tout en intégrant le contexte dans notre modèle. Ces applications doivent être s'adapter et connecter aux différents acteurs médicaux.

Tout au long de jour, les personnes exploitent ses terminaux mobiles (un téléphone, un ordinateur portable) en différentes manière, mais la plupart de ces utilisateur a besoin de plusieurs services permettent de l'aider dans leur vie de la santé, notamment la surveillance des données médicales (glycémie, ECG, EEG...), les formations de la santé (médicament, nourriture, Symptômes de maladies, etc.), et d'autres informations sur des entités à proximité (pharmacie, centre médical, hôpital, etc.). Ces terminaux ajoutent les différents avantages à ce domaine, malgré il existe de nombreuses contraintes de ces terminaux en termes de la capacité du traitement des données médicaux, de taux d'installation et d'exécution des applications et la durée de vie de la batterie.

Par exemple, pour mieux illustrer le type des applications que nous intéressons dans cette étude, prenant l'exemple suivant. Nous considérons trois personnes « *Mouad, Iyed et Mourad* » qui pratiquent leur vie quotidienne de façon normale et chacun possède un Smartphone. *Mouad* souffre d'une maladie diabétique et *Iyed* souffre d'une maladie cardiaque mais *Mourad* souffre des deux maladies (le diabète et le cardiaque). De plus, ces patients ont installé de nombreuses applications sur leurs Smartphones. Ils trouvent ces applications nécessaires afin d'améliorer leurs confort et palier leurs incapacités : le suivi de leurs états de santé 24h/24h, envoi d'alarme d'urgence, l'éducation...

Notre modèle proposé est testé selon deux pathologies (cardiaque et diabète) pour valider leur performance (fiabilité, efficacité et souplesse). Ces deux catégories sont choisies en raison de l'hétérogénéité dans le processus de leurs traitements.

I. 4.3 Présentation de notre problématique

A partir de l'exemple analysé auparavant, nous distinguons les problèmes suivants:

- Les données médicales sont caractérisées par : l'hétérogénéité, la quantité, la diversité ainsi que les différents modes de présentation. Il n'y a pas une identification claire de la relation entre les paramètres médicaux.
- L'adaptation selon la situation courante (les entrées contextuelles).
- Implémentation d'une stratégie permettant d'établir la dépendance entre les données médicales et les fonctionnalités nécessaires à la création des services mobiles.

Pour assurer le meilleur comportement de ces services, les fonctionnalités doivent être installées, démarrées et arrêtées selon nos besoins dans le cadre de ce travail. Donc, il n'existe pas une plateforme unifiée pour la gestion de ces fonctionnalités. Ceci rend intéressant d'avoir un seul prototype qui intègre plusieurs entités logicielles : analyser les données hétérogènes, évaluer et identifier le niveau de risque, adaptation au comportement (alerte selon la situation en cours), transfert immédiat des bilans d'analyse.

Dans notre travail, nous envisageons de développer une plateforme sensible au contexte, capable de gérer les applications liées au M-health. Le modèle proposé s'articule autour des mécanismes qui implémentent le processus de l'adaptation des fonctionnalités des applications et la gestion des informations contextuelles (collection des données, identification des relations entre les éléments du contexte).

Pour atteindre notre but, les travaux de cette thèse se focalisent sur les trois axes destinés à *l'identification et à la classification du contexte, la gestion du contexte et l'adaptation*.

I. 4.3.1 Contexte – Identification et classification

Selon [56], la plupart des travaux définit le contexte suivant leur axe de recherche. Les différentes définitions données par ces projets sont claires et simples pour bien préciser leurs objectifs ciblés. Dans le cadre de cette thèse, nous nous intéressons particulièrement aux applications de *M-health*. Pour cette raison, Il est donc nécessaire de définir la notion de contexte afin de couvrir toutes les informations le plus importantes dans l'exécution de ces applications.

Deux étapes doivent être implémentées pour exploiter la sensibilité de contexte dans le développement des applications et des services mobiles.

La première consiste à l'identification des éléments de contexte (*CE*) pour spécifier les contraintes des environnements mobiles. Les applications doivent être adaptées à ces types d'environnements. Comme nous l'avons vu précédemment, chaque projet intègre certains éléments dans leur conception suivants de leur besoin (**TAB. I.2**). Pour cela, nous allons identifier tous les éléments liés au domaine du *M-health* (chapitre II) afin de satisfaire les besoins de la plateforme proposée. La seconde est la classification de ces éléments par groupe pour nous faciliter la tâche de leurs gestions. Cette classification est effectuée via les caractéristiques des informations contextuelles et les domaines d'applications. De nombreuses classifications ont été proposées dans l'état de l'art mais elles ne sont pas appropriées à nos travaux, la raison pour laquelle notre classification est propre et personnelle.

I.4.3.2 Approche de la gestion de contexte

Le but de la gestion de contexte est d'extraire les informations de haut niveau via l'environnement d'utilisateur. Selon les travaux mentionnés dans la section I.3.1, il y a plusieurs approches pour gérer et modéliser ce contexte. De plus, chaque approche est basée sur les différentes techniques destinées à résoudre un ensemble de problèmes. Le processus de la gestion est une série des opérations effectuées pour obtenir une situation contextuelle plus

claire et simple. Dans ce cadre, nous envisageons une approche de la gestion selon le besoin de notre étude.

I.4.3.3 Mécanisme de l'adaptation au contexte

Le système sensible au contexte est basé sur un ensemble des processus pour assurer le meilleur fonctionnement des applications mobiles dans leur environnement qui est très hétérogène et évolutif. Parmi ces processus, il y a le mécanisme d'adaptation permettant de gérer ces applications suivant leurs contraintes d'exécution. Principalement, il existe plusieurs types d'adaptation : *l'adaptation de contenu*, *l'adaptation du comportement* (service) et *l'adaptation de présentation* (ou interface). Dans le cadre de notre travail, nous nous intéressons à implémenter un adaptateur consacré pour le comportement et la présentation. Généralement, le mécanisme d'adaptation du comportement est basé sur le déclenchement d'une version du service appropriée à l'évènement produit dans l'environnement d'exécution [43]. L'approche d'adaptation de présentation consiste à présenter les contenus d'une façon simple et claire.

Dans cette thèse, nous nous focalisons sur le développement des applications qui s'exécutent dans les environnements mobiles tout en respectant les contraintes imposées par ces types des terminaux (ressources limitées, faible calcul, capacité et résolution...).

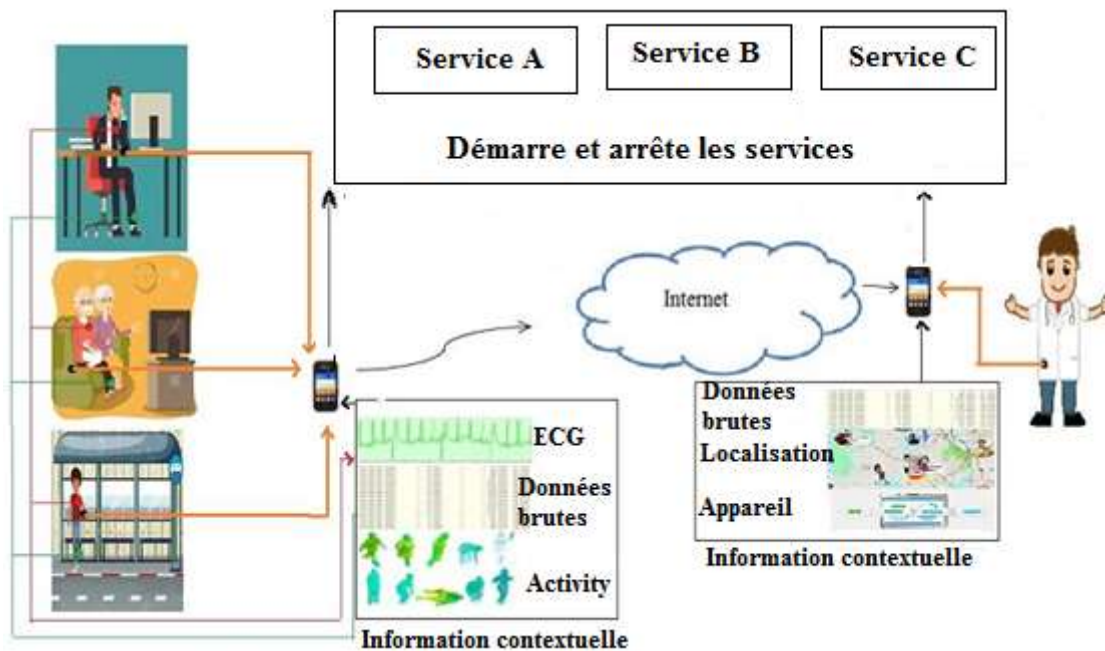


FIG. I.12 – Mécanisme d'adaptation sensible au contexte

Notre objectif est de marier les deux types d'adaptation pour une exploitation potentielle des services de la plateforme, comme illustré sur la figure (FIG. I.12). Le défi de cette proposition consiste à rendre les applications autonomes par rapport au contexte sans l'interruption d'exécution.

I.5 Notre contribution

La plupart des travaux montrent que le système sensible au contexte passe par plusieurs étapes pour atteindre leurs objectifs (Section I.3.1).

Tout d'abord, nous identifions les éléments de contexte pertinents pour la santé mobile. L'idée ici, c'est qu'en partant de ce constat, une fois les situations contextuelles sont découvertes via les relations entre ces éléments, nous procédons à l'adaptation des applications en fonction de

ces situations. Ce qui permet à ces applications de changer leur comportement et par conséquent d'être en exécution plus flexible.

Alors la conception de l'architecture de notre modèle proposé est basée sur une structure multicouche. L'avantage de cette architecture réside dans la constitution facile des fonctionnalités supplémentaires. Chaque couche implémente une fonctionnalité spécifique. De cette manière, le système est plus rapide et léger grâce à la répartition et la division des tâches complexes.

Dans cette thèse, nous proposons de concevoir une architecture de composantes modulaires tout en intégrant l'approche d'adaptation. Dans cette vision, beaucoup de travaux s'intéressent à la gestion et l'adaptation. La particularité de notre travail réside dans l'ajout d'une sous-couche entre ces deux points permettant la création des situations contextuelles.

En outre, l'architecture en couches permet aux développeurs de gérer facilement la construction du logiciel avec une compréhension conceptuelle intuitive et une distribution des fonctionnalités de manière hiérarchique.

Il existe plusieurs architectures modulaires tels que : CBSE (*Component-based software engineering*), SOA et la combinaison des deux SCA (*Service Component Architecture*). Le choix de type d'architecture dépend essentiellement de plusieurs paramètres [3] :

- *Réutilisation du code* : Plusieurs applications sont capables de remployer le même module.
- *Mise à jour plus facile* : La propriété de la division de code en modules permet de faciliter de mettre à jour (la partie de code qui doit être changé) ce dernier.
- *Flexibilité* : La décomposition d'une application facilite le changement de comportement de cette dernière.

Après avoir étudié les différents standards selon nos besoins bien sûr, nous avons opté pour le choix de l'architecture orientée service (SOA) en raison de la connexion automatique des services à l'exécution.

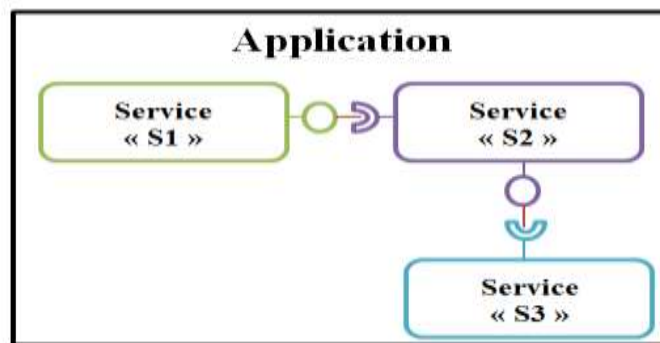


FIG. I.13 – Modèle de l'application à base des services

Pour assurer la flexibilité, nous nous focalisons sur les applications composées des services (**FIG. I.13**), comme proposées dans les travaux [57-59]. Cette architecture nous permet de résoudre les problèmes cités précédemment.

Avec plus de détail, une application doit être divisée de façon logique selon ses fonctionnalités et chaque fonctionnalité exprime un service particulier. Cet assemblage de plusieurs services forme une seule application comme illustré sur la figure (**FIG. I.13**). Dans cette cadre, nous avons pris en compte dans notre cas, les producteurs et les consommateurs locaux (sur la même machine), pour fournir une bonne implémentation dans toutes les conditions (déconnexion).

L'implémentation du modèle proposé pour la gestion des applications sensibles au contexte (TICs & santé) est divisée en deux parties :

I. 5.1 Implémentation sous Android

La première partie est réalisée sous *Java et SDK Androïde* [60] pour l'interface mobile d'accès, d'exécutions des packages des plateformes modulaires (OSGi [61]: *Open Service Gateway Initiative* et iPOJO [62]: *injected Plain Old Java Object*) et d'installer les bundles via différente sources. L'idée ici, est d'intégrer d'une part, le Framework OSGi / iPOJO sur Android et d'autre part, le déploiement des composants d'application au-dessus de ce Framework (**FIG. I.14**).

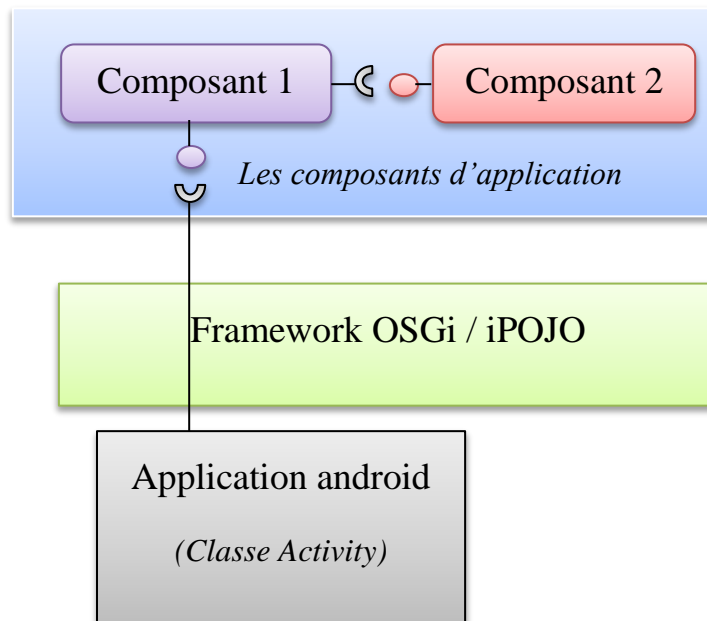


FIG. I.14 – Technologies OSGi / iPOJO sur android

I. 5.2 Utilisation OSGi et iPOJO sur les applications Android

Comme il a été mentionné auparavant, notre but est de concevoir des applications modulaires. Pour cela, nous avons exploité les deux plateformes OSGi et iPOJO en raison de la reconfiguration dynamique des comportements des applications sans les redémarrer et de la simplicité de développement grâce à l'injection de code non-fonctionnel au moment d'exécution et le management de dépendance de services.

Le problème réside dans la complexité d'intégration de ces deux technologies sur la plateforme Android.

Actuellement, de nombreux travaux de recherche se focalisent sur la compatibilité d'OSGi et d'Ipojo avec Android et sur les avantages que ces technologies peuvent ajouter à l'informatique mobile. Nous mentionnons ici quelques exemples de ces travaux. Dans [41], D.Popovici et al implémentent une plateforme d'exécution des applications accompagnant et assistant l'utilisateur dans les activités de transport (conduire, visiter une ville, etc.). Le travail présenté en [63] introduit une plateforme qui est capable de suivre l'état des applications industrielles. M. Khemaja et al [64] proposent une nouvelle architecture d'application adaptable et reconfigurable.

Notre préoccupation ne réside pas seulement dans la compatibilité des technologies (OSGi, iPOJO et Android). Mais nous nous intéressons aussi à la mise en place notre plateforme proposée grâce à l'exploitation de ces technologies. Alors, nous essayons d'implémenter tous

les modules de cette plateforme selon les capacités et les permissions, qui sont fournis par ces technologies.

I. 6 Conclusion

Ce chapitre présente un état de l'art sur les travaux liés au système sensible au contexte. Il s'agit de spécifier l'ensemble des processus nécessaires pour la création et le développement de ce type de système dont chaque projet est caractérisé par ses avantages et inconvénients dans les communautés scientifique et industriel. Nous nous focalisons sur la gestion du contexte et le mécanisme d'adaptation dans le domaine de santé mobile. Tout en étudiant le fonctionnement des applications sur les terminaux à faibles ressources. Ce qui permet de concevoir une plateforme adaptable, efficace et fiable aux environnements mobiles.

Le chapitre suivant sera consacré à une étude approfondie sur la notion du contexte en M-health en identifiant tous ses éléments liés à notre proposition et les classifiant selon leurs caractéristiques afin d'extraire les situations contextuelles.

Chapitre 2

Contexte lié au domaine de la santé mobile

On définit le contexte en général comme étant les informations qui lient à une activité et les informations additionnelles sur son environnement d'exécution. L'emploi de cette notion dans le développement des systèmes informatique permet d'assurer les exigences et les besoins des utilisateurs via leur contexte. Il concerne l'ensemble des informations entourées leurs environnements pour se comprendre leurs comportements. La compréhension de ce terme est nécessaire pour mieux l'exploité dans la conception et l'implémentation des systèmes informatique.

Un tel système sensible au contexte effectue plusieurs étapes afin d'atteindre leurs objectifs ciblé. Ce qui nécessite une modélisation basée sur la création des informations contextuelle abstraite en termes de la structure de données et de la sémantique.

Le but alors de ce chapitre est d'étudier le concept du contexte en système informatique mobile et d'identifier ses éléments influençant l'exécution des applications et des services de santé mobile. Une classification de ces éléments sous forme d'un arbre sera détaillée par la suite tout en effectuant une comparaison entre les modèles de contexte existants.

II. 1 Contexte en système informatique

Ces dernières années, l'idée du contexte est devenue très courante dans les travaux scientifiques. Il considère les informations existées dans l'environnement d'un système informatique ou les critères qui spécifient un événement influencé sur le fonctionnement de ce système, comme schématisé dans la figure (FIG. II.1). Pour cela, il est indispensable de définir l'ensemble de ces paramètres qui doivent être étudiées, analysées et traitées.

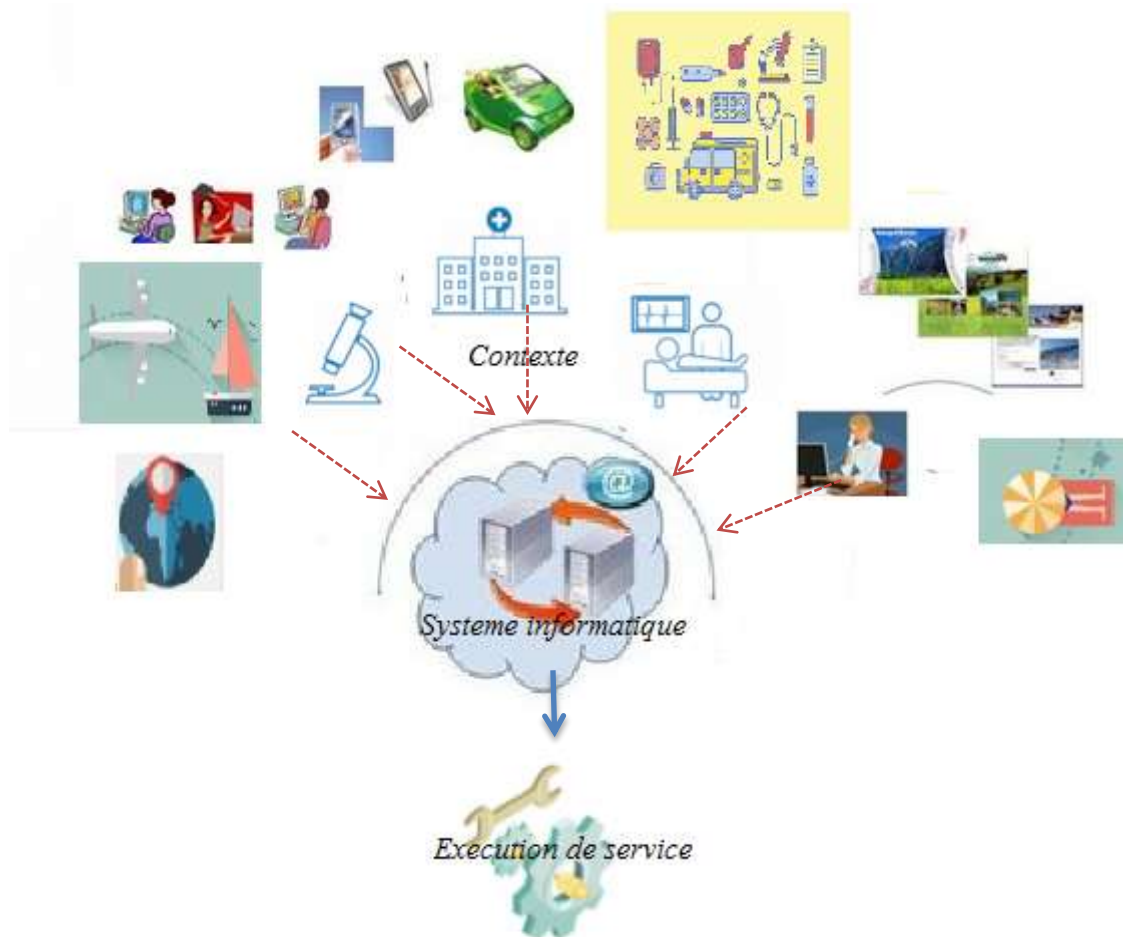


FIG. II.1 - Contexte en système informatique

De nombreux chercheurs tentent de définir le contexte selon leurs buts ciblés et leurs environnements d'exécution. Quelques chercheurs [65-70] présentent seulement l'énumération des informations contextuelles. Par contre d'autres travaux donnent des définitions exhaustives. Par exemple, A K, Dey et al [5] introduisent le contexte comme toute information qui peut être exploitée pour caractériser une situation d'entités (personne, lieu ou objet). Pour M. Moeiz et al [43], le contexte est toute information permettant de démarrer ou modifier (la forme) un service. Selon L. Xin et al [56] le contexte est un ensemble d'information pour représenter des changements de circonstances (statiques ou dynamiques). En adoptant un point de vue holistique sur les définitions mentionnées, nous pouvons constater que la plupart d'entre elles s'articulent uniquement sur une application particulière ou simplement énumèrent les entités en fonction du contexte. En d'autres termes, ils manquent de standardisation. Dans les sections suivantes, nous présenterons notre vision pour définir le contexte en précisant l'exploitation de ce terme dans le cadre de nos travaux.

II. 2 Contexte en santé mobile

Le but général de santé mobile est de fournir des services mobiles appropriés aux patients selon les informations contextuelles et sans intervention explicite des utilisateurs (**FIG. II.2**). Cette adaptation est réalisée via le changement de cycle de vie des services (démarrer, arrêter, installer et désinstaller). Ce qui permet de faire l'abstraction du contexte et de le spécifier selon les services destinés à la santé mobile.

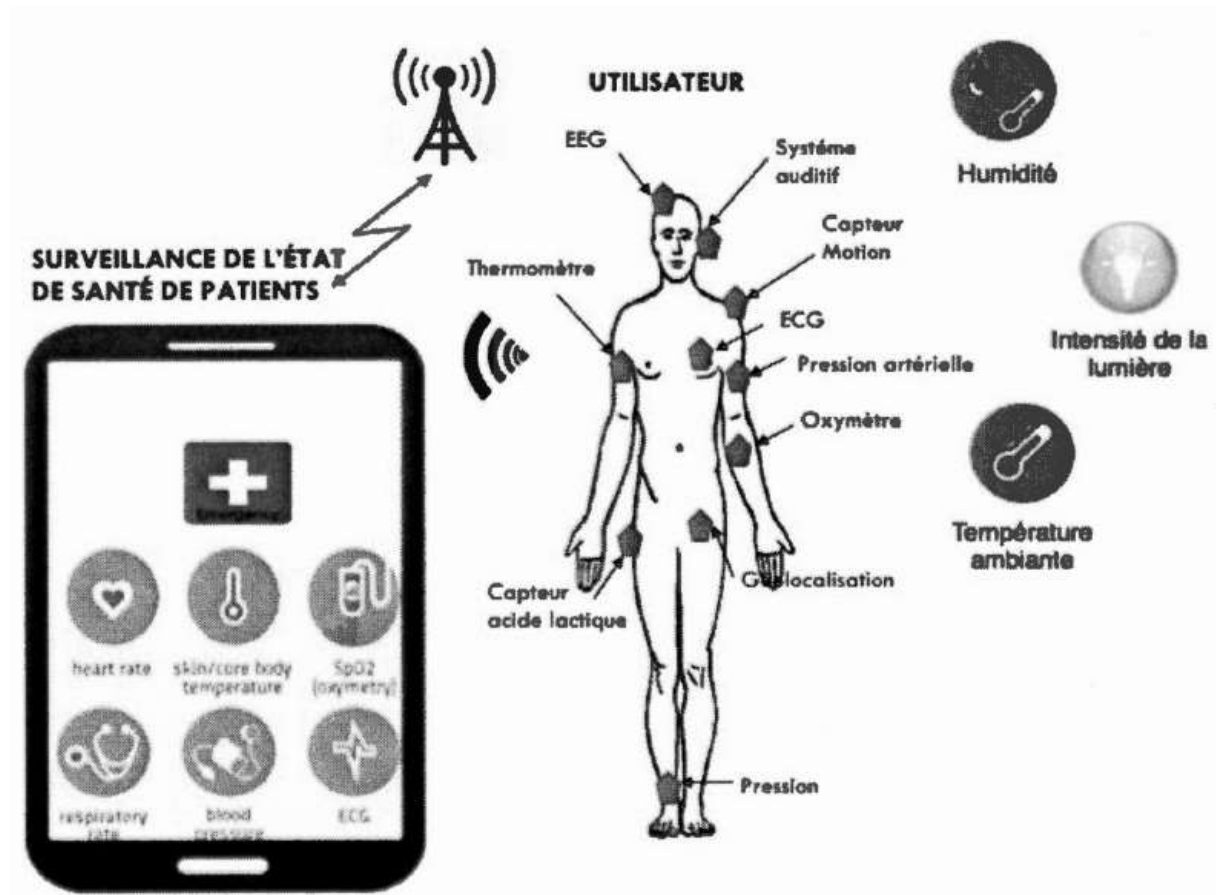


FIG. II.2 - Contexte en santé mobile

Après les définitions citées par les différents chercheurs, nous constatons que la proposition de dey [5] explique le contexte de façon générale. Cette généralisation donne au contexte un avantage d'être proportionnel à plusieurs applications. Pour cette raison, nos travaux s'articulent autour de cette dernière pour comprendre ce concept en santé mobile.

II. 2.1 Identification des éléments de contexte

Plusieurs chercheurs ont exploité le terme du contexte dans leurs travaux, comme mentionné dans l'état d'art [56]. L'objectif de cette utilisation réside dans l'amélioration de l'expérience d'utilisateur afin d'offrir les services pertinents par rapport à sa tâche [5].

Pour cela, il est nécessaire d'intégrer les informations de contexte durant l'exécution des applications pour optimiser la qualité de service.

On peut considérer ces informations contextuelles comme un ensemble des éléments, dont chaque élément représente un défi d'exécution des applications de M-Health.

Certains travaux ont spécifié les éléments du contexte en santé, l'idée qu'en partant de ce constat pour quoi ne pas identifier et exploiter ces éléments dans la conception et développement des services de M-health. Ce qui permet d'implémenter des systèmes intelligents et autonomes issus de l'intelligence artificielle. La figure (FIG. II.3) représente les éléments visés dans nos travaux.

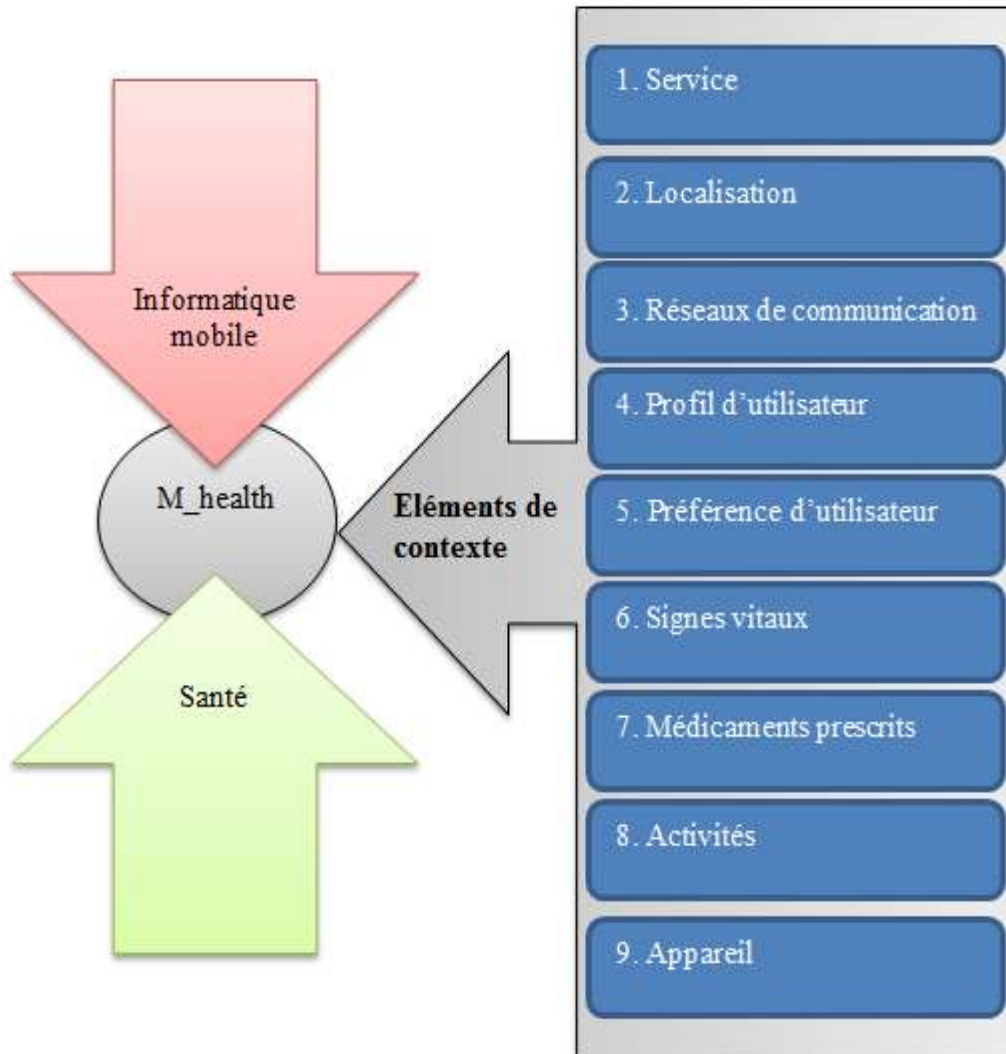


FIG. II.3 – *Eléments de contexte*

II. 2.1.1 Contexte de service

Les applications et les services doivent être adaptés en fonction de la situation contextuelle courante. Ce qui implique d'exploiter des fonctionnalités adéquates. Donc, il est important d'identifier la description de service en question pour atteindre la tâche appropriée. Cette description permet non seulement, de connaître son disponibilité ou non mais aussi d'assurer son adaptabilité.

II. 2.1.2 Localisation

Puisque nous nous intéressons sur la mobilité d'utilisateur, il est nécessaire de spécifier la position de l'utilisateur, ce qui peut être obtenue par les différentes technologies (GPS, WIFI, ...).

II. 2.1.3 Réseaux de communication

Cet élément comporte des informations liées à la communication, notamment la connectivité, la bande passante, le protocole et le flux. Une spécificité de ce type d'élément réside principalement dans l'utilisation de deux modes: avec ou sans infrastructure. Cet élément joue un rôle crucial en domaine de santé mobile, ce qui nécessite d'être évalué périodiquement.

II. 2.1.4 Profil d'utilisateur

Il fournit toutes les informations d'utilisateur : le nom, l'âge, le sexe et le type d'utilisateur. Dans notre travail, il est indispensable de préciser le type d'utilisateur (patient ou médecin) car chaque type comporte un ensemble de services appropriés.

II. 2.1.5 Préférence d'utilisateur

Les applications devraient également permettre de configurer les préférences d'utilisateur afin de personnaliser les résultats qu'elles génèrent. Ce paramètre influence l'interaction d'utilisateur avec l'application.

II. 2.1.6 Signes vitaux

Cet élément offre les différentes informations permettant d'évaluer la situation médicale du patient. Ces informations peuvent être représenté non seulement de différentes façons (un signal ECG ou une seule valeur) mais aussi de les exploiter de différentes manières (envoyer directement ou stocker dans une base de données intermédiaire). Donc, le traitement de telle information est spécifique dans le cas de chaque patient.

II. 2.1.7 Médicaments prescrits

Ce paramètre répertorie les médicaments qui sont pris par le patient en spécifiant l'heure à laquelle il les prend. Il peut être caractérisé par les informations suivantes : le nom, le temps de prise et la dose pris de médicament.

II. 2.1.8 Activités

Il présente des activités qui sont effectués par l'utilisateur en fonction de temps, par exemple la marche, la course, l'effort personnel...

II. 2.1.9 Contexte d'appareil

Cet élément consiste à définir les caractéristiques d'un dispositif. Ces informations permettent de décrire les périphériques en deux façons : le système d'exploitation et les plateformes logicielles existantes comme Java et android, ainsi que le hard qu'est défini par les interfaces d'entrée/sortie, l'espace de la mémoire et le niveau de la batterie.

Après avoir identifié les principaux éléments du contexte exploités dans notre simulation, nous présenterons par la suite notre méthodologie de leurs classifications pour une utilisation fiable et efficace de ces éléments dans une telle plateforme d'exécution des services.

II. 2.2 Classification des éléments

Avant d'entamer notre classification, il faut passer par définir les relations entre les éléments cités auparavant. Pour cela, ces éléments regroupent plusieurs tâches communes tels que : le démarrage et l'arrêt des mêmes moniteurs (évaluateurs des éléments). L'exécution consécutive de ces moniteurs permet de générer une seule situation contextuelle plus

sémantique (Sc_{ij}). Cette situation décrit la relation entre les éléments de la même catégorie (même pathologie).

En raison de la diversité, de l'hétérogénéité et de la qualité des informations contextuelles, il est suggéré de classer ou de catégoriser ces informations afin de faciliter la stratégie d'adaptation.

Plusieurs catégorisations ont été proposées, par exemple, M. Moeiz [71] propose deux classes. La première contient des paramètres qui déclenchent un service et la deuxième change la forme d'un service. Les auteurs de [72] réalisent deux sortes de classification dynamique et statique. D'autre proposition par [73] consiste à implémenter plusieurs classes qui regroupent le contexte physique d'une personne (lieu, heure...), le contexte social (relations sociales...) et le contexte d'activité (regarder la télévision, écouter la musique, appeler quelqu'un au téléphone...).

La plupart des classifications réalisées ne couvrent pas tous les domaines. Elles ciblent beaucoup plus l'axe d'étude en question.

Dans le cadre de notre travail, la classification effectuée est orientée vers la santé mobile.

II. 2.2.1 Proposition d'une classification

A travers les relations établies entre ces éléments, nous suggérons une classification schématisée sous forme de l'arbre ci-dessous (FIG. II.4). Cette dernière comprend les catégories suivantes:

Contexte physique: Vu que l'utilisateur est mobile, la collecte de la totalité des informations doit prendre en considération le déplacement de l'utilisateur. Cette classe est fondée sur le contexte qui englobe la localisation et les réseaux de la communication.

Contexte d'utilisateur : Cette classe est la catégorie qui regroupe des informations personnelles. Autrement dit, les applications sont adaptées selon les besoins de l'utilisateur (son profil et ses préférences).

Contexte d'environnement de santé : les applications de la santé mobile sont conçues dans le but de surveiller la situation médicale des patients. Dans ce cadre, les informations médicales deviennent un outil indispensable pour le suivi médical pertinent. Autour des données environnementales de la santé gravitent des éléments primordiaux: signes vitaux, médicaments prescrits et activités.

Le contexte de l'appareil et celui de service forment deux classes différentes ; chacune se caractérise par des aspects spécifiques clairement détaillés dans la section II.2.1.

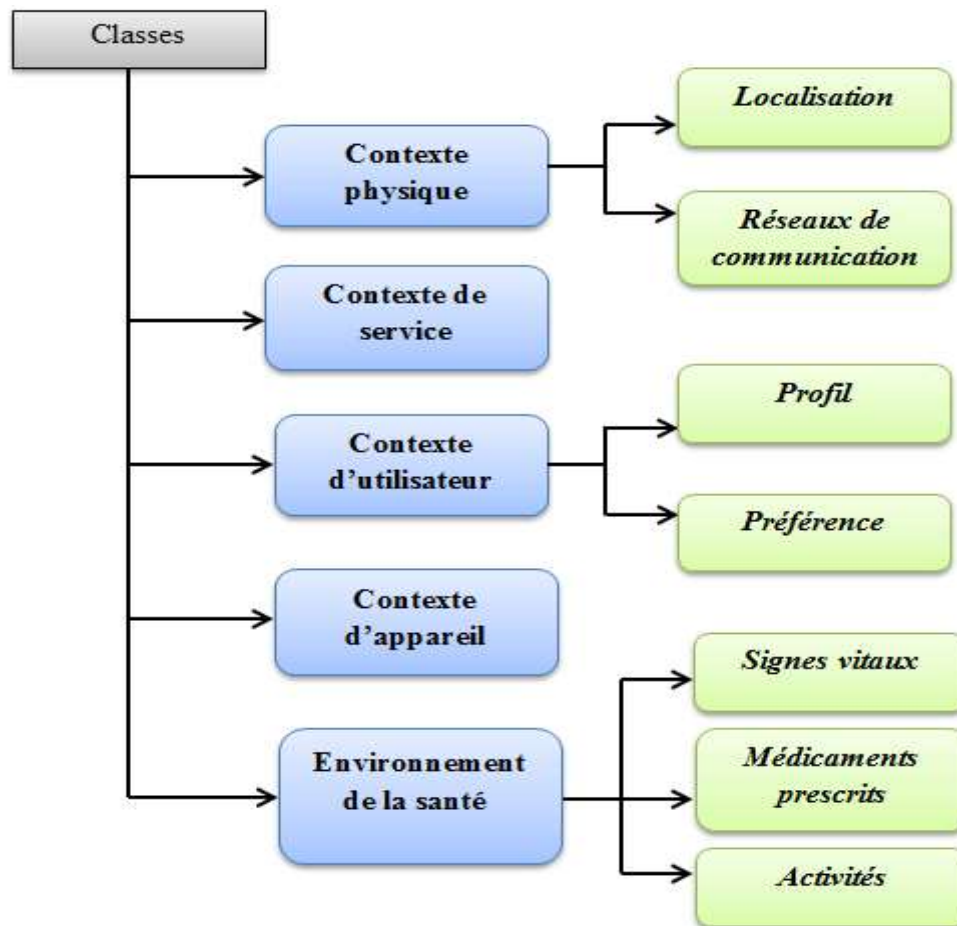


FIG. II.4 – Arbre des éléments de contexte

Cette classification est à la fois complète et expressive: d'une part, elle inclut tous les paramètres influençant l'exécution des applications de la santé mobile. Ce qui facilite le processus d'adaptation au contexte. D'autre part, elle se distingue par une organisation claire et cohérente; en d'autres termes, chaque classe contient des éléments ayant une relation bien précise. A ce niveau, nous citons le cas d'un patient hypertendu, une modification de la valeur du contexte de l'activité peut avoir une répercussion sur la valeur de ses signes vitaux, à l'instar de l'augmentation de sa pression artérielle. Le gestionnaire du contexte s'appuie sur les relations précédentes afin de simplifier son tâche.

Après avoir suggéré cette catégorisation des éléments nécessaires, nous étudions par la suite les différents modèles de contexte simplificateurs et facilitateurs de processus de l'exploitation de ces éléments dans le cadre de développement des applications mobiles.

II. 3 Modélisation du contexte

Pour proposer la modélisation du contexte adéquate, nous analysons et comparons les approches exploitées dans les travaux scientifiques. Ces approches ont été définies dans la littérature [74].

II. 3.1 Modèles de contexte

Les éléments contextuels doivent être adéquatement exprimés et modélisés pour favoriser leur exploitation dans le système informatique tout en fournissant un haut niveau d'abstraction qui facilite les mécanismes d'adaptation.

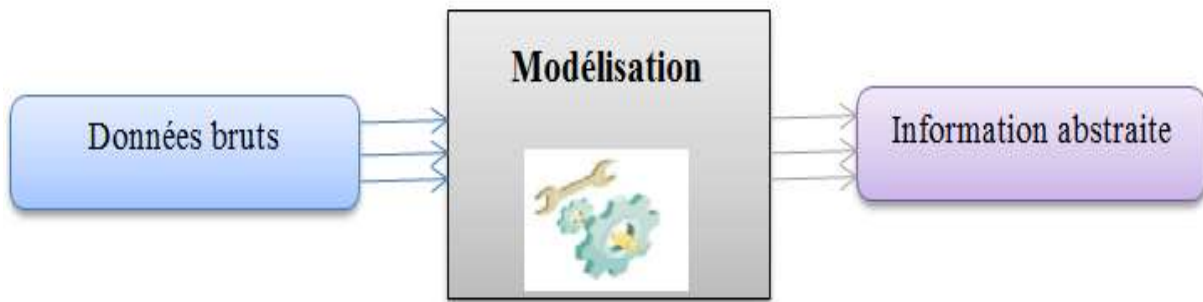


FIG. II.5 – *Objectif de la modélisation de contexte*

La figure (FIG. II.5) dépeint les principes de cette modélisation qui sont la création et l'interprétation représentées selon une forme abstraite en termes de la structure de données et de la sémantique.

II. 3.1.1 Modèle clé-valeur

Dans ce modèle, la structure des données est très simple et claire (où la clé est le contexte et la valeur est l'information détectée correspondante), comme représenté dans la figure (FIG. II.6).



FIG. II.6 – *Structure des données clé-valeur*

En raison de la simplicité et la facilité de son utilisation, il est exploité dans plusieurs travaux de recherches par exemple Dey [75], Yamabe [76] et Schilit [77]. Cependant, il lui manque les capacités de la structuration complexe, ce qui rend très difficile la récupération de ces données. Il ne supporte pas l'interopérabilité et la représentation de la relation entre les informations contextuelles.

II. 3.1.2 Modèle à balise

Ce modèle offre une présentation hiérarchique des données (FIG. II.7) avec des balises structurées, des attributs et des contenus. Les auteurs de ces travaux [78-80] se sont basés sur le langage XML générique pour mettre en œuvre la modélisation du contexte.

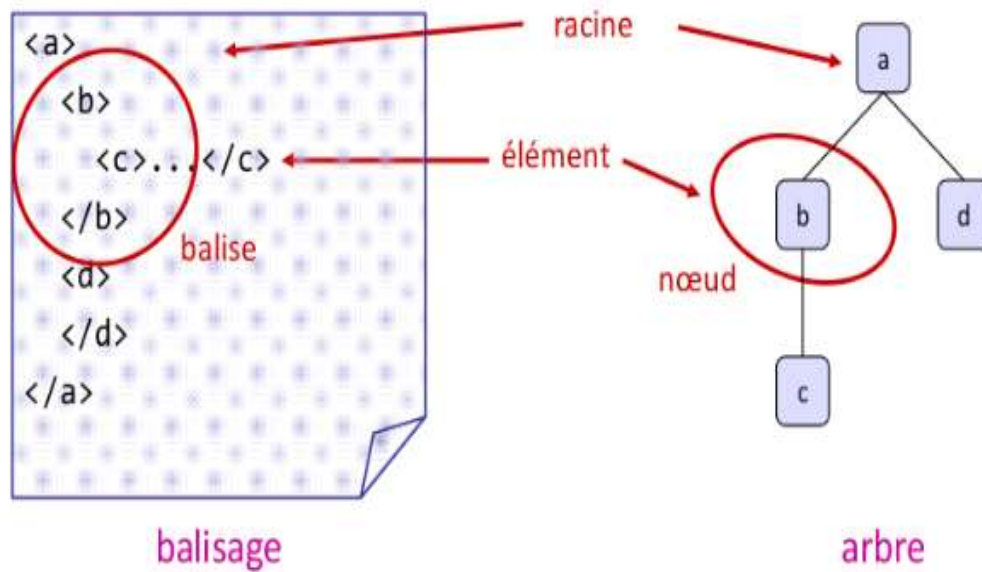


FIG. II.7 – Structure hiérarchique [81]

La limite de ce modèle réside dans sa structure hiérarchique qui doit être prédéfinie. Donc ce modèle ne peut pas modéliser le contexte qui change dynamiquement.

II. 3.1.3 Modèle graphique

Les diagrammes graphiques présentés par ce modèle peuvent identifier des relations. Il y a trois modes issus de ce modèle: le langage de modélisation unifié (UML), le modèle de relation d'entité (ERM) et le modèle de rôle d'objet (ORM) [82]. UML est un langage normalisé qui permet d'exprimer les différents types des informations contextuelles dans une notation graphique propre, tandis qu'ERM et ORM fonctionnent dans le cadre de la conception et l'interrogation de base de données au niveau conceptuel. Cependant, l'interopérabilité entre les différentes bases de données de stockage qui sont exploitées en bas niveau réel du modèle graphique expose un défi.

II. 3.1.4 Modèle orienté objet

C'est la modélisation qui permet de bénéficier de tous les avantages de l'orientation de l'objet (l'encapsulation (FIG. II.8), la réutilisabilité, l'héritage) [83]. Les objets sont exploités pour représenter les différentes informations contextuelles (la température, l'emplacement, etc.).

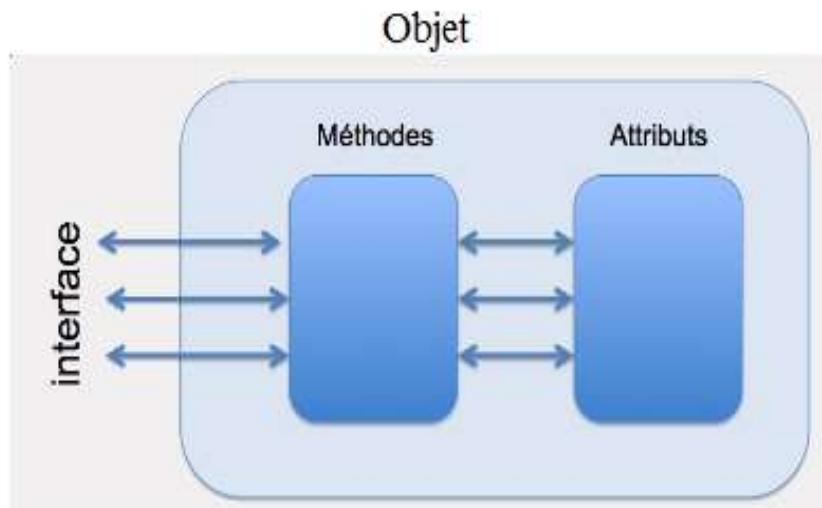


FIG. II.8 – Illustration du principe d'encapsulation [84]

Les instances peuvent accéder au contexte grâce aux mécanismes d'héritage. Le composant de base est appelé l'entité et forme l'objet des informations contextuelles modélisées. Une entité est liée à d'autres entités au moyen d'attributs qui sont également appelés l'association.

II. 3.1.5 Modèle basé sur la logique

Dans un modèle logique, le contexte doit être représenté sous forme formelle notamment les faits, les expressions et les règles. Un système logique est capable d'ajouter, de mettre à jour ou de supprimer de nouveaux faits. Les modèles basés sur la logique ont une capacité très élevée de formalité. Le processus d'inférence (appelé aussi raisonnement) peut être utilisé pour dériver de nouveaux faits en fonction des règles existantes dans les systèmes. Cependant, les normes et les outils de validation se font toujours par défaut.

II. 3.1.6 Ontologie

Studer [85] a défini l'ontologie comme "*une spécification formelle et explicite d'une conceptualisation partagée*" dans le domaine sémantique. Les ontologies représentent une description des concepts et des relations. Cette méthode de modélisation est considérée comme la méthode la plus prometteuse dans [86, 87], et elle peut aborder la confusion conceptuelle entre les personnes et les systèmes parce qu'elle partage la compréhension commune.

Après avoir analysé et mentionné les différentes modélisations du contexte, nous introduisons dans le tableau (TAB. II.1) une étude comparative entre ces modèles en présentant les avantages et les inconvénients de chaque modèle afin de définir le modèle approprié pour modéliser les éléments identifiés dans notre travail :

Modèle de contexte	Avantage	Désavantage	Applicabilité
clef-valeur	Simple; Facilité d'utilisation; Flexible.	Manque de normes; Ne peut pas représenter des relations; Difficultés en matière de la récupération des	Suffisant pour modéliser le contexte limité et auto-indépendant les applications qui n'ont pas besoin de partager

		informations ; Manque des outils de validation; Défaillance en termes d'évolution mis à part l'appariement exact ;	des données avec d'autres applications. Exemples: [19,88]
à balises	Structuré; Certains outils de validation sont disponibles; Flexible.	Insuffisance des normes; Problèmes dans la capture des relations; Intemporalité; Dépendances; Vérification d'incohérence; Raisonnement et incertitude.	Efficace comme mode de transfert de données à propos du contexte peu profond sur réseau; Applications dans lesquelles les niveaux d'information sont peu nombreux. Exemples: [89,90]
graphique	Richesse d'expression ; Les relations sont permises ; La validation est possible à travers des contraintes; Différentes normes et implémentations sont disponibles.	L'interopérabilité non résolue ; La configuration doit être requise ;	Particulièrement applicable pour dériver un modèle ER qui est utile en tant qu'instrument structurant pour une base de données relationnelle. Exemples: [91,92]
orienté objet	Les relations sont permises; Certains outils de développement sont disponibles; Peut-être fusionnable en utilisant des langages de programmation.	Manque des normes; Manque de la validation ; Lacunes liées à la récupération des informations ; Le raisonnement est insupportable.	Possibilité d'intégration dans les applications basées sur le code avec une capacité de calcul élevée. Exemples: [83-93]
basé sur la logique	Richesse d'expression Soutien du raisonnement; Contrôle de cohérence; Simplicité;	Déficit au niveau des normes; Manque de validation.	appropriation aux applications dans lesquelles des informations de haut niveau sont nécessaires et les

	Disponibilité des outils de traitement.		développeurs sont prêts à spécifier des contraintes. Exemples: [94 -96]
Ontologie	Soutenir le raisonnement; La richesse d'expression ; Les relations sont permises ; Forte la validation; Les outils de traitement disponible; Les normes matures; L'interopérabilité	La représentation peut être compliquée. Il sera complexe de récupérer les informations contextuelles; Impossible de traitement l'incertitude.	pour les applications qui sont dans le plus besoin d'échanger les informations avec d'autres; les compétences suffisantes relatives à l'ingénierie des connaissances sont disponibles. Exemples: [85, 97, 98, 99]

TAB. II.1 – Comparaison entre les modèles de contexte

Selon le tableau ci-dessus (**TAB. II.1**), il semble évident que chacun de ces modèles est applicable dans un certain mode des applications. Parmi des modèles précédents, certains nécessitent des dispositifs puissants en termes de capacité, stockage...

En effet, ils se sont basés sur les expressions des raisonnements et des connaissances visant la modélisation de ces informations.

Nous exposons par la suite nos deux approches : la première consiste à modéliser l'arbre de la catégorisation (**FIG. II.4**) et la seconde s'oriente vers la modélisation des éléments du contexte. Le but de ces approches réside dans l'amélioration de l'exploitation de ces éléments dans les ressources limitées.

II. 3.2 Notre approche de la modélisation

Nous avons précédemment représenté l'aperçu général du contexte en prenant particulièrement ce concept dans l'optique santé mobile. Vu que la modélisation du contexte est une étape essentielle pour développer le système de la gestion d'exécution des applications; notre approche s'articule autour de deux parties fondamentales :

II. 3.2.1 Modélisation de la classification

Cette modélisation vise l'extraction des relations qui se manifeste entre les éléments contextuels. Ces relations peuvent être explicitement exploitées dans la stratégie de la gestion de contexte. Notre approche s'articule autour d'un modèle à balise (XML). Loin d'être aléatoire, notre choix se fonde sur l'originalité du sujet sélectionné, la clarté de la structure de données, le format libre, la simplicité et la facilité de la lecture et de la programmation ainsi que leur capacité de stockage des informations. La figure (**FIG. II.9**) représente un extrait du document XML qui montre la modélisation de l'arbre des éléments (**FIG. II.4**).

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ContextElement>
  <C1 name="HEcontext">
    <CE11 name="signs vital"/>
    <CE12 name="Activity"/>
    <CE13 name="Médécale décrits"/>
  </C1>

  <C2 name="Device">
    <CE21 name="handler"/>
    <CE22 name="software"/>
  </C2>

  <C3 name="Physique">
    <CE31 name="Network"/>
    <CE32 name="Localisation"/>
  </C3>

  <C4 name="User">
    <CE41 name="Profil"/>
    <CE42 name="User Preference"/>
  </C4>
</ContextElement>
    
```

FIG. II.9 – Modélisation d’arbre des éléments de contexte

Elle permet de réaliser une représentation très simple, claire et cohérente aux dépendances entre les éléments du contexte. Ces relations ne doivent pas être rétablies avant chaque opération, l’argument solide qui nous mène à confirmer que le modèle XML est le plus adéquat pour modéliser notre classification proposée.

II. 3.2.2 Modélisation des éléments du contexte

Les éléments de contexte expriment des informations liées aux environnements d’exécution des applications. Pour cela, ces éléments doivent être représentés sous forme compréhensible afin de les exploiter dans l’adaptation des applications.

Nous choisissons le modèle orienté objet pour modéliser les entités définissant le contexte. Ce dernier dont le contexte est évalué au préalable rend les réactions plus rapides où le contexte a été évalué. Cette modélisation nécessite plusieurs paramètres pour interpréter plus clairement ces éléments. Le tableau suivant (TAB. II.2) récapitule ces paramètres.

Paramètre	Définition
Nom	Nom d’un élément de contexte.
Type	Type d’information de cet élément.
Valeur	Valeur ou ensemble des valeurs de cet élément.
Classe	Classe à laquelle appartient cet élément.

TAB. II.2 – Paramètres d’un élément de contexte

Chaque élément se caractérise par un ensemble des paramètres. Parmi lesquelles, il y a la valeur, qu’est donnée à un certain moment.

Par ailleurs, il existe d'autre paramètre appelé *Nom* permettant d'identifier l'élément de contexte. Nous distinguons les différents *types* d'information contextuelle:

Discret – Pour ce type, un élément prend une seule valeur qui est exprimée par un nombre réel.

Continu un élément prend un ensemble des valeurs, qui ne peuvent être représentées que par des intervalles de nombres réels.

Booléen ce type est simple. La valeur de celui-ci varie entre vrai ou false. La valeur est égale à vrai si la ressource est disponible.

String nous classons qu'un élément est de type String où la valeur prend une chaîne de caractères.

Le dernier paramètre de l'élément s'identifie par la *classe*, qui peut spécifier les relations avec les autres éléments. Pour mieux expliquer ces paramètres, le tableau (TAB. II.3) illustre les exemples des éléments pour chaque type.

<i>CE</i>	CE1	CE2	CE3
<i>Nom</i>	ECG	Glycémie	Activité
<i>Type</i>	Continue	Discret	String
<i>Valeur</i>	ECG1.txt	140	Travail
<i>Class</i>	Health environment	Health environment	Health environment

TAB. II.3 – *Éléments de contexte de chaque type*

À travers du tableau (TAB. II.3), CE1 présente un élément : *Signal ECG*, de type *continu*. Le signal ECG est représenté par un ensemble de nombres réels. C'est pour ça qu'on doit manipuler ce paramètre sous forme d'un fichier texte contenant tous ses valeurs. Toutes les applications qui exploitent ce genre des informations doivent s'exécuter quand tous les nombres collectés indiquent la possibilité de leurs utilisations.

Pour cela, il est nécessaire d'implémenter un algorithme de l'adaptation qui permet de gérer l'exécution selon les ressources disponibles du terminal mobile.

Comme nous l'avons représenté auparavant dans le tableau ci-dessus (TAB. II.3), l'élément CE2 porte le nom *glycémie*. Sa valeur est définie par un nombre naturel. Pour ceci, le type de cette valeur est *discret*. Les applications basées sur ce genre des paramètres sont plus simples en termes d'exécution. Malgré ce dernier, il faut mettre en œuvre le processus d'adaptation pour une meilleure exploitation des ressources limitées avec d'évaluation correcte de la situation médicale.

Le tableau précédent (TAB. II.3) mentionne que CE3 représente les activités de l'utilisateur. Sa valeur est introduite par une chaîne de caractères par exemple "*marche*", "*mange*" ou "*travail*". Les applications utilisent cet élément pour évaluer et analyser la situation médicale des patients.

La plupart des travaux liés à la modélisation des éléments issus de domaine de la santé sont basés sur l'extraction des nouvelles connaissances [100] ou les raisonnements logiques [71], ce qui nécessite des appareils à grande capacité. La complexité de la modélisation de contexte ralentit le temps de la réaction et de l'adaptation. Par conséquent, les applications deviennent moins qualitatives. À cet effet, nous ciblons la mise en place d'une modélisation à la fois simple et compréhensible pour une meilleure performance des ressources limitées (Smartphone). L'avantage le plus important du modèle simple réside dans l'efficacité de modéliser tous les types des valeurs, en tenant compte les ressources limitées des terminaux mobiles ainsi que la grande quantité et la diversité des données. L'utilisation des terminaux mobiles tels que des Smartphones impose la simplicité de contexte afin d'assurer des réactions rapides.

II. 4 Insertion du contexte dans l'application

Comme nous l'avons représenté auparavant, l'implémentation de notre plateforme est conditionnée par deux modèles basiques. Le premier modèle exprime l'arbre des éléments de contexte, ce qui facilite la gestion de contexte. Alors que le deuxième consiste à modéliser les éléments de contexte garantissant la gestion d'exécution des applications.

Nous avons étudié les différents axes liés au concept du contexte dans le cadre des applications de la santé mobile.

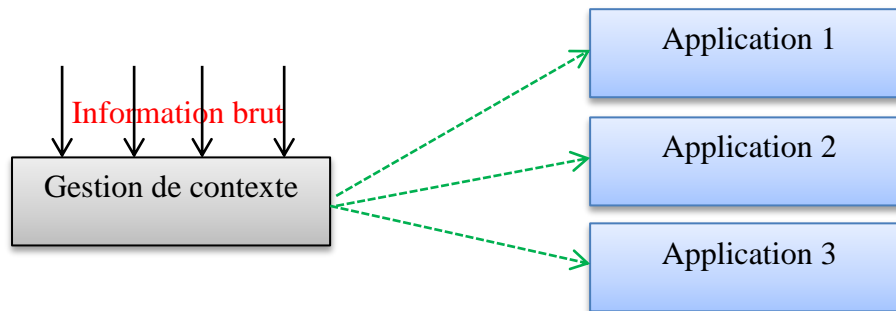


FIG. II.10 – *Gestion de contexte externe*

Parfois, ces applications ont besoin de s'arrêter ou démarrer en fonction des changements des situations d'environnement d'exécution. Pour cela, nous préférons d'implémenter la gestion de contexte en dehors des applications (**FIG. II.10**). Ce qui permet de définir génériquement le contexte pour ajouter les éléments selon le besoin.

Récupération des valeurs des éléments du contexte :

Nous avons introduit l'approche de la modélisation qui est une étape clé pour développer les applications mobiles. Mais il y a un autre défi que nous devons prendre en considération, la récupération des informations. Il existe de différentes méthodes pour la récupération des informations, l'exemple est clairement mentionné dans l'état de l'art [74]. H. Chen [101] représente trois façons pour l'exploitation des informations: *l'accès direct aux capteurs, l'infrastructure de middleware et le serveur de contexte.*



FIG. II.11 – *Code non fonctionnel dépendant aux capteurs*

Le problème de l'accès direct aux capteurs réside dans la flexibilité des applications qui peuvent être peu flexible. En effet, ces dernières doivent intégrer le code non fonctionnel dépendant aux capteurs (**FIG. II.11**). En revanche, les serveurs de contexte ont besoin de tous les manipulateurs de l'information, ce qui conduit à séparer les applications par rapport les

serveurs. Pour cela, il est indispensable d'établir la connexion pour récupérer les valeurs [34], comme illustré par la figure (FIG. II.12).

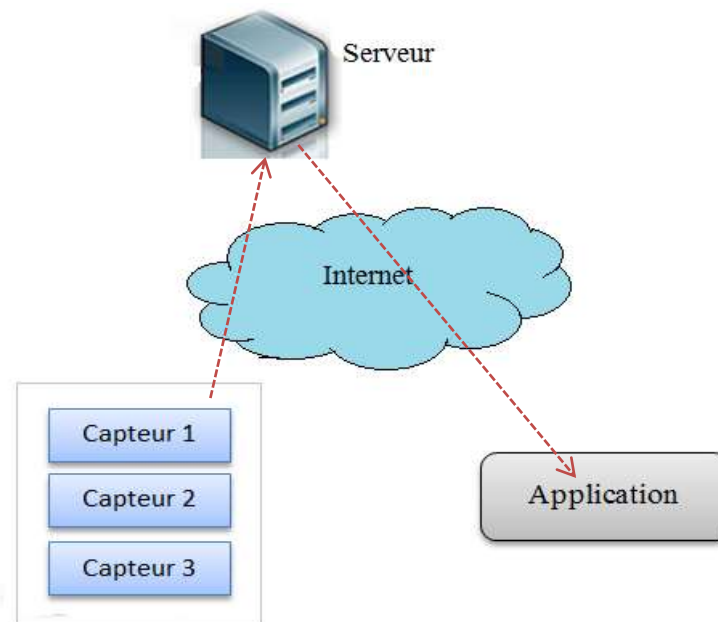


FIG. II.12 – *Serveur de contexte*

Nous focalisons sur l'autonomie du terminal par rapport aux objets externes, en exploitant les applications modulaires et découplées du contexte [72]. Cette dernière est une meilleure solution dans le cadre de notre travail.

Nous avons cité auparavant que notre particularité réside dans l'utilisation des relations entre les éléments de contexte pour évaluer et gérer ces éléments ainsi que la création de la nouvelle situation contextuelle. Par ailleurs, en ce qui concerne les informations contextuelles, nous suggérons qu'il est indispensable d'exploiter les modules légers permettant de gérer et de récupérer ces informations à l'aide d'un modèle de contexte proposé, sans oublier les relations entre ces informations. Ces modules permettent de gérer et de manipuler de manière indépendante le contexte par rapport aux applications

II. 5 Conclusion

Dans ce chapitre, nous avons exposé un aperçu général de terme du contexte dans le système informatique. De ce fait, nous avons étudié les différents axes liés aux informations contextuelles notamment l'identification et la classification des importants éléments de contexte en santé mobile. La proposition principale abordée dans ce chapitre est la modélisation de l'arbre de la classification et des éléments de contexte, en exploitant le modèle XML pour la première catégorisations et le modèle de l'orienté d'objet pour la deuxième hiérarchisation. Ces données assurent une utilisation efficace et potentielle des informations contextuelles dans le processus d'adaptation des applications. Ce qui rend la réaction de ces applications plus rapide quand les conditions externes changent.

Nos modèles proposés offrent une présentation sous forme générique des éléments de contexte en s'appuyant sur les relations sémantiques entre ces éléments. En effet, l'exploitation de ces relations est indispensable dans notre plateforme gérant l'exécution des services qui nécessitent les cohérences entre les éléments pour obtenir la meilleure performance des applications. Enfin, Nous avons discuté les méthodes d'intégration et d'insertion des informations contextuelles dans le fonctionnement des applications. Ces

applications sont conçues en tenant compte la sensibilité au contexte et la capacité de l'adaptation selon les situations externes.

Le chapitre suivant, est consacré à l'analyse et l'étude approfondie de notre plateforme en termes de l'architecture conçue, l'approche d'adaptation, la méthodologie de la conception des applications et la stratégie de la gestion de contexte.

Chapitre 3

Méthodologie et modélisation : implémentation du modèle CAHS

Antérieurement, nous avons étudié le terme de contexte en santé mobile. La mise en œuvre de la plateforme de M-health est considérée comme une opération plus complexe, car elle est liée aux grandes quantités des informations hétérogènes et de même qu'elle vise à créer des applications modulaires dédiées aux terminaux mobiles. Pour cela, il est nécessaire de choisir les technologies adéquates à la création de notre plateforme. Par ailleurs, nous confirmons que le système d'exploitation mobile Android est plus approprié pour l'implémentation de prototype de notre plateforme **CAHS**.

Dans ce chapitre, nous décrivons la méthodologie suivie pour la conception des applications qui doivent être sensible au contexte et aussi capables de changer leurs comportements. Enfin, nous allons mentionner l'architecture et le fonctionnement de notre plateforme permettant la gestion et l'adaptation des services mobiles.

III.1 Application à base des services sensibles au contexte

Nous avons constaté dans le premier chapitre que les applications composées de multiples fonctionnalités sont plus appropriées à nos travaux dont chaque fonctionnalité présente un certain service. Ce dernier peut introduire des nombreuses versions d'implémentation selon le contexte courant. Notre suggestion se rapproche de proposition de Herault [102], tel que son idée s'articule autour des services techniques avec plusieurs personnalités ainsi que la suggestion des familles de produits SPL [103]. Pour cette raison, nous étudions par la suite le concept de services dans le cadre de notre thèse.

III. 1.1 Concept de Service

L'émergence de terme du service correspond à l'apparition de l'architecture orientée service permettant de résoudre le problème d'interopérabilité entre les applications et les architectures hétérogènes [104,105]. De nombreux chercheurs suggèrent les différentes définitions de service selon le but de leurs projets. Par exemple, Hall et al. [106] affirment qu'un service est une interface offrant un accord entre le fournisseur et les consommateurs. Selon le travail [107], un service est une brique logicielle autonome fournissant une fonction bien définie.

Notre vision fondée sur ces deux propositions est conçu dans le but de définir et mieux éclairer ce terme. Pour une meilleure explication de ce concept, nous donnons l'exemple suivant: un service S est un code programmation pour atteindre au résultat prédéfini. A ce niveau nous citons le cas d'un fournisseur F_j et un consommateur C_i . F_j est capable de fournir S par sa propre méthode. Quand C_i nécessite S , il doit appeler F qui exécute S approprié pour obtenir le meilleur résultat.

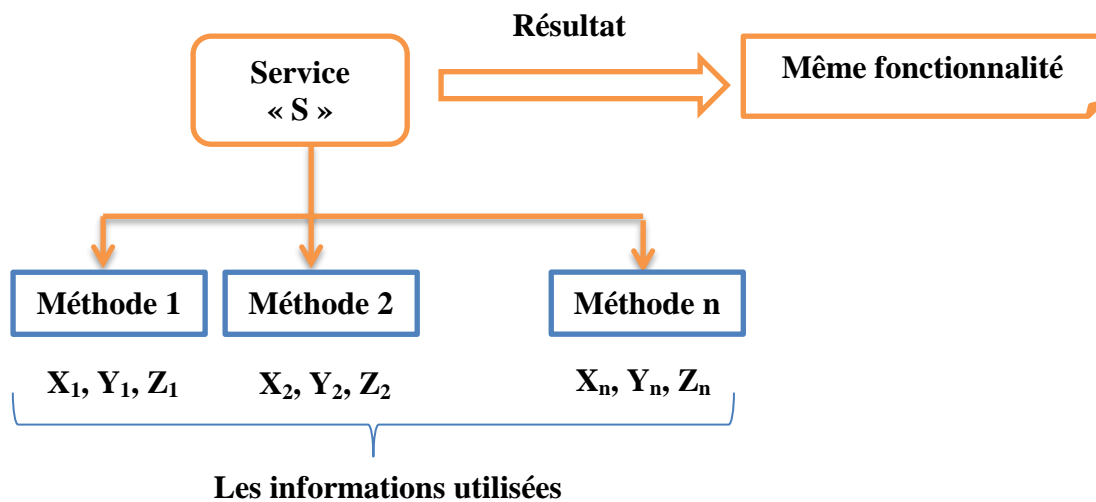


FIG. III.1 – Service S avec leurs méthodes de calculs

Nous pouvons décrire les services par leurs interfaces qui définissent le résultat, qui doit être fourni. En plus, une interface ne spécifie pas la méthode suivie pour calculer le résultat final. Cette notion est soigneusement illustrée dans la figure (FIG. III.1).

Il existe une multitude de méthodes qui aboutit au même résultat, néanmoins, chacune d'elles représente une version particulière. C'est dans cette perspective que chaque fonctionnalité est actionnée en fonction de sa manière personnalisée. Nous exploitons cette caractéristique pour implémenter l'adaptation de services en fonction de contexte actuel.

L'adaptation des applications conduit à changer leur comportement selon le contexte. A cet effet, notre approche est fondée sur la conception des applications à base de services. Ce qui rend ces applications légères et flexibles. Notre défi le plus important est la méthodologie d'intégration des éléments contextuels dans l'architecture harmonieuse des applications et de leur exploitation dans le processus d'adaptation, sans l'interruption d'exécution.

III. 1.2 Services dépendants de la situation de contexte (ScDS)

Comme nous l'avons proposé auparavant, l'application est structurellement présentée sous forme d'un assemblage des services. La méthodologie d'adaptation permet de gérer l'exécution de ces services selon les informations contextuelles. Pour cette fin, nous nous sommes basés sur le concept des services dépendants du contexte (CDS) [3]. La dite notion esquisse la relation entre les versions de services et les éléments de contexte. Dans notre proposition, nous nous focalisons sur les services dépendants de la situation de contexte *ScDS* (*Situation of context Dependent Service*) pour assurer le meilleur comportement d'application. De prime à bord, nous devons étudier les différentes situations où les applications ont besoin d'exploiter un service bien défini. Donc, il est nécessaire de créer plusieurs versions d'un service pour couvrir toutes les situations potentielles.

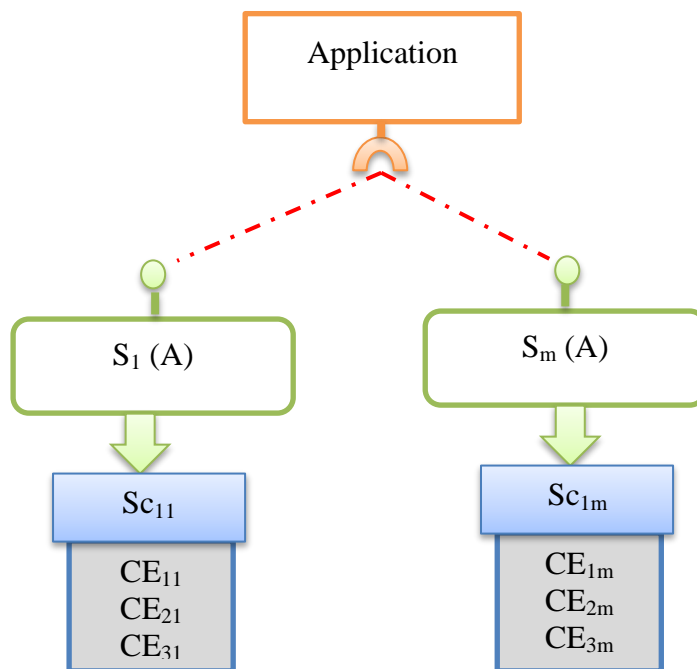


FIG. III.2 – Services dépendants de la situation de contexte *ScDS*

Chaque version possède une description spécifique destinée à déterminer les conditions dans laquelle cette version pourrait fonctionner. Il y a deux façons pour l'utilisation de la situation contextuelle afin que les applications puissent s'adapter.

A- Spécification du service

La première exploitation des *ScDS* consiste à identifier le service employé dans les conditions contextuelles particulières. Dans certains cas spécifiques, le service ne couvre pas forcément toutes les situations de contexte. A titre d'exemple, le service d'analyse doit traiter les données médicales et envoyer des notifications en cas d'urgence. Si l'utilisateur n'est pas le patient le service serait automatiquement stoppé.

B- Choix de la version de service

La deuxième voie pour utiliser les *ScDS* consiste à impliquer toute les situations contextuelles (Sc_{ij}) issues d'un ou plusieurs éléments. Si Sc_{ij} est créé, il est indispensable de choisir une version du service adéquat à cette situation (une version par défaut est toujours possible). De tout ce qui précède, le choix de module d'adaptation fonctionne en harmonie avec les situations contextuelles identifiées; la version sélectionnée est celle la plus appropriée au contexte.

III. 1.3 Description des versions de service

Notre objectif réside dans l'assemblage dynamique et autonome des applications au cours d'exécution, ce qui permet d'améliorer le processus d'adaptation. Pour cela, ces applications ne nécessitent pas l'intervention d'utilisateur. Donc le terminal est seul capable de gérer le couplage des services. Cette proposition se rapproche des buts de l'informatique autonome. Pour mieux sélectionner les versions de services appropriées, il en est obligatoire de définir une description de ces versions dans le but d'assurer l'assemblage correct et cohérent des applications.

Les langages de la description des services se manifestent par le langage UDDI [108] ou le WSDL [109]. Ces deux techniques garantissent l'accès aux services via Internet de même que leur description, en définissant les protocoles de communication et la totalité des informations nécessaires. Notre proposition se limite à l'exploitation des services disponibles sur les terminaux pour éviter l'interruption d'exécution des applications due à l'indisponibilité des services appropriés.

Service S	<ul style="list-style-type: none"> • Interface 	Interface implémentée (le contrat)
	<ul style="list-style-type: none"> • Attributs d'identité 	Nom de la version Numéro de la version
	<ul style="list-style-type: none"> • Attributs généraux 	Service fourni Service consommé

TAB. III.1 – Description d'une version de service

La description des versions de service se proclame comme les informations d'identification. L'objectif de cette description est l'amélioration du processus d'adaptation où le choix de la version se fait de manière rapide, facile et correct. Le tableau (TAB. III.1) décortique ce genre de choses selon nos besoins.

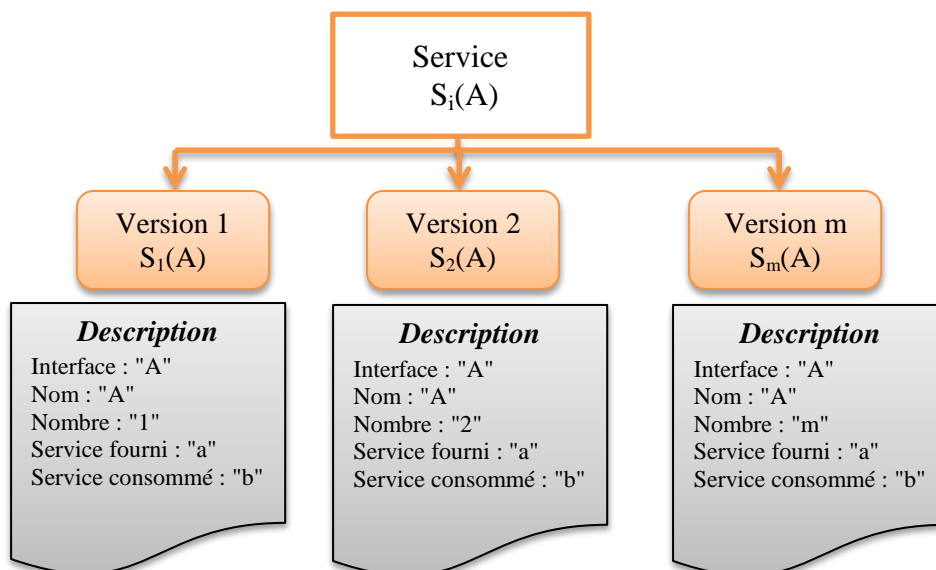


FIG. III.3 – Description des versions de service

La description esquissée dans le tableau (**FIG. III.3**) précédent englobe les informations liées à l'interface implémentée qui vise la définition de la fonctionnalité fournie. Elle comporte les attributs généraux qualifiés d'identiques par rapport à toutes les versions de même service. Il y a aussi les attributs d'identité qui permettent de choisir la version appropriée parmi les versions de même service. Les attributs d'identité contiennent deux paramètres : le nom de version et le numéro de la version. Toutes les versions s'inscrivant dans le même service portent la même appellation. Par contre, le numéro de la version est noté par i , qui varie entre 1, ..., m .

III. 2 Mécanisme d'adaptation des services sensibles au contexte

Dans la section ci-dessus, nous avons suggéré le concept des applications sous forme du groupement de services sachant que chaque service possède plusieurs versions qui sont créées spécialement pour couvrir toutes les situations contextuelles.

Notre approche d'adaptation s'articule autour de la gestion des versions de service appropriées à la situation contextuelle. Notre idée se rapproche dans une certaine mesure à celle proposée par M. Moeiz [70], que nous l'avons mentionné dans le chapitre I.

Le fonctionnement des services compte sur ScDS. une version de service $S_k(T)$ fonctionne selon la situation contextuelle Sc_{ij} courante. Pour cela, il est indispensable de situer toutes les versions de services adaptées par rapport aux situations de contexte possibles. Ce qui assure une exécution correcte des applications, quel que soit les situations contextuelles actuelles. Il est possible d'implémenter des versions de service « *par défaut* ». Ces versions fonctionnent dans le cas où aucune version n'est appropriée à la situation de contexte courante.

Notre processus d'adaptation est effectué grâce à des conditions bien délimitées et spécifiées par des règles clairement explicites. En faveur de la simplicité de ce système, nous exploitons une matrice de classe. Elle se caractérise par le repérage de nombre des situations contextuelles. Dans le chapitre précédent, nous avons proposé un arbre de classification (**FIG. II.4**) qui comporte cinq classes $C_{ct} = \{\text{Service context, Physical context, User context, Health environment context, Device Context}\}$. Alors que chaque classe possède une matrice (MC) particulière.

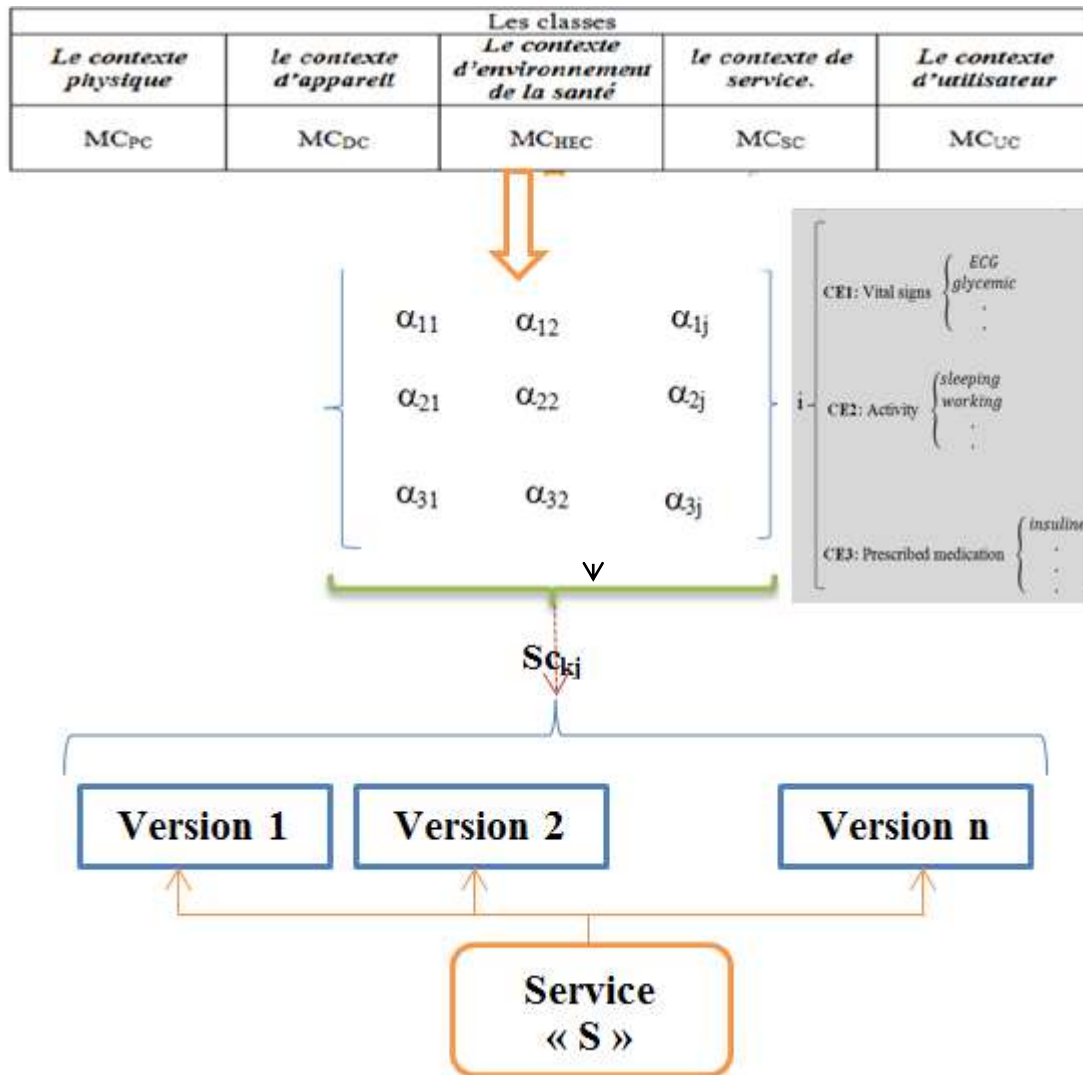


FIG. III.4 – Méthodologie de l'adaptation au contexte

Par exemple, il y a la classe de « *contexte d'environnement de la santé (HEC)* » qui contient trois éléments du contexte (signes vitaux, activité, médicaments prescrits). Cette matrice (MC) est représentée dans la figure (FIG. III.4). Elle est considérée comme la corrélation entre les versions et les éléments de contexte. Le mécanisme d'adaptation est effectué selon les vérifications de conditions suivant :

Si $k=j$ alors la version $S_k(T)$ démarre en fonction de Sc_{ij}III.1

En partant des MCs créées, le démarrage de toutes les versions nécessaires est conditionné par les situations de contexte ; autrement dit, le nombre des deux données doit être identique. Par ailleurs, nous préférons d'implémenter l'adaptation par une entité externe. Les applications s'effectuent en écartant le processus d'adaptation pour éviter l'interruption au cours de leur exécution. Donc, Il est nécessaire de sélectionner les versions des services appropriées avant d'exécuter les applications. Grâce à cette opération, notre plateforme gère le contexte et l'adaptation qui seront réactivés automatiquement pour toutes les applications. Ce qui la rend plus souple et efficace en termes des exécutions des applications.

III. 3 Description de la plateforme des Services de la Santé Sensibles au Contexte (CAHS)

Pour améliorer la qualité de la vie d'un patient, nous avons étudié les applications exploitées dans leur vie y compris la santé. De nos jours, le processus le plus important dans le développement des applications est l'adaptation au contexte, notamment pour les applications mobiles.

III. 3.1 Architecture de la plateforme proposée

Dans le chapitre précédent, nous avons étudié les éléments contextuels ayant un rôle essentiel dans l'exécution des applications de la M-santé. Nous avons déjà abordé une conception pour les applications adaptables. Nous analysons par la suite le fonctionnement de notre plateforme proposée ainsi que les unités de la gestion que nous considérons indispensables.

Nous visons la mise en place d'une plateforme pour les Services de Santé Sensibles au Contexte, que nous appelons **CAHS** (*Context-Aware Health Services Framework*). Le nom de celle-ci provient de domaine des applications faisant partie de notre recherche de même que l'approche que nous avons adoptée.

La plateforme que nous avons proposée indissociablement liée aux modules des tâches de la gestion qui garantissent l'efficacité et la simplicité des applications.

La **CAHS** facilite l'exploitation des informations contextuelles pour la gestion d'exécution des applications de la santé mobile. Donc, les finalités de cette plateforme s'articulent autour de l'adaptation de plusieurs applications et de partage des tâches de gestion inter-applications, qui se distinguent par les différents principes de la M-health (le diagnostic de la maladie, la référence de médicament, la calculatrice médicale, la communication clinique, le systèmes d'information hospitaliers, l'éducation médicale et l'enseignement, l'éducation des patients et l'engagement, la gestion des maladies chroniques, la télésanté et la télémédecine).

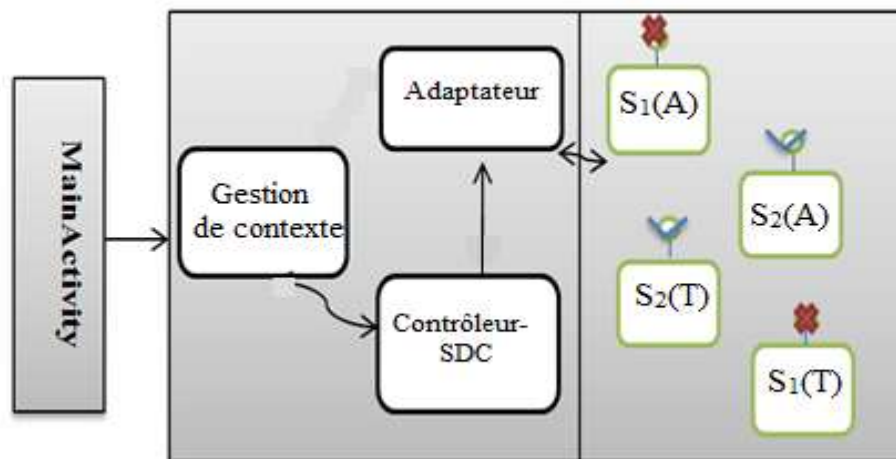


FIG. III.5 – Architecture de la plateforme CAHS

Hormis ces applications, notre plateforme comporte trois modules de gestion illustré précédemment dans la figure (FIG. III.5) : (1) la *gestion de contexte* (manager de contexte), (2) l'*adaptateur* (adaptor), (3) le *contrôleur-SDC* (controller-SDC). Chaque module implémente une tâche bien définie de la gestion. La conception de notre plateforme est fondée sur l'architecture orientée service ce qui la rend très simple et compréhensible en matière de la programmation et de l'implémentation de ce modèle.

Nous allons introduire en détail les buts et les outils d'implémentation de chaque module de la gestion.

III. 3.2 Implémentation de CAHS

Nous avons présenté auparavant (chapitre I) les technologies exploitées pour mettre en place notre prototype **CAHS**. L'objectif de cette section est l'implémentation des modules de la gestion et des applications de la M-health. Cette implémentation nous permet d'évaluer le comportement et les performances des mécanismes de l'adaptation et de la gestion du contexte qui sont proposés.

Nous avons choisi auparavant le système d'exploitation mobile *Android* pour l'évaluation de nos propositions. De prime à bord, nous développons une application Android en utilisant la version 4.4.2 du système d'exploitation (*API level 19*). En considérant que l'application Android est un hôte pour notre plateforme, cette application consiste à regrouper un ensemble classe permettant de concevoir notre prototype de **CAHS**. Dans laquelle, la classe la plus importante est *Activity*. Le but de l'application Android créée est le déclenchement de plateformes OSGi, le modèle iPOJO et les modules du management de **CAHS**, sans oublier les applications qui fonctionnent sur **CAHS**. De plus, l'Android nous permet de contrôler et d'évaluer l'exécution des expériences.

La figure (FIG. III.6) illustre le défi de l'incompatibilité des fichiers jar de composant iPOJO avec la machine virtuelle Dalvik sachant qu'Android utilise Dalvik. Pour cela, il est nécessaire d'extraire le fichier classes.dex pour chaque bundle d'iPOJO. Pour atteindre ce but, nous utilisons un dossier jar2dex, en exécutant la commande suivante :

```
d2j-jar2dex class.jar -o classes.dex
```

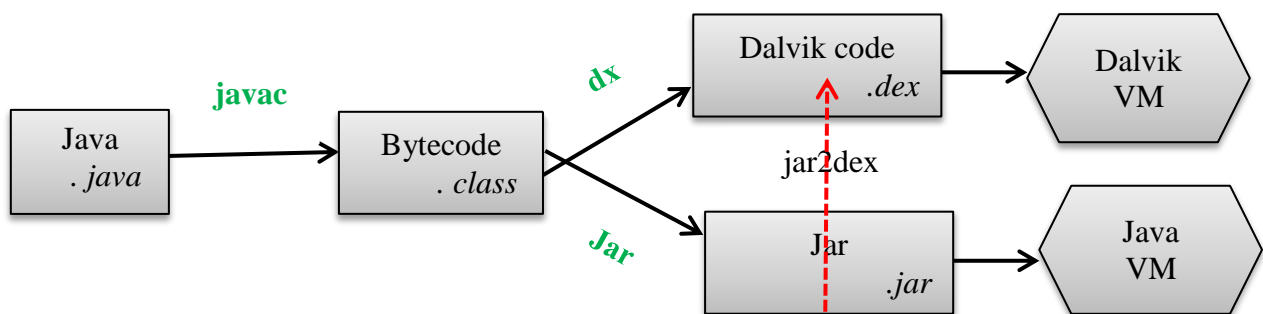


FIG. III.6 – Compatibilité d'OSGi avec Android

En outre, la plateforme **CAHS** est introduite par un fichier.apk installable sur le terminal mobile, comme une application normale. Les paramètres intégrés dans la plateforme **CAHS** sont mentionnés sur l'expression suivante :

Plateforme CAHS = OSGi + iPOJO + modules du management + applicationsIII.2

III. 3.3 Modules de la gestion de CAHS

III. 3.3.1 Gestion de contexte

Nous avons constaté la nécessité de l'implication de la sensibilité de contexte dans le développement des applications mobiles afin d'adapter leur comportement au cours de leurs d'exécutions. Pour cela, nous nous focalisons sur le système sensible au contexte pour gérer ce mode des applications. La section I.2.2 montre que la gestion de contexte est le module crucial dans ce système.

Généralement, le but de ce module est d'évaluer certains éléments de contexte dans des moments différents. La particularité de notre proposition est l'exécution d'un groupe des évaluateurs d'éléments de contexte appartenant à la même classe dans un cadre temporel bien

précis. Le processus de gestionnaire de contexte illustré dans la figure (FIG. III.7) se compose de deux parties:

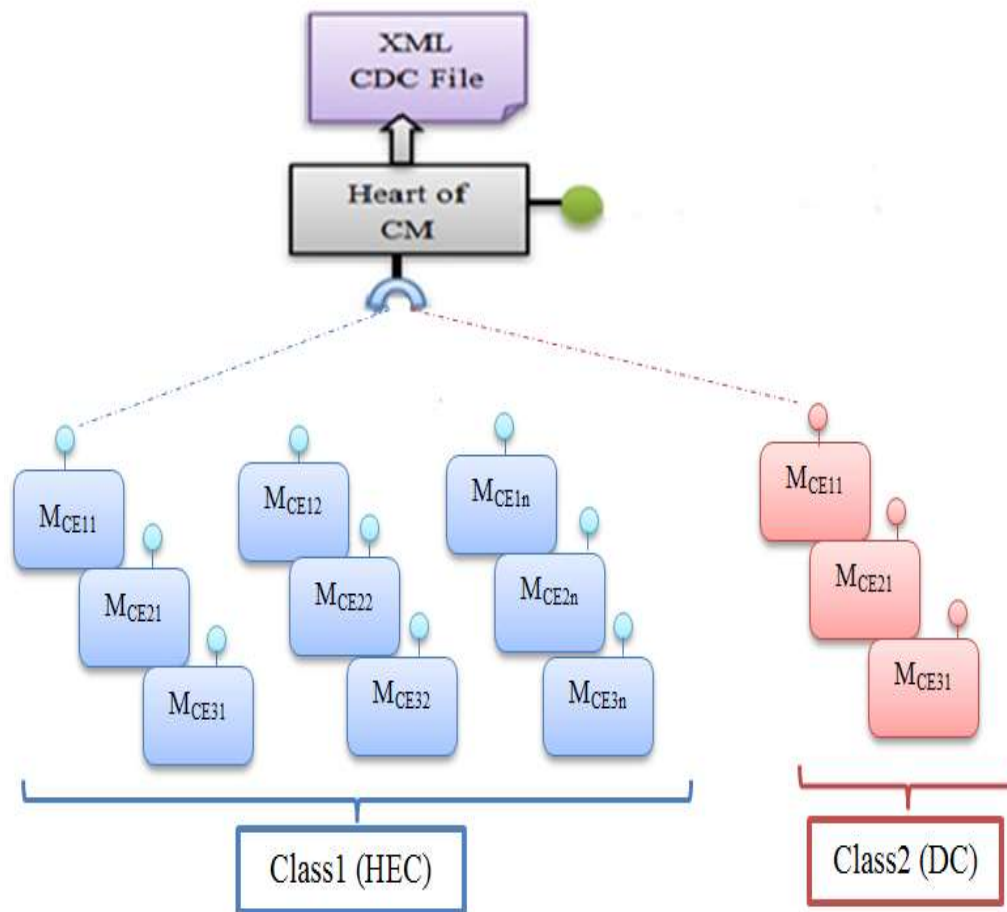


FIG. III.7 – Gestion du Contexte

Moniteur d'élément de contexte. Le besoin de la surveillance de situation médicale nécessite d'évaluer de manière permanente tous les éléments ayant un lien avec les informations liées à la santé. Ce qui nous conduit à proposer un module, nommé *Context Element Monitor*. Ce module est capable de suivre les valeurs des éléments. En effet, chaque élément est doté d'un moniteur noté M_{CEij} où i donne une représentation numérique de l'élément de contexte dans la classe et j fournit une représentation numérique de la situation créée par un ensemble d'éléments de contexte.

La figure (FIG. III.8) dépeint le principe de la création de M_{CEij} via la classe est HEC. Nous présentons un exemple : CE est un signe vital, alors i égal à 1 et lorsque le signe vital représente le signal ECG, j est égal à 1. Quant au moniteur est abrégé par M_{CE11} .

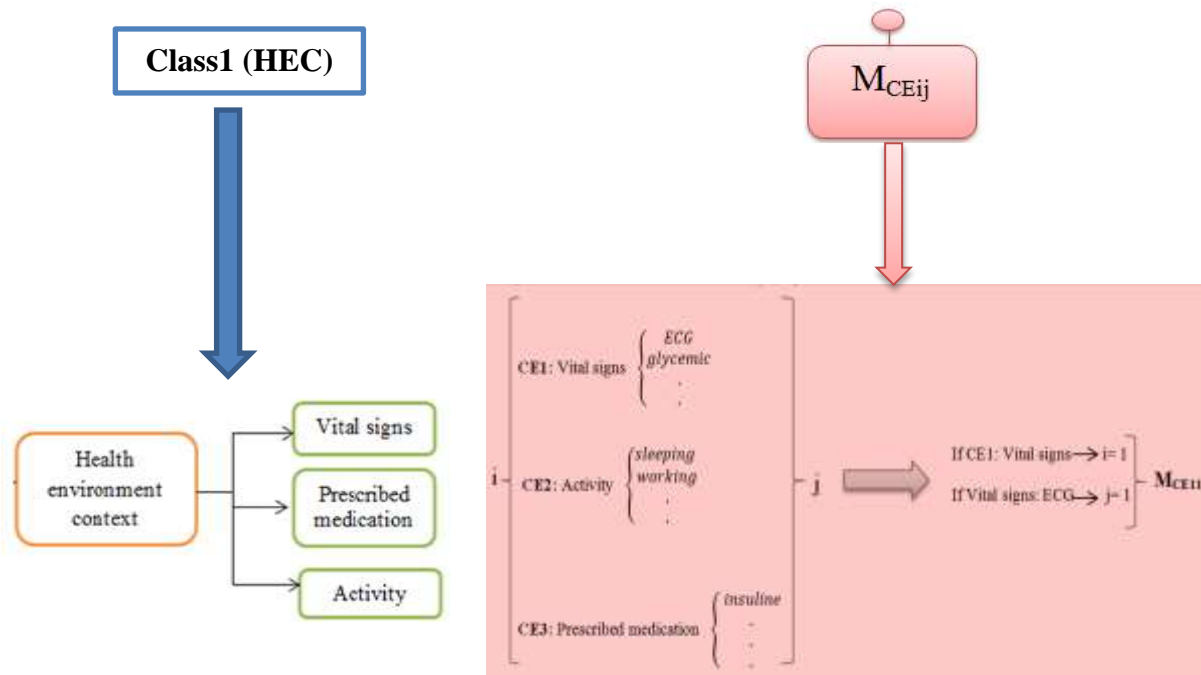


FIG. III.8 - Création du moniteur d'élément de contexte (M_{CEEij})

Noyau du gestionnaire de contexte. Etant le noyau de gestionnaire de contexte représenté par "Noyau de CM", ce sous-module en occupe une partie primordiale. De plus, il peut accéder au fichier XML de la classification puis, extraire les informations nécessaires afin de démarrer les moniteurs appropriés des éléments. Ensuite, il notifie l'état d'exécution des moniteurs au contrôleur-SDC, qui crée la situation contextuelle.

Il existe trois fichiers texte: le premier fichier, appelé Profile.txt qui contient les informations de l'utilisateur (le prénom, le nom, l'âge, le statut du patient ou du médecin, le nom d'utilisateur et le mot de passe) et le second fichier se nomme Status-User.txt. Le deuxième type apparaît lorsque l'utilisateur est un patient. Le fichier Presmedmedication.txt contient des renseignements relatifs au patient (la maladie dont il souffre, le numéro, le nom et la dose de médicament, ainsi que le meilleur moment pour le prendre).

III.3.3.2 Implémentation de la gestion de contexte

En faisant appel à l'illustration désignée par la figure (FIG. III.7), la gestion de contexte comporte trois parties fondamentales: Noyau du gestionnaire de contexte, les moniteurs des éléments de contexte et un fichier CDC XML qui présente l'arbre des éléments de contexte. En effet, le Noyau du gestionnaire de contexte fonctionne de manière permanente. Contrairement aux moniteurs qui s'exécutent de manière optionnelle. Malgré ceci, le Noyau du gestionnaire de contexte ne sera pas avantageux si les éléments de contexte ne sont pas évalués par des moniteurs.

Le moniteur dans la CAHS doit périodiquement évaluer un élément contextuel. Donc, pour chaque classe, nous consacrons un moment à l'évaluation des éléments de contexte appartenant à cette classe. Pour cette raison, le Noyau du MC avant toute opération effectuée une lecture de fichier CDC XML en utilisant DOM parse pour préciser les éléments de contexte qui s'inscrivent dans la même classe. À partir de résultat issu de cette lecture, le Noyau de CM prend la décision pour l'exécution d'un ensemble de moniteurs dans un temps précis. Ces moniteurs permettent de suivre les éléments ayant une influence sur les mêmes services. Ce qui facilite le processus de la collection et de l'évaluation des informations contextuelles, quelle que soit sa quantité et sa qualité. Les étapes pour gérer l'état des

moniteurs, sont décrites dans le diagramme (FIG. III.9). Dans cette optique, nous spécifions la signification de chaque paramètre dans ce diagramme :

K: identifier le cadre temporel nécessaire pour d'évaluer les éléments contextuels de chaque classe.

m: le numéro de premier élément de contexte dans la classe spécifique.

$C_{ancienne}$: la classe ancienne.

$C_{actuelle}$: la classe actuelle.

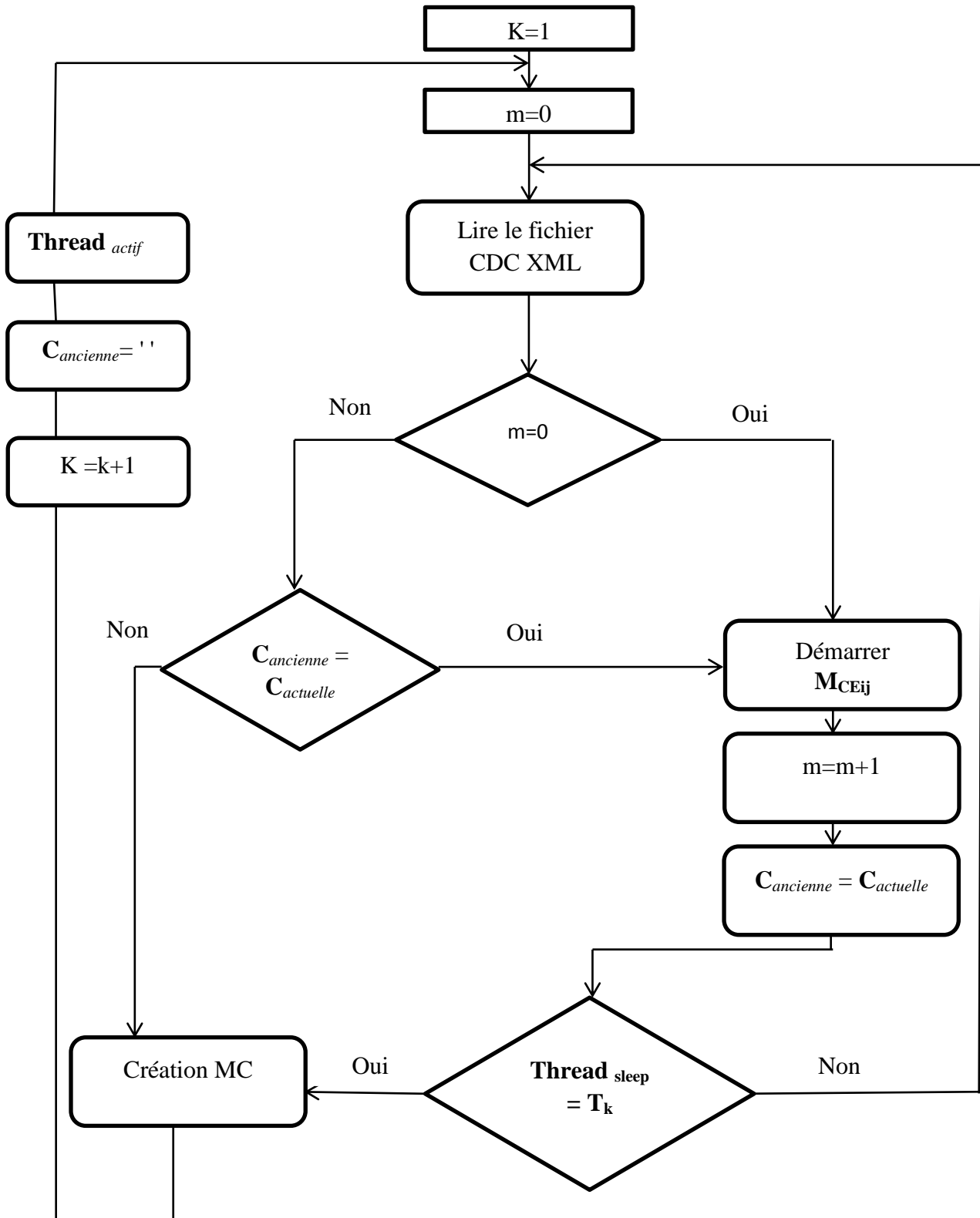


FIG. III.9 – Diagramme du module de la gestion de contexte

D'un côté, pour rassembler le gestionnaire de contexte et le module de contrôleur-SDC, nous exploitons un service de type fournisseur/consommateur. Le gestionnaire de contexte fournit un ensemble des informations liés aux états des moniteurs (*ACTIVE* ou *DESACTIVE*). D'un autre côté, le contrôleur-SDC produit une situation contextuelle alimentée par ces informations. Nous exposons par la suite la description de module *Contrôleur-SDC* et son implémentation.

III. 3.3.3 Contrôleur-SDC (Contrôleur de service dépendant de contexte)

Le deuxième module de la gestion est le contrôleur de service dépendant de contexte (Contrôleur-SDC). L'objectif principal en est la facilité et la simplicité de la concrétisation de la notion du contexte dans le mécanisme d'adaptation. Le Contrôleur-SDC est capable de créer une matrice de classe selon l'état des moniteurs (*ACTIVE* ou *DESACTIVE*) pour obtenir la situation contextuelle.

La figure (**FIG. III.10**) intègre le principe de la conception de cette matrice ayant le coefficient α_{ij} qui prend la valeur 0 ou 1. Comme nous l'avons exposé précédemment, notre idée de la création de MC gravite autour de l'exécution des moniteurs en misant sur deux conditions distinguées par leur simplicité et clarté. Elles sont nettement représentées par (**III.3**).

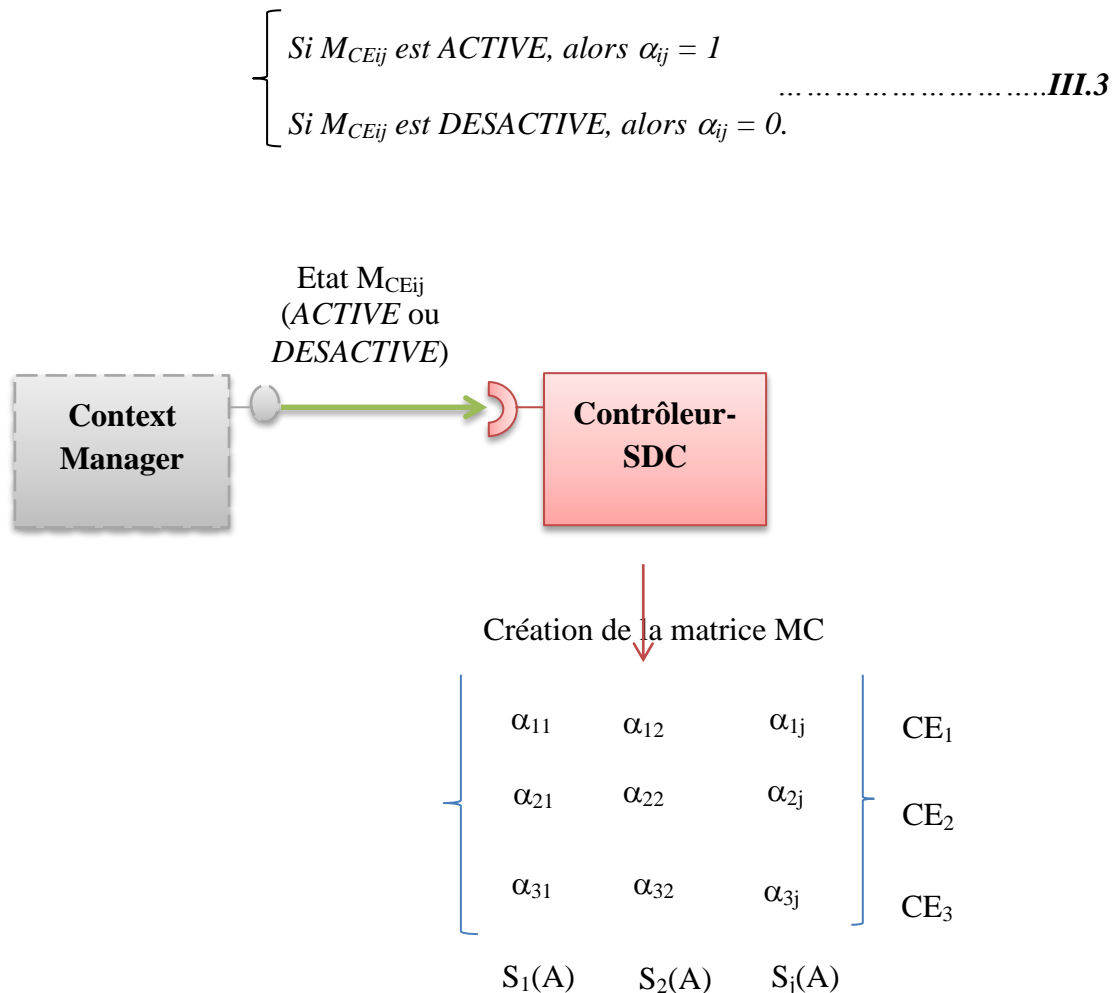


FIG. III.10 - *Contrôleur-SDC*

Le Contrôleur-SDC garantit des avantages énumérés dans les points suivants:

- La simplification des informations compliquées de contexte.
- La création de la situation contextuelle la plus sémantique et raisonnable.
- La simplicité et la clarté de la méthodologie d'exploitation des relations entre les éléments.

III. 3.3.3 Implémentation de contrôleur-SDC

Comme nous avons antérieurement mentionné, l'objectif principal de ce module est la création de la matrice de classe, qui assure la dépendance entre les versions de services et les éléments de contexte. Par conséquent, il est obligatoire de simplifier les informations contextuelles pour provoquer une réaction plus vite en cas de besoin. Le gestionnaire de contexte informe le module sur les états des moniteurs. C'est grâce au facteur précédent que la matrice MC trouve son existence. La figure (FIG. III.11) illustre l'algorithme programmé pour la création de MC, sachant que le paramètre (*sec11*) identifie le moniteur qui a mis en activité.

```
if(sec11-10<10){
    for(int i=1;i<4;i++){
        for(int j=1;j<3;j++){
            if(j==sec11-10){
                MC1[i][j]=1;
            }
            else{MC1[i][j]=0;}
        }
    }
}
else{}
System.out.println("La matrice de Classel: ");
```

FIG. III.11 - Algorithme de la création la matrice de classe HEC

Pour faciliter le fonctionnement de l'adaptateur, le contrôleur-SDC spécifie les versions des services appropriés selon la situation contextuelle. Cette procédure joue un rôle déterminant dans la réduction du temps nécessaire pour adapter les applications. L'algorithme implémenté (FIG. III.12) définit les versions des services en fonction de la matrice MC.


```

int a=0;
for(int i=1;i<4;i++){
    for(int j=1;j<3;j++){
        if(CE1[i][j]==1){
            Vs_affiche="S_affiche"+j;
            a=1;} else{m_statue=true;
        }l=j;}}
for(int i2=1;i2<3;i2++){
    if(CE2[i2]==1){ m_statue2=true;}
if(a==0 && m_statue2==true && m_statue==true ){
    int s=0;
    s=l+1;
Vs_affiche="S_affiche"+s;
m_statue=false;
m_statue2=false;}
else if(a==0 && m_statue2==false && m_statue==true ) {
    Vs_affiche="S_affiche_default";
    System.out.println("S_affiche_default:"+l);
}
else{}
return Vs_affiche;

```

FIG. III.12 - *Algorithme de la détection des versions de service « affichage »*

Après la sélection des versions de services appropriés, ledit module informe l'adaptateur pour gérer l'exécution de ces versions. Nous expliquons ci-dessous le fonctionnement de module d'adaptateur.

III. 3.3.3 Adaptateur

Nous avons constaté qu'il est indispensable de séparer le module de la gestion et les applications. Dans cette section, nous nous focalisons sur un module externe appelé *Adaptateur*. Il gère le cycle de vie des services sur la plateforme **CAHS** pour assurer un assemblage correct des applications. En vue d'une gestion efficace des services, l'adaptateur doit tenir compte les informations suivantes :

La composition des applications. C'est le module qui prend en considération les informations relatives aux dépendances entre les fournisseurs et les consommateurs.

La description de la version de service. Étant donné que la version de service s'identifie par le nombre des versions, l'adaptateur peut accéder à la description de chacune pour connaître ce nombre.

La version spécifiée. L'adaptateur doit connaître la version repérée par le contrôleur-SDC afin de la démarrer.

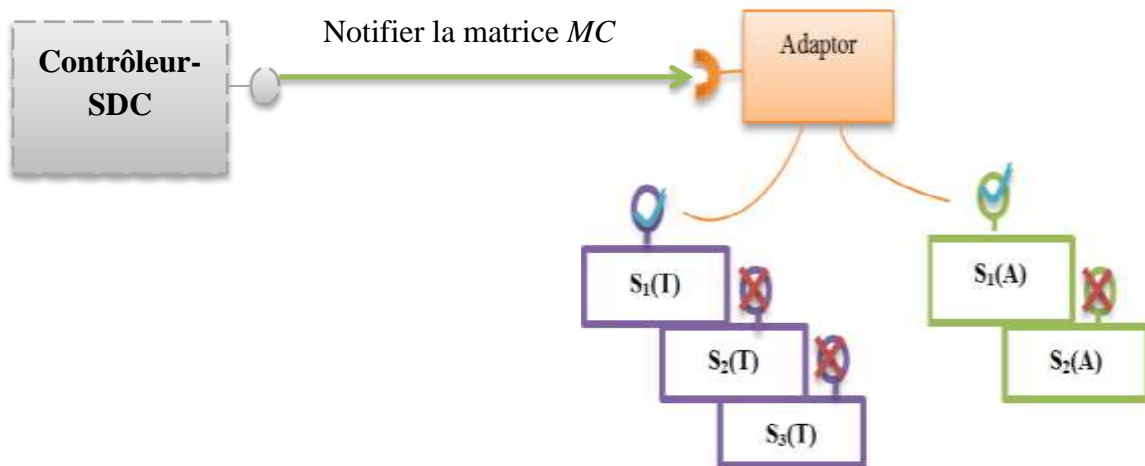


FIG. III.13 – *Adaptateur*

L'adaptateur analyse toutes les versions pour sélectionner celle qui correspond au choix de contrôleur-SDC (FIG. III.13). De plus, ces versions sont disponibles au niveau local afin de gagner le temps de recherche et de téléchargement.

III. 3.3.6 Implémentation d'adaptateur

L'adaptateur joue un rôle décisif sur l'état des services (l'installation, le démarrage, l'arrêt). Le but de ce module est l'ajustement des services à la situation contextuelle, en exploitant le concept de ScDS. L'adaptateur obtient la liste des versions nécessaires à l'aide de contrôleur SDC et en s'appuyant sur la méthode `m_controller.controller_Saffiche()`.

```

int a=0;
for(int i=1;i<4;i++){
  for(int j=1;j<3;j++){
    if(CE1[i][j]==1){
      Vs_affiche="S_affiche"+j;
      a=1;} else(m_statue=true;
    }l=j;)}
for(int i2=1;i2<3;i2++){
  if(CE2[i2]==1){ m_statue2=true;}
}
if(a==0 && m_statue2==true && m_statue==true ){
  int s=0;
  s=l+1;
  Vs_affiche="S_affiche"+s;
  m_statue=false;
  m_statue2=false;}
  else if(a==0 && m_statue2==false && m_statue==true ) {
    Vs_affiche="S_affiche_default";
    System.out.println("S_affiche_default:"+l);
  }
  else{}
return Vs_affiche;

```

FIG. III.14 - *Algorithme d'adaptation d'un service selon la situation de contexte*

L'adaptateur est capable d'accéder au fichier `manifest.mf` des versions de service pour extraire le nom. Ceci se fait grâce à l'instruction `bundle.getSymbolicName()`. La tâche suivante assignée à l'adaptateur est purement analogique. Ainsi, il compare la notification de contrôleur-SDC au résultat de recherche. Sauf lorsqu'ils sont identiques que l'adaptateur décide de démarrer la version par l'intermédiaire de l'instruction `bundle.start()`. La figure (FIG. III.14) expose l'algorithme nécessaire pour adapter le service à la situation de contexte, par le biais d'une méthodologie d'adaptation qualifiée de simple et claire, la chose qui apporte

un changement radical au comportement des applications en accélérant leur réaction. La conception de ces applications sera minutieusement détaillée dans le prochain titre.

III.3.4 Application à base des services de la santé mobile

Comme nous l'avons introduit dans la section III.1, la conception des applications est basée sur une composition des services. Chacun d'eux assume une certaine fonctionnalité. Dans nos travaux, nous nous focalisons sur les applications de la santé mobile ayant l'ambition d'assister les patients dans leur vie quotidienne.

Dans ce cadre nous avons choisi les applications de la surveillance dédiées aux personnes qui souffrent des maladies chroniques (diabète, pathologies cardiovasculaires). Ce genre d'applications mène à bonne fin notre recherche car il permet de mesurer le degré de la performance des modules de la gestion. D'une part, la sélection de ces applications est due au même objectif et fonctionnalité qu'elles partagent. D'autre part, ce choix renvoie aussi à la différence existante entre les manières de la présentation et d'analyse des données médicales. Dans notre recherche l'étude de la conception des applications en termes des fonctionnalités nécessaires précède leur implémentation.

Concernant le diabète

Etant une maladie chronique, le diabète se propage largement parmi la population mondiale [109]. Cette pathologie est dû soit à l'insuffisance de sécrétion d'une hormone nommée l'insuline, soit à l'absence de la production de cette hormone par le pancréas, chez les diabétiques le glucose s'accumule dans le sang et déséquilibre le taux de sucre [111]. En ce fait, le patient attaqué par le diabète est l'objet d'une surveillance permanente pour éviter les complications issues de cette maladie telles: l'endommagement de certains organes (les yeux, les reins, le cœur, les nerfs et les vaisseaux sanguins) [112]. Alors, les informaticiens développent de nombreuses applications qui permettent de suivre la situation médicale de patient à tout moment, et à tout lieu. De plus ces applications nécessitent plusieurs fonctionnalités afin de fournir le meilleur résultat en termes de l'analyse et d'affichage.

Fonctionnalité de l'affichage – Elle est chargée de présenter la situation médicale telle la glycémie, l'état de la santé (normal/anormal). Elle a besoin de plusieurs informations pour assurer la bonne présentation des données médicales. Ces renseignements concernent les informations d'utilisateur (patient ou médecin), la taille de l'écran de terminal, ainsi que le type des données médicales (valeur ou signal). Une présentation appropriée favorise une lecture claire et cohérente.

Fonctionnalité de l'analyse – Elle permet d'analyser les données médicales pour déduire l'état de la santé. Elle nécessite les informations qui représentent le taux du sucre, l'activité et le médicament décrit. Dans ce cadre, notre proposition se repose sur une collection des éléments qui se font de manière regroupée pour éviter la perte du temps en attendant l'acquisition de toutes les informations obligatoires. En outre, nous tenons compte de taux de la glycémie comprise entre 1 et 1.26 g/l [113]. L'échelle à partir de laquelle on évalue la situation médicale de patient.

Fonctionnalité de la communication – Elle assure l'envoi et la réception des informations de la santé où la connexion joue un rôle central dans l'échange des données médicales.

Concernant les maladies cardiaques.

Etant nombreuses, les pathologies cardiaques sont liées au dysfonctionnement du cœur. La maladie cardiaque lie à un mauvais fonctionnement du cœur [114]. Ces maladies sont détectées par l'analyse du signal électrocardiogramme ECG qui contient plusieurs ondes (P, Q, R, S, T), comme nous avons illustré dans la figure (FIG. III.15).

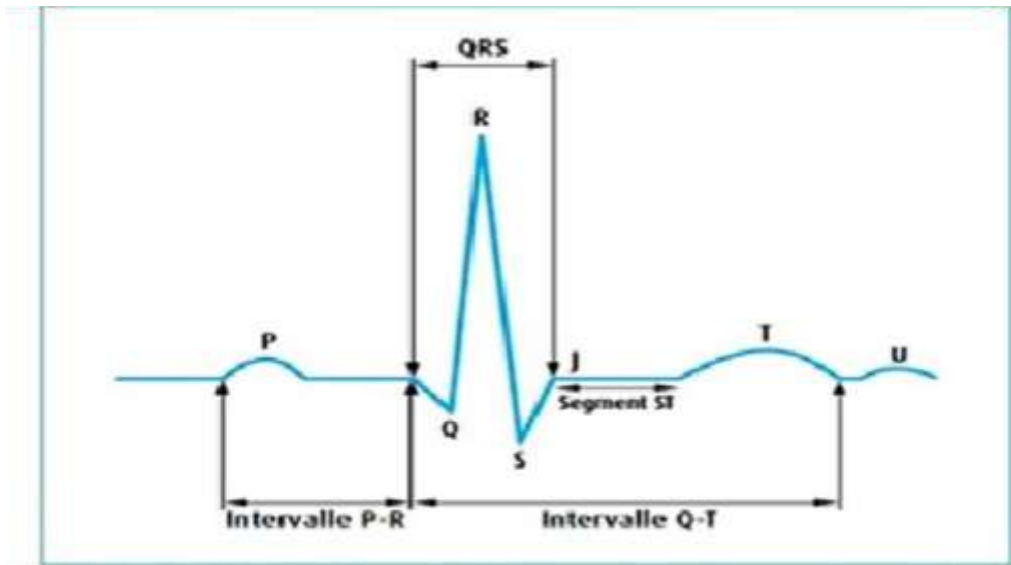


FIG. III.15 – Signal ECG [115]

Un nombre considérable de gens meurent chaque année en raison de maladie cardiaque que de toute autre cause. Ceci conduit à suivre les patients qui souffrent de ce problème. Le rythme cardiaque, l'intervalle QRS et l'amplitude R sont des paramètres primordiaux qui doivent être constamment évalués et analysés, si bien que les informaticiens inventent et rénovent des applications performantes en termes de détection de la situation anormale du signal ECG.

L'application « *Moniteur fréquence cardiaque* » [116] peut préciser la fréquence cardiaque via le Smartphone. En outre, l'application « *Rythme cardiaque plus-cardiofrequencemetre* » [117] mesure le rythme cardiaque partout et à tout moment, en se servant de capteur de l'appareil photo Smartphone. Multiples sont les applications mobiles qui surveillent le signal ECG et leur bonne marche est soumise aux fonctionnalités suivantes:

Fonctionnalité d'affichage – Elle permet d'afficher la situation médicale, notamment le signal ECG, les paramètres détecté (complexe QRS, Rythme cardiaque, Peak R, etc.), l'état de la santé (normal, anormal). Cette fonctionnalité nécessite plusieurs informations tel que le type d'utilisateur (patient, médecin), la quantité des données médicales, la taille de l'écran. Ces données fournissent une bonne plateforme pour interpréter l'état de l'utilisateur.

Fonctionnalité d'analyse – Elle est chargée de traiter et analyser le signal ECG à travers des algorithmes [118-123] qui spécifient les paramètres du signal ECG (complexe QRS, Rythme cardiaque, Peak R, etc.). La situation médicale est cernée par l'intermédiaire des paramètres en respectant les données communiquées par le tableau ci-dessous (TAB. III.2):

	Onde P	Intervalle PQ	Complexe QRS	Intervalle ST	Intervalle QT	Onde T
Durée (s)	0.08 - 0.1	0.12 - 0.2	0.08	0.20	0.36	0.2
Amplitude (Mv)	0.25	Isoélectrique 0	Qa<0, Ra>0, Sa<0	Isoélectrique 0	---	Ta>0

TAB. III.2 – Paramètres normaux du signal ECG [124]

Fonctionnalité de la communication – Elle assure le transfert des informations médicales (les paramètres détectées, l'état de la santé) entre le patient et le médecin.

Après avoir étudié les applications, nous expliquons leur mode d'exploitation. Pour cela, nous considérons la fonctionnalité d'affichage. Nous distinguons deux modes de la fonctionnalité d'affichage. Le premier mode assume la mission de l'affichage des informations médicales de

la maladie de diabète et le second type représente les paramètres médicaux dédiés à la maladie cardiaque. Alors, nous cernons deux méthodes différentes visant la même finalité (objectif). Cette fonctionnalité introduit le service d'affichage mentionné S_i (affiche) sachant que ce service possède deux versions qui sont exprimées comme suit :

$$S_i(\text{affiche}) = \{ S_1(\text{affiche}), S_2(\text{affiche}), \dots, S_n(\text{affiche}) \}, \text{ tel que } i=1, \dots, n \dots \text{III.4}$$

Chaque version s'exécute dans une situation contextuelle spécifique. Dans cette partie, nous nous basons sur le concept de ScDS (Section III.1.2) pour établir la dépendance entre les versions de services et la situation contextuelle courante.

La figure (FIG. III.16) illustre les versions de services d'affichage dépendantes de certaines situations contextuelles, créées par les éléments contextuels. Ainsi, chaque version s'appuie sur une description simple pour identifier la situation contextuelle qui correspond à l'exécution de cette version. Notons que ladite situation contient un ou plusieurs éléments de contexte (CEs) faisant partie de la même classe.

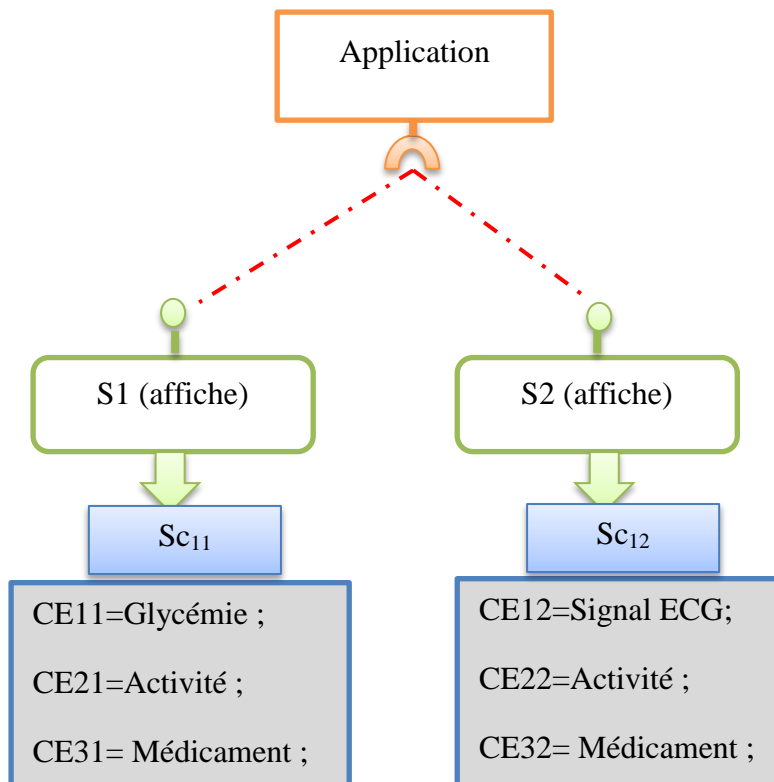


FIG. III.16 – Services d'affichage dépendant de la situation de contexte

Exemple de la première version. S_1 (affiche) Elle vise à présenter la situation médicale de patient à travers le Smartphone. La version dépend donc des trois éléments de contexte, nommés Glycémie, Activité et Médicament. Le premier élément est de type *Discret*, tandis que le deuxième élément est *String*, par contre, le troisième est de type *Booléen*. Cette version est évidemment conçue pour fonctionner si la situation de contexte est créée par ces éléments précédents.

Exemple de la deuxième version. S_2 (affiche) Nous considérons une version qui affiche les situations médicales dédiées à la maladie cardiaque par le biais de Smartphone. Ladite version est soumise à trois éléments, notamment Signal ECG, activité et médicaments. Le type de premier élément est *Continu*, le deuxième est *String* ainsi que le troisième est *Booléen*. Donc, nous pouvons dire que chaque situation de contexte est responsable de fonctionnement d'une version particulière. En plus, chaque version permet de présenter un certain mode d'affichage des informations médicales liée aux différents éléments de contexte.

Vu l'existence de plusieurs applications de la surveillance des pathologies (diabète et cardiaque) dans le marché Play store, notre objectif n'est pas l'algorithme de leur mise en œuvre, mais il réside dans l'implémentation des applications à base des services. Ce qui permet de faciliter le changement et l'adaptation de leur comportement selon le contexte. De plus, ces applications choisies nous permettent d'évaluer la performance des mécanismes que nous avons proposés (la gestion de contexte et l'adaptation).

III. 3.5 Implémentation des applications à base des services

Dans les sections précédentes, nous avons supposé que les applications sont divisées selon les fonctionnalités qui représentent les services. Chacun d'eux est doté de plusieurs versions notées par $S_i(A)$. La création de ces applications compte sur l'assemblage de versions des services. En outre, les dites applications profitent des intérêts de la technologie OSGi assurant leur flexibilité et dynamicité. Nous pouvons exprimer les applications et les services par les expressions suivantes:

$$\text{Application} = S_i(A) + S_i(B) + \dots + S_i(Z); i=1, \dots, n \dots \dots \dots \text{III.5}$$

$$S_i(A) = \{S_1(A), S_2(A), S_3(A) \dots S_n(A)\} \dots \dots \dots \text{III.6}$$

Grâce à la spécification OSGi, nous pouvons créer les applications, comme nous les avons envisagées dans notre proposition (Section III.1). Nous avons sélectionné les métadonnées sous forme de fichiers XML pour que le code soit indépendant de toutes les informations iPOJO. De plus, chaque bundle exprime une version d'un service, autrement dit, une partie de code (une version de service) encapsule dans le bundle. Les technologies (OSGi, iPOJO) utilisées sont performantes pour définir les fournisseurs et consommateurs de services.

Dans notre travail, nous allons exploiter des tags: `<provides/>` pour les fournisseurs et `<requires field="...">` pour les consommateurs, en concevant la balise qui définit le service qui lui correspond. Le bundle contient un fichier *manifest.mf* incluant les informations (*Bundle-Name*, *Bundle-SymbolicName*, *Bundle-Version*, *Import-Package*, *iPOJO-Components*) liées à ce module. Nous nous intéressons à l'information de *Bundle-SymbolicName* qui prend le nom de la version de service, ce qui permet d'identifier la version appropriée selon la situation de contexte actuelle, tel que la figure suivante (FIG. III.17) l'illustre :

```

Bundle-ManifestVersion: 2
Bundle-Name: S_affiche1
Bundle-RequiredExecutionEnvironment: JavaSE-1.7
Bundle-SymbolicName: S_affiche1
Bundle-Version: 1.0.0.qualifier
Created-By: iPOJO 1.12.1
Import-Package: android.app, android.graphics, android.view, android.w
idget, com.inter.service.analyse, inter.web.service, org.apache.felix
.ipojo;version=0.8.0, org.apache.felix.ipojo.architecture;version=0.8
.0, org.example.Saffiche, org.inter.serclient, org.osgi.service.cm;ve
rsion=1.2, org.osgi.service.log;version=1.3
iPOJO-Components: instance { $component="com.service.affiche.Saff.Serv
ice_affiche" }component

```

FIG. III.17 – *Manifest.mf* d'une version 1 d'un service d'affichage

Pour tester l'efficacité de notre plateforme, le choix que nous avons effectué s'articule autour des applications de la surveillance destinées aux maladies de diabète ainsi que cardiaque.

Bien que la finalité de cette thèse est la gestion de contexte et de l'exécution des applications, nous avons procuré les algorithmes simples de classification de signal ECG ainsi que l'analyse de taux de glycémie pour extraire les paramètres nécessaires de l'évaluation de la situation médicale, précisément montrés dans la figure (FIG. III.18).

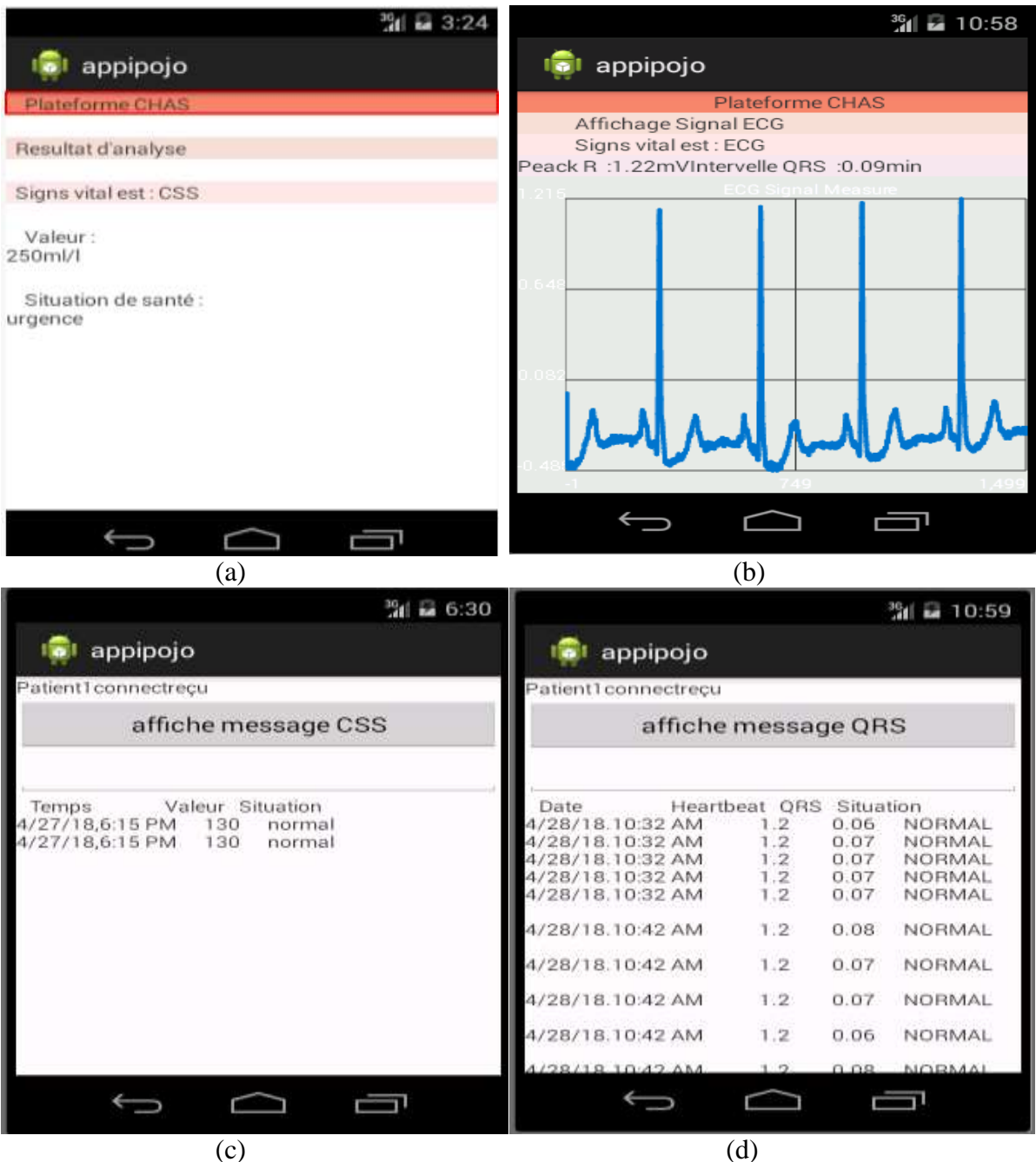


FIG. III.18 – Versions des services

La figure (FIG. III.18 (a)) représente le style d'affichage de données médicales pour la maladie de diabète. La figure (FIG. III.18 (b)) présente le signal ECG avec les résultats d'analyse de ce signal. Ces dernières figures (III.18 (a) et (b)) exposent les versions de service d'affichage destinées au patient, sans écarter les figures (FIG. III.18 (c) et (d)), qui fournissent les informations médicales reçues par les médecins.

Après avoir exécuté les applications à base des services, nous avons constaté qu'elles sont adaptables et flexibles. Cette méthodologie garantit deux genres d'adaptations: la première est étroitement liée au service et la seconde est collée à l'interface graphique (représentations). Ce qui leur permet de fonctionner de manière correcte, efficace et optimale. Notre préoccupation ne se réduit pas seulement à la réalisation des applications, mais elle cible la rapidité de leur réaction, dans le cas d'un évènement externe qui se produit.

L'étape suivante consiste à introduire le principe de fonctionnement de module de la gestion.

III. 4 Fonctionnement de la plateforme CAHS

La plateforme **CAHS** comporte deux parties, la première représente les modules de la gestion, notamment *Context Manager*, *Controlor-SDC* et *Adaptator*. Par contre la deuxième englobe les applications sensibles au contexte conçues spécifiquement pour l'évaluation de notre proposition de départ. Nous avons suggéré que les applications et la conception de la plateforme proposées sont basées sur les services. Ces données permettent de faciliter la mise à jour et la sensibilité au contexte.

Tout d'abord, le démarrage de **CAHS** conduit à lancer tous les modules de la gestion. Ensuite, les versions qui nécessitent des services doivent s'installer pour structurer les applications que l'utilisateur a besoin dans sa vie quotidienne.

Chaque module de la gestion cible une finalité. Le management du contexte est le garant de la simplification et la gestion de l'exploitation des informations contextuelles. Or le contrôleur-SDC facilite l'accès à l'insertion des situations contextuelles dans l'exécution des applications. Enfin, l'adaptateur dirige l'état des services selon ces situations.

L'objectif commun des modules abordés est la mise en fonction harmonieuse des services appropriés avec la situation contextuelle courante.

L'exploitation de module courtier [3] qui assume la tâche de téléchargement des services est considérée comme une perspective pour une future recherche. Dans cette démarche il sera additionné à notre plateforme.

III.5 Conclusion

Pour récapituler et conclure, ce chapitre est principalement consacré à la description de la plateforme **CAHS** ayant la capacité de l'adaptation et de la gestion des applications qui sont sensibles au contexte et dédiées au secteur de la M-santé. Cette plateforme se construit à base de l'architecture orientée services. De plus, la séparation entre les modules de gestion et les applications permet à cette plateforme de fonctionner de manière rapide. Sans négliger l'avantage de la gestion commune qui opte pour une seule et unique activation d'un code partagé par toutes les applications. Pour l'intégration de contexte dans les applications, nous nous sommes basés sur les services dépendants de la situation contextuelle. Autrement dit, ces services peuvent fonctionner selon la situation de contexte particulière. La méthodologie d'adaptation s'articule autour de la gestion de services (démarrage et arrêt), ces opérations clairement expliquées ne sont que le fruit de la contextualisation.

L'architecture de **CHAS** comporte trois modules principaux de la gestion : le premier est la gestion de contexte qui contient deux parties (le moniteur des éléments de contexte et le noyau de la gestion de contexte). Ceci permet de répartir les tâches de la gestion de contexte afin d'augmenter sa vitesse et sa capacité pour gérer une énorme quantité des informations contextuelles. La seconde est représentée par le contrôleur-SDC qui crée une matrice MC garantissant l'implantation des éléments de contexte dans le management de service. Enfin, l'adaptateur est le seul système accomplissant la mission de décideur de cycle de vie de chaque version du service exploitée pour la mise en place des applications.

Le dernier chapitre sera consacré à l'évaluation de notre modèle **CAHS**. Nous allons tester la performance de la stratégie de l'adaptation selon les différents cas d'étude.

Chapitre 4

Résultats: Evaluation de la plateforme proposée CAHS

Dans ce chapitre, nous allons introduire, l'évaluation des performances de la plateforme proposée pour la gestion des applications de la santé mobile. Notre plateforme d'exécution gère les services médicaux correspondant à un ensemble hétérogène des informations contextuelles interdépendantes soumises à l'environnement d'utilisateur. Le modèle proposé s'appuie également sur plusieurs technologies OSCi et IPOJO dont l'objectif est l'implémentation de nos suggestions.

Dans cette dimension, nous nous intéressons initialement au temps nécessaire pour le lancement de la plateforme, sans négliger le temps de retard pour le démarrage des moniteurs (bundles).

Ensuite, nous nous focalisons sur l'approche d'adaptation en nous bénéficiant des avantages d'OSGi et IPOJO. Nous confirmons par la suite l'efficacité de nos propositions, qui se composent de deux parties : la gestion de contexte et le mécanisme d'adaptation.

Enfin, nous essayons de valider notre méthodologie en comparant nos résultats obtenus avec ceux des autres travaux antérieurs. Nous optons pour une démarche visant l'optimisation des performances de la plateforme **CAHS** qui marquera un tournant dans le domaine de la santé mobile.

IV.1 Notre plateforme CAHS conçu

La finalité de notre travail est d'élaborer une plateforme de la gestion des applications par le biais des terminaux mobiles. Nous nous sommes basés sur le système modulaire afin d'implémenter les applications de la M-health. Il se caractérise par sa reconfiguration automatique et son choix des modules appropriés à la situation courante ou aux ressources du terminal. Ce genre d'application est destiné uniquement aux modules qui subissent des changements brusques.

Pour construire cette plateforme, nous avons répertorié les approches et les méthodes en présentant les diverses étapes : *la gestion de contexte, l'élément de contexte dépendant des autres éléments, les services attachés à la situation de contexte et la gestion des services de la santé* (la détection de la situation médicale, l'annonce de médecin sur la situation patient ...) sur un téléphone portable.

Comme nous l'avons déjà expliqué, l'environnement Android a été exploité pour l'implémentation du prototype **CAHS**. Ce processus gère les éléments de contexte et les services sur le mobile pour assurer la procédure d'adaptation au contexte.

Cette partie regroupe la présentation et l'évaluation de différentes étapes d'exécution des algorithmes. Toute cette série de tests a été effectuée grâce à l'émulateur de téléphone.

Lors de la mise en marche de l'application, la classe **ACTIVITY** (Android) permet de lancer les technologies **OSGI** et **iPOJO** ainsi que les modules de la gestion de notre plateforme. Dès sa mise en service, un groupe des moniteurs évaluent simultanément les éléments de contexte sachant que ces éléments appartiennent à la même classe. Ces moniteurs se chargent de collecter les informations contextuelles afin de les exploiter dans les services nécessaires. **CAHS** dirige ces services suivant la situation contextuelle grâce à la matrice de la classe créée par le contrôleur-SDC.

Dans le domaine de la santé, le temps est un paramètre incontournable car un certain retard de la détection des situations critiques peut entraîner la mort du patient. Pour cela, nous avons évalué les performances de **CAHS** via la mesure du temps de retard pour le démarrage des moniteurs évaluateurs des éléments qui appartiennent à la même classe. Nous avons aussi estimé le temps nécessaire pour adapter les applications.

IV.2 Evaluation des performances de la gestion de contexte

Notre plateforme gère les éléments contextuels et exécute les services mobiles selon ces éléments. Dans les premières évaluations, nous visons à tester la mise en place de notre démarche liée au management de contexte. Tout d'abord, nous avons créé le composant principal de la gestion de contexte et les premiers bundles qui sont les moniteurs. Ensuite, nous avons mesuré le temps suffisant pour démarrer chaque moniteur de CE, sur l'émulateur.

IV.2.1 Test d'expérimentation

Étant donné nous intéressons au domaine de la télémédecine ayant de diverses caractéristiques, notamment les différents utilisateurs (patient ou médecin), l'hétérogénéité des informations médicales (Signal ECG, glycémie, signal EEG) et la grande quantité des données.

Pour cette raison, nous avons réalisé une étude approfondie du management de contexte avec une prise en compte de toutes les probabilités possibles. A ce niveau, nous distinguons trois utilisateurs en axant notre recherche sur deux classes, notre corpus comprend trois cas et nous mène à l'évaluation de notre proposition de départ. Le tableau ci-après (**TAB. IV.1**) résume notre expérimentation:

	Cas 1	Cas 2	Cas 3
Utilisateur	le patient qui souffre de la maladie de diabète.	le patient qui souffre la maladie cardiaque.	Le médecin
Classe	Le contexte de l'environnement de la santé Le contexte de l'appareil	Le contexte de l'environnement de la santé Le contexte de l'appareil	Le contexte de l'appareil

TAB. IV.1 – Cas étudiés

Le gestionnaire de contexte s'appuie sur deux types de modélisation : le premier est orienté d'objet qui régularise les éléments de contexte et la seconde catégorisation renvoie au modèle XML qui décrit l'arbre de la classification contextuelle. Dans notre exemple suggéré, nous avons uniquement considéré les éléments qui appartiennent aux classes (le contexte de l'environnement de la santé et celui de l'appareil). Nous avons mis en œuvre six bundles qui implémentent les moniteurs de ces éléments. Les figures (FIG. IV.1 et FIG. IV.2) délimitent ces paramètres contextuels évalués par les bundles qui représentent leurs moniteurs.

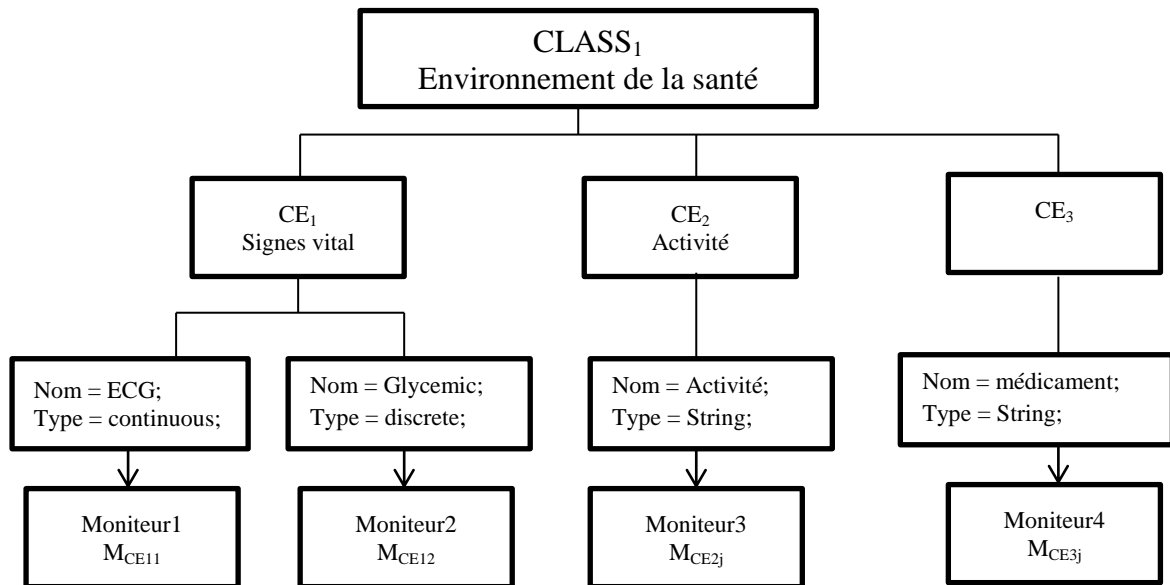


FIG. IV.1 – Eléments appartiennent à la class1

La première classe (Environnement de la santé) introduit les informations liées au l'environnement de la santé. Cette classe inclue trois principaux éléments de contexte, comme montré l'arbre des éléments de contexte (FIG. II.2). Un élément peut avoir les différents paramètres (Nom et Type). Nous distinguons quatre moniteurs, notamment M_{CE11}, M_{CE12}, M_{CE2j} et M_{CE3j} (quel que soit j, l'élément de contexte prend les mêmes paramètres tels que le nom et le type). Chaque moniteur peut évaluer un élément selon leurs paramètres.

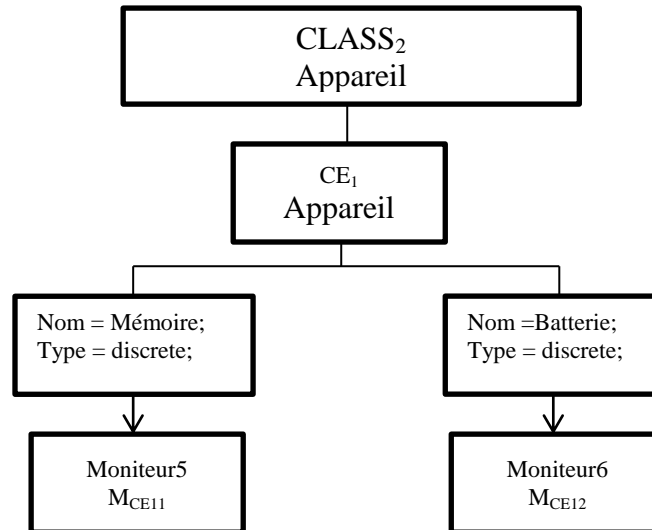


FIG. IV.2 – Eléments appartiennent à la class2

La deuxième classe (contexte d’appareil) exprime les informations qui tournent autour de l’appareil. Cette catégorie est dotée d’un élément de contexte, présenté au niveau de l’arbre classificateur (FIG. II.2). Nous choisissons d’évaluer uniquement deux ressources d’appareil (mémoire et batterie). Donc, nous affirmons l’existence de deux moniteurs, qui sont respectivement le M_{CE11} et le M_{CE12} .

Pour cette expérimentation nous avons exploité les évaluateurs M_{CEij} , expliqués minutieusement dans les figures (FIG. IV.1 et FIG. IV.2). En ce qui concerne les tests ci-dessous nous avons estimé le temps selon l’état des bundles (ACTIVE / DESACTIVE), autrement dit, à partir de moment où elles sont mises en activité. Nous avons mesuré le temps de retard de lancement par la différence entre le moment de déclenchement de premier moniteur et celui de second, sachant que ces moniteurs évaluent les éléments s’inscrivant dans la même classe.

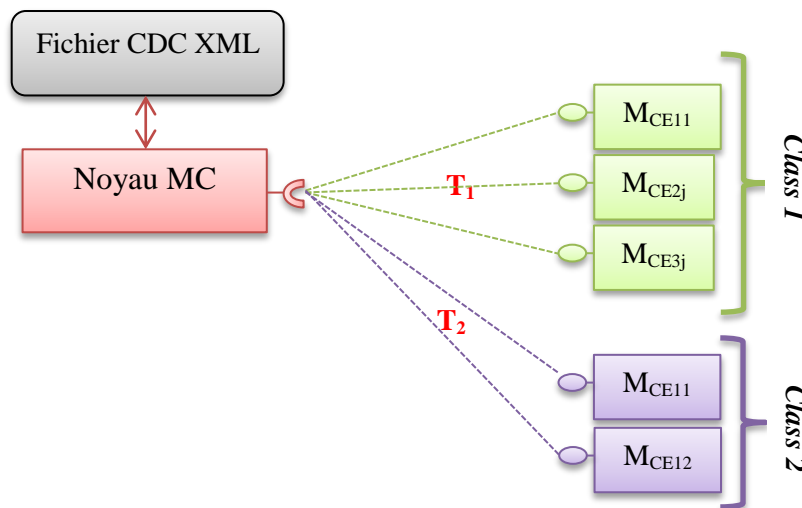


FIG. IV.3 – Structure de test

La figure (FIG. IV.3) introduit la structure de test que nous avons exploitée. Il s’agit d’un module de la gestion (la gestion de contexte) qui dépend d’un moniteur M_{CEij} . Pour chaque classe, nous avons implémenté un ensemble des moniteurs qui doivent être disponibles. Le noyau MC est chargé de la gestion de l’état de ces moniteurs. Ce module est connecté

simultanément aux moniteurs qui évaluent les éléments contextuels regroupés dans la même classe. Pour cette raison, il est indispensable de mesurer la variante temporelle entre le déclenchement de plusieurs moniteurs successivement mis en activité.

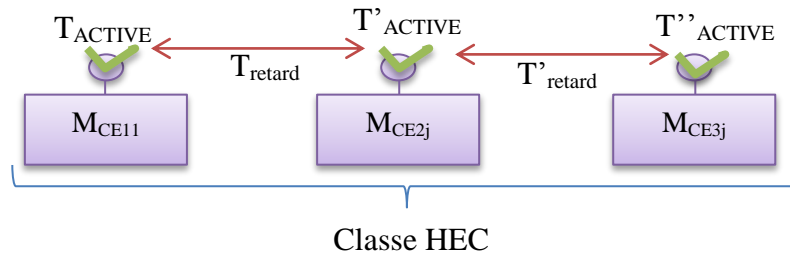


FIG. IV.4 – Méthode de calcul $T_{\text{retard}}(\text{HEC})$

La figure ci-dessus (**FIG. IV.4**) illustre la méthode que nous avons adoptée afin de calculer le temps de retard pour chaque classe sachant que $T_{\text{retard}}(\text{HEC}) = ((T_{\text{retard}}) + (T'_{\text{retard}}))/2$. Ce paramètre nous permet d'argumenter l'efficacité de la méthodologie destinée à la gestion des informations contextuelles caractérisées par son hétérogénéité et sa grande quantité. Ensuite, nous allons étudier et analyser les résultats de ce test.

IV.2.1 Mesures de performances

Premièrement, nous avons estimé le temps nécessaire pour démarrer notre plateforme **CAHS**, c'est-à-dire, nous avons prévu l'intervalle du temps qui existe entre le lancement de la **CAHS** et sa mise en exploitation. Notre expérimentation nous amène à déduire le moment mesuré est en moyenne de 7.450 secondes pour le démarrage de CAHS sur l'émulateur. Ces premiers résultats nous ont prouvé la possibilité de créer une plateforme fondée sur les technologies OSGi et iPOJO à partir du système d'exploitation mobile Android.

Nous introduisons dans la figure (**FIG. IV.5**) les résultats d'installation des bundles (Moniteurs) sur le prototype de **CAHS**. Alors, nous avons remarqué que le temps d'installation d'un bundle varie entre 0.02 et 0.39 secondes. Ce temps est indépendant de la taille de bundle. En somme, ce paramètre n'a aucun impact sur la performance de la gestion de contexte.

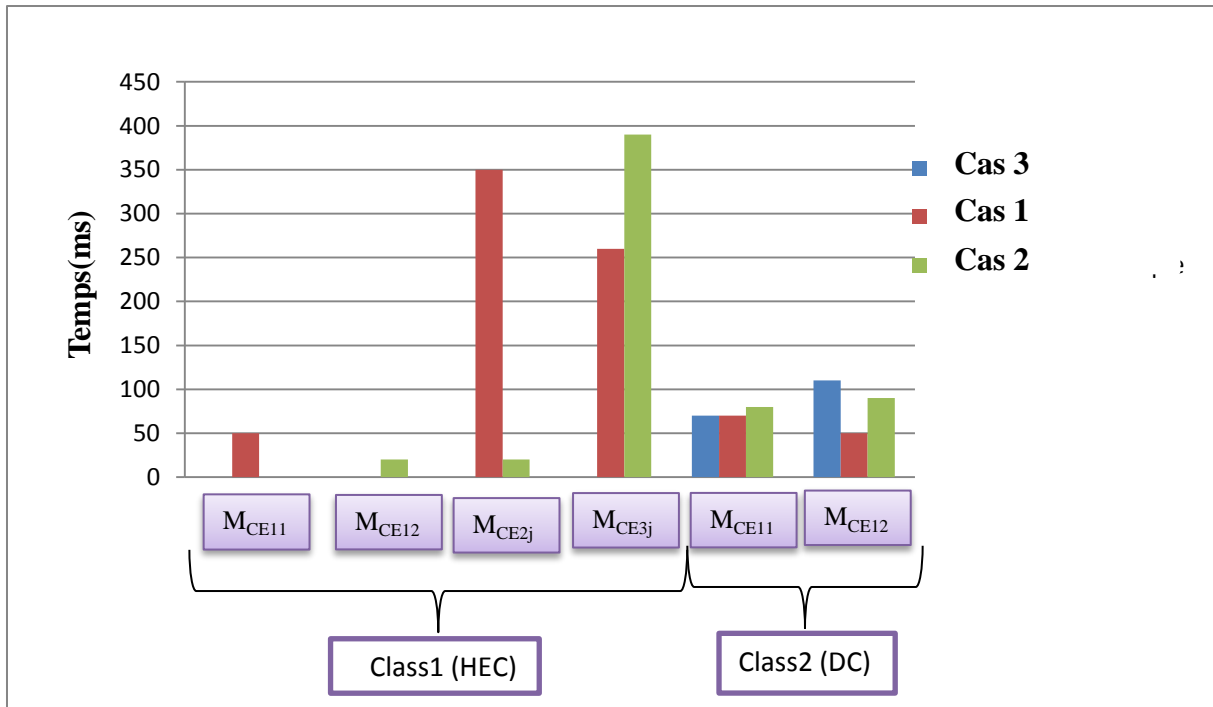


FIG. IV.5 – Temps d'installation des bundles (Moniteurs)

Nous intégrons dans la figure (FIG. IV.6) le résultat d'estimation de temps de retard entre le démarrage des moniteurs évaluant les éléments contextuels qui appartiennent à Class 1. L'objectif de ce test est la vérification de l'existence d'une répercussion qui entrave le fonctionnement de la gestion de contexte, en termes de la qualité et la quantité des informations contextuelles.

Tout d'abord, Class1 est une catégorie principale de domaine de la santé, regroupant toutes les données informant sur l'état médical du patient. Dans ce cadre, nous avons tenu en compte le cas où l'utilisateur est un patient tout en prenant en considération les deux maladies chroniques mentionnées dans le tableau précédent (TAB. IV.1).

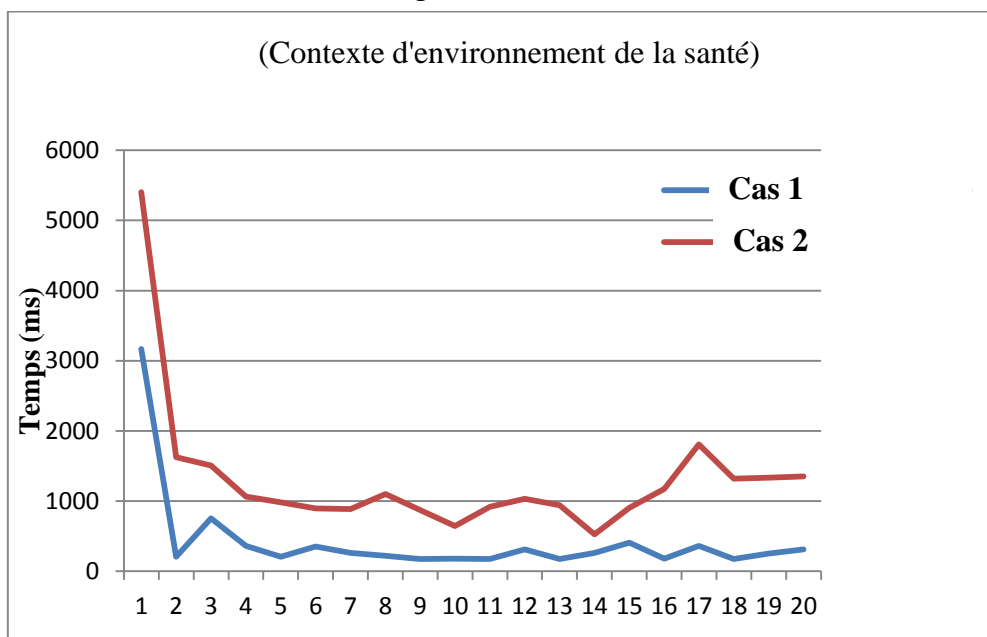


FIG. IV.6 – Temps de retard entre le démarrage M_{CEij} de class1

Dans la figure (FIG. IV.6), nous avons observé que, $T_{\text{retard}}(\text{HEC})$ au début nécessite 3.1 secondes dans le cas 1 et 5.4 secondes dans le cas 2. En revanche, nous constatons que $T_{\text{retard}}(\text{HEC})$ est en diminution constante jusqu'à ce qu'il varie entre 0.17 et 0.74 secondes dans le cas 1 et entre 0.53 et 1.8 secondes dans le cas 2.

La figure (FIG. IV.7) représente le temps de retard qui se manifeste lors de démarrage de chaque moniteur collectant les éléments de la Class2. Cette catégorie contient toutes les informations liées à l'appareil. La raison pour laquelle, nous avons choisi d'implémenter deux moniteurs : le premier évalue la mémoire et le second apprécie la batterie (FIG. IV.2). Dans cette partie, nous avons considéré tous les cas (1, 2 et 3) cités dans le tableau (TAB. IV.1).

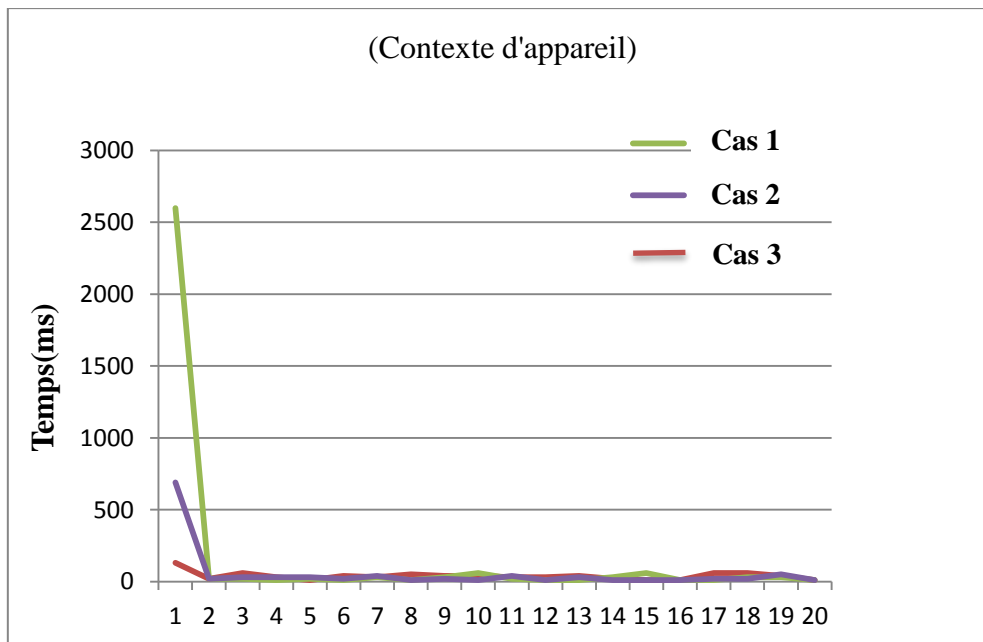


FIG. IV.7 – Temps de retard entre le démarrage M_{CEij} de class2

Selon la figure (FIG. IV.7), le $T_{\text{retard}}(\text{DC})$ prend initialement 2.6 secondes dans le premier cas, 0.67 secondes dans le deuxième cas et 0.13 secondes dans le troisième cas. Ensuite ce temps se réduit jusqu'à ce qu'il diffère entre 0.01 et 0.06 secondes dans le premier cas, entre 0.01 et 0.04 secondes dans le deuxième cas, et entre 0.01 et 0.06 secondes dans le troisième cas.

A travers les courbes (FIG. IV.6 et FIG. IV.7), nous concluons par le constat que le T_{retard} acquiert au début les plus hautes valeurs à cause de l'installation des moniteurs de même que le lancement de CAHS.

La class2 regroupe les éléments ayant des valeurs de type discret. Pour cette raison, $T_{\text{retard}}(\text{DC})$ sont presque identiques dans les trois cas. Par ailleurs, la class1 comprend un élément de type continu, ce qui rend $T_{\text{retard}}(\text{HEC})$ différent dans les deux cas de notre étude. Dans cette optique, nous déduisons que le type des valeurs joue le rôle d'influenceur dans une certaine mesure sur la rapidité d'exécution de ce module. Donc, la qualité des informations exerce un impact sur l'efficacité de la gestion de contexte, mais on ne peut jamais généraliser cette influence de fait qu'elle est très faible et négligeable.

Pour mieux analyser les performances de notre proposition, nous avons estimé la mémoire consommée au niveau de fonctionnement de ce module. Nous introduisons les résultats d'exploitation de la mémoire dans la figure ci-dessus (FIG. IV.8).

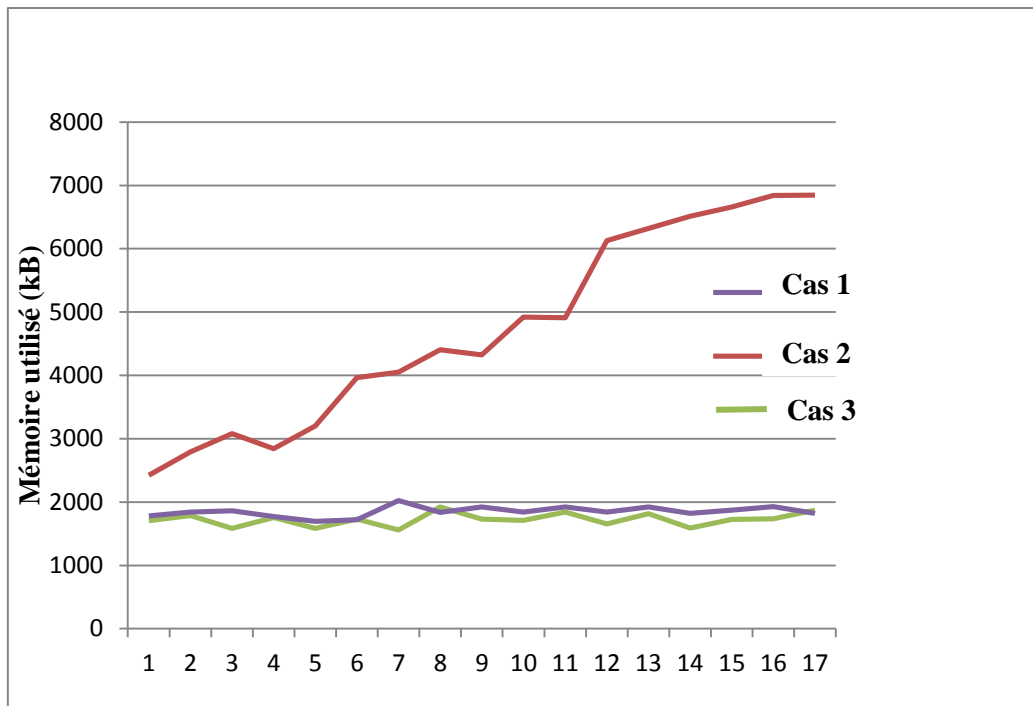


FIG. IV.8 – Mémoire utilisée pour gérer les Moniteurs de CE

A partir de la figure (FIG. IV.8), nous constatons que la consommation de la mémoire est presque identique pour le premier et le troisième cas, sachant que sa consommation est variée entre 1556 et 2026 KB. Contrairement dans le deuxième cas, la quantité exploitée par la mémoire est importante, elle varie entre 2424 et 6848 KB.

Donc, nous déduisons que la consommation importante de la mémoire en fonction de type des valeurs. Dans ce cas le type des valeurs est *continu*. Ce qui nécessite de spécifier la quantité des données sans les contraintes de la détection de la situation médicale en cours du patient.

IV.2.3 Conclusion

Cette première partie d'évaluation nous a permis de prouver la fiabilité de notre suggestion, concernant la gestion du contexte. En général, le fonctionnement de ce module se fait d'une manière rapide, quel que soit le type des valeurs discret ou continu.

Donc, ces premiers résultats nous ont montré que la plateforme Felix (OSGI et iPOJO) fonctionne avec une rapidité suffisante sur le système d'exploitation mobile Android. Pour cela, nous pouvons compter sur notre proposition qui est basée sur la conception des applications modulaires tout en exploitant une architecture orientée services.

La gestion de contexte se distingue par une performance satisfaisante, ce qui nous mène à envisager la proposition d'un nouveau mécanisme d'adaptation reposé sur l'installation et le démarrage des services (bundles), tenant en considération les types de CE.

IV.3 Évaluation de mécanisme d'adaptation au contexte

Nous présentons une plateforme capable de réagir aux changements de contexte. Dans ce cadre, nous nous intéressons à proposer un mécanisme d'adaptation qui rend rapidement les applications réactives à la situation contextuelle. Grâce aux caractéristiques de la plateforme OSGi, les états des services (bundles) peuvent être adaptés en fonction des informations contextuelles. Nous effectuons un ensemble des expériences pour l'évaluation de notre mécanisme suggéré en mettant en œuvre la **CAHS** à l'aide des technologies OSGI et IPOJO.

IV.3.1 Tests de performance

Comme nous l'avons proposé dans le chapitre **III**, les applications peuvent réagir automatiquement au contexte. Ce qui rend leur comportement convenable aux informations contextuelles. Dans ce test, nous distinguons quatre cas d'étude pour l'estimation de l'efficacité de notre plateforme. Le tableau (**TAB. IV.2**) récapitule ces cas:

	Cas 1	Cas 2	Cas 3	Cas 4
Utilisateur	Le patient qui souffre de la maladie de diabète.	Le patient qui souffre de la maladie cardiaque.	Le médecin qui suit le patient diabétique.	Le médecin qui se spécialise en cardiologie.

TAB. IV.2 – Cas étudiés

Dans cette thèse nous nous focalisons sur l'implémentation d'une plateforme qui permet de gérer l'exécution des applications de la surveillance des situations médicales sur des unités portatives. Nous avons conçu une configuration complète, avec tous les modules de gestion et les applications à base des services. Nous avons exposé en détail la conception de ces applications dans le chapitre **III**. De ce fait, nous prenons en considération les éléments suivants:

- *Les moniteurs*: Ces modules permettent d'évaluer les éléments qui appartiennent aux classes (le contexte d'environnement de la santé et le contexte d'appareil). Ces moniteurs sont minutieusement présentés dans les figures (**FIG. IV.1** et **FIG. IV.2**).
- *Les services*: Chaque service possède plusieurs versions (implémentations) sachant que chaque version dépend de la situation contextuelle créée par un ensemble des éléments de contexte. Les applications considérées dans cette évaluation sont réparties en termes des fonctionnalités représentées dans la section **III.3.4**.

Dans la figure (**FIG. IV.9**), nous détaillons l'explication de la méthodologie suivie pour créer les versions de service d'affichage. L'adaptateur se charge de gérer l'état des versions selon la situation contextuelle. Cette dernière est déduite à partir de la matrice de classe (Mc), générée par le contrôleur-SDC via des états des moniteurs.

Sans écarter, les autres services notamment S (analyse) et S (communication) qui se basent sur le même principe de la conception des versions de service d'affichage.

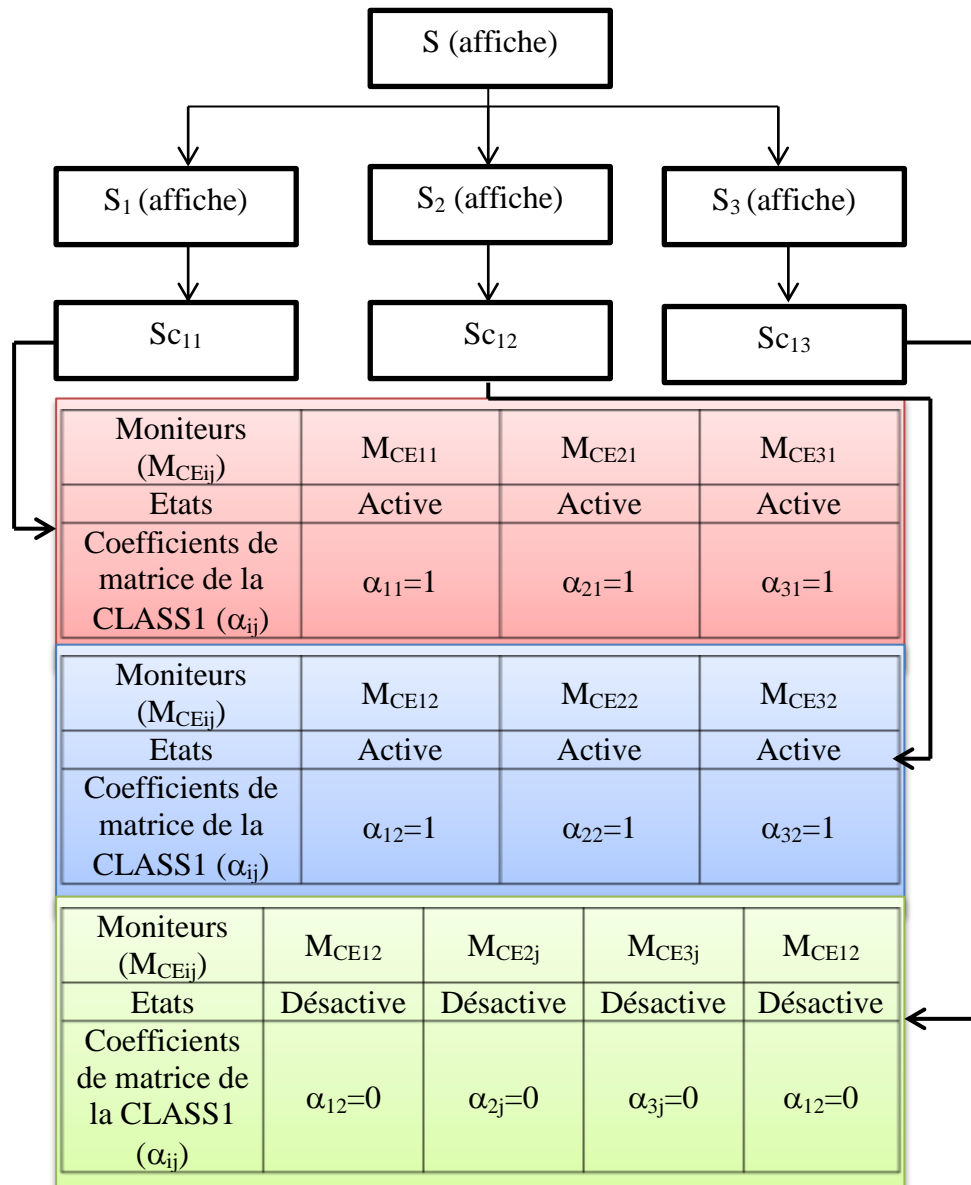


FIG. IV.9 – Versions de service avec leur situation contextuelle

Dans cette partie, nous prouvons la fiabilité de notre plateforme CAHS en estimant la rapidité de processus de l'adaptation et de la réaction des applications. Cette évaluation est représentée par le temps nécessaire afin que toutes les versions appropriées soient mises en activité. Donc, nous avons mesuré le temps indispensable pour le démarrage des versions de services formant une application spécifique.

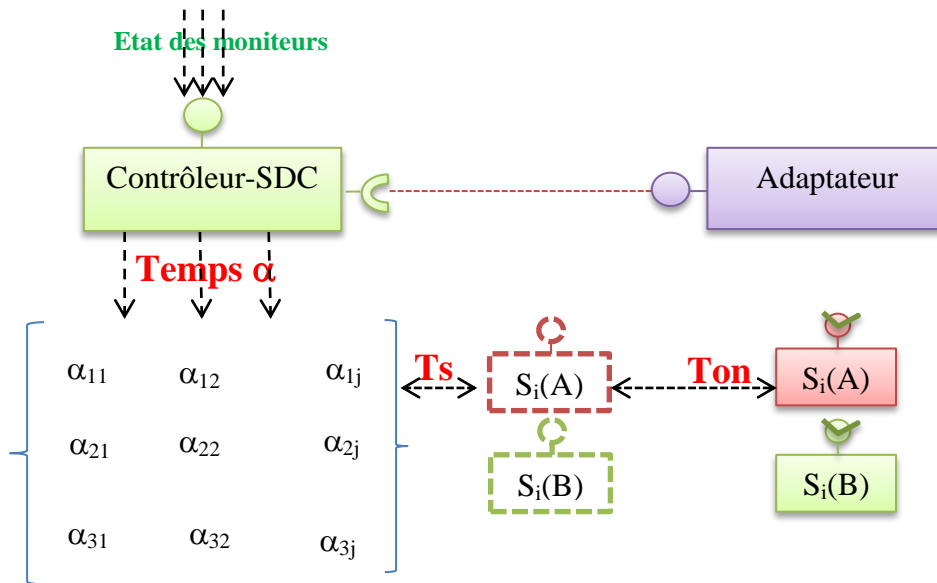


FIG. IV.10 – Configuration de l'expérimentation

La figure (FIG. IV.10) présente la configuration de l'expérimentation que nous exploitons dans cette section. Il s'agit des modules (Contrôleur-SDC et Adaptateur) chargés de la gestion de l'état des versions de services. Pour atteindre ce but, il y a plusieurs opérations obligatoires notamment la création de la matrice MC, le choix de la version adéquate, la décision sur l'état de la version et le changement de cet état, sachant que chaque opération prend un certain temps. Pour cette raison, il est indispensable de mesurer la période temporelle de chaque opération pour savoir laquelle des opérations exerce un impact sur la rapidité de l'adaptation. Par ailleurs, il existe d'autres paramètres qui doivent être estimés pour étudier leur influence sur la rapidité de l'adaptation. Ces paramètres sont respectivement la taille des services et le type des valeurs des éléments. Dans cette optique, nous étudions et analysons les normes nécessaires afin d'évaluer la fiabilité et la performance de la plateforme CAHS.

IV.3.2 Déroulement de l'expérience

Le but de nos expérimentations est d'estimer le temps total de la réaction d'une application quand la situation contextuelle (Sc_{ij}) est créée. Nous mesurons le moment à partir d'évaluation des éléments contextuels jusqu'à la fin de l'adaptation.

Pour réaliser ces tests, il est indispensable d'avoir un prototype complet. Pour ceci, tous les modules de la gestion doivent être fonctionnés complètement. La gestion de contexte évalue les informations contextuelles grâce aux modules d'évaluation (moniteurs).

Ensuite, le contrôleur-SDC permet de créer la matrice de classe à partir des états de ces moniteurs (*ACTIVE / DESACTIVE*). Ces dernières sont exploitées par l'adaptateur pour décider sur les états des différentes versions de services. Par conséquent, l'ordre de fonctionnement des modules de cette plateforme joue un rôle majeur car il assure un meilleur fonctionnement de CAHS et des applications.

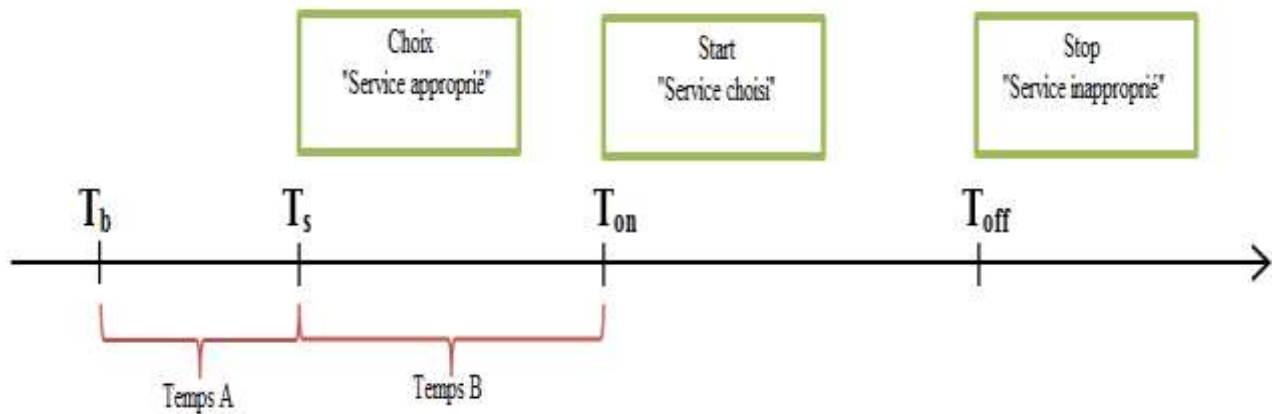


FIG. IV.11 – Déroulement de l’expérience

Pour démontrer la capacité et l’efficacité de la plateforme **CAHS**, nous avons décidé de présenter les événements survenus pendant son exécution, comme nous l’avons illustré dans la figure (FIG. IV.11). Le temps **Tb** considère le moment de la surveillance de contexte. Le paramètre **Ts** est le moment où le service est sélectionné, **Ton** définit le temps de la mise en activité de service sélectionné et **Toff** représente le moment où le service inapproprié est désactivé.

Temps A - Temps A exprime la période temporelle entre l’évaluation des éléments et la sélection de service approprié. Donc, le Temps A comporte le moment requis (α) pour la création de la matrice MC. Il est représenté par l’équation suivante:

$$\text{Temps A [Si (T)]} = Ts [Si (T)] - Tb \dots\dots\dots \text{IV.1}$$

Temps B - Il s’agit de moment requis pour démarrer la version du service choisie. Ce temps est introduit par l’équation suivante:

$$\text{Temps B [Si (T)]} = Ton [Si (T)] - Ts [Si (T)] \dots\dots\dots \text{IV.2}$$

Tout d’abord, les versions de service ont été sélectionnées selon la situation contextuelle courante, pendant la période Temps A [Si (T)], avec $i = 1, \dots, m$. Ces versions choisies ont été démarrées pendant la période Temps B [Si (T)], avec $i = 1, \dots, m$. Dans notre évaluation, nous considérons trois versions du service d’affichage et deux versions du service d’analyse, afin que $m = 3$ pour le premier service et $m = 2$ pour le deuxième.

III. 3.3 Estimations de performances

Dans la figure (FIG. IV.12), on compare les résultats obtenus en mesurant les temps (A), (B), et Time_adapt. Le temps A est presque identique dans tous les cas. Pour cette raison, nous concluons que ce paramètre n’est pas influencé par la taille et le nombre des services de même que les types des valeurs (continu, discret, String,...).

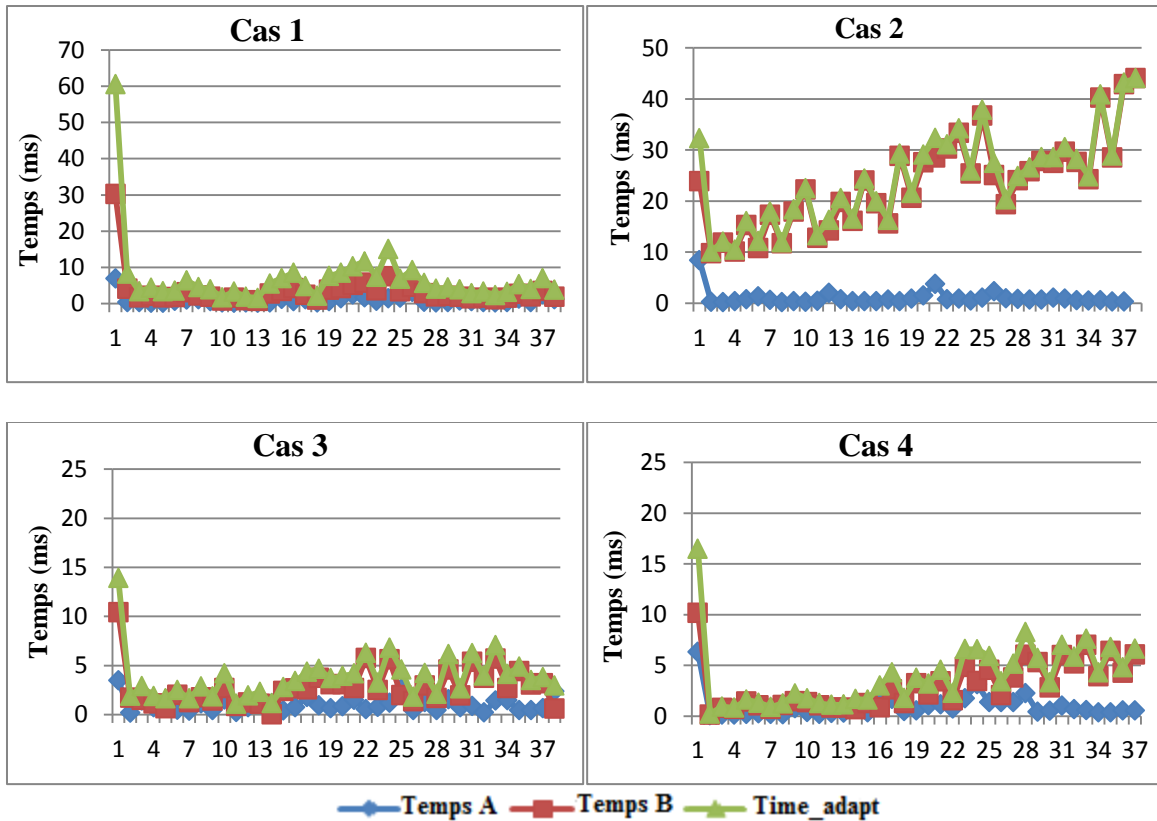


FIG. IV.12 – Moyenne de Temps A, Temps B et Time_adapt

Par contre, nous constatons que la variante de Temps B dans les quatre cas est observable. Pour tous les cas précédents, nous avons clairement remarqué que le Temps B est supérieur au Temps A.

A partir de ces résultats, nous déduisons que la plateforme CAHS prend un délai précis pour le lancement de service même si ce service était déjà installé.

Notre approche de l'adaptation est basée sur la gestion de l'état des services (INSTALL, ACTIVE et DESACTIVE). En effet, on peut estimer le temps de l'adaptation en fonction des temps (A) et (B). Le temps d'adaptation est exprimé par l'expression suivante:

$$\text{Time_adapt [Si (T)]} = \text{Temps A [Si (T)]} + \text{Temps B [Si (T)]}$$

La figure (FIG. IV.12), montre que Time-adapt suit le même chemin de Temps B. on peut ainsi conclure que Time-adapt et Temps B sont d'une certaine mesure identique.

	Cas 1	Cas 2	Cas 3	Cas 4
La moyenne de Time_adapt avec α (ms)	3,417	24,831	4,103	3,922
La moyenne de Time_adapt sans α (ms)	3,344	24,782	4	3,891
Les écarts types (%)	0,052	0,035	0,073	0,022

TAB. IV.3 – Moyenne de Time_adapt avec et sans α

Par ailleurs, nous estimons le temps (α) de la création de la matrice MC. L'observation de tableau (TAB. IV.3) nous conduit à exclure tout impact qui se résulte de paramètre (α) sur le Time_adapt. Bref, le Time_adapt est significativement lié au Temps B.

En ce qui concerne la taille de la version, nous introduisons le Time_adapt de chaque version avec sa taille dans la figure (FIG. IV.13). Nous notons que Time_adapt mesuré dans chaque

version ne solidarise pas directement de sa taille. Dans ce cadre, nous déduisons que le processus de l'adaptation d'une version n'est point influencé par sa taille.

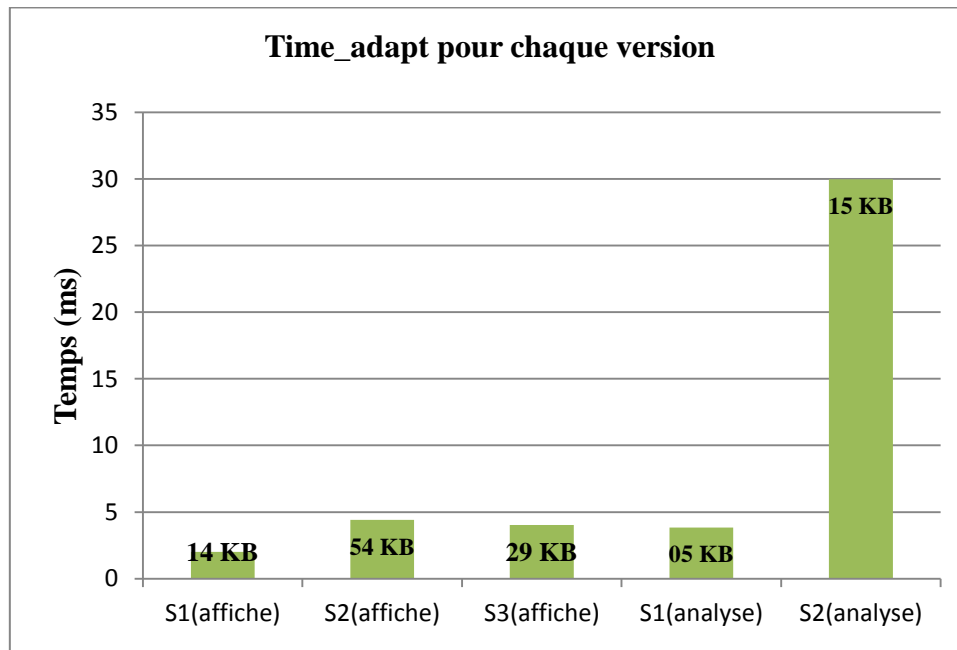


FIG. IV.13 – Time_adapt selon la taille de version

Comme nous l'avons suggéré dans la section III.2, notre approche de l'adaptation est basée sur la gestion des services qui constituent une application spécifique. Pour cette raison, nous essayons d'estimer le temps requis pour démarrer ces services tenant en considération le nombre des versions de services et les types des éléments de contexte. Dans cette dimension, nous exploitons le nouveau paramètre appelé β , qui indique la variante temporelle de démarrage des différentes versions de services qui forment la même application. De plus, nous récapitulons les caractéristiques des cas (TAB. IV.4) de notre étude dans le tableau (TAB. IV.3); sachant que chaque cas utilise un certain nombre de service et les différents types des éléments contextuels.

	Cas 1	Cas 2	Cas 3	Cas 4
Application	S(affiche) S(analyse) S(connection)	S(affiche) S(analyse) S(connection)	S(affiche) S(connection)	S(affiche) S(connection)

TAB. IV.4 – Caractéristiques de chaque cas

Dans la figure (FIG. IV.14), nous introduisons les valeurs moyennes de paramètres β dans les quatre cas. Nous observons initialement que la valeur moyenne de β prend respectivement 0.11 ms et 0.17 ms pour les cas (3) et (4). Par contre dans les deux autres cas (premier et deuxième), l'application nécessite environ 0.2 ms afin d'être disponible pour l'exploitation.

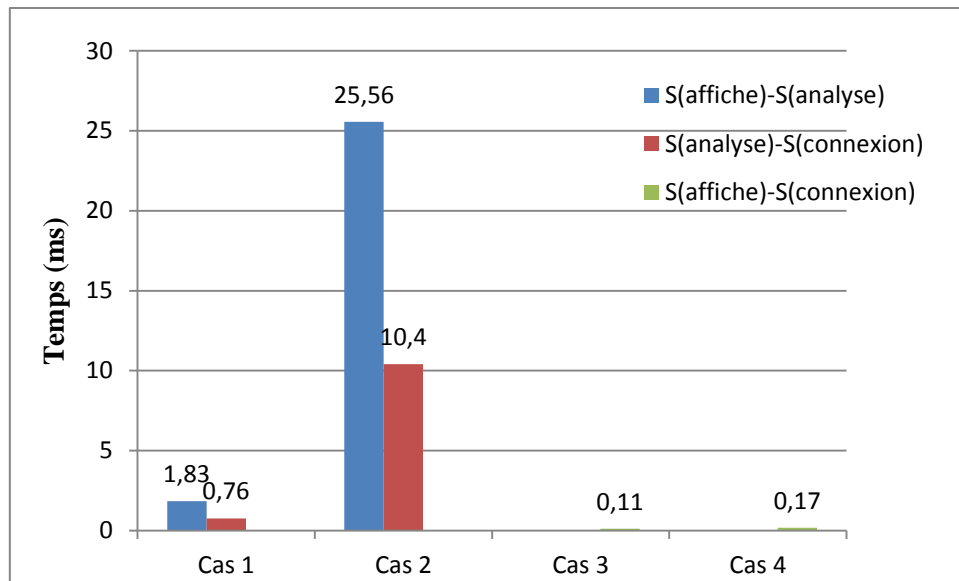


FIG. IV.14 – Moyenne de β dans tous les cas

Dans le deuxième cas, nous constatons que S (analyse) exige environ 25 ms en moyenne pour son démarrage depuis le lancement de S (affiche). Dès le déclenchement de S (analyse), S (connexion) requiert environ 11 ms en moyenne pour son fonctionnement.

Par conséquent, l'application demande environ 36 ms afin de démarrer la totalité des services adéquats. Ainsi, elle requiert cet intervalle temporel pour la découverte de la situation médicale de patient et l'envoi au médecin

Par contre, dans le premier cas, l'application prend environ 3 ms pour lancer tous les services appropriés, sachant que ces applications ont le même nombre des services (TAB. IV.4) dans les deux premiers cas. La différence entre eux réside dans le type des éléments contextuels. Dans la première situation, tous les éléments sont de type discret. En revanche, dans la deuxième situation, il y a deux types des éléments exploités (discret et continu).

Ces résultats obtenus nous permettent de conclure que le temps de la disponibilité des applications est influencé par le type continu des éléments ce qui présente un ensemble de valeurs appelées *les échantillons*. Pour cela, nous essayons de mesurer le paramètre β en fonction de nombre d'échantillons.

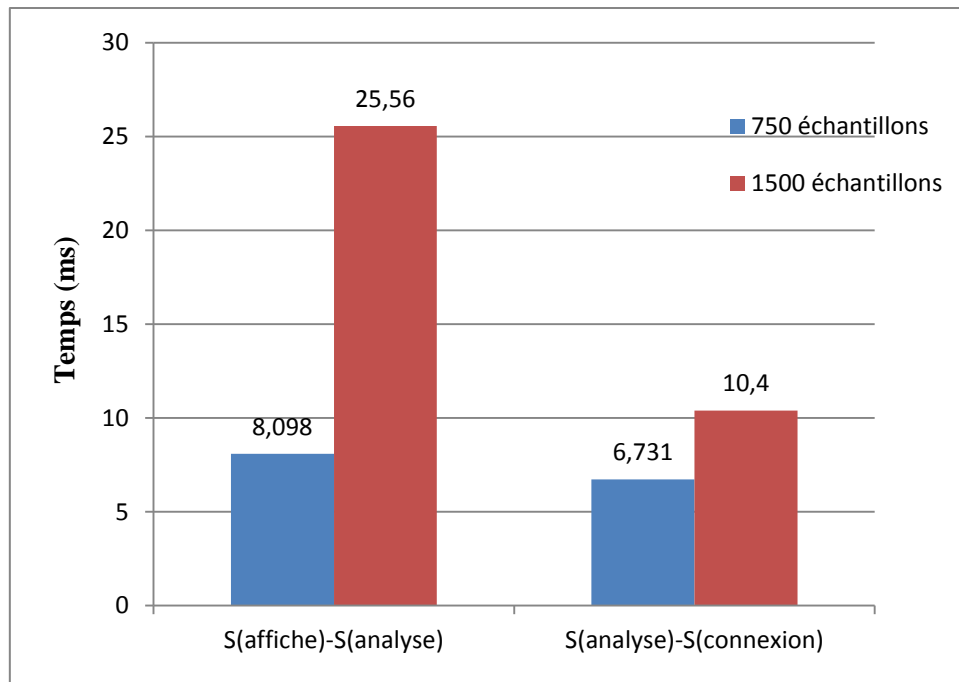


FIG. IV.15 – Moyenne de β selon les différents nombres d'échantillons

La figure (FIG. IV.15) démontre que le temps nécessaire pour découvrir la situation médicale et l'envoyer au médecin varie entre 15 ms et 36 ms pour 750 échantillons et 1500 échantillons respectivement. La moyenne de β accroît avec l'augmentation du nombre d'échantillons. Ces résultats affirment que le temps d'adaptation est influencé par le nombre d'échantillon, lorsque le type d'élément de contexte est continu.

Pour les quatre cas, l'écart type de temps d'adaptation est de 10%. La dispersion est prise en considération, mais dans des limites raisonnables. En dépit de ces données, nous pouvons optimiser la performance de notre plateforme par la spécification d'un nombre approprié d'échantillons requis par les services pour obtenir les meilleurs résultats, afin que ce nombre soit diminué le plus possible. Une autre solution peut avoir lieu en exploitant les services qui permettent de filtrer ou de transférer les données médicales.

Pour une évaluation perfectionniste de l'efficacité de notre plateforme **CHAS**, nous nous intéressons à l'estimation de la mémoire consommée pour l'adaptation des applications après le processus de la gestion des informations contextuelles. La figure (FIG. IV.15) expose l'évolution de la mémoire exploitée dans les quatre cas faisant l'objet de notre étude.

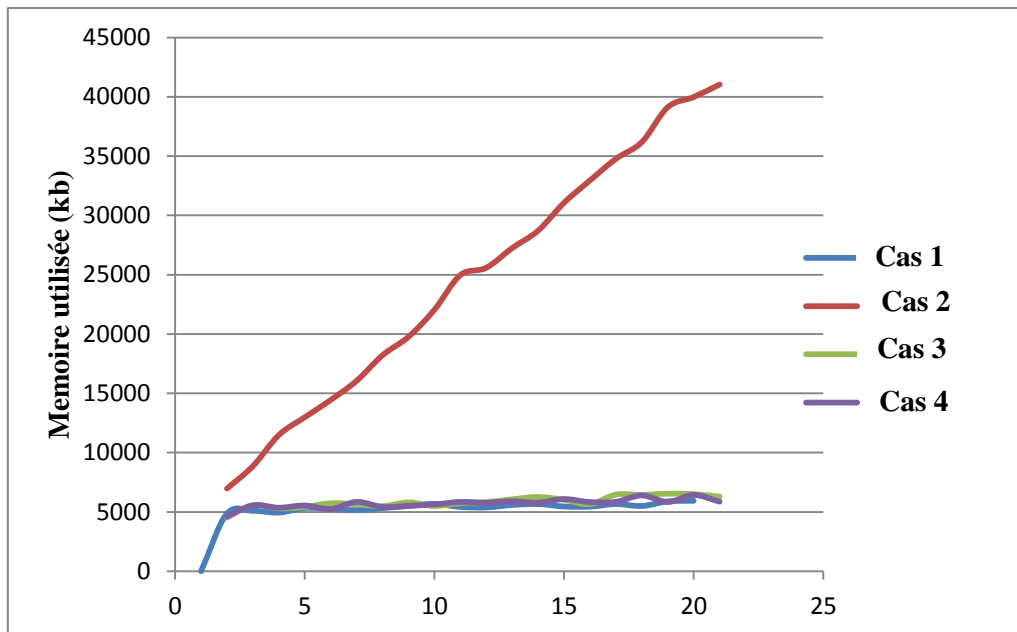


FIG. IV.16 – Mémoire exploitée pour le processus d'adaptation.

À travers de la figure (**FIG. IV.16**), le mécanisme d'adaptation occupe un immense espace de la mémoire pour gérer les services, plus de 40000 kb. Cette capacité augmente de manière exponentielle dans le cas 2. Néanmoins, nous observons des stabilités de consommation de la mémoire, ainsi que des petites variations dans trois cas (premier, troisième et quatrième).

Selon ces mesures, nous constatons qu'il existe un impact clair du type continu des éléments contextuels sur la consommation de la mémoire. Nous interprétons ces résultats par la machine virtuelle qui n'ait pas le moment nécessaire de libérer la mémoire entre deux essais successifs, si bien que nous prenons en considération cet aspect afin que les opérations des adaptations s'effectuent à des intervalles trop longs.

IV.2.3 Conclusion

Cette deuxième partie nous a permis de présenter une étude approfondie des différents paramètres dont l'impact perturbe la vitesse de procédure de l'adaptation. Par une suite logique, nous avons tenu en considération le moment indispensable dans le processus de l'adaptation c'est-à-dire la création de la matrice de la classe, la sélection de la version appropriée de service, le démarrage des versions de services et le moment dans lequel l'application a été adaptée.

Partant de cette estimation, nous inférons que la taille et le nombre des services n'ont aucune répercussion sur le déroulement de l'opération d'adaptation. En effet, l'influenceur clef de la réactivité d'adaptation et de la consommation de mémoire est le type des valeurs des éléments contextuels. Cependant, les résultats de cette évaluation restent dans des limites raisonnables.

IV.3 Validation des résultats

Le but de cette section réside dans la confirmation des résultats obtenus par des mesures effectués sur les processus des algorithmes implémentées selon le modèle proposé auparavant (chapitre III).

La plateforme **CAHS** a mise en place sous l'environnement Android pour la gestion des différents services destinés à la surveillance de plusieurs maladies et aux divers profils d'utilisateurs. La complexité du modèle liée particulièrement à l'accumulation des données et

à la diversité des informations contextuelles. Ce qui contraint la gestion de ces informations permettant d'adapter les services afin de constituer une certaine application.

La validation de notre méthodologie se bâtit sur la comparaison de nos propositions avec celles des autres travaux antérieurs en termes des paramètres d'évaluation, tel que le temps d'adaptation, la mémoire exploitée, la vie de la batterie.

Alors, nous nous centrons sur la suggestion de [3] pour valider nos résultats obtenus. Cette proposition introduit une plateforme CATS qui est capable de gérer les services dédiés au domaine de transport. Par ce résultat nous prouvons l'efficacité et la fiabilité des méthodes que nous avons proposées, tout en considérant l'approche d'adaptation qui est l'opération clefs de cette évaluation.

Donc, la **CAHS** réalise une multitude de tâches, par le fait de suivre plusieurs pathologies sur un seul téléphone portable et dans un temps réduit.

Par ailleurs, cette évaluation met en évidence les points forts de notre plateforme par rapport aux travaux liés aux systèmes sensibles au contexte. La complexité et la particularité de notre plateforme résident dans la surveillance de plusieurs pathologies sur le terminal mobile.

IV.4 Evaluation

Notre plateforme proposée s'articule autour de deux points fondamentaux: d'abord le processus de la gestion de contexte prenant en compte les différentes informations médicales, puis la méthodologie suivie pour adapter les services de la santé en vue d'offrir un meilleur comportement d'application. Chaque service implémente plusieurs versions afin que chacune d'elles corresponde à une situation bien déterminée.

Dans ce cadre, les applications fonctionnées sur cette plateforme sont basées sur l'assemblage des services appropriés en exploitant le mécanisme développé tout au long de cette thèse. Ainsi, elles peuvent être plus flexibles et légères, la particularité qui permet de les reconfigurer sans interrompre leur exécution. Pour cette raison, les applications de la santé mobile proposées sont capables de suivre plusieurs maladies et d'être exploitées par les différents utilisateurs (patient, médecin, infirmière, ...). La **CAHS** assure la gestion des applications et la diminution de leur nombre sur le terminal mobile ainsi que la minimisation des tâches effectuées grâce au système de la gestion commune.

IV.4 Conclusion

Nous avons implémenté le prototype de la plateforme **CAHS** sur le système d'exploitation mobile Android. Par ailleurs, nous avons prouvé qu'était possible d'effectuer des applications modulaires sur le terminal mobile grâce à une architecture orientée services. Nous avons débutés par l'exemple d'un utilisateur, Jean, qui avait besoin de suivre l'état de sa santé par les applications mobiles de son téléphone : détection de la situation médicale, affichage des informations médicales, contact avec le médecin, etc. le besoin de cet utilisateur est l'adaptation de ses applications à l'environnement qui change constamment, sans que cela soit perturbant pour l'exécution des applications.

Les performances de modèle **CAHS** sont requises selon les périodes temporelles requis pour effectuer des certaines tâches. Cette évaluation estime la rapidité de deux processus fondamentaux: le premier est la gestion des informations qui les regroupent selon certaines relations. Le second consiste à l'adaptation fondée sur le changement de l'état des services en fonction des situations contextuelles courantes. On s'intéresse en particulier à l'existence d'une masse de données diverses destinées à être gérées par des algorithmes implémentés sur les ressources limitées et conçus pour la détection de différentes situations contextuelles afin d'adapter les services. Par cette démarche nous cherchons à juger l'efficacité de notre système en fonction de plusieurs environnements d'utilisateurs possibles.

Ainsi, nous nous sommes focalisés sur la gestion de contexte afin de permettre la surveillance simultanée des éléments contextuels qui assemblent une relation spécifique. Nous avons également distingué différents cas pour tester la méthodologie implémentée l'adaptation en démarrant les services appropriés selon la situation contextuelle. Nos résultats nous ont permis de prouver la fiabilité et l'efficacité de notre approche à plusieurs reprises des utilisateurs, pour rassurer l'utilisateur par le mécanisme d'adaptation même lorsque les informations médicales se qualifient par leur énorme quantité et hétérogénéité.

Conclusion générale et perspectives

Notre thèse aborde la conception et l'implémentation des applications de la M-santé gérées selon les situations contextuelles. Elle a pour objectif de proposer une plateforme sensible au contexte et implémentée sur les terminaux mobiles. Son but réside dans la gestion et l'adaptation des services en fonction de contexte.

Tout d'abord, nous avons présenté les besoins imposés par la conception de système régi par le contexte. Les éléments cités sont essentiellement liés, d'une part, à la gestion de contexte (l'évaluation et la modélisation des informations contextuelles) sur des terminaux aux ressources limitées (téléphones mobiles) et à l'exploitation de ces informations dans les applications. D'autre part, à la capacité d'adaptation des applications et des services de la santé mobile à ces informations. Trois aspects fondamentaux gravitent autour de notre recherche : le système sensible au contexte, les applications de la santé mobiles et leur adaptabilité au contexte. Ensuite, nous avons offert un aperçu global sur les concepts généraux, l'architecture de système sensible au contexte, les travaux de recherche antérieurs qui sont en cohérence avec notre proposition.

Notre démarche s'articule autour de la création et de l'évolution de système sensible au contexte, tenant en considération la complexité et les finalités des services de la santé mobile. Le principe en est la résolution du problème de l'adaptation et la gestion d'exécution selon les informations contextuelles pour assurer le meilleur fonctionnement des applications. Dans le cadre de cette thèse, notre travail est subdivisé en trois parties :

La première partie se consacre à l'étude de la notion de contexte pour développer le système informatique et identifier les éléments contextuels clefs à la santé mobile. Nous avons suggéré un arbre de la classification de ces éléments selon des caractéristiques spécifiques.

Par ailleurs, la modélisation de contexte est basée sur deux modèles de contexte :

Le premier modèle, compte sur un encodage XML chargé de la description de l'arbre de la classification. Cette opération occupe une place primordiale dans le management de contexte. Le second s'appuie sur le modèle orienté objets permettant de réguler les éléments contextuels. Notre proposition vise l'extraction des données les plus abstraites en termes de la structure de données et de la sémantique.

La deuxième partie introduit la conception des applications adaptables aux situations contextuelles préexistantes. Pour la conception de nos applications, nous nous sommes focalisés sur les applications modulaires pour garantir la reconfiguration et la flexibilité des applications de la santé mobile. Nous avons exploité la notion de contexte dans les applications grâce aux services dépendants de la situation de contexte afin de l'intégrer dans l'exécution des applications. Dans cette optique, nous avons implémenté plusieurs versions pour chaque service afin de couvrir toutes les situations contextuelles possibles. Notre méthodologie est construite sur les applications à base de services, en sélectionnant les versions des services appropriées aux situations courantes.

L'approche d'adaptation proposée se distingue par sa cohérence, simplicité et efficacité, ce qui la permet de se perfectionner dans le domaine qui nécessite la rapidité de la réaction des applications.

La troisième partie esquisse l'architecture de notre plateforme gérant les services de la santé mobile (CAHS). Au-delà des applications, notre recherche aborde trois autres modules de gestion (la gestion de contexte, contrôleur-SDC et l'adaptateur). Notre plateforme assume la gestion d'exécution des applications selon les situations actuelles.

Ensuite, nous avons évalué les performances de nos méthodologies suivies pour la conception et le développement de la plateforme CAHS, grâce à son implémentation tout en exploitant les technologies OSGI et iPOJO. Notre jugement de valeur a prouvé que CAHS fonctionne de manière harmonieuse et leurs résultats sont satisfaisants sur le système d'exploitation mobile "Android". Nous avons étudié la réactivité de la plateforme dans les divers cas d'utilisation. En outre, nous avons estimé l'espace de mémoire exploité dans les processus du management de contexte et de l'adaptation.

Notre thèse se rapporte à proposer une solution pour la conception des applications sensibles au contexte y compris la santé mobile. Nous considérons la technologie de la santé mobile comme un axe fondamental dans l'optimisation des applications mobiles. En effet, les utilisateurs mobiles, notamment ceux qui souffrent des pathologies chroniques, auront l'avantage de profiter de l'aide apportée par ce type d'application. Plus loin de cette qualité, elles peuvent même sauver des vies, en détectant l'état alarmant à travers la surveillance permanente des situations médicales.

Étant donné la diversité du contexte où ce type d'applications s'exécute, il est indispensable de l'adapter rapidement. Notre proposition répond au problème lequel elle veut résoudre: accompagner le suivi de l'état médical d'utilisateur quel que soit la maladie dont il est sujet et dans toutes les situations possibles. La plateforme CAHS réagit parfaitement à ce besoin en se servant des modules de gestion du contexte et d'adaptateur. La CAHS se caractérise par la capacité de traitement, analyse et surveillance de plusieurs types de données médicales.

Ainsi, elle sera exploitée par de différentes catégories d'utilisateurs et exercera un contrôle performant d'une multitude de maladies

À travers ce manuscrit, nous présentons une solution adéquate à notre domaine étudié, néanmoins, notre travail reste inachevé :

D'une part, notre sujet peut être un point de départ pour de nouveaux travaux et suggestions dans le but d'améliorer notre recherche ou d'ouvrir une nouvelle piste dans le même thème.

D'autre part, la concrétisation de notre proposition devient une nécessité indépassable pour qu'elle soit réellement exploitée sur un téléphone portable

Dans cette partie, nous présentons les perspectives de cette thèse axées sur les points suivants:

Téléchargement de nouvelles fonctionnalités. Nous visons à améliorer notre travail par l'addition d'un autre module qui s'occupe de télécharger de nouvelles fonctionnalités indispensables en tenant compte des issues de la sécurité de la vie privée et la disponibilité de la connexion.

Application universelle de la santé mobile. Le développement auquel nous avons pensé vise un objectif à moyen terme qui consiste à concevoir une application globale pour la santé mobile, ce qui permet de surveiller la plupart des maladies. Selon cette dimension, cette application serait la boussole qui guide un utilisateur souffrant des pathologies qui nécessitent un suivi permanent.

Coopération entre les services. Le domaine de notre étude est basé sur la santé mobile. Pour cette raison, nous envisagerons le calcul de pourcentage qui définit l'impact des différents signaux vitaux lorsque l'un influence l'autre ; à l'instar d'un patient qui est sujet à plusieurs maladies, ainsi, la glycémie a une répercussion directe sur les battements du cœur.

Dans ce cadre, nous visons l'implémentation d'un nouvel algorithme permettant de coopérer entre les services pour assurer une analyse minutieuse de la situation médicale de patient atteint de plusieurs maladies.

Références

- [1] <https://mobilemarketing.fr/2019/07/12/30-milliards-dapplications-ont-ete-telechargees-au-deuxieme-trimestre-2019/>
- [2] Chen, Guanling and Kotz, David, "A Survey of Context-Aware Mobile Computing Research" (2000). Open Dartmouth: Faculty Open Access Articles. 3212. <https://digitalcommons.dartmouth.edu/facoa/3212>
- [3] D. Popovici. Gestion du contexte pour des applications mobiles dédiées aux transports. Thèse de doctorat. Université de Valenciennes et du Hainaut-Cambresis, 2012.
- [4] Yazid Benazzouz. Découverte de contexte pour une adaptation automatique de services en intelligence ambiante. Autre. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2011. Français. ffNNT : 2011EMSE0617ff. fftel-00733013f
- [5] Anind K. Dey and Gregory D. Abowd, Towards a better understanding of context and context-awareness, In HUC '99 : Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, Springer-Verlag, 1999, pp. 304–307.
- [6] M. Weiser, The computer for the 21st century, Scientific American, 1991.
- [7] M. Friedewald and O. Raabe, Ubiquitous computing: An overview of technology impacts, Telematics and Informatics, Vol. 38, pp. 55-65, May 2011.
- [8] Antonio J. J., Miguel A. Z., Antonio F. G. S., "An internet of things-based personal device for diabetes therapy management in ambient assisted living (AAL).", "Personal and Ubiquitous Computing.", Vol. 15., pp. 431–440., April 2011.
- [9] Parisa . R., Alex . M., " A Survey on Ambient-Assisted Living Tools for Older Adults.," IEEE Journal of Biomedical and Health Informatics., Vol. 17.,pp. 579-590., May 2013.
- [10] Peter. W., Andreas. S., Javier Parada. O., Michael. K., Sebastian. R., Birgitta. K., Torsten. D., Aygul. G., "openAAL - the open source middleware for ambient-assisted living (AAL).,"
- [11] Eun-Young J., Jong Tak K., Jaeyoung S., Dong Kyun P., " Development of U-healthcare monitoring system based on context-aware for knowledge service.," Multimedia Tools and Applications., Vol. 74., pp. 2467–2482., April 2015.
- [12] Satyanarayanan. M., Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (MCS), 2010.
- [13] Satyanarayanan. M., Fundamental challenges in mobile computing. "In Proceedings of the 5th annual ACMsymposium on Principles of distributed computing.," pp. 1–7., 1996.
- [14] <http://www.google.com/landing/now/>. (2013). Accessed: 2013-05-23.
- [15] Agnes Gruenerbl, Venet Osmani, Gernot Bahle, Jose C Carrasco, Stefan Oehler, Oscar Mayora, Christian Haring, and Paul Lukowicz. 2014. Using Smart Phone Mobility Traces for the Diagnosis of Depressive and Manic Episodes in Bipolar Patients. In Proceedings of the 5th Augmented Human International Conference. Kobe, Japan.
- [16] Predrag Klasnja, Sunny Consolvo, David W. McDonald, James A. Landay, and Wanda Pratt. 2009. Using Mobile & Personal Sensing Technologies to Support Health Behavior Change in Everyday Life: Lessons Learned. In AMIA'09. San Francisco, CA, USA.
- [17] Margaret E Morris, Qusai Kathawala, Todd K Leen, Ethan E Gorenstein, Farzin Guilak, Michael Labhard, and William Deleeuw. 2010. Mobile Therapy: Case Study Evaluations of a Cell Phone Application for Emotional Self-Awareness. Journal of Medical Internet Research 12, 2 (2010).
- [18] Lucy Yardley, Leanne Morrison, Laura Dennison, Sharon Lin, Charlie Hargood, Scott Lloyd, Derek Johnston, Marie Johnston, Paul Roderick, Elizabeth Murray, Susan Michie, Paul Little, Peter Smith, and Mark Weal. 2013. Uptake And Usage Of Digital Self-Management Interventions: Triangulating Mixed Methods Studies Of A Weight Management Intervention. In Medicine 2.0 '13 World Congress on Social Media and Mobile Apps. London, UK.

- [19] World Health Organization. 2010. Hidden cities: unmasking and overcoming health inequities in urban settings. Technical Report. Centre for Health Development, and United Nations Human Settlements Programme Un-habitat.
- [20] Bret Hull, Vladimir Bychkovsky, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Yang Zhang, Hari Balakrishnan, and Samuel Madden. 2006. CarTel: A Distributed Mobile Sensor Computing System. In *SenSys'06*. Boulder, CO, USA.
- [21] Suhas Mathur, Tong Jin, Nikhil Kasturirangan, Janani Chandrasekaran, Wenzhi Xue, Marco Gruteser, and Wade Trappe. 2010. Parknet: Drive-By Sensing of Road-Side Parking Statistics. In *MobiSys'10*. San Francisco, CA, USA.
- [22] Prabal Dutta, Paul M. Aoki, Neil Kumar, Alan Mainwaring, Chris Myers, Wesley Willett, and Allison Woodruff. 2009. Demo Abstract: Common Sense - Participatory Urban Sensing Using a Network of Handheld Air Quality Monitors. In *SenSys'09*. Berkeley, CA, USA.
- [23] S. Akter, P. Ray, mHealth-an ultimate platform to serve the unserved, *MIA Yearbook of Medical Informatics 2010*, pp- 75-81.
- [24] mHealth for Development: The opportunity of mobile technology for healthcare in developing world, <http://www.vitalwaveconsulting.com/insights/mHealth.htm>. Last Access May 12, 2010.
- [25] Istepanian R. Introduction to the Special Section on M-Health: Beyond Seamless Mobility and Global Wireless Health-care Connectivity. *IEEE Trans Inf Technol Biomed* 2004;8(4):405-13.
- [26] Michael P. The Case for mHealth in Developing Countries. *Mobilizing Markets: Special Edition of MIT Innovations Journal for the GSMA Mobile World Congress 2009*, Cambridge: MIT Press; 2009. p. 153-68.
- [27] Bardram JE, Baldus H, Favela J. Pervasive computing in hospitals. In: Bardram JE, Mihailidis A, Wan D. Boca Raton, FL, editors. *Pervasive Computing in Healthcare*. CRC Press; 2007. p. 49-77.
- [28] Akter S, D'Ambra J, Ray P. User perceived services quality of mHealth Services around the world. In the proceedings of the 18 th European Conference on Information Systems, Pretoria, South Africa, 2010.
- [29] Pascoe, J. Adding generic contextual capabilities to wearable computers. In *Proceedings of the Second International Symposium on Wearable Computers*, Pittsburgh, PA, USA, 19–20 October 1998; pp. 92–99.
- [30] Salber, D.; Dey, A.K.; Abowd, G.D. *Ubiquitous Computing: Defining an Hci Research Agenda for an Emerging Interaction Paradigm*; Georgia Institute of Technology: Atlanta, GA, USA, 1998.
- [31] Brown, P.J. Triggering information by context. *Pers. Technol.* 1998, 2, 18–27.
- [32] Ryan, N. *Mobile Computing in a Fieldwork Environment: Metadata Elements*; University of Kent: Canterbury, Kent, UK, 1997.
- [33] Anind K, D., Gregory D, A.: Towards a Better Understanding of Context and Context-Awareness. In: *Handheld and Ubiquitous Computing*, vol. 1707, pp. 304—307. Springer, (2001)
- [34] Baldauf, M., Dustdar, S., Rosenberg, F.: A Survey on Context-Aware Systems. *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 2, No. 4, pp.263-277. (2007)
- [35] Loke, S.: *Context-Aware Pervasive Systems: Architectures for New Breed of Applications*. Auerbach Publications, ISBN 0-8493-7255-0 (2006)
- [36] V. Adzic, H. Kalva, B. Furht, A survey of multimedia content adaptation for mobile devices, *Multimedia Tools and Applications*, Vol. 51, pp 379–396, January 2011.
- [37] J. Nam ; Yong Man Ro ; Y. Huh ; M. Kim, Visual Content Adaptation According to User Perception Characteristics, *IEEE Transactions on Multimedia*, vol. 7, pp. 435 – 445, June 2005.
- [38] S. Ardon, P. Gunningberg, B. Landfeldt, Y. Ismailov, M. Portmann, A. Seneviratne, MARCH: A Distributed Content Adaptation Architecture, *International Journal of Communication Systems*, vol.16, February 2003.
- [39] Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, and Frank Leymann, Service-oriented computing : State of the art and research challenges, *Computer* 40 (2007), 38–45.
- [40] Context-Aware Service Composition in a Home Control Gateway. In: *International Conference on Pervasive Services*, pp. 223-231. IEEE Xplore. Istanbul (2007)
- [41] L. Srinivasan and J. Treadwell, An overview of service-oriented architecture, web services and grid computing, HP Software Global Business Unit (2005).

- [42] D. Popovici, M. Desertot, S. Lecomte, Seamless Context Adaptation on a Service-Oriented Framework, International Conference on Mobile Wireless Middleware, Operating Systems, and Applications, Vol. 65, pp 207-220, 2012.
- [43] M. Miraoui ; Ch.Tadj, Ch. Ben Amar, Dynamic Context-Aware Service Adaptation in a Pervasive Computing System, 2009 Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Sliema, Malta, 11-16 Oct. 2009.
- [44] T. CHAARI, F. LAFOREST et A. FLORY, «Adaptation des applications médicales à des contextes multiples», Journées Francophones d’Informatique Médicale, Laboratoire d’informatique en images et systèmes d’information LIRIS UMR CNRS 5205, Lille, France, Mai 2005.
- [45] T. Lemlouma, N. Layaida, Context-Aware Adaptation for Mobile Devices, IEEE International Conference on Mobile Data Management, Berkeley, CA, USA, 2004.
- [46] T. Chaari, F. Laforest, A. Flory, Adaptation des applications au contexte en utilisant les services WEB, 2ème Journées Francophones Ubiquité et Mobilité, UbiMob’05, May 2005, Grenoble, France. pp.111-118.
- [47] Pung, H.K.; Tao, G.; Xue, W.W. Context-aware Middleware for Pervasive Elderly Homecare. IEEE Sel. Areas Commun. 2009, 27, 510–524.
- [48] Wei, E.J.Y.; Chan, A.T.S. CAMPUS: A middleware for automated context-aware adaptation decision making at run time. Pervasive Mob. Comput. 2013, 9, 35–56.
- [49] Kang, J.; Park, S. Context-Aware Services Framework Based on Semantic Web Services for Automatic Discovery and Integration of Context. Int. J. Adv. Comput. Technol. (IJACT) 2013, 5, doi:10.4156/ijact.vol5.issue4.52.
- [50] Forkan, A.; Khalil, I.; Tari, Z. CoCaMAAL: A cloud-oriented context-aware middleware in ambient assisted living. Future Gener. Comput. Syst. 2014, 35, 114–127.
- [51] Forkan, A.R.M.; Khalil, I.; Ibaida, A.; Tari, Z. BDCaM: Big Data for Context-aware Monitoring—A Personalized Knowledge Discovery Framework for Assisted Healthcare. IEEE Trans. Cloud Comput. 2015, 99, doi:10.1109/TCC.2015.2440269
- [52] Wu, G.; Zhang, H.X.; Qiu, M.K.; Ming, Z.; Li, J.Y.; Qin, X. A decentralized approach for mining event correlations in distributed system monitoring. *J. Parallel Distrib. Comput.* 2012, 73, 330–340.
- [53] Z. Cao., X. Zhang., W. Zhang., X. Xie., J. Shi., H. Xu, » A Context-aware Adaptive Web service Composition Framework, » Proceedings of IEEE International Conference on Computational Intelligence & Communication Technology, Ghaziabad, India, 13-14 Feb. 2015.
- [54] Étude sur l’usage du Smartphone, » Décembre 2017.
- [55] B. M.C. Silva., J. J.P.C. Rodrigues., Isabel de la T. Díez., M. López-Coronado., K. Saleem, “ Mobile-health: A review of current state in 2015,” Journal of Biomedical Informatics, 56, 2015, PP. 265–272.
- [56] Xin Li, Martina Eckert, José-Fernán Martínez, and Gregorio Rubio.,” Context Aware Middleware Architectures: Survey and Challenges,” sensors.,Vol. 15.,pp. 20570-20607., 2015.
- [57]V. Krishnamurthy, Versioning Based Dynamic Reconfiguration for SOA Applications, Research Journal of Applied Sciences, Engineering and Technology 9(11): 926-934, 2015.
- [58]Yuvraj Sahni, Jiannong Cao, Xuefeng Liu and Hong Kong, MidSHM: A Flexible Middleware for SHM Application Based on Service-Oriented Architecture, IEEE Symposium on Service-Oriented System Engineering (SOSE), Oxford, UK, 2016, DOI: 10.1109/SOSE.2016.43.
- [59] J. Vizcarrondo, J. Aguilar, E. Exposito and A. Subias, ARMISCOM: self-healing service composition, Service Oriented Computing and Applications (11), pp 345–365, 2017.
- [60] S.Brahler, “Analysis of the Android architecture, Karlsruher Institut für Technologie KIT,” 2010. <<http://os.ibds.kit.edu/>>.

- [61] <https://www.osgi.org/>
- [62] C. Escoffier. iPOJO : Un modèle à composant à service flexible pour les systèmes dynamiques. Génie logiciel [cs.SE]. Université Joseph-Fourier - Grenoble I, 2008. Français.
- [63] Matko Kuna, Hrvoje Kolarić, Iva Bojić, Mario Kušek, and Gordan Ježić, Android/osgi-based machine-to-machine context-aware system, Proceedings of the 11th International Conference on Telecommunications ConTEL 2011 / Plank, Thomas, 2011.
- [64] M. Khemaja and T. Mastour, "Semantic Reasoning for Enabling Mobility and Context-Awareness: Application to an OSGi Based Intelligent Tutoring System », TEEM '14, October 01 - 03 2014, Salamanca, Spain, <http://dx.doi.org/10.1145/2669711.2669925>.
- [65] Schilit, W.N. A System Architecture for Context-Aware Mobile Computing; Columbia University: New York, NY, USA, 1995.
- [66] Brown, P.J.; Bovey, J.D.; Xian, C. Context-aware applications: From the laboratory to the marketplace. IEEE Pers. Commun. 1997, 4, 58–64.
- [67] Pascoe, J.; Ryan, N.; Morse, D. Human Computer Giraffe Interaction: HCI in the Field. In Proceedings of the 2nd International Symposium on Wearable Computers (ISWC 98), Glasgow, UK, 19–20 October, 1998.
- [68] Flachsbarth, D.F. All gadget and no representation makes jack a dull environment. In Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments, Palo Alto, CA, USA, 23–25 March 1998.
- [69] Ward, A.; Jones, A.; Hopper, A. A new location technique for the active office. IEEE Pers. Commun. 1997, 4, 42–47.
- [70] Hull, R.; Neaves, P.; Bedford-Roberts, J. Towards situated computing. In Proceedings of the Digest of Papers First International Symposium on Wearable Computers, Cambridge, MA, USA, 13–14 October 1997; pp. 146–153.
- [71] Miraoui, M.: Software Architecture for pervasive computing: context modeling and dynamic adaptation of services. Thesis of PhD. Montreal: École de technologie supérieure, (2009)
- [72] Dana, P.: Context Management for mobile applications dedicated to transportation. Thesis of PhD: Engineering Sciences. Valenciennes (2012)
- [73] M. Mohsin, S., Natalia Díaz, R., Johan, L., Iván, Porres.: A Framework for Context-Aware Applications for Smart Spaces. In: Smart Spaces and Next Generation Wired/Wireless Networking, Vol. 6869, pp. 14–25. Springer, (2011)
- [74] Strang, Thomas and Linnhoff-Popien, Claudia (2004) *A Context Modeling Survey*. In: Workshop Proceedings. First International Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp 2004, Nottingham, England, September 7, 2004, 2004-09-07, Nottingham, UK.
- [75] Dey, A.K., Salber, D., Abowd, G.D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Anchor article of a special issue on context-aware computing in the Human-Computer Interaction (HCI) Journal, Volume 16 (2-4), pp. 97-166, (2001)
- [76] Yamabe, T., Takagi, A., Nakajima, T.: Citron: A Context Information Acquisition Framework for Personal Devices. Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05), (2005)
- [77] Schilit, B.; Adams, N.; Want, R. Context-aware computing applications. In Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA, 8–9 December 1994; IEEE Computer Society: Washington, DC, USA, 1994; pp. 85–90.
- [78] Venezia, C., Lamorte, L.: Pervasive ICT Social Aware Services enablers. 14th Int. Conf. Intelligence in Next Generation Networks (ICIN). Berlin, Germany (2010)
- [79] Knappmeyer, M., Kiani, S.L., Fra, C., Moltchanov, B., Baker, N.: ContextML: A light-weight context representation and context management schema. 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC). Modena, Italy (2010)

- [80] Ferscha, A., Hechinger, M., Riener, A., Schmitzberger, H., Franz, M., Rocha, Md.S., Zeidler, A.: Context-Aware Profiles. International Conference on Autonomic and Autonomous Systems, ICAS '06. Silicon Valley, CA (2006)
- [81] <https://wdi.supelec.fr/appliouaibe/Cours/XML>.
- [82] HENRICKSEN, K., INDULSKA, J., AND RAKOTONIRAINY, A. Generating Context Management Infrastructure from High-Level Context Models. In *Industrial Track Proceedings of the 4th International Conference on Mobile Data Management (MDM2003)* (Melbourne/Australia, January 2003), pp. 1–6.
- [83] Cheverst, K.; Mitchell, K.; Davies, N. Design of an object model for a context sensitive tourist guide. *Comput. Graph.* **1999**, *23*, 883–891.
- [84] <https://www.supinfo.com/cours/2OOP/chapitres/01-introduction-modelisation-objet>.
- [85] Studer, R.; Benjamins, V.R.; Fensel, D. Knowledge engineering: Principles and methods. *Data Knowl. Eng.* 1998, *25*, 161–197. [111]
- [86] Perera, C.; Zaslavsky, A.; Christen, P.; Georgakopoulos, D. Context aware computing for the internet of things: A survey. *IEEE Commun. Surv. Tutor.* 2014, *16*, 414–454.
- [87] Zhang, D.; Huang, H.; Lai, C.F.; Liang, X.; Zou, Q.; Guo, M. Survey on context-awareness in ubiquitous media. *Multimed. Tools Appl.* 2013, *67*, 179–211.
- [88] Aiello, F.; Fortino, G.; Gravina, R.; Guerrieri, A. A java-based agent platform for programming wireless sensor networks. *Comput. J.* 2011, *54*, 439–454.
- [89] McDonald, H.; Nugent, C.D.; Finlay, D.D.; Moore, G.; Burns, W.; Hallberg, J. Assessing the impact of the homeml format and the homeml suite within the research community. *J. UCS* 2013, *19*, 2559–2576.
- [90] Gonçalves, B.; Pereira Filho, J.G.; Andreão, R.V. ECGWARE: An ECG Markup Language for Ambulatory Telemonitoring and Decision Making Support. In *Proceedings of the International Conference on Health Informatics*, Funchal, Portugal, 28–31 January 2008; pp. 37–43.
- [91] Rialle, V.; Lamy, J.-B.; Noury, N.; Bajolle, L. Telemonitoring of patients at home: A software agent approach. *Comput. Meth. Prog. Biol.* 2003, *72*, 257–268.
- [92] Henricksen, K.; Indulska, J. A software engineering framework for context-aware pervasive computing. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications*, Orlando, FL, USA, 14–17 March 2004; pp. 77–86.
- [93] Zhang, D.; Gu, T.; Wang, X. Enabling context-aware smart home with semantic web technologies. *Int. J. Hum. Welf. Robot. Syst.* 2005, *6*, 12–20.
- [94] Xu, C.; Cheung, S.C. Inconsistency detection and resolution for context-aware middleware support. *SIGSOFT Softw. Eng. Notes* 2005, *30*, 336–345.
- [95] Bouchard, B.; Giroux, S.; Bouzouane, A. A smart home agent for plan recognition of cognitively-impaired patients. *J. Comput.* 2006, *1*, 53–62.
- [96] Chen, L.; Nugent, C.; Mulvenna, M.; Finlay, D.; Hong, X.; Poland, M. Using event calculus for behaviour reasoning and assistance in a smart home. In *Smart Homes and Health Telematics*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5120, pp. 81–89.
- [97] Okeyo, G.; Chen, L.; Wang, H. Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes. *Future Gener. Comput. Syst.* 2014, *39*, 29–43.
- [98] Chen, H.; Perich, F.; Finin, T.; Joshi, A. SOUPA: Standard ontology for ubiquitous and pervasive applications. In *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, MA, USA, 22–26 August 2004; pp. 258–267.
- [99] Ye, J.; Stevenson, G.; Dobson, S. USMART: An unsupervised semantic mining activity recognition technique. *ACM Trans Interact. Intell. Syst.* 2014, *4*, 16:1–16:27.
- [100] Denis Conan, Romain Rouvoy, and Lionel Seinturier, *COSMOS : composition de noeuds de contexte*, *Technique et Science Informatiques (TSI)* **27** (2008), 1189–1224.
- [101] Harry Chen, *An intelligent broker architecture for pervasive context-aware systems*, Ph.D. thesis, University of Maryland, Baltimore County, 2004.
- [102] Colombe Herault, *Adaptabilite des services techniques dans le modele a composants*, These de doctorat, Universite de Valenciennes et du Hainaut Cambresis, 2005.

- [103] Carlos Parra, Xavier Blanc, and Laurence Duchien, *Context awareness for dynamic service-oriented product lines*, 13th International Software Product Line Conference SPLC 2009 (John McGregor and Dirk Muthig, eds.), vol. 1, 08 2009, pp. 131–140.
- [104] Papazoglou, M.P., et Georgakopoulos, D. (2003). Introduction : Service-oriented computing. *Communication of the ACM* 46(10), 24–28.
- [105] Papazoglou, M.P. (2003). Service-oriented computing: concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE 2003)*, pp. 3–12.
- [106] Richard S. Hall, Karl Pauls, Stuart McCulloch, and David Savage, *Osgi in action : Creating modular applications in java*, Manning Publications, 2010
- [107] DK Barry, *Web services, service-oriented architectures, and cloud computing*, 2003, <https://www.service-architecture.com/>.
- [108] *Universal description discovery and integration* <https://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm>, 2002. [109] *Web services description language (wsdl) version 2.0* <http://www.w3.org/TR/wsdl20/>, juin 2007.
- [110] M. Adasy, « Le diabète », 2016.
H. Labud, N. Jenni et N. Marcoz, « Définition du diabète Conseils pour la prise en charge des patients diabétiques », 2015.
- [112] Organisation mondiale de la santé, « rapport mondiale sur le diabete »,.
- [113] Association Luxembourgeoise du Diabète asbl, « Le guide du diabète », 2018.
- [114] M. Juneau, «Les troubles cardiaques, maladies cardio-vasculaires (angine et crise cardiaque) », « PasseportSanté.
- [115] http://e-cardiogram.com/ecg-lexique_theme.php?id_th=3&id_lex=251&PHPSESSID=
- [116] Azumio Inc, Moniteur Fréquence Cardiaque, <https://play.google.com/store/apps/details?id=si.modula.android.instantheartate>, 2018.
- [117] PVDApps, Rythme Cardiaque Plus – Cardiofréquence, <https://play.google.com/store/apps/details?id=com.dungelin.heartrate>, 2018.
- [118] Chiarugi F, Sakkalis V, Emmanouilidou D, Krontiris T, Varanini M, et al, “Adaptive threshold qrs detector with best channel selection based on a noise rating system,” *Proc. IEEE Computers in Cardiology*, 2007, pp. 157–160. doi:10.1109/CIC.2007.4745445.
- [119] Christov II, “Real time electrocardiogram QRS detection using combined adaptive threshold,” *Biomedical Engineering Online*, 2004, pp. 3- 28.
- [120] Zidelmal Z, Amirou A, Adnane M, Belouchrani A, “QRS detection based on wavelet coefficients,” *Computer Methods and Programs in Biomedicine*, 2012, 107: 490–496.
- [121] Chouakri SA, Bereksi-Reguig F, Taleb-Ahmed A, “QRS complex detection based on multi wavelet packet decomposition,” *Applied Mathematics and Computation*, 2011, 217: 9508–9525.
- [122] Li C, Zheng C, Tai C, “Detection of ECG characteristic points using wavelet transforms,” *IEEE Transactions on Biomedical Engineering*, 1995, 42: 21–28.
- [123] Arzeno N, Deng Z, Poon C, “Analysis of first derivative based QRS detection algorithms,” *IEEE Transactions on Biomedical Engineering*, 2008, 55: 478–484.
- [124] R. SLAMA, G. MOTTÉ, A.LEENHARDT, C.SE BAG, « Aide Mémoire de Rythmologie », Édition Flammarion, Paris, 04/2003. ISBN 10: 2257124065.

Publications

- Revues internationales

- Siham, B and R. Merzougui. “A Framework Survey for Context Aware Adaptation,” *International Journal of Computer Science and Information Security*, vol.14, no. 11, pp. 732-738, 2016.
- Siham Belhadi and Rachid Merzougui; Design and implementation of a context-aware health service platform (CAHS); *International Journal of Electrical and Computer Engineering (IJECE)*; December 2019; Vol. 9; pp. 4993-5005; ISSN: 2088-8708, URL : DOI: 10.11591/ijece.v9i6.pp4993-5005

- Conférence internationale

- Belhadi Siham , et Merzougui Rachid, Context Element Monitor Dedicated to Health Domain, in the *proceeding of: 1st international Conference on Electrical Engineering*, 2019, Algeria.

Résumé :

Actuellement, la création et le développement des applications et services mobiles se propagent de plus en plus en ouvrant de nouvelles pistes dépendantes aux particularités de contexte et à la situation de l'utilisateur. La majorité des applications disponibles rencontrent des lacunes qui entravent leur fonctionnement. Etablies en dehors de leur contexte, elles deviennent inadaptées aux conditions d'exécution. C'est cette constatation qui nous a suscité à créer un modèle unificateur qui tend à gérer les services de manière simultanée pour former une application solide.

Les travaux portent, dans un premier temps, sur une étude approfondie des applications mobiles et de leur comportement, en fonction de contexte d'exécution. Nous considérons le cas des applications dédiées à la santé, conçues pour accompagner et assister un patient dans sa vie quotidienne (suivi médicale, sécurité permanente, applications des analyses et traitement à distance, etc.). Dans un second temps, nous proposons un modèle dynamique et sensible au contexte, permettant la gestion de contexte ainsi que l'adaptation afin d'harmoniser les potentialités du service développé avec son environnement et pour une meilleure utilisation des ressources dans des milieux très évolutifs. Nous nous focalisons sur le côté réactif, permettant d'adapter le comportement des applications à leurs conditions d'utilisation.

Le prototype de notre plateforme CAHS (Context-Aware Health Services) proposée dans le cadre de cette thèse est implémenté sous la plateforme Java IPOJO et basé sur l'adaptation orientée services SOA, elle offre la sensibilité des applications au contexte et permet d'assurer un meilleur fonctionnement de l'ensemble des applications dédiées à la santé. Notre démarche tend vers une nouvelle génération de plateformes mobiles intégralement ouvertes et unanimes en toutes les catégories d'applications dimensionnant la santé mobile.

Mot-clé— Contexte ; Système sensible au contexte ; Architecture orientée service ; Santé mobile ; IPOJO.

Abstract :

Currently, the conception and the development of mobiles applications and services are still growing by the fact of more strong new horizons and services related to the context and the user situation.

The majority of available applications face the set of constraints which obstruct their execution. This is because designed applications are unsuitable for the conditions of execution. This makes it interesting to use a unified conception and execution model that can manage these applications.

Our work in a first step, presents an in-depth study of mobiles applications and their behavior, depending on the execution context. We consider the case of applications dedicated to health, designed to accompany and to assist a patient in his daily life (medical monitoring, permanent security, applications of remote analysis and treatment, etc.). In a second step, we propose a dynamic and context-aware model, allowing the context manager and an adaptation in order to harmonize the potentialities of the developed service with its environment and to assure a better use of resources in much evolved environments. We focus on the reactive side, allowing adapting the behavior of applications to their use conditions.

In this Thesis, the prototype of our Context-Aware Health Services (CAHS) platform is implemented under the Java IPOJO platform and based on the SOA. This platform provides the sensitivity of applications to context and ensures the proper functioning of all applications dedicated to health. The goal of our approach is to move towards a new generation of fully open and unified mobile platforms in all categories of applications that scale mobile health.

Keywords— Context; Context-aware system; Service oriented architecture; Mobile health; IPOJO.

ملخص:

يعرف إنشاء تطبيقات وخدمات الهاتف المحمول تطوراً واسعاً في الآونة الأخيرة، من خلال الاعتماد على مفهوم السياق وحالة المستخدم. معظم التطبيقات المتاحة تواجه ثغرات تعيق عملها. ويعود ذلك لأنشائها خارج السياق، لهذا تصبح غير مناسبة لشروط التنفيذ. هذه الإشكالية هي التي دفعتنا إلى إنشاء نموذج موحد يركز على إدارة الخدمات في وقت واحد بهدف تشكيل تطبيق قوي.

في بداية عملنا نقوم بدراسة متعمقة للتطبيقات المحمولة وسلوكها، اعتماداً على سياق التنفيذ، مع الأخذ في الاعتبار مختلف التطبيقات المخصصة للصحة، التي تهدف لدعم وللمساعدة المريض في حياته اليومية (المراقبة الطبية، الأمن الدائم، تطبيقات التحليلات والعلاج عن بُعد، إلخ).

أما في خطوة ثانية، نقترح نموذجاً ديناميكياً وحساساً للسياق، يسمح بإدارة السياق والتكيف من أجل موازنة إمكانات الخدمة التي تم تطويرها مع بيئتها وللاستخدام الأفضل للموارد في بيئات شديدة التغير. نحن نركز على الجانب التفاعلي، مما يسمح لنا بتكييف سلوك التطبيقات مع ظروف استخدامها.

في إطار هذه الأطروحة، يتم تنفيذ النموذج الأولي لمنصة CAHS (خدمات الصحية الحساسة للسياق) المقترحة ضمن نظام Java IPOJO واستناداً إلى التكيف مع خدمة SOA. هذه المنصة تعمل على توفير حساسية التطبيقات للسياق وضمان أداء أفضل لجميع التطبيقات المخصصة للصحة. يهدف النهج الذي نتبعه إلى تصميم جيل جديد من المنصات المحمولة المفتوحة والجامعة لمختلف أنواع التطبيقات المحددة لصحة المحمول.

Mot-clé— Contexte ; Système sensible au contexte ; Architecture orientée service ; Santé mobile ; IPOJO.