UNIVERSITÉ LARBI TÉBESSI, TÉBESSA

Faculté des Sciences Exactes et Sciences de la Nature et de la Vie

Département des Mathématiques et Informatique

Laboratoire des Mathématiques, Informatique et Systèmes (LAMIS)

Thèse

Spécialité : Informatique

Présentée et soutenue publiquement par

ACHOURI MOUNIR

En vue de l'obtention du diplôme de

Docteur (cycle LMD)

Intitulée :

Profil sémantique unifié et Cloud Computing pour les environnements intelligents

Devant le jury

Dr. Makhlouf Derdour	Université Larbi Tébessi	Directeur de thèse
Dr. Adel ALTI (MCA)	Université de Sétif	Co-Directeur de thèse
Professeur Med Ridda LAOUAR	Université Larbi Tébessi	Président
Dr. Mohamed AMROUNE (MCA)	Université Larbi Tébessi	Examinateur
Dr. Hichem TALBI (MCA)	Université de Constantine 2	Examinateur
Dr. Mohamed Amine FERRAG (MCA)	Université de Guelma	Examinateur

Année Universitaire 2019 / 2020

Profil sémantique unifié et Cloud Computing pour les environnements intelligents

Achouri Mounir

UNIVERSITÉ DE LARBI TÉBESSI

Laboratoire LAMIS

Résumé

L'intelligence ambiante (AmI) est un domaine scientifique qui concerne les environnements constitués d'appareils intelligents (capteurs et actionneurs) capables de connaître et de réagir à la présence de personnes. AmI doit fonctionner de manière simple et naturelle, en utilisant les données fournies par les capteurs, de manière à soutenir les activités et les tâches de la vie quotidienne des personnes. En raison des progrès de la technologie des capteurs, les capteurs sont devenus plus puissants, moins chers et de plus petite taille, stimulant ainsi les déploiements à grande échelle. De ce fait, nous avons aujourd'hui un grand nombre de capteurs déjà déployés et leur nombre devrait augmenter rapidement au cours de la prochaine décennie. Ces capteurs génèrent en permanence d'énormes quantités de données. Cependant, pour ajouter de la valeur aux données brutes des capteurs, nous devons les comprendre. La collecte, la modélisation, le raisonnement et la distribution du contexte en relation avec les données de capteurs jouent un rôle crucial. Les ontologies sont considérées comme un pilier essentiel de la gestion avancée de la situation dans divers domaines intelligents. Il joue un rôle essentiel dans la compréhension du contexte de l'utilisateur afin de déterminer la sécurité des personnes, l'identification de la situation et personnalisé le confort. Nous avons proposé une nouvelle ontologie générique de profils sensibles à la situation des utilisateurs (GUSP-Onto) pour une description sémantique de profils d'utilisateurs hétérogènes avec une gestion efficace de la situation des utilisateurs. Sur la base de l'ontologie de gestion de la situation des utilisateurs, une architecture à deux couches a été proposée. La première couche est utilisée pour réaliser un diagnostic de qualité des situations urgentes, y compris un Fog Computing intelligent optimisé avec une modélisation de profil sémantique offrant une gestion efficace des situations. La deuxième couche permet une analyse de situation plus détaillée pour les patients et des services enrichis et enrichis utilisant le Cloud Computing, offrant une bonne évolutivité. La plus innovante de cette architecture est les avantages potentiels de la représentation sémantique pour mener un raisonnement de connaissance des situations d'urgence et finalement réaliser un processus de sélection et d'adaptation de service précoce. Les résultats expérimentaux montrent une réponse temporelle réduite et une précision accrue de l'approche proposée.

Mots Clés : *profil utilisateur, sensible au contexte, Cloud Computing, Fog Computing, Intelligence ambiante.*

Abstract

Ambient intelligence (AmI) is a scientific field that concerns environments consisting of intelligent devices (sensors and actuators) able to know and react to the presence of people. AmI must operate in a simple and natural way, using the data provided by the sensors, in order to support the activities and tasks of the daily life of the people. Due to advances in sensor technology, sensors have become more powerful, cheaper, and smaller, boosting large-scale deployments. As a result, we now have a large number of sensors already deployed and their number is expected to increase rapidly over the next decade. These sensors constantly generate huge amounts of data. However, to add value to the raw sensor data, we need to understand them. Collection, modeling, reasoning and context distribution in relation to sensor data play a crucial role. Ontologies are seen as a key pillar of advanced situational management in a variety of smart areas. It plays a vital role in understanding the context of the user in order to determine the safety of people, the identification of the situation and personalized comfort. We have proposed a new generic ontology of profiles sensitive to the situation of the users (GUSP-Onto) for a semantic description of heterogeneous user profiles with an effective management of the situation of the users. Based on the user situation management ontology, a two-layer architecture has been proposed. The first layer is used to perform quality diagnostics of urgent situations, including intelligent Fog Computing optimized with semantic profile modeling for effective situation management. The second layer provides a more detailed situation analysis for patients and enriched and enriched services using Cloud Computing, offering good scalability. The most innovative of this architecture is the potential benefits of semantic representation for leading reasoning of knowledge of emergencies and ultimately achieving an early service selection and adaptation process. The experimental results show a reduced temporal response and an increased precision of the proposed approach.

Key Words: *user profile, context-aware, Cloud Computing, Fog Computing, Ambient Intelligence.*

ملخص

الذكاء المحيط (AmI) هو مجال علمي يتعلق بالبيئات التي تتكون من أجهزة ذكية (أجهزة استشعار ومشغلات) قادرة على رصد وجود الأشخاص والتفاعل معهم. يجب أن تعمل أنظمة الذكاء المحيط بطريقة بسيطة وطبيعية، وذلك باستخدام البيانات المقدمة من أجهزة الاستشعار، وذلك لدعم أنشطة ومهام الأشخاص في حياتهم اليومية. نظرًا للتقدم الحاصل في تقنية المستشعرات، حيث أصبحت أكثر قوة وأصغر حجما وبثمن بسيط، مما يعزز عمليات نشرها في أماكن واسعة النطاق. نتيجة لذلك، لدينا الآن عدد كبير من أجهزة الاستشعار التي تم نشرها ومن المتوقع أن يزداد عددها بسرعة خلال العقد المقبل. تولد هذه المستشعرات باستمرار كميات هائلة من البيانات. ولإضافة قيمة إلى هذه البيانات القادمة من المستشعرات، نحتاج إلى فهمها. يلعب الجمع والنمذجة والاستنتاج وتوزيع السياق فيما يتعلق ببيانات المستشعر دوراً هاما. الأنطولوجيا هو الركن الأساسي للإدارة وتحديد الحالة في مجموعة متنوعة من المجالات الذكية. تلعب الأنطولوجيا دورًا حيويًا في فهم سياق المستخدم من أجل تحديد الحالة والموقف لضمان الراحة والسلامة للأشخاص. لقد اقترحنا أنطولوجيا عامة جديدة من الملفات الشخصية الحساسة لحالة المستخدمين (GUSP-Onto) للحصول على وصف دلالي لملفات تعريف المستخدمين الغير المتجانسة مع إدارة فعالة لحالة المستخدمين. على أساس الأنطولوجيا المقترحة لإدارة الوضع للمستخدم، اقترحنا هيكل من طبقتين. تُستخدم الطبقة الأولى لتحديد الحالات العاجلة، بما في ذلك الحوسبة الضبابية الذكية مع نمذجة الملف التعريفي الدلالي من أجل الإدارة الفعالة للتعرف على الحالة. توفر الطبقة الثانية تحليلًا أكثر تفصيلًا للحالة بالنسبة للأشخاص وتوفير خدمات متنوعة باستخدام الحوسبة السحابية، مما يوفر قابلية جيدة للتوسع. الأكثر إبداعًا لهذا الهيكل هو الفوائد المحتملة للتمثيل الدلالي لقيادة التفكير في معرفة حالات الطوارئ والتي تحقق في النهاية اختيار الخدمة المناسبة وتكييفها حسب متطلبات المستعمل. تظهر النتائج التجريبية انخفاض مدة الاستجابة الزمنية وزيادة دقة تحديد الحالة.

Dédicace

À mes parents ;
À toute ma famille ;
À tous ceux qui me sont chers.

Remerciement

Je fais partie des personnes qui croient mordicus qu'il n'y a de force ni de puissance que par Dieu. Mes remerciements vont tout premièrement au dieu tout puissant pour la volonté, la santé et la patience qui m'a donné durant tous ces années d'études.

Cette thèse a été réalisée avec le soutien et l'assistance d'un grand nombre de personnes. Je profite cette occasion pour exprimer ma gratitude à ces personnes pour leurs contributions comme mes collègues, ma famille et mes amis.

Je suis très reconnaissant aux membres du jury qui ont accepté de réviser et d'évaluer ma thèse et de fournir des critiques pertinentes. Je vous assure de ma profonde gratitude pour l'attention que vous avez portée à ce manuscrit et le temps que vous avez consacré à son évaluation.

Je voudrai exprimer tout d'abord ma gratitude à ma belle-famille et surtout, à ma mère, mon père, mes frères, mes sœurs. Ils sont toujours intéressées à mes études avec le succès et m'ont aidé beaucoup plus qu'elles ne peuvent le croire.

Je voudrai exprimer mes sincères remerciements à mon encadreur, le Dr. Derdour Makhlouf, pour avoir acceptée de diriger ma thèse. Il m'a fait connaître un nouveau domaine de recherche réseau, de l'informatique de nuage véhiculaire. Au cours de ma thèse, Dr. Derdour Makhlouf m'a orienté en proposant de nombreuses idées et perspectives. Grâce à son pensé logique dans ce domaine, mes idées et mes propositions initiales ont été améliorés et développés.

Je tiens également ma gratitude à mon co-encadreur le Dr. Philippe Roose pour avoir acceptée de diriger cette thèse et pour ses conseils et son suivi durant mon stage en France, n'oublie pas non plus toute sa famille.

Je voudrais m'adresser aussi sincèrement que possible aux membres du jury : Prof. Mohammed Rida Laouar, Dr. Mohamed AMROUNE, Dr. Hichem TALBI et Dr. Mohamed Amine FERRAG pour avoir accepté de participer au jury de cette thèse. Je voudrais également exprimer mes sincères remerciements à tous les membres de notre laboratoire de recherche (LAMIS), qui cherchent à développer le travail de recherche de ce laboratoire dans des domaines particuliers de l'informatique, et à valoriser le travail de recherche en organisant et en présentant des journées doctorales et des conférences internationales Je voudrais également remercier tous mes collègues pour leur patience et leurs encouragements durant mon doctorat. Donc, heureux d'avoir des amis aimants comme vous. Vraiment merci.

Merci à vous tous !

Table des matières

Liste des tables

Liste des figures

Introduction générale

1. Contexte

En raison des progrès et la miniaturisation de la technologie, diverses technologies intelligentes mobiles intégrées dans la vie quotidienne ont été rapidement développés, les capteurs deviennent de plus en plus puissants, économiques et de plus petites tailles, ce qui a encouragé leur déploiement à grande échelle. Cela inclut les Smartphones, les tablettes ou les capteurs sans fil utilisés pour réaliser un environnement intelligent. Ils constituent des sources clés pour un environnement intelligent. Ainsi, les utilisateurs sont de plus en plus dépendants de l'utilisation de ces technologies en tant que partie intégrante de l'environnement intelligent, une solution émergente axée sur la technologie pour répondre aux actions des personnes et des objets et satisfaire leurs besoins. En conséquence, un grand nombre de capteurs ont déjà été déployés et il est prévu que leur nombre augmente rapidement au cours de la prochaine décennie, lorsqu'un grand nombre de capteurs est déployé et commence à générer des données que nous recueillerons, ceci ne peut avoir aucune valeur si nous ne pouvons analyser, interpréter et comprendre. L'approche traditionnelle basée sur les applications (c'est-à-dire connecter les capteurs directement aux applications individuellement et manuellement) devient irréalisable.

En raison de la diversité de ces capteurs de bas niveau, la gamme ambiante et intelligente de données observées à partir des systèmes peut varier considérablement. En conséquence, la conversion des données en formats structurés interchangeables et leur agrégation et gestion ultérieures devient un défi important. Les implémentations réussies reposent non seulement sur des bases matérielles et logicielles solides, mais également sur des installations de calcul distantes [5]. Ainsi, l'adoption de modèles de Cloud Computing jouera un rôle vital, en particulier avec l'augmentation de la performance globale du réseau.

2. Problématique

Les systèmes sensibles au contexte sont la clé de tout environnement intelligent, car ils doivent comprendre le contexte de la situation des données collectées et fournir des services personnalisés adaptés aux besoins de l'environnement et des utilisateurs. Un contexte approprié permet de mieux comprendre l'état des personnes, des lieux ou des objets, et leur relation avec l'interaction entre les utilisateurs et le système. Les données physiologiques d'une personne varient selon les activités (Le rythme cardiaque dans un état normal par rapport aux exercices) et même à différents endroits (la chaleur corporelle lorsqu'il court dehors ou sur un tapis roulant à la maison). Les données varient aussi selon l'âge, ou varient en fonction de l'individu.

Les données collectées d'après les capteurs d'une manière dynamique sur des périodes différents, génèrent une énorme quantité d'informations de contexte pour les systèmes ambiants intelligents. Les systèmes ambiants ont besoin d'un référentiel complet de connaissances et doivent rester sensibles au contexte afin de satisfaire différents profils de comportement en fonction des besoins spécifiques de chaque individu. Cependant, la mise en œuvre de telles opérations à partir des serveurs centraux entraînera une rigidité du système et sera vulnérable de tomber en panne et ne sera pas évolutive.

3. Objectif de la thèse

Une plate-forme basée sur le Cloud facilite la gestion de ces grands systèmes sensibles au contexte, en simplifiant l'accès des utilisateurs et en gérant efficacement l'élasticité de la demande [8].La nécessité d'un grand espace de calcul et de la disponibilité immédiate des services de Cloud Computing nous ont amenés à envisager et à concevoir un Framework sensible au contexte en temps réel basé sur le Cloud. En effet, la variété des situations et de contextes dans lesquels interagissent les utilisateurs sont variés et pour la plupart manipulent des contenus orientés multimédia provenant de sources variées (*locales, Cloud, etc.*). Cet accès est réalisé en utilisant des terminaux aux caractéristiques physiques et d'interactions différentes (*Smartphones, téléviseurs standards ou Smart TV, ordinateurs, tablettes, Set-top boxes, etc.*). En plus, l'usage de services différents à partir de périphériques selon des scénarios mouvants qui affectent à la fois la présentation des informations mais également les services accédés ainsi que les différentes interactions dont l'usager peut et doit disposer. Le concept des environnements intelligents vise à offrir une vie autonome, à plusieurs domaines médicaux et sociaux comme le domaine du Smart-Health, Smart-Home, Smart-Car, etc. Notre principal objectif est d'offrir un support à la réalisation d'une plateforme autonomique longue durée fondée sur l'utilisation d'ontologies c'est-à-dire prendre des décisions plus intelligentes et automatisées qui s'adapteraient aux besoins du moment comme la localisation de l'utilisateur et ses ressources disponibles.

4. Motivation

Le paradigme de l'intelligence Ambiant (AmI) représente la vision future de l'informatique intelligente dans laquelle les environnements prennent en charge les personnes qui s'y trouvent. Dans ce nouveau paradigme informatique, les supports d'entrée et de sortie classiques n'existent plus, mais les capteurs et les processeurs seront intégrés aux objets du quotidien, travaillant

ensemble en harmonie afin de satisfaire les besoins des utilisateurs de manière transparente et anticipée.

Les solutions architecturales existantes des systèmes sensibles au contexte se limitent à des services spécifiques et reposent principalement sur un agent intelligent local (c'est-à-dire un périphérique mobile) pour la découverte et la gestion du contexte. Le manque de stockage et de puissance dans les capteurs portables et les appareils mobiles les empêche de traiter des données de capteurs sans limites en utilisant des méthodes de calcul convenables. De plus, la découverte de nouveaux capteurs et appareils intelligents accroît la demande de services d'assistance plus intelligents et plus complexes. Le Cloud Computing augmente la capacité de traitement de données volumineuses. Cela nous a convaincu de construire un bon système coopératif en transférant la tâche de gestion du contexte d'un périphérique intelligent local vers un environnement en Cloud distribué afin d'améliorer le temps de traitement de la création du contexte et du transfert de services complexes.

Les systèmes traditionnels de gestion de contexte sont incapables de gérer ensemble un grand nombre de systèmes sensible au contexte, ces applications indépendantes reposant sur un serveur local [18,19].Cet inconvénient nous incite à concevoir une architecture orientée Cloud qui sera suffisamment capable de gérer simultanément un bon nombre de clients. Le contexte dérivé d'un système sensible au contexte deviendra une banque de connaissances de contexte pour un autre système dans le Cloud.

L'intégration du Cloud Computing étendra la diversité des services. Nous souhaitons développer un système capable de fournir chaque type de service en utilisant un seul modèle. En d’autres termes, de la création du contexte à la prestation des services, tout est possible avec la plate-forme Cloud.

5. Structure de la thèse

Après cette introduction générale, cette thèse est organisée comme suite :

• Le premier chapitre a pour objectif de décrire l’évolution de l’informatique et l’impact de la technologie de communication aboutissant à l’intelligence ambiante, ensuite nous présenterons quelques notions de base concernant les systèmes ambiants, caractéristiques des systèmes ambiants,

• Le deuxième chapitre présente le contexte scientifique de notre travail de thèse. Nous y détaillons la notion du contexte, la notion de la sensibilité au contexte et nous y présente une architecture générale d'un système sensible au contexte.

• Le troisième chapitre Architectures et profile sémantique pour les environnements intelligents. Ce chapitre est destiné à présenter les différents travaux connexes pertinents aux systèmes sensibles au contexte. Nous terminons ce chapitre par une étude comparative de ces travaux.

• Le quatrième chapitre, contribution d'une nouvelle ontologie générique de profils sensibles à la situation des utilisateurs (GUSP-Onto) pour une description sémantique de profils d'utilisateurs hétérogènes avec une gestion efficace de la situation. Nous y présentons une conception et une implémentation de nos propositions

•Le cinquième sur la base de l'ontologie de gestion de la situation des utilisateurs (GUSP-Onto), une architecture à deux couches a été proposée. La première couche est utilisée pour réaliser un diagnostic de qualité des situations urgentes, y compris Fog Computing amélioré avec une modélisation de profil sémantique offrant une gestion efficace des situations. La deuxième couche permet une analyse de la situation plus en profondeur pour les clients et des services améliorés riches en utilisant le Cloud Computing qui offre une bonne évolutivité. Le document se termine par une conclusion générale qui présente une synthèse de nos contributions ainsi que les perspectives que nous avons tracées.

Chapitre 1 : Système ambiant et Cloud Computing

1. Introduction

La prochaine vague de l'ère de l'informatique sortira du domaine des ordinateurs de bureau traditionnels. Dans le paradigme de L'intelligence ambiante, un grand nombre d'objets communicants qui nous entourent représentant un capteur, un actionneur ou un objet physique quelconque (TV, Smartphone, tablette tactile, etc.) seront accessibles à grande échelle selon le paradigme de l'Internet des objets. L'identification par fréquence radio (RFID) et les technologies de réseau de capteurs vont faire leur apparition pour relever ce nouveau défi, dans lequel les systèmes d'information et de communication sont intégrés de manière invisible à l'environnement. Cela se traduit par la génération d'énormes quantités de données qui doivent être stockées, traitées et présentées sous forme transparente, efficace et facilement interprétable. Ce modèle comprendra des services qui sont des produits et fournira de manière similaire aux produits traditionnels. L'intelligence ambiante peut fournir aux utilisateurs des services personnalisés en tenant compte du contexte, lorsqu'ils interagissent et échangent des informations avec l'environnement. De nos jours, L'intelligence ambiante évolue de son origine en tant que sujet de recherche universitaire à une réalité commerciale [1]. Il y a de nombreuses applications potentielles, des lieux de travail intelligents et des maisons de soins de santé intelligentes, aux salles de jeux, aux systèmes de loisirs et aux transports en commun [2]. Les environnements ambiants sont de plus en plus hétérogènes, décentralisés et autonomes. D'une part, l'informatique mobile est déjà au cœur de nombreuses applications basées sur la localisation (GPS) des utilisateurs et des dispositifs et l'accès aux services par des terminaux utilisateurs spécifiques : Laptop, PDA, Smartphone, tablettes, Smart-Watch et autres terminaux mobiles sont maintenant courants et composent une grande partie des systèmes informatiques. D'autre part, les ressources de l'environnement sont souvent réparties sur de larges zones à travers des réseaux qui peuvent aller d'un réseau mondial (vaste) come Internet jusqu'aux réseaux locaux pair-à-pair. L'un des problèmes majeurs de ce type d'application est l'adaptation autonomique sémantique des services aux contextes d'usages. Dans ce genre de système, l'adaptation automatique des services à un ensemble de contraintes et de paramètres (QoS, largeur de bande passante, localisation de l'utilisateur, temps, type de dispositifs disponibles : vitesse CPU, taille RAM, etc.) doit être assurée pour garantir une utilisation confortable et transparente.

2. Informatique Ubiquitaire

De nos jours, la miniaturisation des dispositifs électroniques et les moyens de communication sans fil à faible consommation électrique, l'informatique est devenue embarquée dans tout type d'objet de la vie quotidienne, grâce à cette évolution, sont nés de nouveaux systèmes d'information dits pervasifs ou ubiquitaire [3]. La notion d'informatique ubiquitaire «Ubiquitous Computing» a été développé par Mark Weiser [4] au cours des années 80 à Xerox PARC, pour donner sa vision de l'ordinateur du 21ième siècle comme un terminal plus intelligent intégré dans les objets de la vie quotidienne et il deviendra invisible dans notre environnement de telle sorte qu'il rendrait continuellement des services indispensables sans l'intervention de l'être humain et pas un assistant de bureau [5]. Ainsi, l'informatique ubiquitaire (i.e. omniprésente à rendre la technologie intangible dans l'environnement quotidien et non pas simplement à la cacher, C'est la notion de disparition de la technologie dans l'environnement de la vie quotidienne, comme le souligne Weiser : «*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it* » [4]. La vision révolutionnaire de Weiser a évoqué de nombreux travaux dans des domaines divers tels que les réseaux de capteur, les systèmes distribués, les architectures matérielles et logicielles, la mobilité, la domotique, etc. de nombreuses études et expériences restent encore à faire pour poursuivre l'exploration de l'informatique ubiquitaire.

Selon Saha et Mukherjee [6] Grâce aux capteurs, l'informatique ubiquitaire peut effectuer certaines tâches automatiquement, qui fournissent des informations sur le contexte des environnements intelligents, Ces données facilitent une interaction plus naturelle avec les utilisateurs. D'après Friedewald [7] l'informatique ubiquitaire est un paradigme qui réfère au concept "every-where" où l'informatique est intégrée dans l'environnement par l'installation de plus en plus de dispositifs au lieu d'avoir les ordinateurs comme des objets distincts (voitures, meubles, PDA, smart phones) dans notre vie quotidienne, l'accès est transparent et invisible aux ressources vis-à-vis des utilisateurs.

3. Internet des objets

Vers la fin des années 1960, la communication entre deux ordinateurs était rendue possible par un réseau informatique [8]. Au début des années 1980, la pile TCP / IP a été introduite. Ensuite, l'utilisation commerciale d'Internet a commencé à la fin des années 1980. Plus tard, le World Wide

Web (WWW) est devenu disponible en 1991, ce qui a rendu Internet plus populaire et a ainsi stimulé la croissance rapide. Web of Things (WoT) [9], basé sur WWW, fait partie de l'IdO. Plus tard, des appareils mobiles connectés à Internet et formant l'Internet mobile [10].

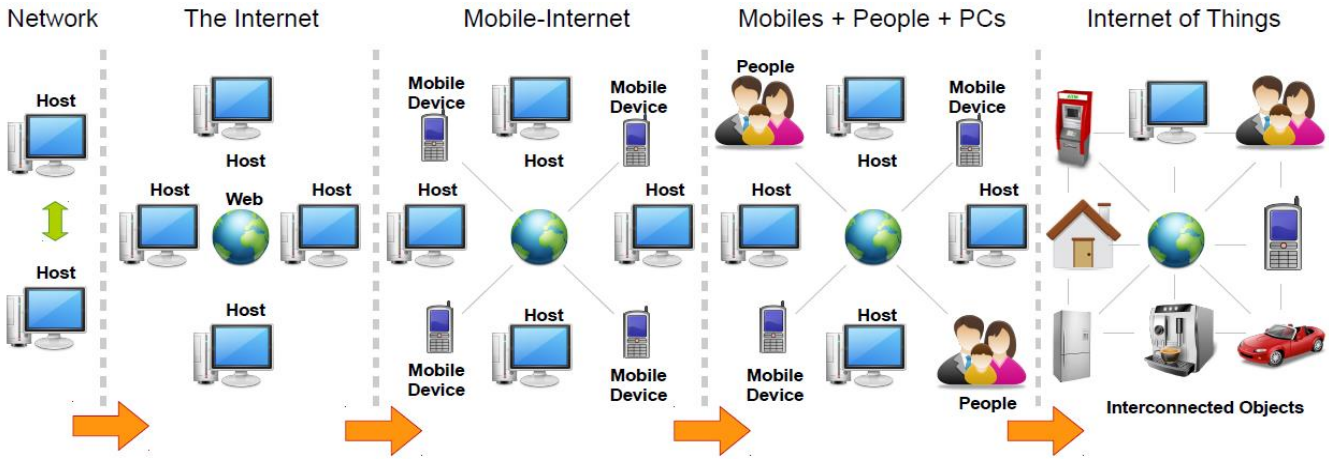

Figure 1.1 L'évolution de l'internet [10]

Avec l'émergence des réseaux sociaux, les utilisateurs ont commencé à se connecter via Internet. La prochaine étape de l'IdO consiste à permettre aux objets de notre entourage de se connecter les uns aux autres (par exemple, machine à machine) et de communiquer via Internet [11]. L'évolution d'Internet commence par la connexion de deux ordinateurs, puis vers la création d'un World Wide Web par la connexion d'un grand nombre d'ordinateurs. L'Internet mobile est apparu en connectant des appareils mobiles à Internet. Ensuite, les identités des peuples ont rejoint Internet via les réseaux sociaux. Enfin, on se dirige vers l'Internet des objets en connectant chaque jour des objets à Internet. La figure 1.1 illustre les cinq phases de l'évolution d'Internet.

Au cours de la dernière décennie, l'IdO a attiré une attention considérable dans les milieux universitaires et industriels. Les principales raisons de cet intérêt sont les capacités offertes par l'IdO [12], [13]. Il promet de créer un monde où tous les objets (également appelés objets intelligents [24]) qui nous entourent sont connectés à Internet et communiquent les uns avec les autres avec un minimum d'intervention humaine [15]. Le but ultime est de créer "un monde meilleur pour les êtres humains", dans lequel les objets qui nous entourent savent ce que nous aimons, ce que nous voulons et ce dont nous avons besoin et agissent en conséquence sans instructions explicites [16].

Le terme «Internet des objets» a été inventé pour la première fois par Kevin Ashton [17] lors d'une présentation en 1998. Il a mentionné que «l'Internet des objets a le potentiel de changer le

monde, tout comme Internet. Peut-être même plus encore ». Ensuite, le centre d'identification automatique du MIT a présenté sa vision de l'IdO en 2001 [18]. Plus tard, l'IoT a été officiellement introduit par l'Union internationale des télécommunications (UIT) par le rapport Internet de l'UIT en 2005 [19].L'IdO englobe un nombre important de technologies qui orientent sa vision. Dans le document intitulé Vision et défis pour la réalisation de l'Internet des objets, par CERP-IoT [4], un ensemble complet de technologies a été répertorié. L'IdO est une vision très large. La recherche sur l'IdO en est encore à ses balbutiements. Par conséquent, il n'existe pas de définition standard pour l'IoT. Les définitions suivantes ont été fournies par différents chercheurs. Selon [20]: "Les identités et les personnalités virtuelles des objets fonctionnent dans des espaces intelligents à l'aide d'interfaces intelligentes permettant de se connecter et de communiquer au sein de contextes sociaux, environnementaux et utilisateurs." Définition de [10] : "L'origine sémantique de l'expression est composée de deux mots et concepts : "Internet" et "Thing", où Internet peut être défini comme le réseau mondial de réseaux informatiques interconnectés, reposant sur un protocole de communication standard. alors que "Thing" est un objet non précisément identifiable. Par conséquent, Internet des objets est sémantiquement un réseau mondial d'objets interconnectés, adressables de manière unique, basés sur des protocoles de communication standard". Cette évolution a donné naissance à un nouveau paradigme appelé L'intelligence ambiante, offrant un environnement personnalisé, qui répond intelligemment à des besoins utilisateurs et améliore la vie des personnes, Cette nouvelle orientation de l'informatique sera développée dans la suite de ce chapitre.

4. L'intelligence ambiante

Concevoir des environnements intelligents est un objectif qui séduit les chercheurs de diverses disciplines, notamment l'informatique ubiquitaire, l'internet des objets, les réseaux de capteurs, l'intelligence artificielle, la robotique, l'informatique multimédia, les middlewares et les logiciels basés sur des agents [21]. Les progrès réalisés dans ces domaines connexes ont entraîné une augmentation considérable du nombre de projets d'environnement intelligent. L'évolution de la technologie est un facteur important dans la naissance du domaine de l'AmI. Les ordinateurs étaient à l'origine très chers, mais difficiles à comprendre et à utiliser. Chaque ordinateur était une ressource rare et précieuse. Un seul ordinateur serait généralement utilisé par de nombreuses personnes (voir la figure 1.2). Dans la prochaine étape de l'évolution, de nombreux utilisateurs n'avaient plus besoin d'utiliser un ordinateur à tour de rôle, car beaucoup pouvaient y accéder

simultanément. La révolution informatique dans les années 80 a changé le ratio à un utilisateur par ordinateur. Au fur et à mesure que l'industrie progressait et que les coûts diminuaient, un utilisateur était souvent capable d'accéder à plus d'un ordinateur. Le type de ressources informatiques dont nous disposons aujourd'hui est considérablement plus varié qu'il y a quelques décennies. La société Philips a présenté le terme Ambiant Intelligence en 1998 dans le cadre du projet «Vision of the Future» [22]. Dans ce projet, Philips a mené une réflexion et une analyse prospective en interne sur l'évolution de l'électronique grand public. La Commission européenne a pour la première fois tracée la voie à suivre par la recherche AmI en 2001 [23].

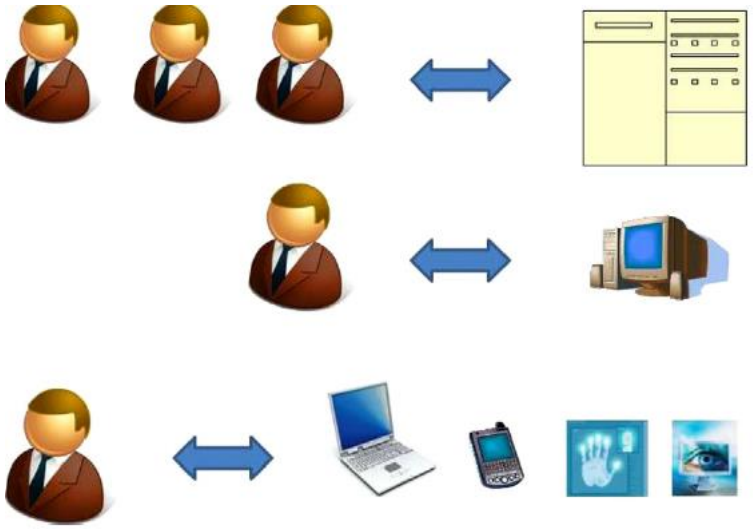

Figure 1.2 Un changement dans le ratio ordinateur / personnes [26]

L'intelligence ambiante est un concept développé au cours des dernières décennies du siècle dernier, C'est un paradigme large et multidisciplinaire qui établit un nouveau type de relation entre l'homme, son environnement et la technologie. Pour cette raison, un certain nombre de définitions peuvent être trouvées dans la littérature, chacune se focalisant sur un aspect différent. Une définition largement acceptée du concept d'AmI provient de l'ISTAG [22], groupe chargé de conseiller la Commission européenne sur la stratégie globale à suivre pour réaliser la priorité thématique Information et Communication dans le cadre de la recherche européenne. "*Le concept d'Ambiant Intelligence (AmI) fournit une vision de la société de l'information qui met l'accent sur une plus grande convivialité, un soutien plus efficace des services, l'autonomisation des utilisateurs et le soutien des interactions humaines. Les gens sont entourés d'interfaces intuitives intelligentes intégrées à toutes sortes d'objets et d'un environnement capable de*

reconnaître et de réagir à la présence de personnes différentes de manière transparente, discrète et souvent invisible."

Les environnements intelligents ambiants montreront leur "intelligence" d'une part par la nature sociale de l'interface, une sorte de dialogue avec l'utilisateur, et d'autre part par la capacité du système à s'adapter à ses utilisateurs et à ses environnements. Le caractère social de l'interface utilisateur sera déterminé par la mesure dans laquelle le système répondra au contexte social et culturel de l'utilisateur et sa capacité d'adaptation dépendra de la capacité du système à comprendre le contexte et à réagir à ses changements.

D'après Aarts [25], la notion d'ambiance dans AmI fait référence à l'environnement et reflète la nécessité d'intégrer la technologie de manière à s'intégrer discrètement dans les objets de la vie quotidienne, l'intelligence étant liée à la capacité de l'environnement numérique à présenter une forme spécifique. D'interaction sociale avec les personnes qui vivent dans l'ambiant.

Les auteurs identifient les principales caractéristiques d'AmI comme suit :

- ✓ intégration par l'intégration à grande échelle de l'électronique dans l'environnement ;
- ✓ connaissance du contexte grâce à l'identification de l'utilisateur, de l'emplacement et de la situation ;
- ✓ personnalisation par ajustement d'interface et de service ;
- ✓ adaptation par apprentissage
- ✓ anticipation par le raisonnement.

Cette définition met également l'accent sur la caractéristique d'AmI d'améliorer la qualité de vie des personnes. Le facteur technologique, même s'il est important et qu'il est un catalyseur, n'est pas suffisant pour atteindre la pleine expression d'AmI. En effet, une enquête sur les facteurs humains, les interactions naturelles et le comportement humain est nécessaire pour permettre une réelle autonomisation de l'utilisateur.

Cook et coll. [26] ont mené une enquête sur différentes définitions de l'ImA. Dans leurs recherches, ils ont mis en évidence les caractéristiques attendues dans un paradigme AmI (et dans les technologies impliquées) :

- ✓ sensible : capable de percevoir des informations sur le contexte ;

- ✓ réactif : capable de réagir à la présence de personnes dans l'environnement ;
- ✓ Adaptatif : capable de réagir différemment en s'adaptant à différentes situations ;
- ✓ Transparent : invisible pour l'utilisateur ;
- ✓ Omniprésent : présent partout ;
- ✓ Intelligent : capable de réagir et de s'adapter de manière intelligente (ici le concept d'intelligence est lié aux paradigmes de l'intelligence artificielle).

Ces recherches soulignent l'importance de la technologie utilisée pour percevoir les informations relatives à l'environnement et le rôle central de l'utilisateur dans la conception des services à offrir de manière "intelligente".

Souvent, l'informatique Ubiquitaire et Pervasive sont utilisées comme synonymes d'intelligence ambiante mais, comme l'a observé Augusto [27], les deux premiers concepts mettent l'accent sur la présence physique et la disponibilité des ressources, mais manque l'élément clé de "l'intelligence". Selon Augusto « Intelligence » devrait être conçu comme lié au domaine de l'intelligence ambiante. Les fonctions clés des systèmes AmI étant la flexibilité, l'adaptation, l'anticipation et une interface adéquate avec les humains, il est nécessaire de disposer d'une sorte d'intelligence capable de percevoir et d'apprendre de l'environnement et de raisonner de manière à répondre aux besoins des utilisateurs.

Le but de toutes ces définitions compatibles avec le contenu est de créer un environnement ou un système intelligent pour améliorer la qualité de vie des utilisateurs. Dans l'environnement intelligent, les appareils et équipements de la vie quotidienne sont transformés en objets

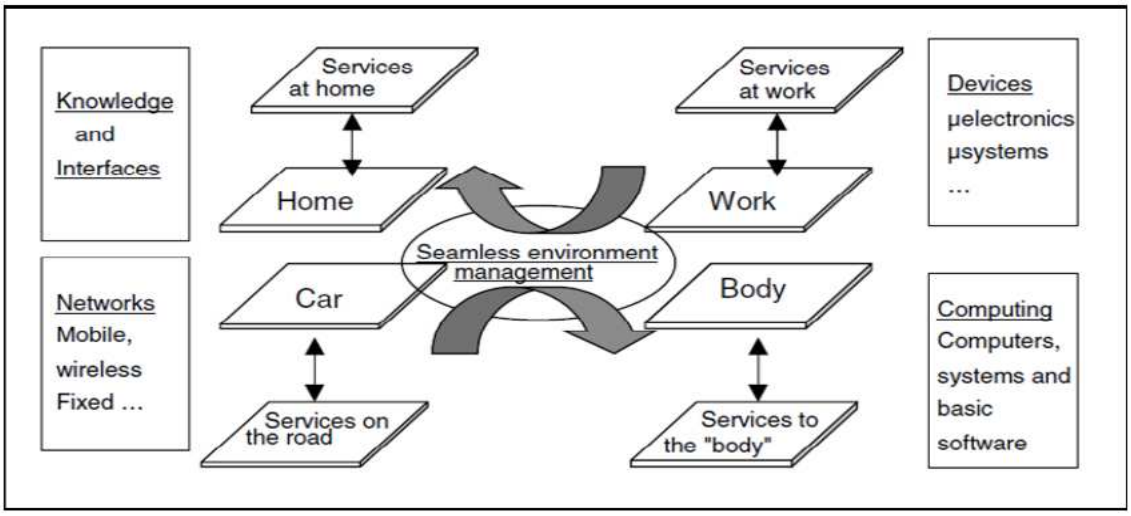

Figure 1.3 Espace AmI [28]

fournissant des services de support répondant aux besoins des utilisateurs de manière continue et proactive. Selon [28], les auteurs considèrent que l'interaction directe et naturelle entre les utilisateurs, les applications et les services dans l'environnement intelligent est évidente à la suite du chevauchement et de la convergence entre deux directions importantes, "l'informatique partout" et les "interfaces utilisateur". (Figure 1.3).

Les systèmes intelligents environnants reposent sur un système capable de gérer de manière permanente et transparente en contrôlant les espaces intelligents grâce à la possibilité de :

- ✓ Il doit être au courant des événements résultant des entités qui nous entourent, qu'il s'agisse d'une personne, d'un appareil, d'un lieu ou d'une application.
- ✓ Collecter et interpréter les informations à partir de sources hétérogènes, interpréter l'intention de l'humain .
- ✓ Réagir en conséquence de manière appropriée et transparente.

5. Le Cloud Computing

Le Cloud Computing est populaire dans la mesure où il fournit aux utilisateurs des services et des ressources informatiques rémunérés à la consommation [29]. Le Cloud Computing est une extension du cluster et du Grid Computing qui a été utilisé pour collecter des ressources en un lieu central et les utiliser pour un calcul haute performance. L'architecture du Cloud Computing fournit trois types de services tels que les logiciels en tant que service (SaaS), les plates-formes en tant que services (PaaS) et les infrastructures en tant que services (SaaS) [30]. NIST (*National Institute of Standards and Technology*) donne la définition du Cloud dans la conception de l'architecture informatique, qui fournit des ressources très puissantes pour l'informatique, le stockage, l'environnement de développement d'applications avec de multiples plates-formes avec une facilité de gestion et de coordination de tous les dispositifs de ressources en un seul endroit [31]. Le Cloud Computing fournit également des fonctionnalités de mobilité connues sous le nom de Cloud Computing mobile. Le Cloud Computing mobile est définie comme une «nouvelle vision du monde pour les applications portables, où le traitement et le stockage des informations sont transférés des clients à proximité vers des plates-formes informatiques intenses et centralisées situées dans le Cloud» [32]. Les modèles de Cloud Computing diffèrent en fonction du type de services qu'ils fournissent, tels que les services SaaS, tels que la comptabilité, les logiciels de bases

de données et les courriels, mais l'utilisateur n'a pas traité l'infrastructure technique de Cloud Computing [33]. PaaS fournit une plate-forme pour les développeurs d'applications telle qu'un environnement de programmation. Ces outils permettent de développer des applications et disposent des droits nécessaires pour configurer et gérer techniquement le Cloud [34]. IaaS fournit des services sur l'infrastructure de Cloud tels que la gestion de serveurs, le stockage et les périphériques réseau [35]. IaaS fournit le droit de gérer, de modifier ou de configurer une infrastructure Cloud en fonction de leurs besoins. Les modèles de Cloud Computing sont donnés à la figure 1.4.

5.1. Modèles de déploiement Cloud

Le Cloud Computing déployé selon quatre méthodes différentes : public, privé, communautaire et hybride [36].

- Cloud public : cette conception du Cloud est ouverte au public et est conçue et gérée par des organisations gouvernementales, des organisations du secteur de l'éducation ou des entreprises ou par une combinaison de certaines d'entre elles. Elle dépend des fournisseurs de services [37].

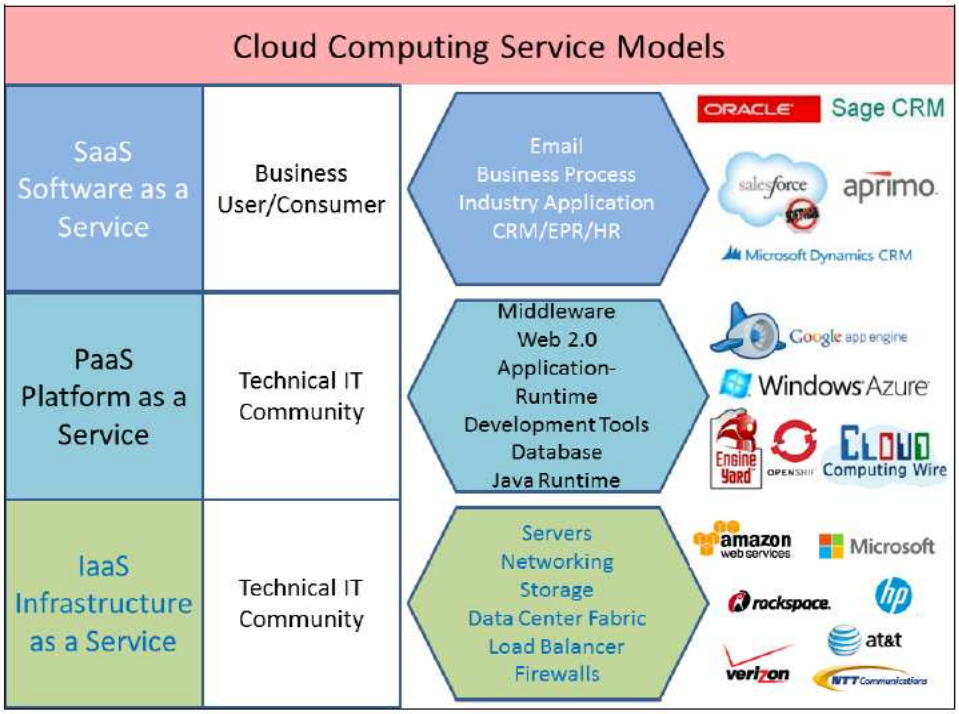

Figure 1.4 Modèle de service Cloud Computing [29]

- Cloud privé : le Cloud public est conçu et développé pour l'usage privé d'organisations telles que les agences de l'éducation, des entreprises et de la sécurité, ou géré par plusieurs organisations pour les consommateurs [38].
- Cloud communautaire : le Cloud conçu et géré par une communauté spécifique pour l'utiliser à des fins de sécurité ou d'affaires. Il est géré par un ou deux organismes basés sur la communauté [39].
- Cloud hybride : cette base dépend de la disposition de plusieurs conceptions de Cloud, telles que publiques, communautaires ou privées, qui s'unissent pour renforcer les informations sur les ressources informatiques et la transportabilité des applications [40].

5.2. Le Fog Computing

FOG est « Cloud plus proche de la masse ». Il s'agit d'une nouvelle architecture qui étend l'architecture traditionnelle du Cloud Computing aux limites du réseau. Avec le brouillard, le traitement de certains composants d'application (par exemple, ceux sensibles au temps de latence) peut avoir lieu à la périphérie du réseau, tandis que d'autres (par exemple, des composants à forte intensité de traitement et à tolérance de délai) peuvent se produire dans le Cloud. Le calcul, le stockage et les services sont les éléments constitutifs du CLOUD et du FOG qu'il élargit. Cependant, le FOG offre des avantages supplémentaires, tels qu'une faible latence, en permettant au traitement de se dérouler à la périphérie du réseau, à proximité des périphériques finaux, grâce aux nœuds appelés Fog et à la possibilité d'activer le traitement à des emplacements spécifiques [51] .Il propose également des points densément distribués pour la collecte des données générées par les périphériques finaux. Cela se fait par l'intermédiaire de serveurs proxy, de points d'accès et de routeurs situés à la périphérie du réseau, près des sources. Dans la littérature [52][13], il est largement reconnu que le Cloud Computing n'est pas viable pour la plupart des applications de l'Internet des objets (IoT) et que le Fog Computing pourrait être utilisé comme alternative. Le Fog Computing définie comme une variante améliorée où le calcul se produit au niveau du bord du système comme le montre dans la figure 1.5.

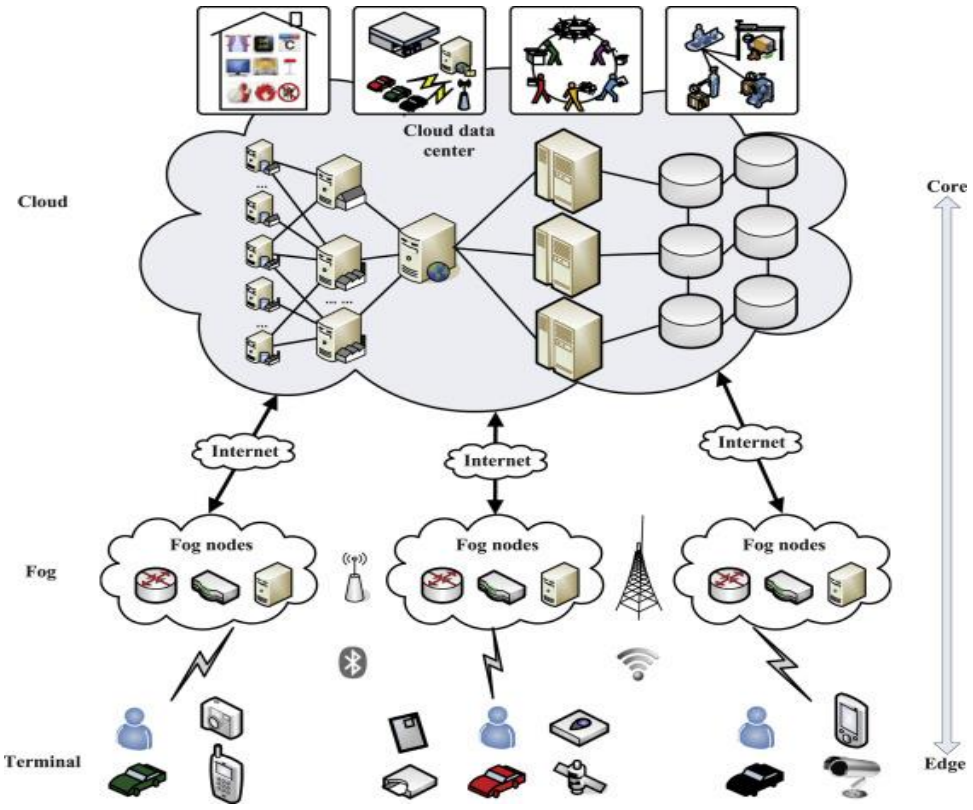

Figure 1.5 Architecture hiérarchique de Fog Computing [50]

6. Caractéristiques des systèmes ambiants

L'une des caractéristiques des systèmes ambiants est l'évolutivité de l'environnement. Des ressources qui sont disponibles à certains moments, et peuvent apparaître ou disparaître à tout moment. L'un des objectifs des systèmes ambiants est la prise en compte de cette évolutivité de l'environnement. L'évolution de l'environnement ne doit pas perturber l'utilisateur. Avec un grand nombre d'objets impliqués dans le traitement d'un problème, l'interaction des divers composants appelle de nouvelles stratégies. Cela est rendu plus difficile par le fait que les appareils doivent «s'associer» pour former des ensembles. Trois points principaux doivent être considérés :

6.1. Interopérabilité

Il s'agit de la capacité des unités hétérogènes indépendantes à coopérer et à échanger des données. Les dispositifs de plusieurs constructeurs doivent coopérer, ce qui signifie que des normes générales doivent être définies, décrivant non seulement les interfaces, mais également la sémantique des données échangées par les différents dispositifs.

6.2. Hétérogénéité

Ce terme désigne la possibilité d'exécuter un logiciel sur différents périphériques dont la puissance de calcul diffère considérablement. Dans un ensemble d'appareils, l'infrastructure logicielle utilisée doit être capable de faire évoluer les opérations informatiques nécessaires vers différentes unités et d'utiliser des concepts d'échange de données compatibles avec différents protocoles de communication.

6.3. Dynamique

Dans la plupart des systèmes ambiants, les infrastructures logicielles évoluent dynamiquement sous l'impulsion des apparitions et disparitions des objets et dispositifs. Ces évolutions peuvent être dues à la mobilité de ces objets et dispositifs, mais aussi à des mesures d'économie d'énergie ou encore à des mesures de l'environnement de l'application (*environnement physique, logiciel, de l'utilisateur, etc.*). Il est impératif de pouvoir considérer ces multiples mesures, non plus comme un ensemble des valeurs mais comme représentatives d'une situation ayant un sens du point de vue de l'objectif visé ; c'est à dire de la satisfaction de l'usage. Le système doit être en mesure de répondre avec souplesse aux changements dans l'infrastructure.

7. Technologies de l'intelligence ambiante

D'après sa définition, nous pouvons voir que AmI a une relation décisive avec de nombreux domaines de l'informatique. Nous organisons les technologies contributives dans cinq domaines, illustrés à la Figure 1.6. La présence de l'intelligence est un facteur clé de la recherche sur AmI. Nous adoptons la notion d'agent intelligent telle que définie par Russell et Norvig [54]. En tant que tel, l'algorithme AmI perçoit l'état de l'environnement et des utilisateurs dotés des capteurs, explique les données à l'aide de diverses techniques d'intelligence artificielle et agit sur l'environnement à l'aide d'un contrôleur, de manière à ce que l'algorithme atteigne le but recherché. Nous nous concentrons donc sur les technologies permettant la détection, le raisonnement et l'action.

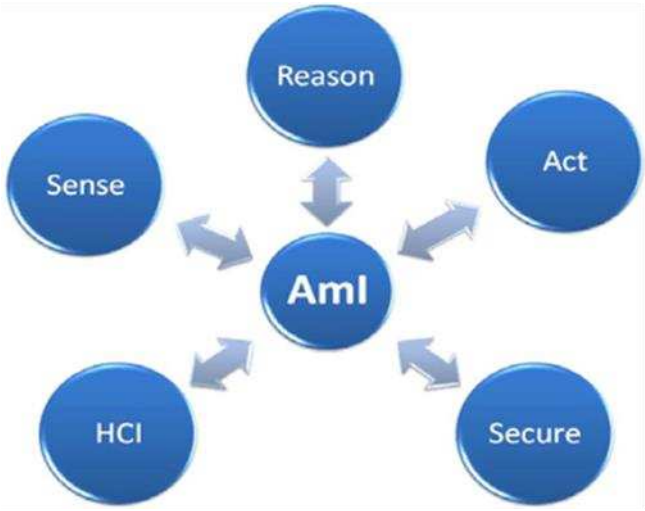

Figure 1.6 Relation entre AmI et d'autres domaines de la science informatique [26]

7.1.**La détection**

L'intelligence Ambiante est conçue pour des environnements physiques réels, il est essentiel d'utiliser efficacement les capteurs. Sans composants physiques permettant à un agent intelligent de détecter l'environnement et d'agir sur celui-ci, nous nous retrouvons avec des algorithmes théoriques sans application pratique. Les capteurs sont la clé qui relie la puissance de calcul disponible aux applications physiques. Les systèmes d'intelligence ambiante reposent sur des données capturées du monde réel. Le système perçoit l'environnement et utilise ces informations pour raisonner sur l'environnement et sur l'action pouvant être engagée pour modifier l'état de l'environnement. La perception est obtenue à l'aide d'une variété de capteurs. Les capteurs ont été conçus pour la mesure de position [19], ainsi que pour les lectures de luminosité, rayonnement, température, bruit, contrainte, pression, position, vitesse et direction, ainsi que la surveillance de la santé [55]. Les capteurs sont généralement assez petits et peuvent donc être intégrés dans presque tous les environnements.

Le suivi et l'identification des personnes dans un environnement est un problème important dans les systèmes AmI. Si l'emplacement d'une personne est connu, le système peut la servir en anticipant les besoins en fonction de ses préférences et en fournissant des services en fonction des moments où ils sont généralement requis. La technologie souvent utilisée pour suivre des individus est le capteur de mouvement. Les capteurs de mouvement sont utilisés depuis des décennies comme colonne vertébrale des systèmes de sécurité. Cependant, bien qu'ils puissent détecter un mouvement, ils ne peuvent pas fournir d'informations permettant de distinguer qui (ou quoi) a

produit le mouvement. Les personnes et les objets peuvent porter un capteur permettant de les suivre. Un exemple de cette technologie, ce sont les étiquettes RFID qui peuvent être couplées à un lecteur RFID pour surveiller le mouvement des objets étiquetés. Cette technologie repose sur des individus et des objets marqués. De plus, la sensibilité des balises peut présenter des défis pour le système. Par exemple, si un lecteur RFID est placé dans un cadre de porte pour identifier les personnes faisant la transition entre des pièces, la personne peut déclencher le lecteur si elle s'approche de la porte, sans nécessairement passer à la pièce suivante. Ces ambiguïtés peuvent être résolues en intégrant plusieurs technologies. Par exemple, des capteurs de mouvement peuvent être placés de chaque côté de la porte en combinaison avec le lecteur RFID pour distinguer la proximité des transitions de pièce.

Toutes ces méthodes ont des limites et ne peuvent pas garantir une identification correcte dans tous les cas. Cependant, elles fournissent des outils intéressants et il appartient à l'intelligence du système de les associer à des techniques logicielles permettant d'atteindre l'objectif d'identification et de suivi. D'autres dispositifs de détection pouvant être utilisés pour identifier des personnes sont les microphones (via la façon de parler et l'identification explicite verbale) et les caméras vidéo (via la reconnaissance faciale ou des badges d'identification explicites).

Lors de l'analyse de données de capteurs, les systèmes AmI peuvent utiliser un modèle centralisé ou distribué [56]. Les capteurs du modèle centralisé transmettent les données à un serveur central, qui fusionne et analyse les données qu'il reçoit. Dans le modèle distribué, chaque capteur dispose de capacités de traitement intégrées et effectue un calcul local avant de communiquer des résultats partiels aux autres nœuds du réseau de capteurs. Le choix du modèle aura un effet considérable sur l'architecture de calcul et le type de capteur utilisé pour la tâche [57,58]. Dans les deux cas, les données des capteurs sont collectées à partir de sources différentes, puis combinées pour produire des informations plus précises et plus complètes.

7.2. **Raisonnement**

La détection et l'action permettent d'établir des liens entre les environnements et le système intelligent dans lequel elles fonctionnent. Afin de rendre ces systèmes réactifs, adaptatifs et avantageux pour les utilisateurs, un certain nombre de types de raisonnement doivent avoir lieu. Ceux-ci incluent la modélisation de l'utilisateur, la prédiction et la reconnaissance d'activité, la prise de décision et le raisonnement spatio-temporel.

7.2.1. La modélisation

Une caractéristique qui sépare les algorithmes informatiques généraux de ceux répondant à l'utilisateur est la possibilité de modéliser son comportement. Si un tel modèle peut être construit, il peut être utilisé pour personnaliser le comportement du logiciel AmI vis-à-vis de l'utilisateur. Si le modèle donne une ligne de base suffisamment précise, elle peut servir de base pour détecter les anomalies et les modifications des profils de résidents. Si le modèle a la capacité de s'affiner, l'environnement peut alors potentiellement s'adapter à ces modèles changeants. Dans cet aperçu, nous caractérisons les approches de modélisation d'utilisateur AmI en fonction de trois caractéristiques : (a) les données utilisées pour construire le modèle, (b) le type de modèle construit, la source de données la plus commune et (c) la nature de l'algorithme de construction du modèle. (Supervisé, non surveillé).

La source de données la plus commune de bas niveau constitue la source de données la plus courante pour la modélisation. Ces données sont faciles à collecter et à traiter. Cependant, un défi présent à l'utilisation de ces données à faible niveau est la nature volumineuse de la collecte des données. Dans le projet de maison intelligente MavHome [59], par exemple, les seules informations collectées sur le mouvement et l'éclairage génèrent en moyenne 10 310 événements par jour. Dans ce projet, un préprocesseur d'exploration de données identifie les modèles séquentiels communs dans ces données, puis les utilise pour créer un modèle hiérarchique du comportement des résidents. Loke [60] s'appuie également sur ces données de capteur pour déterminer l'action résidente et l'état du périphérique, puis extrait des informations de situations similaires pour créer un environnement sensible au contexte. A l'instar du projet MavHome, la recherche iDorm menée par Doctor, et al. [61] se concentre sur l'automatisation d'un cadre de vie. Cependant, au lieu d'un modèle de Markov, ils modélisent le comportement des résidents en apprenant des règles floues qui mappent l'état du capteur sur les lectures d'actionneur représentant les actions des résidents.

La quantité de données créée par les capteurs peut être un défi de calcul pour les algorithmes de modélisation. Cependant, le défi est encore plus grand pour les chercheurs qui incorporent des données audio et visuelles dans le modèle résident. Luhr [62] utilise les données vidéo pour trouver des règles d'association (séquentielles) dans les actions résidentes. Ces règles constituent ensuite la base permettant d'identifier les comportements émergents et anormaux dans un environnement

intelligent. Brdiczka [63] a recours à la détection de la parole pour modéliser automatiquement des groupes d'individus en interaction. Moncrieff [64] utilise également des données audio pour générer des modèles résidents. Cependant, ces données sont combinées avec des données de capteur et des décalages horaires enregistrés, puis utilisées pour détecter des situations dangereuses en maintenant un niveau d'anxiété de l'environnement.

L'identification des interactions sociales est un thème récurrent dans les recherches AmI. En plus des travaux de Brdiczka, Laibowitz [65] ont également utilisé des réseaux de capteurs sans fil pour analyser la dynamique sociale dans les grandes réunions. Ils ont pu détecter des caractéristiques d'interaction clés telles que l'intérêt et l'affiliation à partir de données de capteurs dans des groupes de plus de 100 personnes.

7.2.2. Prédiction d'activité et reconnaissance

C'est la capacité de prédire et de reconnaître les activités qui se produisent dans des environnements intelligents. Une grande partie de ces travaux ont été réalisés dans le cadre d'enquêtes sur les environnements intelligents, où l'application AmI est axée sur un environnement unique doté de capteurs et conçu pour améliorer l'expérience du résident dans l'environnement [66]. Des exemples de tâches de reconnaissance sont énumérés dans le tableau 4. Notez que chacune de ces tâches de reconnaissance de l'activité constitue des services de base assez attendus des systèmes AmI énumérés dans chaque cas, la réalisation de la reconnaissance de ces activités à un niveau très satisfaisant est un formidable défi dans chaque Cas.

Les projets Neural Network House [67], Intelligent Home [68] et MavHome [69,70] contrôlent de manière adaptative les environnements domestiques en anticipant l'emplacement, les itinéraires et les activités des résidents. La prévision des lieux de résidence, voire des actions des résidents, permet au système AmI d'anticiper les besoins du résident et d'assister (ou éventuellement d'automatiser) l'exécution de l'action [71]. Les chercheurs ont exploré un certain nombre d'approches pour résoudre la reconnaissance des activités. Les approches diffèrent en fonction du type de données de capteur utilisé pour la classification et du modèle conçu pour apprendre les définitions d'activité.

Environnements	Activités	Exemples
Maison intelligente	modèles de style de vie	consommation de nourriture et sommeil appropriés
Hôpital	Prendre le médicament	assurez-vous que le bon médicament est pris en quantité suffisante
Bureau intelligent	Utilisation des ressources	des documents et des salles de réunion
Voiture intelligente	Comportement de conduite	pour augmenter la sécurité si le conducteur s'endort
Classe intelligente	Interaction professeur / étudiant	focaliser la caméra sur une partie du tableau ou sur le professeur
Rue surveillée	Surveillance du comportement	se concentrer sur la plaque d'immatriculation d'une voiture qui roule vite

Table 1-1 Exemples de problèmes de reconnaissance d'activité dans différents environnements.

7.3. **Action**

Les systèmes AmI relient le raisonnement sur le monde réel à travers les capteurs et les actions. Les dispositifs intelligents offre un mécanisme par lequel les systèmes AmI peuvent exécuter des actions et affecter les utilisateurs du système. Un autre mécanisme est à travers des robots. Les relations entre l'homme et les machines ont été largement explorées dans des histoires de science-fiction. Cependant, comme le souligne Turkle [72], observé que les enfants et les personnes âgées en interaction tendrement avec des animaux domestiques robotisés qui introduisent " la science-fiction dans la vie quotidienne ".. La recherche en robotique a progressé à un point tel que les utilisateurs n'ont plus besoin de savoir comment leur donner le moyen de se déplacer, mais peuvent formuler des requêtes telles que «Apporte-moi le médicament sur le comptoir». En effet, de tels assistants robots se trouvent déjà dans les maisons de retraite [73] et constituent un débouché pour entretenir le contact avec les personnes âgées. Les robots sont en mesure de fournir une gamme encore plus large de tâches d'assistance pour soutenir AmI. Ils peuvent surveiller les signes vitaux de leurs maîtres et fournir une stimulation conversationnelle.

Les robots sont maintenant capables d'exposer des émotions et des expressions beaucoup plus humaines qu'auparavant [74] et peuvent même influer sur les décisions humaines.

7.4. **Interaction homme machine**

Un aspect important de l'intelligence ambiante concerne l'interaction. D'un côté, il y a une motivation à réduire l'interaction homme-machine [75], car le système est sensé utiliser son intelligence pour déduire les situations et les besoins des utilisateurs à partir des activités enregistrées, comme si un assistant humain passif observait les activités se déroulant dans l'attente. Pour aider si (et seulement si) nécessaire. D'autre part, une diversité d'utilisateurs peut avoir besoin ou rechercher volontairement une interaction directe avec le système pour indiquer ses préférences et ses besoins.

Bien que les concepteurs de systèmes d'intelligence ambiante sont encouragés par les progrès qui ont été accomplis dans le domaine au cours des dernières années, une grande partie de ces progrès sont inutilisés du fait que les technologies sont difficiles ou contre nature pour les utilisateurs. Abowd et Mynatt [76] indiquent que l'entrée explicite doit maintenant être remplacée par d'autres moyens de communication de la vie-humaine et des actions implicites. La maturation de technologies telles que le suivi des mouvements, la reconnaissance des gestes [77], la reconnaissance de l'expression faciale [78] et la reconnaissance des émotions [79], le traitement de la parole [99] facilite les interactions naturelles avec des environnements intelligents. Dans certains cas, divers mécanismes d'interface sont combinés pour former des interfaces multimodales [80].

Le paradigme classique de Windows, icônes, souris et pointeur (WIMP), basé uniquement sur l'utilisation du clavier, de la souris et de l'affichage, ne suffit pas pour répondre à tous les besoins qui naissent de l'interaction avec Smart Environnements. Les utilisateurs interagissent en utilisant leurs corps et leurs sens, de manière multimodale. De plus, le contexte acquiert un rôle fondamental dans le dialogue entre utilisateurs et technologies. Contrairement au paradigme pc, où il existe un ensemble fixe de périphériques d'E / S et dont l'utilisation est largement normalisée, dans le nouveau domaine des environnements AmI et Smart Environnements, le vocabulaire de l'interaction et les périphériques d'E / S à utiliser phase exploratoire, loin d'une définition et d'une normalisation fixes, reste ouverte à un éventail presque infini de possibilités.

En effet, la phase d'évolution actuelle de l'Infrastructure pour l'intelligence ambiante est toujours confrontée à ces défis :

- interaction multimodale ;
- De nouvelles métaphores pouvant s'ajouter aux nouveaux scénarios ;
- Nouveaux dispositifs d'entrée / sortie pouvant intercepter les interactions implicites / explicites de l'utilisateur et communiquer avec tous ses sens ;
- Nouveau vocabulaire et grammaire d'interaction.

La vaste gamme de dispositifs informatiques disponibles, avec des puissances de calcul et des capacités d'entrée / sortie différentes, signifie que l'avenir de l'informatique comprendra vraisemblablement de nouveaux modes d'interaction. Certaines de ces méthodes incluent les gestes, la parole, le regard et bien d'autres. Pour donner un bref aperçu des technologies qui caractérisent actuellement les interfaces utilisateur du système AmI, nous pouvons commencer par la façon dont les êtres humains les perçoivent et dont elles interagissent.

7.5. **Les défis de la confidentialité et de la sécurité**

La sécurité des systèmes informatiques et la protection de la confidentialité et de la vie privée des usagers sont deux caractéristiques importantes que doivent posséder les services d'intelligence ambiante. Les environnements d'intelligence ambiante peuvent comporter de multiples sources d'informations dont il faut contrôler et protéger l'accès tout en respectant les aspects éthiques et juridiques. Mettre en place des mécanismes fiables de contrôle d'accès aux services, et des systèmes de chiffrement et déchiffrement des données de la vie privée, sont des problématiques difficiles et complexes qu'il faut impérativement aborder pour permettre à des personnes dépendantes d'évoluer et d'interagir de manière sécurisée avec les services de l'intelligence ambiante sans atteinte à leur vie privée.

8. Domaines d'application du système ambiant intelligent

L'intelligence ambiante peut avoir un impact considérable sur nos vies dans de nombreux domaines. Il existe de nombreuses utilisations potentielles pour un environnement intelligent. En effet, nous prévoyons que les caractéristiques des environnements intelligents se répandent dans nos vies entières. Elles automatiseront certains aspects de notre vie, augmenteront la productivité au travail, personnaliseront nos expériences d'achat et accompliront toutes ces tâches, amélioreront

également l'utilisation de ressources telles que l'eau et l'électricité. Dans cette section, nous citerons les différents domaines d'applications AmI .(Voir figure 1.7).

8.1. **Maison intelligente**

Un exemple d'environnement enrichi avec l'intelligence ambiante est une "maison intelligente". Plusieurs artefacts et objets dans une maison peuvent être enrichis de capteurs pour collecter des informations sur leur utilisation et, dans certains cas, même pour agir indépendamment, sans intervention humaine. Certains exemples de tels dispositifs sont l'électrodomestique (par exemple une cuisinière et un réfrigérateur), des articles ménagers (par exemple des robinets, un lit et un canapé) et des dispositifs de traitement de la température (par exemple la climatisation et des radiateurs). Les avantages attendus de cette technologie peuvent être : (a) une sécurité accrue (par exemple, en surveillant les habitudes de vie ou les activités les plus récentes et en fournissant une assistance lorsqu'une situation potentiellement dangereuse se développe), (b) un confort (en ajustant automatiquement la température), et (c) économie (par exemple, en contrôlant l'utilisation des lumières). Il s'agit d'une utilisation répandue de nombreuses technologies telles que les badges actifs [81] et les systèmes de positionnement intérieurs [82].

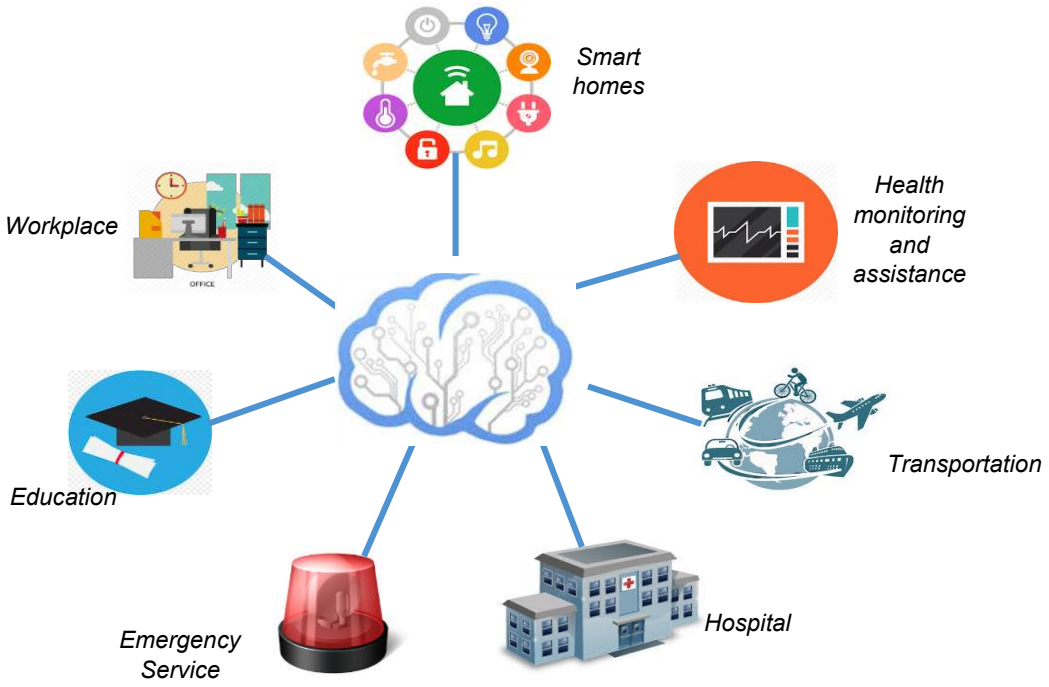

Figure 1.7 Domaine d'application

8.2. **Surveillance de la santé et assistance**

Beaucoup de raisons ont incité les développeurs à intégrer les technologies intelligentes dans le domaine de la santé. Une des raisons est l'émergence d'entreprises qui prennent des initiatives pour introduire des technologies intelligentes de soins aux personnes âgées à la maison [85]. Ainsi que l'énorme besoin de recherche sur l'intelligence ambiante pour améliorer la qualité de vie des personnes handicapées. La nécessité de la technologie dans ce domaine est évidement de regarder nos données démographiques actuelles et futures projets. En 2040, 23% de la population sera âgée de plus de 65 ans [83] et plus de 11 millions de personnes souffriront de démence liée à la maladie d'Alzheimer [84], il est prévu à long terme des pertes totales à l'économie américaine de près de 2 trillion de dollars [86]. Certaines de ces technologies mettent l'accent sur l'assurance, ou de faire en sorte que nos amis et proches soient en sécurité et en bonne santé à la maison par des techniques AmI pour reconnaître les activités, surveiller le régime alimentaire et détecter les changements ou les anomalies.

8.3. **Les hôpitaux**

Les applications d'AmI dans les hôpitaux peuvent aller de l'amélioration de la sécurité pour les patients et les professionnels au suivi de l'évolution des patients après une intervention chirurgicale. De nombreuses technologies AmI présentes dans les maisons intelligentes peuvent être adaptées pour être utilisées dans des pièces ou des zones spécifiques d'un hôpital [87].Les hôpitaux peuvent accroître l'efficacité de leurs services en surveillant la santé et les progrès des patients en analysant les activités menées dans leurs chambres. Ils peuvent également accroître la sécurité en ne permettant, par exemple, qu'au personnel autorisé et aux patients d'avoir accès à des zones et à des dispositifs spécifiques. Les capacités d'intelligence ambiante peuvent être utilisées dans ce contexte pour réduire la lourde charge des infirmières dans les établissements de soins assistés et de les sensibiliser plus rapidement aux besoins des résidents.

8.4. **Transport**

Les moyens de transport sont également des paramètres précieux pour les technologies AmI. Nous passons une partie importante de notre vie à voyager de différentes manières. Les gares, les bus et les voitures peuvent être équipés d'une technologie capable de fournir des connaissances fondamentales sur le fonctionnement du système à tout moment. Sur la base de cette connaissance,

des actions préventives peuvent être appliquées et l'expérience des personnes utilisant ce transport peut être augmenté en utilisant le système plus efficacement. Les transports publics peuvent tirer parti de la technologie AmI, notamment la localisation spatiale basée sur le GPS, l'identification des véhicules et le traitement des images, afin de rendre les transports plus fluides et donc plus efficaces et plus sûrs. Pour l'exemple, nous pouvons citer le projet I-VAITs [88] visant à aider les conducteurs en collectant des informations importantes par le biais de la façon dont ils utilisent les différents éléments de la voiture (pression sur les pauses) ou de leurs mouvements et du traitement d'image des expressions du visage du conducteur (comme indicateurs d'humeur).

8.5. **Services d'urgence**

Les services liés à la sécurité, tels que les pompiers, peuvent améliorer la réaction à un danger en localisant plus efficacement le site d'un accident et en préparant un itinéraire pour atteindre le lieu en liaison avec les services de voirie. Cela peut être réalisé à l'aide du traitement des images et de la surveillance du trafic, comme dans le projet e-Road [89]. Ce service peut également localiser rapidement un endroit où un danger est présent ou susceptible de se produire et en préparer un meilleur accès pour le personnel de sécurité. De même, le projet PRISMATICA [90] utilise des caméras pour surveiller les emplacements des transports en commun. En détectant des situations telles que le surpeuplement, la présence de personnes ou d'objets qui ne bougent pas, le mouvement dans une direction interdite et l'intrusion. L'environnement et les responsables peuvent réagir rapidement pour assurer la sécurité des personnes utilisant les transports en commun.

8.6. **L'Éducation**

L'intelligence ambiante peut également aider à améliorer l'expérience d'apprentissage des étudiants. Les établissements d'enseignement peuvent utiliser la technologie pour suivre la progression des étudiants dans leurs tâches et la fréquence de leur participation aux événements clés. Le projet Georgia Tech Classroom 2000 [91], fournit des interfaces homme-machine via des dispositifs tels qu'un tableau blanc interactif qui stocke le contenu dans une base de données. La classe intelligente de Shi et al. [92] utilise également un tableau blanc interactif et permet aux intervenants d'écrire des notes directement au tableau à l'aide d'un stylo numérique. Cette expérience en classe est encore renforcée par une vidéo et des microphones reconnaissant un ensemble de gestes, de mouvements et de paroles pouvant être utilisés pour récupérer des informations ou attirer l'attention sur des affichages et des supports appropriés.

8.7. **Les lieux de travail**

Les environnements de production peuvent également être enrichis avec la technologie AmI afin d'accroître des aspects importants du processus, tels que la sécurité des employés. Le système MOSES [93] utilise AmI pour déduire où se trouve le personnel et quelles tâches sont effectuées. Le système s'appuie sur la technologie RFID pour reconnaître le positionnement des éléments importants de l'environnement. Comme les travailleurs sont équipés de lecteurs RFID, le système peut suivre le développement des activités et peut donc indiquer à l'employé quelles tâches restent à accomplir. IShopFloor [94] fournit une architecture pour la planification, la détection et le contrôle de processus intelligent de fabrication. Le système repose sur trois agents principaux : les agents de ressources (dispositifs de fabrication), les agents de produit / pièces (pièces) et les agents de service (coordination des agents de ressources et des pièces).

9. Conclusion

L'intelligence ambiante a changé notre façon de vivre, déplaçant les interactions entre les personnes à un niveau virtuel dans plusieurs contextes allant de la vie professionnelle aux relations sociales. Dans ce chapitre, nous avons examiné la notion d'intelligence ambiante et les domaines émergents associés au sein de l'informatique. Nous avons souligné qu'un élément essentiel de ce domaine est la distribution de technologie intelligemment orchestrée pour permettre à un environnement d'être bénéfique à ses utilisateurs. Nous avons illustré le concept en décrivant un certain nombre de domaines d'application possibles.

Chapitre 2 : Les systèmes sensibles au contexte dans les environnements intelligents

1. Introduction

Ambiant Intelligence (AmI) concerne des environnements électroniques sensibles et adaptatifs qui répondent aux actions des personnes et des objets et à leurs besoins. Cette approche inclut tout l'environnement, y compris chaque objet physique, et l'associe à une interaction humaine. L'option d'une interaction plus étendue et plus intuitive devrait permettre d'améliorer l'efficacité, la créativité et le bien-être personnel. En raison des progrès de la technologie des capteurs, ces derniers sont devenus de plus en plus puissants, économiques et de plus petite taille, ce qui a stimulé les déploiements à grande échelle. En conséquence, nous avons aujourd'hui un grand nombre de capteurs déjà déployés et il est prévu que leur nombre augmentera rapidement au cours de la prochaine décennie [95].Ces capteurs génèrent en permanence d'énormes quantités de données. Cependant, pour ajouter une valeur aux données brutes des capteurs, nous devons les comprendre. La collecte, la modélisation, le raisonnement et la distribution du contexte en relation avec les données du capteur jouent un rôle crucial dans ce défi. L'informatique Sensible au contexte a joué un rôle important dans la résolution de ce problème dans les environnements intelligents, ce qui nous porte à penser que le paradigme de l'intelligence ambiante continuerait à être efficace. La sensibilité au contexte nous permet de stocker des informations de contexte liées aux données du capteur afin que l'interprétation puisse être faite facilement et de manière plus significative.

Ce chapitre présente un état de l'art sur la notion de contexte et sur les Systèmes sensibles au contexte. Nous présentons la notion de contexte, nécessaire à ces systèmes, exposés sous différentes définitions, caractéristiques et dimensions. Dans cette partie, nous attribuons une attention particulière aux différentes modélisations de contexte illustrées à la littérature et au processus de gestion de contexte qui représentent le fondement des systèmes sensibles au contexte.

2. Définition du contexte

Les caractéristiques communes dans les définitions d'un système AmI est qu'il doit être sensible au contexte, capable de comprendre et de répondre aux besoins des utilisateurs, sans être intrusif. Cela signifie que le système a besoin d'acquérir toutes les informations qui sont utiles pour comprendre la situation et de soutenir correctement l'utilisateur. Aujourd'hui, nous sommes habitués à traiter les services qui tirent parti de certaines connaissances de notre contexte. Pour utiliser efficacement la notion de contexte, de nombreux chercheurs ont commencé à comprendre la signification de contexte et comment il peut être utilisé. Schilit et Theimer [96] ont été les premiers à suggérer une définition du contexte comme étant l'emplacement de l'utilisateur, les identités et le statut des personnes et des objets qui l'entourent, Ils indiquent tout simplement que les aspects les plus importants du concept de contexte peuvent être identifiés en répondant aux questions suivantes : "Où est l'utilisateur ?", "Avec qui est-il ? ", " Quelles ressources sont à proximité ? ".Une autre définition est ensuite proposée par Brown et al [97], qui considère le contexte en tant qu'identité de l'utilisateur, des personnes et des objets qui l'entourent, sa localisation géographique, son orientation, la saison et la température où il évolue. Ryan, Pascoe, et Morse [98] définissent le contexte comme la localisation, l'environnement, l'identité, et le temps de l'utilisateur. Par conséquent, ces définitions ne peuvent pas non plus être utilisées pour identifier un nouveau contexte. Abowd et Mynatt [99] ont identifié les cinq questions (Qui, Quoi, Où, Quand, Pourquoi) comme informations minimales qui sont nécessaires pour comprendre le contexte. Dey [100] fournit une autre définition qui présente le contexte comme étant l'état émotionnel de l'utilisateur, le centre d'intérêt, la localisation et l'orientation, la date et le temps, les objets et les personnes dans l'environnement de l'utilisateur. De nombreuses définitions de contexte ont été écrites. Selon Zimmermann [101], il existe des définitions qui essayent d'être générales, mais elles risquent d'être récursives et inutiles, tandis que d'autres définitions énumèrent des éléments qui constituent le contexte, mais ils risquent de ne pas être complets ou relatifs uniquement à certains scénarios spécifiques. La définition du contexte donnée par Franklin et Flaschbart [102] est la situation de l'utilisateur. De même, Hull et al. [103] décrivent le contexte comme les aspects de la situation actuelle.

Dey et al. [104] ont évalué et souligné les faiblesses de ces définitions. Dey a assuré que la définition fournie par Schilit et Theimer [96] était basée sur des exemples et ne pouvait pas être

utilisée pour identifier un nouveau contexte. En outre, Dey a indiqué que les définitions fournies par Brown [97], Franklin et Flachsbart [102], Rodden et al. [105], Hull et al. [106] ont utilisé des synonymes pour faire référence au contexte, tels que l'environnement et la situation. Il a affirmé que ces définitions étaient trop spécifiques et ne pouvait pas être utilisées pour identifier le contexte dans un sens plus large et a fourni une définition du contexte comme suit : *"toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité est une personne, un endroit ou un objet considéré comme pertinents pour l'interaction entre un utilisateur et une application, y compris l'utilisateur et les applications elles-mêmes* ".

Selon cet aperçu dans les définitions que nous avons vu précédemment, on peut noter que le contexte prend à chaque fois la même direction, avec des vues différentes. Toutes les définitions précédentes se rencontraient dans la définition générale de Dey, et il est considéré comme la référence dans le domaine de l'informatique sensible au contexte.

3. La sensibilité au contexte

Le terme de la sensibilité au contexte, également appelé en anglais "Context-awareness", a été introduit pour la première fois par Schilit et Theimer [97] en 1994 comme la capacité d'une application à découvrir et à réagir aux modifications dans l'environnement où se trouve l'utilisateur. Après quelques années, il a été défini par Ryan et al [98], comme étant la sensibilité au contexte qui décrit la capacité d'un système à percevoir et à agir sur des informations sur son environnement, telles que la localisation, le temps, la température ou l'identité de l'utilisateur. Dans les deux cas, l'accent était mis sur les applications et les systèmes informatiques. Comme indiqué par Abowd et al [106], ces définitions sont trop spécifiques et ne peuvent pas être utilisées pour déterminer si un système donné est un système sensible au contexte ou non. Par conséquent, Dey [109] considère qu'un système est sensible au contexte s'il utilise ce contexte pour fournir à l'utilisateur des informations ou des services pertinents, où la pertinence dépend de la tâche exécutée par l'utilisateur. Lieberman et Selker [107] considèrent que les applications sensibles prennent en compte les données qui ne sont pas explicitement fournies (c'est-à-dire les données collectées ou récupérées par elles-mêmes). Dans une telle vue de la sensibilité contextuelle, aucune donnée n'appartient à l'entrée ou à la sortie explicite de l'application, mais affecte néanmoins son traitement et appartient ensuite à son contexte. Contrairement aux applications traditionnelles où

le système reçoit des données d'entrée fournies explicitement par l'homme (entrée) et s'appuie uniquement sur ces entrées pour produire une sortie claire (sortie).Selon Dourish [108] lorsque l'informatique sort de l'environnement de bureau, qui est un environnement statique qui ne change pas comme aujourd'hui, il était nécessaire d'identifier l'environnement, de suivre l'état dans lequel la technologie a été utilisée.

Les systèmes sensibles au contexte proposent une adaptation guidée par le contexte pour personnaliser le contenu et les services qu'il convient d'utiliser sous des circonstances (Temporelles, spatiales, matérielles, physiques et environnementales).

4. Architecture des systèmes sensibles au contexte

Les systèmes sensibles au contexte ont pour objectif l'adaptation au changement lors de la détection de situations pertinentes, par le biais d'informations obtenues à partir de capteurs, ainsi que l'interprétation et l'analyse de données pour détecter les modifications pertinentes. Dey[109] est l'un des premiers chercheurs à avoir généralisé la notion de contexte en spécifiant trois étapes nécessaires à savoir : (i) la capture du contexte, (ii) son interprétation qui permet de passer à une représentation de haut niveau plus exploitable pour l'application et finalement (iii) fournir ces informations à l'application et procéder à la séparation de l'acquisition de contexte de son utilisation dans les applications. Il y a un accord universel, dans ces travaux [110][111][112], sur la séparation entre la capture des informations de contexte et l'utilisation de ces informations afin de garantir l'extensibilité et la réutilisation du système.

On peut distinguer deux cycles principaux dans les systèmes sensibles au contexte (cycle de vie de contexte, cycle d'adaptation).

4.1. **Cycle de vie de contexte**

Le cycle de vie de contexte montre comment les données brutes se déplacent d'une phase à l'autre dans les systèmes logiciels (par exemple une application, un middleware). Plus précisément, il explique où les données sont générées et où elles sont consommées. Les interactions de l'utilisateur avec le système et l'environnement peuvent entraîner des changements de contexte. Il est donc nécessaire que la gestion du contexte tienne compte de ces changements grâce à un processus itératif qui capture, publie, modifie, traite et stocke les informations

contextuelles. En outre, des décisions sont prises pour déterminer si certaines actions seront prises en fonction du contexte. Pour que les adaptations puissent prendre en compte ces changements, Bernardos et al. [113] ont identifié trois phases dans un système de gestion de contexte typique : l'acquisition du contexte, le traitement de l'information et le raisonnement et la décision. Charith Perera et al.[10] propose un cycle de vie de contexte comprenant tous les éléments essentiels avec un nombre minimal de phases, comme illustré à la figure 2.1.

Ce cycle de vie de contexte comprend quatre phases. Premièrement, le contexte doit être acquis de différentes sources. Les sources peuvent être des capteurs physiques ou des capteurs virtuels

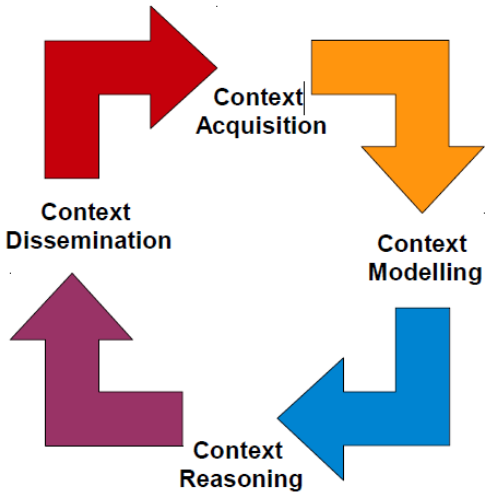

Figure 2.1 Cycle de vie de contexte [10]

(acquisition de contexte). Deuxièmement, les données collectées doivent être modélisées et représentées de manière significative (modélisation du contexte). Troisièmement, les données modélisées doivent être traitées pour dériver des informations de contexte de haut niveau à partir de données de capteur brutes de bas niveau (raisonnement de contexte). Enfin, les contextes de haut niveau et de bas niveau doivent être distribués aux consommateurs intéressés par le contexte (diffusion du contexte).

4.1.1. *Acquisition de contexte*

Les techniques utilisées pour acquérir le contexte peuvent varier en fonction du type de capteur et du processus d'acquisition.

✓ ***Acquisition basé sur les types de capteurs***

Différents types de capteurs peuvent être utilisés pour acquérir un contexte. En règle générale, le terme «capteur» est utilisé pour désigner des dispositifs matériels de capteurs tangibles. Cependant, dans la communauté technique, les capteurs sont appelés sources de données fournissant un contexte pertinent. Par conséquent, les capteurs peuvent être divisés en trois catégories [115] : physique, virtuelle et logique.

- **Capteurs physiques :** Ce sont les types de capteurs les plus couramment utilisés et ils sont tangibles. Ces capteurs génèrent des données de capteur par eux-mêmes. La plupart des appareils que nous utilisons aujourd'hui sont équipés d'une variété de capteurs (par exemple, la température, l'humidité, le microphone, le toucher). Une discussion sur les types de données de capteur et les capteurs couramment utilisés est présentée dans [116]. Les données extraites des capteurs physiques sont appelées contexte de bas niveau. Ils sont moins significatifs, vulnérables aux petits changements.
- **Capteurs virtuels :** Ces capteurs ne produisent pas forcément des données de capteur par eux-mêmes. Les capteurs virtuels sont basés sur des composants logiciels qui extraient des données de nombreuses sources et les publient sous forme de données de capteur (par exemple, calendrier, répertoire du numéro de contact, états Twitter, applications de messagerie électronique et de discussion en ligne). Ces capteurs n'ont pas de présence physique. Ils utilisent couramment la technologie des services Web pour envoyer et recevoir des données.
- **Capteurs logiques (également appelés capteurs logiciels) :** ils combinent des capteurs physiques et des capteurs virtuels afin de produire des informations plus significatives. Chaque type de capteur doit être lié à un composant logiciel permettant l'accès aux informations capturées. Ces composants sont généralement fournis avec des pilotes logiciels (drivers) et une API de communication avec les capteurs. Un service Web dédié à la fourniture d'informations météorologiques peut être appelé capteur logique. Les stations météorologiques utilisent des milliers de capteurs physiques pour collecter des informations météorologiques. Ils collectent également des informations à partir de capteurs virtuels tels que des cartes, des

calendriers et des données historiques. Enfin, les informations météorologiques sont produites en combinant des capteurs physiques et virtuels. En outre, le système d'exploitation mobile Android comprend un certain nombre de capteurs logiciels tels que des capteurs de gravité, d'accéléromètre linéaire, de vecteur de rotation et d'orientation.

- ✓ ***Acquisition basé sur le processus d'acquisition***

Il existe trois manières d'acquérir un contexte : La détection, La dérivation et la fourniture manuelle.

- **La détection** : Les données sont détectées par des capteurs, y compris les données détectées stockées dans des bases de données (par exemple, récupérer la température à partir d'un capteur, extraire les détails de rendez-vous à partir d'un calendrier).

- **La dérivation** : Les informations sont générées en effectuant des opérations de calcul sur les données du capteur. Ces opérations peuvent être aussi simples que des appels de service Web ou aussi complexes que des fonctions mathématiques exploitant des données détectées (par exemple, calculer la distance entre deux capteurs à l'aide de coordonnées GPS). Les données nécessaires devraient être disponibles pour appliquer toute technique de raisonnement numérique ou logique.

- **Fourniture manuellement :** Les utilisateurs fournissent manuellement des informations de contexte via des options de paramètres prédéfinies telles que les préférences (par exemple, comprenez qu'ils n'aiment pas recevoir de notifications d'événements entre 22h00 et 6h00). Cette méthode peut être utilisée pour récupérer tout type d'information.

4.1.2. *Modélisation de contexte*

Le contexte joue un rôle clé dans les systèmes sensibles au contexte pour la personnalisation du contenu et des services à l'aide du mécanisme d'adaptation. Les informations

de contexte sont utilisées en fonction des informations regardées et de leur représentation. Selon Brézillon [117], la représentation effective du contexte dans la machine du point de vue de la programmation ou de l'utilisation du contexte est un problème qui doit être résolu, à la fois en termes de modélisation des connaissances et de raisonnement. Les informations contextuelles peuvent être stockées, interprétées et gérées par la représentation explicite des informations contextuelles représentées dans le modèle de contexte. Dans un environnement intelligent, la modélisation de ces informations contextuelles dans un système est nécessaire car elle permet de gérer la sensibilité au contexte et l'adaptation. En d'autres termes, les capacités d'adaptation d'un Système Sensible au Contexte dépendent du modèle de contexte utilisé [118]. Ainsi, un modèle de contexte bien conçu est le fondement d'un Système Sensible au Contexte [119]. De toute évidence, le formalisme choisi pour représenter ce modèle est important, car il détermine les méthodes de raisonnement que le système peut utiliser pour effectuer certaines adaptations. Grâce à la littérature, nous pouvons observer que de nombreux modèles de contexte ont été proposés par la communauté de recherche [120]. Chaque modèle présente différents points de vue de la notion de contexte qui ont été étudiés dans différents domaines d'application (e.g. intelligence ambiante, systèmes mobiles de tourisme). Un modèle de contexte assure la définition de processus d'adaptation indépendant et isole ce processus des techniques d'acquisition de contexte, représentant ainsi la première exigence pour la maintenance et l'évolution des Systèmes Sensibles au Contexte [118].

4.1.3. *Raisonnement de contexte*

Le raisonnement contextuel peut être défini comme une méthode permettant de déduire de nouvelles connaissances et de mieux comprendre en fonction du contexte disponible [121]. Cela peut également être expliqué comme un processus consistant à donner des déductions de contexte de haut niveau à partir d'un ensemble de contextes [122]. L'exigence de raisonnement est également apparue en raison de deux caractéristiques du contexte brut : les imperfections (inconnues, ambiguës, imprécises ou erronées) et les incertitudes. Les performances de raisonnement peuvent être mesurées à l'aide de l'efficacité, de la solidité, de la complétude et de l'interopérabilité [123]. Le raisonnement s'appelle également l'inférence. Le raisonnement du concours comprend plusieurs étapes. En gros, nous pouvons les diviser en trois phases [124].

- **Prétraitement du contexte :** cette phase nettoie les données de capteur collectées. En raison de l'inefficacité du matériel du capteur et de la communication réseau, les données collectées peuvent être inexactes ou manquantes. Par conséquent, les données doivent être nettoyées en remplissant les valeurs manquantes, en supprimant les valeurs éloignées, en validant le contexte via plusieurs sources, etc. Ces tâches ont fait l'objet de nombreuses recherches de la part des communautés de recherche sur les bases de données, l'exploration de données et les réseaux de capteurs au cours de nombreuses années.
- **Fusion de données de capteurs :** Il s'agit d'une méthode de combinaison de données de capteurs provenant de plusieurs capteurs afin de produire des informations plus précises, plus complètes et plus fiables qui ne pourraient pas être obtenues avec un seul capteur [125]. Dans l'IoT, la fusion est extrêmement importante, car des milliards de capteurs seront disponibles. En conséquence, il existera un grand nombre de sources alternatives pour fournir les mêmes informations.
- **Inférence de contexte :** Génération d'informations de contexte de haut niveau à l'aide d'un contexte de niveau inférieur. L'inférence peut être faite dans une interaction unique ou dans plusieurs interactions. Reprenant un exemple sous un angle différent, W4 Diary [126] représentait le contexte sous forme de tuples (par exemple, qui : John, quoi : marcher : 4 km / h, où : ANU, Canberra, quand : 2013-01-05: 9h30). Ce contexte de bas niveau peut être déduit par un certain nombre de mécanismes de raisonnement pour générer les résultats finaux. Par exemple, lors de la première itération, les valeurs de longitude et de latitude d'un capteur GPS peuvent être déduites en tant que café PurplePickle à canberra. Lors de la prochaine itération, le café PurplePickle de canberra pourrait être considéré comme le café préféré de John. Chaque itération donne des informations plus précises et significatives.

4.1.4. *Diffusion de contexte*

La distribution de contexte est une tâche assez simple. Elle consiste en des méthodes qui fournissent un contexte aux consommateurs. Du point de vue du consommateur, cette tâche peut être appelée acquisition de contexte. Par conséquent, tous les facteurs dont nous avons discuté lors de l'acquisition du contexte doivent également être pris en compte pour la

distribution du contexte. En dehors de cela, il existe deux autres méthodes couramment utilisées dans la distribution de contexte :

- **Requête :** le consommateur de contexte effectue une demande sous forme de requête afin que le système de gestion de contexte puisse utiliser cette requête pour produire des résultats.
- **Abonnement (également appelé publier / souscrire) :** le consommateur de contexte peut être autorisé à s'abonner à un système de gestion de contexte en décrivant les exigences. Le système renvoie ensuite les résultats périodiquement ou lorsqu'un événement survient (dépassement du seuil). En d'autres termes, les consommateurs peuvent s'abonner à un capteur spécifique ou à un événement. Toutefois, dans les implémentations soulignées, les requêtes peuvent également être utilisées pour définir des abonnements. En outre, cette méthode est généralement utilisée dans le traitement en temps réel.

4.2. **Cycle d'adaptation**

La boucle générale d'un système adaptatif selon Da [127] inclut l'observation de l'environnement, le choix des adaptations et leur mise en œuvre. Ils l'ont appelé boucle CADA qui

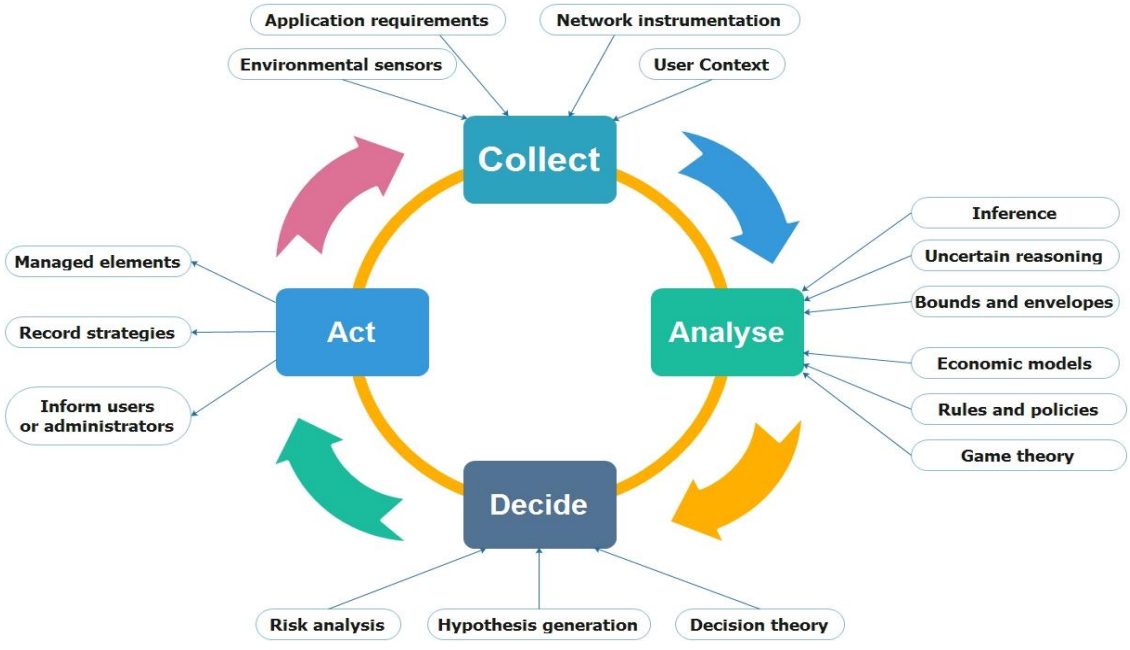

Figure 2.2 Boucle CADA : Collection, Analysis, Decision and Action [127].

signifie "*Collection, Analysis, Decision and Action*". La Figure 2.2 décrit une telle boucle d'adaptation dans laquelle le système peut être autonome ou impliquer l'humain.

Au début, les capteurs recueillent les données de l'environnement et l'utilisateur. Ces données sont exploitées pour générer un modèle abstrait de haut niveau afin de représenter les informations du contexte. L'analyseur estime ce modèle et génère des schémas d'adaptations. Il analyse le risque d'erreur pour chaque schéma et favorise le meilleur programme d'exécution pour l'adaptation [127] [128].

D'après le cycle de MAPE-k [129], [130] et l'idée de Da [127] L'adaptation dans l'informatique ubiquitaire est comprise comme étant le processus réactif déclenché par un événement spécifique ou un ensemble d'événements dans le contexte, l'objectif ultime étant d'améliorer la qualité de service perçue par l'utilisateur final. Ainsi, la condition fondamentale pour les applications qui suivent le paradigme informatique omniprésent est la capacité de détecter leur environnement, de raisonner en cas de changement de contexte et de réagir (si nécessaire) en conséquence. Nous définissons l'adaptation dans l'informatique ubiquitaire comme une boucle fermée (figure 2.3) comprenant les phases consécutives suivantes :

- Détection et traitement du contexte : pendant cette phase, toutes les données du contexte de l'utilisateur (telles que le bruit ambiant, la température actuelle et les préférences de l'utilisateur) et le contexte du système (telles que les ressources de calcul disponibles et les périphériques partagés) est collecté et souvent traduit en événements de contexte de haut niveau susceptibles de déclencher une adaptation du système.
- Raisonnement et planification de l'adaptation : Dans cette phase, le système auto adaptatif est appelé à raisonner sur le nouveau contexte et à décider sur ce qui doit être changé, dans le but d'atteindre l'objectif d'adaptation globale ;
- Action pour l'adaptation : Dans cette phase, les mécanismes d'adaptation appropriée sont utilisés pour mettre en œuvre les décisions d'adaptation prises par le processus de raisonnement.

Nous remarquons que ces deux approches se ressemblent énormément puisqu'elles contiennent les mêmes phases. Nous pouvons donc déduire qu'un système d'adaptation dans un environnement

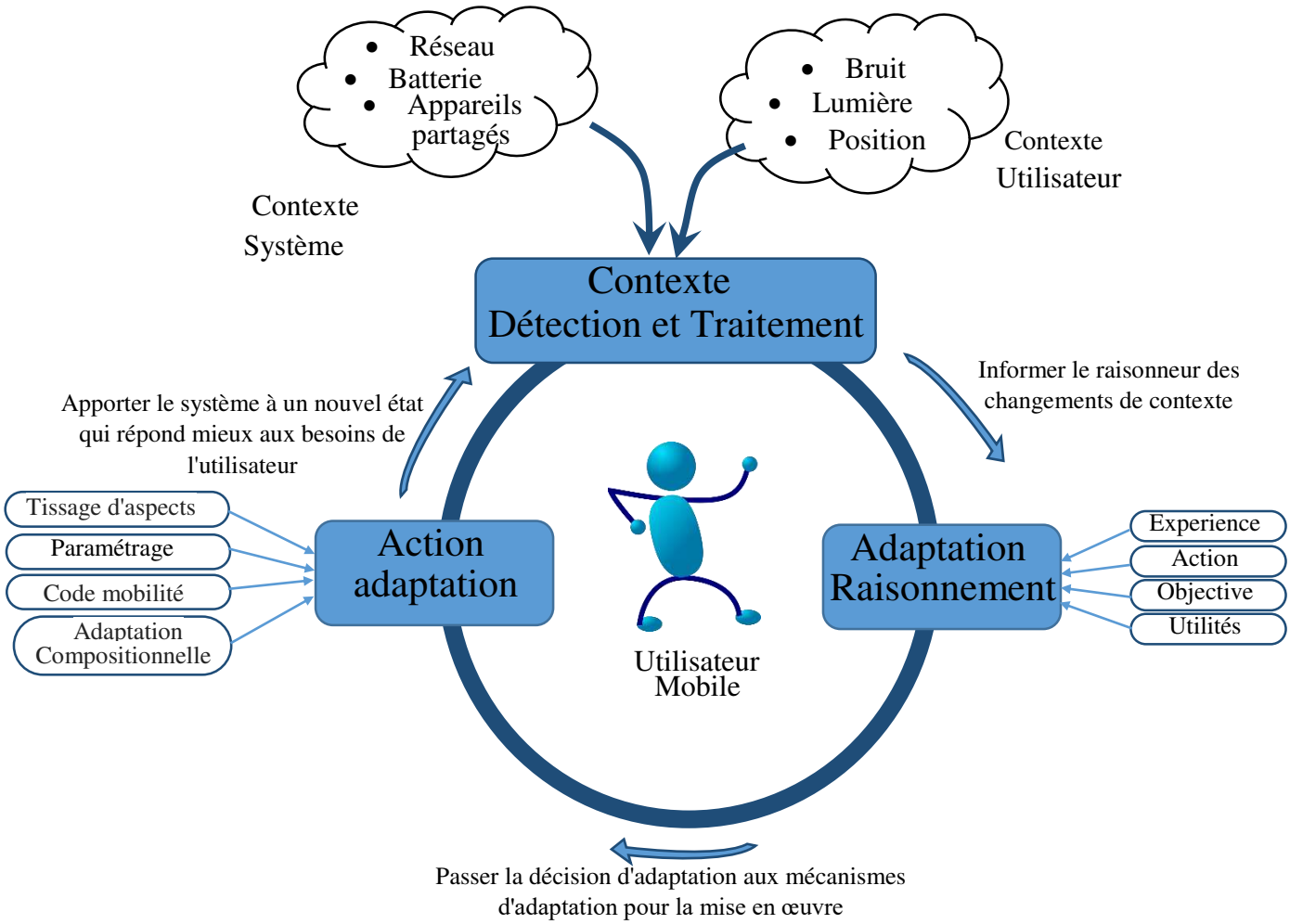

Figure 2.3 Boucle d'adaptation selon [129]

ubiquitaire possède quatre rôles : gestionnaire du contexte, planificateur, décisionnaire et middleware. Le rôle de planificateur et du décisionnaire étant les plus importants.

Tous ces travaux convergent vers une architecture qui se compose de cinq couches où chacune d'elle est responsable d'une fonction bien déterminée : (1) la couche capture du contexte, (2) interprétation ou modélisation du contexte, (3) la couche stockage du contexte, (4) la couche diffusion du contexte et enfin (5) la couche application.

5. Approches de Modélisation de contexte

Pour décrire la sensibilité d'une application à son contexte d'exécution, il faut déterminer les contextes auxquels cette application est sensible et les décrire dans un modèle. Par conséquent, la modélisation du contexte est la première étape dans le processus de création d'applications sensibles au contexte. Cette modélisation permet à l'application et/ou aux intergiciels de faciliter

l'interaction avec le contexte en fournissant une description abstraite des observables. La diversité des informations de contexte et leur utilisation dans divers domaines engendrent différentes façons de les modéliser.

Les techniques de modélisation de contexte les plus populaires sont décrites dans [131], [132]. Ces enquêtes portent sur un certain nombre de systèmes développés à l'aide des techniques suivantes. Chacune des techniques suivantes à ses propres forces et faiblesses. Nous discutons des techniques de modélisation de contexte à un niveau élevé. Les implémentations réelles de ces techniques peuvent varier considérablement en fonction du domaine d'application (par exemple, les détails de la mise en œuvre peuvent différer des environnements intégrés aux environnements mobiles ou aux environnements basés sur le Cloud). Par conséquent, nous nous concentrons sur la perspective conceptuelle de chaque technique de modélisation non sur une implémentation spécifique. Notre discussion est basée sur les six techniques de modélisation de contexte les plus populaires : clé-valeur, schémas de balisage, modélisation graphique, basée sur les objets, basée sur la logique et basée sur l'ontologie.

5.1. **Modélisation clé-Valeur**

Il modélise les informations de contexte sous forme de paires clé-valeur dans différents formats, tels que des fichiers texte et des fichiers binaires. C'est la forme de représentation de contexte la plus simple parmi toutes les autres techniques. Ils sont faciles à gérer lorsqu'ils contiennent moins de données. Cependant, la modélisation clé-valeur n'est ni évolutive ni adaptée au stockage de structures de données complexes. En outre, les structures hiérarchiques ou les relations ne peuvent pas être modélisées à l'aide de paires clé-valeur. Par conséquent, le manque de capacité de structuration des données rend difficile l'extraction efficace des informations modélisées. En outre, l'attachement d'une méta-information n'est pas possible. La technique clé-valeur est une technique orientée vers l'application et délimitée par l'application qui convient à l'objectif du stockage temporaire, tel que des configurations d'application moins complexes et des préférences utilisateur.

5.2. **Modélisation basée sur schéma de balisage**

Il modélise les données à l'aide de balises. Par conséquent, le contexte est stocké dans les balises. Cette technique est une amélioration par rapport à la technique de modélisation clé-valeur. L'utilisation des balises de marquage présente l'avantage de permettre une récupération efficace

des données. En outre, la validation est prise en charge via des définitions de schéma. Des outils de validation sophistiqués sont disponibles pour les techniques de marquage populaires telles que XML. La vérification de la plage est également possible dans une certaine mesure pour les valeurs numériques. Les schémas de balisage tels que XML sont largement utilisés dans presque tous les domaines d'application pour stocker temporairement des données, les transférer entre applications et entre des composants d'application. En revanche, les langages de balisage ne fournissent pas de fonctionnalités expressives avancées permettant le raisonnement. En outre, en raison de l'absence de spécifications de conception, la modélisation du contexte, la récupération, l'interopérabilité et la possibilité de réutilisation sur différents systèmes de balisage peuvent être difficiles. Une application courante de la modélisation à base de balisage est la modélisation de profils. Les profils sont couramment développés à l'aide de langages tels que XML. Cependant, le concept de langages de balisage ne se limite pas à XML. Tout langage ou mécanisme (JSON, par exemple) prenant en charge le stockage basé sur des balises permet la modélisation de schéma de balisage. Un exemple de modélisation de schéma de balisage populaire est Profils de capacités / préférences composites (CC / PP [133]). Il existe un nombre important d'applications émergentes similaires telles que ContextML [134] dans l'informatique contextuelle. Les tuples sont également utilisés pour modéliser le contexte [135].

5.3. **Modélisation graphique**

Il modélise le contexte avec des relations. Quelques exemples de cette technique de modélisation sont le langage UML (Unified Modeling Language) [136] et la modélisation ORL (Object Role Modeling) [137]. En termes de richesse expressive, la modélisation graphique est préférable à la modélisation de balisage et de clé-valeur, car elle permet de capturer des relations dans le modèle de contexte. La représentation réelle de bas niveau de la technique de modélisation graphique pourrait être modifiée. Par exemple, cela pourrait être une base de données SQL, une base de données noSQL, XML, etc. De nombreuses autres extensions ont également été proposées et implémentées en utilisant cette technique [138]. De plus, comme nous sommes familiers avec les bases de données, la modélisation graphique est une technique bien connue, facile à apprendre et à utiliser. Les bases de données peuvent contenir des quantités énormes de données et permettre des opérations de récupération de données simples pouvant être effectuées relativement rapidement. En revanche, le nombre de mises en œuvre différentes (c'est-à-dire différentes bases

de données et autres solutions) rend l'interopérabilité difficile. En outre, il existe des limitations sur les mécanismes de récupération de données tels que SQL. De plus, les exigences sophistiquées de récupération de contexte peuvent nécessiter l'emploi de requêtes SQL très complexes. Les requêtes peuvent être difficiles à créer, à utiliser et à gérer, même avec les outils sophistiqués existants. L'ajout d'informations de contexte et la modification de la structure de données sont également difficiles à des étapes ultérieures. Toutefois, certaines des tendances et solutions récentes du mouvement noSQL [129] permettent de résoudre ces problèmes de modification de structure. Par conséquent, les techniques de modélisation graphique peuvent être utilisées comme stockage persistant de contexte.

5.4. **Modélisation basée sur les objets**

Les concepts basés sur les objets (ou orientés objet) sont utilisés pour modéliser les données à l'aide de hiérarchies et de relations de classes. Le paradigme orienté objet favorise l'encapsulation et la réutilisation. Comme la plupart des langages de programmation de haut niveau prennent en charge les concepts orientés objet, la modélisation peut être facilement intégrée à des systèmes sensibles au contexte. Par conséquent, la modélisation basée sur les objets convient pour être utilisée en tant que mécanisme de modélisation, de manipulation et de stockage du contexte d'exécution interne, non partagé, basé sur du code. Cependant, il ne fournit pas de capacités de raisonnement intégrées. La validation des conceptions orientées objet est également difficile en raison de l'absence de normes et de spécifications.

5.5. **Modélisation basée sur la logique**

Les faits, les expressions et les règles sont utilisés pour représenter des informations sur le contexte. Les règles sont également utilisées par d'autres techniques de modélisation, telles que les ontologies. Les règles sont principalement utilisées pour exprimer des stratégies, des contraintes et des préférences. Elles offrent une richesse beaucoup plus expressive par rapport aux autres modèles discutés précédemment. Par conséquent, le raisonnement est possible jusqu'à un certain niveau. Les structures et langages spécifiques pouvant être utilisés pour modéliser le contexte à l'aide de règles sont variés. Cependant, le manque de standardisation réduit les possibilités de réutilisation et d'applicabilité. De plus, des techniques graphiques hautement sophistiquées et interactives peuvent être utilisées pour développer des représentations basées sur la logique ou sur des règles. Ainsi, même les utilisateurs non techniciens peuvent ajouter des règles et une logique

aux systèmes pendant l'exécution. La modélisation logique permet d'extraire de nouvelles informations de contexte de haut niveau en utilisant un contexte de bas niveau. Par conséquent, il a la capacité d'améliorer d'autres techniques de modélisation de contexte en jouant le rôle de complément.

5.6. **Modélisation basée sur l'ontologie**

Le contexte est organisé en ontologies à l'aide de technologies sémantiques. Un certain nombre de normes différentes (RDF, RDFS, OWL) et des capacités de raisonnement peuvent être utilisées en fonction des besoins. Une large gamme d'outils de développement et de moteurs de raisonnement est également disponible. Cependant, la récupération de contexte peut nécessiter beaucoup de temps de calcul et d'informatique lorsque la quantité de données est augmentée. Selon de nombreuses enquêtes, dans le contexte de l'informatique contextuelle et de la gestion des données de capteurs, les ontologies constituent le mécanisme privilégié de gestion et de modélisation du contexte, malgré ses faiblesses. En raison de sa popularité et de son adaptation plus large au cou rs des cinq dernières années, tant dans le monde universitaire que dans l'industrie, nous présentons une brève discussion sur la modélisation sémantique et le raisonnement. Cependant, notre intention n'est pas d'examiner les technologies sémantiques, mais de mettre en évidence l'applicabilité de la sémantique dans un domaine sensible au contexte du point de vue de l'AmI.

6. Le Web sémantique

Le Web sémantique (SW) peut être défini comme une extension du World Wide Web (WWW), qui permet aux systèmes de rechercher, combiner et traiter intelligemment des données Web en se basant sur la signification que ces données ont pour l'homme. [140]. Cette extension exploite le potentiel du Web car elle permet de partager efficacement des données sur Internet [141]. Les applications Web sémantiques peuvent être définies comme des solutions logicielles utilisant l'un des langages d'ontologie bien standardisés, tels que le RDF (Resource Description Framework) [142] ou Ontology Web Language (OWL) [143] pour prendre en charge l'échange et l'intégration de données, ainsi que la représentation et le raisonnement des connaissances [140]. Les instances de domaines d'application pour le SW comprennent l'échange de données Web et la syndication,

les wikis sémantiques, portails sémantiques, des métadonnées sémantiques dans des formats de données, web sémantique dans les sciences de la vie et ontologies pour la normalisation.

Le Web sémantique est une intuition qui surmonte les difficultés et le codage des services Web est compris sans ambiguïté. Comme souligné dans [139], la liaison et l'identification de ressources Web est l'un des principaux objectifs du Web sémantique, qui vise à faciliter leur utilisation, leur découverte, leur intégration et leur réutilisation. Grâce à des descriptions sémantiques exploitables et compréhensibles par les appareils, elles accèdent aux ressources que le Web sémantique doit fournir.

6.1. **L'Architecture du web sémantique**

La recherche sur le web sémantique a connu un développement important après la normalisation du langage Web ontologique (OWL) par le W3C [143]. Ce dernier est le résultat de la combinaison du langage de représentation de l'ontologie DAML (Darpa Agent Markup Language) avec la couche OIL (Ontology Inference Layer) [144]. OWL est une suite logique des normes RDF et XML. Basé sur XML, RDF est le langage de balisage sémantique pour les données. C'est le langage de représentation des ontologies OWL. RDF et sa suite RDF-S fournissent un modèle standard et simple de représentation des connaissances, qui peut être automatiquement exploité par une machine via des URI Web [145]. La figure 2.4 illustre cette architecture en couches du web sémantique [146].

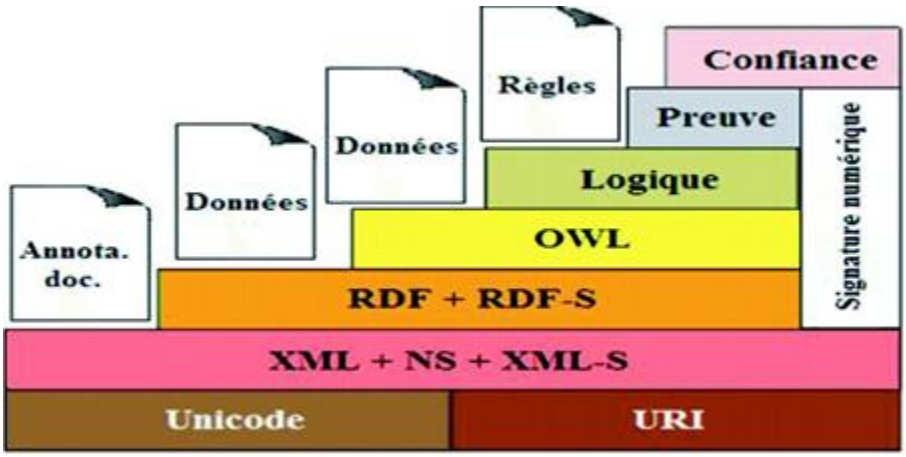

Figure 2.4 Architecture en couches du web sémantique [146]

6.2. **Les Ontologie**

L'ontologie découle d'un ancien concept philosophique, à savoir un système objectif existant de la catégorie philosophie, et concerne la nature abstraite de la réalité objective [147]. Au cours de la dernière décennie, la recherche en ontologie a de plus en plus mûri et dépasse le cadre de la philosophie, dans de nombreux domaines, tels que l'intelligence artificielle et les technologies de l'information.

En 1991, Neches, Fikes et al. [148] ont défini pour la première fois une ontologie dans le domaine de l'intelligence artificielle : «Une ontologie définit les termes de base et les relations constituant le vocabulaire d'un sujet, ainsi que les règles de combinaison des termes et des relations pour définir l'extension. Au vocabulaire. »Selon cette définition, l'ontologie indique que les termes ne sont pas seulement définis explicitement, mais dérivent également de règles. Gruber [149] donne une définition répandue de l'ontologie : «L'ontologie est la spécification claire d'un modèle conceptuel», impliquant deux significations : premièrement, elle fait référence à l'abstraction, à l'induction, conceptualisée dans un champ ; deuxièmement, il fournit une conceptualisation des résultats exprimés sous une forme unifiée pouvant être comprise par les humains et les ordinateurs. En se basant sur cette définition, Borst [150] est allé encore plus loin : «l'ontologie est la spécification claire d'un modèle conceptuel de partage». En 1998, Studer et al. [151] ont proposé une alternative plus détaillée, comprenant quatre aspects :

- **Conceptualisation** : modèles obtenus en faisant abstraction de phénomènes apparentés dans le monde objectif, qui se déroulent indépendamment de l'état spécifique de l'environnement.
- **Explicit :** définitions explicites pour les concepts utilisés et leur restriction.
- **Formel :** le modèle d'ontologie peut être traité par ordinateur.
- **Partager :** l'ontologie vise le consensus de groupe plutôt que le consensus individuel.

La définition ci-dessus de l'ontologie révèle que celle-ci est impliquée : terminologie (glossaire), relations, règles, spécification formelle, connaissance du domaine, expression et partage. En fait, l'ontologie définit la structure de connaissances de base d'un domaine par les concepts, la terminologie et la description normalisée des relations.

6.2.1. Langage de description d'ontologies

Récemment, l'industrie et les instituts de recherche ont développé divers langages de description d'ontologies basés sur différentes formes de représentation, illustrés à la figure 2.5. Le langage de description est basé sur la syntaxe XML (Extensible Markup Language), notamment : RDFS «Resource Description Framework», DAML + OIL (langage de balisage d'agent DARPA + langage d'échange d'ontologies) («DAML + OIL», 2001), OWL "OWL Web Ontology Language", etc. Le W3C (World Wide Web Consortium) a recommandé trois de ces langages comme modèles standards liés à l'ontologie : XML, RDF / RDFS et OWL.

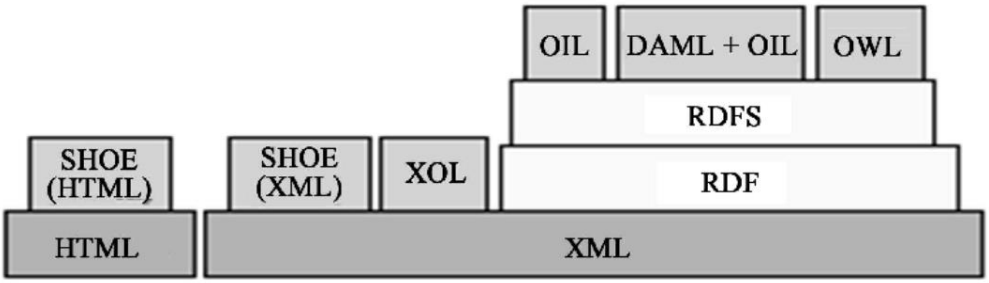

Figure 2.5 Famille de langages ontologiques [181]

- **XML**

XML est un langage de balisage qui définit un ensemble de règles pour coder des documents dans un format unifié «XML». Il s'agit d'un format de données textuel prenant fortement en charge, via Unicode, les langues du monde, mettant l'accent sur la simplicité, la généralité et la convivialité sur Internet. La technologie XML est un ensemble de normes et de protocoles recommandés et approuvés par le W3C, allant du réseau sous-jacent à la couche application impliquant la représentation, le traitement, l'échange et la remise de messages, etc. Elle contient plusieurs autres spécifications connexes, des normes ouvertes gratuites, telles que normes sous-jacentes : DTD (définition de type de document), normes de style : CSS (feuilles de style en cascade), critères de requête : XQL (langage de requête XML) et norme d'analyse : DOM (modèle d'objet de document). XML ouvre la voie à d'autres langages de description d'ontologies, en fournissant une base bien structurée.

- **RDF/RDF Schema**

RDF (Resource Description Framework) localise ci-dessus le XML dans la pile de langues ontologiques. RDF est un langage permettant de représenter des informations sur les ressources sur le Web, contenant des ressources, des propriétés et des instructions. Les instructions sont composées de trois composants : Subject, Predicate et Object. RDF fournit un modèle de données permettant d'exprimer les métadonnées Web à l'aide d'objets et de leurs relations. RDF définit non seulement un modèle triple, composé d'objet, d'attribut et de valeur, en tant que primitives de modélisation de base, mais également une syntaxe standard représentant un modèle triple. Selon une caractéristique du RDF, toute relation complexe peut être exprimée par la composition de plusieurs relations doubles simples. Par conséquent, le modèle de données RDF peut être utilisé pour décrire toutes les ressources sur le Web. RDF fournit un format lisible à la machine et à la lecture humaine pour permettre l'échange entre système et application, palliant ainsi le manque de sémantique de XML.

RDFS (schéma RDF) complète le RDFS. Il ajoute quelques nouvelles primitives sémantiques basées sur XML, étendant la description sémantique des données. RDFS fournit un modèle sémantique simple à lire par machine et constitue la base d'autres langages de description d'ontologies. Cependant, il ne prend pas en compte les conflits sémantiques.

- **DAML-OIL**

XML et RDF peuvent fournir une solution simple pour la description d'ontologies mais n'ont qu'une capacité d'expression limitée. Les chercheurs ont donc cherché à développer un langage modèle plus expressif. OIL (Ontology Interchange Language) est l'un d'entre eux, développé par le programme On-To-Knowledge en Europe. Ce langage modèle étend pour la première fois la syntaxe et la logique de description de RDF et la logique de description conjointe, le langage cadre et les normes Web.

Le DAML (DARPA Agent Markup Language) est le nom d'un programme de financement américain mis en place par la DARPA (Agence américaine de recherche sur les projets de défense avancée) en 1999. De nombreux instituts de recherche américains ont participé à cette étude. C'est une extension de RDF basée sur une technologie orientée objet. La version la plus ancienne de

DAML s'appelait DAML-ONT, avant de passer à DAML + OIL. C'est un langage issu de DAML et d'OIL qui combine les meilleures fonctionnalités des deux. DAML + OIL décrivent la structure du domaine par des classes et des attributs, améliorant ainsi la capacité de description du langage. Contrairement à RDF, DAML + OIL n'est pas un modèle de données, mais un langage de structure permettant de décrire le modèle de données RDF. Dans une certaine mesure, il peut être considéré comme un autre langage RDF, de sorte qu'il hérite également des faiblesses de RDF.

- **OWL**

OWL (Web Ontology Language) est une famille de langages de représentation des connaissances pour la création d'ontologies. Il a été approuvé par le directeur du W3C en tant que recommandation du W3C «langage d'ontologie Web OWL», situé en haut de la pile de langues d'ontologies. OWL a commencé comme une amélioration de DAML-OIL, s'appuyant sur sa conception et son expérience en matière d'application pour enrichir le mécanisme de définition sémantique. OWL, dans la syntaxe abstraite, contient une séquence d'annotations, d'axiomes et de faits. Les annotations sur les ontologies OWL peuvent être utilisées pour enregistrer la paternité et d'autres informations associées à l'ontologie, y compris les références d'importation à d'autres ontologies.

Le contenu principal de l'ontologie OWL est exprimé dans ses axiomes et faits, qui fournissent des informations sur les classes, les propriétés et les individus de l'ontologie «OWL Web Ontology Language». OWL fournit plus de moyens d'expression que XML et RDF / RDFS dans une expression sémantique, ajoutant plus de vocabulaire pour décrire les propriétés et les classes : entre autres, les relations entre les classes (par exemple la disjonction), la cardinalité (par exemple "exactement un"), l'égalité, le typage plus riche des propriétés et caractéristiques des propriétés (par exemple, symétrie) et des classes énumérées. Par conséquent, OWL fournit un puissant pouvoir d'expression de l'ontologie. En se concentrant sur différentes exigences, OWL fournit trois sous-langages de plus en plus expressifs : OWL Lite, OWL DL et OWL Full.

- **OWL Lite**

Fournit aux utilisateurs le sous-ensemble fonctionnel principal et de base d'OWL. Il supporte des fonctions de contrainte simples et est relativement facile à impliquer. Le W3C utilise un exemple simple pour interpréter cela : OWL Lite prend en charge la cardinalité, mais n'autorise que les

valeurs de cardinalité de 0 ou 1. Il existe plus d'outils de développement prenant en charge OWL lite que les deux autres sous-langues. Développer le processus est relativement facile.

- **OWL DL**

Une extension de OWL Lite, plus expressive que OWL Lite. OWL DL contient toutes les constructions du langage OWL, mais elles ne peuvent être utilisées que sous certaines restrictions. Le W3C utilise un exemple simple pour interpréter ceci : une classe peut être une sous-classe de nombreuses classes, une classe ne peut pas être une instance d'une autre classe. OWL DL fournit une logique de description pour les applications nécessitant une expressivité maximale tout en conservant la complétude informatique et la décidabilité, mais son vocabulaire est limité.

- **OWL Full**

Possède le sous-langage OWL le plus expressif, considéré comme une extension de OWL DL. Il est conçu pour les utilisateurs nécessitant une expressivité maximale et la liberté syntaxique de RDF sans aucune garantie informatique. Le W3C utilise un exemple simple pour interpréter cela : une classe peut être traitée à la fois comme un ensemble d'individus et comme un individu à part entière. Il permet d'ajouter une nouvelle ontologie au vocabulaire RDF, OWL RDFS prédéfini. Par conséquent, aucun logiciel de raisonnement ne pourrait prendre en charge le raisonnement complet pour chaque fonctionnalité de OWL Full.

Comme analysé ci-dessus, OWL a été choisi pour construire notre modèle de Profile sémantique. Dernier langage de représentation des connaissances approuvé par le W3C, il tire parti des anciens langages de description d'ontologies, en exploitant la capacité de XML à définir des schémas de balisage personnalisés et l'approche flexible de RDF pour la représentation des données. OWL fournit une compréhension commune des informations contextuelles afin de faciliter la modélisation et le raisonnement contextuels d'informations contextuelles imparfaites et ambiguës et de permettre le partage et la réutilisation des connaissances contextuelles.

7. Conclusion

Au début de ce chapitre, nous avons donné une définition du terme contexte puis nous avons montré l'importance dans les systèmes sensibles au contexte. On a décrit l'architecture d'un système sensible au contexte ainsi que les deux cycles principaux (cycle de vie de contexte, cycle

d'adaptation) et les verrous liés à l'acquisition, à la modélisation, au raisonnement, à l'adaptation et la personnalisation.

Nous avons aussi présenté les Web sémantique qui fournissent les bases de la représentation du contexte et du raisonnement dans notre solution de gestion de contexte dynamique, un cadre pour la représentation du contexte sous la forme d'informations RDF et pour l'échange de modèles de contexte sous la forme de graphiques RDF avec les fournisseurs de contexte et les consommateurs. RDFS et OWL-Lite fournissent l'expressivité requise pour raisonner sur des données contextuelles. Nous avons ainsi remarqué que les ontologies sont de plus en plus utilisées. Ceci est principalement dû aux propriétés formelles et à l'expressivité des ontologies, ainsi qu'aux moteurs d'inférence associés. Les ontologies permettent également d'assurer l'interopérabilité désirée au niveau sémantique, et donc la réutilisation de ces modèles avec une certaine cohérence sémantique. Ceci facilite donc la gestion

Chapitre 3 : Architectures et profil sémantique pour les environnements intelligents

1. Introduction

Un modèle de contexte est nécessaire pour définir et stocker les données de contexte dans un format pouvant être traité par une machine [152]. Un modèle bien conçu est un élément clé du contexte dans tout système sensible au contexte [153]. L'importance des modèles de contexte pour les systèmes sensibles au contexte ne saurait être suffisamment soulignée. Dans ce chapitre, nous examinerons d'abord les méthodes de modélisation du contexte existantes et les architectures sensible au contexte, ainsi que les comparaisons et les analyses.

2. Modélisation des profils sensibles au contexte

CC/PP [154] est une recommandation du W3C pour spécifier les capacités des terminaux ainsi que les préférences des utilisateurs selon le formalisme RDF. La structure d'un tel profil est très descriptive car elle liste des ensembles de valeurs, tels que la taille de l'écran, la version du navigateur, etc. Toutefois, CC/PP manque de structuration, en effet ce langage limite la description de profils complexes notamment en contraignant une hiérarchie stricte à deux niveaux. De plus, ce langage ne permet pas la description de relations et de contraintes complexes entre les informations de contexte. Enfin, pour intégrer de nouveaux éléments, il est nécessaire d'étendre le vocabulaire de CC/PP.

UAProf [155] constitue une extension du langage CC/PP pour les téléphones mobiles. Les éléments de son vocabulaire sont utilisés dans la même structure de base que celle employée pour CC/PP, néanmoins UAProf permet une description précise des capacités des dispositifs sans fil. Entre autres, il décrit des éléments tels que la taille de l'écran, les capacités multimédias, de caractères Unicode… etc. UAProf est un standard exploité par des milliers de téléphones mobiles mais ce langage se limite exclusivement à la description de caractéristiques matérielles de téléphones sans fils.

CSCP [156] se base également sur le formalisme RDF. Ce langage permet de décrire la localisation, les caractéristiques du réseau et les dépendances des applications. Il présente également une structure multiniveaux qui est extensible. Contrairement à CC/PP, il permet de

spécifier une description du contexte non limitée à deux niveaux hiérarchiques. Toutefois, il ne s'agit pas d'un standard international et cette proposition ne permet pas de décrire des relations ou des dépendances entre les informations du profil. Néanmoins, CSCP modélise des contraintes mais de très bas niveau, telles que certaines informations du profil qui ont des valeurs différentes selon certaines conditions. Malgré cette extension, ce langage reste non intuitif et difficile à utiliser pour décrire des informations complexes [157].

Saleemi et al [158] ont proposé la classification des informations de contexte à l'aide d'aspects multidimensionnels : le contexte physique d'une personne peut inclure le (lieu, heure, etc.) ; le contexte social peut inclure les (relations sociales famille, etc.), le contexte d'activité peut inclure toutes les tâches effectuées dans la vie quotidienne (regarder la télévision, écouter de la musique, parler au téléphone, etc.). Pour être efficace et partager entre de grands profils de contexte, le profil de contexte doit être aussi générique que possible et prendre en charge le mappage automatique entre les profils de différents utilisateurs. De plus, ce modèle ne prend pas en charge la mise en cluster de contextes d'utilisateurs similaires et leur situation actuelle.

Dromzée et al [159] ont proposé un modèle de profil d'utilisateur basé sur un service sémantique appelé profil générique sémantique. Ils ont représenté différents contextes d'informations sur l'utilisateur, le périphérique et le document en tant qu'ensemble de services. Ce travail peut être utile pour intégrer les normes de profils d'utilisateurs de différents utilisateurs susceptibles de donner lieu à une interopérabilité entre différents grands services en les mappant automatiquement à un profil d'utilisateur générique. Cependant, ils ne généralisent pas plusieurs profils de différents périphériques et utilisateurs.

Yus et al [160] ont présenté des informations de contexte composées de deux classes. La première est dynamique liée aux éléments de contexte qui changent de manière dynamique dans le temps. La seconde est une catégorie statique dans laquelle les propriétés de contexte ne changent pas. Récemment, de grandes sources de contexte dans différents domaines intelligents sont devenues disponibles. Les objets intelligents connectés permettent l'identification de situations urgentes. Ils fournissent une surveillance du contexte et des services appropriés. Les auteurs considèrent les services qui ne peuvent fournir que des informations contextuelles à une application spécifique et ne sont pas suffisants de notre point de vue. Ce travail manque de relations sémantiques intelligentes entre les contextes (par exemple, une activité alimentaire continue peut augmenter la

pression systolique), ce qui aide les utilisateurs à identifier rapidement les situations urgentes et à adapter les modifications du contexte de manière transparente et optimisée.

3. Les techniques d'identification de situations

L'identification de la situation est classée en deux catégories : les techniques basées sur la connaissance et les techniques non basées sur la connaissance. L'identification de situation basée sur la connaissance repose sur une conceptualisation du modèle de contexte encodé dans un format de machine interprétable par le cadre de définition de ressources (RDF). L'identification de situation non basée sur la connaissance utilise des règles événement-condition-action et d'autres techniques d'apprentissage qui permettent l'apprentissage automatique à partir de l'historique des événements et la détection de situations de la vie quotidienne à partir du contexte de données. Ces techniques ne permettent pas d'identifier simultanément plusieurs utilisateurs sur la base d'une situation sémantique en temps réel. Nous sommes intéressés par l'identification de situations urgentes basées sur le brouillard et l'ontologie en raison de la mobilité de l'utilisateur et des contextes d'utilisation. Le modèle de contexte sémantique est extrait et calculé en fonction de mesures de similarité d'ontologies. Gyrard et al. [161] ont proposé une approche basée sur une ontologie pour décrire formellement les métadonnées de contexte d'utilisateur afin d'améliorer l'assistance des utilisateurs dans leurs activités de la vie quotidienne et la prévision de certaines situations urgentes. L'outil proposé vise à collecter des données de contexte, à inférer et à raisonner sur ces données pour l'identification de la situation et la prise de décision. L'outil est basé sur des règles d'inférence fournies par des experts de domaine afin de générer les services appropriés. Plus récemment, Chabridon et al [162] ont proposé un mécanisme efficace pour identifier et calculer la qualité de la situation en utilisant une approche basée sur une ontologie et des critères de qualité agrégés en utilisant l'opérateur fuzzy Choquet. Cependant, ces travaux souffrent toujours de la sélection de service qui fournit des services appropriés aux utilisateurs.

4. Architecture et middleware sensible au contexte pour les environnements intelligents.

Avec la récente amélioration du Cloud Computing [163] les Framework et les middlewares sont devenus populaires et efficaces pour fournir des services appropriés aux utilisateurs au bon moment, au bon endroit et de la bonne manière. Cependant, le Cloud Computing est capable de faire face à la situation des utilisateurs dans différents lieux et conditions d'éclairage, en fonction du réseau Internet. Si un problème de connexion Internet est rencontré, le Cloud affectera potentiellement les performances dans des situations réellement urgentes et entravera la continuité du service.

4.1. SOCAM

Service-Oriented Context-Aware Middleware (SOCAM), développé par Gu et al [164], est une architecture destinée à la création et au prototypage rapide de services contextuels. SOCAM est conçu pour l'acquisition, la découverte, l'interprétation et la diffusion de contexte sur la base d'un ensemble de services indépendants. L'architecture est montrée ci-dessous. Elle se compose de cinq parties : fournisseur de contexte, interprète de contexte, base de données de contexte, services sensibles au contexte et services de localisation de services.

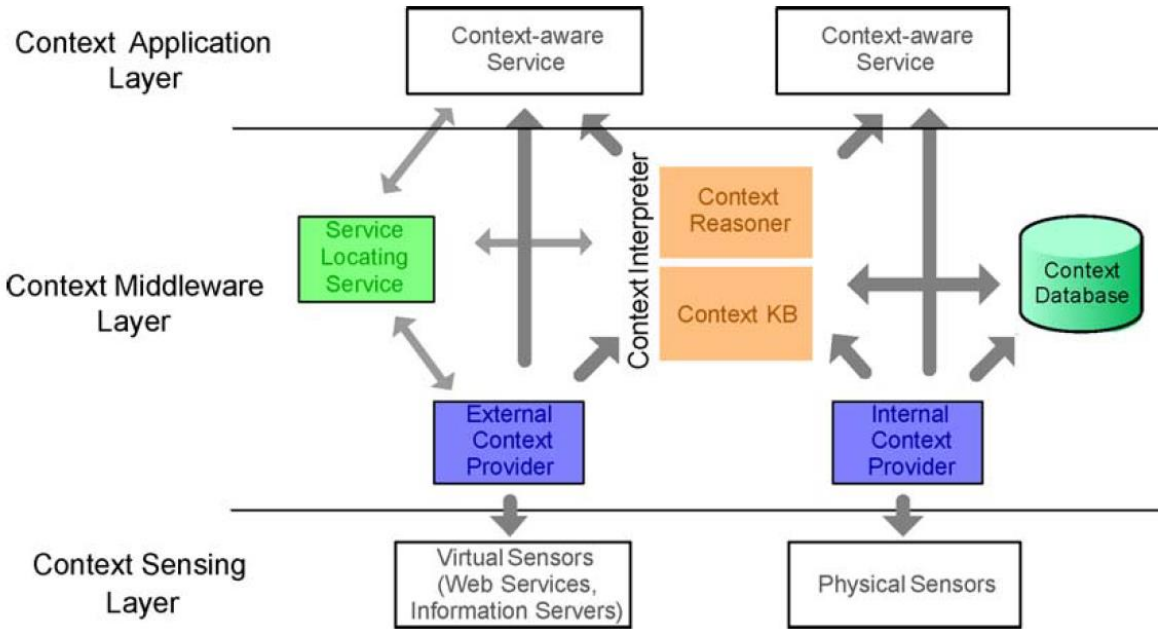

Figure 3.1 Architecture SOCAM [164]

Les fournisseurs de contextes ont deux tâches primordiales : ils peuvent extraire les contextes à partir de capteurs virtuels et physiques, et convertir le contexte brut en représentation OWL pour le partager et le réutiliser facilement par d'autres composants de service.

L'interpréteur de contexte est un composant fournissant des services de raisonnement logique pour traiter les informations de contexte. Il comprend la déduction de contextes de haut niveau à partir de contextes de bas niveau, l'interrogation de la connaissance, le maintien et la résolution des conflits de contextes. Il se compose de deux parties : un raisonneur et une base de connaissances de contexte. Le raisonneur de contexte prend en charge deux types de raisonnement : le raisonnement d'ontologie et le raisonnement basé sur des règles définies par l'utilisateur. En plus de contenir une ontologie de contexte et leurs instances, la base de connaissances de contexte fournit un ensemble d'API permettant à d'autres composants de service d'interroger, d'ajouter, de supprimer ou de modifier les connaissances de contexte.

La base de données contextuelle stocke des ontologies de contexte ainsi que des contextes passés pour un sous-domaine.

Les services contextuels font référence à des applications et à des services utilisant différents niveaux de contextes en adaptant leur comportement en fonction du contexte actuel.

Les services de localisation de services proposent le mécanisme de service de localisation de services [165]. Ils peuvent suivre et s'adapter aux changements dynamiques dans le fournisseur de contexte. Ils peuvent également permettre aux utilisateurs ou aux applications de localiser ces services.

SOCAM fournit un support efficace pour acquérir, découvrir, interpréter et accéder à divers contextes afin de créer des services sensibles au contexte. SOCAM utilise également une ontologie pour modéliser un contexte similaire à CoBrA. Mais, Il adopte une approche hiérarchique à deux couches pour la conception d'ontologies de contexte, qui sont divisés en ontologie supérieure commune pour les concepts généraux et en ontologie spécifique à un domaine pour différents sous-domaines. Il ajoute en outre divers caractères d'informations de contexte aux modèles de contexte tels que la classification et dépendance. Cependant, SOCAM se contente d'adopter la règle basée sur la logique du premier ordre pour déterminer le contexte de haut niveau et est inadéquate pour traiter la reconnaissance de contexte d'activité complexe dans l'informatique ubiquitaire.

4.2. Federated-Q

Naqvi et al [166] ont proposé des services en nuage à fourniture de contexte faiblement couplés, sensibles au contexte et à la qualité, pour adapter les services au contexte de l'utilisateur et de son appareil mobile. L'inconvénient est qu'il est nécessaire de réduire l'hétérogénéité sémantique dans un modèle générique permettant une adaptation plus avancée et automatisée à partir d'un grand nombre de profils d'utilisateurs hétérogènes. La figure 3.2 illustre le schéma directeur de notre infrastructure fédérée dans le Cloud et sur le périphérique mobile. Des modules d'adaptation aux

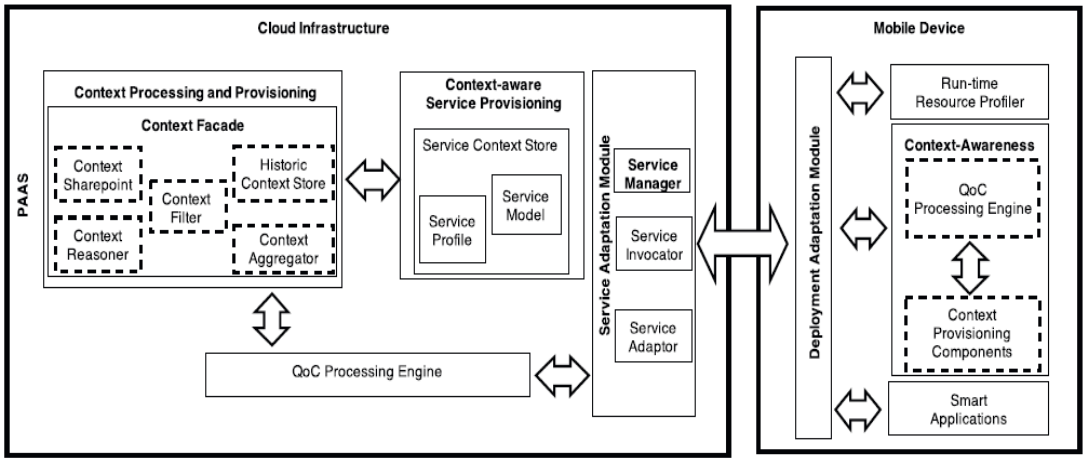

Figure 3.2 Schéma directeur de l'architecture de structure fédérée [166]

deux extrémités sont envisagés pour réaliser la fédération entre les deux paradigmes pour le développement et le déploiement d'applications intelligentes. Les principales caractéristiques du cadre sont décrites comme suit :

Les services de Cloud faiblement couplés sont conçus pour la fourniture et le traitement de contexte. Cela permet à la gestion du contexte d'évoluer à mesure que de nouvelles sources d'information deviennent disponibles ou que des sources existantes disparaissent. Le cadre est ouvert et extensible pour intégrer de nouveaux services de gestion de contexte, tout en restant évolutif et élastique pour gérer un nombre croissant d'utilisateurs. Les composants sont déployés au moment de l'exécution entre les paradigmes mobiles et Cloud. Les composants avec des lignes en pointillés sur la figure 1 sont les composants à déploiement dynamique.

Les modules d'adaptation sont les principaux composants permettant de décider de manière dynamique de la prise en charge de la reconfiguration et du déploiement de l'exécution.

Dans l'infrastructure de Cloud Computing, il existe déjà un mécanisme aux services Web de fourniture et sur la création de volée de nouveaux services qui ne perturbent pas le tissu existant. Cependant, un gestionnaire de service est utilisé pour faire correspondre le contexte pour l'adaptation de services de Cloud contextuels.

Un mécanisme permettant de vérifier la qualité des décisions d'adaptation est mis en place à l'aide de QoC Processing Engine aux deux extrémités. Ce mécanisme fonctionne avec les modules d'adaptation qui prennent en compte la QoC pour les différents contextes lors de l'aide à la décision.

Une approche d'analyse des compromis d'exécution est utilisée dans les modules d'adaptation pour permettre à la décision de répartition de déterminer de manière dynamique le choix du moment et du moment du déploiement dans le Cloud sans affecter la qualité de service.

Ce travail ne donne pas suffisamment de performances pour les situations en temps réel, sur des machines moins puissantes et avec des profils d'utilisateurs hétérogènes en augmentation.

4.3. CAMSPF

Pan et al [167] ont proposé un framework basé sur le Cloud, couramment utilisé dans les environnements informatiques omniprésents pour déployer et adapter des services mobiles. Il permet une approche d'adaptation plus efficace pour chaque utilisateur mobile du Cloud en exploitant la gestion adaptative de la puissance du service basé sur un agent mobile. Comme le

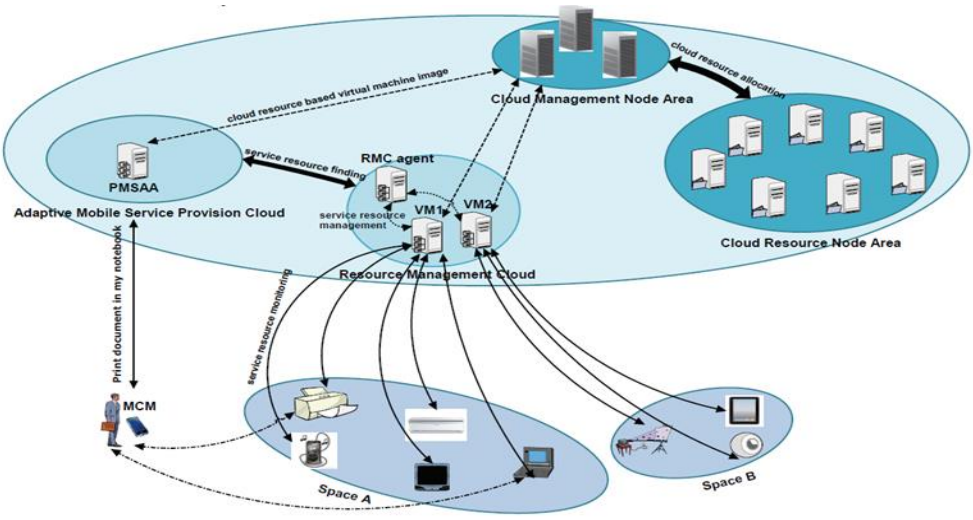

Figure 3.3 La carte conceptuelle du framework CAMSP[167]

montre la figure 3.3, RMC (*Resource Management Cloud*) est le Cloud dédié à la construction d'une

plate-forme unifiée de gestion des ressources destinée à fournir un support d'infrastructure pour la fourniture de services mobiles. Il est responsable de la gestion des ressources enregistrées pour chaque espace et de la fourniture d'une variété d'API permettant de rechercher et d'accéder à des ressources spécifiques. Grâce à RMC, diverses ressources disponibles dans l'environnement sont résumées en objets virtuels dans le Cloud.

Spécialisés pour chaque utilisateur mobile, ils ont conçu un agent d'adaptation de service mobile personnalisé (PMSAA) dans AMSPC. Le PMSAA peut résoudre automatiquement la sémantique du service et mapper de manière dynamique les composants du service sur les ressources disponibles en fonction des préférences et du contexte des utilisateurs, puis réaliser l'adaptation d'exécution du service mobile.

MSM (Mobile Service Middleware) est, comme son nom l'indique, le middleware utilisé dans le contexte des terminaux mobiles. Il est placé entre la couche d'application mobile et la couche de système d'exploitation. Il obtient le support matériel sous-jacent via le système d'exploitation mobile et peut détecter les demandes de service mobile de la couche d'application mobile. Une fois que le MSM a détecté la demande de service, il envoie la demande de service avec les informations de contexte requises à PMSAA pour obtenir la prise en charge de l'adaptation de service du Cloud.

Cependant, ce Framework ne prend pas en compte les informations contextuelles pouvant être collectées par les capteurs intelligents.

4.4. CoCaMAAL

Forkan et al [168] ont proposé un middleware sensible au contexte orienté Cloud, basé sur une architecture orientée service et le Web sémantique pour la gestion d'espaces de contexte volumineux pour des systèmes de vie assistée hétérogènes. Dans ce travail, les données contextuelles volumineuses des systèmes sensibles au contexte sont traitées dans le Cloud et non sur des ordinateurs serveurs locaux. En cas de problème, tel que la mobilité de l'utilisateur, la disponibilité des informations de contexte, l'instabilité du réseau de communication et les données de contexte, ne pouvant pas être transmis au Cloud, ce qui entraîne une interruption du système. Notre approche doit réagir rapidement à de nouvelles situations parmi celles précédemment observées et regroupées dans un cluster en analysant les emplacements, les profils et les environnements physiques actuels des utilisateurs. L'architecture abstraite du système

CoCaMAAL est illustrée à la Figure. 3.4. Une description détaillée de chacun des composants est donnée ci-dessous.

Systèmes AAL: Notre modèle proposé peut desservir un grand nombre de clients AAL. Les clients

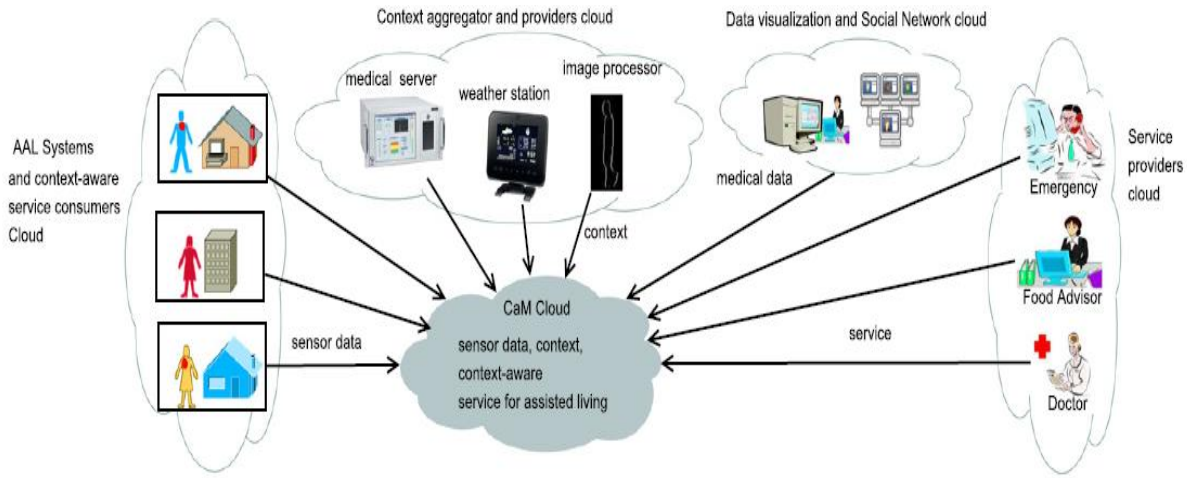

Figure 3.2 L'architecture générique du modèle CoCaMAAL [168]

AAL agissent en tant que fournisseurs de données de capteurs et consommateurs de services sensibles au contexte. Tous les systèmes AAL forment ensemble une structure de Cloud distribuée. Ils génèrent des données de capteur brutes pour le modèle, qui deviennent l'entrée pour la génération de contexte de haut niveau et consomment des services sensibles au contexte.

Agrégateur de contexte et fournisseurs (CAP) : le Cloud CAP contient la logique de calcul et le processus de conversion des données brutes de bas niveau en une représentation de contexte abstraite reconnaissable par tous les composants de l'architecture.

Fournisseurs de services : les fournisseurs de services (SP) sont des applications et des services liés à la connaissance du contexte. Un SP peut être une application logicielle s'exécutant sur le périphérique mobile d'un système AAL qui rappelle à l'utilisateur les rendez-vous, ou bien un fournisseur de soins externe qui surveille les situations d'urgence.

Middleware sensible au contexte (CaM): le nuage CaM est le composant fonctionnel le plus important du modèle. Le nuage CaM dispose d'une infrastructure pour le traitement des données de contexte, le stockage et la récupération de contexte, la gestion de services sensibles au contexte, les mécanismes de contrôle d'accès des enregistrements médicaux, la cartographie contexte-

service, la fourniture d'actions d'assistance et de nombreuses autres tâches informatiques complexes.

Visualisation des données de contexte : Les données de contexte contiennent les informations médicales précieuses de l'utilisateur. Une interface visuelle appropriée est nécessaire pour que le professionnel de la santé puisse visualiser les informations. Certains services de visualisation de données fournissent des interfaces utilisateur utiles à cet effet.

Ce travail n'atténue pas l'hétérogénéité sémantique dans un modèle générique permettant une adaptation dérivée plus automatisée à partir d'un grand nombre de profils d'utilisateurs hétérogènes et la mobilité des utilisateurs et l'instabilité du réseau de communication ne peuvent pas être effectuée automatiquement.

4.5. CARMiCLOC

Aguilar et al [169] ont proposé un middleware de sensibilisation au contexte dans le Cloud Computing appelé CARMiCLOC, un middleware basé sur un service Web pouvant se comporter comme un logiciel en tant que service (SaaS). Les bases philosophiques de l'architecture CARMiCLOC sont l'informatique réflexive, l'informatique autonome et l'informatique en nuage.

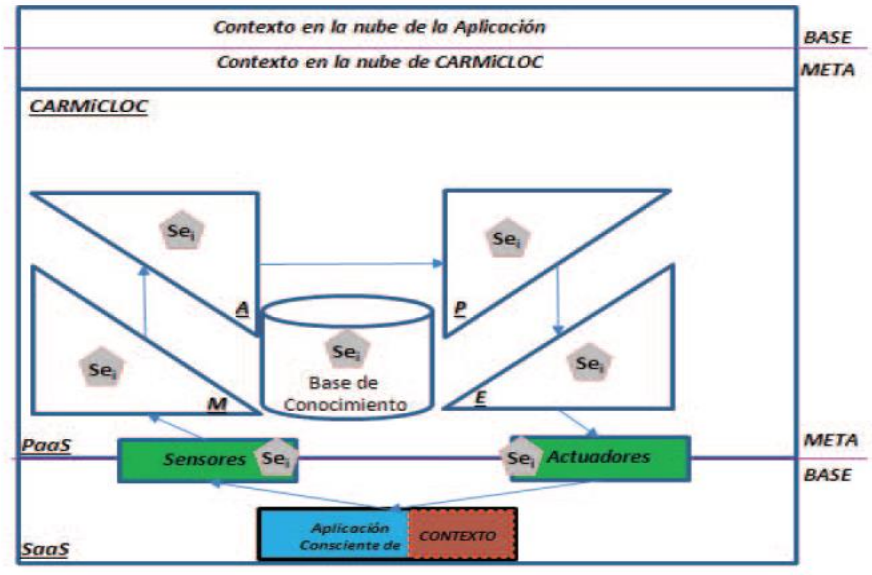

Figure 3.3 Architecture de base de CARMICLOC [169]

La réflexion est la capacité de notre middleware à surveiller et à modifier son propre comportement, ainsi que les aspects de son implémentation (syntaxe, sémantique, etc.), ce qui lui

permet d'être sensible à son contexte. En ce sens, CARMiCLOC a un comportement dynamique et une architecture adaptative, c'est-à-dire qu'il peut gérer des services capables de changer ou d'évoluer de manière dynamique. En général, cela a deux processus :

Introspection : La capacité d'observer et de raisonner sur votre propre état d'exécution.

L'intersection : C'est la capacité de modifier son propre état d'exécution, ou de modifier son propre sens ou interprétation.

En raison de la quantité de données impliquées pour définir le contexte, ce travail offre une excellente gestion de contexte utilisant un Cloud Computing, mais il n'est pas en mesure d'analyser un grand nombre d'utilisateurs simultanés pour identifier des services avec des profils génériques similaires.

4.6. E-HAMC

Aazam et Huh [170] ont proposé un service basé sur un smartphone, Emergency Help Alert Mobile Cloud (E-HAMC), qui utilise les services de brouillard à des fins de prétraitement et de déchargement pour fournir un moyen instantané de notifier le service d'urgence concerné (par exemple, une ambulance) à partir des détails stockés du contact. Ce service envoie également le lieu de l'incident afin de faciliter la recherche des patients.

L'application est capable d'envoyer automatiquement un message aux membres de la famille proche déjà stockés, dont la liste est maintenue par l'application. Cela présente un autre avantage: si la victime n'est pas en mesure d'informer les membres de sa famille, tout témoin ou tout passant peut le faire en appuyant simplement sur un bouton en utilisant E-HAMC sur son propre téléphone. L'application fournit également le mode témoin pour l'utilisateur. La seule exception dans ce cas serait que les membres de la famille ne soient pas informés. Le mode par défaut est le mode victime. Une fois l'alerte déclenchée par le brouillard, les données sont prétraitées pour être affinées, puis envoyées au nuage. Les autorités concernées peuvent collecter les données du nuage, le cas échéant, pour analyser le type de situations d'urgence qui se sont multipliées, à quelle fréquence, dans telle ou telle zone et quelles en sont les raisons. Cela permettra de prévenir et d'éviter de telles situations à l'avenir et d'assurer une meilleure vie publique. Le modèle de communication de E-HAMC proposé avec le Cloud Computing est indiqué sur la figure 3.6.

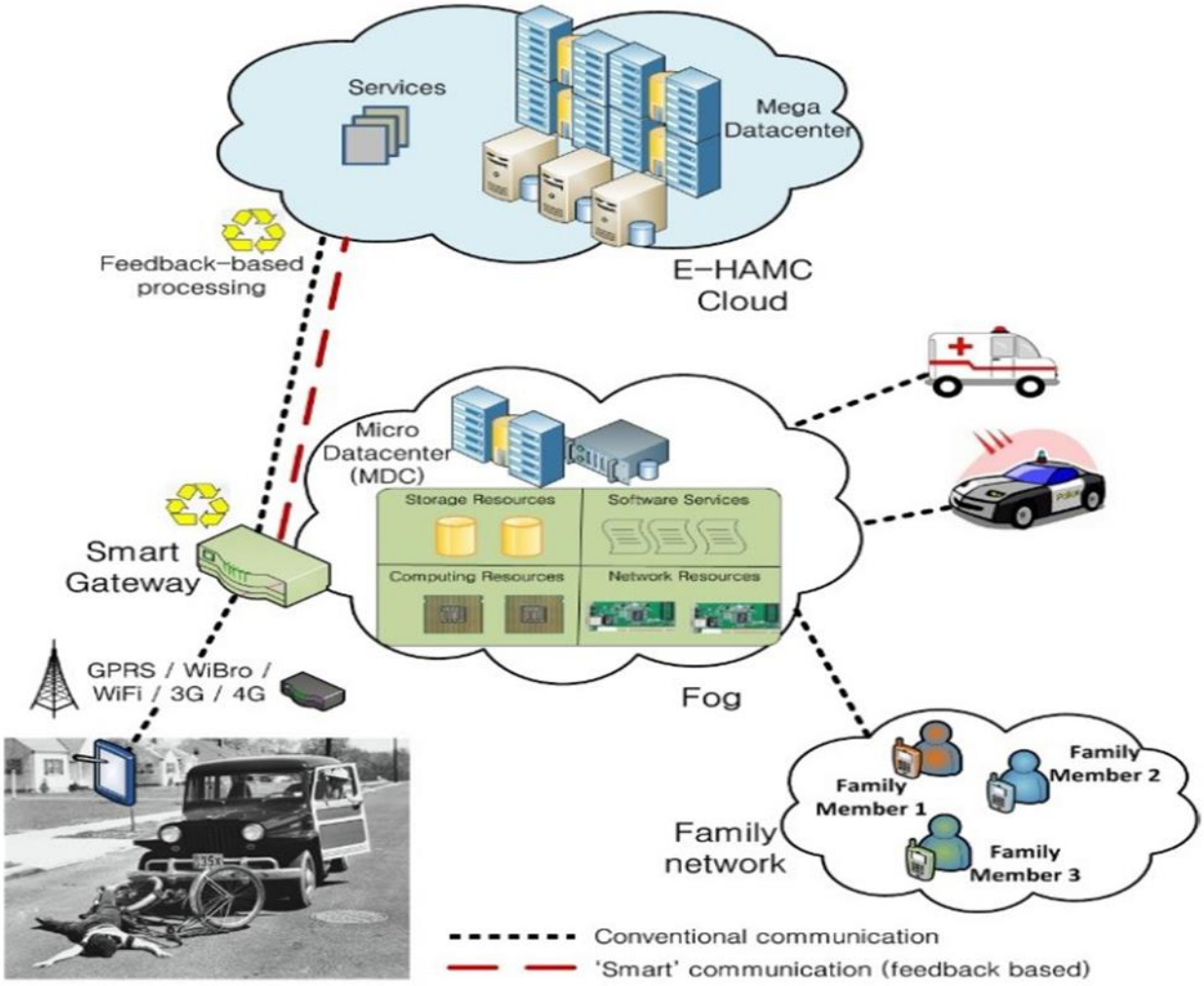

Figure 3.4 Modèle de communication E-HAMC [170]

Traitement des notifications d'urgence : Dans son système, une fois la situation d'urgence créée, la victime envoie une photo de l'événement, qui est ensuite envoyée automatiquement au Cloud Computing via l'application. Étant donné que le service sera utilisé avec les téléphones intelligents, le fait d'avoir une caméra dans l'équipement n'est pas un problème.

Mise à jour des contacts en fonction de l'emplacement de l'utilisateur : Lorsqu'un utilisateur a changé de lieu et a déménagé dans une autre ville ou un autre pays, les contacts des services de traitement des urgences doivent être mis à jour. Dans le nouvel emplacement, l'application contacte le Cloud et synchronise les listes des contacts, ainsi que la disponibilité de différents types de départements, traitant différents types de catastrophes. Par cela, l'utilisateur n'a jamais à mettre à jour manuellement. Les utilisateurs sont toujours prêts à utiliser l'application.

Éviter l'usurpation d'identité de localisation : Dans les méthodes traditionnelles disponibles de gestion des situations d'urgence, l'usurpation d'emplacement se produit beaucoup et devient une gêne pour les départements traitant d'urgence. En raison de ce type de problème, le réseau reste occupé et les victimes ne parviennent parfois pas au service de secours. Dans notre système, nous avons géré le problème en rendant automatique la détection de la position. Lorsqu'un utilisateur / victime envoie une notification d'urgence, la position est automatiquement prise à partir du GPS ou du BTS de l'appareil mobile de l'utilisateur auquel il est connecté. Ce mécanisme rend impossible l'usurpation de l'emplacement.

5. Discussion

Comme décrit ci-dessus, le système tenant compte du contexte en ce qui concerne l'architecture est divisée en trois caractéristiques critiques : modèle du contexte, scalabilité et possibilité de réutilisation. Ces systèmes représentatifs sensibles au contexte ont choisi respectivement différents types de modèles de communication et de composants logiciels de base. Cependant, presque tous supportent les caractéristiques de la généralité, d'extensibilité et de réutilisabilité du contexte. L'environnement intelligent évoluant plus rapidement qu'auparavant, il est encore plus important de prendre en charge l'extensibilité et la réutilisabilité. Une autre tendance du système sensible au contexte est que, sans exception, tous les systèmes récents séparent les ressources de contexte du traitement de contexte, ce qui contribue au développement rapide d'applications sensibles au contexte, contrairement aux systèmes antérieurs sensibles au contexte. Sur la base d'analyses et de comparaisons antérieures, nous avons résumé les principales caractéristiques des systèmes sensibles au contexte, devenus des critères importants.

Le système sensible au contexte doit prendre en charge autant de types de contextes que possible pour s'adapter à différentes situations. Cependant, il n'est pas possible de fournir tous les types de contexte. Le modèle basé sur une ontologie fournit des solutions efficaces, prenant en charge à la fois la réutilisation et le partage des connaissances, ainsi que l'inférence de contexte, de sorte que les systèmes de plus en plus sensibles au contexte adoptent ce modèle pour organiser les informations contextuelles.

En outre, les chercheurs commencent à attirer l'attention sur le contexte historique. Ces données de contexte peuvent être utilisées pour prédire le comportement des utilisateurs. Apparemment, ce sera l'une des nouvelles orientations à venir.

Chaque cadre présenté ci-dessus calcule toutes les situations spécifiques pour un profil d'utilisateur particulier et explore individuellement chaque description de service et calcule la distance correspondante afin de sélectionner les services pertinents. Dans de nombreux cas, cette situation peut surcharger le système. De nombreux profils spécifiques auraient pu être spécifiés, de nombreuses comparaisons doivent donc être calculées et mises à jour. Notre objectif dans ce travail est atteint en regroupant des contextes et des services spécifiques dans le profil générique et en manipulant des préférences et des services sémantiquement équivalents (catégorie, emplacement, heure, qualité de service).

Le Tableau 4-1 illustre une étude comparative entre les différentes approches de description des profils utilisateurs. La plupart des travaux et middlewares existants améliorent la connaissance du contexte sur la base de plates-formes d'adaptation dynamiques basées sur des composants, telles que MUSIC [171], Kalimucho-A [172] et Kali-Smart [173], CoCaMAAL [168], CARMiCLOC (Aguilar et al., 2011), Le tableau 2 présente un résumé de ces travaux. Comme nous l'avons montré dans cette section, de nombreux travaux existants fournissent des composants spécifiques adaptés à chaque profil spécifique. Ce type d'adaptation présente de nombreux inconvénients :

- Aucun d'entre eux n'utilisent pas un profil générique, à savoir qu'il est impossible d'automatiser l'adaptation provenant d'un grand nombre de profils d'utilisateurs hétérogènes. Par exemple, si un nouveau profil est intégré, nous devons fournir des composants spécifiques adaptés à ce profil spécifique. Par conséquent, si le contexte utilisateur évolue rapidement (par exemple, la mobilité de l'utilisateur) le système peut être surchargé par un profil spécifique.
- Aucun d'entre eux ne permet une gestion intelligente des situations urgentes des utilisateurs, c'est-à-dire qu'il n'est pas possible de calculer toutes les situations spécifiques d'un profil utilisateur particulier, d'explorer chaque description de service individuellement et de calculer la distance de mise en correspondance afin de sélectionner les services pertinents. Les appareils changent de manière dynamique en fonction de différents contextes (profil utilisateur, environnement utilisateur, surveillance, activités

sociales, etc.). Il suffit de marquer les utilisateurs qui ont changé de manière dynamique leur contexte au sein de leur communauté afin de recalculer localement leurs situations en tenant compte de la mobilité d'un utilisateur et de leurs utilisations du contexte ou de l'instabilité du réseau de communication.

- Aucun des profils n'expriment des contraintes riches, comme par exemple « si je participe à une réunion, je ne peux pas jouer de contenus audio ». C'est le premier objectif de ce travail qui est de permettre à l'utilisateur de créer ses propres contraintes.

	Modèle de contexte	**Information de contexte**	**Mobilité des utilisateurs**	**Scalabilité du contexte**	**Généralité du contexte**	**Raisonneur de situation**	**Stockage de contexte**	**regroupement de contexte**
SOCAM	Basé ontologies	Location Activité Utilisateur Dispositif	Faible	Faible	Faible	Centralisé	Serveur	Aucun
Federated-Q	Basé ontologies	Location Utilisateur Dispositif	Faible	élevé	Faible	Distribué	Cloud	Aucun
CAMSPF	Basé ontologies	Dispositif Services réseau	Faible	élevé	Aucun	Centralisé	Cloud	Aucun
CoCaMAAL	Basé ontologies	Location Activité Utilisateur Dispositif Environnement	Faible	élevé	Faible	Centralisé	Cloud	Aucun
CARMiCLOC	Basé ontologies	Dispositif Utilisateur	Faible	élevé	Aucun	Centralisé	Cloud	Aucun
E-HAMC	Basé ontologies	Location Utilisateur Dispositif Environnement	Faible	élevé	Aucun	Centralisé	Cloud	Aucun

Table 2-1 Comparaison des architectures sensibles au contexte

- Ils n'exploitent pas de contraintes sémantiques riches liées aux paramètres du contexte de l'utilisateur alors que ce sont des informations importantes pour gérer de façon intelligente un système d'adaptation.

6. Conclusion

Dans ce chapitre, nous avons fait un tour d'horizon des différents profils sémantiques et architectures sensibles au contexte pour les environnements intelligents. Après avoir conclu sur le caractère général des systèmes sensibles au contexte, de ces derniers, les plus représentatifs seront choisis pour être analysés. Nous fournissons ensuite une analyse et une comparaison plus détaillées des aspects fondamentaux des systèmes sensibles au contexte : architecture, représentations, détection, stockage et raisonnement du contexte. La plupart des solutions proposées, les fonctionnalités du middleware sont obtenues en répartissant les tâches dans une architecture en couches. Plusieurs applications exploitent les ressources du Cloud pour concevoir des systèmes efficaces. Certaines des recherches ont principalement suggéré comment l'infrastructure de Cloud Computing peut être utilisée pour le traitement de l'information de contexte. La modélisation et la conception d'un système intelligent basé sur l'ontologie, guidé par les contraintes utilisateurs fait l'objet du chapitre suivant.

Chapitre 4. GUSP-Onto : une Ontologie de Profil Unifie Sémantique pour les environnements intelligents

1 Introduction

Les défis liés au contexte sont la surveillance, l'agrégation et l'analyse des informations de contexte de manière sémantique et la sélection de services sensibles à la situation pour accéder / diffuser des services à l'aide de plates-formes middleware. Notre travail consiste à gérer efficacement les situations des utilisateurs grâce à la modélisation du profil sémantique sensible au contexte, à une stratégie d'identification adaptée à la situation et à la fourniture de tous les services d'adaptation distribués facilitant le partage de contenus multimédias à l'aide du middleware Kalimucho basé sur le Cloud Computing [172] dans un environnement variable et dynamique. Dans ce chapitre, nous proposons un profil sémantique unifié, utilisé pour organiser le contexte dans différentes situations. Après la représentation de ce modèle, Nous présentons les outils techniques associés sont également décrits. Enfin, ce chapitre se termine par une conclusion.

2. Ontologie de Profil générique sémantique sensible aux situations

Le but principal de notre ontologie est de bien comprendre les utilisateurs afin d'améliorer l'identification de leur situation. De plus, cette ontologie accélère l'identification de la situation en utilisant des informations de contexte pertinentes. GUSP-Onto est une ontologie générique pour les environnements intelligents. Elle inclut des définitions des concepts de base interprétables par la machine dans l'environnement intelligent et des relations entre eux. Construire une ontologie générique en utilisant un contexte formel basé sur une ontologie peut jouer un rôle vital en facilitant le raisonnement en représentant formellement les connaissances du domaine intelligent et en facilitant la sélection du service approprié parmi un grand nombre de services en fonction des situations inférées. Les membres de la communauté de tel domaine peuvent communiquer et partager des connaissances entre eux en utilisant GUSP-Onto. Pour développer notre ontologie, nous avons utilisé une combinaison de l'approche descendante. Premièrement, nous avons défini les concepts importants dans le domaine de l'environnement intelligent à partir d'ontologies existantes : entité de contexte, situation et service. Deuxièmement, nous avons généralisé et spécialisé. Pour chaque concept, nous avons créé une taxonomie de concepts afin de construire la hiérarchie de GUSP-Onto. Par exemple, la situation peut être urgente et normale. Les situations urgentes font référence à des situations anormales : situation d'alarme, intrusion et incendie, les autres étant des situations normales, au vu des activités quotidiennes de l'utilisateur et de son appareil. Troisièmement, nous avons ajouté pour chaque concept ses propriétés. Enfin, nous avons

créé une relation logique entre ces concepts pour permettre le raisonnement à l'aide de notre ontologie. À la fin du processus de création de l'ontologie, nous avons obtenu notre GUSP-Onto qui contient plusieurs concepts spécifiques tels que le contexte utilisateur, le contexte d'objet dynamique, le contexte d'environnement, la situation urgente et le service.

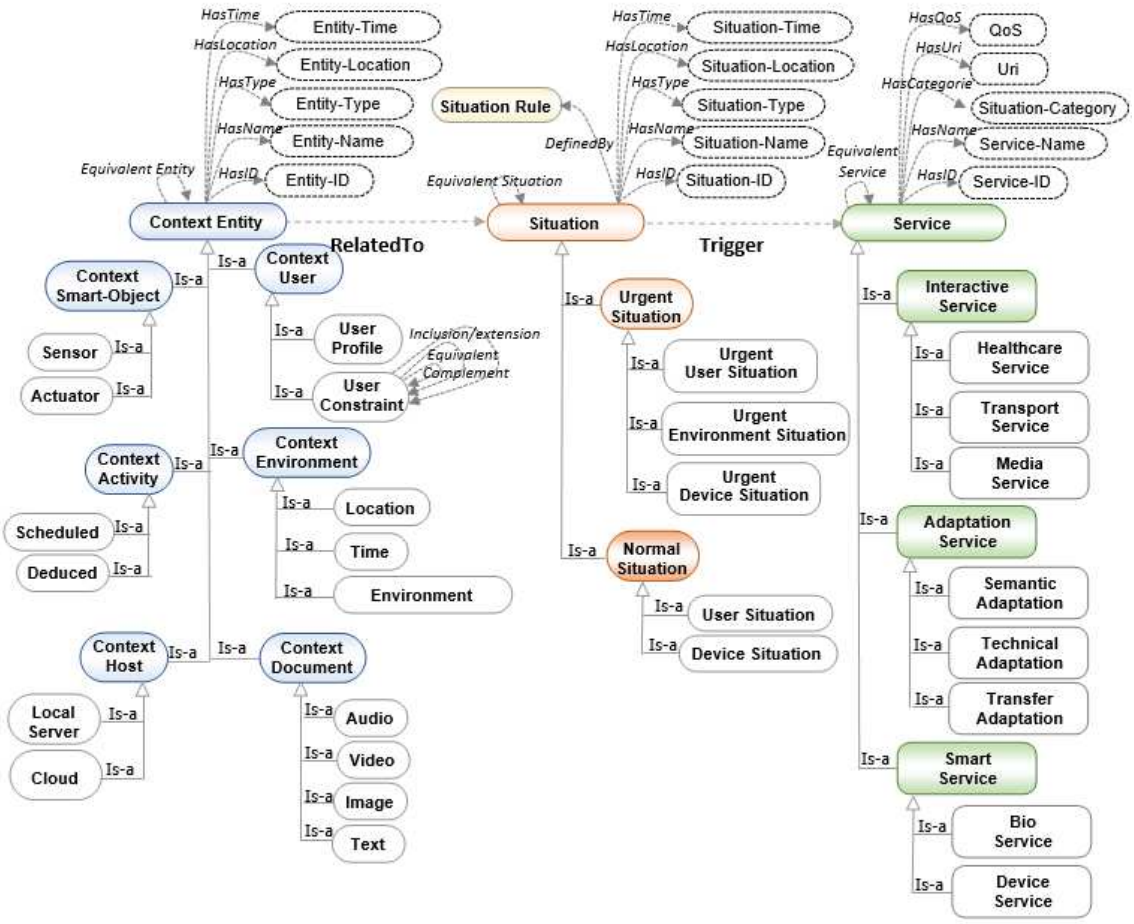

Figure 4.1 Ontologie de Profil générique sensible à la situation

2.1. Modélisation sémantique de contexte

Ce concept a de nombreuses propriétés pour le caractériser, telles que ID-entité, Nom-entité, Type-entité, Emplacement-entité et autres. Les concepts d'entité de contexte ont un certain nombre de sous-concepts : Contexte d'utilisateur qui inclut des propriétés telles que nom, âge, poids, taille, sexe et autres. Le contexte d'objet intelligent comprend plusieurs capteurs et actionneurs hétérogènes, chacun ayant un identificateur unique, un nom local, un emplacement et d'autres

attributs décrivant ses propriétés. Le contexte de document multimédia et le contexte d'environnement ont des paramètres d'environnement tels que la température, l'humidité, etc. qui font référence à un utilisateur dans un environnement intelligent peut effectuer plusieurs activités et contextes d'hôte.

Nous avons divisé la sous-ontologie ***Context*** en huit sous-contextes :

- Classe «***Profil Utilisateur***» : décrit les informations sur l'utilisateur (*Nom, localisation, âge, état de santé, préférences, profession, langage...etc*), qui peut faire varier le service d'adaptation. Les préférences de l'utilisateur comprennent les langues et les modalités préférées (*voix, geste, clic de stylo, clic de souris, etc.*). Un utilisateur peut sélectionner le multimédia préféré (*image, texte, vidéo ou audio*) afin de l'adapter à son contexte. Par exemple, s'il reçoit de l'audio quand il est au travail, il préfère la réception d'un texte à la place de ce dernier ; dans ce cas nous avons besoin d'un service d'adaptation de changement de l'audio en un texte. Nous pouvons trouver aussi une description de la situation de la santé de l'utilisateur : utilisateur en bonne santé ou handicapé (*cf. Figure 4.2*).

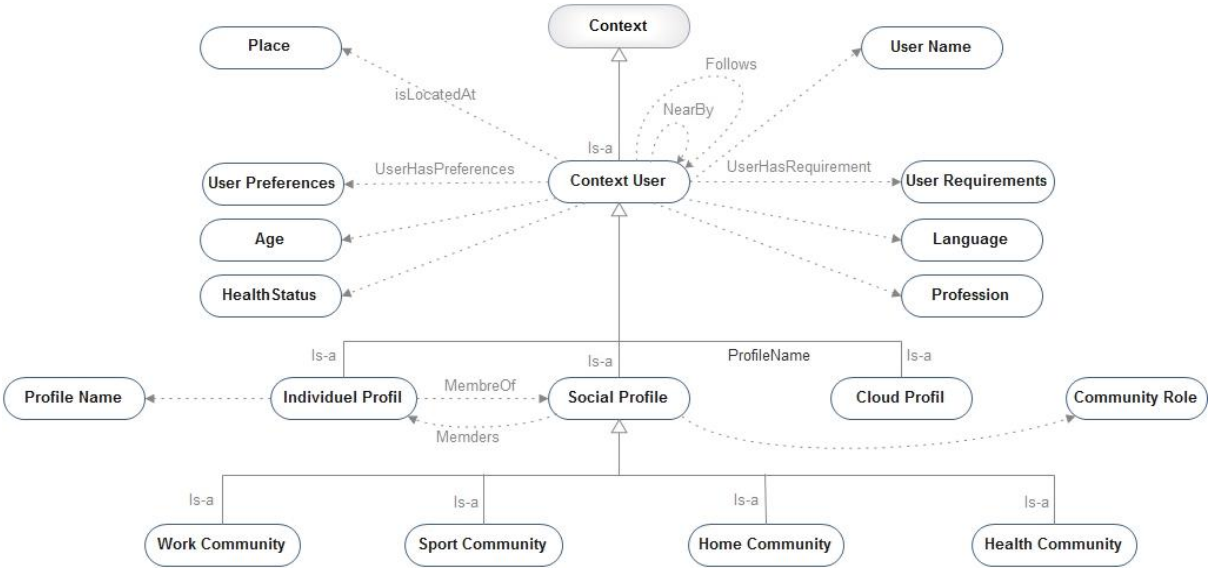

Figure 4.2 Ontologie de contexte utilisateur

- Classe « ***Smart Object***» : décrit les données qui sont collectées à partir des différents capteurs ainsi que les chaînes actions des différents actionneurs dans

des environnements intelligents. Nous distinguons trois types de capteurs (*cf. Figure 4.3*) :

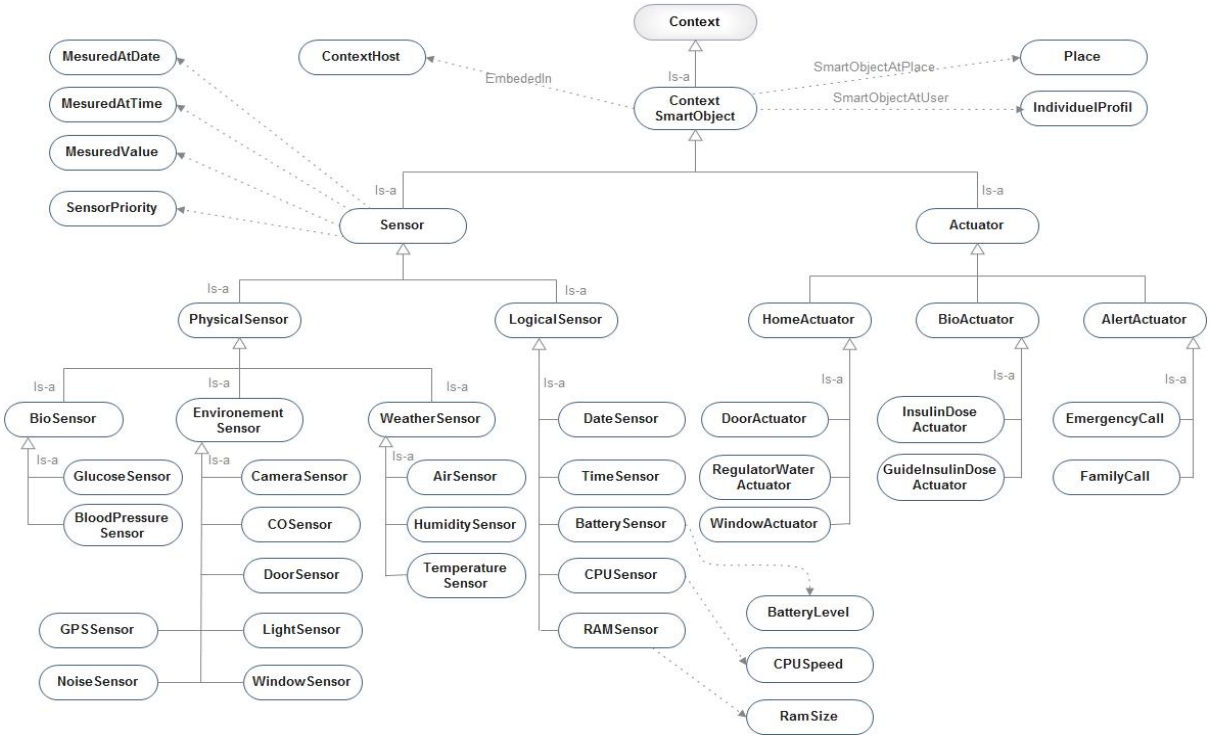

Figure 4.3 Ontologie des capteurs et des actuateurs intelligents.

- ***BioSensor*** : représente les données capturées par le *BioSensor*, comme la pression artérielle, la glycémie et la température du corps,
- ***EnvironmentSensor*** : ce sont les données fournis par les capteurs d'environnement, comme la température de la maison, l'humidité, etc,
- ***LogicalSensor*** : représente des données qui sont capturées par le dispositif, comme la vitesse du processeur, l'énergie de la batterie, etc.

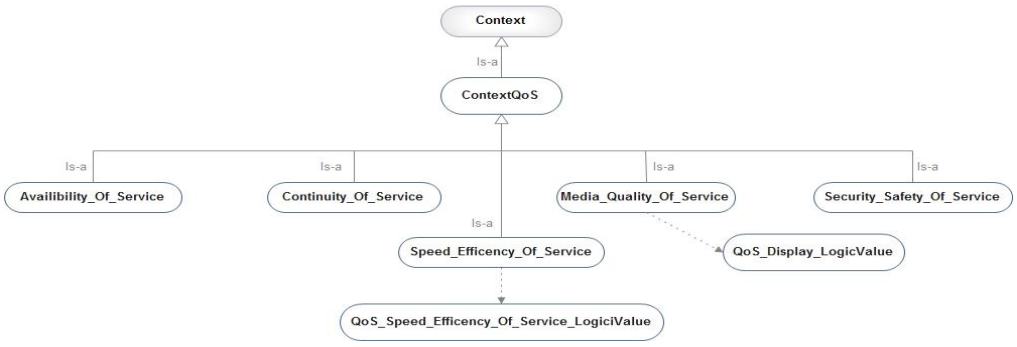

Figure 4.4 Ontologie de qualité de service.

- Classe « ***ContextQoS***» : permet de décrire la qualité d'une application mobile, représentée sur notre ontologie, elle est définie comme un ensemble de paramètres de métadonnées. Ces paramètres de qualité de service sont les suivantes : 1) - la continuité du service, 2) - la durabilité du service, 3) - la vitesse et l'efficacité du service,4) - la sécurité. 5) - la disponibilité du service (*cf.Figure 4.4*).

- Classe « ***Context Interaction***» : définit les différents types de modalités (*clic, geste, touche, voix*), en respectant les préférences définies par l'utilisateur ainsi que les modalités supportés par les dispositifs situés près de l'utilisateur (*cf. Figure 4.5*).

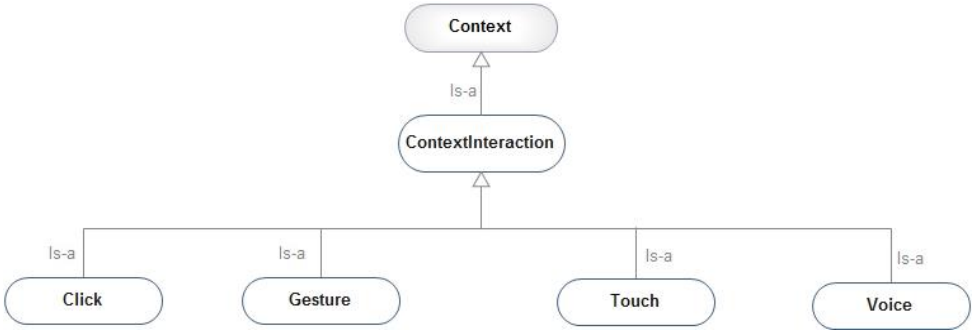

Figure 4.5 Ontologie des interactions utilisateur.

- Classe « ***Context Host***» : est divisé en deux sous-classes principales ***Cloud*** et ***Device***. Les services peuvent être hébergés sur les dispositifs locaux ou sur le Cloud. La classe ***Device*** contient des informations sur les dispositifs fixes ***FixedDevice*** ou sur des dispositifs mobiles ***MobileDevice***. Les dispositifs mobiles ont des ressources limitées telles que la batterie, la mémoire, CPU …etc. La classe Cloud contient des informations sur le serveur de Cloud (*par exemple Google Cloud*) qui peut être utilisé pour les services d'hébergement. Le service est déployé et migré sur les dispositifs en respectant les contraintes et les exigences de déploiements du service (*supportedPlateform, HasQoS, HasModality, HasMedia,* …) de sorte qu'il pourrait se produire une substitution d'un service équivalent (*un scénario est possible, lorsque le niveau de la batterie est faible, le service Vidéo Acquisition doit être substitué par un autre service équivalent comme par exemple*

le service Image Acquisition afin de minimiser la consommation de l'énergie) (*cf. Figure 4.6*).

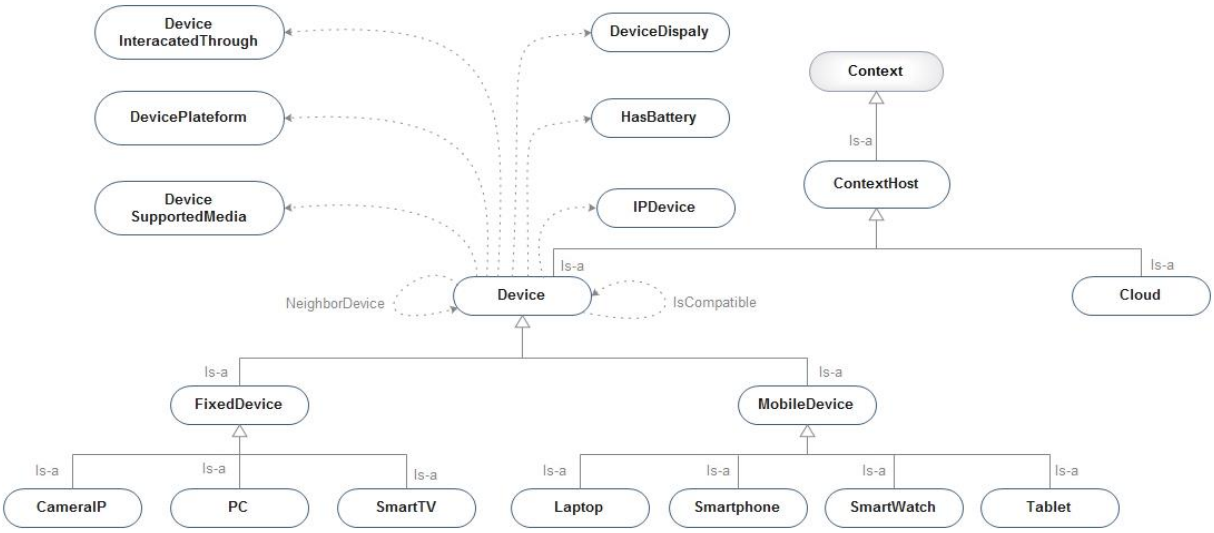

Figure 4.6 Ontologie des dispositifs mobiles et fixes.

- Classe « ***Context Environnement***» : décrit les informations spatio-temporelles (longitude, latitude, temps, date). Malgré l'évolution dynamique des infrastructures logicielles (apparitions, disparitions des objets et dispositifs), on peut définir pour chaque emplacement et à tout moment donné, l'ensemble des ressources disponibles (***Device, Smart-Object***).

  Ces évolutions peuvent être dues à la mobilité de ces objets et dispositifs, mais aussi à des mesures d'économie d'énergie ou encore à des pannes (*cf.Figure 4.7*).

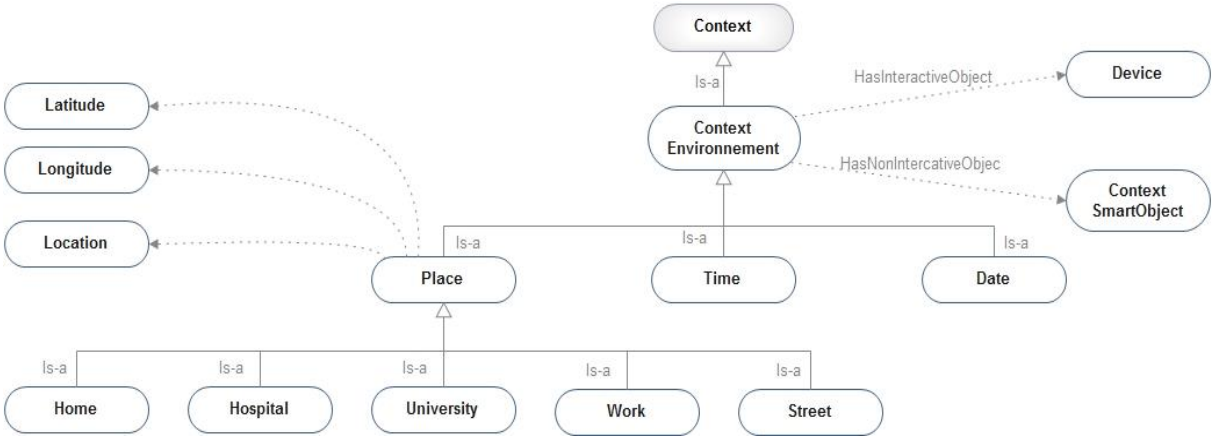

Figure 4.7 Ontologie de contexte de l'environnement.

- Classe « ***Context Document***» : décrit la nature des documents (texte, vidéo, audio). Le contexte de document spécifie un ensemble de propriétés liées à un type de support spécifique : (1) Texte : alignement, police, couleur, format, …etc. (2) Image : hauteur, largeur, résolution, taille, format, etc. (3) Vidéo : le titre, la couleur, la résolution, la taille, la codification etc. (4) Son : fréquence, taille, résolution et format (*Figure 4.8*).

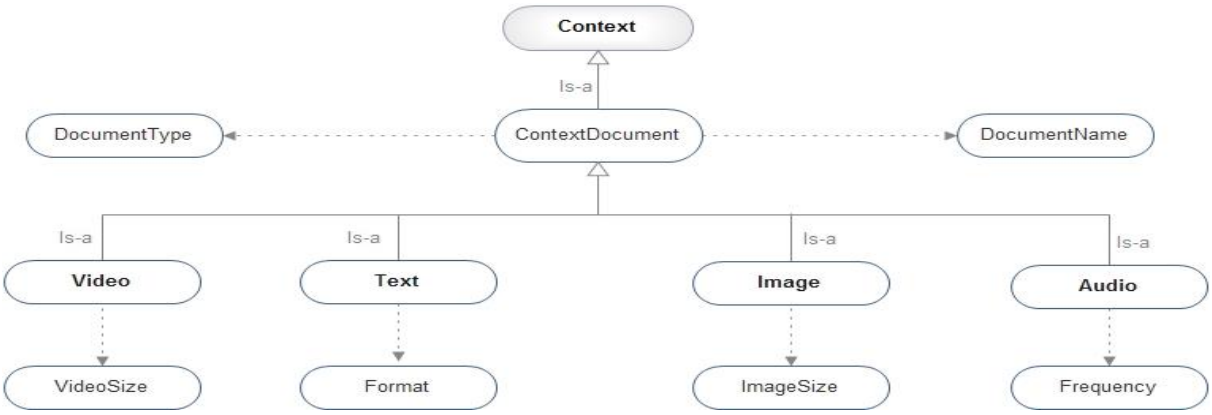

Figure 4.8 Ontologie de context du document.

- Classe « ***ContextActivity***» : contient deux sous classes : *Task* et *Schedule*. La classe *Task* permet de décrire les activités définies par un utilisateur (*WorkTask, UniversityTask, HealthTask …etc.*) ou par un développeur (*Deploy, Remove, Stop*

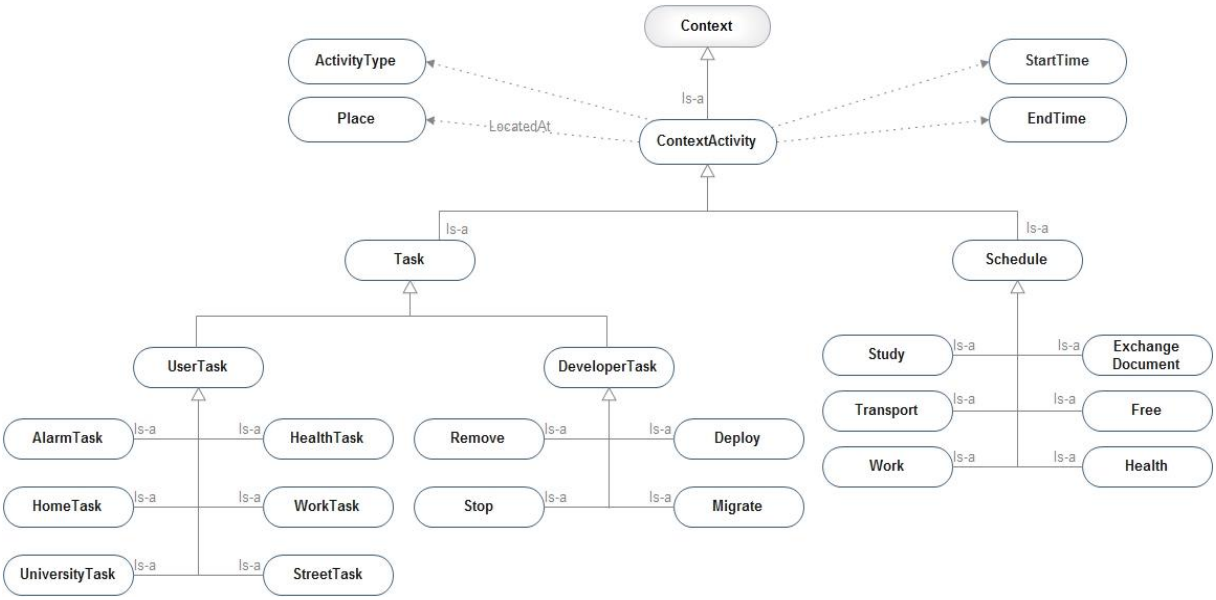

Figure 4.7 Ontologie des Activités.

etMigrate). La classe Schedule contient un calendrier qui permet à un utilisateur d'exercer une activité prévue (*Figure 4.9*).

2.2. Modélisation sémantique des services

Pour répondre aux demandes des utilisateurs à tout moment et n'importe où selon son emplacement, l'interaction entre les utilisateurs cherche à obtenir de meilleurs services des fournisseurs. Le service pourrait être soit intelligent soit interactif. Tout service intelligent peut être utilisé d'une manière locale ou en utilisant le Cloud, ce qui lui permet de gérer le stockage des données, dont les utilisateurs ont besoin pour exécuter leurs applications. Les services interactifs ont des interactions uni-modales et multimodales. Un service peut être exécuté sous divers formes avec différentes qualités de service. Chaque utilisateur attend ses propres QoS lorsqu'il utilise le service. Pour assurer le niveau de qualité du service, le service a ses contraintes d'appareils mobiles spécifiques (taille de la mémoire, vitesse du processeur et durée de vie de la batterie). Notre classe ***Service*** contient trois types de services : service intelligent (***NoInteractiveService***), service interactif (***InteractiveService***) et service d'adaptation (**AdaptationService**) (*Figure 4.10*).

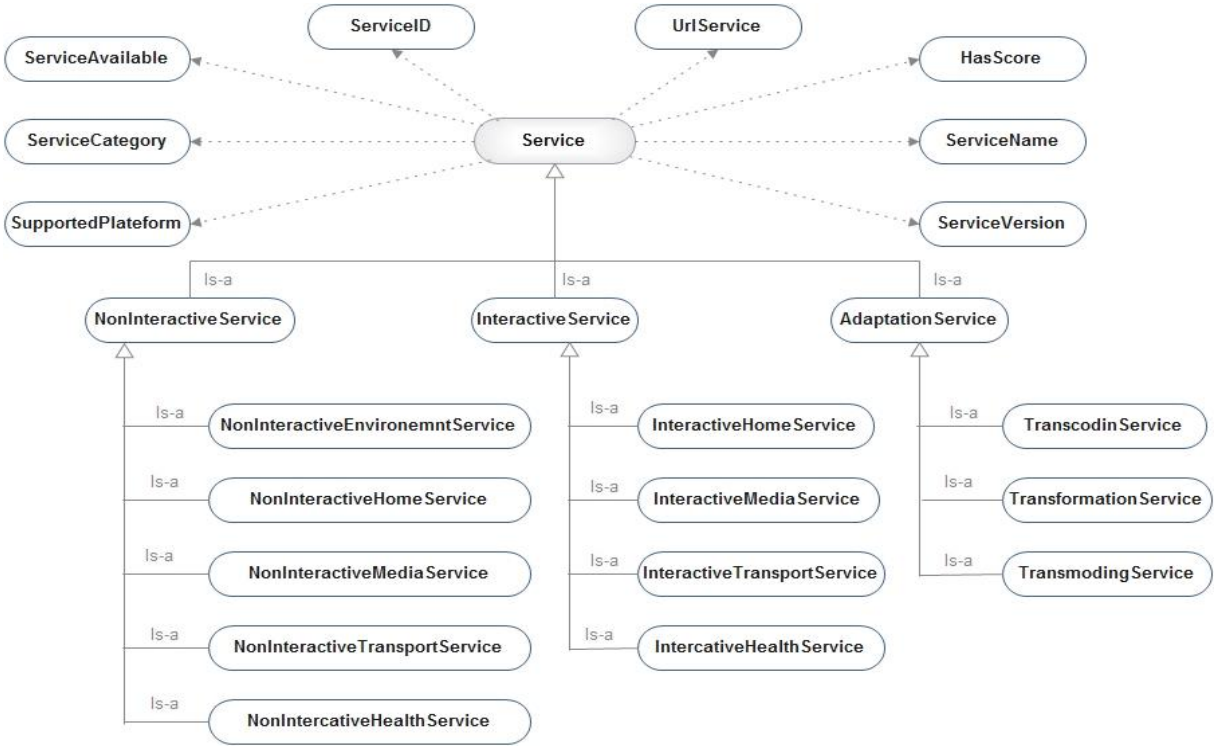

Figure 4.8 Ontologie des services.

2.3. Ontologie des Situations

Chaque utilisateur, appareil et élément dans un environnement intelligent peut contenir plusieurs situations. Ces situations sont divisées en deux catégories : urgente et normale. Chaque situation peut faire partie d'autres situations. Elle a son heure de début et de fin. La situation est effectuée dans un lieu et consiste en deux ou plusieurs conditions contextuelles. La classe Situation urgente représente des situations liées à une entité de contexte spécifique, telle que l'état de santé d'une personne, comme les situations de glycémie et les situations de pression artérielle. La classe de situation normale représente des situations liées aux activités sociales des utilisateurs et de leurs appareils, telles que les situations de température domestique et les situations de batterie. (*Figure 4.11*).

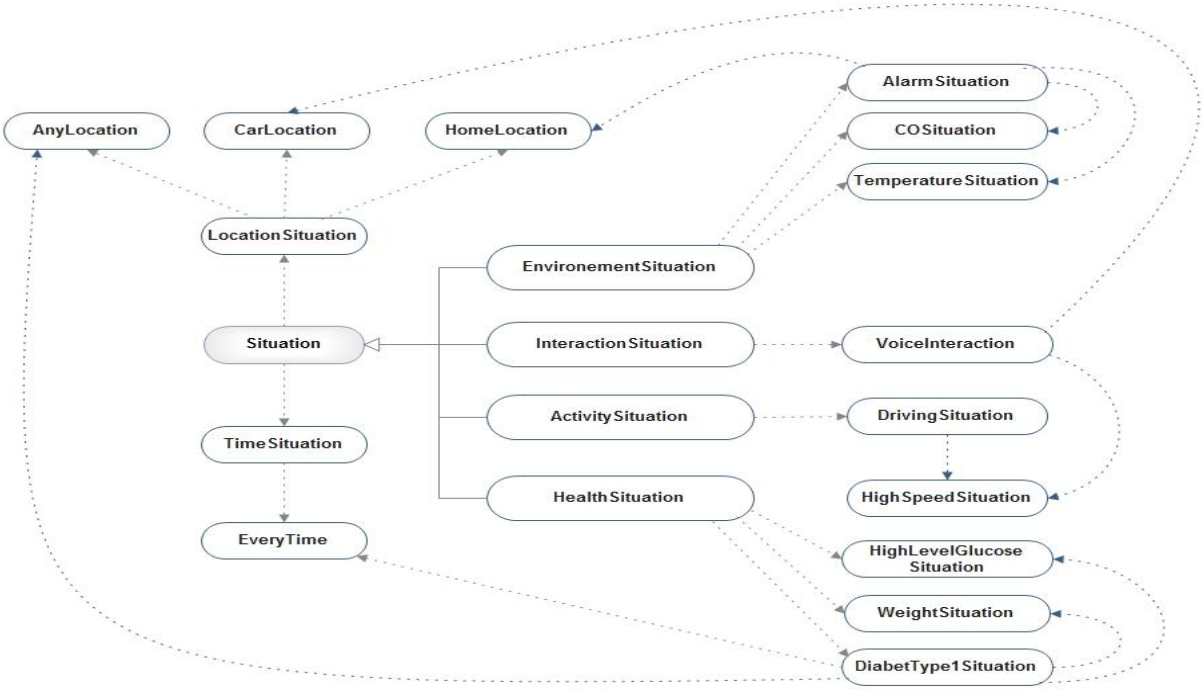

Figure 4.9 Ontologie des situations.

3. Outils de développement de modèles basés sur des ontologies

Nous avons choisi un langage de description d'ontologies, OWL, pour construire le modèle de contexte spatio-temporel. Bien que OWL soit un langage basé sur XML, le développement d'OWL est différent d'un XML classique. Face à un environnement de capteurs hétérogènes, nous utilisons

l'éditeur Protégé et Jena pour développer OWL en Java. Généralement, l'éditeur Ontologie Protégé fournit une plate-forme réalisable prenant en charge WYSIWYG pour la construction du modèle initial. Jena fournit un ensemble d'API permettant aux développeurs de gérer le contexte dans JAVA. Des informations détaillées seront fournies ci-dessous.

3.1. L'éditeur Protégé

L'éditeur Protégé, développé par le Centre de recherche en informatique biomédicale de Stanford, est un éditeur d'ontologies libre open source et un framework basé sur la connaissance [174]. La plate-forme Protégée prend en charge les ontologies de modélisation via un client Web ou un client de bureau, basé sur Java, extensible et fournissant un environnement plug-and-play, ce qui est en fait une base flexible pour le prototypage rapide et le développement d'applications.

La plate-forme Protégée prend en charge deux principales méthodes d'ontologie de modélisation : l'éditeur Protégé-Frames et l'éditeur Protégé-OWL.

L'éditeur Protégé-Frames permet aux utilisateurs de créer et de gérer des ontologies basées sur des cadres, conformément au protocole OKBC (Open Knowledge Base Connectivity). Dans ce modèle, une ontologie consiste en un ensemble de classes organisées dans une hiérarchie de sous-consommation afin de représenter les concepts saillants d'un domaine, un ensemble d'emplacements associés à des classes pour décrire leurs propriétés et relations, et un ensemble d'instances de ces classes - des exemples individuels du concepts qui contiennent des valeurs spécifiques pour leurs propriétés.

L'éditeur Protégé-OWL permet aux utilisateurs de créer des ontologies pour le Web sémantique. Il implémente un riche ensemble de structures et d'actions de modélisation des connaissances qui prennent en charge la création, la visualisation et la gestion d'ontologies dans divers formats de représentation, notamment OWL, RDF (S) et XML Schema. Il peut être personnalisé pour fournir un support adapté au domaine pour la création de modèles de connaissances et la saisie de données. L'éditeur Protégé-OWL permet aux utilisateurs de:

- ✓ Charger et enregistrer des ontologies OWL et RDF.
- ✓ Éditer et visualiser des classes, des propriétés et des règles SWRL.
- ✓ Définir les caractéristiques de classe logique telles que les expressions OWL.
- ✓ Exécuter des raisonneurs tels que des classificateurs de logique de description.

- ✓ Modifier les individus OWL pour le balisage Web sémantique.

L'architecture flexible de l'OWL-Protégé facilite la configuration et l'extension de l'outil. OWL-Protégé est étroitement intégré à Jena et dispose d'une API Java open source pour le développement de composants d'interface utilisateur personnalisés ou de services Web sémantiques arbitraires. En outre, l'OWL-Protégé s'appuie sur une solide communauté de développeurs et d'utilisateurs universitaires, gouvernementaux et d'entreprises, qui l'utilisent pour ses solutions de connaissances dans des domaines aussi divers que la biomédecine, la collecte de renseignements et la modélisation d'entreprise. Dans notre système, il est utilisé pour construire notre modèle de contexte initial.

3.2. Jena

Jena, pris en charge par Apache Software Foundation, est un framework Java permettant de créer des applications Web sémantique. Il regroupe un ensemble d'outils et de bibliothèques Java permettant de développer des applications, des outils et des serveurs Web sémantique et à données liées [175].

Par le biais de l'API d'ontologie, Jena souhaite fournir une interface de programmation cohérente pour le développement d'applications d'ontologies, quel que soit le langage d'ontologie utilisé dans les programmes.

Jena peut être utilisé pour créer et gérer des graphiques RDF. Il a des classes d'objets pour représenter les graphiques, les ressources, les propriétés et les littéraux. Les interfaces représentant les ressources, les propriétés et les littéraux s'appellent respectivement Ressource, Propriété et Littéral. En Iéna, un graphe s'appelle un modèle et est représenté par l'interface du modèle. Un modèle d'ontologie est une extension du modèle Jena RDF, offrant des fonctionnalités supplémentaires pour la gestion des ontologies. Les modèles d'ontologies sont créés via Jena ModelFactory. Le concept fondamental en ontologie est la classe, la propriété est l'individu. Jena fournit une solution réalisable pour les gérer en Java.

Les classes sont les blocs de construction de base de l'ontologie. Une classe simple est représentée dans Jena par un objet OntClass. Une classe d'ontologie est une facette d'une ressource RDF. Une façon, par conséquent, d'obtenir une classe d'ontologie qui consiste à convertir une ressource RDF simple en facette de classe.

La propriété désigne le nom d'une relation entre des ressources ou entre une ressource et une valeur de données dans l'ontologie. Cela correspond à un prédicat dans les représentations logiques. Les applications ontologiques doivent stocker, récupérer et faire des affirmations directement sur les propriétés. Ainsi, Jena dispose d'un ensemble de classes Java qui permettent aux utilisateurs de gérer facilement les propriétés représentées dans un modèle ontologique. Une propriété dans un modèle d'ontologie est une extension de la propriété de base de la classe de l'API Jena et permet d'accéder aux informations supplémentaires pouvant être affirmées à propos des propriétés dans un langage d'ontologie. La super-classe d'API commune pour la représentation des propriétés d'ontologie en Java est OntProperty.

Dans OWL Full, toute valeur peut être un individu - les déclarations d'ontologie sont donc le sujet des triples dans le graphe RDF. Dans OWL Lite et DL, les termes de langage et les données d'instance utilisés par l'application sont conservés séparément, par définition du langage. Jena soutient donc une simple notion d'individu, qui est essentiellement un alias pour une ressource. Bien que les individus soient en grande partie synonymes de ressources, ils fournissent une interface de programmation cohérente avec les autres classes Java de l'API d'ontologie.

4. L'implémentation de l'ontologie GUSP-Onto

La conception des ontologies de domaine et du profil utilisateur était une des étapes les plus importantes de notre projet. Ces dernières sont implémentées en utilisant l'éditeur Protégé

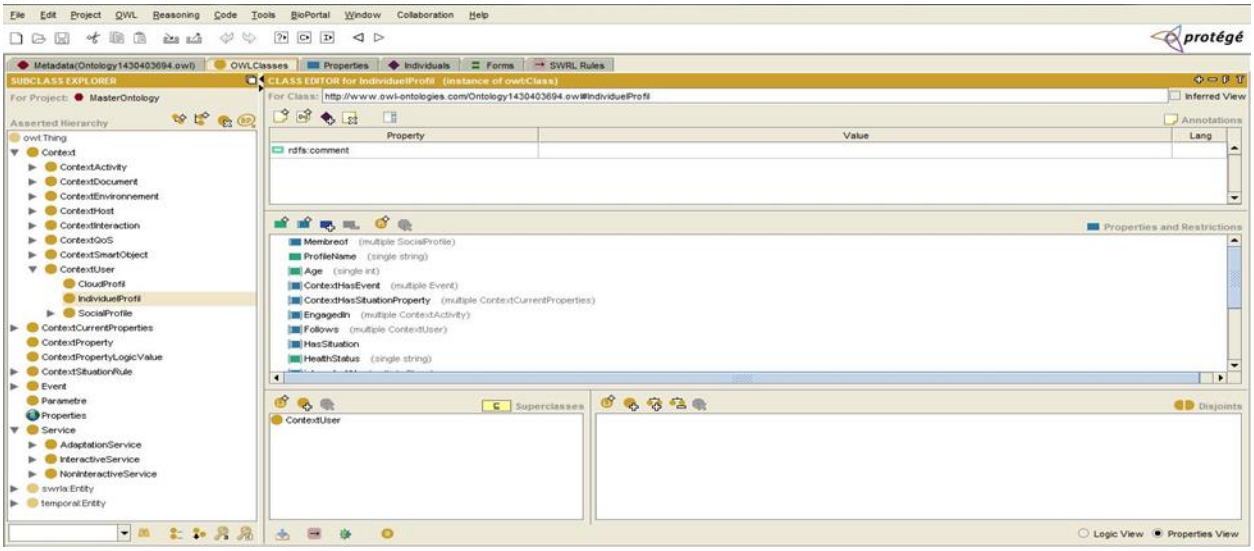

Figure 4.10 Aperçu de notre ontologie sous Protégé.

sauvegardé sous le fichier : "*GUSP-Onto.owl*". La *Figure 4.12* illustre notre ontologie qui contient les classes principales suivantes : *Context, Event,ContextSituationRule, ContextProperty, ContextProperty,LogicValue, Service,ContextCurrentProperties*

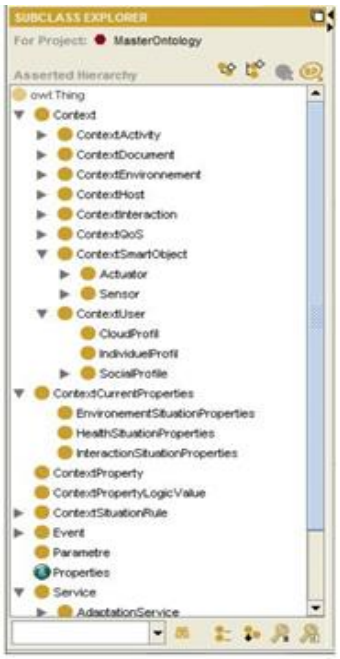

Figure 4.13 : Classes et sous classes.

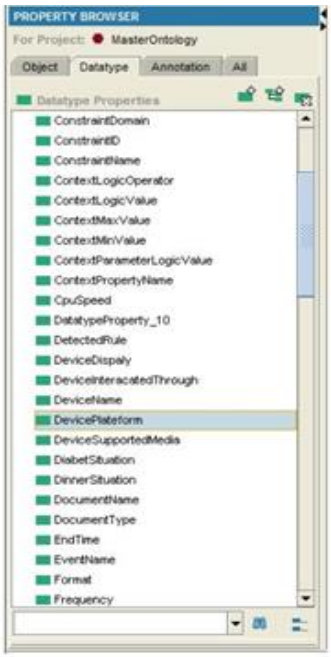

Figure 4.14 : Propriétés des classes.

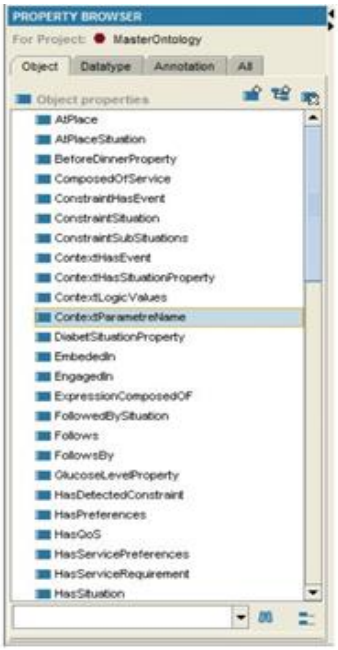

Figure 4.15 : Relations entre les classes.

La *Figure 4.13* illustre l'ensemble des classes et sous classes de notre ontologie où ces propriétés sont présentées dans la *Figure 4.14* et la relation entre ces classes dans la *Figure 4.15*.

Après la définition des classes et les relations entre eux, on procède à l'étape d'instanciation des individus. La *Figure 4.16* illustre quelques exemples d'individus.

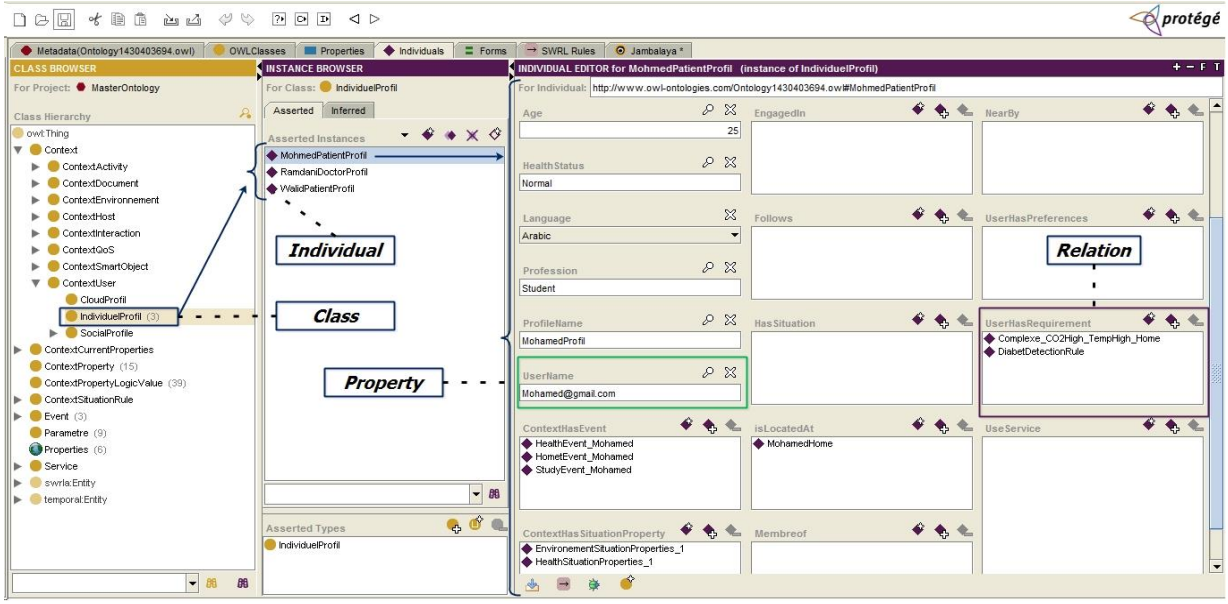

Figure 4.16 Aperçu de notre ontologie sous Protégé.

On a enrichi aussi la sémantique de notre ontologie OWL par des règles SWRL (cf.Figure 4.14). On a opté pour l'utilisation du moteur d'inférence JESS pour réaliser des inférences de la logique de premier ordre pour fournir des services appropriées selon les situations des utilisateurs et leurs informations contextuelles. Deux types de règles sont à respecter :

- Les règles d'interrogation des individus (données contextuelles, services contextuels, dispositifs...) (figure 4.17) et de déploiement des services appropriés aux utilisateurs.

- Les règles d'inférence pour raisonner sur les individus et déduire de nouvelles connaissances sur ces individus (figure 4.19).

Règle	Langage SWRL
IF *Profil.UserName= "Mohamed"* AND *Profil.Location ="Home"* AND Device.Location = "Home" THEN select device	**IndividuelProfil(?profil) ^ UserName(?profil,"Mohamed@gmail.com") ^ IslocatedAt(?profil, ?place) ^ Location(?place,"Home") ^ PlaceHaseInterativeObject(?place, ?device) ^ IPDevice(?device, ?ip) ^ DevicePlatform(?device, ?platform) ^ HasBattery(?device, ?hasBattery) ^ DeviceName(?device, ?deviceName)→sqwrl:select(?profil, ?place, ?device, ?ip, ?platform, ?hasBattery, ?deviceName)**

Figure 4.17 Exemple de règle d'interrogation SWRL.

Règle	Langage SWRL
IF *GlucoseLevel > 1.45* AND *GlucoseLevel < 2.50* THEN *GlucoseSituation :="High"*	**IndividuelProfil(?profil) ^ ContextHasSituationProperty(?p,?property) ^ HealthSituationPrpperties(?property)^ SmartObjectAtUser(?s,?p) ^ GlucoseSensor(?s) ^ MesuredValue(?s, ?v) ^ ContextProperty(?cp) ^ ContextPropertyName(?cp, ?np) ^ swrlb:equal(?np,"Glucose") ^ ContextLogicValues(?cp,?clv) ^ ContextMaxValues(?cp,?max) ^ ContextMinValues(?cp,?max) ^ sqwrl:greaterThanOrEqual(?v, ?min) ^ sqwrl:lessThanOrEqual(?v, ?max)^ contextLogicValue(?clv,?logicValue) ^ MesuredAtTime(?s, ?time) ^ MesuredAtDate(?s, ?Date) ^ isLocatedAt(?p, ?place)→GlucoseStuation(?property, ?logicValue)**

Figure 4.18 Exemple de règles d'inférence SWRL.

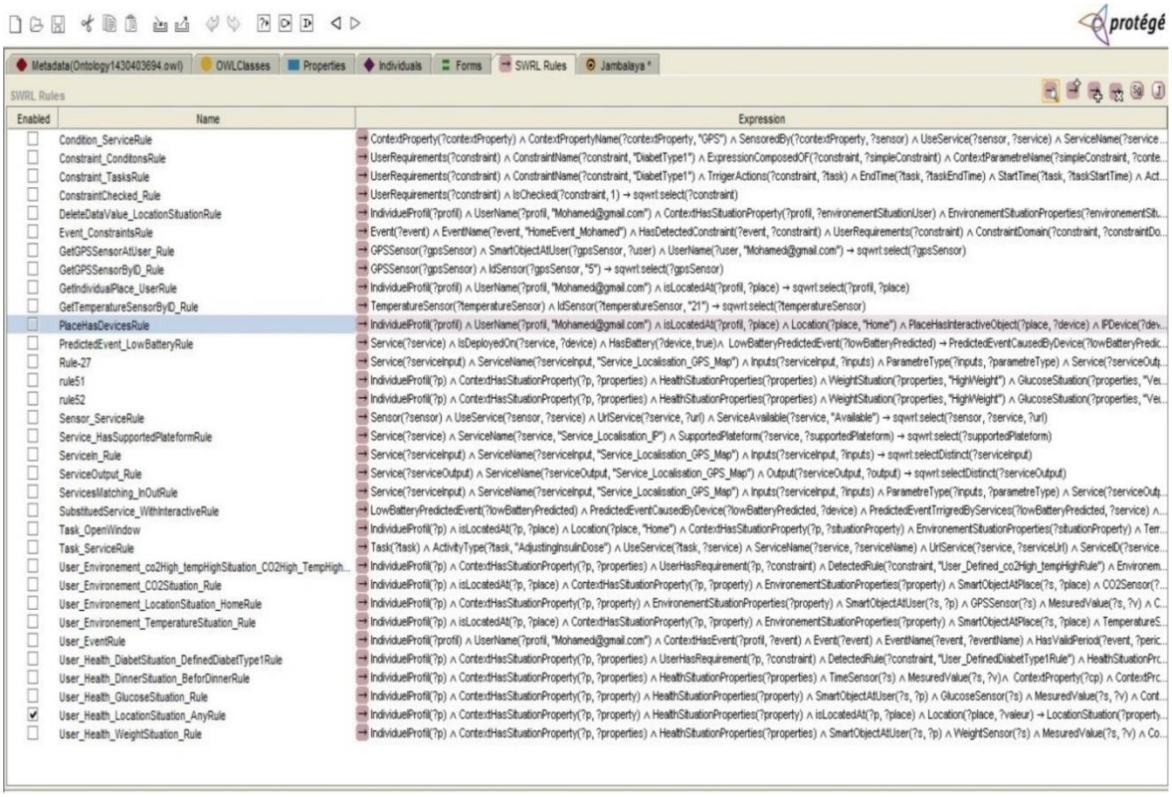

Figure 4.19 Liste des règles d'inférence et d'interrogation.

5. Conclusion

Dans ce chapitre, l'étude se concentre sur le modèle de contexte. Ce modèle organise le profil dans un format lisible par la machine, jetant ainsi les bases du système sensible au contexte. Après avoir reclassé le profil sémantique et analysé l'approche du modèle de contexte existant, nous proposons un modèle basé sur une ontologie. Ce modèle offre une solution pour organiser le contexte caractérisant la situation de l'entité selon un axe temporel et spatial. Les développeurs peuvent facilement utiliser ce modèle pour définir des scénarios spécifiques et obtenir un support historique. La technique d'ontologie permet de prendre en charge le partage de contexte, sa réutilisation et son inférence.

Chapitre 5 : Architecture basé GUSP-Onto pour le déploiement des services sensible aux contextes sur le Cloud Computing

1. Introduction

L'intelligence ambiante (AmI) fait référence à un environnement réel numériquement augmenté, sensible et réactif à la présence de personnes [176]. Dans un monde d'intelligence ambiante, les objets et les appareils de communication travaillent ensemble pour aider les personnes à effectuer leurs activités et tâches quotidiennes (ou professionnelles) de manière naturelle et simple en utilisant des informations même cachées dans le réseau d'objets et d'appareils connectés. AmI correspond à une nouvelle vision de la vie quotidienne (ou professionnelle), composée de différents types de capteurs, actionneurs, objets de communication et dispositifs informatiques, qui génèrent une intelligence omniprésente dans l'environnement qui soutient les activités et les interactions des utilisateurs. Cela signifie que la technologie informatique existera dans tout ce qui nous entoure (appareils, appareils ménagers, objets, vêtements, matériaux) et que tout sera interconnecté par un réseau omniprésent. Le système formé par toutes ces choses intelligentes interconnectées (également appelé l'internet des objets) communique avec l'homme au moyen d'interfaces avancées, naturelles, flexibles et adaptables aux besoins et aux préférences de chaque utilisateur. L'objectif final est d'acquérir un système adaptatif et "intelligent" qui assiste les êtres humains dans leurs activités quotidiennes (ou professionnelles).
Dans la suite de ce chapitre, un scénario de situation AmI est présenté, permettant une contextualisation à la suite d'une explication technique. Ensuite, plusieurs aspects de notre architecture sont expliqués, en commençant par la Couche de contrôleur d'adaptation de service sémantique basée sur le Fog-Computing (Fog-Based SSAC) .L'étape suivante consiste à décrire la couche du contrôleur d'adaptation de service sémantique basée sur le Cloud (Cloud-Cased SSAC). Nous présentons la conception et la réalisation de notre prototype. Avant de conclure nous validons notre architecture.

2. Fog basé GUSP-Onto pour les environnements intelligente (FGIE)

L'approche de Fog Computing intelligente proposée est un système sémantique, souple et prenant en compte la situation, permettant de gérer un grand nombre de profils spécifiques hétérogènes. Il fournit une identification automatique de la situation qui utilise les technologies du Web sémantique, le Fog Computing et les méthodes de mesure de similarité. Il offre deux fonctionnalités. La première assure le traitement simultané du contexte et la gestion efficace des

situations urgentes avec la description sémantique de données de contexte hétérogènes basées sur l'ontologie et le Fog Computing. Par exemple, un capteur de glycémie fournit des données sur le taux de glycémie indépendamment de l'unité de mesure. La deuxième fonctionnalité permet une analyse de situation plus approfondie et des services améliorés pour les cas d'urgence et des services de soins de santé pour les patients. L'approche met en évidence, d'une part, le regroupement de profils des utilisateurs équivalents dans un modèle de profil de Cloud générique basé sur la situation des utilisateurs, et, d'autre part, facilite le déploiement de services appropriés en fonction des utilisateurs (médecin, infirmière, spécialistes ...) pour effectuer des analyses en profondeur. -analyse et services d'urgence.

3. Fog Computing et Architecture de framework basée sur les ontologies

Dans les environnements intelligents, les documents multimédias sont accessibles sur une grande variété de périphériques (objets intelligents multimédias et périphériques mobiles intelligents). L'hétérogénéité de tels dispositifs et les différents besoins des utilisateurs varient toujours chaque jour de la vie réelle. Le stockage de grandes quantités de profils d'utilisateurs spécifiques nécessite de nombreuses comparaisons qui doivent être calculées et mises à jour pour

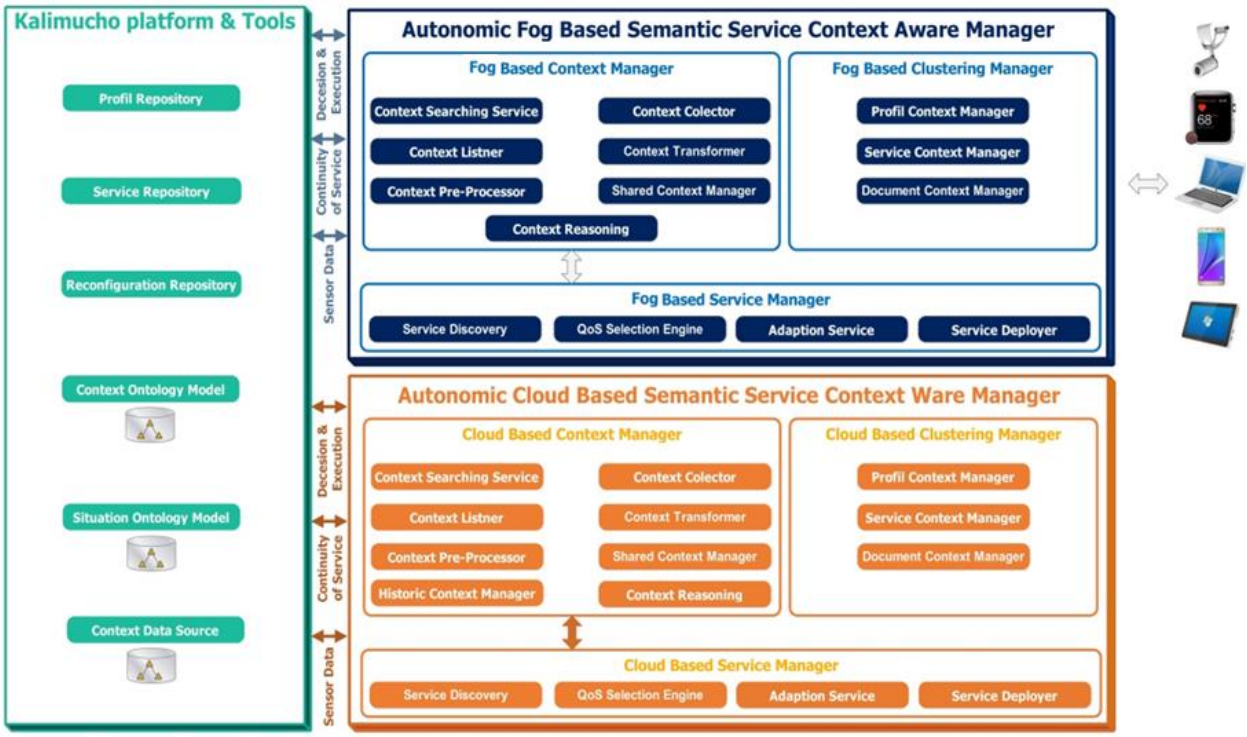

Figure 5.1. L'architecture générale de notre plateforme [114]

chaque changement de contexte. Le cadre doit donc être générique, dynamique et flexible. Nous relevons ce défi en fournissant une nouvelle architecture à deux couches comprenant un serveur local (Fog Computing), le Fog Computing central fournissant une couche de gestion de contexte sémantique pour une gestion efficace des situations et une bonne évolutivité.

L'architecture générale de notre framework est présentée dans cette section. La figure 5.1 détaille nos composants de structure dans un centre de données local (Fog-Computing) et dans le Cloud. Les principales caractéristiques de notre framework sont les suivantes :(1)- les composants sont déployés à l'exécution entre les paradigmes de Fog Computing et de Cloud Computing. (2)- fournissent des mécanismes intelligents qui permettent d'extraire et d'agréger des données contextuelles recueillies à partir de différents objets et périphériques intelligents hétérogènes et soumis à des contraintes, (3)- regrouper certaines contraintes et certains services de haut niveau dans une structure d'ontologies génériques de premier niveau. Plus précisément, si deux profils sont assez similaires grâce à la distance adaptée proposée, (4)- détection précoce des situations urgentes et fourniture de tous les services multimédia distribués aidant les utilisateurs à accéder aux documents multimédia et à les diffuser. Un environnement intelligent est composé de différents objets intelligents portables et prend en charge différents protocoles de connectivité tels que Bluetooth, Wi-Fi, ZigBee, etc. pour connecter un centre de données local et un Cloud. L'architecture est composée de deux couches:

- ✓ Couche de contrôleur d'adaptation de service sémantique basée sur le Fog-Computing (Fog-Based SSAC)
- ✓ Couche du contrôleur d'adaptation de service sémantique basée sur le Cloud (Cloud-Cased SSAC)

3.1. Couche de contrôleur d'adaptation de service (FOG-BASED SSAC)

Le Fog intelligent est une couche de base permettant de déterminer rapidement de manière dynamique la reconfiguration au moment de l'exécution de la prise en charge du déploiement pour les situations urgentes. Il fournit un regroupement de profils, un traitement de contexte, une identification de situation urgente et une diffusion d'informations multimédia liées à des services intelligents.

3.1.1.Gestionnaire de contexte basé sur le Fog-Computing

Ce composant est responsable de la vérification du contexte de plusieurs utilisateurs d'un environnement intelligent donné. Il est également responsable de la détermination des situations urgentes en fonction d'un groupe d'utilisateurs contextuels sémantiquement similaires. Il est composé des composants suivants : (1)- ***Context Searching Service***, qui permet aux consommateurs de rechercher des sources d'informations contextuelles de bas niveau dans le domaine intelligent à partir de descriptions sémantiques, (2)- ***Context Listener,*** est chargé de surveiller le changement de contexte de l'utilisateur et d'appeler le préprocesseur de contexte. (3)- ***Context Collector,*** responsable de la gestion et de l'agrégation des données de contexte brutes provenant de différents dispositifs de détection, (4)- ***Context Pre-processor,*** qui est responsable du prétraitement des données de contexte brutes provenant de différents dispositifs de détection, (5)- ***Context Transformer,*** chargé de transformer les données de contexte de bas niveau en données sémantiques de haut niveau et de les enregistrer dans le modèle ontologique. (6)- ***The Shared Context Manager,*** chargé de la gestion des données de contexte partagées, des contraintes de périphérique équivalentes et des propriétés de document commune. (7)- ***Context Reasoning,*** est chargé d'identifier les situations urgentes à l'aide de l'ontologie et des règles de situation.

- ✓ ***Clustering contexte basé sur le Fog-Computing***

Ce composant sert de module de clustering pour regrouper certains profils, services et documents spécifiques. Ce regroupement accélère le processus de recherche des services d'adaptation pertinents. L'unification de certaines descriptions de profils procurera aux analystes une meilleure visibilité et une meilleure connaissance de la situation pour le regroupement de profils. Les trois composants principaux du clustering de contexte basé sur le brouillard sont : (1)- le service de recherche de contexte, qui permet aux consommateurs de rechercher des sources d'informations contextuelles de bas niveau dans le domaine intelligent à partir de descriptions sémantiques. (2)- l'écouteur de contexte, responsable de la surveillance du contexte de l'utilisateur change et appelle le préprocesseur de contexte et (3)- Context Collector, responsable de la gestion des données de contexte brutes provenant de différents dispositifs de détection.

✓ ***Gestionnaire de service basé sur le Fog-Computing***

Ce composant est chargé d'assurer la continuité du service, en fournissant tous les services multimédias distribués qui aident les utilisateurs à accéder à des services d'accès / de diffusion de documents multimédias, d'urgences et de problèmes de santé, et effectuer une adaptation des documents de qualité aux membres de leur famille.

Le gestionnaire de services basé sur le brouillard est composé de quatre composants :

- Découverte de services, responsable de la découverte des services disponibles à partir de situations identifiées,
- le moteur de sélection de service QoS, permet d'analyser la liste des services sémantiquement équivalents (location, temps, catégorie) avec différentes QoS (temps d'exécution, type de modalité entrée/sortie, coût de service). Il fait appel au composant gestionnaire de contexte service.
- le service d'adaptation, qui est responsable du déploiement et de la continuité du service sur les appareils mobiles et
- le service de déploiement, permet de créer automatiquement des scripts de (re)configuration selon les situations des utilisateurs qui vont être placés dans le Fog-Computing.

3.2. Couche de contrôleur d'adaptation de service (CLOUD-BASED SSAC)

Ce composant fonctionne sur une infrastructure de Cloud qui doit évoluer horizontalement pour prendre en charge le grand nombre d'objets intelligents connectés, ainsi que verticalement pour traiter la diversité des situations des utilisateurs dans les domaines intelligents. Les composants principaux de cette couche comprennent trois sous-composants : (i) clustering de contexte basé sur le Cloud, (ii) gestionnaire de contexte basé sur le Cloud et (iii) Gestionnaire de services basé sur le Cloud.

- ***Clustering de contexte basé sur le Cloud*** : ce composant est responsable de la mise en cluster de profils de contexte évolutifs prenant en charge le volume et la variété de données de contexte. Il est capable de traiter un très grand nombre de services hétérogènes et de profils d'utilisateurs et d'intégrer facilement de nouveaux profils.

- ***gestionnaire de contexte basé sur le Cloud*** : ce composant dispose de capacités de traitement de contexte évolutives, ainsi que de la possibilité de consolider et d'analyser les données de contexte et de déterminer les situations des utilisateurs.
- ***Gestionnaire de services basé sur le Cloud*** : Ce composant est responsable de la création de nouveaux services à partir d'une infrastructure informatique en nuage pour les situations identifiées. Il est également capable de fournir de nouvelles mises à jour de services et de gérer les situations des utilisateurs.
- ***Ontologie de Profil générique sensible à la situation :*** La description sémantique des données de contexte est une caractéristique importante pour permettre le raisonnement de contexte et l'interopérabilité de données de contexte dans des domaines intelligents hétérogènes.
- ***Référentiels de profils contextuels*** : Les profils doivent être synchronisés automatiquement d'un centre de données local (Fog) vers le Cloud. Les profils sont des services, des utilisateurs et des documents.

3.3. Plate-forme de Kalimucho

Kalimucho est une plate-forme middleware utile pour la gestion d'applications mobiles distribuées avec déploiement (ré) déploiement et migration de composants dynamiques. Il offre une excellente stratégie de gestion des services intelligents et de déploiement de politiques prédéfinies dédiées à la gestion du contexte distribué sur le domaine partagé. Nous étendons la plateforme Kalimucho pour gérer efficacement l'explosion de profils d'utilisateurs, identifier les situations et fournir des services appropriés en temps réel à plusieurs utilisateurs simultanément.

4. Le modèle fonctionnel de notre architecture

Le principal objectif de notre Framework est de gérer un grand nombre de profils d'utilisateurs hétérogènes afin de fournir simultanément des services multimédias personnalisés à plusieurs utilisateurs. La nouveauté de notre stratégie consiste à introduire le concept de la sensibilité au contexte des utilisateurs et de sa réactivité afin d'accélérer le processus d'identification de situations pertinentes en fonction de profils d'utilisateurs similaires au moment du traitement. Notre suggestion Fog Computing intelligente est accompagnée de trois nouvelles étapes. La première étape regroupe des profils de contexte spécifiques dans un profil de contexte générique.

La deuxième étape consiste à identifier et à gérer toute situation urgente. Afin de gérer l'évolutivité lors de la résolution du processus de sélection et de découverte de service Cloud sensible à la situation, le processus de sélection de service dynamique est amélioré dans la troisième étape en regroupant les contraintes de périphérique d'utilisateurs sémantiquement équivalents pour la sélection d'un ensemble de services d'adaptation pertinents en fonction de la situation actuelle .

4.1. Etape 1 : Modélisation des profils génériques des utilisateurs et le regroupement

Comme illustré à la figure5.2, La première étape consiste à collecter le traitement préalable des données de contexte brutes provenant de plusieurs dispositifs hétérogènes de détection omniprésents à l'aide du gestionnaire de contexte basé sur Fog Computing. Certains sont récupérés automatiquement à partir de capteurs portables et d'appareils mobiles. D'autres sont spécifiques à

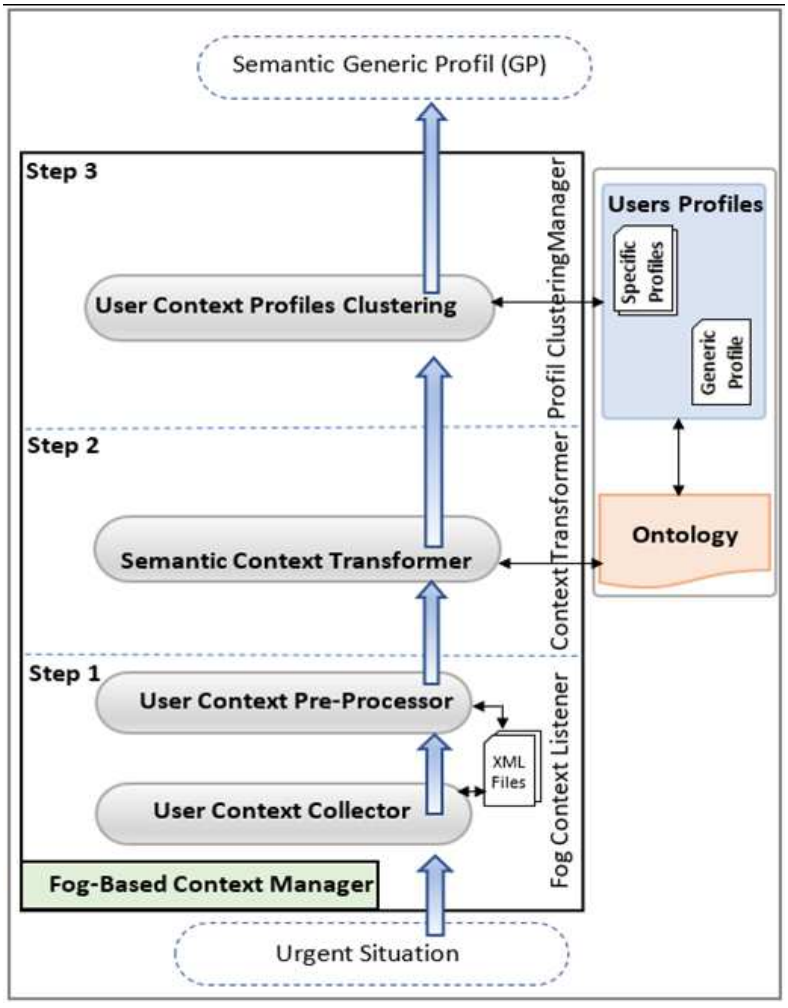

Figure 5.2. Modélisation des profils générique des utilisateurs et le regroupement [114]

l'utilisateur. Les données brutes personnelles sont enregistrées au format XML dans un référentiel de profils utilisateur. La réception du contexte de bas niveau, représenté à la figure 3 par le composant de transformation de contexte basé sur FOG, le transformera en un contexte sémantique de haut niveau. Le modèle d'ontologie de profil d'utilisateur sera mis à jour avec les nouvelles données sémantiques transformées de l'étape précédente. Après avoir reçu et transformé les données de contexte, le gestionnaire de contexte partagé est prêt à appeler le gestionnaire de groupement de profils pour regrouper des profils d'utilisateurs équivalents partageant un contexte similaire (étape 3 de la figure 5.2). L'objectif est de minimiser l'espace de contraintes pour la stratégie d'identification de situation et d'accélérer le processus de découverte des services pertinents. Le gestionnaire de contexte partagé se base sur Fog Computing afin de regrouper les profils des utilisateurs selon des techniques basées sur les ontologies [177][166][178]. L'idée du regroupement de profil est de tenir compte des propriétés de contexte équivalents et leurs données de contexte similaires.

Nous avons calculé une mesure de similarité entre chaque paire de concepts équivalents reflétant le même contexte sémantique. Deux profils spécifiques (SP1, SP2) sont assez similaires en fonction de la mesure suivante :

$$Score = Lexical_{sim} + Property_{sim} + Contextual_{sim} \quad (1)$$

Où

$$Lexical_{sim} = W_{lexi} * Sim_{lexi}(SP_1, SP_2) \quad (2)$$

$$Property_{sim} = W_{prop} * Sim_{prop}(SP_1, SP_2) \quad (3)$$

$$Contextual_{sim} = W_{ctx} * Sim_{ctx}(SP_1, SP_2) \quad (4)$$

Où W_{lexi}, W_{prop} et W_{ctx} sont respectivement les poids pour déterminer les méthodes de similarité importance Sim_{lexi} (similarité lexicale), Sim_{prop} (similarité de propriété), Sim_{ctx} (similarité de contexte) telle que $W_{lexi} + W_{prop} + W_{ctx} = 1$. La valeur du score entre les propriétés de contexte de chaque paire de profils d'utilisateur doit être supérieure à 0,8 pour regrouper les profils équivalents. Toutes les méthodes de similarité sont basées sur la mesure normalisée suivante :

$$Sim_{\text{Thesaurus}}(e1, e2) = \alpha^{index} \qquad (5)$$

Nous avons déterminé toutes les paires de profils spécifiques dont les propriétés de contexte d'approximation correspondaient. L'algorithme de matching sémantique, tel qu'illustré à la figure 5.3, génère automatiquement un profil d'utilisateur générique à partir de profils d'utilisateurs de contextes spécifiques différents Sp1 et Sp2. À partir des lignes 1 à 2, l'algorithme construit les listes de nœuds suivantes : L1 et L2, où chaque nœud contient une référence «parent», une référence «enfant» et une relation parent-enfant de chaque profil utilisateur spécifique. À partir des lignes 3 à 6, nous avons déterminé le degré de similarité pour des profils spécifiques Sp1 et

```
Inputs: Sp1, Sp2                          /* Specific Services Profiles of Sp1, Sp2 */
W_lexi , W_prop , W_ctx , threshold       /* weight and threshold values */
L1: set TNode                             /* All relation between pair of elements (e1,e2) of Sp1 */
L2: set TNode                             /* All relation between pair of elements (e1,e2) of Sp2 */
Output: Gp                                /* Generic Profile */
        Matching_Matrix [][]              /* Matching matrix between all pair (e1,e2) of Sp1 and Sp2 */
        LG: set of TNode                  /* list of couples where its specific and generic elements are identical */
1. Begin
2.      L1 = ConstructSpecificList (Sp1);
3.      L2 = ConstructSpecificList (Sp1);
4.      For each concept c1 ϵ Sp1 do      /* Compute matching matrix between Sp1 and Sp2 */
5.          For each concept c2 ϵ Sp2 do
6.              Matching_Matrix [c1][c2] = Similarity_Degree (c1,c2); /* similarity degree defined in [1]*/
7.          End
8.      End
9.      global_score = ComputeSimilarityDegree (Matching_Matrix);
10.     If (global_score>threshold) then
11.         LG = Merge_liste (L1,L2);
12.         Gp = Construct_OWLModel (LG);
13.     End
14.     Return Gp
15. End
```

Figure 5.1 L'algorithme de construction de profil d'utilisateur générique [114].

Sp2, en fonction de la distance entre chaque paire de propriétés de profil et de leurs valeurs de contexte. L'algorithme calcule la matrice correspondante pour chaque couple c1 et c2 de profils d'utilisateurs spécifiques à l'aide de la fonction Similarity_Degree. La question est d'identifier les profils SP1 et SP2 les plus similaires ou les plus proches en ce sens qu'ils présentent la plus grande similarité.

4.2. Étape 2 : Gestion de la situation

Le processus de gestion d'une situation (étape 4) est décrit à la figure 5.4. Le profile Clustering Manager appelle le composant de raisonnement par contexte pour identifier les nouvelles situations urgentes de plusieurs utilisateurs sur la base des profils des utilisateurs génériques à l'aide d'une mesure de similarité de situation. Le composant de raisonnement contextuel envoie la liste des situations identifiées au modèle d'ontologie à enregistrer. Les informations de contexte relatives aux utilisateurs sont synchronisées automatiquement du SSAC basé sur le Fog au SSAC basé sur le Cloud, permettant une analyse de situation plus approfondie. L'identification de la situation est une partie importante de l'approche présentée qui détecte les situations à partir des paramètres de contexte pertinents actuels. Il existe deux techniques principales pour identifier les situations dans l'environnement intelligent et sélectionner le service approprié en fonction des situations : les techniques basées sur le raisonnement et celles basées sur la similarité [177][166]. La technique basée sur la similarité représente la méthode qui applique la décision d'identification de situation directement après le traitement des informations de contexte d'utilisateur actuelles. Notre mesure de similarité étend les propriétés de Sim définies dans Alti, Lakehal [173]. Il est formalisé comme suit :

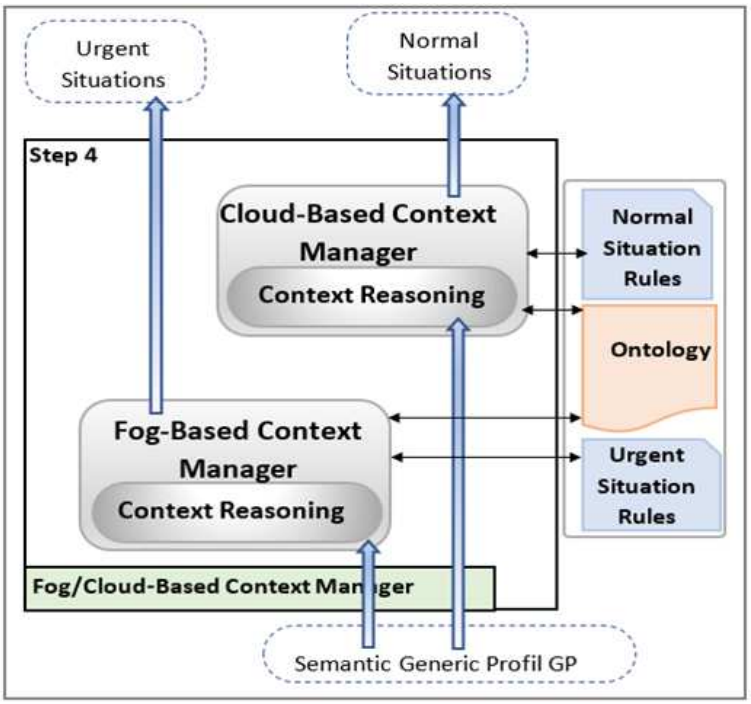

Figure 5.2 Gestion de la situation [114]

$$Sim(Q,S) = \frac{a}{a+b} = \frac{a\sum_{i=1}^{a} w_i * sima(Q_i, Sj)}{\left(a\sum_{i=1}^{a} w_i\right) + \left(b\sum_{i=1}^{b} w_{a+i}\right)}$$

Où « b » est un attribut de contexte inhabituel d'une situation donnée S, et « a » est un certain nombre de contextes attributs communs entre un profil générique Gp et une situation S. La fonction Sim détermine le score de matching entre Gp et une situation donnée S en fonction de ses poids associés et de la valeur de similarité atomique de chaque attribut de contexte commun (Gp_i,S_i). Nous identifions les situations d'un groupe d'utilisateurs avec un score de correspondance plus élevé.

La figure 6 montre l'algorithme d'identification de situation dynamique dans son concept. Il prend en entrée un modèle de profil générique utilisateur et un ensemble de situations (ligne 1) et en sortie le score de correspondance le plus élevé (ligne 2). Initialement, pour chaque couple de concepts communs de contraintes de contexte et de profil, un ensemble de valeurs de similarité atomiques est initialisé et calculé (ligne 4-7). Ensuite, une valeur de similarité globale Sim est calculée entre un profil de contexte générique et chaque situation urgente (Eq.1; ligne 8). Enfin, le jeu de résultats est trié et une mesure de meilleure correspondance avec une relation de mise en correspondance de situation est renvoyée (ligne 10).

4.3. Étape 3 : diffusion de l'information multimédia

Une fois que le raisonnement contextuel basé sur Fog Computing a identifié les situations urgentes des utilisateurs, nous nous concentrons sur l'envoi simultané des informations multimédia aux

```
1.  Inputs: A User Generic Profile Gp, Situation S []
2.  Outputs: Overall similarity score and the identified situation.
3.  Begin
4.      For each urgent situation Do
5.          For i=1 To Number_Of_Common_Context_Attributes Do
6.              Get each context condition of S, the atomic similarity "sima" value
7.          End For
8.              Compute an overall similarity value defined as Eq.1
9.      End For
10.     Output the higher matching score.
11. End.
```

Figure 5.3 Algorithme d'identification de situation dynamique [114]

utilisateurs intéressés qui devraient pouvoir s'exécuter sur différents types de périphériques. Le processus de déploiement de services distribués et de diffusion d'informations multimédias de santé aux utilisateurs intéressés (étape 5 à l'étape 9) est décrit à la figure 5.5. Le composant Contexte partagé basé sur FOG regroupe de manière sémantique les contraintes de périphérique des utilisateurs (étape 5 de la figure 5.6). La découverte de services basée sur Fog Computing est déclenchée par les situations urgentes signalées par le raisonnement de contexte basé sur FOG Computing (étape 6 de la figure 5.6). Il détermine ensuite les services appropriés aux situations urgentes identifiées à partir du référentiel de services. Afin de parvenir à une meilleure découverte de service, nous regroupons sémantiquement des services similaires. Le mécanisme de la mise en cluster de services consiste à grouper sémantiquement un grand nombre de services Cloud Computing hétérogènes en fonction de leur catégorie de services, du même type de situations pouvant déclencher ces services, ainsi que des descriptions fonctionnelles et contextuelles. Cette méthode garantit une sélection rapide des services afin de satisfaire la plupart des contraintes des utilisateurs cibles sur les équipements cibles, constituant ainsi une qualité de service (QoS) importante. Nous utilisons un profil de service générique pour sélectionner le meilleur service de plusieurs profils spécifiques possibles. (Étape 7 Figure 5.6). Ensuite, le moteur de sélection QoS sélectionne les meilleurs services d'adaptation multimédia en fonction de différentes politiques (temps de réponse, qualité du support, disponibilité totale et économie d'énergie). Le composant Adaptation au service effectue des adaptations partielles ou complètes en fonction des contraintes actuelles : lecteurs, codecs, ressources, profil utilisateur et matériel (étape 8 de la figure 7). En raison de la mobilité des utilisateurs et des contraintes de batterie faible pouvant interrompre l'exécution des services mobiles dans les systèmes omniprésents, nous recherchons une manière intelligente d'assurer la continuité du service sur les appareils mobiles en déployant les différents services hétérogènes. Par conséquent, ***Service Deployer*** déploiera les services sélectionnés par l'adaptation de service sur le périphérique mobile des utilisateurs. Par conséquent, ***Service Deployer*** déploiera les services sélectionnés par ***Service Adaptation*** sur le périphérique mobile des utilisateurs. Il génère automatiquement les fichiers de configuration par le générateur de code JAVA après analyse de l'environnement et les envoie à la plate-forme Kalimucho (étape 9 de la figure 7). Il est responsable de l'exécution des services, de la continuité de ces services et de la sauvegarde de la nouvelle configuration de service en cas de ressources insuffisantes sur les

périphériques mobiles. Parallèlement, les patients sont automatiquement analysés sur le Cloud, ce qui permet d'améliorer les services d'urgence.

5. Implémentation du prototype

Notre modèle d'ontologie est implémenté dans l'outil Protégé [179]. Cet outil de modélisation visuelle prend en charge la création et la gestion de modèles d'ontologies pour les domaines intelligents et leurs applications. L'outil protégé s'appuie sur une solide communauté d'utilisateurs issus des universités, des gouvernements et des entreprises pour concevoir une sémantique formelle, commune et partagée avec le langage OWL (Web Ontology Language). L'architecture de plug-in Protégé peut être adaptée pour créer des applications aussi bien simples que complexes basées sur des ontologies. Les développeurs peuvent intégrer la sortie de cet outil protégé à des systèmes de règles afin de construire une large gamme de systèmes intelligents. Nous avons utilisé

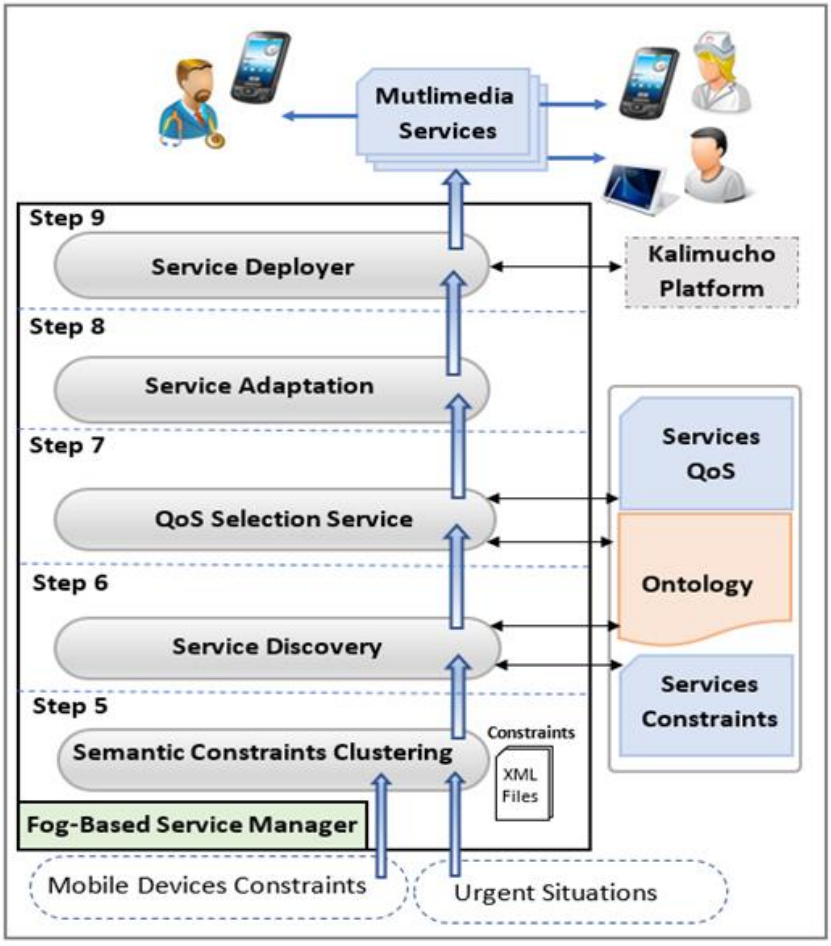

Figure 5.4 Diffusion de l'information multimédia [114]

l'outil Protégé pour définir explicitement différentes classes de GUSP-Onto, leurs propriétés et leurs instances.

Le prototype est développé avec une architecture à deux couches : basée sur Fog et aussi sur le Cloud. Les objets intelligents et les appareils mobiles utilisent un smartphone Android 1.5 GPS Wi-Fi Galaxy 512 Mo de RAM et 4 Go de ROM. Un ordinateur de bureau simule le Fog Computing. Il s'agit du PC de bureau Dell doté d'un processeur Intel Core i5 4460 (3,20 GHz) et de 4 Go de mémoire. Nous avons déployé le SSAC basé sur FOG intelligent sur le PC, qui inclut des composants de traitement de contexte implémentés dans NetBeans 6.1. Le SSAC basé sur la technologie FOG intelligente a quatre fonctions : recevoir des données de contexte d'objets intelligents, raisonner en fonction de la situation, envoyer du multimédia aux utilisateurs mobiles intéressés et envoyer des données de contexte au Cloud.

Le serveur de Cloud utilise Amazon Extra Large Cloud. Il dispose de 160 Go de mémoire et de 124,5 unités de calcul EC2. Il implémente Socket et notre framework de moteur de raisonnement sémantique. Le serveur de Cloud représente et raisonne sur un grand nombre d'informations contextuelles. Il dispose également de services Web fournissant une ontologie à un groupe d'utilisateurs. Dans le cadre de notre travail, les cas d'utilisateurs possibles sont variés et vastes. Considérez l'étude de cas suivante sur le diabète.

5.1. Étude de cas sur le diabète et scénarios de la vie réelle

Notre approche offre de grands potentiels dans les hôpitaux intelligents afin de faciliter le partage d'informations de santé contextuelles provenant de profils de patients différents. Pour commencer, la plate-forme d'intelligence smart-health basée sur le Fog Computing assure le traitement simultané du contexte et la gestion efficace des situations urgentes avec une représentation sémantique de différentes données de contexte (par exemple, un capteur de glycémie fournit des données de niveau de glycémie avec différents types et formats de données).

Deuxièmement, il analyse les données contextuelles contrôlées et collectées de patients équivalents et identifie leurs situations urgentes. Enfin, il déploie de manière dynamique les services appropriés chez les utilisateurs les plus proches (médecins, infirmières, spécialistes) pour une préparation immédiate à l'accueil du patient. Parallèlement, les informations relatives aux patients sont également analysées automatiquement sur le Cloud, ce qui permet une analyse plus

approfondie de la situation et des services améliorés pour les urgences et les services de soins de santé. Comme le montre la figure 5.7, l'hôpital intelligent est composé de plusieurs étages, chaque étage comprend un certain nombre de chambres de patients. Chaque chambre est équipée d'une caméra IP. Le patient est équipé de biocapteurs portables (glucomètre, balance de poids, bracelet intelligent et GPS) avec divers protocoles de réseau (Wi-Fi, GSM, etc.). Ces capteurs sont utilisés pour promouvoir des comportements sains chez les patients. Les informations de contexte (niveau de glucose, poids, courte vidéo du patient, coordonnées GPS) sont collectées par le boîtier décodeur. Des données d'un contexte bien déterminé peuvent être détectées par différents dispositifs (c'est-à-dire qu'un patient est localisé à l'aide d'une caméra IP ou de dispositifs GPS avec différents types et formats de données). Tout objet intelligent situé autour de l'utilisateur peut être un hôte potentiel ou une source d'informations importante. Ces informations contextuelles sont utiles pour identifier un ensemble d'activités d'utilisateurs. Les informations de contexte surveillées sont collectées dans une passerelle intelligente. Cette passerelle collecte des données de contexte surveillées par des capteurs. Les données de santé contextuelles agrégées et les signes vitaux sont transmis à smart-health. En cas de situation critique (situation de coma diabétique hypoglycémique), le FOG smart-health déclenche une réponse immédiate au service d'urgence disponible le plus proche. Le ***Context Reasoning*** à base de FOG identifie la situation de coma diabétique hypoglycémique comme suit: *"IF Glucose is very low AND Temperature of the user is very High, AND Location is Inside_ Hospital AND Period is before Dinner THEN hypoglycemic diabetic coma"*.

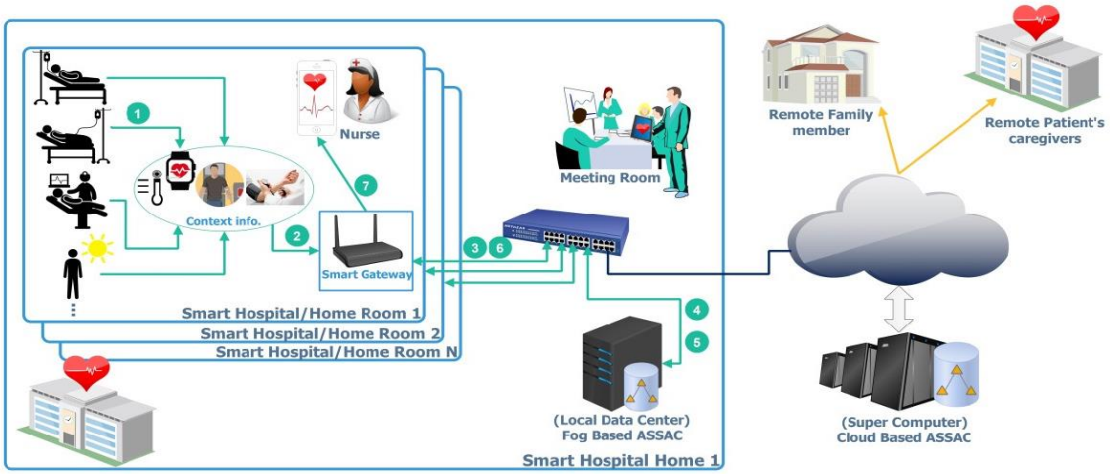

Figure 5.5 Smart hôpital avec leurs scénarios possibles [114]

Dans le premier scénario présenté dans notre architecture, le composant principal du SSAC basé sur FOG consiste à superviser plusieurs patients afin d'identifier les situations urgentes. Le ***Context Collector*** collecte et prétraite les données de contexte brutes des périphériques intelligents et les stocke dans le référentiel de profils. Ensuite, il transforme les données de contexte brutes en sémantique à l'aide du composant ***Context Transformer*** basé sur le Fog, qui est ensuite enregistré dans notre ontologie.

La composante ***Profile Clustering*** regroupe les profils de patients équivalents dans un modèle de profil générique tenant compte du contexte, liés à leurs informations de contexte similaires et à leur emplacement (voir le tableau 1). Les situations liées aux patients sont également analysées automatiquement sur le Cloud.

Le coma diabétique hypoglycémique en tant que situation anormale a été identifié par le raisonnement contextuel basé sur FOG en utilisant notre mesure de similarité (équation 1) et un profil générique sensible au contexte qui déclenche le contrôleur de service basé sur FOG. Ce dernier est responsable de l'envoi / de la diffusion des notifications et des documents multimédia à plusieurs utilisateurs disponibles les plus proches (médecin, infirmière, urgence sanitaire) Le composant Service Discovery basé sur FOG est responsable de la découverte des services

Profils de patients spécifiques	Profil générique
Glucose Level (2.5 mmol /L) Temperature of patient (42°C) Longitude 43°29.4201’ Latitude 5°28.19000’ Date Time: 2017-07-08 09.30pm Glucose Level (50 mg /L) Temperature of patient (313.15 K) IP Camera (video + camera position) Date Time: 2013-07-08 09.34 pm	VeryLowGlucoseLevel VeryHighTemperatureLevel CloseBayonneCity InsideHospitalLocation BeforeDinnerPeriod

Tableaux 5.1. Mapping des profils de patients spécifiques dans une ontologie générique

multimédias d’adaptation disponibles en fonction de la disponibilité des utilisateurs et de leurs contraintes actuelles en matière de dispositifs mobiles. Premièrement, les contraintes multimédias équivalentes spécifiées par le médecin et l’infirmière sont regroupées sémantiquement (comme indiqué dans le tableau 2). Ensuite, le moteur de sélection QoS sélectionne les meilleurs services d’adaptation multimédia en fonction de différentes politiques (bande passante, type de support, langue, temps de réponse et modalités d’interaction). Sur la base de nos travaux précédents [173],

le tableau 3 présente les meilleurs résultats d'évaluation pour lesquels les composants de service «Composants de diabète et de codeur / décodeur vidéo d'urgence de haute qualité» sont sélectionnés. Cette sélection est basée sur la valeur du score élevé lorsque l'objectif est de minimiser les coûts d'adaptation et le temps de réponse, ainsi que d'optimiser la qualité du support. Enfin, la sélection de service active le déployeur de services basé sur FOG en déclenchant une action de reconfiguration. Le déployeur de services basé sur FOG est responsable du déploiement et de la continuité du service sur les appareils mobiles. Par conséquent, il utilise le framework

Kalimucho [172] afin de déployer de nouveaux composants de service

Profils de patients spécifiques (Sp1)	**Profil de périphérique spécifique (Sp1)**	**Profil générique**
C1: Bandwidth > 4 Mo ***C2: Media = {video, images}*** ***C3: Modality = {sound}*** ***C4: Language ={Fr.}*** ***C5: Low cost preferred*** ***C6: Location = {inside}***	*C1: Bandwidth > 1 Mo* *C2: Media = {images, text}* *C3: Modality = {click}* *C4: Language ={Eng.}* *C5: Fast time preferred* *C6: Location = {inside}*	*C1: Bandwidth > 1 Mo* *C2: Media = {video, images}* *C3: Modality = {click, sound}* *C4: Language = {Fr., Eng.}* *C5: Low cost preferred* *C6: Location = {inside}*

Table 5-1 Mappage de profils de périphériques spécifiques dans une ontologie générique.

Orchestration des services	**Score**
Emergency Service + High Quality Video Encoder/Decoder	0.83
Emergency Service + High Quality Image Encoder/Decoder	0.73
Guide Service + High Quality Image Encoder/Decoder	0.38
Guide Service + High Quality Image Encoder/Decoder	0.26

Table 5-2 Résultats d'évaluation.

Le second scénario est lorsqu'un médecin participe à une réunion, il peut maintenant utiliser sa tablette pour une vue plus large et ne peut pas recevoir le son pendant une réunion. Une fois que l'absence de bon codec de média pour exécuter le service vidéo est détectée, le gestionnaire d'adaptation peut fournir une vidéo adaptée afin de répondre à toutes les nouvelles contraintes, contextes de réunion et d'exécution. Le contrôleur de service basé sur FOG recherche les services pertinents dans le Cloud et, après sa sélection et son déploiement, le médecin peut suivre le service d'urgence tout en commençant une réunion. Après avoir reçu la demande du médecin, le contrôleur de service basé sur la FOG envoie un message pour informer les infirmières de la première aide nécessaire à effectuer (injecter une dose d'insuline) ou de l'heure de son arrivée.

6. Expérimentations et discussion

Toutes nos expériences de simulation sont réalisées à l'aide d'un PC 4 Go de RAM à 3,4 GHz. L'évaluation comprend 100 instances de patients qui utilisent le SSAC basé sur le Fog intelligent et le SSAC basé sur le Cloud. Il notifie différentes urgences aux spécialistes concernés, aux membres de la famille et aux services d'urgence. Deux types de données contextuelles ont été utilisés : (a) des données contextuelles simples et (b) un fichier vidéo. Un fichier vidéo représente la courte vidéo du patient capturée pendant environ 10 minutes à l'aide d'une adresse IP de la caméra, qui est téléchargée sur le smart Fog. Les données contextuelles simples concernent la situation ou le lieu, l'heure, les mesures de santé (par exemple, le taux de glucose, le poids, etc.) et d'autres données pertinentes sont téléchargés dans le FOG. Pour différents types de données de contexte, différents profils de patients sont créés, nous validons notre approche en présentant les résultats obtenus dans le Cloud et le FOG.

Deux expériences ont été conçues pour étudier les performances et les problèmes connexes des systèmes intelligents basés sur FOG. La première expérience a montré la taille de l'ontologie à la performance du système. La deuxième expérience a analysé le temps de calcul basé sur une technologie différente : Cloud vs. Fog.

6.1. Évaluation de l'efficacité de FOG Computing intelligente

La première performance a été utilisée pour évaluer l'approche proposée en termes de temps de vérification des contraintes et d'identification de la situation. En utilisant le Fog Computing intelligente, nous avons comparé notre algorithme d'identification de situation basé sur une ontologie générique [170].

Nous avons implémenté le modèle de simulation «*Poisson Event Based*» afin de générer un grand nombre de profils de patients et de leurs données de contexte. Les contraintes de profils de patients passent de 10 à 100. Nous avons étudié les performances de l'algorithme et les avons comparées à des résultats de travail similaires [170] afin d'illustrer l'efficacité de notre approche. Ce travail est basé sur FOG Computing sans regroupement de profils d'utilisateurs similaires. Le système est modélisé par un processus de Poisson. Chaque niveau de glucose de la température du corps humain) est initialisé avec une valeur aléatoire dans la plage de [50, 100] (respectivement dans la plage de [37, 39]) et incrémenté automatiquement par une valeur aléatoire dans la plage de [zéro,

5] (dans la gamme de [0.1, 1]). Nous avons recueilli des résultats comprenant les profils des patients d'un centre de données local, le nombre de situations identifiées et leurs précisions.

Notre composant SSAC basé sur FOG regroupe des contraintes similaires pour optimiser le temps d'identification de la situation. La figure 5.8 révèle que dans la majorité des cas, le temps de réponse de la situation d'identification des patients était supérieur à celui trouvé dans le résultat dAazam [170]. Nous avons également remarqué que notre algorithme, comparé à Aazam [170], minimisait le nombre de profils testés à 34,23%. Cela explique que notre travail regroupe certains profils de patients individuels, qui partagent des propriétés de contexte similaires, et accélère le processus de recherche des services pertinents au moment de l'exécution, mieux que les travaux de Aazam [170].

6.2. Évaluer l'efficacité du raisonnement situationnel à l'aide du Cloud Computing et du FOG Computing

La deuxième performance a été utilisée pour comparer le raisonnement de situation en nuage et le raisonnement de situation de brouillard intelligent. Lors de notre expérimentation, nous avons mesuré le transfert de données à partir d'appareils mobiles, puis évalué le temps de raisonnement depuis le Cloud. Nous avons également mesuré le temps de raisonnement entre le brouillard smart-health et le Cloud, comme illustré à la figure 10. Les résqultats montrent que le brouillard smart-

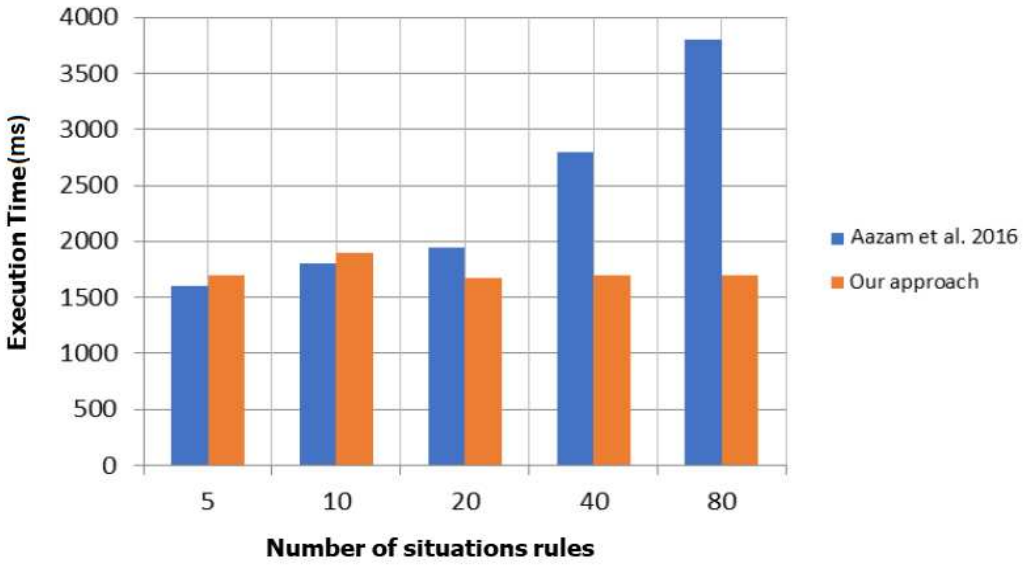

Figure 5.6 Temps d'exécution de l'identification de la situation avec / sans regroupement de contraintes [114]

health a généré des résultats plus tôt que le Cloud. Cela signifie que la méthode proposée utilise un raisonnement de situation intelligent pour identifier les premières situations urgentes. Toutefois, le Fog smart-health s'est avéré plus efficace lors de la mise à jour de nouvelles situations au moment de l'exécution, ce qui a amené le moniteur de contexte à envoyer plus fréquemment les informations de contexte pertinentes.

6.3. Discussion

L'objectif principal de l'approche présentée est de parvenir à une identification efficace de la situation à l'aide de smart FOG Computing. Nous avons effectué les expériences pour smart FOG et d'autres travaux similaires avec un nombre de contraintes différent. Comme le montre la figure 9, Smart FOG tente de minimiser le nombre de contraintes des utilisateurs contrôlés, ce qui permet d'obtenir de meilleures performances en termes de temps, de la situation (figure 9). Ce résultat confirme qu'en utilisant smart FOG, le temps d'identification de la situation est inférieur à 39% par rapport à un travail similaire Aazam et Huh, [170] dans 20 à 40 contraintes.

Depuis l'approche proposée, regroupe de manière dynamique les règles de situations équivalentes, il vérifie les situations urgentes en tenant compte des règles de raisonnement java intelligentes. Toutefois, lors du dernier test avec un nombre élevé de contraintes (80 contraintes), les temps d'identification et de gestion de la situation dans le FOG intelligent sont respectivement inférieurs à 3 s. (80 contraintes), les temps d'identification et de gestion de la situation dans le FOG intelligent sont respectivement inférieurs à 3 s. Ceci est dû à une élimination élevée des contraintes. Cette méthode peut être appliquée au domaine intelligent e-Health en temps réel, où le temps requis est calculé en fractions de seconde. Nous appliquons un raisonnement de situation distribué basé sur FOG pour gérer tous les aspects de l'identification de situation dans tous les environnements intelligents connectés. Ces nouvelles solutions incluent la gestion intelligente des situations (capteurs et appareils intelligents), ainsi que des outils d'aide à la décision et des outils de communication intelligents et distribués. L'approche présentée offre aux utilisateurs la possibilité d'identifier des situations urgentes destinées à les assister dans leurs besoins quotidiens et à gérer tous les services en même temps.

Pour prouver l'efficacité de la technologie FOG intelligente, nous avons mesuré le transfert de données à partir d'appareils mobiles et le temps de raisonnement depuis le Cloud. La figure 5.9 montre les résultats obtenus pour le raisonnement de situation CLOUD avec / sans FOG intelligent.

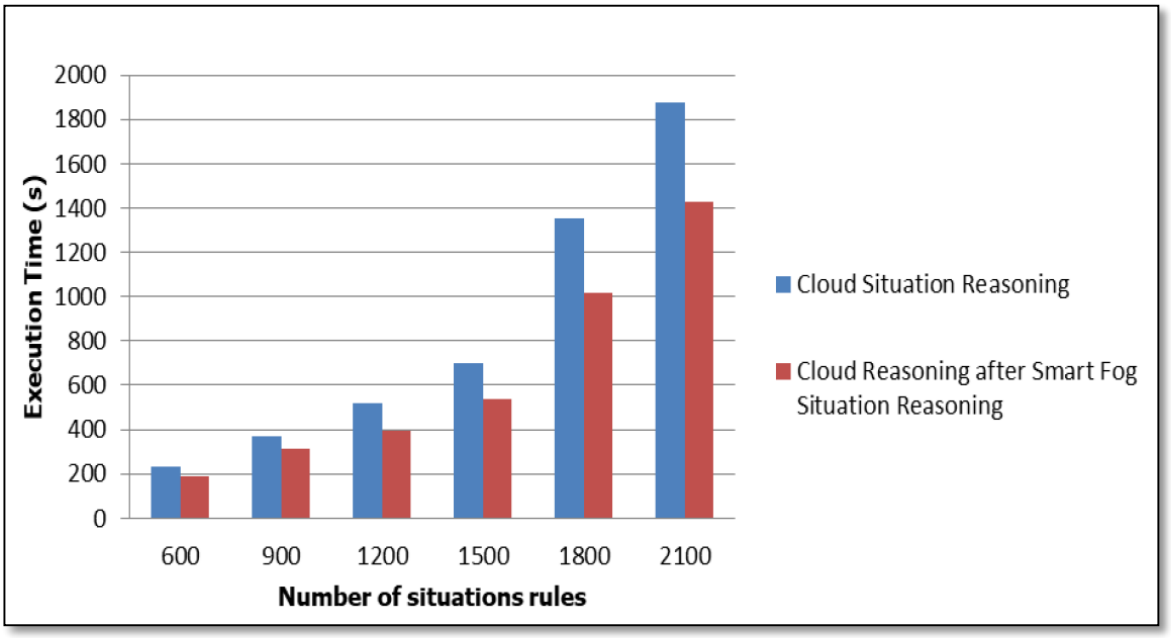

Figure 4.7 Cloud Situation Raisonnement vs Fog intelligent Raisonnement de la situation [114]

À partir de cette figure, on peut en conclure que, pour un rapport de transfert de données plus élevé avec une contrainte de bande passante réduite, le raisonnement en nuage avec des archives de brouillard intelligentes donne de meilleurs résultats que le seul raisonnement en situation de nuage. Cependant, le temps d'identification est important (1800ms) avec un nombre élevé de contraintes (2100 contraintes). Cela signifie que le transfert de données est un facteur clé dans l'évaluation du temps de réponse. La technique proposée donne une meilleure efficacité que le raisonnement en situation de nuage, mais reste relativement rare et doit être améliorée dans les travaux futurs. Notons également que l'approche proposée facilite l'extensibilité. Nous pouvons ajouter un nouveau profil de patient en enregistrant une nouvelle description de ce profil dans l'ontologie. De même, lorsque nous supprimons un profil sur le serveur local, nous ne supprimons que la description de ce profil dans l'ontologie. L'approche proposée peut être réutilisée sur d'autres domaines (véhicule intelligent, ville intelligente, etc.) pour la gestion de la qualité des situations d'urgence.

7. Conclusion

De nos jours, la conscience du contexte est devenue un terme de plus en plus courant dans le domaine de l'intelligence ambiante. Ce concept est répandu et appliqué dans une variété de domaines. Et pourtant, développer des applications sensibles au contexte reste une tâche ardue. Dans ce chapitre, l'architecture sensible au contexte à deux couches est proposée comme solution efficace pour la gestion de situation dans des environnements intelligents. Alors que les ressources contextuelles (capteurs physiques et services virtuels) deviennent de plus en plus diverses et complexes, la facilité de gestion est souvent une préoccupation. L'approche proposée consiste en un serveur local (Fog Computing) et un Cloud central assurant une gestion efficace de la situation et une bonne évolutivité. Notre travail est divisé en trois phases : regroupement de profils, gestion de la situation et diffusion d'informations multimédias liées aux services intelligents. Cette approche est nécessaire pour regrouper certaines situations et certains services afin que les utilisateurs puissent partager des ressources de calcul et des ressources en bande passante et accélérer la sélection de services spécifiques pertinents.

Conclusion Générale et Perspectives

1. Conclusion générale

Au fur et à mesure que nous progressons vers l'intelligence ambiante, le nombre de capteurs déployés dans le monde augmente rapidement. Les études de marché ont montré une croissance significative du nombre de déploiements de capteurs au cours de la dernière décennie et ont prédit une augmentation significative du taux de croissance à l'avenir. Ces capteurs génèrent en permanence d'énormes quantités de données. Cependant, pour ajouter de la valeur aux données brutes de ces capteurs, nous devons les comprendre. La collecte, la modélisation, le raisonnement et la distribution du contexte en relation avec les données de capteurs jouent un rôle crucial dans ce défi. Nous avons assisté au développement de services et d'applications dédiés à l'amélioration de l'état et du bien-être de ces personnes dans leurs vies quotidiennes. Dans les applications de téléassistance et de télémédecine, L'intelligence ambiante a permis d'introduire plus de confort et d'apporter plus d'autonomie à ces personnes dépendantes. Les données collectées dans ces espaces de vie sont exploitées par les techniques d'apprentissage, de modélisation, de raisonnement pour fournir des services sensibles au contexte. On constate une certaine complexité lors de la gestion de ces données contextuelles mais cela fournit des solutions intelligentes, flexibles et plus adaptées aux besoins des personnes.

Le but de cette recherche est de proposer une architecture système facilitant le développement d'applications contextuelles. Les principales contributions à la recherche sont présentées dans trois catégories : Modélisation unifiée des profils sémantique pour les environnements intelligents, une nouvelle architecture à deux couches, sensible au contexte, comprenant un serveur local (Fog Computing), Cloud central fournissant une couche de gestion de contexte. Il garantit une gestion efficace de la situation et une bonne évolutivité. Notre travail est divisé en trois phases : regroupement de profils, gestion de la situation et diffusion d'informations multimédias liées aux services intelligents.

En ce qui concerne la représentation de profil sémantique unifiée, un modèle de représentation de contexte est proposé pour notre architecture sensible au contexte via une ontologie. Nous avons développé notre ontologie, en utilisant une combinaison de l'approche descendante. Premièrement, nous avons défini les concepts importants dans le domaine de l'environnement intelligent à partir d'ontologies existantes : entité de contexte, situation et service. Deuxièmement, nous avons

généralisés et spécialisés. Pour chaque concept, nous avons créé une taxonomie de concepts afin de construire la hiérarchie de GUSP-Onto. La couche supérieure traite du contexte commun générique, composé de classes de base interdépendantes, tandis que la couche inférieure est utilisée pour compléter et mettre à niveau la couche supérieure à l'aide d'informations plus précises relatives à une application particulière ou à un domaine d'application. Contrairement aux modèles existants, outre la prise en compte des facteurs de localisation, notre modèle de contexte propose également des informations contextuelles tout au long des périodes. Ces contextes historiques peuvent être utilisés pour prédire et inférer le contexte. En outre, l'approche de modélisation basée sur une ontologie facilite le développement du système sensible au contexte, tel que le partage des connaissances, la réutilisation des connaissances et l'inférence logique.

Pour mettre en œuvre cette approche plus approfondie des services sensibles au contexte dans AmI, nous avons conçu une architecture sensible au contexte basé sur un profil sémantique unifié d'environnement intelligent, organisé en deux couches. La couche inférieure basée sur le Fog Computing intelligent est une couche de base permettant de déterminer rapidement de manière dynamique la reconfiguration au moment de l'exécution sur la prise en charge du déploiement pour les situations urgentes. Il fournit un regroupement de profils, un traitement de contexte, une identification de situation urgente et une diffusion d'informations multimédia liées à des services intelligents. La couche supérieure fonctionne sur une infrastructure de Cloud qui doit évoluer horizontalement pour prendre en charge le plus grand nombre d'objets intelligents connectés, ainsi que verticalement pour traiter la diversité des situations des utilisateurs dans les domaines intelligents. Les composants principaux de cette couche comprennent trois sous-composants : (i) clustering de contexte basé sur le Cloud, (ii) gestionnaire de contexte basé sur le Cloud et (iii) Gestionnaire de services basé sur le Cloud.

2. Perspective

Ce travail de thèse soulève de nouvelles questions de recherche. Il ouvre la voie à de nouvelles perspectives que nous considérons intéressantes. Nos travaux futurs ont pour objectif d'étudier la combinaison multi-domaines de l'informatique ubiquitaire et de la recherche Big Data basée sur notre système via l'apprentissage automatique. Le «Big Data» est actuellement l'un des termes les plus populaires en informatique, considéré comme la prochaine frontière de l'innovation. Dans une certaine mesure, Big DATA se développe grâce à l'informatique ubiquitaire. Les progrès dans

les capteurs numériques, les communications, le calcul et le stockage ont créé de vastes collections de données, capturant des informations provenant de la science et de la société [180].

La montée en puissance du Big Data exige de plus en plus que nous puissions accéder aux ressources de données à tout moment et en tout lieu. Ces énormes volumes de données proviennent de multiples ressources hétérogènes, sous diverses formes, notamment de diversité des capteurs, et même des milliards de réseaux sociaux. Avant de tirer pleinement parti des bénéfices du big data, de nombreux problèmes doivent être résolus, tels que l'accès aux données, leur fusion et leur intégration. Découvrir des connaissances spécifiques à l'utilisateur à partir des corrélations entre différents contextes dans des données antérieures et les utiliser pour prédire des anomalies futures. Détecter les changements de comportement à long terme à l'aide des historiques de contexte et de leurs relations réciproques. Rendre les modèles d'apprentissage adaptatifs et continus afin qu'ils puissent s'adapter en fonction des changements de contextes tout en permettant de prédire en permanence les événements futurs avec une bonne précision.

Travaux scientifiques

Publication

- Mounir, A., Adel, A., Makhlouf, D., Sébastien, L., & Philippe, R. (2019). A New Two-Level Clustering Approach for Situations Management in Distributed Smart Environments. International Journal of Ambient Computing and Intelligence (IJACI), 10(2), 91-111.
- Achouri, M., Alti, A., Derdour, M., Laborie, S., & Roose, P. (2018). Smart fog computing for efficient situations management in smart health environments. Journal of ICT, 17(4), 537-567.

Communication

- Achouri, M., Alti, A., Derdour, M., Laborie, S., & Roose, P. (2017, November). A new two-Layered architecture for efficient situations management in smart environments. In *Proceedings of the 9th international conference on management of digital EcoSystems* (pp. 6-13).
- Achouri, M., Alti, A., Derdour, M., Laborie, S., & Roose, P. (2017, November). A new two-Layered architecture for efficient situations management in smart environments. In Proceedings of the 9th international conference on management of digital EcoSystems (pp. 6-13).
- Achouri, M., Alti, A., Derdour, M., Laborie, S., & Roose, P. (2017, November). A new two-Layered architecture for efficient situations management in smart environments. In Proceedings of the 9th international conference on management of digital EcoSystems (pp. 6-13).
- Achouri, M., Alti, A., Derdour, M., Laborie, S., & Roose, P. (2017, November). A new two-Layered architecture for efficient situations management in smart environments. In Proceedings of the 9th international conference on management of digital EcoSystems (pp. 6-13).

Référence

[1] K. Henricksen, J. Indulska, Developing context-aware pervasive Computing applications: models and approach, Pervasive and Mobile Computing 2 (1) (2006) 37–64.

[2] D.J. Cook, S.K. Das, How smart are our environments? An updated look at the state of the art, Pervasive Mobile Computing 3 (2) (2007) 53–73.

[3]Birnbaum, J. (1997). Pervasive information systems. Communications of the ACM, 40(2), 40-42.

[4] M. Weiser. “The computer for the 21st century”, Scientific American, 1991.

[5]Weiser, M. (1993). Some computer science issues in ubiquitous Computing. Communications of the ACM, 36(7), 75-84.

[6]Saha, D., & Mukherjee, A. (2003). Pervasive Computing: a paradigm for the 21st century. Computer, (3), 25-31.

[7]Friedewald, M., & Raabe, O. (2011). Ubiquitous Computing: An overview of technology impacts. Telematics and Informatics, 28(2), 55-65.

[8]Olifer, N., & Olifer, V. (2006). Computer networks: Principles, technologies and protocols for network design. New York, NY, USA: John Wiley & Sons.

[9]Guinard, D., & Trifa, V. (2009, April). Towards the web of things: Web mashups for embedded devices. In Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain (Vol. 15).

[10]Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware Computing for the internet of things: A survey. IEEE communications surveys & tutorials, 16(1), 414-454.

[11]Bassi, A., & Horn, G. (2008). Internet of Things in 2020: A Roadmap for the Future. European Commission: Information Society and Media, 22, 97-114.

[12]Institutes, C., & Alliance, I. M. C. (2011). Smart networked objects and internet of things. Carnot Institutes' Information Communication Technologies and Micro Nano Technologies alliance, White Paper.

[13]Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. Computer networks, 54(15), 2787-2805.

[14]Kortuem, G., Kawsar, F., Sundramoorthy, V., & Fitton, D. (2010). Smart objects as building blocks for the internet of things. IEEE Internet Computing, 14(1), 44-51.

[15]Le-Phuoc, D., Polleres, A., Hauswirth, M., Tummarello, G., & Morbidoni, C. (2009, April). Rapid prototyping of semantic mash-ups through semantic web pipes. In Proceedings of the 18th international conference on World wide web (pp. 581-590). ACM.

[16]Dohr, A., Modre-Opsrian, R., Drobics, M., Hayn, D., & Schreier, G. (2010, April). The internet of things for ambient assisted living. In Information technology: new generations (ITNG), 2010 seventh international conference on (pp. 804-809). IEEE.

[17]Ashton, K. (2009). That 'internet of things' thing in the real world, things matter more than ideas. RFID J.

[18]Brock, D. The Electronic Product Code-A Naming Scheme for physical Objects, Auto-ID White Paper (January 2001).

[19]Strategy, I. T. U., & Unit, P. (2005). ITU Internet Reports 2005: The internet of things. Geneva: International Telecommunication Union (ITU), 1, 62.

[20]Tan, L., & Wang, N. (2010, August). Future internet: The internet of things. In Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on (Vol. 5, pp. V5-376). IEEE.

[21]Youngblood, G. M., Heierman, E. O., Holder, L. B., & Cook, D. J. (2005, July). Automation intelligence for the smart environment. In International Joint Conference on Artificial Intelligence (Vol. 19, p. 1513). LAWRENCE ERLBAUM ASSOCIATES LTD.

[22] Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., & Burgelman, J. C. (2003). Ambient intelligence: From vision to reality. IST Advisory Group Draft Report, European Commission.

[23] IA Group. (2001). Scenarios for ambient intelligence in 2010. Technical report, IST.

[24] Cook, D. J., Augusto, J. C., & Jakkula, V. R. (2009). Ambient intelligence: Technologies, applications, and opportunities. Pervasive and Mobile Computing, 5(4), 277-298.

[25] Aarts, E. H., & Encarnação, J. L. (Eds.). (2006). True visions: The emergence of ambient intelligence. Springer Science & Business Media.

[26] Cook, D. J., Augusto, J. C., & Jakkula, V. R. (2009). Ambient intelligence: Technologies, applications, and opportunities. Pervasive and Mobile Computing, 5(4), 277-298.

[27] Augusto, J. C. (2007). Ambient intelligence: the confluence of ubiquitous/pervasive Computing and artificial intelligence. In Intelligent Computing everywhere (pp. 213-234). Springer, London.

[28] ISTAG, S. O. (2002). Priorities for IST in FP6. European Commission Report.

[29] Laghari, A. A., He, H., Halepoto, I. A., Memon, M. S., & Parveen, S. (2017). Analysis of quality of experience frameworks for Cloud Computing. IJCSNS, 17(12), 228.

[30] Hashemi, S. M., & Bardsiri, A. K. (2012). Cloud Computing vs. grid Computing. ARPN journal of systems and software, 2(5), 188-194.

[31] Mell, P., & Grance, T. (2011). The NIST definition of Cloud Computing.

[32] Laghari, A. A., He, H., Shafiq, M., & Khan, A. (2016, October). Assessing effect of Cloud distance on end user's Quality of Experience (QoE). In 2016 2nd IEEE international conference on computer and communications (ICCC) (pp. 500-505). IEEE.

[33] Dempsey, D., & Kelliher, F. (2018). Revenue Models and Pricing Strategies in the B2B SaaS Market. In Industry Trends in Cloud Computing (pp. 45-82). Palgrave Macmillan, Cham.

[34] Krancher, O., Luther, P., & Jost, M. (2018). Key affordances of platform-as-a-service: Self-organization and continuous feedback. Journal of Management Information Systems, 35(3), 776-812.

[35] Kashif, U. A., Memon, Z. A., Siddiqui, S., Balouch, A. R., & Batra, R. (2019). Architectural design of trusted platform for IaaS Cloud Computing. In Cloud Security: Concepts, Methodologies, Tools, and Applications (pp. 393-411). IGI Global.

[36] Briggs, Benjamin David, Lawrence A. Clevengen, Bartlet H. DeProspo, and Michael Rizzolo. "Structure, system, method, and recording medium of implementing a directed self-assembled security pattern." U.S. Patent Application 15/055,835, filed August 31, 2017.

[37] Hu, H., Wen, Y., & Niyato, D. (2017). Public Cloud storage-assisted mobile social video sharing: A supermodular game approach. IEEE journal on selected areas in communications, 35(3), 545-556.

[38] Manu, A. R., Agrawal, V. K., & Murthy, K. B. S. (2017, January). An empirical hunt for ally co-operative Cloud Computing utility. In 2017 11th International Conference on Intelligent Systems and Control (ISCO) (pp. 422-438). IEEE.

[39] Goyal, S. (2014). Public vs private vs hybrid vs community-Cloud Computing: a critical review. International Journal of Computer Network and Information Security, 6(3), 20.

[40] Ramanathan, R., & Latha, B. (2018). Towards optimal resource provisioning for Hadoop-MapReduce jobs using scale-out strategy and its performance analysis in private Cloud environment. Cluster Computing, 1-11.

[50] Saharan, K. P., & Kumar, A. (2015). Fog in comparison to Cloud: A survey. International Journal of Computer Applications, 122(3).

[51] Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R. H., Morrow, M. J., & Polakos, P. A. (2017). A comprehensive survey on fog Computing: State-of-the-art and research challenges. IEEE Communications Surveys & Tutorials, 20(1), 416-464.

[52] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog Computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile Cloud Computing (pp. 13-16). ACM.

[53] Bonomi, F., Milito, R., Natarajan, P., & Zhu, J. (2014). Fog Computing: A platform for internet of things and analytics. In Big data and internet of things: A roadmap for smart environments (pp. 169-186). Springer, Cham.

[54] Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited.

[55] M. Ermes, J. Parkka, J. Mantyjarvi, I. Korhonen I. (2008). Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. IEEE transactions on information technology in biomedicine, 12(1), 20-26.

[56] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. IEEE Communications magazine, 40(8), 102-114.

[57] Benini, L., & Poncino, M. (2003). Ambient intelligence : A computational platform perspective. In Ambient intelligence: impact on embedded sytem design (pp. 31-50). Springer, Boston, MA.

[58] Jayasimha, D. N., Iyengar, S. S., & Kashyap, R. L. (1991). Information integration and synchronization in distributed sensor networks. IEEE Transactions on Systems, Man, and Cybernetics, 21(5), 1032-1043.

[59] Youngblood, G. M. (2005). Automating inhabitant interactions in home and workplace environments through data-driven generation of hierarchical partially-observable Markov decision processes (Doctoral dissertation, The University of Texas at Arlington).

[60] Loke, S. W. (2004). Representing and reasoning with situations for context-aware pervasive Computing: a logic programming perspective. The Knowledge Engineering Review, 19(3), 213-233.

[61] Doctor, F., Hagras, H., & Callaghan, V. (2004). A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 35(1), 55-65.

[62] Lühr, S., West, G., & Venkatesh, S. (2007). Recognition of emergent human behaviour in a smart home: A data mining approach. Pervasive and Mobile Computing, 3(2), 95-116.

[63] Zancanaro, M., Lepri, B., & Pianesi, F. (2006, November). Automatic detection of group functional roles in face to face interactions. In Proceedings of the 8th international conference on Multimodal interfaces (pp. 28-34). ACM.

[64] Moncrieff, S., Venkatesh, S., West, G., & Greenhill, S. (2007). Multi-modal emotive Computing in a smart house environment. Pervasive and Mobile Computing, 3(2), 74-94.

[65] Laibowitz, M., Gips, J., AyIward, R., Pentland, A., & Paradiso, J. A. (2006, April). A sensor network for social dynamics. In 2006 5th International Conference on Information Processing in Sensor Networks (pp. 483-491). IEEE.

[66] Cook, D. J., & Das, S. K. (2007). How smart are our environments? An updated look at the state of the art. Pervasive and mobile Computing, 3(2), 53-73.

[67] Cook, D., & Das, S. K. (2004). Smart environments: technology, protocols, and applications (Vol. 43). John Wiley & Sons.

[68] Lesser, V., Atighetchi, M., Benyo, B., Horling, B., Raja, A., Vincent, R., ... & Zhang, S. X. (1999). The intelligent home testbed. environment, 2, 15.

[69] Das, S. K., Cook, D. J., Battacharya, A., Heierman, E. O., & Lin, T. Y. (2002). The role of prediction algorithms in the MavHome smart home architecture. IEEE Wireless Communications, 9(6), 77-84.

[70] Youngblood, G. M., Cook, D. J., & Holder, L. B. (2005). Managing adaptive versatile environments. Pervasive and Mobile Computing, 1(4), 373-403.

[71] Helal, S., Winkler, B., Lee, C., Kaddoura, Y., Ran, L., Giraldo, C., ... & Mann, W. (2003, March). Enabling location-aware pervasive Computing applications for the elderly. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003). (pp. 531-536). IEEE.

[72] Turkle, S. (2004). Whither psychoanalysis in computer culture?. Psychoanalytic psychology, 21(1), 16.

[73] Pineau, J., Montemerlo, M., Pollack, M., Roy, N., & Thrun, S. (2003). Towards robotic assistants in nursing homes: Challenges and results. Robotics and autonomous systems, 42(3-4), 271-281.

[74] Breazeal, C., & Scassellati, B. (1999). A context-dependent attention system for a social robot. rn, 255, 3.

[75] Carroll, J. M. (2003). Introduction: Toward a multidisciplinary science of human-computer interaction. In HCI models, theories, and frameworks: Toward a multidisciplinary science (pp. 1-9). Elsevier Inc.

[76] Abowd, G. D., & Mynatt, E. D. (2004). Designing for the human experience in smart environments. Smart environments: technologies, protocols, and applications, 151-174.

[77] Ge, S. S., Yang, Y., & Lee, T. H. (2008). Hand gesture recognition and tracking based on distributed locally linear embedding. Image and Vision Computing, 26(12), 1607-1620.

[78] Pantic, M. (2006). Face for ambient interface. In Ambient Intelligence in Everyday Life (pp. 32-66). Springer, Berlin, Heidelberg.

[79] Nakatsu, R. (2002, August). Integration of multimedia and art for new human-computer communications. In Pacific Rim International Conference on Artificial Intelligence (pp. 19-28). Springer, Berlin, Heidelberg.

[80] Aghajan, H., Augusto, J. C., & Delgado, R. L. C. (Eds.). (2009). Human-centric interfaces for ambient intelligence. Academic Press.

[81] Ward, A. M. R. (1998). Sensor driven Computing (Doctoral dissertation, University of Cambridge).

[82] Hazas, M., & Hopper, A. (2006). Broadband ultrasonic location systems for improved indoor positioning. IEEE Transactions on mobile Computing, 5(5), 536-547.

[83] Lanspery, S., & Hyde, J. (2018). Staying put: Adapting the places instead of the people. Routledge.

[84] Hebert, L. F., Scherr, P. A., Bienial, J. L., Bennell, D. A., & Evans, D. A. (2004). Alzheimer's disease in the US population: Prevalence estimates using the 2000 census.

[85] Larson, C. (2007). Infielder care, signing on becomes a way to drop by. The New York Times.

[86] Ernst, R. L., & Hay, J. W. (1994). The US economic and social costs of Alzheimer's disease revisited. American Journal of Public Health, 84(8), 1261-1264.

[87] Sánchez, D., Tentori, M., & Favela, J. (2008). Activity recognition for the smart hospital. IEEE intelligent systems, 23(2), 50-57.

[88] Rakotonirainy, A., & Tay, R. (2004, October). In-vehicle ambient intelligent transport systems (i-vaits): Towards an integrated research. In Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749) (pp. 648-651). IEEE.

[89] Dashtinezhad, S., Nadeem, T., Dorohonceanu, B., Borcea, C., Kang, P., & Iftode, L. (2004, May). TrafficView: a driver assistant device for traffic monitoring based on car-to-car communication. In 2004 IEEE 59th Vehicular Technology Conference. VTC 2004-Spring (IEEE Cat. No. 04CH37514) (Vol. 5, pp. 2946-2950). IEEE.

[90] Velastin, S. A., Boghossian, B. A., Lo, B. P., Sun, J., & Vicencio-Silva, M. A. (2004). PRISMATICA: toward ambient intelligence in public transport environments. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 35(1), 164-182.

[91] Abowd, G. D. (1999). Classroom 2000: An experiment with the instrumentation of a living educational environment. IBM systems journal, 38(4), 508-530.

[92] Shi, Y., Xie, W., Xu, G., Shi, R., Chen, E., Mao, Y., & Liu, F. (2003). The smart classroom: merging technologies for seamless tele-education. IEEE Pervasive Computing, (2), 47-55.

[93] Stöttinger, M. (2004). Context-Awareness in industrial environments. Software Engineering. Hagenberg, FH Hagenber, 68.

[94] Shen, W., Lang, S. Y., & Wang, L. (2005). iShopFloor: an Internet-enabled agent-based intelligent shop floor. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 35(3), 371-381.

[95] Sundmaeker, H., Guillemin, P., Friess, P., & Woelfflé, S. (2010). Vision and challenges for realising the Internet of Things. Cluster of European Research Projects on the Internet of Things, European Commision, 3(3), 34-36.

[96] 1. Schilit, B. N., & Theimer, M. M. (1994). Disseminating Active Mop Infonncition to Mobile Hosts. IEEE network.

[97] 2. Brown, P. J., Bovey, J. D., & Chen, X. (1997). Context-aware applications: from the laboratory to the marketplace. IEEE personal communications, 4(5), 58-64.

[98] 3. Ryan, N. S., Pascoe, J., & Morse, D. R. (1998). Enhanced reality fieldwork: the context-aware archaeological assistant. In Computer applications in archaeology. Tempus Reparatum.

[99]4. Abowd, G. D., & Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous Computing. ACM Transactions on Computer-Human Interaction (TOCHI), 7(1), 29-58.

[100]5. Dey, A. K., Abowd, G. D., & Wood, A. (1998). CyberDesk: A framework for providing self-integrating context-aware services. Knowledge-based systems, 11(1), 3-13.

[101]6. Zimmermann, A., Lorenz, A., & Oppermann, R. (2007, August). An operational definition of context. In International and Interdisciplinary Conference on Modeling and Using Context (pp. 558-571). Springer, Berlin, Heidelberg.

[102] Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing, 2(4), 263-277.

[102]7. Franklin, D., & Flaschbart, J. (1998, March). All gadget and no representation makes jack a dull environment. In Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments (pp. 155-160).

[103]8. Hull, R., Neaves, P., & Bedford-Roberts, J. (1997). Towards situated Computing (pp. 146-153). Hewlett Packard Laboratories.

[104]9. Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human–Computer Interaction, 16(2-4), 97-166.

[105]10. Rodden, T., Cheverst, K., Davies, K., & Dix, A. (1998, May). Exploiting context in HCI design for mobile systems. In Workshop on human computer interaction with mobile devices (pp. 21-22).

[106]11. Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999, September). Towards a better understanding of context and context-awareness. In International symposium on handheld and ubiquitous Computing (pp. 304-307). Springer, Berlin, Heidelberg.

[107]12. Lieberman, H., & Selker, T. (2000). Out of context: Computer systems that adapt to, and learn from, context. IBM systems journal, 39(3.4), 617-632.

[108]13. Dourish, P. (2004). What we talk about when we talk about context. Personal and ubiquitous Computing, 8(1), 19-30.

[109] Dey, A. K. (2001). Understanding and using context. Personal and ubiquitous Computing, 5(1), 4-7.

[110] Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing, 2(4), 263-277.

[111] Kirsch-Pinheiro, M., Villanova-Oliver, M., Gensel, J., & Martin, H. (2005, March). Context-aware filtering for collaborative web systems: adapting the awareness information to the user's context. In Proceedings of the 2005 ACM symposium on Applied Computing (pp. 1668-1673). ACM.

[112] Miraoui, M. (2009). Architecture logicielle pour l'informatique diffuse: modélisation du contexte et adaptation dynamique des services (Doctoral dissertation, École de technologie supérieure).

[113] Bernardos, A. M., Tarrio, P., & Casar, J. R. (2008, August). A data fusion framework for context-aware mobile services. In 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (pp. 606-613). IEEE.

[114] Achouri, M. (2020). Smart fog computing for efficient situations management in smart health environments. Journal of Information and Communication Technology, 17(4), 537-567.

[115] Indulska, J., & Sutton, P. (2003, January). Location management in pervasive systems. In Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21 (pp. 143-151). Australian Computer Society, Inc.

[116] Schmidt, A., & Van Laerhoven, K. (2001). How to build smart appliances?. IEEE Personal Communications, 8(4), 66-71.

[117] Brézillon, P. (2002). Expliciter le contexte dans les objets communicants. In Objets Communicants, C. Kintzig, G. Poulain, G. Privat, et P.N. Favennec, eds. (Hermes Science Publications), pp. 295–303.

[118] Najar, S., Saidani, O., Kirsch-Pinheiro, M., Souveyet, C., & Nurcan, S. (2009, June). Semantic representation of context models: a framework for analyzing and understanding. In Proceedings of the first Workshop on Context, Information and Ontologies (p. 6). ACM.

[119] Strang, T., & Linnhoff-Popien, C. (2004, September). A context modeling survey. In Workshop Proceedings.

[120] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing, 6(2), 161-180.

[121] Bikakis, A., Patkos, T., Antoniou, G., & Plexousakis, D. (2007, November). A survey of semantics-based approaches for context reasoning in ambient intelligence. In European Conference on Ambient Intelligence (pp. 14-23). Springer, Berlin, Heidelberg.

[122] Guan, D., Yuan, W., Lee, S., & Lee, Y. K. (2007, October). Context selection and reasoning in ubiquitous Computing. In The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007) (pp. 184-187). IEEE.

[123] Perttunen, M., Riekki, J., & Lassila, O. (2009). Context representation and reasoning in pervasive Computing: a review. International Journal of Multimedia and Ubiquitous Engineering, 4(4), 1-28.

[124] Nurmi, P., & Floréen, P. (2004). Reasoning in context-aware systems. Helsinki Institute for Information Technology, Position paper.

[125] Hall, D. L., & Llinas, J. (1997). An introduction to multisensor data fusion. Proceedings of the IEEE, 85(1), 6-23.

[126] Castelli, G., Mamei, M., Rosi, A., & Zambonelli, F. (2009). Extracting high-level information from location data: the w4 diary example. Mobile Networks and Applications, 14(1), 107-119.

[127] Da, K., Dalmau, M., & Roose, P. (2011). A Survey of adaptation systems. International Journal on Internet and Distributed Computing Systems, 2(1), 1-18.

[128] Cheng, B. H., de Lemos, R., Giese, H., Inverardi, P., Magee, J., Andersson, J., ... & Serugendo, G. D. M. (2009). Software engineering for self-adaptive systems: A research roadmap. In Software engineering for self-adaptive systems (pp. 1-26). Springer, Berlin, Heidelberg.

[129] Dobson, S., Denazis, S., Fernández, A., Gaïti, D., Gelenbe, E., Massacci, F., ... & Zambonelli, F. (2006). A survey of autonomic communications. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 1(2), 223-259.

[130] Kakousis, K., Paspallis, N., & Papadopoulos, G. A. (2010). A survey of software adaptation in mobile and ubiquitous Computing. Enterprise Information Systems, 4(4), 355-389.

[131] Chen, G., & Kotz, D. (2000). A survey of context-aware mobile Computing research. Dartmouth Computer Science Technical Report TR2000-381.

[132] Strang, T., & Linnhoff-Popien, C. (2004, September). A context modeling survey. In Workshop Proceedings.

[133] Klyne, G. R. A. H. A. M., Reynolds, F. R. A. N. K. L. I. N., Woodrow, C. H. R. I. S., Ohto, H., Hjelm, J. O. H. A. N., Butler, M., & Tran, L. (2004). Composite capabilities/preference profiles: Structure and vocabularies. Technical report W3C Ubiquitous Web Application Working Group.

[134] Knappmeyer, M., Kiani, S. L., Frà, C., Moltchanov, B., & Baker, N. (2010, May). Contextml: A light-weight context representation and context management schema. In IEEE 5th International Symposium on Wireless Pervasive Computing 2010 (pp. 367-372). IEEE.

[135] Yanwei, S., Guangzhou, Z., & Haitao, P. (2011, December). Research on the context model of intelligent interaction system in the internet of things. In 2011 IEEE International Symposium on IT in Medicine and Education (Vol. 2, pp. 379-382). IEEE.

[136] Rumbaugh, J., Jacobson, I., & Booch, G. (2004). Unified modeling language reference manual, the. Pearson Higher Education.

[137] Halpin, T. (1998). Object-role modeling (ORM/NIAM). In Handbook on architectures of information systems (pp. 81-103). Springer, Berlin, Heidelberg.

[138] Henricksen, K. (2003). A framework for context-aware pervasive Computing applications. Queensland: University of Queensland.

[139] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. Scientific american, 284(5), 28-37.

[140] Hitzler, P., Krotzsch, M., & Rudolph, S. (2009). Foundations of semantic web technologies. Chapman and Hall/CRC.

[141] Brun, Y., Serugendo, G. D. M., Gacek, C., Giese, H., Kienle, H., Litoiu, M., ... & Shaw, M. (2009). Engineering self-adaptive systems through feedback loops. In Software engineering for self-adaptive systems (pp. 48-70). Springer, Berlin, Heidelberg.

[142] Manola, F., Miller, E., & McBride, B. (2004). RDF primer. W3C recommendation, 10(1-107), 6.

[143] McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. W3C recommendation, 10(10), 2004.

[144] Horrocks, I. (2002). DAML+OIL: A Description Logic for the Semantic Web. IEEE Data Eng. Bull., 25(1), 4-9.

[145] Klyne, G. (2004). Resource description framework (RDF): Concepts and abstract syntax. http://www. w3. org/TR/2004/REC-rdf-concepts-20040210/.

[146] Hyvönen, E. (2002). Semantic web kick-off in finland: Vision, technologies, research, and applications. HIIT Publications 2002-01.

[147] Fu, Z., Yue, J., & Li, Z. (2009). Ontology and Its Application in Supply Chain Information Management. In Supply Chain the Way to Flat Organisation. IntechOpen.

[148] Neches, R., Fikes, R. E., Finin, T., Gruber, T., Patil, R., Senator, T., & Swartout, W. R. (1991). Enabling technology for knowledge sharing. AI magazine, 12(3), 36-36.

[149] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge acquisition, 5(2), 199-220.

[150] Borst, W. N. (1999). Construction of engineering ontologies for knowledge sharing and reuse.

[151] Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. Data & knowledge engineering, 25(1-2), 161-197.

[152] Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing, 2(4), 263-277.

[153] Strang, T., & Linnhoff-Popien, C. (2004, September). A context modeling survey. In Workshop Proceedings.

[154] Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M. H., & Tran, L. (2004). Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation, W3C, January 2004.

[155] Butler, M. H. (2002, March). CC/PP and UAProf: Issues, improvements and future directions. In Proceedings of W3C Delivery Context Workshop (DIWS 2002).

[156] Buchholz, S., Hamann, T., & Hubsch, G. (2004, March). Comprehensive structured context profiles (cscp): Design and experiences. In IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second (pp. 43-47). IEEE.

[157] Indulska, J., Robinson, R., Rakotonirainy, A., & Henricksen, K. (2003, January). Experiences in using cc/pp in context-aware systems. In International Conference on Mobile Data Management (pp. 247-261). Springer, Berlin, Heidelberg.

[158] Saleemi, M. M., Rodríguez, N. D., Lilius, J., & Porres, I. (2011). A framework for context-aware applications for smart spaces. In Smart Spaces and Next Generation Wired/Wireless Networking (pp. 14-25). Springer, Berlin, Heidelberg.

[159] Dromzée, C., Laborie, S., & Roose, P. (2013). A Semantic Generic Profile for Multimedia Document Adaptation. In Intelligent Multimedia technologies for networking applications: techniques and tools (pp. 225-246). IGI Global.

[160] Yus, R., Mena, E., Ilarri, S., & Illarramendi, A. (2014). SHERLOCK: Semantic management of location-based services in wireless environments. Pervasive and Mobile Computing, 15, 87-99.

[161] Gyrard, A., Bonnet, C., Boudaoud, K., & Serrano, M. (2016, August). LOV4IoT: A second life for ontology-based domain knowledge to build Semantic Web of Things applications. In 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud) (pp. 254-261). IEEE.

[162] Chabridon, S., Bouzeghoub, A., Ahmed-Nacer, A., Marie, P., & Desprats, T. (2017, June). Unified modeling of quality of context and quality of situation for context-aware applications in the internet of things. In International and Interdisciplinary Conference on Modeling and Using Context (pp. 370-374). Springer, Cham.

[163] Tran, H. T., & Feuerlicht, G. (2016, September). Improving reliability of Cloud-based applications. In European Conference on Service-Oriented and Cloud Computing (pp. 235-247). Springer, Cham.

[164] Gu, T., Pung, H. K., & Zhang, D. Q. (2005). A service-oriented middleware for building context-aware services. Journal of Network and computer applications, 28(1), 1-18.

[165] Gu, T., Pung, H. K., & Yao, J. K. (2005). Towards a flexible service discovery. Journal of network and computer Applications, 28(3), 233-248.

[166] Naqvi, N. Z., Preuveneers, D., & Berbers, Y. (2014). A quality-aware federated framework for smart mobile applications in the Cloud. Procedia Computer Science, 32, 253-260.

[167] Pan, B., Wang, X., Song, E., Lai, C. F., & Chen, M. (2013, July). CAMSPF: Cloud-assisted mobile service provision framework supporting personalized user demands in pervasive Computing environment. In 2013 9th international wireless communications and mobile Computing conference (IWCMC) (pp. 649-654). IEEE.

[168] Forkan, A., Khalil, I., & Tari, Z. (2014). CoCaMAAL: A Cloud-oriented context-aware middleware in ambient assisted living. Future Generation Computer Systems, 35, 114-127.

[169] Aguilar, J., Jerez, M., Exposito, E., & Villemur, T. (2015, October). CARMiCLOC: context awareness middleware in Cloud Computing. In 2015 Latin American Computing Conference (CLEI) (pp. 1-10). IEEE.

[170] Aazam, M., & Huh, E. N. (2015, March). E-HAMC: Leveraging Fog Computing for emergency alert service. In 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops) (pp. 518-523). IEEE.

[171] Rouvoy, R., Barone, P., Ding, Y., Eliassen, F., Hallsteinsen, S., Lorenzo, J., ... & Scholz, U. (2009). Music: Middleware support for self-adaptation in ubiquitous and service-oriented environments. In Software engineering for self-adaptive systems (pp. 164-182). Springer, Berlin, Heidelberg.

[172] Da, K., Dalmau, M., & Roose, P. (2014, March). Kalimucho: middleware for mobile applications. In Proceedings of the 29th Annual ACM Symposium on Applied Computing (pp. 413-419). ACM.

[173] Alti, A., Lakehal, A., Laborie, S., & Roose, P. (2016). Autonomic semantic-based context-aware platform for mobile applications in pervasive environments. Future Internet, 8(4), 48.

[174] Knublauch, H., Fergerson, R. W., Noy, N. F., & Musen, M. A. (2004, November). The Protégé OWL plugin: An open development environment for semantic web applications. In International Semantic Web Conference (pp. 229-243). Springer, Berlin, Heidelberg.

[175] Jena, A. (2015). A free and open source Java framework for building Semantic Web and Linked Data applications. Available online: jena. apache. org/(accessed on 28 April 2015).

[176] Nakashima, H., Aghajan, H., & Augusto, J. C. (Eds.). (2009). Handbook of ambient intelligence and smart environments. Springer Science & Business Media.

[177] Anagnostopoulos, C., & Hadjiefthymiades, S. (2008). Enhancing situation-aware systems through imprecise reasoning. IEEE Transactions on Mobile Computing, 7(10), 1153-1168.

[178] Sulaiman, M. S., Nordin, S., & Jamil, N. (2017). An object properties filter for multi-modality ontology semantic image retrieval. Journal of Information and Communication Technologuy, 16(1), 1-19.

[179] Horridge, M., Tsarkov, D., & Redmond, T. (2006, November). Supporting Early Adoption of OWL 1.1 with Protege-OWL and FaCT++. In OWLED.

[180] Bryant, Randal, Randy H. Katz, and Edward D. Lazowska. "Big-data Computing: creating revolutionary breakthroughs in commerce, science and society." (2008).

[181] Lavrischeva, E. M. (2015). Ontology of Domains. Ontological Description Software Engineering Domain—the Standard Life Cycle. Journal of Software Engineering and Applications, 8(07), 324.