Une nouvelle approche de modélisation et d'adaptation contextuelle des systèmes pervasifs : la plateforme COALA (COntext Adaptation Platform)

Naima Nebhani

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UFC UNIVERSITÉ DE FRANCHE-COMTÉ

Une nouvelle approche de modélisation et d'adaptation contextuelle des systèmes pervasifs

La plateforme COALA (*COntext Adaptation Platform*)

Thèse en co-tutelle (Tunisie-France)

NAIMA NEBHANI

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE FRANCHE-COMTÉ

N° U P M - 2 0 1 6 - 0 1

THÈSE présentée par

NAIMA NEBHANI

pour obtenir le

Grade de Docteur de
l'Université de Franche-Comté

Spécialité : **Informatique**

Une nouvelle approche de modélisation et d'adaptation contextuelle des systèmes pervasifs

La plateforme COALA (*COntext Adaptation Platform*)

Thèse en co-tutelle (Tunisie-France)

Unité de Recherche :
Institut FEMTO-ST/DISC UMR CNRS 6174

Soutenue le 18 Novembre 2016 devant le Jury :

PHILIPPE ROOSE	Rapporteur	MCF HdR Univ. Pau et Pays de l'Adour - France
SAÏD TAZI	Rapporteur	MCF HdR Univ. de Toulouse - France
MOUNIR ZRIGUI	Rapporteur	Prof. Univ. de Mounastir - Tunisie
CHRISTOPHE NICOLLE	Examinateur	Prof. Univ. Bourgogne - France
J-CHRISTOPHE LAPAYRE	Directeur de thèse	Prof. Univ. Franche-Comté - France
M-SALIM BOUHLEL	Directeur de thèse	Prof. à l'Univ. Sfax - Tunisie

SOMMAIRE

REMERCIEMENTS

Je voudrais tout d'abord remercier mes encadrants, Monsieur BOUHLEL Mohamed Salim, Professeur à l 'Institut Supérieur de Biologie de Sfax (Université de Sfax), pour m'avoir proposé ce sujet de thèse original et m'avoir permis de l'approfondir au cours de ces années. Monsieur LAPAYRE jean-Christophe, Professeur à l'Université de Bourgogne Franche-Comté, qui a encadré ce présent travail en co-tutelle, pour sa disponibilité, ses qualités humaines, son aide, son soutien moral, sa patience, ses conseils et sa bonne connaissance qui a rendu cette thèse possible. Je désire lui exprimer ma profonde gratitude pour m'avoir accueillie dans son équipe.

Je tiens à remercier les Professeurs, Philippe ROOSE,Mounir ZRIGUI et Said TAZI qui m'ont fait l'honneur d'être rapporteurs de ma thèse. Pour leur lecture attentive et critique de ce manuscrit et l'intérêt porté à ces travaux.

Un immense merci également à Monsieur Christophe NICOLLE, professeur à la Faculté des Sciences Mirande (Université de Bourgogne), pour m'avoir fait l'honneur d'accepter de présider le jury de ma thèse et de juger mon travail.

À tous les membres du laboratoire Femto-st département DISC, ceux et celles qui m'ont accompagné au cours de l'élaboration de cette thèse, qu'ils trouvent ici mon témoignage de reconnaissance.

Merci à tous mes amis, mes proches pour leur aide en diverses occasions et de leur encouragement continu.

Finalement, je tiens à remercier toute ma famille et particulièrement « mon sublime PaPa et mon adorable MaMan » qui m'ont toujours encouragé au cours de mes études pour aller plus loin pour demander la noble science. je vous dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, longue vie et bonheur. Ma thèse n'était jamais facile mais la voilà possible.

À cœur vaillant rien d'impossible
À conscience tranquille tout est accessible

INTRODUCTION

INTRODUCTION GÉNÉRALE

Domaine très récent, l'informatique mobile est le fruit de l'intégration de la technologie cellulaire et du web pour donner la possibilité aux différents utilisateurs d'accéder facilement à l'information de n'importe quel endroit, à n'importe quel moment et en utilisant n'importe quel terminal. L'accroissement est en particulier dû à la réduction des coûts, qu'ils soient matériels ou de services réseaux. C'est en quelque sorte la deuxième étape vers l'informatique pervasive «*le n'importe où, le n'importe quand et le n'importe comment*» [Nar00], après l'apparition des systèmes distribués qui constituent la rencontre entre les ordinateurs personnels et les réseaux locaux favorisant le partage des ressources à travers un réseau et une infrastructure de communication (cf figure 1).

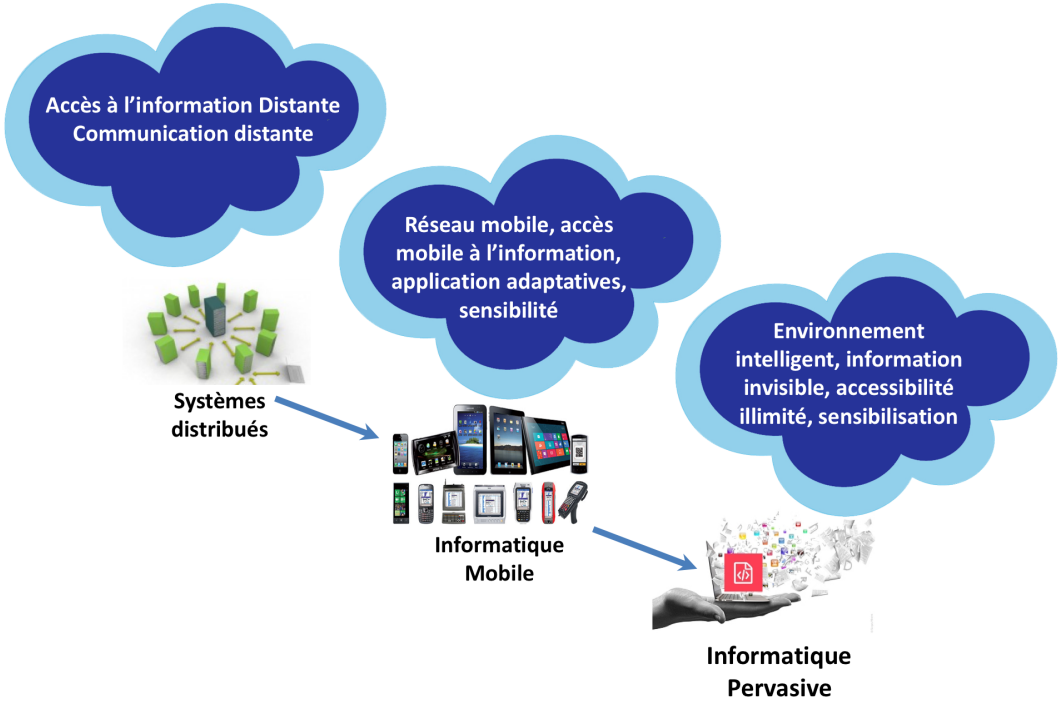

FIGURE 1 – Aperçu de l'évolution vers l'informatique pervasive

De nos jours, l'évolution technologique des moyens de communication informatique ne cesse de croître en donnant naissance à des nouveaux systèmes d'information dits pervasifs ou ubiquitaires [Birn97]. Dans une tendance mondiale où l'accès au savoir et à l'information est primordial, le passage massif à l'exploitation des dernières technologies de communication ainsi que leur intégration ont contribué à l'émergence de ces technologies dans différents domaines. Dans les années 90, Mark Weiser [Weise93] est

le premier à introduire la notion de l'informatique pervasive en décrivant l'ordinateur du 21ème siècle comme étant un terminal compagnon actif plus intelligent qu'un assistant de bureau comme un ordinateur standard.

Cette vision dévoile une nouvelle technologie considérée comme invisible à l'utilisateur qui combine les aspects de l'informatique distribuée et de l'informatique mobile. Elle adopte une nouvelle vision des équipements, des infrastructures et des applications. Les travaux [Agost00] ont donné une définition universelle des systèmes pervasifs, tout en reprenant les idées initiées par Weiser : *l'informatique pervasive rend l'information disponible partout et à tout moment et vise à assister implicitement et discrètement un utilisateur dans les tâches qu'il accomplit au quotidien*.

En effet, l'avènement des réseaux de communications sans fil d'une part et des terminaux informatiques mobiles, d'autre part, tels que les assistants personnels (PDA), les smartphones (et tablettes), ... a permis la mobilité et la coopération. Ainsi, les ordinateurs *de poche* ne sont plus isolés de la technologie du Web, ce que favorise les interactions avec de multiples services distants. L'avantage réside dans le fait que la réalisation de nombreuses tâches de la vie quotidienne s'effectue de manière transparente à l'utilisateur, sans son intervention explicite.

L'intégration des terminaux mobiles dans les nouveaux systèmes d'interaction pervasifs exige la prise en considération de toutes les caractéristiques logicielles et matérielles. L'un des aspects contribuant à la réalisation de la notion de l'informatique pervasive est de permettre aux systèmes informatiques de s'adapter aux divers changements du contexte des utilisateurs et des applications.

CONTEXTE ET SENSIBILITÉ AU CONTEXTE DANS LES SYSTÈMES D'INFORMATIONS PERVASIFS

Dans le domaine de l'informatique, la notion du contexte désigne d'une manière générale l'ensemble des informations qui entourent l'activité de l'utilisateur, tout en tenant compte de leur contexte. Avec l'avènement des systèmes d'informations pervasifs, l'application doit assurer une adaptation au type d'utilisateur, au type de son terminal, à sa localisation géographique, à ses tâches... Ainsi de nouveaux domaines sont apparus et connus sous le nom de *sensibilité au contexte* ou de *context-awareness*.

Une application dite sensible au contexte doit percevoir la situation de l'utilisateur dans son environnement, tout en lui permettant d'interagir de manière transparente en fonction de son contexte et de l'assister dans ses tâches de tous les jours. Ainsi cette application adapte son comportement aux différentes situations contextuelles instantanées et courantes.

L'utilisation de la sensibilité au contexte pour améliorer les interactions entre homme et machine du point de vue de l'ergonomie des interfaces et de la connaissance d'informations sur l'utilisateur s'est considérablement développée avec l'arrivée du Web. Les interfaces graphiques suivent le même chemin et se reconfigurent compte tenu des fonctions les plus utilisées par l'utilisateur. Un module de sensibilité au contexte dans un système pervasif regroupe des informations et des événements qui ont lieu dans l'environnement des entités du système informatique. Ces entités sont les usagers, le matériel, les logiciels, ...

Les paramètres du contexte sont utilisés par le système pervasif pour fournir du soutien

et pour réagir proactivement aux demandes instantanées des utilisateurs. On dit d'un système qu'il est sensible au contexte s'il peut récupérer, interpréter et utiliser des informations issues du contexte et adapter sa réponse en fonction du contexte d'utilisation. L'étude de la littérature montre que la sensibilité au contexte est devenue un élément central pour la conception et la mise en place de services adaptatifs.

ADAPTATION CONTEXTUELLE AUX SERVICES DE SYSTÈME PERVASIF

Devant la diversité de contexte d'utilisation des applications interactives pervasives, l'adaptation automatique de services à l'utilisateur est vue comme une nécessité. Elle n'est pas considérée non plus comme une simple personnalisation mais plutôt c'est une exigence de satisfaction. L'adaptation automatique de services est caractérisée comme étant une interaction implicite avec le service afin de changer de manière dynamique son comportement à l'égard de l'utilisateur. Les données sur l'utilisateur et son environnement constituent les sources principales de ce changement. La prise en compte de l'adaptation des services permet d'assurer à la fois la continuité des services à l'utilisateur et d'améliorer l'interaction de l'utilisateur avec son environnement. Ainsi, l'adaptation des services à l'utilisateur s'avère être un facteur déterminant pour assurer la continuité de ces services et veiller à ce que le service ait bien effectué la tâche demandée. Et afin d'améliorer les interactions de l'utilisateur avec son environnement, les systèmes pervasifs, via les différents outils dédiés à la sensibilité au contexte d'utilisation, permettent d'assurer une interaction plus souple de l'utilisateur avec son environnement. L'adaptation des services à l'utilisateur joue un rôle important dans l'amélioration de cette interaction en facilitant et en aidant l'individu dans sa relation avec son environnement.

CONTEXTE DE LA THÈSE

Dans ce paysage, l'adaptation au sein de tels systèmes devient un point central. Mettre en œuvre une telle adaptation est loin d'être une tâche facile. De nombreuses questions de recherche doivent être résolues avant de permettre une adaptation automatique En s'appuyant sur deux aspects de base des systèmes pervasifs (qui sont "la sensibilité" et "l'adaptation contextuelle") la grande majorité des contribution existantes dans ce domaine s'intéresse à la manière de capturer et de collecter des informations de contexte, d'interpréter et de fournir ces informations à l'application. De plus, la plupart des travaux existants se focalisent sur la création des prototypes d'applications sensibles au contexte en incorporant le code d'adaptation dans le code métier de l'application, et donc sans tenir compte des particularités des systèmes pervasifs comme la mobilité et les changements dynamiques de contenus du contexte d'utilisation.

Dans notre travail nous nous intéressons particulièrement à la modélisation et à la gestion de la sensibilité au contexte pour l'adaptation des services proposés par des systèmes pervasifs. L'étude de ces systèmes intégrant ces fonctionnalités est une thématique de recherche multidisciplinaire. Le domaine d'application retenu dans cette thèse se situe à l'intersection de cette thématique et des ontologies pour la modélisation de la sensibilité de contexte. Cette thèse s'intéresse à la problématique de l'adaptation de services dans les applications pervasives dans le domaine de la sensibles au contexte. L'étude de la littérature montre que la sensibilité au contexte est devenue un élément

central pour la conception et la mise en place des systèmes pervasifs dotés de services adaptatifs. Cependant, sa prise en compte se limite généralement à des descriptions élémentaires de situations ou à des modèles prédéfinis. Afin de permettre une adaptation aux changements d'habitudes des utilisateurs, à la dynamique de l'environnement et à l'hétérogénéité des sources de perception, nous proposons des mécanismes de traçabilité de contexte et de situations déclenchant l'adaptation qui permettent de fournir des ontologies d'informations de contexte. Ces mécanismes s'appuient sur des techniques de mapping et de cartographies pour la génération des ontologies qui seront intégrées au sein de notre architecture d'adaptation de services au contexte : cette architecture se nomme *COALA*, pour *COntext Adaptation Platform*.

OBJECTIF DE CES TRAVAUX

Le cadre général de nos recherches est donc le domaine de la sensibilité au contexte des systèmes pervasifs. Notre intérêt porte en général sur la proposition de la nouvelle plateforme *COALA* pour l'adaptation des systèmes pervasifs sensibles au contexte. Ceci comprend la modélisation et la gestion de la sensibilité au contexte à travers la catégorisation et la représentation formelles des information de contexte. Ainsi cela comprend l'adaptation dynamique des applications interactives pervasives aux changements dynamiques de l'environnement et aux contextes des divers utilisateurs.

Nous nous appuyons sur des ontologies générales pour la modélisation des éléments de contexte, ainsi que sur des mécanismes de récupération des informations des situations contextuelles courantes sous formes des cartes de visite contextuelle. Cette démarche nous permet de générer des ontologies de traçabilité, de traiter la complexité de modélisation des informations de contexte ainsi que l'adaptation des services à la dynamique de l'environnement (aux changements des services, d'usages et à l'hétérogénéité et au volume important des données de contexte).

Au final, nous contribuons au développement de notre plateforme informatique permettant de mettre en œuvre l'architecture finale de la solution proposée et nous la validons sur des scénarios d'usage dans le domaine de médecine.

PLAN DU MÉMOIRE

Nous avons organisé ce document en deux parties : un état de l'art permettant de mettre en évidence les différentes problématiques de notre thèse (sur deux chapitres), et la contribution de la thèse, avec validation de notre proposition sur les scénarios d'usage retenus (sur deux chapitres également).

Dans la première partie, le premier chapitre fait l'objet d'une revue de la littérature qui permet de définir précisément ce que nous considérons dans ce document lorsque nous parlons de contexte et d'informatique sensible au contexte. Il constitue le cadre général de notre recherche, ses différents aspects et approches. Ce chapitre présente et discute diverses questions relatives à la modélisation et à l'adaptation au contexte dans les environnements utilisateurs.

Nous présenterons, dans le deuxième chapitre, une classification des différentes approches qui abordent le domaine de la sensibilité au contexte. Nous exposons également les catégorisations existantes de contexte disponibles dans la littérature. Nous finissons notre étude par un tableau comparatif qui récapitule les travaux antérieurs les plus importants dans notre domaine d'étude tout en analysant les limitations de ces travaux. Enfin, nous justifions, dans ce chapitre, notre choix du modèle *PIVon* qui aborde la sensibilité au contexte à base d'ontologies comme une référence pour notre travail.

Dans la deuxième partie, le chapitre trois expose notre ontologie de traçabilité pour la modélisation et l'adaptation de contexte. Nous présenterons les différents modèles généraux ontologiques de contexte que nous utilisons dans notre approche, puis nous présentons notre modèle de contexte à base d'ontologie de traçabilité. Nous exposons dans ce même chapitre les différentes méthodes et outils suivis dans notre approche.

Le chapitre quatre fait l'objet de la présentation de notre nouvelle plateforme d'adaptation *COALA*, pour *COntext Adaptation Platform*, qui intègre notre modèle de contexte à base d'ontologie de traçabilité. Ce même chapitre nous permet également de définir le cadre expérimental utilisé pour la validation de notre proposition. Nous détaillons particulièrement nos expérimentations à travers une étude de cas dans le domaine de la médecine.

Nous terminons ce document en synthétisant les différentes contributions que nous avons proposées étant donnés les objectifs que nous nous étions fixés pour ce travail. Nous finalisons ce chapitre par différentes perspectives de recherche que soulève notre travail et que nous envisageons pour des travaux futurs.

QUELQUES INDICATIONS POUR LA LECTURE DE CE RAPPORT

Le plan est organisé sur quatre niveaux : Parties, Chapitres, Sections et Sous-Sections. Il est utile de savoir que la Partie II, présentant les contributions et résultats, est indépendante de la première Partie.

Les résultats de nos travaux ont pour la plupart été publiés. La liste des publications personnelles est donnée avant la bibliographie placée en fin du document.

I

État de l'art

L'informatique pervasive a été introduite initialement avec l'avènement des technologies mobiles. En effet, l'utilisation de l'informatique à ses début prenait une forme standard : un utilisateur d'une application est associé à un ordinateur situé dans un bureau. L'informatique pervasive rend l'utilisation de l'informatique possible en tout moment et indépendamment du positionnement de l'utilisateur.

De nos jours, l'interaction avec les applications informatiques exploite de plus en plus les technologies mobiles. L'adoption de ces technologies assure plus de flexibilité et permet de créer de nouvelles formes d'utilisation dans des environnements différents : de nouveaux usages. En effet, les informations caractérisant ces environnements sont dites des informations de contexte. En conséquence, la conception des applications interactives doit être sensible à ces informations de contexte.

L'étude de la sensibilité au contexte nous amène à étudier les travaux de recherche réalisés pour définir et modéliser le contexte. En effet, la modélisation permet d'offrir les fondations pour une représentation expressive du contexte et de simplifier son utilisation. Les modèles de description du contexte varient selon leur expressivité et le type d'informations de contexte qu'ils permettent de décrire.

La variété au niveau de la modélisation provient de l'utilisation du contexte dans les domaines variés de l'ingénierie comme le traitement du langage, les systèmes d'aide à la prise de décision ou les systèmes sensibles au contexte. Les actions fournies par ces applications doivent être dotées de mécanismes qui collectent les informations de contexte, stockent et gèrent ces informations et qui réagissent à base de ces informations. Ces applications doivent réagir selon les exigences imposées par les informations de contexte. Les applications qui intègrent ces mécanismes sont les applications sensibles au contexte.

Dans cette première partie de thèse, nous nous intéressons aux travaux sensibles au contexte déjà effectués dans la littérature. En effet, les applications interactives sensibles au contexte sont caractérisées par le fait qu'elles s'exécutent dans un environnement hétérogène et dynamique. Cet environnement est soumis aux caractéristiques variables des ressources des dispositifs utilisés et à la mobilité des utilisateurs, ce qui rend nécessaire l'utilisation des applications sensibles au contexte qui détectent les variations de l'environnement et adaptent leurs comportements en conséquence.

Nous commençons dans le premier chapitre par une étude sur les travaux existants des systèmes sensibles au contexte. Cette étude est réalisée en se référant à l'ensemble des modèles fournis par chaque approche centrée sur la sensibilisation au contexte. L'étude de ces modèles a donné l'objet de plusieurs nouveaux débats. Tout d'abord, nous avons constaté que les travaux étudiés proposent des modèles formels explicites de la sensibilité au contexte de l'environnement de travail. Or, chaque modèle de l'environnement dans lequel se situe l'utilisateur est indispensable pour la conception d'une application sensible au contexte, ce que incite à travailler de manière très précise sur les caractéristiques de ces environnements. Dans le premier chapitre, nous abordons la définition de la sensibilité au contexte tout en mettant en évidence l'importance de celle-ci dans le développement des applications interactives dans des environnements mobiles. Nous passons en revue différents modèles dans lesquels le contexte est bien modélisé. Puis, nous décrivons les caractéristiques des informations de contexte dans chaque modèle etudié et les différentes méthodes d'acquisition et de traitement de ces informations. Puisque la modélisation de la sensibilité au contexte représente l'une des premières phases de prise en compte du contexte dans le processus de création d'appli-

cations sensibles au contexte, nous dressons un état de l'art des travaux de modélisation du contexte, puis nous analysons ces différentes approches pour choisir l'approche la mieux adaptée aux différents environnements des utilisateurs.

Puis, nous avons déduit de cette étude que les approches de modélisation de la sensibilité au contexte sont prises en compte en même temps l'adaptation des situations contextuelles. Or, une adaptation au contexte est un ensemble de règles qui prennent la forme de condition-action : les conditions portent sur les variables de contexte et les actions expriment les changements à apporter à l'application. Nous définissons l'association d'une perception d'une information de contexte à une action d'adaptation. Pour cela, nous avons décrit la méthodologie d'adaptation au contexte dans la deuxième section de ce chapitre.

Dans la deuxième chapitre, nous avons réalisé une étude comparative entre les différentes approches (que ce soit au niveau de la modélisation de la sensibilité du contexte ou au niveau de l'adaptation), afin de mettre en valeur les approches qui peuvent répondre au mieux à nos besoins et qui seront utilisées dans notre contribution.

1

LE CONTEXTE DE L'INFORMATIQUE SENSIBLE AU CONTEXTE

INTRODUCTION

L'informatique contextuelle est l'informatique qui tient compte de la position et/ou du comportement de l'utilisateur en vue de lui fournir un service. Ce domaine est très prometteur. En effet, après les ordinateurs haute performance (*mainframes*) et les ordinateurs personnels, ce domaine est un troisième paradigme de l'informatique ubiquitaire. Notre sujet de thèse se situe au croisement de plusieurs domaines de recherche tels que la modélisation contextuelle, l'informatique ambiante, et la gestion de connaissances.

1.1/ LE CONTEXTE EN INFORMATIQUE UBIQUITAIRE

L'informatique contextuelle est apparue dans le milieu des années quatre-vingt dix inspirée par les travaux de Schilit et Theimer [Schi94]. À cette époque, la définition de contexte dans le domaine de l'informatique est plus orientée vers l'énumération des éléments spécifiques, qui représentent le contexte associé à une application (comme la localisation par exemple), que vers la généralisation de l'information de contexte :

- En effet, pour le premier type de définition de contexte, les chercheurs ont essayé de présenter le contexte à travers l'ensemble des éléments qui peuvent être considérés comme une information de contexte. Ces éléments constituent des informations non fonctionnelles de l'application qui peuvent apporter un impact sur ces dernières. Parmi ces travaux, la définition donnée par de Schilit [Schi94], qui considère que le contexte, est la localisation de l'utilisateur, les identités et les états des personnes et des objets qui l'entourent. La même définition a été reprise par Brown et al. [Bro97] qui définissent le contexte en tant que l'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. Rayn et al. ajoutent également à cette énumération des informations sur l'environnement telles que la température, d'où ils présentent le contexte par la localisation, l'identité et le temps de l'utilisateur et surtout par l'environnement physique qu'il l'entoure.
- Avec l'évolution de nouvelles technologies ubiquitaires et de nouveaux usages, une nouvelle vision est en train de naître. En effet, un aspect conceptuelle qui se focalise sur la structure des informations de contexte à travers une abstrac-

tion et la généralisation de l'information de contexte a vu le jour. Cette évolution, au niveau de la définition de contexte, permet de le centrer sur les utilisateurs et le monde réel qui les entoure. Reposant sur ces nouveaux éléments, les chercheurs tendent de plus en plus de donner une définition générique à l'information de contexte en montrant son utilité et en la regroupant selon plusieurs catégories. Parmi ces travaux, nous pouvons présenter celui d'Amara-Hachimi et al [Amar06] qui considèrent que le contexte regroupe toutes les informations sur l'environnement courant d'exécution. Ces informations sont distribuées selon trois niveaux de contexte : le contexte physique, le contexte social et le contexte utilisateur. Également, les travaux de Brézillon et Pomerol [Most04] définissent le contexte comme étant toutes les connaissances qui participent à la résolution d'un problème.
Chaari et al.[Cha07] définissent, quant à eux, le contexte comme l'ensemble des paramètres externes à l'application pouvant influer sur le comportement d'une application en définissant de nouvelles vues sur ses données et ses services. A. Dey, dans [Dey01], définit également le contexte comme étant *toutes les informations pouvant être utilisées pour caractériser la situation d'une entité. Une entité est une personne, un lieu, ou un objet qui peut être pertinent pour l'interaction entre l'utilisateur et l'application, y compris l'utilisateur et l'application*.

Cette dernière définition est considérée comme étant la plus générale et la plus utilisée. En donnant ces définitions, la majorité des travaux étend la définition de A. Dey qui considère d'une façon générale le contexte comme toute information qui peut être utilisée pour caractériser la situation d'une entité pouvant être jugée significatif à l'interaction entre l'utilisateur et l'application, incluant à la fois l'utilisateur et l'application.

1.1.1/ LA SENSIBILITÉ AU CONTEXTE

Avec l'avènement des environnements mobiles, la sensibilité au contexte est un nouveau besoin des applications interactives. En effet, la sensibilité au contexte est la réaction du système face à un évènement survenu. Cette réaction se fait en réalisant une compatibilité entre le sens de l'action et le sens particulier apporté par le contexte. Le sens de l'action peut être défini par l'objet de l'action, son objectif et surtout les moyens disponibles pour invoquer cette action. La compatibilité est faite en contraignant le processus de construction et d'exécution de ces actions.

Pour les définitions de la sensibilité au contexte, [Ryan98] considèrent que les applications sensibles au contexte sont des applications ayant des mécanismes pour changer dynamiquement ou pour adapter leur comportement en se basant sur le contexte de l'application ou de l'utilisateur.

De même, G. Abowd [Abo99] considèrent qu'un système est sensible au contexte s'il utilise le contexte pour fournir une information pertinente à l'utilisateur, à travers d'une part, la collection et la gestion des informations de contexte et, d'autre part, la gestion et le contrôle des actions du système en fonction de ces informations de contexte. Pour sa part, M. Miraoui [Mira08] considère qu'une application est dite sensible au contexte si elle peut changer automatiquement ses formes de services ou déclencher un service comme une réponse au changement de la valeur d'une information ou d'un ensemble d'informations qui caractérisent le service.

L'étude des différents travaux existants montre que A. Dey[Dey01] est le seul qui a donné

une définition générale du contexte, comme suit :

Définition

"le contexte est construit à partir de tous les éléments d'information qui peuvent être utilisés pour caractériser la situation d'une entité. Une entité correspond ici à toute personne, tout endroit, ou tout objet (en incluant les utilisateurs et les applications) considéré(e) comme pertinent(e) pour l'interaction entre l'utilisateur et l'application".

Dit autrement, un système est sensible au contexte s'il utilise le contexte pour offrir des informations ou des services pertinents pour l'utilisateur.

1.1.2/ L'ACQUISITION DU CONTEXTE

Au cours de ces dernières années, de nombreux travaux se sont intéressés à la conception d'interfaces interactives intelligentes qui peuvent s'adapter dynamiquement au contexte d'interaction, afin de les rendre plus personnalisées. Les éléments de contexte doivent inclure les informations relatives à l'environnement de l'utilisation (par exemple, pour l'utilisateur, les informations liées aux préférences, historique des navigations, ...), ainsi que toutes les autres information pertinentes pouvant être utilisées pour caractériser les conditions d'interaction.

De ce fait, L'acquisition, la modélisation et le traitement de ces contextes d'interaction jouent un rôle fondamental dans le développement de telles applications interactives et intelligentes. En effet, l'acte d'acquisition du contexte représente le processus de collecte des données à partir de l'environnement du système. Bien que la collecte des informations de contexte dépende de la source du contexte et du type des informations collectées (par exemple, les informations du profil d'un utilisateur sont des informations explicitement fournies par ce dernier), la gestion de l'acquisition du contexte ne peut pas être mis en œuvre directement dans une application. Le développeur peut alors utiliser, pour récupérer le profil porteur de l'information, les services d'un intergiciel pour la sensibilité au contexte. Cet intergiciel se charge de collecter le contexte et de le fournir à l'application.

Le plus souvent, l'interaction entre l'intergiciel et l'application est faite en mode requête pour lequel les données collectées sont généralement sauvegardées dans une base de contexte pour être utilisées ultérieurement par le système.

1.1.3/ L'INTERPRÉTATION DU CONTEXTE

La couche d'interprétation du contexte représente le mécanisme qui déduit un contexte de haut niveau en utilisant d'autres informations de contexte. En premier lieu, cette couche a pour but l'interprétation, elle sert ensuite à l'analyse et à la transformation des données brutes fournies par une couche de déduction de contexte.

Les informations utilisées pour déduire un contexte de haut niveau peuvent être des informations interprétées, des informations capturées ou des informations de profil. La

fréquence de mise à jour des informations interprétées dépend de la fréquence de mise à jour des informations utilisées lors de l'interprétation. À chaque fois qu'une nouvelle donnée est collectée, le mécanisme d'interprétation doit être relancé pour déduire les nouvelles valeurs du contexte. Toutes transformations effectuées sur les données brutes concernent l'extraction et la quantification de celles-ci.

Cette couche peut aussi avoir le rôle de résolution de conflits causés par l'utilisation de plusieurs sources de contexte [Cha07]. Ces sources peuvent donner des résultats contradictoires ou peuvent aboutir à des situations imprécises. Cette couche doit donc avoir des moyens pour résoudre ces conflits. Une phase d'analyse doit donc intervenir juste après cette interprétation du contexte. En effet, l'analyse du contexte est un processus qui contrôle continuellement les données collectées. Ce contrôle permet de filtrer les observations et les interprétations du contexte pour détecter les situations pertinentes dans le but d'adapter le système en conséquence ou de notifier les applications sensibles à ces situations pertinentes. Grâce aux résultats obtenus par cette analyse, des décisions sont prises par le processus d'adaptation afin d'adapter le comportement de l'application aux nouvelles situations pertinentes.

1.1.4/ L'ADAPTATION AUX SITUATION DU CONTEXTE

L'adaptation au contexte est l'ensemble des mécanismes de réactions prévues suite aux changements de contexte. En effet, selon P-C. Davi[Davi05] :

Définition

"Une adaptation est une modification d'un système, en réponse à un changement dans son contexte, avec l'objectif que le système résultant soit mieux à même pour réaliser sa fonction dans le nouveau contexte"

L'adaptation se base sur un ensemble de règles d'adaptation qui sont implémentées selon des langages de programmations en utilisant la logique des prédicats. Certains travaux adoptent la logique floue ou la logique probabiliste afin de remédier au caractère incertain du contexte. Dans la littérature, plusieurs formes d'adaptation peuvent se présenter [Raib08] :

- L'adaptation architecturale que désigne les changements faits, en temps réel, dans la structure des composants du système dans les interactions entre eux en utilisant un modèle architectural du système.
- L'adaptation compositionnelle qui permet de définir les modifications de la structure et du comportement d'une entité logiciel, en temps réel, en réponse aux changements arrivés à son environnement d'exécution.
- L'adaptation structurelle qui signifie le changement dynamique des types des composants de l'application tels qu'une signature d'une méthode.
- L'adaptation de contenu qui concerne la transformation et la manipulation des contenus en se basant sur les caractéristiques de l'application et des terminaux utilisés.

Pour finir, l'adaptation est une modification d'une application en réponse à un changement

de contexte en cours : cette modification consiste soit à changer la structure de l'application, soit son comportement. Mais, tous les changements de contexte n'engendrent pas forcément l'adaptation de l'application. Pour ce faire, nous appelons situation pertinente, toute observation du contexte pertinent ou composition d'observations à un instant donné qui nécessite l'adaptation de l'application. En résumé, toute adaptation est conditionnée par la détection d'une ou de plusieurs situations pertinentes.

1.2/ La modélisation du contexte dans les applications interactives

Le contexte est considéré comme une propriété qui caractérise une entité de l'application. Les éléments sensibles au contexte sont exprimés dans un modèle de contexte qui intègre un modèle de données. Ce modèle s'opposent à d'autres modèles qui considèrent le contexte comme étant une donnée additionnelle utilisée par l'application pour réaliser une certaine fonctionnalité. Le but de l'étude des interactions est d'analyser la façon dont les utilisateurs interagissent avec les systèmes informatiques dans différentes situations contextuelles, et ainsi de proposer de nouvelles solutions pour améliorer ces interactions.

Pour les concepteurs d'applications, il s'agit également de proposer des outils qui permettent une réalisation plus facile d'applications mieux adaptées aux utilisateurs. Les modèles d'interaction sont basés uniquement sur la modélisation architecturale, c'est à dire la manière dont un système fonctionnera. En effet, La modélisation des interactions constitue un élément fondamental nécessaire d'une part à la compréhension de la nature de la communication Homme-Machine et d'autre part au processus de développement d'interfaces. Elle décrit le processus général de l'interaction Homme-Machine, c'est à dire la structure des échanges de l'utilisateur avec la machine.

Dans ce chapitre, nous étudierons les différents paramètres d'interaction entre l'homme et la machine qui nous seront utiles pour la suite.

1.2.1/ Les Modèles architecturales classiques

Les modèles d'architecture constituent un des centres d'intérêt des recherches pour les systèmes interactifs depuis une quinzaine d'années. Ces modèles formalisent la façon dont l'utilisateur peut interagir avec le cœur procédural des applications (souvent appelé noyau fonctionnel) via une interface dédiée aux utilisateurs d'une telle application interactive. La modélisation architecturale correspond respectivement aux deux types de modèles d'interaction que distinguent H. Hartson et D.Hix dans [Har89] : les modèles de description générale de l'interaction Homme-Machine et les modèles d'architecture proprement dits.

Les premiers décrivent directement l'interaction en termes de manipulation de l'utilisateur, de messages du système et d'enchaînements entre les séquences d'interaction. Les seconds sont proposés dans le but de décrire, dans un contexte architectural, les relations de l'interface utilisateur avec le reste du système interactif. Ils décrivent la structure interne de l'interface utilisateur : identification des composants et de leur organisation (relations, protocole de communication, . . .).

Dans la littérature, depuis les années 1970, différents modèles de ce type ont été pro-

posés et ont déjà donné lieu à de nombreuses publications et nous centrerons notre travail, dans cette section, sur les rappels des caractéristiques essentielles des modèles les plus représentatifs pour nos travaux de recherche tout en donnant une liste assez exhaustive.

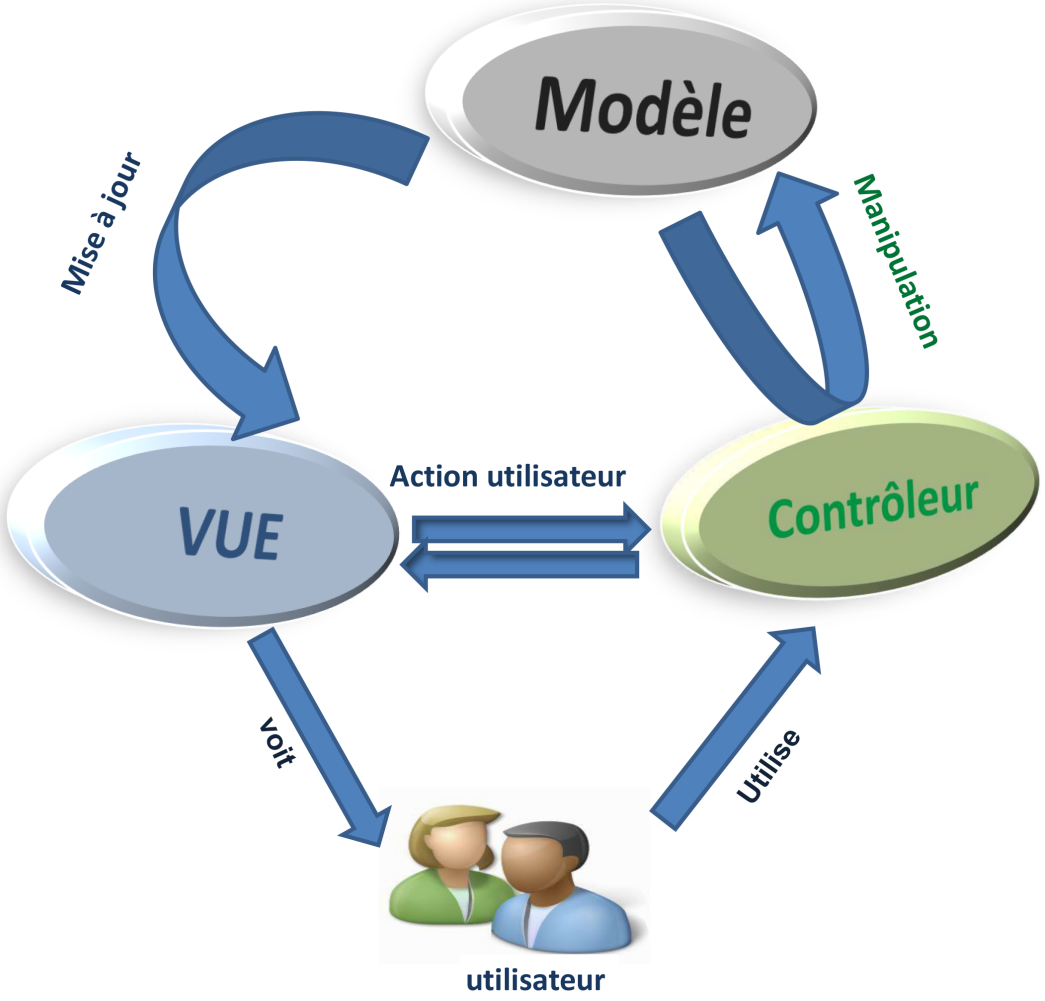

FIGURE 1.1 – L'architecture du Modèle Vue-Contrôleur

LE MODÈLE VUE-CONTRÔLEUR : MVC

Dans les années 1970, G. Krasner a proposé, dans [Kras88], de formaliser en premier les relations entre une l'interface et les données. Dans ses travaux de recherche sur le pattern Modèle Vue-Contrôleur, T. Reenskaug [Ree79] en 1979 a développé le MVC dont le but principal était de proposer une solution générale aux problèmes d'utilisateurs (profils utilisateurs) en termes de volume et de complexité indépendant du langage utilisé.

Le modèle MVC a également été considéré comme un modèle destiné à répondre aux besoins des applications (profils applications) interactives en étant défini par trois entités de base à savoir le modèle, la vue et le contrôleur (cf figure 1.1).

- Le modèle :

Il correspond au noyau fonctionnel d'une application et à ses données métier. Les résultats renvoyés par le modèle ne s'occupent pas de la présentation. Par exemple, dans un programme de visualisation des données d'un recensement, le modèle à base du MVC contiendra toutes les données démographiques.

- La vue :
  Elle permet d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur. Une vue consiste généralement en une interface graphique ou textuelle pour la représentation des informations à destination de l'utilisateur.
- Le contrôleur :
  Il permet de gérer les interactions avec l'utilisateur, et plus particulièrement, il est chargé de la gestion des actions en entrée effectuées par l'utilisateur. Le contrôle agit sur demande de l'utilisateur et modifie le modèle.

Pour une modélisation à base du Modèle Vue-Contrôleur, il existe un mécanisme de notification qui permet de transmettre des notifications de modifications à l'ensemble de ses vues et contrôleurs. Il est possible que le modèle se trouve modifié, soit par l'un quelconque des contrôleurs qui lui sont associés, soit par le noyau fonctionnel lui-même. Le modèle doit alors informer chacune de ses vues que ses données ont changé, afin qu'elles puissent le cas échéant mettre à jour leurs informations.

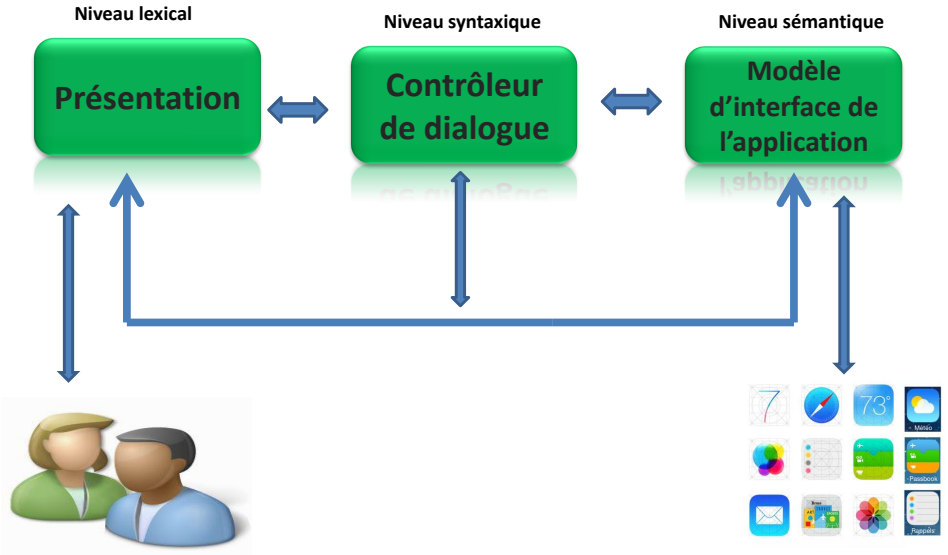

FIGURE 1.2 – La modélisation classique d'une interface d'interaction dans Seeheim/ARCH

LE MODÈLE CONTRÔLEUR DE DIALOGUE-INTERFACE

Le modèle Seeheim [Pfa12] était principalement destiné au traitement lexical des entrées et sorties dans les interfaces textuelles. Bien que tous les modèles partent du principe qu'un système interactif comporte une partie interface et une partie application, le modèle ARCH est le premier modèle de référence pour ajouter une nouvelle partie linguistique correspondant à la présentation physique des informations.

Pour une modélisation de contenu textuel, ce modèle a intégré deux couches, une couche sémantique (sens d'un discours), et une couche syntaxique et lexical (règles d'agencement des mots en phrases, des phrases en paragraphes,...). Par analogie, la couche sémantique correspond au noyau fonctionnel et la couche syntaxique et lexicale (linguistique) est chargée de l'ordonnancement des interactions.

Pour sa modélisation, le modèle Seeheim propose de diviser l'interface en trois couches logicielles (cf figure 1.2) à savoir :

- la couche présentation :
  Elle est chargée de la gestion des entrées et sorties. Elle interprète les actions de l'utilisateur et génère les sorties (affichage) au niveau sémantique.
- Le contrôleur de dialogue :
  Ce contrôleur gère les séquences de l'interaction en entrée et en sortie. Il maintient un état lui permettant de gérer les modes d'interaction et les enchaînements d'écrans.
- L'interface du noyau fonctionnel :
  Elle est la couche intermédiaire entre le système interactif et le noyau fonctionnel. Elle convertit les entrées en appels du noyau fonctionnel et les données abstraites de l'application en des objets présentables à l'utilisateur.

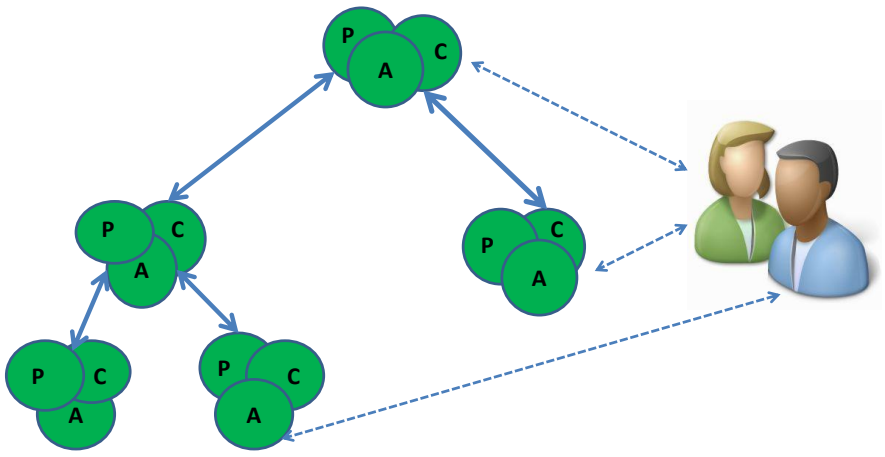

FIGURE 1.3 – L'architecture du modèle PAC

LE MODÈLE PRÉSENTATION-ABSTRACTION-CONTRÔLE : PAC

PAC est un modèle à base d'agents dérivé du modèle MVC. En effet, dans le modèle PAC, Joëlle Coutaz et al. [Cout87] décomposent l'application interactive en une hiérarchie d'agents interactifs communiquant entre eux, de manière à se déléguer les différentes sous-tâches d'un système interactif. Ainsi, chaque agent PAC (cf figure 1.3) dans la communauté d'agents possède trois facettes principales :

- Le composant Présentation :
  Il est chargé principalement de la transmission (visuelle, sonore, ...) des informations à l'utilisateur et de l'acquisition des entrées de l'utilisateur. Il correspond réellement aux vues et aux contrôleurs de l'architecture MVC.
- Le composant Abstraction :
  Il contient généralement les données et les traitements sémantiques.
- Le composant Contrôle : Il est le moteur des dialogues qui s'établissent entre l'agent et l'utilisateur, également entre l'agent et les autres agents puisqu'il est chargé de la communication avec les autres agents PAC au sein de leur hiérarchie comme le montre la figure suivante.

L. Nigay et J. Coutaz [Cout95] ont développé par l a suite le modèle PAC/Amodeus comme étant une nouvelle extension du PAC. En effet, PAC/Amodeus fait l'objet d'une combinaison des avantages d'une part du modèle Arch qui intègre des aspects de génie logiciel comme la modifiabilité et la portabilité, et d'autre part, du modèle PAC, qui permet de structurer efficacement le contrôleur de dialogue. Il réutilise alors le modèle Arch et décrit son contrôleur de dialogue avec une hiérarchie d'agents PAC. En effet, dans le fonctionnement de PAC/Amodeus (cf figure 1.4), toutes les entrées sont décrites par des symboles et des grammaires, et des mécanismes de fusion de haut-niveau sont implémentés dans le dialogue par des agents PAC.

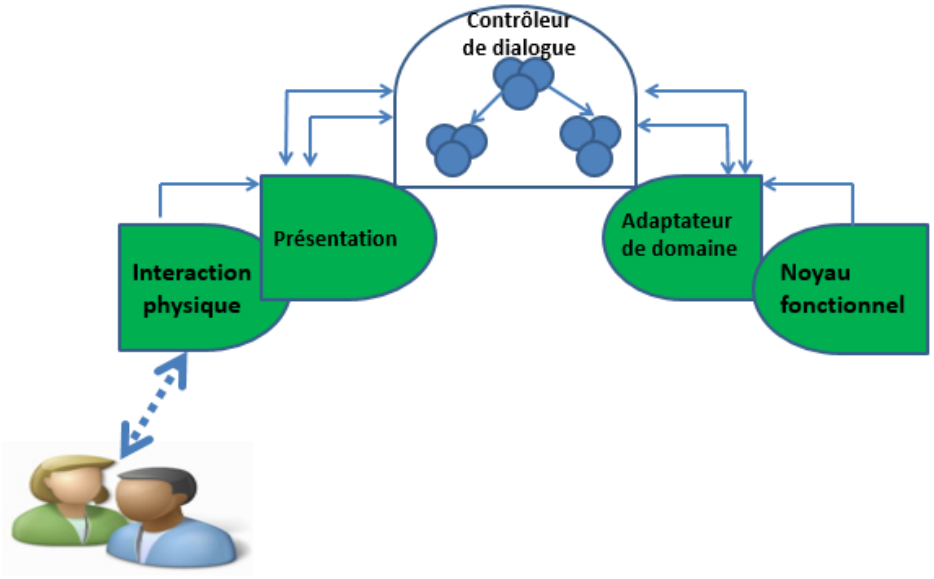

FIGURE 1.4 – L'architecture du modèle PAC/AMODEUS.

Discussion

Un modèle d'architecture permet le développement facile des interfaces d'interaction tout en séparant les différents modules (séparation de l 'interface et du noyau fonctionnel dès la conception) ce qui facilite le processus de conception de telles applications.
Mais ces modèles reposent sur un noyau fonctionnel stable et rigoureusement défini au moment de la conception de l'application interactive et donc il ne sera jamais réutilisable par d'autres applications.
Il faut également remarquer que, le contexte est absent de ces architectures. Or, dans un modèle d'architecture d'interaction classique comme par exemple MVC ou PAC, l'utilisateur (son profil, ses préférences, ...) n'est jamais représenté explicitement par ces modèles.
Notons également : ces architectures, longtemps utilisées en informatique conventionnelle, ont trouvé leurs limites avec l'informatique ambiante. Elles reposent pratiquement sur une vision centralisée de l'application et de l'interaction et ne peuvent donc répondre à aucun des problèmes de la distribution des application mobiles. De plus, une interaction mobile intelligente doit tenir compte des préférences de ses utilisateurs ainsi que de la sensibilité au contexte.

1.2.2/ LES APPROCHES DE MODÉLISATION ORIENTÉES PAIRES/TRIPLETS

Afin de modéliser les valeurs observées du contexte, les approches paires/triplés utilisent des structures de données très simples pour les décrire. En effet, ces approches présentent le contexte sous forme de paires (attribut, valeur). L'attribut caractérise le nom d'une information contextuelle. Le deuxième élément de la paire représente la valeur courante de cette information. B.schilit et M.Marvin dans [Schi93] proposent d'utiliser un serveur d'environnement dynamique qui se charge d'observer un ensemble de contextes pour le compte d'une application. Chaque observation du contexte est sauvegardée dans une variable d'environnement constituée d'une paire(clé, valeur). S. Schmidt (dans [Schm99]) quant à lui, modélise le contexte comme un triplet constitué de l'attribut, de la valeur observée du contexte, et du degré de certitude sur la cohérence des données observées. Par exemple, le triplet *(Location, ConferenceRoom, 95)* définit la localisation comme le lieu d'une conférence avec un degré de certitude égal à 95.

Généralement, l'approche paires/triplets est très fréquemment utilisée dans les frameworks des services distribués, comme le framework Capeus [Samu02]. On retrouve également dans cette catégorie les travaux qui se focalisent globalement sur l'aspect d'adaptation au changement du contexte et qui utilisent ce type de modélisation pour une description de localisation [Pito94].

Une modélisation de type paires/triplets utilise des structures de données simples à gérer, ce qui facilite son implantation. Cependant, elle ne permet pas une description complète du contexte, ni l'expression des relations qui peuvent exister entre les informations de contexte.

Discussion

Les approches paires/triplets sont caractérisées généralement par un faible niveau d'expressivité ainsi qu'une simplicité des données qu'elles représentent. En effet, une telle approche utilise des structures de données simples à gérer, ce qui facilite son implantation. En revanche, elle ne permet pas une description complète du contexte, ni l'expression des relations qui peuvent exister entre les informations de contexte ou la manière de déduire un contexte de haut niveau. Ce type de modélisation ne prend pas en compte la description des actions d'adaptation ni les conditions qui permettent de les déclencher.

1.2.3/ LES APPROCHES DE MODÉLISATION ORIENTÉES OBJETS

La modélisation du contexte par l'approche orientée objet permet d'offrir la puissance de tels mécanismes (encapsulation, réutilisation, héritage,...). Dans [Bouz97] est exposé une approche de modélisation orientée objet par sa capacité d'héritage et de réutilisation. Ces dernières permettent de définir un petit nombre de propriétés, fonctions et règles dans le but de simplifier la représentation des connaissances dans des systèmes complexes.

HYDROGEN, est une architecture orientée objet à 3 niveaux (cf figure 1.5) pour la modélisation du contexte dans le domaine de l'informatique mobile[Hofe03]. En effet, la modélisation de contexte dans HYDROGEN est fait sous forme des diagrammes de classes UML. Chaque type de contexte est composé de plusieurs objets qui constituent la superclasse de plusieurs éléments du contexte, notamment : le temps, le réseau, la localisation, l'utilisateur, la machine et d'autres éléments qui peuvent être ajoutés par héritage de la super classe. L'approche objet est utilisée pour pouvoir intégrer facilement la représentation du contexte au sein de l'application qui en dépend. Cette représentation du contexte s'appuie sur les propriétés d'encapsulation, de réutilisation et d'héritage. D'une façon générale, les termes sont représentés par les classes et les informations par les attributs de la classe. Le détail du contexte est masqué aux autres objets, grâce à la technique d'encapsulation.

1.2.4/ LES APPROCHES DE MODÉLISATION ORIENTÉES MODÈLES FORMELS

Afin d'offrir la possibilité de modéliser le contexte et de permettre sa réutilisation, les approches de modélisation orientées modèles utilisent des modèles formels bien structurés afin de bien spécifier les paramètres du contexte d'une application interactive. Avec ces modèles formels, un lot d'informations doit être décrit à travers l'ensemble des outils utilisés par chaque modèle pour décrire ces informations de contexte. Dans ce que suit, nous décrivons un ensemble d'approches appartenant à cette catégorie de modélisation.

LA MODÉLISATION CONTEXTUML

La première tentative de modélisation du contexte a été réalisée dans [Bau03] en utilisant le langage UML, afin de modéliser le contexte auquel une application de gestion

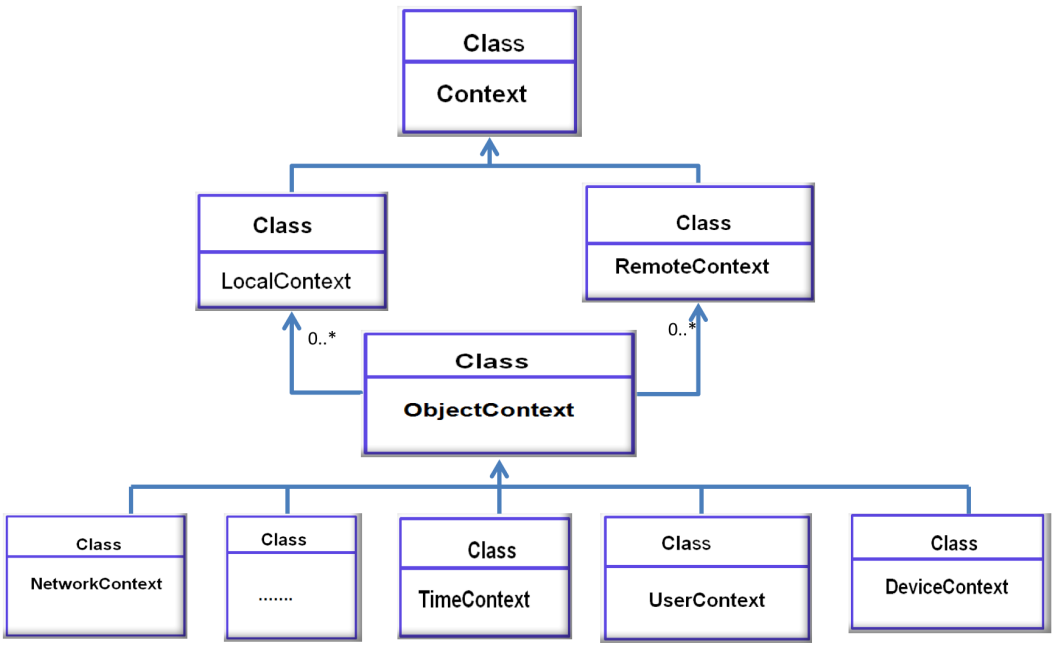

FIGURE 1.5 – Le modèle UML de contexte dans HYDROGEN

du trafic aérien est sensible. Bien que ce modèle ait réussi à catégoriser et à spécifier l'application en cours, sa spécification ne peut être utilisée dans d'autres applications sans modification. En effet, devant la multitude et la complexité de services sensibles au contexte, les chercheurs focalisent leurs travaux sur la manière dont un langage standard UML peut être utilisé pour fournir des informations structurelles relatives à la conception des applications sensibles au contexte.

Sheng et Benatallah, dans [Shen05], ont proposé un méta-modèle basé sur une extension du langage de modélisation unifié UML qui permet de modéliser le contexte auquel des services web sont sensibles. Ce langage est appelé ContextUML. En effet, ContextUML est un méta-modèle qui aborde les services sensibles au contexte *(Context-Aware Services : CAS)*. Ces services sont dotés de conscience sur l'environnement courant de l'exécution de leurs utilisateurs afin de fournir des services sur mesure.

Le développement de *CAS* permet d'un coté l'approvisionnement de l'information de contexte décrivant l'utilisateur telles que les préférences du client (la langue choisie, la résolution d'affichage, ...), la situation courante de l'utilisateur (par exemple l'emplacement géographique), plus d'autres informations (par exemple les informations décrivant les Services Web et les informations supplémentaires spécifiques à l'application, telles que l'heure, les informations de climat , ...). D'un autre coté, il permet de déterminer les mécanismes à utiliser pour adapter son comportement suivant le contexte sans l'intervention explicite de l'utilisateur. En donnant un exemple illustratif, la figure 1.6 montre que le méta-modèle ContextUML est composé de plusieurs classes qui permettent de créer des services sensibles au contexte.

En effet, le contexte est représenté par la classe Context qui permet de décrire un contexte observable. Or, un contexte observable peut être un contexte de bas niveau représenté par la classeAtomicContext, ou bien un contexte interprété représenté par la classe CompositeContext. Pour chaque contexte de bas niveau, le modèle permet de spécifier la source à partir de laquelle il a été collecté. Après la phase de spécification, Il

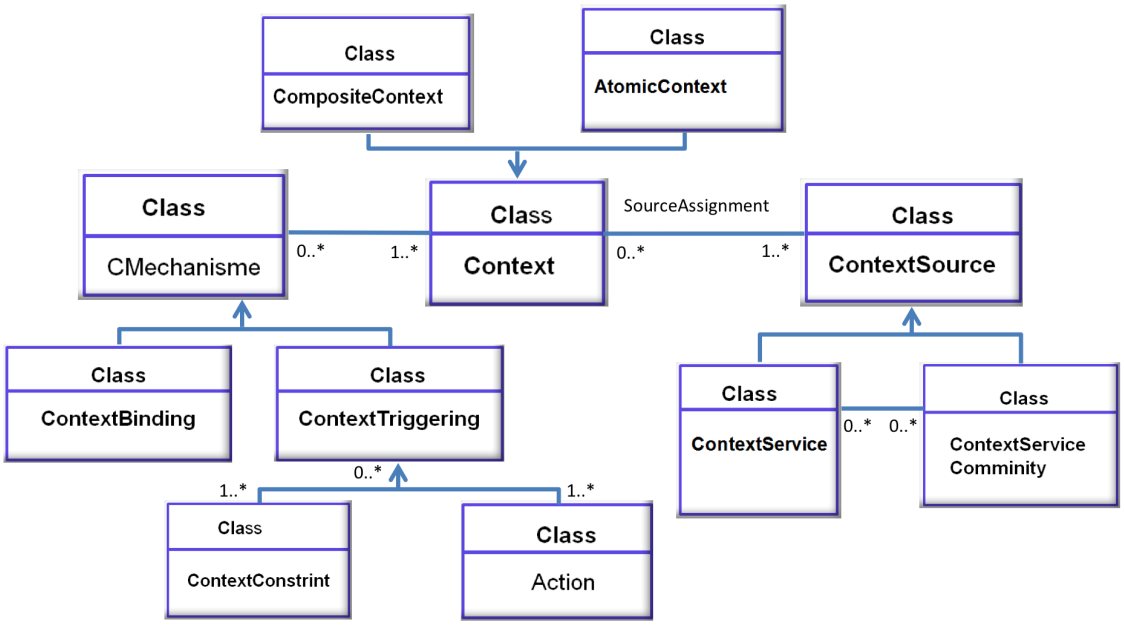

FIGURE 1.6 – Le méta-modèle CONTEXTUML

permet la description des actions d'adaptation et les situations pertinentes qui permettent de les déclencher.

Le méta-modèle ContextUML est caractérisé par sa généricité, ainsi il permet entre autre la description des actions d'adaptation et les situations pertinentes qui permettent de les déclencher. Mais parallèlement, une telle description au niveau des relations de dérivation et de dépendance entre les informations de contexte n'a pas été considérée. De plus, ce méta-modèle n'offre pas le moyen de décrire la qualité des informations de contexte ni leur validité temporelle.

LE LANGAGE CML

Dans cette modélisation, Les concepts à modéliser sont fournis à base d'une représentation graphique inspirée du modèle ORM (*Object-Role Modeling*). En effet, pour capturer les différents types d'informations contextuelles, en s'appuyant sur le raisonnement sur le contexte, ORM décrit les informations imparfaites et résout l'ambiguïté de l'information contextuelle.

Par la suite, Baldauf et al[Bal07] ont développé une approche de modélisation graphique basée sur ORM (*Object Role Modeling*). C'est une méthode dite orientée "fait" pour l'analyse de l'information au niveau conceptuel. La modélisation dans ORM consiste à identifier les types des faits appropriés et les rôles des types d'entités. Cette méthode a subi des améliorations et a donné le langage CML (*Context Modeling Language*) et elle apparaît comme une extension d'une représentation basée sur XML.

Ce langage permet de modéliser le contexte auquel une application est sensible dans son environnement. Il offre au concepteur d'une application sensible au contexte un moyen de décrire les caractéristiques des informations de contexte (capturées, statiques, dérivées

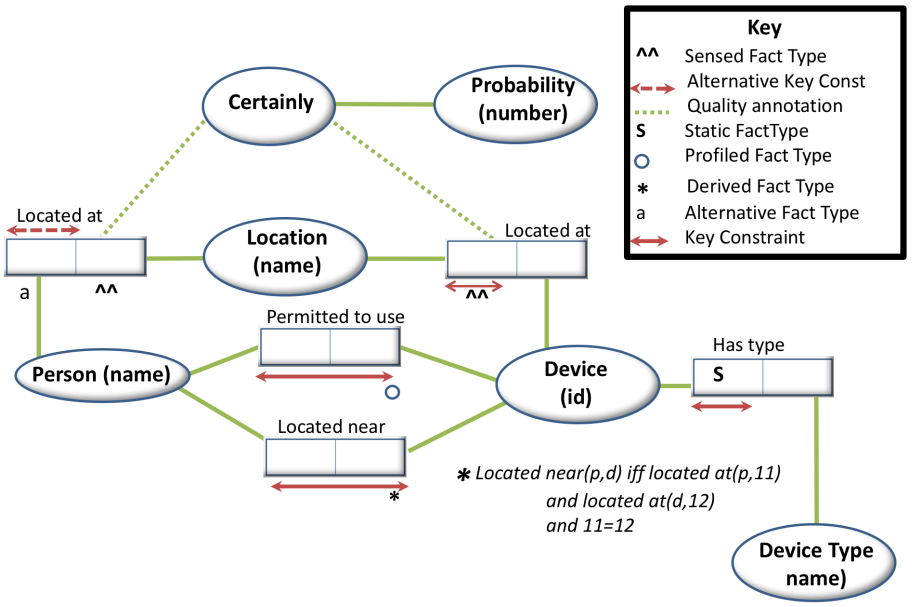

FIGURE 1.7 – Exemple d'une modélisation contextuelle avec CML

ou informations de profil) et les dépendances entre ces informations. Il permet alors de spécifier la qualité de chaque information observée et sa validité temporelle. Dans cette modélisation, les informations de contexte sont groupées en un ensemble d'entités. Chacune d'elle décrit un objet conceptuel ou physique tel qu'une personne, un dispositif. Les propriétés des entités telles que le nom de la personne, le nom du dispositif sont représentées par des attributs. Les entités sont liées à leurs attributs ou à d'autres entités à travers des relations. À l'aide de ces relations, le concepteur peut classifier les informations de contexte associées à chaque entité. La figure 1.7 illustre un exemple de modélisation de contexte avec l'outil CML en détaillant les associations entre les informations modélisées.

Le *Context Modeling Language* offre un modèle complet avec visualisation graphique qui permet de typer le contexte, et de décrire un ensemble de relations entre plusieurs contextes observables. Cependant, les outils associés à CML afin d'éditer ces différentes relations sont des outils classiques des bases de données.

LA MODÉLISATION PAR BALISES

Pour une modélisation contextuelle basée sur des balises, le langage utilisé est celui d'XML (*eXtensible Markup Language*), afin de décrire à la fois le contexte associé à l'utilisateur ainsi que le profil de l'application en cours, qui caractérise le contexte adéquat pour son exécution. En effet, les balises sont définies dans une DTD (*Document Type Definition*) afin d'avoir une description de différents profils typiques de ce type de modélisation.

L'un des travaux qui a contribué à une utilisation concrète d'une telle modélisation est celui de Pascoe (cf [Pasc97]) qui a utilisé un protocole d'échange de données contextuelles au format XML entre un serveur et un utilisateur mobile dans le cadre d'une application appelée *Stick-e Note*. En effet, Cette modélisation consiste à mettre en œuvre un ensemble de balises avec des attributs. Le protocole ConteXtML qui a été utilisé [Ryan99]

pour décrire le contexte observé associé à l'utilisateur, puis pour décrire le profil que représente le contexte nécessaire à une telle application pour qu'elle puisse s'exécuter.

```
//context Source
RCSMContext dc {
    char[] string location;
    boolean light;
}

//beginning of context-sensitive interface
interface instructor-object {
    //context variables
    RCSMContext_var dc C1
    where location="sceen";
    RCSMContext_var dc C2
    where light=true;
    RCSMContext_var dc C3
    where light=false;
    //context-sensitive method
    [out going]
    [activate when] C1^([C2->c3])
    void distribute(String lectures);
}
//end of context-sensitive interface
```

FIGURE 1.8 – Exemple d'une modélisation contextuelle avec CA-IDL

La modélisation à base de protocole d'échange CONTEXTML permet une description riche et détaillée à travers sa grammaire DTD ce qui donne une modélisation de données hiérarchiques : mais elle ne définit qu'un format d'échange de données. En effet, elle ne permet pas de décrire les contextes interprétés ni le profil de l'utilisateur. De plus, aucune description des relations entre les informations de contexte n'est prévue. Elle est spécifique à un ensemble limité d'applications.

LA GRAMMAIRE CA-IDL

```
RCSMContext
Device Specific Context{
  Doublebattery_power;
  Doublelight_intensity;
  Doublemenet_transmission_rate;
}
```

```
RCSMContext
Environment Specific Context{
  Using intnumber_peer_device;
  char[16]location;
}
```

```
RCSMContext
User Specific context{
  Usigned int
  calendar_usage_rate;
}
```

FIGURE 1.9 – Les catégories de contexte utilisée par CA-IDL

Pour la modélisation de la sensibilité au contexte des applications réparties orientées objet, Stephen et al. [Yau01] ont proposé une grammaire permettant la description d'interfaces sensibles au contexte baptisée CA-IDL (*Context Aware-Interface Description Language*. Ce langage permet de décrire les contextes (cf figure 1.8) auxquels l'application est sensible, les situations pertinentes et les actions d'adaptation d'une telle application

orientée objet. Pour le développeur, au lieu d'utiliser des mécanismes de transformation statiques de contexte, il suffit de spécifier l'interface à modéliser dans un fichier CA-IDL, et ce dernier permet la génération de la sensibilité au contexte de l'interface désignée dans son ADC (*Adaptive Object container*).

La plateforme, qui fournit le langage CA-IDL [Tsel04], organise l'environnement de travail à travers la définition principalement de trois catégories de contexte observable (cf figure 1.9). Chaque catégorie est constituée d'un nombre limité de types de contexte. La catégorie *DeviceSpecificContext* permet de décrire des informations de contexte spécifiques à une machine. La catégorie *EnvironmentSpecificContext* permet de décrire des informations de contexte spécifiques à l'environnement qui entoure l'application. Enfin, la catégorie *UserSpecificContext* permet de décrire des informations de contexte spécifiques à l'utilisateur.

Discussion

Par rapport aux approches paires/triplets, les approches orientées modèle utilisent un support formel plus riche pour décrire le contexte. En effet, ces approches orientées modèle sont des approches prometteuses car elles utilisent non seulement un modèle formel pour décrire le contexte, mais elle offrent également un méta-modèle de description qui peut être réutilisé par plusieurs applications. Cela offre aux développeurs d'applications la possibilité de réutiliser le modèle pour d'autres applications.
Pour les modélisations existantes basées sur UML (comme ContextUML par exemple) la description des relations entre les informations de contexte est une étape indispensable dans le processus de modélisation, mais en revanche elles ne prennent pas en compte la description des dépendances entre ces informations, ni la qualité ou la validité temporelle des données décrites. Dans le même cercle des approches orientées modèle, CML est venu remédier à certains de ces manques en proposant un modèle avec visualisation graphique qui permet de typer le contexte, et de décrire un ensemble de relations entre plusieurs contextes observables.
Les modélisations basées sur un langage de balises permettent, quant à elles, de fournir une description des contextes simples, sans offrir la possibilité de décrire des relations de dérivation et de dépendance entre ces différentes informations.
Enfin le langage CA-IDL permet non seulement la description des situations pertinentes de l'application sensible au contexte mais également l'ensemble d'actions d'adaptation avec les conditions de leur déclenchement. Généralement, ces différentes conditions représentent des expressions régulières entre les situations pertinentes. De ce fait, CA-IDL offre une grammaire qui permet de décrire la sensibilité au contexte, mais cette grammaire reste limitée dans la mesure où elle n'offre pas la possibilité d'ajouter de nouvelles sources de contexte et de nouveaux observables sans modifier la grammaire de CA-IDL, ni le moyen de décrire l'interprétation d'un contexte de haut niveau.

Le tableau suivant donne un récapitulatif sur les approches orientées modèle formel. En effet, pour cette étude des modèles formels en termes de modélisation du contexte,

une comparaison est faite sur leurs éléments fondamentaux. Notre comparaison porte particulièrement sur :

- Catégorie du contexte :
  Une catégorie de l'information de contexte est distinguée selon Utilisateur (U), Physique (P), Spatio-Temporel (ST) et Organisationnel (O). Pour chaque travail, les catégories de contexte considérées sont vérifiées.
- La base de modélisation :
  Elle permet de décrire le langage utilisé par chaque approche d'adaptation au contexte.
- Le type de modélisation :
  Il permet la description des situations pertinentes, à travers les différentes méthodes d'adaptation au contexte, de modélisation des relations de dépendance (RD), des règles d'adaptation (RA) et des sources du contexte (SC) qui semblent être très intéressantes pour automatiser le processus d'adaptation au contexte.
- Le type d'adaptation :
  Une adaptation au contexte peut être différenciée selon une réaction ou une intégration.
- La qualité :
  Elle permet de déterminer la qualité des informations de contexte données dans l'ensemble de l'approche d'adaptation : chaque approche ayant son propre modéle formel.

D'après le résumé de l'ensemble de caractéristiques des approches orientées modèles formels (cf table 1.1), nous constatons que la plupart des modèles ne permettent pas de décrire les états du contexte qui nécessitent une adaptation de l'application. Les modélisations à base de ContextUML, CML et CA-IDL se distinguent par le fait qu'elles permettent la description des situations pertinentes à travers la modélisation des méthodes d'adaptation, la modélisation des relations de dépendance, des règles d'adaptation et des sources du contexte que semblent être intéressantes pour automatiser le processus d'adaptation au contexte. Mais malheureusement, ces méthodes ne permettent pas de décrire la manière d'interpréter un contexte de haut niveau. De plus, les modélisations CA-IDL n'offrent pas la possibilité de décrire les sources de contexte.

Modèles	Catégorie de contexte	Modélisation à base	Type de modélisation			Type d'adaptation	Qualité
			RA	SC	RD		
ContextUML	U,P,ST et O	**Langage UML**	Oui	Oui	Oui	Réaction	✓Q
CML	U,P et ST	**Modèle ORM**	Non	Non	Oui	Réaction	✓Q
ContextML	U,P	**Langage de balise XML**	Non	Non	Non	Intégration	
CA-IDL	U,P et ST	**Grammaire**	Oui	Non	Non	Intégration	✓Q

TABLE 1.1 – Récapitulatif des approches orientées modèles formels

1.2.5/ LES APPROCHES DE LA SENSIBILISATION AU CONTEXTE ORIENTÉES PROFILS

LE LANGAGE COMPOSITE CAPABILITY/PREFERENCE PROFILE

La première définition de profils de contexte a été réalisée à l'aide du langage *Composite Capabilities/Preference Profiles* : nommé CC/PP. En effet, CC/PP est une recommandation du W3C (*World Wide Web Consortium*) basée sur RDF (*Resource Description Framework*) pour la représentation de profils qui permet de décrire les capacités d'un dispositif mobile ainsi que les préférences de l'utilisateur. CC/PP est utilisé pour personnaliser le contenu sur la base des capacités d'un terminal et des préférences de l'utilisateur à travers une structure composée d'une hiérarchie à deux niveaux : des composants (entités observables) et des étiquettes (observations). Cette structure de profil est constituée de :

- Une description en RDF avec une grammaire XML pour une représentation par graphes orientés étiquetés.
- Un regroupement de différentes caractéristiques du terminal et de préférences utilisateurs.
- Une hiérarchisation de description des deux niveaux (composants et étiquettes).

Dans cette structure, chaque profil a un certain nombre de composants et chaque composant possède des caractéristiques qui représentent une description de profil d'une personne (client).

La structure de profil CC/PP est très descriptive, mais il manque la 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 car il n'est pas riche et il est restreint à la description de profil.

Certaines tentatives d'extension de CC/PP ont été effectuées par Held et al. Dans [Held02], les auteurs ont proposé le *Comprehensive Structured Context Profile* (CSCP) qui permet d'avoir une description du contexte qui ne se limite pas seulement à deux niveaux hiérarchiques.

LE LANGAGE DE DESCRIPTION *User Agent Profile*

UAProf (*User Agent Profile* décrit dans [Chan11] propose une extension du langage CC/PP dans lequel Les éléments de vocabulaire dans UAProf sont utilisés dans la même structure de base que celle employée pour CC/PP. En effet, 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, ...

UAProf est un standard exploité par des milliers d'applications mobiles mais ce langage se limite exclusivement à la description de caractéristiques matérielles des applications sans fils.

LE LANGAGE DE DESCRIPTION *Comprehensive Structured Context Profile*

Le langage de description *Comprehensive Structured Context Profiles* (CSCP) est une extension qui a été proposée pour résoudre les problèmes de *Composite Capability/Preference Profile*. En effet, contrairement à CC/PP, CSCP permet de spécifier une description du contexte non limitée à deux niveaux hiérarchiques.

CSCP se base également sur le formalisme RDF (cf [Buch03]). Ce langage permet de décrire la localisation, les caractéristiques du réseau et les dépendances des applications. Il présente également une structure multi-niveaux qui est extensible. Il n'impose pas de structure hiérarchique fixe pour la notion de contexte. Ainsi, il permet la fusion des fragments de profil qui sont dynamiquement récupérés. Comme l'illustre la figure 1.10, ce langage de description exprime les informations de contexte au moyen d'un ensemble de sessions de profils (cf [Ach12]) :

- La session de profil de utilisateur :
  Cette session est composée de caractéristiques statiques de l'utilisateur (nom, prénom, ...) et de ses caractéristiques évolutives qui sont définies par son environnement (lieu, temps, ...) ainsi que de ses préférences.
- La session de profil de l'appareil :
  Cette session intègre le contexte matériel (le type de dispositif, la taille de l'écran, ...), ainsi que le contexte du logiciel de système d'exploitation, la version, ...
- La session de profil de connectivité :
  Cette session décrit la connexion présentée par l'utilisateur du système (la durée de connexion, la date de connexion, ...).

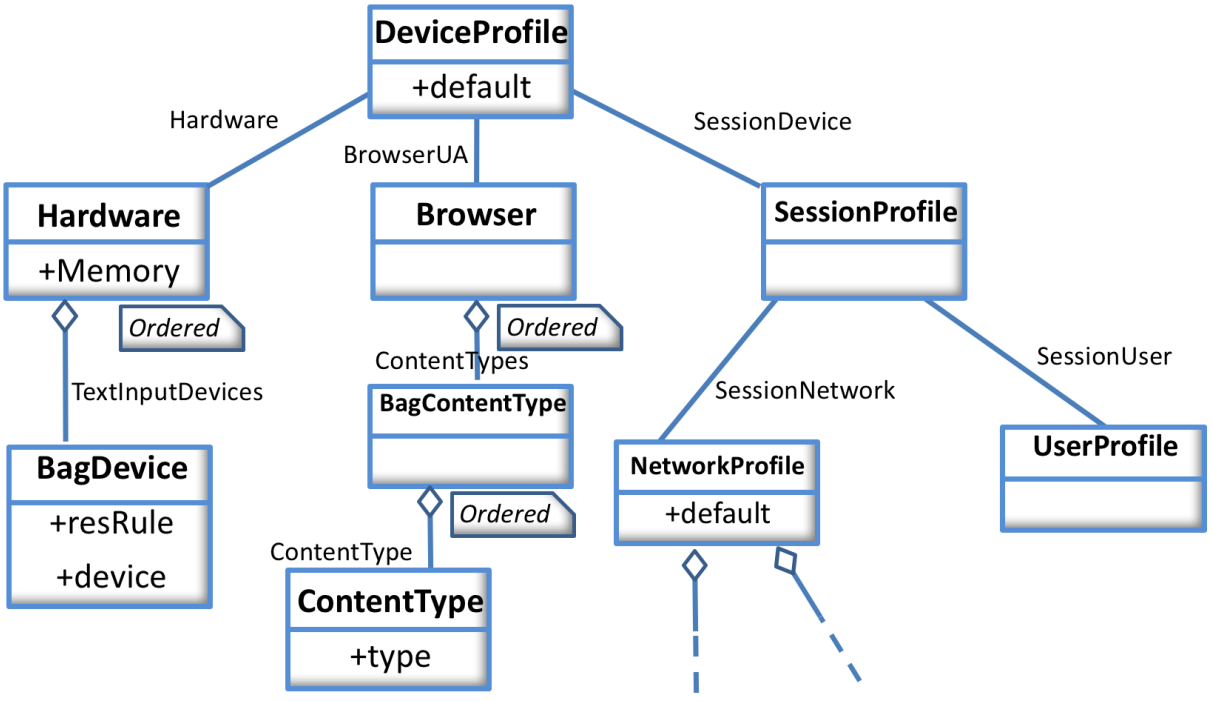

FIGURE 1.10 – Exemple d'une modélisation contextuelle basée sur CSCP

Toutefois, il ne s'agit pas d'un standard international puisque cette proposition ne per-

met 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 peuvent avoir des valeurs différentes selon certaines conditions. Malgré cette extension, ce langage reste non intuitif et difficile à utiliser pour décrire des informations complexes [Ind03].

1.2.6/ LES APPROCHES BASÉES SUR LA LOGIQUE

Pour une modélisation contextuelle basée sur la logique, le contexte est défini comme des faits, des règles ou des conditions qui nécessitent de déduire des faits ou des expressions à partir d'un autre ensemble d'expressions ou de faits. Ces modèles utilisent souvent l'algèbre booléenne et la logique du premier ordre pour une telle tâche de modélisation fondée sur une réification du contexte.

Ce type de modélisation est utilisé souvent dans le domaine de l'intelligence artificielle dans lequel les connaissances sont regroupées en un ensemble des micro-théories. En 1993 , [Mcca93] définit, dans sa première approche de modélisation du contexte basée sur la logique, le contexte comme une entité mathématique abstraite ayant un ensemble des propriétés bien définies. Cette formalisation logique ainsi que ses théories permettent de placer la situation de contexte à un moment donné, selon ses valeurs, en un ensemble des micro-théories.

Or, dans une modélisation contextuelle basée sur la logique, les services d'adaptation à une telle situation peuvent être automatiquement déclenchés ou modifiés à base de l'information de contexte, car cette information est composée généralement d'un ensemble d'actions et de règles (contraintes) bien structurées qui sont implémentées selon des langages de programmations ou bien en utilisant la logique de prédicats. De ce fait, le mécanisme d'adaptation ne peut déclencher des actions que si l'ensemble des contraintes de contexte correspondant à cette action est satisfait.

Certains travaux utilisent la logique à base d'heuristique afin de définir l'ensemble de ces contraintes. Dans [Cassa09], [Loub11], à travers une heuristique de déploiement contextuel, les auteurs ont developpés une nouvelle plateforme nommée *Kalimucho* pour l'adaptation des applications mobiles sensibles au contexte. En effet, *Kalimucho* implémente une heuristique de déploiement contextuel permettant de trouver une configuration satisfaisant les conditions de contexte et de QdS (adaptation dynamique des applications guidée par la qualité de service). *Kalimucho* a été testée avec le modèle de composant Osagaia/Korrontea et plusieurs périphériques. Dans son évaluation, les résultats confirment que *Kalimucho* fournit des adaptations en un temps d'exécution satisfaisant.

Plusieurs formes d'adaptation au contexte peuvent être modélisées [Cha04] :

- La modélisation de l'adaptation architecturale :
  Elle désigne le changement de l'ensemble des faits, en temps réel, dans les interactions entre les composants du système en utilisant un modèle architectural du système.
- La modélisation de l'adaptation comportementale :
  Elle consiste à définir l'ensemble des changements dynamiques dans la phase de l'exécution d'un composant logiciel.
- La modélisation de l'adaptation structurelle :
  Elle consiste en la mise à jour de la structure initiale en préservant son comporte-

ment et ses services.
- La modélisation de l'adaptation de contenu :
  Elle concerne la transformation et la manipulation des contenus en se basant sur les préférences des utilisateurs, les caractéristiques de l'application, et les terminaux utilisés.

Discussion

Bien que les approches de modélisation de contexte basées sur le logique soient très efficaces en termes de raisonnement sur les informations collectées, puisqu'elles utilisent l'algèbre booléenne ou la logique du premier ordre, elles ne permettent pas de décrire la validité temporelle des informations ni les relations qui peuvent exister entre les informations de contexte.

1.2.7/ LES APPROCHES DE SENSIBILITÉ CONTEXTUELLE À BASE DE COLLABORATION ENTRE COMPOSANTS

Une interface intelligente, à travers les environnements ambiants, vise à offrir un espace quotidien intelligent permettant l'accès à l'information ou à des services numériques permettant une utilisation adaptée, naturelle et conviviale de celle-ci. ce genre d'application nécessite souvent une modélisation contextuelle rigoureuse basée sur une collaboration entre les différents composants dans le processus d'interaction.

LE MODELE *Knowledge-User-Presentation*

Un espace technologique adapté est capable de comprendre les caractéristiques de l'environnement, de répondre intelligemment aux demandes ou de réagir de façon appropriée. Cette intelligence est rendue possible, à travers la collaboration entre composants, par la convergence des technologies mêlant objets intelligents, réseaux de communication et interfaces interactives pour fournir de nouveaux services aux utilisateurs.

Dans ce cadre, le modèle KUP (Knowledge, User, Presentation) [Jac06] vise à proposer un modèle théorique basé sur une collaboration entre le différents composants d'une application interactive, pour la spécification et l'implémentation d'une interface intelligente à des utilisateurs mobiles, dans lequel le noyau fonctionnel du système, les utilisateurs et les dispositifs de présentation sont représentés par des entités logiques (agents intelligents). Ce modèle permet de modéliser les différents composants d'une application interactive dans le cadre de l'ambiant entre, d'une part, la fourniture d'une information par le noyau fonctionnel à l'entité utilisateur, et d'autre part, la présentation de cette information par un dispositif adéquat sur demande de l'entité utilisateur.

À travers son système d'interaction à base d'intelligence ambiante, le modèle KUP permet de collecter les informations liées aux utilisateurs lors de leurs déplacements habituels. Ainsi, il permet de garder toute connaissance liée à la mobilité de dispositifs de présentation ainsi qu'à l'environnement d'interaction à travers un ensemble de profils :

- Le profil d'un utilisateur :

Il décrit les capacités et préférences d'un utilisateur donné. Il possède une composante dynamique, mais qui varie lentement.

- Le profil d'un dispositif de présentation :
  Il décrit les capacités du dispositif, en relation avec les contraintes extérieures imposées par l'environnement. Les pannes du dispositif ainsi que la variation des conditions de l'environnement imposent à ce profil une grande dynamicité.
- Le profil d'une unité sémantique :
  Ce profil statique décrit les modalités selon lesquelles l'unité sémantique sait générer un contenu concret. Lors de la présentation d'une unité sémantique donnée, pour un utilisateur donné, sur un dispositif de présentation donné, il est donc nécessaire de mettre en relation ces trois profils.
  Dans KUP, afin de sélectionner une modalité utilisable, C. Jacquet propose dans [Jac07] de construire une taxonomie qui permette de grouper l'ensemble des modalités possible. Cette taxonomie vise à placer de façon arborescente ces différentes modalités (cf figure 1.11). En effet, dans les feuilles, seules les modalités concrètes seront représentées alors que dans les nœuds internes, les modalités abstraites seront représentés. Concernant la racine de l'arbre taxonomique, la modalité abstraite englobe le reste des modalités. Des attributs seront affectés à l'ensemble des modalités qui caractérisent les présentations concrètes qui les utilisent. Ces attributs peuvent avoir des valeurs discrètes (par exemple une police de caractères parmi une liste donnée) ou aussi des valeurs continues (par exemple la taille d'un texte). Finalement, au fur et à mesure de la représentation d'une information selon une modalité donnée, toutes les valeurs de ses attributs doivent être instanciées.

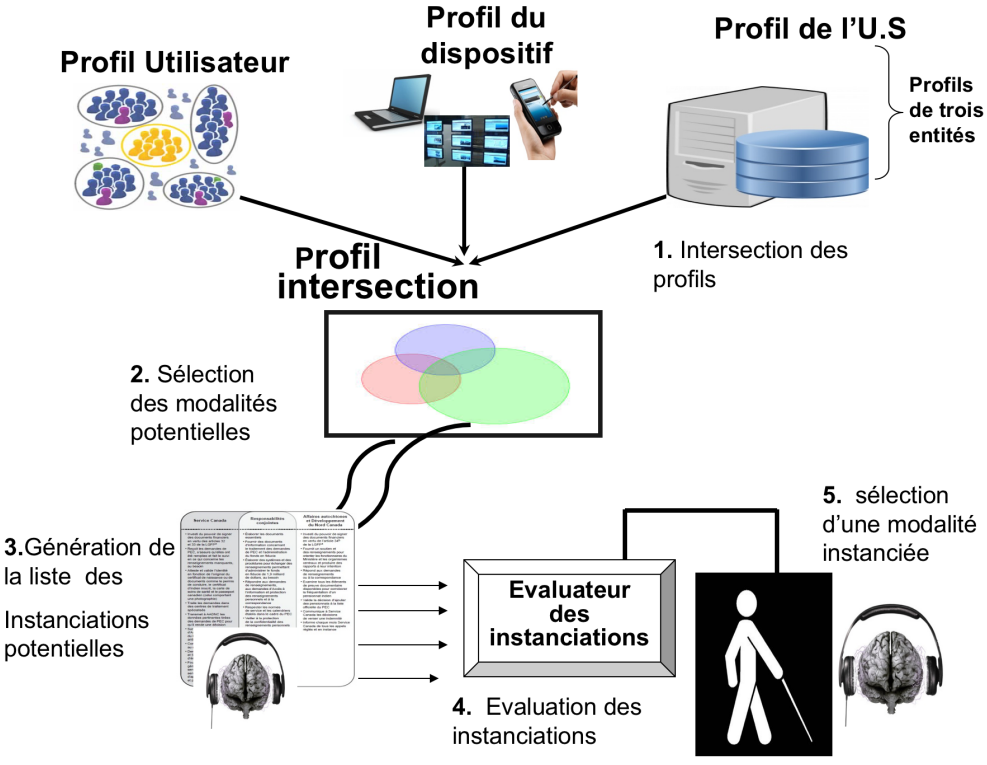

FIGURE 1.11 – Vue synoptique de la méthode de choix d'une modalité à travers les trois modèles de base de KUP

En effet, Lorsqu'un utilisateur se trouve à proximité de dispositifs de présentation, il faut déterminer quel dispositif et quelle modalité utiliser. Un premier algorithme, conçu de façon incrémentale, permet de choisir le dispositif tout en respectant trois contraintes ergonomiques : complétude, stabilité et optimisation de l'espace. Cette dernière contrainte permet à des dispositifs voisins de collaborer de façon à opérer une répartition du contenu à présenter, ce qui permet d'éviter une surcharge des dispositifs. Un second algorithme permet de sélectionner et d'instancier les modalités, en essayant de donner satisfaction aux utilisateurs concernés, et surtout, en évitant d'en défavoriser certains. Cet algorithme est basé sur l'intersection de profils sensori-cognitifs des utilisateurs, du dispositif considéré, ainsi que de l'information à présenter.

Pour qu'une application interactive puisse s'adapter à une situation ambiante, le modèle KUP, dans sa modélisation contextuelle, se base sur deux notions clés à savoir l'espace de rayonnement et l'espace perceptuel :

En effet, l'espace perceptuel d'un utilisateur (ou l'espace de rayonnement d'un dispositif de présentation) est défini en tenant compte des valeurs maximales des paramètres du dispositif (le plus haut volume sonore, la plus grande taille de caractères, ...). Ceci permet de prendre en considération tous les dispositifs potentiels lors de la recherche d'un dispositif de présentation. Si ensuite il n'est pas possible de donner les valeurs maximales à certains paramètres d'un dispositif donné, ce dernier sera tout simplement éliminé par le processus de négociation. La modélisation du contexte dans le modèle KUP est définie à travers l'ensemble des agents. Chaque agent dispose d'un espace de rayonnement, zone géographique dont la forme et la distance dépend de l'agent et qui correspond à la possibilité de cet agent d'interagir avec un autre agent. Lorsque l'agent représente un média par exemple, l'espace de rayonnement est l'ensemble des points de l'espace où ce média peut être perçu (avec le sens adapté au mode employé par le média). L'espace dual de l'espace de rayonnement est appelé espace perceptuel. Il s'agit de l'ensemble des points de l'espace, décrits par rapport à un agent (par exemple l'utilisateur), où ce dernier peut percevoir une autre entité.

- les agents-informateurs (K) : ils correspondent aux sources d'information et ils fournissent des informations aux agents-utilisateurs.
- les agents-utilisateurs (U) : ils correspondent aux utilisateurs et ils connaissent leurs préférences.
- les agents présentateurs (P) : ils correspondent aux dispositifs de présentation.

L'APPROCHE *Ambient Intelligence System for Knowledge*

L'approche *Ambient Intelligence System for Knowledge* (ASK-IT), proposée dans [Span06, Mora07], est constituée d'une architecture multi-agents permettant la gestion des données sensibles au contexte de l'environnement ambiant tout en permettant aux utilisateurs d'accéder facilement aux données et aux services qui les environnent.

En effet, le fonctionnement sous ASK-IT se fait en parallèle du coté client, en fournissant à l'utilisateur des services disponibles, et du coté serveur, en permettant à travers ses agent de spécialiser les réponses aux requêtes en tenant compte des besoins et pré-requis liés aux différentes catégories éventuelles de l'utilisateur. Les données contextuelles manipulées par ces différents agents sont modélisées grâce à une base de connaissance partagée entre tous les agents.

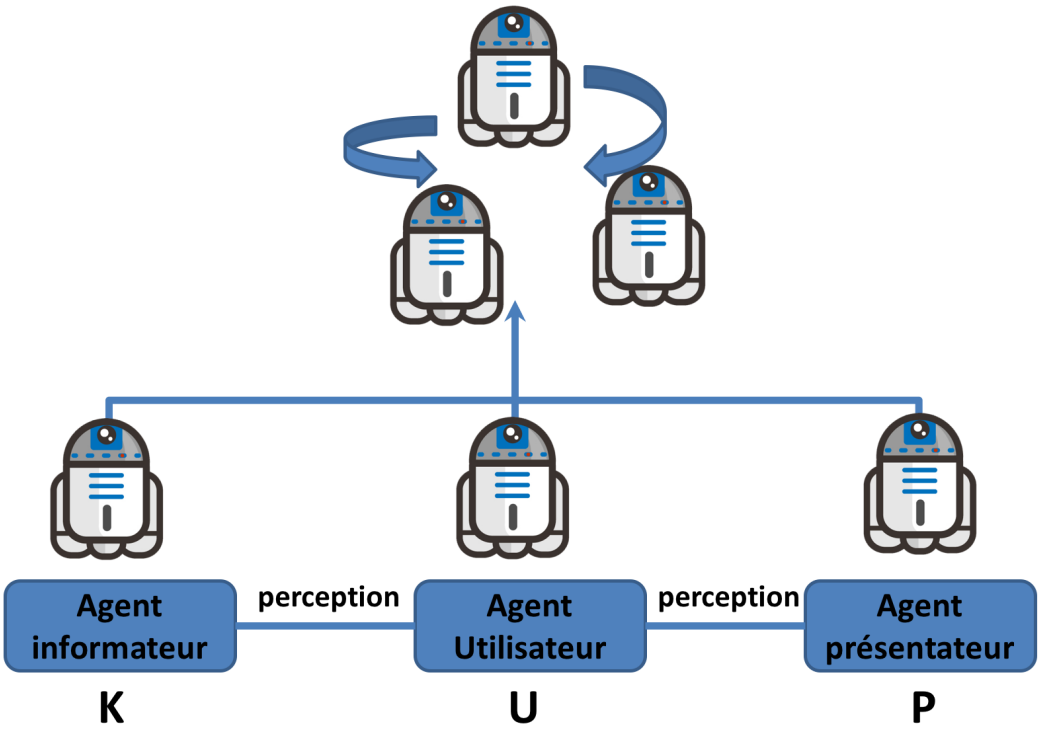

FIGURE 1.12 – La collaboration multi-composants pour l'adaptation au contexte dans KUP

1.2.8/ LA SENSIBILITÉ AU CONTEXTE À BASE DES POLITIQUES SÉMANTIQUES : ONTOLOGIES

La modélisation de contexte s'est faite dans plusieurs travaux de recherche à base des ontologies en utilisant le langage OWL. La représentation de contexte basée sur des ontologies est un ensemble structuré de concepts qui sont organisés dans un graphe dont les relations peuvent être des relations sémantiques ou des relations de composition et d'héritage. Cette ontologie fournit un vocabulaire représentatif lié à un domaine donné [Hef04] : un ensemble de définitions et d'axiomes qui contraignent le sens des termes de ce vocabulaire de manière suffisante pour permettre une interprétation consistante des données représentées au moyen de ce vocabulaire.

En effet, certains outils ontologiques peuvent effectuer des raisonnements automatisés et ainsi apporter des services évolués aux applications telles que la recherche sémantique, le support de décision, la gestion des connaissances et les bases de données intelligentes. Les ontologies sont caractérisées par la possibilité de partager des connaissances entre plusieurs systèmes. En effet, les langages d'ontologies offrent le moyen de publier, d'étendre des ontologies existantes et d'employer différentes ontologies existantes pour compléter une nouvelle ontologie. Dans ce contexte, la gestion des informations de contexte se fait par une modélisation sémantique du contexte, exprimée par le langage OWL. Pour une modélisation de contexte, plusieurs modèles d'ontologies ont été proposés et sont présentés dans la prochaine section. Ces modèles permettent non seulement de modéliser le contexte, mais également de raisonner sur les données décrites.

LE MODÈLE *Context Management Framework*

Le *Context Management Framework* (CMF) [Korp03] se base sur un gestionnaire de contexte centralisé qui communique avec tous les autres composants de la plateforme (cf figure 1.13). En effet, le modèle CMF permet de capturer le contexte à travers le *"ressource server"*, l'interprétation du contexte via le *"context recognition"*, d'où l'utilisation de la logique floue pour atteindre des informations contextuelles de haut niveau. Ainsi, il permet la gestion du contexte via le *"context manager"* et finalement la création de l'application via sa couche applicative. D'une maniére générale, dans CMF, le gestionnaire de contexte récupère les informations contextuelles à l'aide du module du *"ressource server"*, la description de l'environnement de l'utilisateur fait à travers une description dans des profils du contexte.

Puis, l'interprétation de ces informations contextuelles sera faite via un mécanisme basé sur la logique floue pour obtenir des comportements de haut niveau. Il les diffuse à la couche application où cette dernière peut communiquer avec les autres modules de la plateforme.

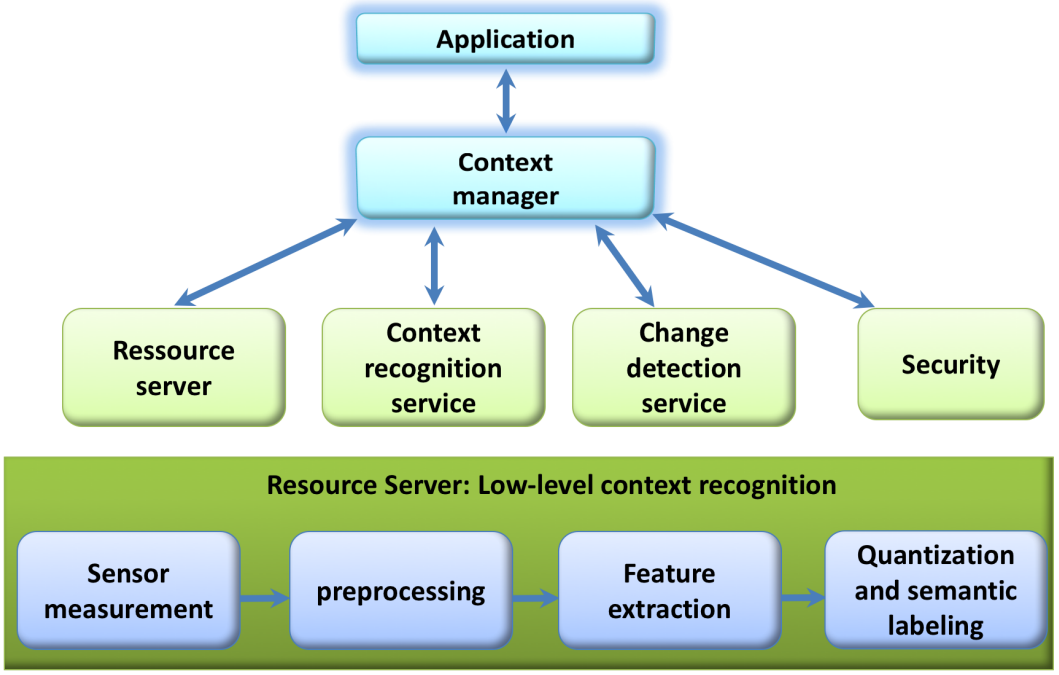

FIGURE 1.13 – L'architecture générale du *Context Management Framework*

LE MODÈLE *Context Broker Architecture*

Plusieurs modèles d'ontologies ont été proposés pour décrire le contexte. Ces approches permettent non seulement de modéliser le contexte, mais aussi de raisonner sur les données décrites. Pour une modélisation de contexte détaillé, l'ontologie CoBrA-ONT (*Context Broaker Architecture*, [Chen03]) se focalise sur la modélisation des contextes et offre au système la possibilité de raisonner sur les informations collectées.

Elle permet de décrire les relations communes et les attributs associés avec les personnes, les endroits et les activités pour une telle application interactive. En effet, cette ontologie a été créée pour décrire le contexte que les applications multi-agents de maison intelligente utilisent. CoBrA-ONT se compose de quatre sous-ontologies qui décrivent des informations de localisation, des informations d'agents (des personnes ou des agents logiciels), la localisation de ces agents et leurs activités. Et dans l'un de ces différents cas, CoBrA-ONT a pour rôle :

- d'acquérir et d'interpréter les informations de contexte,
- de renforcer la communication entre les agents,
- de garder les informations personnelles de l'utilisateur au partage des informations de contexte,
- d'assurer un support de coordination entre les agents.

En utilisant CoBrA-ONT, la création d'applications multi-agents sensibles au contexte est bien possible et chaque capteur (considéré comme un agent) décrit les données qu'il a collectées. Par conséquent, ces données sont partagées entre les agents de chaque application. Ce partage de données permet au système multi-agent CoBrA [Chen04] de raisonner sur les informations de contexte et d'enrichir l'ontologie avec les données qu'il a déduites.

Par exemple, si un capteur détecte la présence d'un badge électronique dans une pièce, le système déduit que le propriétaire du badge se trouve dans la même pièce. Ce raisonnement permet au système non seulement de détecter de nouvelles informations de contexte, mais il lui offre aussi le moyen de résoudre les incohérences des données qui peuvent apparaître. CoBrA-ONT se focalise sur la modélisation des contextes observables et observés utilisés par une application dans le but de les partager entre les agents de l'application. Ce partage offre au système la possibilité de raisonner sur les informations collectées. Cette modélisation ne prend pas en compte la description des situations pertinentes ni les actions d'adaptation. En revanche, CoBrA-ONT permet de décrire les dépendances entre les contextes observables à travers des propriétés ou en utilisant la notion d'héritage. L'avantage majeur de CoBrA-ONT est l'utilisation de l'ontologie qui par définition permet le partage des données, et le raisonnement sur son contenu.

LE MIDDLEWARE *Service Oriented Context-Aware*

Le projet SOCAM (*A Service-Oriented Context-Aware Middleware*) [Gu04] propose un middleware distribué pour la sensibilité au contexte dans les environnements ambiants intelligents. Il vise à convertir les divers espaces physiques (contexte capturé) en un espace sémantique dont le contexte peut être partagé et fourni à des services sensibles au contexte. Pour sa modélisation contextuelle, SOCAM utilise une première ontologie dédiée à l'environnement pervasif qui décrit les informations caractérisant l'environnement d'exécution de l'application dans sa globalité. L'architecture SOCAM comprend une

ontologie spécifique au domaine de l'application ainsi que des autres différents composants :

- Des *context providers* : ils capturent les informations contextuelles de différentes sources hétérogènes de l'environnement de l'utilisateur et de l'application. L'étape qui suit permet la conversion en des représentations OWL pour le partage et la réutilisation de ce contexte.
- Un *context interpreter* : il permet de fournir des services de raisonnement logique basé sur les représentations OWL du contexte en appliquant des enchaînements de règles d'interprétation.
- Une *context Database* : elle permet de stocker les différents éléments de l'ontologie spécifique au domaine de l'application décrivant l'environnement de l'utilisateur.
- Des *context-aware services* : ils utilisent les connaissances déjà stockées dans la base de données de contexte (*context database*) pour modifier leur comportement selon le contexte courant.

LE MODÈLE *CONtext ONtology*

Wang et ses co-auteurs ont proposé, dans [Wang04], une ontologie du contexte CONON codée en OWL pour la modélisation du contexte dans un environnement pervasif et pour le support de raisonnement logique sur le contexte.

CONON fournit une première ontologie du contexte supérieur qui capte les concepts généraux du contexte de base, et fournit par la suite une extension pour ajouter des ontologies spécifiques à un domaine particulier d'une manière hiérarchique. Les concepts généraux concernent généralement les informations de localisation, les informations sur l'utilisateur, les informations sur des activités et les entités de calcul.

La deuxième ontologie est une spécialisation de la première pour un domaine du contexte. Elle constitue un ensemble de sous-classes qui viennent se greffer à l'ontologie de base afin de détailler la modélisation suivant un environnement tel que la maison ou le lieu de travail.

Pour le côté adaptation, CONON ne prend pas en compte la description des règles d'interprétation ni des politiques d'adaptation or, elle se limite à la description du contexte et de certaines règles qui sont écrites grâce à des prédicats associés à la logique de premier ordre. Elles sont introduites pour permettre de caractériser la situation de l'utilisateur. Dans CONON, un téléphone mobile, par exemple, pourrait s'adapter en fonction du contexte de l'utilisateur. Si l'utilisateur, par exemple, est en train de dormir dans sa chambre ou s'il est en train de prendre une douche dans la salle de bain, les appels entrants seront transférés vers la boite vocale du téléphone.

Elle ne modélise pas les capteurs à partir desquels les informations de contexte sont collectées. CONON ne décrit que le contexte et certaines relations existantes entre les informations de contexte en ajoutant la notion de qualité des données.

LA PLATEFORME AMADEUS

Dans la plateforme Amadeus [Gui12, Gui13], le gestionnaire contextuel permet d'assurer la sensibilisation au contexte d'un environnement ambiant. Il lui permet d'adapter son comportement et ainsi d'améliorer le confort de ses utilisateurs. À travers les différents sources de collecte des informations, Amadeus doit acquérir assez d'informations pour anticiper les actions des utilisateurs.

Afin de respecter le critère d'expressivité sémantique, Amadeus fonctionne en exploitant les données sous leur forme la plus générique possible, à savoir directement avec leur valeur numérique, ce qui lui permet de prendre en compte la majorité des données. Cependant, rien n'empêche que ces données soient malgré tout enrichies d'une sémantique qui permettrait éventuellement de réaliser des pré-traitements. Tandis que pour la proactivité, les entités dans le gestionnaire Amadeus sont basées sur ses capacités à observer les actions des utilisateurs afin de déterminer quelle action réaliser face aux différentes situations qui se présentent.

LA PLATEFORME ATRACO

Le système du projet ATRACO, décrit par [Mink10] cherche à établir l'activité dans laquelle est engagé un utilisateur, afin de créer une *"sphère d'activité"* composée des ressources (dispositifs, connaissances, ...) pour l'assister dans cette activité. En effet, le système ATRACO modélise l'ensemble des connaissances grâce à différentes ontologies, tout en intégrant des mécanismes d'alignement entre ces différentes ontologies. Après, dans la phase de traitement, l'ensemble de ces connaissances sera basé sur la logique floue afin d'observer les actions des utilisateurs pour en déduire un comportement à attribuer aux dispositifs du système ambiant.

LA PLATEFORME PIVON : *Pervasive Information Visualization Ontology*

Comme le contexte est un concept large, imprécis et non délimité, R. Hervas et al [Herv10] dans le modèle PIVon ont proposé un modèle de contexte générique et adaptable en utilisant quatre principaux modèles ontologiques qui se basent sur le niveau supérieur d'abstraction du contexte (méta-contexte) comme le montre la figure 1.14.

Ce modèle générique s'appuie sur une classification plus générale centrée utilisateur, environnement, dispositifs et services. Or, l'emplacement utilisateur, l'identité de celui-ci, l'activité et le temps sont des propriétés de ce modèle. Par exemple pour décrire l'ontologie de l'utilisateur, le module *user* a été conçu pour poser trois questions fondamentales : 1. quelles sont les caractéristiques de l'utilisateur ? 2. que souhaite-t-il faire ? 3.et que fait-il ?

Pour adapter dans PIVon, les auteurs se sont basés sur des moteurs de raisonnement pour obtenir de nouvelles informations à travers l'information précédente et donc rendre l'architecture sensible au contexte.

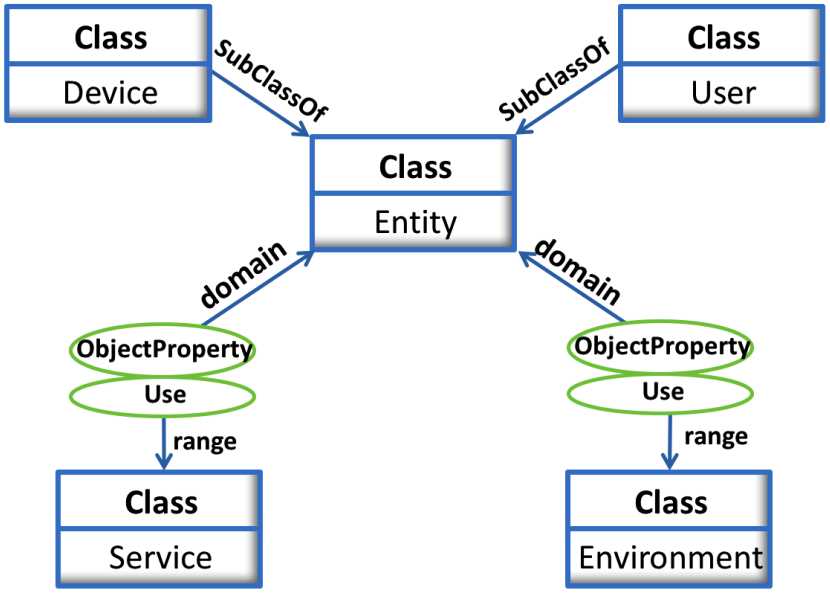

FIGURE 1.14 – Niveau supérieur du contexte

LE MODÈLE COIVA : UNE EXTENSION DE PIVON

L'extension COIVA (*Context-aware and Ontology-powered Information Visualization Architecture*) de PIVon [Herv11] exploite l'application de la sémantique du contexte dans l'adaptation de l'exécution, notamment pour les services dans des environnements intelligents. Les services contextuels sont généralement concentrés sur leurs propres informations sans prétentions d'interopération. Il est nécessaire pour permettre à des modèles et des systèmes de contexte commun de rendre les applications sensibles au contexte interopérable.

En outre, les systèmes de gestion de contexte ont mis en œuvre des mécanismes d'interprétation pour soutenir le comportement dynamique des utilisateurs et de leur environnement, y compris les techniques d'adaptation à leurs futurs besoins, afin de maintenir les informations de contexte au moment de l'exécution et de l'interopérabilité avec les modèles contextuels externes.

En effet, pour adapter COIVA, La sémantique formelle exploitée permet l'utilisation du raisonnement décidable comme un mécanisme puissant pour déduire de nouvelles informations de contexte. Cette sémantique formelle permet de surveiller les règles de raisonnement par le gestionnaire dynamique du contexte, qui prend en charge des réactions à chaque changement de contexte. Ensuite, dans [Vill14] une nouvelle application dans le cadre de modélisation basée sur les ontologies permet le suivi des patients par le biais de la biométrie et d'appareils mobiles pour les maladies chroniques.

La contribution dans cette application est fonction de trois composants qui permettent le développement semi-automatique et indépendant de la maladie cible et adaptable aux besoins des utilisateurs particuliers. Tout d'abord, une modélisation à base d'ontologies classe les éléments médicaux tels que les maladies, les recommandations, les préventions, les aliments, les appareils mobiles et les suggestions de régime. La deuxième tâche consiste à la distribution des dispositifs à couches, ce qui permet la génération d'applications finales distribuées dans un contexte médical. La troisième et la

plus importante partie consiste à développer les modèles en un ensemble de *MobiPatterns*. Un *MobiPatterns* définit le schéma de chaque module de l'application qui fait partie de l'application finale. Ces modules comprennent des modèles formels qui cherchent à découvrir les principes caractéristiques ainsi que les algorithmes essentiels qui doivent être revues dans le cadre prévu par la mobilité. Par exemple, le *MobiPatterns* est responsable de la génération du profil du patient. Le profil associé à la maladie du patient doit présenter la structure du module de profil du patient et sa relation avec le *MobiPatterns* pour générer le profil en cours.

Discussion

Dans ces différentes approches à base des politiques sémantiques, on retrouve un point commun sur la séparation de l'acquisition du contexte d'une part et de son utilisation d'autre part (cf table 1.2).
COBRA permet de modéliser les données contextuelles en un ensemble de modules partant de la phase d'acquisition jusqu'à la phase de la création de la base de connaissance tout en respectant la séparation entre ces différents modules. Par rapport au projet SOCAM, ce modèle utilise une approche de collecte et de gestion du contexte : les capteurs sont encapsulés par des services web, puis la communication avec l'interpréteur du contexte est réalisée par un échange d'évènements en utilisant des représentations OWL des information échangées tout en respectant l'aspect de séparation entre la phase d'acquisition et la phase d'utilisation. Voici un tableau comparatif de différentes approches de la modélisation contextuelle à base des ontologies.

	Type d'architecture	Modèle de contexte	Interprétation du contexte	Stockage du contexte
CoBra	Basées sur des Widgets	Ontologie (OWL)	Base de connaissances	Distribué
SOCAM	Middleware distribué	Ontologie (OWL)	Moteur d'inférence	Base de données
CMF	Gestionnaire de contexte	Ontologie (RDF)	Service D'interprétation	Non disponible

TABLE 1.2 – Comparatif de trois plateformes à base d'ontologies

1.3/ LA MÉTHODOLOGIE D'ADAPTATION AU CONTEXTE

Adapter, c'est le fait de rendre une application interactive adaptée aux variations pertinentes du contexte. Pour ce faire, quelques méthodologies d'adaptation ont été envisagées dans la littérature. En effet, une première étape d'adaptation au contexte est de définir les éléments constituant l'environnement de travail. Ces derniers définissent les

connaissances régularisant cet environnement ainsi les paramètres liés aux références et aux préférences de l'utilisateur. Généralement, tous les systèmes sensibles au contexte passent par les étapes méthodologiques suivantes :

- L'identification de toutes les actions du système sensibles au contexte : celles qui représentent la fonctionnalité du système, et qui sont souvent supposées indépendantes du contexte.
- La définition des points d'adaptations : comme les régularités, les paramétres de références et les préférences.
- La définition des éléments pertinents de contexte.
- Le choix pour chaque ensemble de contexte, des actions qui répondent principalement aux objectives d'adaptation, ainsi que de la configuration de contexte.
- La construction des règles d'adaptation : suite aux différentes configurations extraites et des actions, il faut déterminer l'ensemble des règles d'adaptations.
- La manipulation des actions selon les règles d'adaptation.

L'adaptation est une modification d'une application en réponse à un changement de la situation de son contexte. Cette modification consiste soit à changer la structure de l'application, soit son comportement. Un adaptateur au contexte constitue un élément fondamental d'un système sensible au contexte. L'adaptateur au contexte définit un ensemble de changements qui sont décrits à partir d'un ensemble de règles. Ces règles sont définies à partir de la description de l'environnement. Plusieurs travaux ont été réalisés dans ce sens. Dans [Bouas09] par exemple, les auteurs ont proposé une nouvelle approche de modélisation multi-niveaux pour l'adaptation des systèmes ubiquitaires collaboratifs. En effet, dans cette approche les auteurs ont identifié une suite de niveaux d'abstraction pertinents et ont défini l'ensemble des outils et des règles de transformation "entre-niveaux" (*inter-level*) afin de fournir l'adaptation aux changements de contexte. Cette adaptation est guidée par les exigences de haut niveau et les contraintes de bas niveau.

Cependant, La construction des règles d'adaptation constitue un grand défi.

Pour mieux adapter, la stratégie d'adaptation est généralement reconnue comme l'élément le plus important d'un système adaptatif, même si les opérations disponibles sont extrêmement puissantes et si le contexte est connu dans ses moindres détails, tout cela ne sert à rien si la stratégie n'est pas capable de tirer partie de ces informations. Dans les applications interactives sensibles au contexte, la stratégie d'adaptation se présente sous la forme d'un ensemble d'algorithmes et de données qui sont chargés, à partir des informations connues sur le contexte d'utilisation, de décider quand et comment adapter ces applications, en utilisant au mieux les mécanismes de reconfiguration disponibles. Elle est située au cœur des applications adaptatives, puisqu'elle fait le lien entre les mécanismes de reconfiguration et les informations contextuelles.

La conscience de l'environnement et de ce qui l'entoure regroupe la conscience sur les connaissances qui régularisent cet environnement de travail. En effet, ces connaissances apportent des exigences sur les interactions du système dans cet environnement [Agre01]. Une description explicite et implicite doit être définie au début sous forme de conscience. Or, cette conscience constitue le modèle interne de l'environnement de travail qui doit contenir particulièrement des pré-connaissances sur :

- Les régularités :
  Elles sont constituées de l'ensemble des conditions, des normes et des valeurs de l'environnement de travail. Ces régularités limitent les interactions d'une applica-

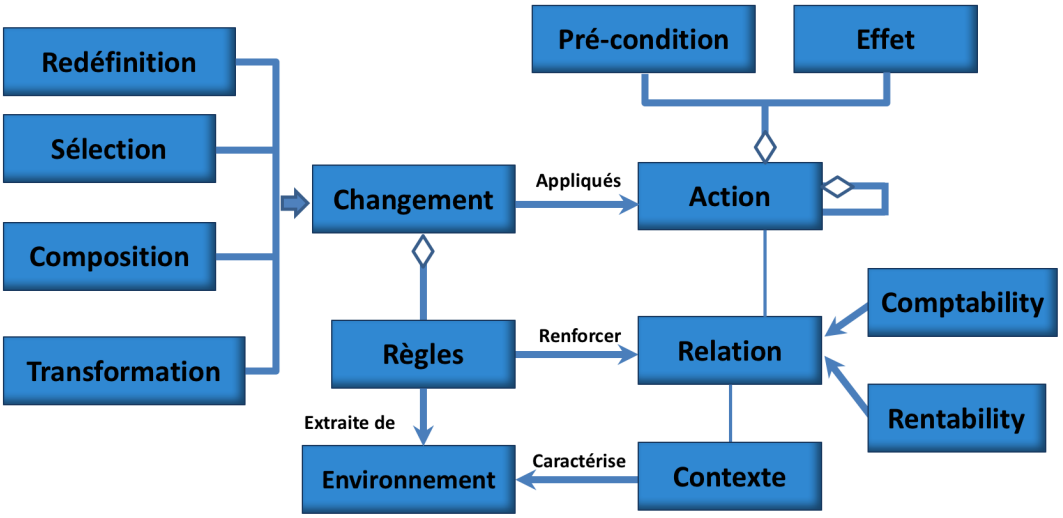

FIGURE 1.15 – Modèle de l'adaptateur au contexte

tion d'une situation à une autre. Par exemple, dans l'environnement de travail en milieu hospitalier, une régularité, définie par cet environnement est qu'un médecin dans une salle de chirurgie ne peut pas utiliser l'outil de message instantanée pour communiquer avec ses collaborateurs, car sa situation limite ses mouvements et qu'il a besoin de plus de concentration sur son travail en cours.

- La référence physique :
  C'est l'ensemble des descriptions des paramètres physiques des ressources composant un environnement de travail. Par exemple, le niveau de batterie d'un terminal et le type d'images supporté. Ces paramètres de référence peuvent changer continuellement des valeurs.
- Les préférences de l'utilisateur :
  Ces dernières regroupent un ensemble d'options imposées par l'utilisateur lui-même.
- La Compatibilité physique :
  La compatibilité désigne aussi les caractéristiques physiques de l'environnement. Un système sensible au contexte a pour but d'établir une compatibilité de ses actions avec les règles sociales et physiques définies dans un environnement de travail [Harr06]. Par exemple, une adaptation de contenu est nécessaire pour l'adaptation de contenu des documents aux caractéristiques physiques d'un terminal.
- La Rentabilité :
  La sensibilité au contexte est aussi destinée à maximiser la rentabilité d'un système dans un environnement de travail.

Pour un système sensible au contexte, les changements du système regroupent toutes les opérations qui peuvent être soumises aux actions du système. Or, une action du système est définie par des pré-conditions et leurs effets et qui peut concerner des données de sortie telle qu'une image ou bien un changement d'état d'une entité de l'environnement. Une pré-condition représente les conditions qui doivent être valides pour l'exécution de l'action. Ces conditions sont exprimées généralement en fonction des paramètres de l'environnement.

La manipulation d'une action se fait par la redéfinition des pré-conditions (cf figure 1.15) ou bien par la modification de l'effet de l'action [Urbi09]. Une redéfinition des pré-

conditions est une manipulation au niveau de la construction alors que la redéfinition de l'effet est une manipulation au niveau de l'exécution de l'action. La manipulation des actions désigne l'ensemble des opérations qui peuvent être regroupées soit selon la sélection qui a pour effet de répondre au mieux à une exigence définie par une description donnée de l'environnement, soit la composition de plusieurs actions simples pour construire une action composée.

1.4/ LES FORMES D'ADAPTATION AU CONTEXTE

1.4.1/ LES TYPES D'ADAPTATION

Pour remédier au caractère incertain de contexte, dans la phase d'adaptation, il est important d'adopter un certain niveau de logique (la logique floue, la logique probabiliste . . . ou bien un apprentissage de comportement). En effet, l'adaptation au contexte est définie comme étant l'ensemble des mécanismes de réactions prévu suite aux changements de contexte, ainsi cette adaptation se base sur un ensemble de règles prédéfinies. Ces règles sont implémentées selon des langages de programmations ou bien en utilisant la logique de prédicats. À ce stade, plusieurs formes d'adaptation peuvent se présenter [Raib08] :

```
-----------------------------------------------------------------------------------------

Fonction : préparer AdaptationContenu (modelAdaptation FM = (f0, F, T))
Begin
  For each (fi ∈ F)
    Begin
      AdaptationContenu ac = instancierAdaptationContenu(fi)
      modelAdaptationAdapte = actualiserAdaptateur(modelAdaptationAdapte,fi,ac)
    Endfor
  Return (modelAdaptationAdaptee)
End

-----------------------------------------------------------------------------------------
```

FIGURE 1.16 – Algorithme général de la préparation de l'adaptation de contenu [Berhe05]

- Adaptation compositionnelle (en temps réel) : elle permet de désigner l'ensemble des modifications liées à la structure et au comportement de l'entité logiciel tout en proposant une réponse aux changements survenus à son environnement d'exécution ;
- Adaptation architecturale (en temps réel) : elle concerne les changements effectués dans l'un des composants du système ou bien dans les interactions entre eux en utilisant un modèle architectural bien défini du système ;
- Adaptation structurelle : elle signifie le changement dynamique de type des composants de l'application. En effet, ce type d'adaptation consiste en la mise à jour de sa structure en préservant son comportement et ses services ;
- Adaptation comportementale : cette forme d'adaptation permet de désigner les changements dynamiques dans la phase de l'exécution d'un composant logiciel d'une manière non intrusive ;

- Adaptation de contenu : elle désigne toute transformation et toute manipulation des contenus concernant l'information de contexte en se basant sur les caractéristiques de l'application et des terminaux utilisés à un moment donné.

1.4.2/ LES ALGORITHMES D'ADAPTATION DE CONTENU

Pour la planification d'adaptation de contenu, il existe plusieurs modules d'adaptation qui ont été utilisés dans le processus d'adaptation. En effet, dans les travaux de G. Berhe, l'adaptateur de contenu fondé sur le moteur d'adaptation des données à base des modules permet de préparer à l'adaptation de contenu à travers l'utilisation d'un algorithme général de la préparation de l'adaptation de contenu (cf figure 1.16.

L'instanciation de l'adaptateur à travers l'algorithme général d'instanciation des adaptateurs de contenu (cf figure 1.17) permet l'adaptation de données à leur contexte d'utilisation. Pour chaque service à adapter par l'application, le module d'adaptation sélectionne les opérateurs d'adaptation de données nécessaires pour adaptater le contenu à leur contexte d'utilisation.

```
---------------------------------------------------------------------------------------------------------

Fonction : instancier adaptationContenu (DescriptionService f)
Begin
   AdaptationContenu ac = new AdaptationContenu (f)
   For each (c_j ∈ f)
      Begin
         planAdaptation = planLocal (c_j)
         ac.descriptionUpdate (planAdaptation, c_j)
      Endfor
   Return (ac)
End

---------------------------------------------------------------------------------------------------------
```

FIGURE 1.17 – Algorithme général d'instanciation des adaptateurs de contenu [Berhe05]

Les algorithmes dans les travaux de G. Berhe, L. Brunie, JM. Pierson [Berhe05] permettent d'aboutir à un résultat adapté au contexte d'utilisation pour un media donné (audio, vidéo, ...). Ainsi, le processus d'adaptation de contenu permet de fournir la possibilité de garder le résultat généré et de le mettre en cache local pour servir à la réutilisation une prochaine situation contextuelle identique.

Synthèse du chapitre

Nous avons proposé, dans ce premier chapitre, un état de l'art des les approches de la sensibilité au contexte les plus utilisées. Dans le but d'étudier la possibilité d'utiliser ces approches dans un environnement sensible au contexte, nous avons étudié ces approches selon la richesse et la qualité des informations qu'elles permettent de décrire, leur degré de formalisme, et la possibilité de leur utilisation dans différents types de systèmes.

- Commençant par les modèles architecturales classiques, la sensibilité au contexte des applications interactives présente de nombreuses difficultés d'utilisation dans le monde réel. Les approches paires/triplets sont caractérisées par une pauvreté d'expressivité et la simplicité des données qu'elles représentent. De ce fait, elles ne permettent pas de décrire les observables, ni les relations entre les informations de contexte, ni les règles d'interprétation.
- Les approches de sensibilité contextuelle à base de collaboration entre composants permettent de décrire des relations entre les informations de contexte à travers une collaboration entre composants, mais leurs applications aux systèmes existants restent limitées : cela est dû à la difficulté de description qu'engendre ce type d'approche.
- Les approches orientées modèle sont des approches prometteuses car elles utilisent non seulement un modèle formel pour décrire le contexte, mais elles offrent aussi un méta-modèle de description qui peut être réutilisé par plusieurs a pplications. Les approches orientées modèle existantes ne prennent pas en compte la description des règles d'interprétation de contextes de haut niveau, ni la modélisation des capteurs. Ces approches offrent une possibilité d'extension pour permettre la description de nouveaux types de relations entre les informations de contexte et ainsi la possibilité d'intégrer les notions qui manquent.
- Les approches basées sur la logique sont des approches formelles. Ces approches permettent de raisonner sur les informations de contexte pour déduire de nouvelles valeurs du contexte ou pour générer des réactions au niveau de l'application ou du système. En pratique, ces approches orientées ontologies sont des approches formelles qui tirent parti des caractéristiques des ontologies pour modéliser le contexte. En effet, les caractéristiques de partage et de distribution des données ont été exploitées afin de définir des méta-modèles de description du contexte. De plus, les moteurs d'inférence fournis par les ontologies ont été utilisés pour déduire des contextes de haut niveau à partir des données collectées.

L'objectif de l'étude que nous avons effectuée consiste à montrer l'apport de chaque approche de modélisation de la sensibilisé au contexte dans le but d'utiliser l'une d'elles pour modéliser les informations de contexte associées aux applications sensibles au contexte.

Ainsi l'objectif du chapitre suivant est une étude comparative des différentes approches les plus expressives et les plus prometteuses pour la description du contexte dans un environnement sensible au contexte.

2

CLASSIFICATION DES DIFFÉRENTES APPROCHES DE LA SENSIBILISATION AU CONTEXTE

INTRODUCTION

Avec les systèmes d'information pervasifs, les équipements d'interactions informatiques communiquent et collaborent ensemble en percevant le contexte global et en réagissant proactivement (sans intervention explicite de l'utilisateur) afin de fournir des services adaptés à l'utilisateur et aux applications. Le fait d'ajouter la sensibilité au contexte à une application interactive dans un environnement pervasif revient à décrire les informations associées à ce contexte et à sa gestion. En effet, la sensibilité s'appuye sur les informations fournies par les modèles de contexte qui se chargent de décrire le contexte associé à ces applications. À ce stade, les modèles à base d'ontologies sont les moyens de représenter le contexte des environnements pervasifs en formulant l'ensemble des connaissances dans ce domaine.

Nous commençons ce chapitre par une première section qui permet de décrire l'étude comparative entre les différentes approches de la sensibilité au contexte déjà citées (qui proviennent de la littérature), puis dans une deuxième section nous décrivons le modèle particulier PIVOn (*Pervasive Information Visualization Ontology*) construit à base de quatre modèles d'ontologies. Nous finissons le chapitre par une dernière section dans laquelle nous proposons une classification des approches à base d'ontologie.

2.1/ NOTRE ÉTUDE COMPARATIVE

2.1.1/ SUIVANT LES CRITÈRES DE COMPARAISON

Après une étude des différentes approches existantes de la sensibilité au contexte, la comparaison s'est faite sur deux critères fondamentaux : la modélisation des éléments de contexte et les paramètres d'adaptation au contexte.

A ce niveau, nous prouvons la nécessité d"utilisation de ce deux critères tout en référents aux différents rôles de chacun dans la réponse aux principaux objectifs de modélisation de contexte lors de la conception d'une application interactive pervasive. En ce qui

concerne les éléments de contexte, la modélisation de la sensibilité au contexte se base sur la définition de ces éléments constituants le contexte courant. Or, les éléments de contexte représentent tous les composants physiques et organisationnels de ce contexte, qui permettent à mieux informer sur les situations courantes contextuelle. En effet, répondant à la définition du contexte, l'élément de contexte en réalité caractérise une entité de l'application. Une cohérence entre cet élément qui a déjà définit comme l'entité de l'application et l'information de contexte est établie dans l'application, qui permet d'apporter l'information la plus directement utilisable en termes de prise en compte du contexte. Si nous prenons l'exemple de l'élément « Localisation » dans les applications mobiles pervasives, nous remarquons que son utilisation comme un élément de contexte est nécessaire et indispensable afin dons le renseignement sur la mobilité dynamique d'utilisateurs. Ainsi, pour modéliser le contexte d'utilisation des systèmes pervasifs, tous les éléments constituant ce contexte se réunissent afin qui peuvent être utilisés pour caractériser la situation courante contextuelle. Pour les paramètres d'adaptation au contexte, la conception des applications pervasives nécessite la prise en considération de l'ensemble de paramètres d'adaptation pour que ces applications puissent être utilisées en adaptant le contexte qui l' entoure au moment de l'utilisation de celui-ci sur des terminaux, par des utilisateurs, dans des environnements et emplacements spécifiques.

LE PREMIER CRITÈRE : LA MODÉLISATION DES ÉLÉMENTS DE CONTEXTE

La modélisation des éléments de contexte est une des caractéristiques importantes pour favoriser et améliorer la sensibilité au contexte dans les environnements pervasifs. En effet, cette modélisation est considérée comme une étape indispensable pour la conception et le développement des systèmes interactifs dans ces environnements. La modélisation des éléments de contexte consiste à traiter et à analyser l'ensemble des informations contextuelles contenues dans un domaine précis sous formes d'une représentation abstraite, soit au niveau de la structure des données, soit au niveau sémantique. Ainsi, un contexte doit être bien modélisé à travers ses éléments sous une forme appropriée afin de favoriser son utilisation, son partage, sa réutilisation et donc son adaptation aux objets de l'environnement.

Le trois tableaux que nous présentons (tables 2.1, 2.2 et 2.3) sont le résultat de notre étude comparative des différentes approches de la sensibilisation au contexte. Nous commençons dans le premier tableau (cf table2.1) par une classification de l'ensemble des approches suivant qu'elles ont des approches ontologiques ou non.

Cette classification nous permet de regrouper les approches selon leurs points communs. Nous commençons dans ce premier tableau, afin de traiter le premier critère lié à la manière de décrire et de modéliser les informations de contexte.

Un modèle de contexte utilise une structure de donnée pour représenter les informations de contexte. Dans notre étude, nous proposons une comparaison entre les différentes approches au niveau de la structuration des données suivant qu'il sagit d'une structure simple (comme dans les approches attribut/valeur) ou d'une structure complexe (comme dans les approches à base d'ontologies à travers la définition de concepts, de sous classes, . . .).

Prenons, tout d'abord, l'exemple de la structuration des données de contexte dans les approches paires/triples (attribut, valeur)/(attribut, valeur, degré) : l'auteur propose l'utilisation de la structure des données la plus simple pour la modélisation des informations

Approches	App non Ontologiques	App Ontologique	Déf de structure complexe
Paires/Triples	Oui	Non	Non
Approches formelles	Oui	Non	Oui
Approches logiques	Oui	Non	Non
HYDROGENE	Oui	Non	Oui
Approches profils	Oui	Non	Non
KUR	Oui	Non	Oui
ASK-IT	Oui	Non	Non
CoBra	Non	Oui	Oui
AMADEUS	Non	Oui	Oui
CONON	Non	Oui	Oui
SOCAM	Non	Oui	Oui
ATRACO	Non	Oui	Oui
PiVon	Non	Oui	Oui
COIVA	Non	Oui	Oui
OCAAR	Non	Oui	Oui

TABLE 2.1 – Étude comparative sur la définition des structures de données

contextuelles. L'attribut représente un élément de contexte comme la localisation de l'utilisateur. La valeur est la valeur actuelle de cette information. Le degré représente la certitude de cohérence. Cette représentation est caractérisée par sa facilité d'implémentation. Cependant, elle manque d'une force d'expressivité et ne permet pas de présenter les relations entre les éléments de contexte.

Prenons, maintenant, l'exemple d'une approche basée sur les ontologies : dans CONON, la structure des données est représentée sous forme de concepts qui sont organisés dans un graphe dont les relations peuvent être des relations sémantiques ou bien des relations de composition et d'héritage. À travers une structure de données hiérarchique, un vocabulaire représentatif peut être facilement fourni et permet l'interprétation des informations contextuelles.

Dans le deuxième tableau (cf table2.2) ,nous commençons notre étude en désignant l'ensemble des éléments de contexte qui ont été le plus souvent utilisés dans les approches de la sensibilité de contexte. Un élément du contexte est celui qui décrit les points d'adaptations. Il fait partie des descriptions des caractéristiques de l'environnement qui décrit des paramètres des références ou bien des préférences. La première étape consiste à étudier la sensibilité au contexte et à définir les éléments constituant l'environnement en cours. Ces éléments servent après dans la phase d'adaptation. Ils définissent les connaissances régularisant un environnement pervasif et les points seuils de cet environnement (les paramètres de références et les préférences de l'utilisateur).

L'ensemble des éléments de contexte est une conjonction de plusieurs sous ensembles : Utilisateur, Physique, Organisationnel, et Spatio-Temporel.

Pour chaque élément de contexte, $C_1 = \{ U_1, P_1, O_1, ST_1)$ définit l'ensemble des éléments

Approches	Description des éléments sensibles au contexte						
	User	Envir^{nt}	Terminal	Localisation	Service	Activité	Temps
Approches non ontologiques	Oui	Oui	Oui	Non	Non	Non	Oui
Approches ontologiques	Oui	Oui	Oui	Oui	Oui	Oui	Oui

TABLE 2.2 – Étude comparative des éléments du contexte

de ce contexte :

Pour chaque élément de contexte, définit le sous-ensemble de cet élément de contexte. Prenant l'exemple suivant de le sous-ensemble de l'élément de contexte spatio-temporel qui est défini par :

P($_{ST}$) = { { C$_{ST1}$ } , { C$_{ST2}$ } } tels que C$_{ST1}$= **Location** et C$_{ST2}$=**Time**.

Dans le cas d'une application de télédiagnostic le contexte **Localisation** peut avoir plusieurs valeurs comme par exemple :

C$_{ST1}$=(CHIRURGICAL ROOM, CONSULTATION ROOM , OFFICE)

Chaque travail sur la sensibilité au contexte utilise une suite des éléments pour le définir et le représenter : notamment le temps, le réseau, la localisation, l'utilisateur, le terminal, l'environnement, l'activité et la tâche. Ces éléments sensibles au contexte sont exprimés dans un modèle de contexte qui intègre un modèle de donnée. Suite à notre étude comparative (cf table2.2), nous remarquons que les approches à base d'ontologies sont les plus complètes en terme d'éléments pris en compte dans la sensibilité au contexte, alors que les autres approches n'utilisent que quelques éléments.

Finissant par le troisième tableau (cf table2.3), nous comparons la modélisation de la sensibilité au contexte dans les différentes approches à travers les langages de modélisations utilisés par chacune d'entre elles. En effet, les approches de la sensibilisation à base des ontologies font appel à une modélisation sémantique du contexte, exprimée par un langage formel (OWL, RDFs) pour la description des différents éléments situés dans ce contexte et ce n'est pas le cas des approches non ontologiques. La représentation du contexte, en particulier au moyen des langages du Web sémantique peuvent fournir une richesse et une définition rigoureuses des éléments de contexte.

Approches	Langage de modélisation							
	CC/PP	OWL/RDF	UML	CA-IDL	CML	UML	ORM	XML
Approches non ontologiques	Oui	Non	Oui	Non	Non	Oui	Non	Non
Approches ontologiques	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui

TABLE 2.3 – Étude comparative des langages de modélisation

Discussion

À ce stade, nous pouvons conclure que les méthodes les plus intéressantes pour la modélisation de la sensibilité de contexte sont celles fondées sur les ontologies. Elles permettent, tout d'abord, de capturer une grande variété d'éléments de contexte surtout dans un environnement informatique omniprésente. Aussi, les ontologies permettent la définition d'un vocabulaire commun des informations de contexte à travers un langage formel, flexible et extensible afin de permettre l'interopérabilité sémantique. Les ontologies permettent de fournir un raisonnement efficace sur les connaissances de contexte.

LE DEUXIÈME CRITÈRE : LES PARAMÈTRES D'ADAPTATION AU CONTEXTE

La conception des applications interactives dans des environnements pervasifs nécessite la prise en considération de l'ensemble de paramètres d'adaptation pour que ces applications puissent être utilisées sur des terminaux, par des utilisateurs, dans des environnements et emplacements spécifiques. De plus, ces applications doivent faire face au dynamisme de changement de contexte d'utilisation pour réaliser des activités et atteindre les objectifs appropriés des utilisateurs. Pour garantir une utilisation confortable et adaptable des applications interactives dans les environnements pervasifs, beaucoup de paramètres entrent en jeu :

- L'adaptation par rapport aux éléments du contexte :
  L'application doit assurer un niveau d'adaptation par la prise en considération d'un grand nombre d'éléments qui peuvent être sensibles au contexte (utilisateur, terminal, environnent, . . .). Le comportement de ces applications interactives doit être en corrélation avec les capacités matérielles et logicielles de l'ensemble des composants du contexte (la diversité des terminaux mobiles, la diversité des préférences d'utilisateurs, . . .) ;
- Le raisonnement logique :
  Ce paramètre permet la déduction et la prédiction des nouvelles informations du contexte ;

- Le type d'adaptation de l'information :
  Généralement le type d'adaptation de l'information au contexte est déterminée selon l'acquisition de l'information de contexte et son adaptation. En effet, l'adaptation au contexte est considérée soit comme étant une réaction ou comme une intégration. Dans [El11], l'adaptation au contexte est considérée comme étant une réaction à l'aspect dynamique de l'environnement qu'elle permet d'associer directement une perception d'une information de contexte à une action d'adaptation qui sera efficace à la sensibilité au contexte. Une adaptation réactive ne possède pas une représentation interne du modèle de l'environnement, ainsi elle ne fournit pas une méthodologie générale d'adaptation au contexte.
  Dans [Harr06], une adaptation par intégration est réalisée à travers une représentation interne d'un modèle de contexte en montrant la structure des interactions dans un environnement donné. Le processus d'adaptation dans le cas d'intégration permet non seulement de donner une structure générale, mais aussi d'imposer le respect de l'ensemble des règles et des propriétés liées aux actions de l'environnement pervasif ;
- La gestion d'adaptation :
  Dans les systèmes interactifs sensibles au contexte, la manipulation des informations de contexte se réalise soit à travers une gestion centralisée soit à travers une gestion distribuée ;
- Les techniques d'adaptation :
  Elles consistent à définir l'ensemble des mécanismes qui permettent de modifier le comportement d'application pour qu'il soit compatible avec le contexte d'utilisation en cours ;
- La description des relations entre les informations de contexte :
  Ces relations permettent dans une application sensible au contexte de déduire de nouvelles situations pertinentes et de les lier avec des politiques d'adaptation et des règles d'interprétation.

Les applications interactives s'exécutent dans un environnement pervasif dynamique. Par conséquent, ces applications doivent détecter les changements de l'environnement et adapter leurs comportements en fonction de ces changements. En abordant une comparaison au niveau de différents paramètres d'adaptation adoptées par les approches à base d'ontologique (cf table2.4), nous constatons que les approches à base d'ontologies sont les plus approches qui ont mené à une adaptation généralisée et réutilisable (par exemple au niveau du type d'adaptation concernée et considérée dans les plus part des approches comme une intégration de l'environnement et non pas comme une réaction).

Approches	Paramètres d'adaptation au contexte				
	Raison^{nt} Logique	Gestion	Type d'adaptation	Technique d'adaptation	DR
Paires/Triples	Non	Centralisée	Réaction	Att/Valeur	Non
Approches formelles	Non	Centralisée	Réaction	-	Non
Approches logiques	Oui	Centralisée	Intégration	Object Formel	Oui
HYDROGENE	Oui	Centralisée	Intégration	Meta-modèle	Oui
Approches profils	Non	Centralisée	Réaction	-	Non
KUP	Oui	Distribuée	Réaction	Raisonneur	Oui
ASK-IT	Oui	Distribuée	Réaction	Raisonneur	Oui
CoBrA	Oui	Distribuée	Intégration	Raisonneur	Oui
AMADEUS	Oui	Distribuée	Intégration	Raisonneur	Oui
CONON	Oui	Distribuée	Intégration	Raisonneur	Oui
SOCAM	Oui	Distribuée	Intégration	Raisonneur	Oui
ATRACO	Oui	Distribuée	Intégration	Raisonneur	Oui
PIVON	Oui	Distribuée	Intégration	Raisonneur	Oui
COIVA	Oui	Distribuée	Intégration	Raisonneur	Oui
OCAAR	Oui	Distribuée	Intégration	Raisonneur	Oui

TABLE 2.4 – Étude comparative au niveau des paramètres d'adaptation au contexte

Discussion

Les caractéristiques fondamentales d'un système interactif sensible au contexte peuvent se résumer en quelques points. En effet, ce système doit offrir un service complet, transparent et qui satisfasse les besoins d'utilisateurs. Ce système doit en plus surveiller le contexte et les paramètres d'accès au service qu'il offre aux utilisateurs. Le système doit aider l'usager dans ses tâches principales et lui permettre de se concentrer sur celles-ci tout en les rendant adaptables à ses exigences. Pour ce faire, un modèle descriptif riche, permettant de modéliser un grand nombre des éléments sensibles au contexte, doit être créé en premier lieu pour définir une méthodologie d'adaptation rigoureuse.

2.1.2/ TABLEAU RÉCAPITULATIF

Après l'étude comparative que nous avons menées sur l'ensemble des approches qui ont abordé la sensibilité au contexte, nous pouvons noter l'importance et l'efficacité des approches à base d'ontologies dans la conception des applications interactives sensibles au contexte d'utilisation. Comme nous pouvons le remarquer dans la table2.5, la plupart des approches à base d'ontologies possèdent des mécanismes efficaces de modélisation et d'interprétation sur les données de contexte, ce qui favorise l'abstraction du contexte et sa réutilisation.

En effet, elles utilisent souvent un serveur, pour la gestion du contexte, qui permet la séparation des processus de description du contexte et d'utilisation/interprétation. Cela permet une abstraction des détails des informations de contexte de bas niveau et favorise l'extensibilité et la réutilisation de ces connaissances dans d'autres applications. Ainsi, pour ces approches, qui sont basées sur une modélisation logique, les mécanismes de déduction utilisés sont les mieux adaptés pour réaliser l'abstraction des informations en concepts.

Approches	Abstraction	Distribution	Extensibilité	Réutilisation
Approches non ontologiques	+	+	-	+
Approches ontologiques	+++	+++	++	+++

TABLE 2.5 – Tableau récapitulatif

2.2/ LA SENSIBILITÉ AU CONTEXTE À BASE DE QUATRE MODÈLES ONTOLOGIQUES PIVON

Pour les conceptions de modèles de sensibilité au contexte, le contexte initial (*background*) est tout d'abord défini. Ce *background* peut être résumé sous forme d'un ensemble d'exigences initiales pour la modélisation et l'adaptation des applications interactives sensibles au contexte [Fuch05]. Les neuf exigences initiales sont les suivantes :

- La généralité :
  L'idée consiste à proposer dès le début un modèle de contexte générale lié à un domaine donné. L'utilisation des mécanismes d'adaptation de ce modèle à d'autres domaines connexes est possible ;
- La richesse en détails :
  Le modèle doit avoir un niveau de détails ;
- La distribution des composantes :
  La plupart des systèmes ubiquitaires proviennent du domaine des systèmes distribués. Le modèle de la sensibilité au contexte doivent gérer les différentes caractéristiques de distribution ;
- La validation partielle :
  Les informations contextuelles ainsi que les relations entre elles sont généralement complexes. Un modèle de contexte doit mettre en œuvre un ensemble de mécanismes de validation ;
- La qualité de l'information :
  Les modèles doivent représenter la qualité et la richesse des annotations ;
- Le traitement des ambiguïtés :

Généralement les informations de contexte sont ambigües. Le modèle doit identifier et quantifier cette caractéristique ;
- L'applicabilité aux environnements réels :
  Chaque modèle de contexte doit être appliqué dans des environnements réels tout en donnant la possibilité de l'interopérabilité entres les systèmes interactifs ;
- Le développement évolutif :
  Le modèle de contexte devrait favoriser l'adaptabilité et la conception évolutive. En outre, des services et des exigences supplémentaires doivent être intégrés dans le modèle au moment de l'exécution ;
- Le raisonnement à base d'inférence :
  Le modèle de sensibilité au contexte doit recueillir les informations à l'aide de techniques d'inférence.

LA MODÉLISATION DE LA SENSIBILITÉ AU CONTEXTE À L'AIDE DES ONTOLOGIES

L'utilisation des ontologies dans l'informatique ambiante met en avant plusieurs avantages et fonctionnalités supplémentaires. En général, l'adaptation des principes des ontologies pour les environnements pervasifs offre des avantages importants à travers une conceptualisation formelle des connaissances dans le domaine de la sensibilité au contexte. Suite à différents travaux de la littérature, publiés dans [Wang02] [Chen05] [Herv10], dans le domaine de la modélisation de la sensibilité au contexte utilisant les ontologies, nous pouvons identifier les avantages et les fonctionnalités importantes dans la phase de modélisation :

- Le développement de l'ontologie à travers son langage formel (OWL) constitue un moyen pour la représentation explicite des connaissances. En général, les ontologies peuvent être considérées comme un puissant mécanisme pour la structuration, l'organisation et la réutilisation de ces connaissances ;
- Les ontologies permettent l'acquisition des informations diversifiées de contexte même dans un contexte de sources hétérogènes ;
- Il est possible d'appliquer des mécanismes de raisonnement et d'inférence au moyen de représentation explicite de la sémantique, ce qui réduit les incohérences et les ambiguïtés des données.
  Les ontologies permettent l'interopérabilité entre les vocabulaires spécifiques à des domaines donnés. En outre, des systèmes hétérogènes peuvent définir la sémantique de l'ensemble des concepts, ce qui permet le partage de celui-ci ;
- Les environnements pervasifs comprennent des diversités technologiques surtout au niveau de l'offre de service. Ainsi les ontologies, qui permettent la réutilisation, peuvent également réduire les difficultés liées à cette diversité, ainsi que les difficultés liées aux efforts d'adaptation. En effet, les ontologies peuvent détecter les incohérences au niveau des connaissances et donc les résoudre par des informations historiques ou en combinant d'autres données de contexte valides ;
- Les ontologies permettent de simplifier la communication entre les humains et les systèmes informatiques.
  Ainsi, La proactivité dans le système sensible au contexte est améliorée grâce aux ontologies. En effet, il est possible de définir des comportements intelligents des entités en fonction de l'environnement de la situation contextuelle en cours ;
- La modélisation de sensibilité au contexte à base des ontologies peut réduire le coût de mise en œuvre et d'implémentation des mesures de sensibilisation ;

- À travers leur puissant langage, les ontologies permettent de fournir des mécanismes de représentation formelle de la connaissance qui améliorent les capacités du modèle de sensibilité non seulement au niveau du traitement de l'information contextuelle, mais aussi au niveau de la capacité d'adaptation, et même au niveau de la réutilisation de ces informations.

Compte tenu des travaux de R. Hervas, nous les prenons comme référence de notre travail compte tenu de l'importance du modèle à base d'ontologie qu'il a proposé. En effet, dans [Herv10], bien que le contexte soit un concept large, imprécis et non délimité, Il a proposé un modèle de contexte adaptable et générique tout en se référant lui-même aux travaux [Dey05].

Le contexte est décomposé en quatre catégories principales : le service, l'utilisateur, l'environnement et le dispositif technique. Il vise à une classification générale centrée sur l'utilisateur et qui est composée de ces quatre dimensions comme le montre la figure (cf table 2.6). En effet, l'emplacement, l'activité et le temps sont des propriétés élémentaires de son niveau supérieur d'abstraction de contexte.

LE *background* DU MODÈLE DE CONTEXTE PIVON À BASE D'ONTOLOGIE

LE MODÈLE D'UTILISATEUR

En effet, dans un environnement pervasif, l'utilisateur peut avoir un nombre important d'offres des services à travers divers dispositifs tout en utilisant une ou plusieurs applications et platesformes.

R. Hervas a proposé que l'utilisateur soit au centre des applications sensibles au contexte. De cela, il a définit le contexte d'utilisateur à travers la Théorie des 5 Ws (*What, Who, Where, When, Why*) [Broo03] pour la conception de ces applications sensibles au contexte.

Cette théorie permet de faciliter le traitement de chaque modèle utilisé dans les quatre catégories de la modélisation de la sensibilité : le modèle d'utilisateur (*User Model*), le modèle de dispositif technique (*Device Model*), le modèle d'environnement (*Environment Model*) et le modèle de service(*Service Model*).

Récemment, la plupart des travaux sur l'étude de la sensibilité au contexte ont proposé des modèles généraux d'utilisateur.

Contrairement à la modélisation de l'entité utilisateur qui s'est basée sur un modèle spécifique à une application donnée, dans les divers environnements ambiants, le modèle d'utilisateur doit supporter des conditions supplémentaires prenant en compte le dynamisme et la diversité de ces environnements :

- La résolution de l'ambiguïté :
  Généralement, les données acquises sur le contexte sont faits à partir de capteurs ou de dispositifs différents, ce qui peut engendrer des ambiguïtés, qu'un modèle utilisateur doit résoudre ;
- Les feedbacks :
  Les systèmes sensibles au contexte doivent être proactifs et doivent effectuer des mises à jour en fonction de la situation de l'utilisateur. Il est important de placer dans le modèle utilisateurs des mécanismes pour surveiller et pour modifier des

Catégorie	What	Who	Where	When	Why
Utilisateur (U)	Quel utilisateur et ce qu'il fait	son profil détaillé et ses relations sociales	Où l'utilisateur effectue ses tâches	Quand l'utilisateur effectue ses tâches	Pourquoi l'utilisateur effectue une tâche à cet endroit
Environ^{nt} (E)	Quel environ^{nt} et quels objets	Quel type d'utilisateurs dans cet environ^{nt}	Où sont placés les objets dans cet environ^{nt}	Quand les objets seront-ils disponibles	Pourquoi l'environ^{nt} a été organisé de cette manière
Services (S)	Quels sont les services offerts	Qui utilise ces services	Où sont utilisés les services	Quand les services seront-ils utilisés	Pourquoi ces services ont-ils été utilisés
Device (D)	Quel terminal est disponible	Qui utilise ce terminal	Où le terminal a-t-il été placé	Quand le terminal sera-t-il utilisé	Pourquoi ce terminal a été utilisé

TABLE 2.6 – Taxonomie de contexte selon la théorie des 5 Ws

informations personnelles des utilisateurs et avoir des feedbacks ;
- La généralité :
  Le modèle utilisateur devrait soutenir des fins d'utilisation diverses ce qul facilite la réutilisation et l'adaptation des systèmes interactives pervasifs.

Pour la création de notre modèle d'utilisateur, nous suivrons, dans la suite de nos travaux, la démarche utilisée dans le modèle d'utilisateur pour la sensibilité au contexte à base d'ontologie(cf figure 2.1). PIVon [Herv10]. Nous développerons cette démarche adaptée, dans la *"partie II. contribution"*.

LE MODÈLE D'ENVIRONNEMENT

La sensibilité au contexte est un besoin imposé avec l'avènement de l'environnement pervasif. Un environnement est un espace physique organisé d'une manière spécifique qui comprend tous les objets susceptibles d'être présentés en face d'un système interactive. Cet environnement favorise la création de nouvelles formes d'applications mobiles qui permettent aux utilisateurs de se déplacer en maintenant les interactions avec l'application. En effet, tous systèmes sensibles au contexte est un système qui est destiné à intégrer un environnement de travail. Dans cet environnement, le nombre des éléments de contexte est grand, ce qui pourrait générer plusieurs situations de contexte.

Les propriétés physiques de cet environnement sont principalement extraites de caractéristiques de type de terminal utilisé (laptop, PDA, ...) et de type de connexion (filaire ou non filaire), de caractéristiques d'utilisateur (préférences, exigences, ...). Plusieurs

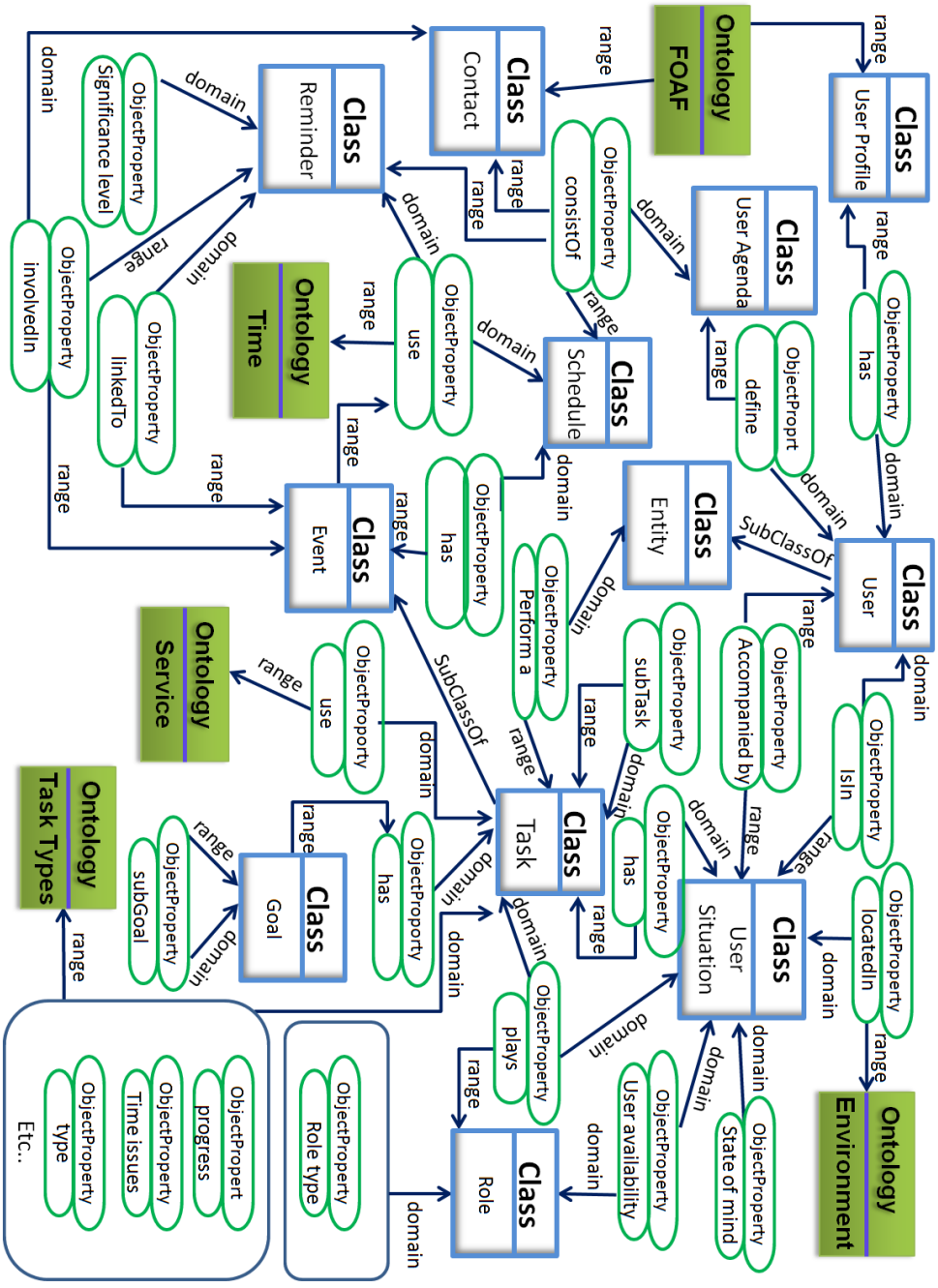

FIGURE 2.1 – Extrait du modèle d'utilisateur à base d'ontologie

modèles d'environnement à base d'ontologies ont été proposés. Nous pouvons citer le modèle d'environnement dans CANON [Wang04] ainsi que le modèle d'environnement dans CoBrA[Chen04] (*Context Broker Architecture*) qui vise à utiliser la représentation formelle des connaissances basée sur des ontologies pour décrire son modèle. Toutefois, l'organisation taxonomique des concepts dans ces approches a des limites au niveau de l'ambiguïté de l'ensemble des modèles ontologiques qui définissent le contexte.

Dans PIVon, R. Hervas a défini son modèle à base d'ontologie d'environnement tout en faisant face aux lacunes des autres approches(cf figure 2.2). Pour ces raisons, dans la

suite de notre travail, nous choisissons de prendre son modèle d'environnement à base d'ontologie.

L'environnement pervasif peut supporter les systèmes sensibles au contexte avec l'ensemble des objets qui l'entourent et qui doivent être bien définis dans le modèle de contexte. Pour la mise en œuvre, dans un environnement mobile distribué, les utilisateurs se déplacent fréquemment et peuvent utiliser des terminaux de différentes caractéristiques (en terme de caractéristiques de l'écran, de l'OS, des réseaux de connexions, ...). Ainsi dans la partie II, notre modèle d'environnement devra comporter l'ensemble des mécanismes permettant l'adaptation aux systèmes pervasifs.

LE MODÈLE DE TERMINAL

L'évolution technologique des dispositifs mobiles a donné naissance à de nouveaux besoins applicatifs pour assurer l'exécution des applications dans des environnements dynamiques. Ces applications appelées applications sensibles au contexte doivent détecter les variations de l'environnement et s'adapter en conséquence. Généralement, les environnements pervasifs comprennent une grande diversité de matériels pour garantir la bonne satisfaction d'un grand nombre de besoins d'utilisateurs au niveau des interaction. L'environnement doit prendre en compte l'intégration dynamique de nouveaux appareils, l'acquisition de leurs caractéristiques et la mise à niveau l'information globale de l'appareil. Afin d'atteindre ces objectifs, un modèle descriptif général concernant les appareils susceptibles d'être intégrés dans ces environnements est nécessaire.

La modélisation et la représentation des informations de ces terminaux à travers un langage formel et expressif permet de garantir une description structurée de l'ensemble des connaissances concernant ces terminaux ainsi que le partage de ces connaissances avec les autres modèles de contexte afin de gérer le dynamisme (l'évolutivité) des environnements pervasifs.

Dans la littérature, le nombre de propositions de modèles ontologiques pour la description des terminaux techniques reste médiocre. La modélisation dans ces travaux se concentre généralement sur un type spécifique de dispositif. En effet, les modèles de descriptions des terminaux comprennent des informations sur l'emplacement de l'appareil à un moment donnée et/ou une situation contextuelle donnée ainsi que les services qui doivent être offerts par ces appareils pour adapter le contexte courant [De06] [Garc08].

Les progrès dans le domaine des ontologiques améliorent les mécanismes de description des terminaux afin de permettre l'obtention de modèles riches, tout en associant la sémantique entre les données ainsi que la conceptualisation hiérarchique qui peuvent être situées dans les profils de ces appareils afin de permettre la réutilisation et le partage des informations sur le périphérique. Dans PIVon, R. Hervas a proposé un nouveau modèle de terminal à base d'ontologie (cf figure 2.3) qui permet de décrire l'ensemble des dispositifs susceptibles d'être utilisés dans un environnement pervasif. Dans son modèle à base d'ontologie, R. Hervas a proposé de décrire d'une manière rigoureuse toutes les caractéristiques de haut et de bas niveau de ces appareils. Il a défini, entre temps, une relation entre les différents modèles du contexte et le modèle d'appareils décrit.

Dans notre travail, nous définirons notre modèle ontologique de terminal tout en prenant comme référence le modèle de terminal de R. Hervas : compte tenu de son importance dans la modélisation des appareils techniques.

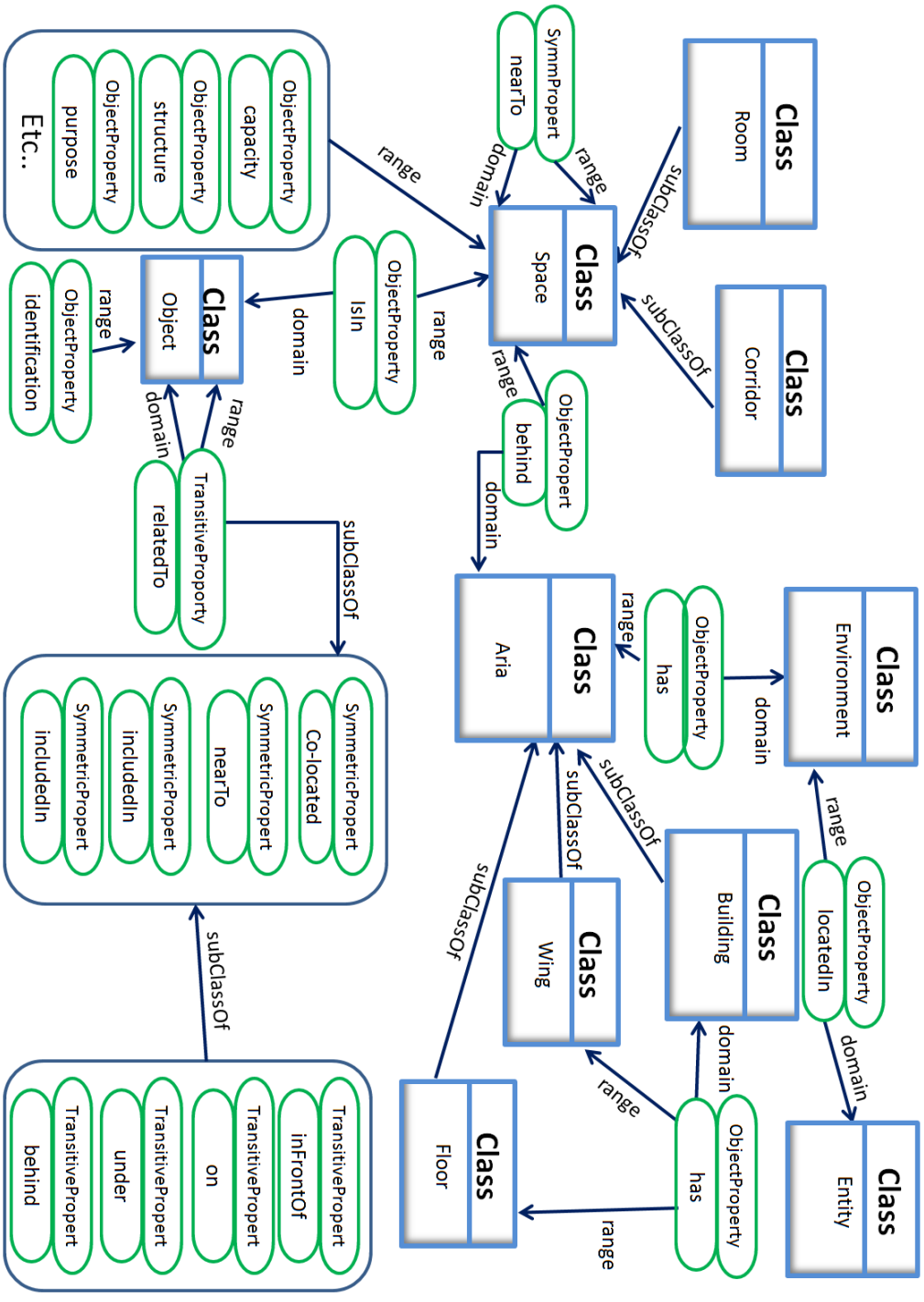

FIGURE 2.2 – Le modèle d'environnement à base d'ontologie (extrait)

LE MODÈLE DE SERVICE

Généralement, avec la diversité des systèmes pervasifs, les services offerts seront en grand nombre. Ces services doivent respecter les exigences imposées par toutes les situation contextuelles courantes.

Ainsi, dans la partie II, notre modèle de service devra être créé de manière universelle.

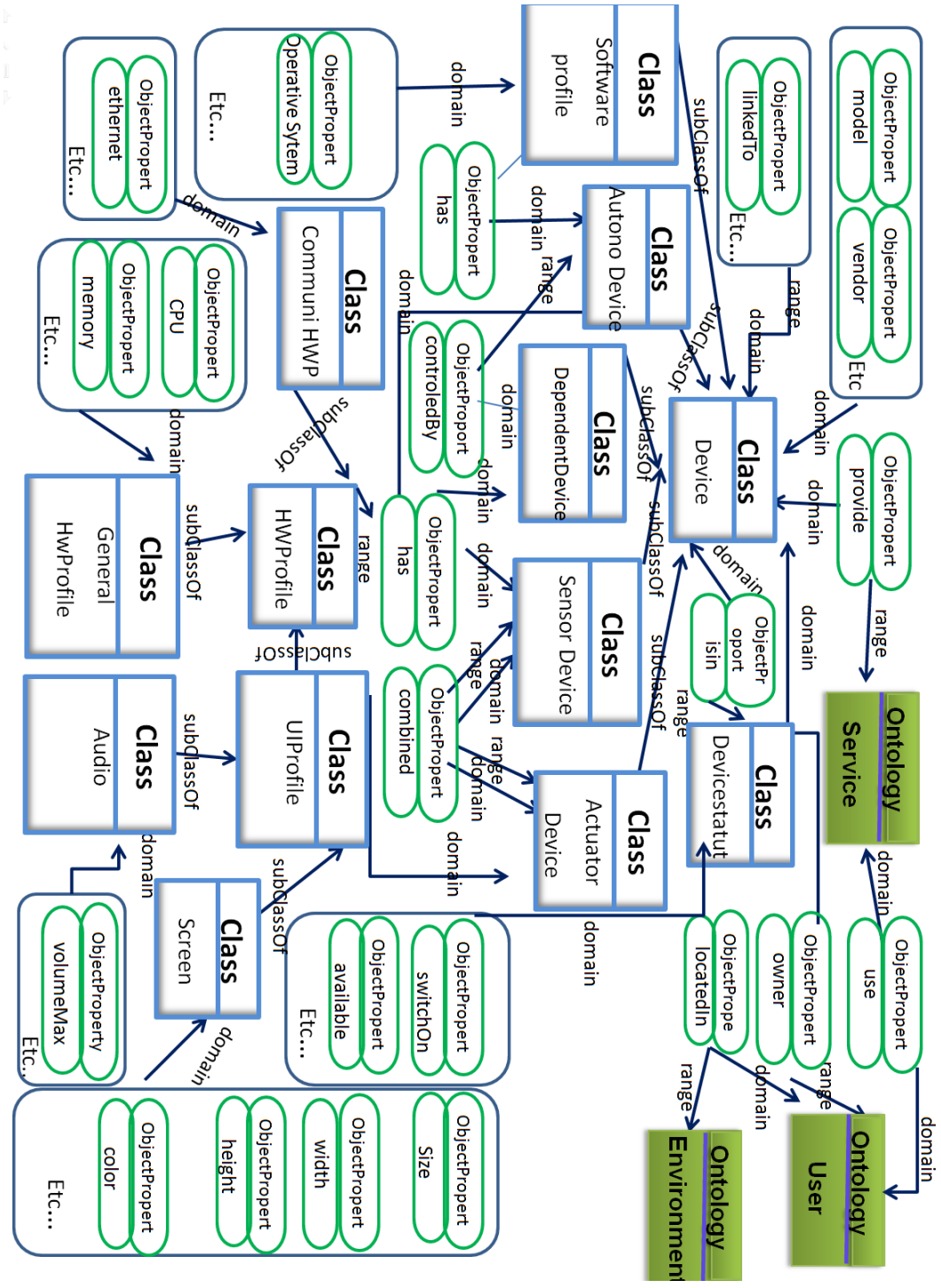

FIGURE 2.3 – Extrait du modèle de Terminal à base d'ontologie

Ce modèle à base d'ontologie devra permettre de définir les différentes spécifications des services qui seront offerts aux utilisateurs, y compris les aspects généraux impliqués dans des environnements pervasifs. L'infrastructure de modèle permettra l'organisation taxonomique et les mécanismes pour intégrer les modèles de services spécifiques. Dans la littérature, le modèle de service décrit dans PIVITA [Herv08], permet de proposer un cadre général pour les services de visualisation de l'information qui offrent un contenu

adapté aux utilisateurs en fonction de leurs profils et des situations contextuelles. Par ailleurs, la conception de ce modèle donne des formalités sur la modélisation de la sensibilité au contexte à travers l'ensemble des services susceptibles d'être présentés dans un contexte donné. Les utilisateurs ont besoin de soutien pour obtenir de l'information nécessaire partout et à tout moment. Par conséquent, R. Hervas a considéré que la progression de l'informatique ambiante oblige à se focaliser également sur les services de visualisation de l'information qui peuvent être proposés à ces utilisateurs.

Une approche taxonomique a été proposée dans [Herv09], elle permet de franchir une étape supplémentaire vers la compréhension de l'espace de conception de visualisation de l'information dans des environnements pervasifs en extrayant les caractéristiques de base de ces services. En effet, cette approche consiste à utiliser une ontologie qui permette de décrire et de modéliser les services de visualisations offerts aux utilisateurs des systèmes pervasifs et qui permette l'adaptation au contexte d'utilisation. Ce modèle a été conçu à travers un ensemble d'éléments d'informations de visualisation en forme de services. En effet, selon les situations des utilisateurs (décrits dans le modèle), les meilleures informations seront sélectionnées sous formes de services à offrir à l'utilisateur dans la phase de visualisation de son interface. Chaque service possède plusieurs caractéristiques associées qui ont été décrites dans l'ontologie de service de visualisation afin de générer de meilleures sorties tout en permettant l'adaptation de l'interface utilisateur en fonction de la situation contextuelle en cours.

Dans PIVon [Herv10] et COIVA (*Context-aware and Ontology-powered Information Visualization Architecture*) [Herv11], cette taxonomie a été enrichie à travers une identification de plusieurs concepts tout en les classant en fonction des critères prédéfinis pour construire une taxonomie plus rigoureuse et générique qui permette ainsi de traiter un grand nombre des domaines (cf figure2.4).

En effet, dans COIVA, une architecture a été proposée afin de permettre de fournir l'infrastructure pour lancer des services de contexte alimentés à travers des dispositifs environnementaux personnels ou publics. Ils partagent une base de connaissances des éléments de contexte et la sémantique nécessaires pour identifier les besoins et les habitudes des utilisateurs, et offrir des services d'adaptation partout et à tout moment. Dans son noyau fonctionnel, le modèle de service a été conçu en identifiant les éléments clés des environnements pervasifs et surtout dans les environnements particuliers. En effet, le modèle ontologique de service dans COIVA a accordé une attention particulière aux environnements pervasifs pour éviter l'inclusion d'éléments qui sont incompatibles avec certains domaines d'application : comme les relations, les contraintes ou les axiomes qui ne sont pas valides dans certains cas d'utilisation.

D'autres modèles ont été conçus pour décrire les services dans les systèmes pervasifs. En effet, Dans OCAAR (*Ontological Context-awareness for Adaptive Augmented Reality*) [Herv13] l'architecture proposée utilise le principe des ontologies, dans son modèle de service, pour adapter les applications interactives à la sensibilité au contexte. Elle permet donc la personnalisation à l'utilisateur. En effet, le modèle de service est conçu pour décrire et supporter l'ensemble des services susceptibles d'être utilisés dans le domaine de la réalité augmentée (cf figure2.5), tout en offrant des fonctionnalités pour rendre des décisions sur quoi et comment ces informations devraient être offertes.

Pour la modélisation de notre propre modèle de services, nous prendrons les précédents modèles descriptifs de service, selon l'axe de recherche ainsi que la problématique de chaque approche, comme une référence de notre travail compte tenu de l'importance de

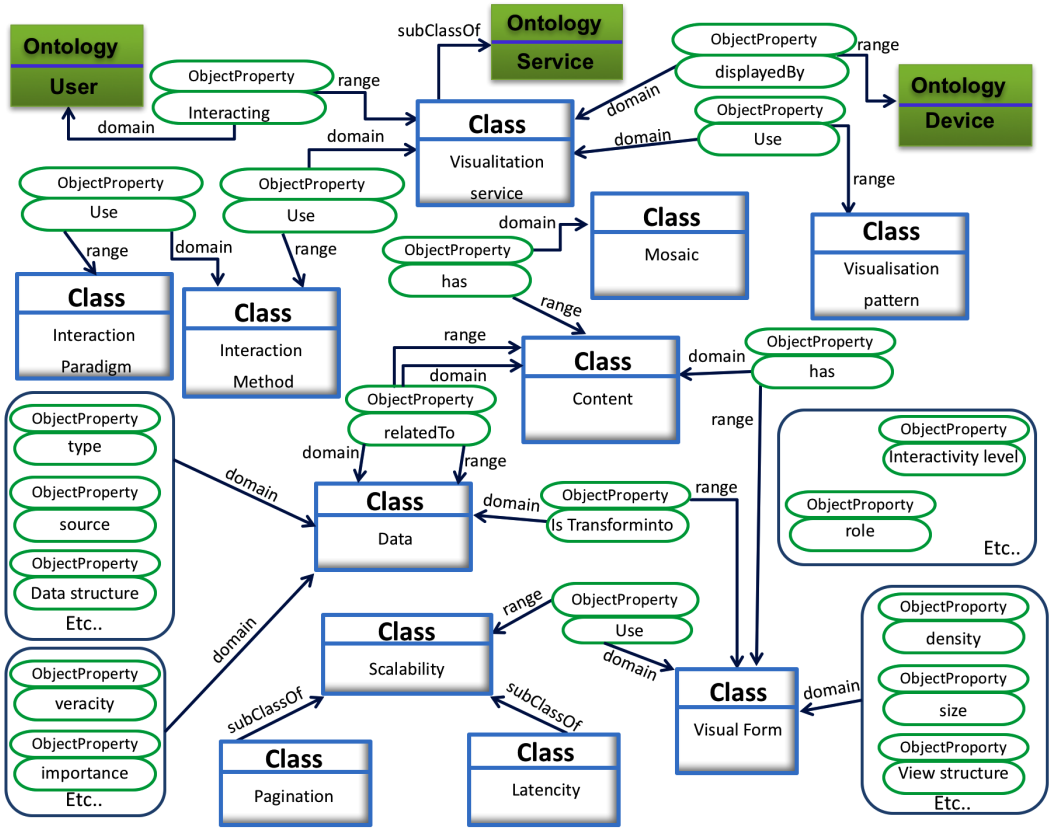

FIGURE 2.4 – Extrait du modèle de service à base d'ontologie dans PIVon

la diversité concernant la description des services offerts afin de proposer notre modèle propre à nos besoins. Nous le développerons avec plus de détails dans notre partie II : contribution.

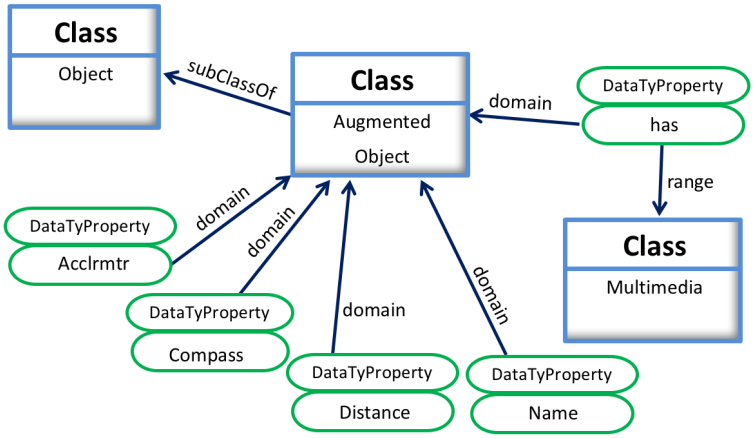

FIGURE 2.5 – Le modèle de service à base d'ontologie dans OCAAR

2.3/ LA CLASSIFICATION DE DIFFÉRENTES APPROCHES EXISTANTES DE LA SENSIBILITÉ AU CONTEXTE

La sensibilité au contexte est le fait de réagir en prenant en compte l'information de contexte. Toutefois, les applications sensibles au contexte sont des applications ayant des mécanismes pour changer dynamiquement ou adapter leurs comportements en se basant sur le contexte de l'application ou de l'utilisateur. Ils utilisent le contexte pour fournir une information pertinente à l'utilisateur à travers un changement automatique de ses formes de services ou le déclenchement d'un service comme réponse au changement de la valeur d'une information ou d'un ensemble d'informations qui caractérisent le service. Bien que la sensibilité au contexte soit l'aptitude à modéliser, interpréter et répondre aux aspects de l'environnement local de l'utilisateur et de terminal (adaptation), nous considérons que cette sensibilité est une sorte de transformation qui commence par la phase d'interprétation d'informations de contexte arrivant à la phase d'adaptation à ce contexte.

Nous constatons dans les approches à base d'ontologie qu'il y a une séparation entre la phase de gestion de contexte (qui consiste à la représentation formelle des informations de contexte à travers des ontologies), et la phase d'adaptation au contexte (qui est constituée de l'ensemble des mécanismes de réactions déclenchées suite aux changements de ce contexte et des règles d'adaptation).

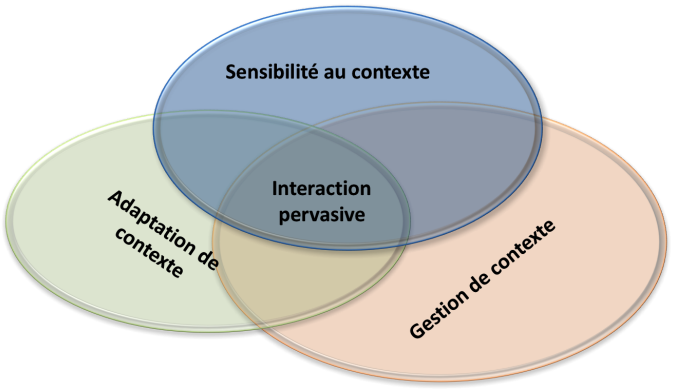

FIGURE 2.6 – Modélisation de la sensibilité au contexte à base d'ontologie

Cette séparation nous permet de proposer une classification de ces différentes approches au niveau du traitement (gestion de contexte) et d'interprétation (adaptation) de la sensibilité au contexte (cf figure 2.6). Notre classification porte sur les méthodes et les techniques liées à la modélisation ainsi qu'à l'adaptation au contexte des systèmes pervasifs. Cette étude porte dans un premier temps sur la classification des approches ontologiques au niveau de la modélisation de la sensibilité au contexte (gestion de contexte, cf figure 2.7) pour passer dans un deuxième temps à une classification selon leur forme d'adaptation (adaptation de contexte, cf figure 2.8).

La représentation de contexte est la structuration de ce contexte selon un modèle. Nous décrirons ici les représentations basées sur des ontologies. Une représentation est dite à base d'ontologie puisqu'elle est constituée d'un ensemble structuré de concepts. Les concepts sont organisés dans un graphe dont les relations peuvent être des relations sémantiques ou des relations de composition et d'héritage. Face à un contexte d'utili-

sation donné, une représentation permet de fournir un vocabulaire représentatif pour ce domaine, un ensemble de définitions et d'axiomes qui contraignent le sens des termes de ce vocabulaire de manière suffisante pour permettre une interprétation consistante des données représentées au moyen de ce vocabulaire.

À travers une étude de l'ensemble des approches de représentation de contexte à base d'ontologie que nous avons décrits, nous proposons une classification (cf figure 2.7) de ces différentes approches en fonctions :

- de la modélisation des objets définis dans le domaine,
- de la perception des informations de contexte,
- du niveau de formalisme du modèle,
- et de la taxonomie des concepts utilisées .

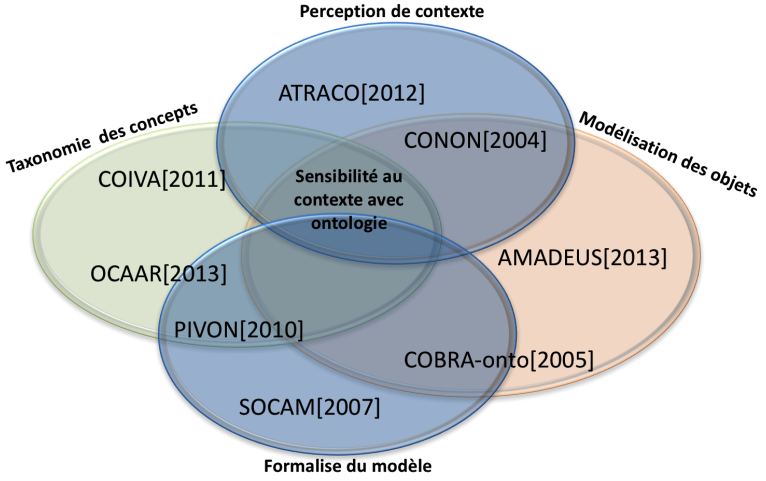

FIGURE 2.7 – Classification des ontologies existantes dans le domaine de la sensibilité au contexte

Pour son fonctionnement, l'application interactive doit être dotée d'un système d'adaptation basé sur des mécanismes d'interprétation et de raisonnement qui peuvent être soit une interprétation logique avec de règles préétablies (la logique du premier ordre, algèbre booléenne, ...), soit une interprétation à travers de l'apprentissage de comportement (comportement des agents par exemple). Or, un système sensible au contexte est un système qui réagit selon un comportement convenable en considérant les informations modélisées de l'environnement contextuel de ce système. L'adaptation au contexte pour ces systèmes est une réaction à l'aspect dynamique de l'environnement. Cette réaction permet d'associer directement une perception d'une information de contexte à une action d'adaptation suite à un changement de l'environnement. Ainsi nous proposons une classification des différentes approches selon la forme d'adaptation adoptée (cf figure 2.8).

Dans CoBrA par exemple, la forme d'adaptation adoptée est la forme structurelle. En effet, l'interprétation des informations de ce contexte est orientée agents. L'objectif de ces agents intelligents est d'apprendre un comportement pertinent pour un système ambiant en se basant sur l'observation des actions récurrentes des utilisateurs, puis d'établir dans quels contextes ces actions sont réalisées afin de suppléer l'utilisateur. L'élément central de cette architecture est l'agent intelligent, appelé *courtier de contexte*, qui maintient un modèle partagé de contexte pour une communauté d'agents, de services et de capteurs. Dans un premier temps, la modélisation de contexte s'est faite à base des ontologies en

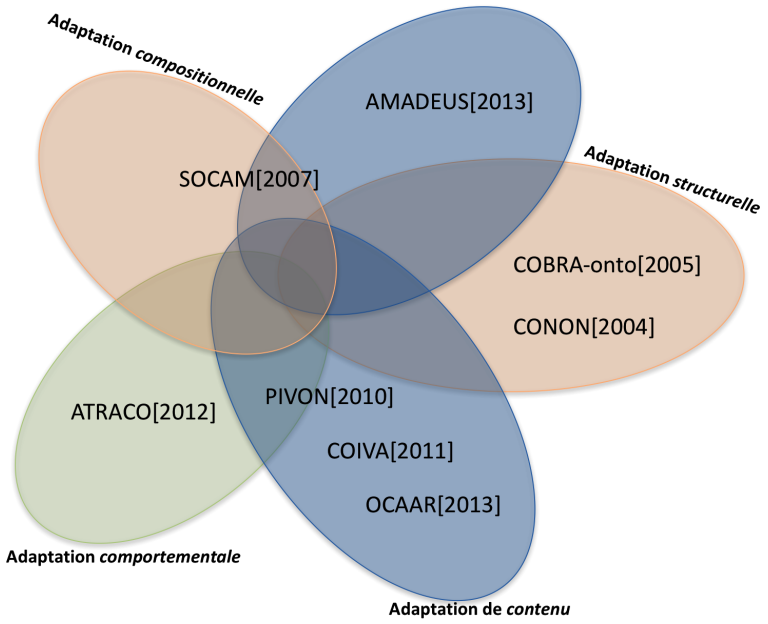

FIGURE 2.8 – Classification des approches de la sensibilité au contexte selon leurs formes d'adaptation

utilisant le langage OWL. Elle définit une ontologie qui décrit les relations communes et les attributs associés aux personnes, endroits et activités. CoBrA permet à travers son ontologie de modéliser les données contextuelles en un ensemble de modules partant de la phase d'acquisition jusqu'à la phase de la création de la base de connaissances tout en respectant la séparation entre ces différents modules. Pour la phase d'interprétation sur les informations de contexte, le *courtier de contexte* permet d'interpréter les informations de contexte, et il représente l'intermédiaire entre les terminaux et le noyau de l'application.

Discussion

Pour décrire la sensibilité au contexte d'exécution d'une application interactive dans un environnent pervasif, il faut déterminer les contextes auxquels cette application est sensible et les décrire dans un modèle. Par conséquent, la modélisation du contexte à base des ontologies est la plus adaptée dans le processus de création d'applications sensibles au contexte. Cette modélisation permet à l'application et aux différents objets de l'environnement de faciliter l'interaction avec le contexte en fournissant une description abstraite. À travers la classification faite, nous retenons que ces différentes approches peuvent avoir des points communs au niveau de la modélisation et des processus d'adaptation des applications sensibles au contexte.

Synthèse du chapitre

Un important défi dans le domaine de l'informatique ambiante porte sur l'optimisation de l'utilisation des mécanismes d'une part, au niveau de la gestion de l'information contextuelle, et d'autre part, au niveau des techniques d'adaptation des applications interactives à la diversité des environnements pervasifs.
Dans ce chapitre, nous avons commencé par une étude comparative qui porte sur l'aspect de la description et de modélisation de contexte dans les différentes approches qui abordent le domaine de la sensibilité au contexte. Nous constatons à ce stade que les approches à base d'ontologies sont celles les plus adaptées pour décrire et modéliser les informations liées à la sensibilité au contexte des applications pervasives. En effet, Les travaux existants sur la modélisation de la sensibilité au contexte décrits dans le chapitre 1, qui n'utilisent pas les ontologies, ne permettent pas de donner un modèle général, extensif et surtout réutilisable pour la gestion et l'adaptation des applications à toute sensibilité contextuelle.
De ce fait, nous nous sommes focalisés dans la deuxième et la troisièmes section sur les approches à base d'ontologies. Dans la deuxième section, Nous avons décrit les modèles ontologiques de la sensibilité au contexte (en particulier les modèles de R. Hervas) pour les prendre comme une référence dans notre travail de recherche (contribution partie suivante).
Nous avons achevé ce chapitre en proposant une classification entre ces différentes approches en fonction de l'ensemble des axes qui on été traités au niveau de la gestion et de l'adaptation de contexte.

Dans ce chapitre, nous avons effectué un tour d'horizon des différents travaux de recherche évoqués dans le domaine de la sensibilité au contexte ainsi que des différentes propositions des plateformes d'adaptation dotées de modèles de gestion et de contexte. Nous constatons que l'utilisation des modèles de contexte à base d'ontologie permettent l'adaptation des utilisateurs. Mais la majorité des solutions existantes proposent des approches où les paramètres de contexte sont généralement identifiés dans la phase de conception de l'application pervasive : ce qui peut poser des limites face aux changements dynamiques de ce contexte, qui nécessitent des réactions systèmes instantanées en temps réel.

Dans la deuxième partie de ce document de thèse, nous présenterons notre contribution. Nous exposerons notre modèle d'ontologie de contexte tout en intégrant notre ontologie de traçabilité pour bien spécifier les situations contextuelles courantes de l'utilisateur. Nous suivrons le processus de développement des ontologies dans notre domaine. Puis nous définirons notre nouvelle plateforme d'adaptation au contexte, nommée COALA, (*COntext Adaptation pLAtform*).

II

CONTRIBUTION

De nos jours, l'interaction avec les applications informatiques dans un environnement pervasif s'inscrit de plus en plus dans les environnements impliquant les technologies mobiles. L'adoption de ces technologies assure plus de flexibilité et permet de créer de nouvelles formes d'utilisation. La prise en compte du contexte d'utilisation dans les applications interactives pervasives est un domaine de recherche connu sous le nom de « *sensibilité au contexte* » ou encore « *context-awareness* » : une application sensible au contexte est une application qui répond aux exigences imposées aux informations de contexte. Bien plus encore, la sensibilité au contexte est devenue un élément central pour la conception et la mise en place de services adaptatifs.

Dans ce domaine, les efforts se focalisent d'une part sur la modélisation et la gestion de contexte, et d'autre part sur l'adaptation et l'actualisation de l'application au contexte. Ces deux aspects deviennent centraux surtout dans le cas où l'on veut ajouter la sensibilité au contexte aux systèmes pervasifs. Ces systèmes intègrent des terminaux mobiles de différentes capacités matérielles et logicielles, et sont utilisés par une variété d'utilisateurs à tous moments et à n'importe quel endroit.

Une application sensible au contexte doit être capable de gérer les informations de contexte en vue d'apporter des services adaptés. La prise en compte de ces deux volets fait l'objet de notre contribution. En effet, pour réaliser la gestion contextuelle, nous adoptons le concept d'ontologie au travers de descriptions universelles élémentaires de domaine de la sensibilité au contexte. La prédéfinition des modèles de gestion de contexte à base d'ontologies sera très importante au niveau du dernier composant de la chaine de sensibilité des applications au contexte pour réaliser l'adaptation.

Devant une grande variété des informations de contexte, la modélisation et la gestion de contexte dans un environnement informatique omniprésent est en effet une tâche difficile. L'utilisation des ontologies nous permet dans cette tâche une description structurée de la connaissance tout en fournissant la sémantique qui caractérise les relations existantes entre les différents concepts du domaine. Cette description permet également de fournir une structure commune des informations de contexte. Ainsi, elle nous permet le partage de ces connaissances, grâce au langage flexible et extensible dont elle est dotée.

Au niveau de l'adaptation aux changements de l'environnement contextuel, à son dynamisme ainsi qu'à l'hétérogénéité des sources de perception, un compte rendu sur les situations contextuelle courantes sera récupéré sous forme d'une *carte visite contextuelle* liée à nos ontologies.

Dans cette partie, nous commençons dans le chapitre 3 par décrire le processus de développement des ontologies dans le domaine de la sensibilité au contexte. Nous présentons par la suite notre modèle général de contexte à base de ces ontologies tout en mettant en avant toutes les exigences imposées par un environnement dynamique pervasif. Puis, nous exposons notre proposition pour réaliser à l'aide d'une ontologie de traçabilité l'adaptation au contexte par l'intégration de la *carte visite contextuelle* (*CVCO*), afin de rendre les applications interactives adaptables au contexte courant.

Dans le quatrième chapitre, nous définissons notre nouvelle plateforme d'adaptatation au contexte, nommée *COALA*, (*COntext Adaptation pLAtform*) qui s'articule sur notre modèle de contexte à base d'ontologie de traçabilité. Puis nous présentons une première implémentation sur un cas d'utilisation : "le suivi de l'évolution de tumeur de cerveau". Les premiers résultats sont très encourageants.

3

NOTRE ONTOLOGIE DE TRAÇABILITÉ POUR LA GESTION ET L'ADAPTATION DE CONTEXTE

INTRODUCTION

La modélisation de divers contextes dans un environnement informatique omniprésente est une tâche difficile, qui fait l'objet de nombreuses recherches. Le domaine de l'informatique omniprésente (ou ubiquitaire) peut être divisé en une collection de sous-domaines géographique tels que le domaine de la maison, le domaine de bureau, le domaine du véhicule,... Il serait facile de préciser le contexte à l'aide d'une gamme de contextes prédéfinis (ensemble d'éléments de contexte terminal, localisation, ...). Pour ce faire, contrairement à la simple hiérarchisation de données utilisées par les modèles balisés ou objet/valeur que nous avons déjà traitée précédemment, les ontologies permettent de spécifier les paramètres et les liens qui existent entre les différentes entités du contexte.

Ainsi, l'utilisation d'ontologies pour la modélisation du contexte est particulièrement appropriée pour représenter la connaissance. Le langage OWL est également utilisé pour modéliser les entités et les relations définissant un contexte. Dans une ontologie du contexte, les liens peuvent être annotés et possèdent des propriétés. Chaque lien existant entre deux entités peut être défini par une sémantique (dépendance, équivalence, sous-propriété, ...), c'est grâce à la définition de la sémantique entre les termes qu'il permet de faciliter les opérations de raisonnement pour extraire des connaissances.

Au cours de ce chapitre, nous commençons par décrire le processus de développement des ontologies dans le domaine de la sensibilité au contexte. Puis, nous présentons par la suite notre propre modèle de contexte à base d'ontologies tout en mettant en avant toutes les exigences imposées par un environnement dynamique pervasif. Nous exposons finalement notre proposition d'intégration d'une ontologie de traçabilité des services de contexte pour rendre les applications interactives adaptables au contexte courant.

3.1/ LES ONTOLOGIES POUR LA MODÉLISATION DE CONTEXTE (UTILISÉES POUR LA GESTION CONTEXTUELLE)

La présente section se focalise sur la description des méthodes adoptées pour la modélisation des éléments de contexte et la gestion de contexte à base d'ontologies. Cette description porte sur les définitions de différents élément sous-jacents du processus de modélisation et de gestion de contexte ainsi que la définition du langage de représentation de connaissances dans le domaine de la sensibilité au contexte.

3.1.1/ LA GESTION DES CONNAISSANCES POUR LA SENSIBILITÉ AU CONTEXTE

L'utilisation de connaissances en informatique a pour but, dans ce cas, de manipuler les informations de domaine en permettant une interaction entre le système et les utilisateurs. Pour cela, le système doit avoir accès non seulement aux termes utilisés par l'être humain mais également à la sémantique associée, afin qu'une communication efficace soit possible. Le processus d'ingénierie des connaissances définit des étapes pour organiser les connaissances au sein de la représentation formelle. Un modèle conceptuel de la connaissance est ensuite traduit en une représentation qui pourra être manipulée par les systèmes informatiques. Il est également nécessaire qu'une sémantique soit associée à des méta-données. Le système doit être capable d'interpréter le rôle de la méta-donnée dans la représentation des données. De plus, il doit être capable d'interpréter les liens entre différentes méta-données associées aux données.

Une représentation de connaissance est une structure, composée de symboles, construite à partir d'un ensemble de règles de formation [Kays97]. L'ensemble de ces règles est défini par le langage de représentation choisi. La représentation de la connaissance s'appuie alors sur des représentations au niveau conceptuel pouvant modéliser la «*structure cognitive*» d'un domaine. Pour le niveau conceptuel, on entend ici une formalisation sur la description des connaissances avant de se préoccuper de la manière dont un système inférentiel pourra les traiter.

Les ontologies qui seront définies dans la prochaine section sont des exemples de telles représentations.

Pour bien maitriser la représentation de connaissances dans un domaine précis, les principales préoccupations de ce domaine sont l'acquisition, la modélisation, le stockage/consultation de connaissances, le raisonnement automatique sur les connaissances stockées et la modification des connaissances stockées.

Les langages à base de frame[Mins00], les logiques de description [Brac85] et les graphes conceptuels [Sowa00] sont des langages permettant ces représentations. Ces langages ont en commun de donner priorité au pouvoir d'expression par rapport à la capacité de raisonnement logique. Ils permettent de représenter pour un domaine de connaissance donné, les concepts, les relations entre les concepts, ainsi que la sémantique de ces relations. Ces dernières notions sont expliquées dans la section qui suit.

3.1.2/ LES ONTOLOGIES POUR LA MODÉLISATION DE LA SENSIBILITÉ CONTEXTUELLE

Une ontologie permet de spécifier la connaissance nécessaire au système pour interpréter le rôle sémantique des méta-données. Thomas Gruber, dans [Grub93], introduit la notion d'ontologie comme *une spécification explicite d'une conceptualisation*. Cette définition a été légèrement modifiée par Willem Borst dans [Bors97]. Une combinaison des deux définitions peut être résumée ainsi : *une spécification explicite et formelle d'une conceptualisation partagée*. Cette définition s'explique ainsi [Stud98] :

- **explicite** signifie que le type des concepts et les contraintes sur leurs utilisations sont explicitement définis,
- **formelle** se réfère au fait que la spécification doit être lisible par une machine,
- **partagée** se rapporte à la notion selon laquelle une ontologie capture la connaissance consensuelle, qui n'est pas propre à un individu mais validée par un groupe,
- **conceptualisation** se réfère à un modèle abstrait d'un certain phénomène du monde reposant sur l'identification des concepts pertinents de ce phénomène.

Comme les ontologies sont utilisées dans de nombreux domaines pour modéliser, il nous est apparu que ces dernières étaient aussi un des meilleurs moyens de modéliser la sensibilité au contexte. En effet, les ontologies permettraient aux différentes dimensions de contexte d'interpréter le sens des mots contenus dans la variété des d'informations. Accéder aux sens des mots, aux concepts sous-jacents, aux relations sémantiques entre les concepts permettrait aux systèmes d'établir une meilleure interprétation du contenu qu'ils ont à gérer. Ces ontologies pourraient de plus apporter de la connaissance utile à d'autres niveaux de compréhension.

Les ontologies peuvent également être utilisées pour les systèmes sensibles au contexte afin de mieux interagir et comprendre les requêtes des l'utilisateurs auxquelles ils doivent répondre. Ceci devient possible en donnant à ces systèmes un accès à une représentation de la connaissance et en fournissant les mécanismes d'accès. Finalement, les ontologies peuvent permettre une meilleure spécification de domaine selon ces différentes structures de connaissances. Dans [Van97], la classification distingue trois types d'ontologies :

- Les ontologies terminologiques ou linguistiques : elles permettent de spécifier les termes utilisés pour représenter la connaissance d'un domaine. On peut citer l'exemple d'ontologie du réseau sémantique *UMLS (Unified Medical Language System)*, [Lind08].
- Les ontologies de l'information : elles permettent de spécifier la structure des enregistrements d'une base de données. Elles proposent un cadre de représentation de la connaissance stockée mais ne spécifient pas de détails sur la sémantique des champs.
- Les ontologies pour la modélisation de la connaissance : elles permettent de spécifier la conceptualisation de la connaissance. Ces ontologies ont une structure beaucoup plus riche que celle des deux autres types. Elles sont généralement conçues en fonction de l'utilisation prévue de la connaissance qu'elles contiennent.

Dans le cas de notre domaine lié à la sensibilité de contexte, la structure de connaissance utile se situe dans les ontologies pour la modélisation de la connaissance. Elles permettent d'expliquer la conceptualisation sous-jacente aux formalismes de représentation

[Davi97]. Ainsi, elles proposent un cadre de représentation abstrait (ou de haut niveau) parce qu'elles permettent de définir des concepts abstraits qui peuvent être réutilisés pour définir des concepts dans un domaine spécifique.

3.1.3/ LA MODÉLISATION DE LA SENSIBILITÉ CONTEXTUELLE À BASE D'ONTOLOGIES

Nous présentons dans ce qui suit la définition de différents éléments sous-jacents du processus de modélisation et de gestion de contexte ainsi que la définition du langages de représentation de connaissances dans le domaine de la sensibilité au contexte.

A. LES ÉLÉMENT SOUS-JACENTS DU PROCESSUS DE MODÉLISATION ET DE GESTION DE CONTEXTE

Le processus de modélisation de contexte à base d'ontologies peut être défini à travers l'ensemble des ressources conceptuelles de ces ontologies. En effet, une ressource conceptuelle est la structure qui permet de décrire le niveau conceptuel et le lexique correspondant au niveau lexical. Le niveau lexical couvre tous les termes ou labels définis pour désigner les concepts. Le niveau conceptuel représente les concepts ainsi que la sémantique qui leur est associée à partir des relations conceptuelles qui les lient [Maed04]. Les ontologies permettent, en particulier, de formaliser de manière cohérente et consensuelle les connaissances d'un domaine donné.

Dans la suite de cette sous section, et afin d'exprimer la notion de l'engagement sémantique dans la formalisation des ontologies liées à la sensibilité au contexte, nous présentons les principaux composants utilisés pour développer des ontologies. Concrètement, chaque système ontologique est formellement défini comme suit (définitions usuelles issues de [Grub93, Guar95, Fan09]) :

Les concepts

Dans une ontologie les concepts sont rassemblés pour fournir les briques élémentaires et exprimer les connaissances dont on dispose dans ce domaine. L'ensemble de concepts ou de classes (sous le logiciele *Protégé* par exemple) est hiérarchisé par une relation de subsomption (entre classe et sous-classes). En effet, la tâche de hiérarchisation nécessite une expertise de domaine étudié à partir d'un corpus spécifique à un domaine donné, d'une taxonomie ou d'un thésaurus. Après le repérage et la classification de l'ensemble de concepts, il est nécessaire de définir les propriétés qui leur sont attachées. Les propriétés peuvent être définies comme l'ensemble des attributs, fonctionnalités, caractéristiques ou paramètres que les objets peuvent posséder et partager dans une ontologie. Une propriété pourra être attachée aux concepts et sa valeur variera suivant le concept auquel on fait référence.

Les relations binaires

Les relations binaires dans une ontologie permettent d'exprimer la sémantique du domaine. Généralement, ces relations englobent les liens et les interactions entre les

concepts de l'ontologie. Parmi ces liens nous pouvons citer : la relation de subsomption *est-sous-classe-de*, la relation *est-partie-de*, la relation associative *est-lié-à*, ... Les relations de subsomption, par exemple, permettent de définir n liens de généralisation : héritages de propriétés. Dans [Guar00], La notion de subsomption est une relation binaire particulière qui implique l'engagement sémantique suivant : un concept c_1 subsume un concept c_2 si toutes relations sémantiques de c_1 est aussi relation sémantique de c_2 : en d'autres termes si le concept c_2 est plus spécifique que le concept c_1.

Les instances se rapportant au concept c_2 seront des instances de c_1, en revanche une partie seulement des instances de c_1 seront des instances de c_2.

Les axiomes

Les axiomes décrivent la structure du modèle, les assertions de l'ontologie qui serviront par la suite au le moteur d'inférences. Ces assertions (ou contraintes) ont pour but de définir ou de préciser la signification des objets de l'ontologie. Les contraintes sur les attributs et les arguments de relations (restriction des domaines), les propriétés de relations comme par exemple la transitivité, la symétrie, inverse de, ...).

- **La symétrie** : cette propriété signifie que si la paire(X,Y) est une instance de P, alors automatiquement la paire(Y,X) est également une instance de P.
- **La transitivité** : lors de définition d'une propriété P comme une propriété transitive, cela signifie que si une paire(X,Y) est bien une instance de P, au même temps, une paire Y,Z est aussi une instance de P, alors nous pouvons déduire que la paire (X,Z) est également une instance de P. Formellement, on peut écrire : (X subsume Y) et (Y subsume Z) $\Rightarrow$ (X subsume Z).
- **inverse de** : cette propriété a pour but de définir une relation symétrique entre des propriétés. Un axiome de la forme P1 *owl :inverseOf* P2 fait valoir que pour chaque couple (X,Y) de la propriété P1, il existe un couple (Y,X) dans l'extension de propriété de P2, et vice versa. La propriété *owl :inverseOf* est donc symétrique.

Les axiomes sont des expressions qui sont toujours vraies. Leur inclusion dans une ontologie peut avoir plusieurs objectifs : définir la signification des composants, définir des restrictions sur la valeur des attributs, définir les arguments d'une relation, vérifier la validité des informations spécifiées ou en déduire de nouvelles.

Les instances

les instances constituent l'univers de base (univers du discours). Elles permettent la définition existentielle de l'ontologie.

Les faits

Ces faits décrivent des situations concrétes particulières entre les instances. Cette structure permettra de donner d'une part un vocabulaire partagé pour décrire un domaine, un typage des données et d'autre part une signature des relations ainsi que la capacité de raisonner (inférence).

B. Les Langages de représentation d'ontologie

les langages orientés Web Sémantique permettent la spécification des ontologies. Les propositions des standards ont été pour la plupart recommandées par le W3C dans le cadre de la croissance et de l'exploitation du Web. La suite de cette section expose différents langages utiles pour ces travaux :

XML

En 1998, le langage de balisage extensible XML (*Extensible Markup Language*) devient une recommandation du W3C. Ce langage permet de générer des balises pour la structuration de données et de documents et la représentation et l'échange de documents semi-structurés [Brad02]. Les documents XML doivent être lisibles par l'homme et raisonnablement clairs. *XML-schema*, [Van02], a été créé pour vérifier la structure de documents XML. Ce langage n'est pas vu comme un langage d'ontologies car les primitives qu'il met en place sont plutôt orientées application que concept. La sémantique définie dans le document est interprétable dans le contexte de l'opération faite sur le document mais ne permet pas d'établir des inférences en dehors de ce contexte. Pour conclure, on peut dire que XML et *XML-schema* sont considérés comme des langages définissant le format de *"message"* alors qu'un langage d'ontologies a pour but de *"définir et représenter"* la connaissance.

Resource Description Framework : RDF

Le *Resource Description Framework* est créé en 1999 pour décrire les métadonnées dans l'objectif de traiter l'information automatiquement, de favoriser l'interopérabilité des connaissances, d'encoder, d'échanger et de réutiliser des méta-données structurées. Ces méta-données peuvent aussi bien être descriptives que relatives aux contenus des granules. RDF, a été créé pour gérer les méta-données de documents XML mais peut également être utilisé pour des ontologies. Il permet de définir des ressources avec des propriétés et des états[Lass99]. La description des connaissances via RDF peut être résumée comme suit :

- RDF est le moyen d'exprimer des relations ;
- Ces relations sont décrites sous forme de graphe ;
- Chaque nœud du graphe est une ressource ou une valeur ;
- Chaque nœud est relié à un autre par un arc "nommé".

RDF-Schema définit les relations entre ces ressources. Le pouvoir sémantique de ces deux langages est limité car les axiomes ne peuvent pas être directement décrits. Le type des relations (symétrique, transitive, . . .) ne peut pas être spécifié.

XOL

XML based Ontology Exchange Language a été créé pour échanger des ontologies se rapportant à la biologie moléculaire mais est applicable à d'autres domaines. Cependant, les relations entre concepts ne peuvent pas être spécifiées correctement[Karp00].

Un *Ontology Inference Layer (OIL)* a été créé juste après. Il présente à la fois un langage de représentation et d'échange pour les ontologies. Il combine les primitives des langages reposant sur les frames avec une sémantique formelle et des possibilités de raisonnement issues de la logique de description. Pour être utilisé sur le Web, il repose sur les standards RDF(S) et XML. *OIL* permet de définir des classes et des relations et un nombre limité d'axiomes. Les relations sont considérées comme des classes et peuvent être organisées hiérarchiquement.

Ontologie Web Language : OWL

En 2004, *Ontologie Web Language* devient un standard actuellement proposé par le W3C pour représenter les ontologies. OWL se veut plus représentatif du contenu du Web que XML, RDF et RDF-Schema en apportant un nouveau vocabulaire avec une sémantique formelle qui permet de traiter le contenu de l'information et non plus uniquement à présenter l'information. Il ajoute du vocabulaire pour décrire les propriétés et classes, comme par exemple la disjonction de classe, la cardinalité (exactement un), l'égalité, les types de propriétés plus riches, les caractéristiques de propriété (symétrie, transitivité, ...) et les classes énumérées. Une ontologie formalisée en OWL comprend la définition des classes, des propriétés et des instances.

OWL est décliné en trois sous langages d'expressivité croissante : OWL lite, OWL DL, OWL Full.

OWL LITE

Il permet d'établir une hiérarchie de concepts simples, contraintes simples. Il permet de définir facilement des thésaurus ou taxonomies.

OWL DL

DL signifie Description Logique. Ce language comprend toutes les structures de OWL, et possède une expressivité plus importante, avec complétude de calcul. Il repose sur les éléments OWL auxquels il associe un grand nombre de restrictions (par exemple, une classe peut être une sous-classe de nombreuses autres classes, mais pas une instance d'une classe). *OWL DL* est conçu pour pouvoir supporter la logique de description. En effet, cette logique appartient à un domaine de recherche qui a pour but d'aider au raisonnement sur une base de connaissances.

OWL FULL

Il permet de fournir une expressivité maximale, une liberté syntaxique sans garantie de calcul : une classe peut également correspondre à l'instance d'une autre classe. Il lève les contraintes imposées par OWL DL pour rendre certaines valeurs disponibles et utilisables dans des bases de données ou de connaissances, mais il ne supporte pas les raisonnements liés à la logique de description. L'utilisation du langage de représentation

OWL dans le cadre de la gestion de la sensibilité au contexte permet d'une part de faire reposer le système sensible au contexte sur un standard mais surtout d'utiliser un langage incrémental. Dans un premier temps, les ontologies que nous considérons pourront être représentées à partir de OWL-Lite, puis elles évolueront vers un autre sous-langage lorsque le système sera capable de prendre en compte le niveau de formalisation spécifié.

3.2/ UN MODÈLE POUR LA SENSIBILITÉ CONTEXTUELLE À BASE D'ONTOLOGIE DE TRAÇABILITÉ

Cette section présente la description du processus de développement d'ontologie dans le domaine de la sensibilité au contexte. Les descriptions portent sur le processus de développement qui est constitué d'une suite des tâches successives pour la définition de nos ontologies dédiées à la gestion de la sensibilité contextuelle.

3.2.1/ LE PROCESSUS DE DÉVELOPPEMENT

Les différentes tâches présentées dans le manuscrit viennent pour décrire le processus de développement et de la conceptualisation d'une base de connaissances. En effet, l'activités de conceptualisation de l'ontologie nécessite le suit d'un ensemble des tâches à réaliser. Par exemple dans les travaux de [Corc03] dans Methontology, il propose d'encapsuler toutes les taches de conceptualisation de l'ontologie dans une seule tache de planification des ontologies crées. Dans notre figure 3.1, nous avons illustré l'ensembles des composant de l'ontologies voir importants et suffisants dans le processu de développement (les concepts, les attributs, etc) construits durant chaque tache.

Tâche 1

La construction de dictionnaire des termes : un glossaire est un réseau de termes contrôlés. Il contient généralement tous les termes importants pour un domaine donné (les concepts, les attributs, les relations entre concepts, . . .). Il doit inclure les descriptions en langage naturel, les synonymes et les acronymes.

Tâche 2

La définition des taxonomies de concepts : généralement chaque glossaire lié à un domaine donné a un nombre important de termes. La taxonomie permet de faciliter la recherche d'un terme en fonction de ses relations hiérarchiques avec d'autres termes. Elle est considérée comme une sorte de vocabulaire contrôlé, introduisant une notion de hiérarchie, ce qui permet d'élargir ou de resserrer une recherche à partir d'un terme. Les taxonomies s'attachent essentiellement à organiser les méta-données portant sur les concepts. Pour créer une telle taxonomie, nous devons tout d'abord suivre une démarche basée sur la sélection de l'ensemble des termes qui vont être considérés comme des

classes (concepts), des sous-classes, puis construire une hiérarchie basée sur des relation taxonomiques (disjonction,partition, . . .).

Les notions de base pour une taxonomie sont :

- Une classe : le concept C_1 permet de définir un groupe d'individus possédant des propriétés similaires. Les classes peuvent inclure des sous-classes équivalentes, des sous-classes disjointes, des partitions. En effet, OWL permet de déclarer que 2 classes sont équivalentes : ***equivalentClass*** lorsqu'elles ont les mêmes instances. Inversement on peut déclarer que 2 classes sont disjointes : ***disjointWith*** lorsqu'elles ne peuvent pas avoir des instances communes. Pour ***la partition***, c'est lorsque l'ensemble de sous-classes d'un concept C qui ne partagent pas de cas commun mais qui couvrent le concept C.
- Une sous-classe de : un concept C_1 est sous-classe d'un autre concpet C_2, si et seulement si toutes les instances de C_1 sont aussi des instances de C_2.

Tâche 3
La validation de diagramme de relations : cette étape consiste à bien valider le diagramme qui permet d'établir les relations entre les concepts d'une ou plusieurs taxonomies de concepts.

Tâche 4
La construction de glossaire de concepts : ce glossaire doit inclure tous les concepts du domaine, leurs relations, leurs instances, leurs attributs de classes et d'instances.

Tâche 5
L'annotation des relations : toutes les relations incluses dans le glossaire des concepts doivent être décrites d'une manière détaillée. En effet, pour chaque relation il faut préciser le nom, les concepts sources et destination, la cardinalité, . . . Pour la déclaration de cardinalités sur les propriétés des classes par exemple, nous pouvons citer les trois niveaux de cardinalités :

- minCardinality : toute instance de la classe est liée par la propriété *a-au-moins* **X** individus. Par exemple, la propriété *mother-of* de la classe Mother : $minCardinality = 3$;
- maxCardinality : toute instance de la classe est liée par la propriété *a-au-plus* **X** individus. Par exemple, la propriété *mother-of* de la classe Mother : $maxCardinality = 10$;
- cardinality : toute instance de la classe est liée par la propriété *a-exactement* **X** individus. Par exemple, la propriété *son-of* de la classe Mother : $cardinality = 1$.

Tâche 6 :
La description des attributs d'instances : les attributs d'instances sont ceux qui décrivent les instances d'un concept et leurs valeurs qui peuvent être différentes pour chaque instance du concept. Il faut spécifier pour chaque attribut d'instance le nom, le concept auquel il appartient, le type de valeur, la cardinalité, . . .

Tâche 7
La description des attributs de classes : pour chaque attribut de classe, il faut spécifier le nom, l'endroit ou le concept est défini, le type de valeur, la valeur et la cardinalité.

Tâche 8
La description des constantes prédéfinies : pour chaque attribut de classe, il faut spécifier le nom, le type de valeur, la valeur et l'unité de mesure.

Tâche 9
La description des axiomes formels : à cette étape, il faut définir toutes les expressions qui sont toujours vraies. Pour chaque définition d'axiome formel, il est important de spécifier le nom, la description, l'expression logique qui le décrit formellement (la logique de premier ordre par exemple), les concepts attachés, les attributs et les relations auxquelles l'axiome fait référence.

Tâche 10
La description des règles : chaque ontologie doit contenir un ensemble de règles bien définies. Pour chaque règle, il faut spécifier le nom, la description, l'expression qui la décrit formellement, les concepts, les attributs, les relations auxquelles elles font référence. Pour la définition des règles, nous utilisons la syntaxe suivante :

Si ⟨condition⟩ alors ⟨conséquences et actions⟩

Tâche 11
La description des instances : pour chaque instance, il faut spécifier le nom, le concept auquel il appartient et les valeurs des attributs. Une fois que ces dernières tâches sont effectuées, l'ontologie est ainsi bien normalisée, formalisée et opérationnelle. En effet, dans la phase de normalisation, il s'agit de créer des primitives du domaine. Les étapes de conceptualisation sont présentées dans la figure 3.1.

Les méthodes d'ingénierie ontologique consistent à définir les procédures de travail, les étapes, qui décrivent le pourquoi et le comment de la conceptualisation puis de l'artefact construit. En effet, les différentes tâches citées précédemment présentent les étapes générales de développement des ontologies qui permettent un engagement sémantique en introduisant une normalisation sémantique des termes manipulés dans l'ontologie. La méthode de normalisation suit trois étapes principales à savoir la normalisation sémantique, la formalisation et l'opérationnalisation des connaissances.

La phase de normalisation consiste à bien choisir les termes du domaine et à les normaliser en explicitant leurs propriétés et en exprimant les identités (l'ensemble des caractéristiques sémantiques génériques) et les différences (l'ensemble des caractéristiques sémantiques spécifiques tout en prennant en considération l'opposition des unes avec les autres) dans leur voisinage proche. Cette structuration de terme en fonction des identités et des différences permet de passer vers une ontologie différentielle.

L'objectif à ce stade est alors de rendre explicite le sens des expressions linguistiques et donc de modéliser les primitives nécessaires à la représentation de l'ensemble des connaissances à partir des expressions linguistique dont on dispose déjà. D'une manière

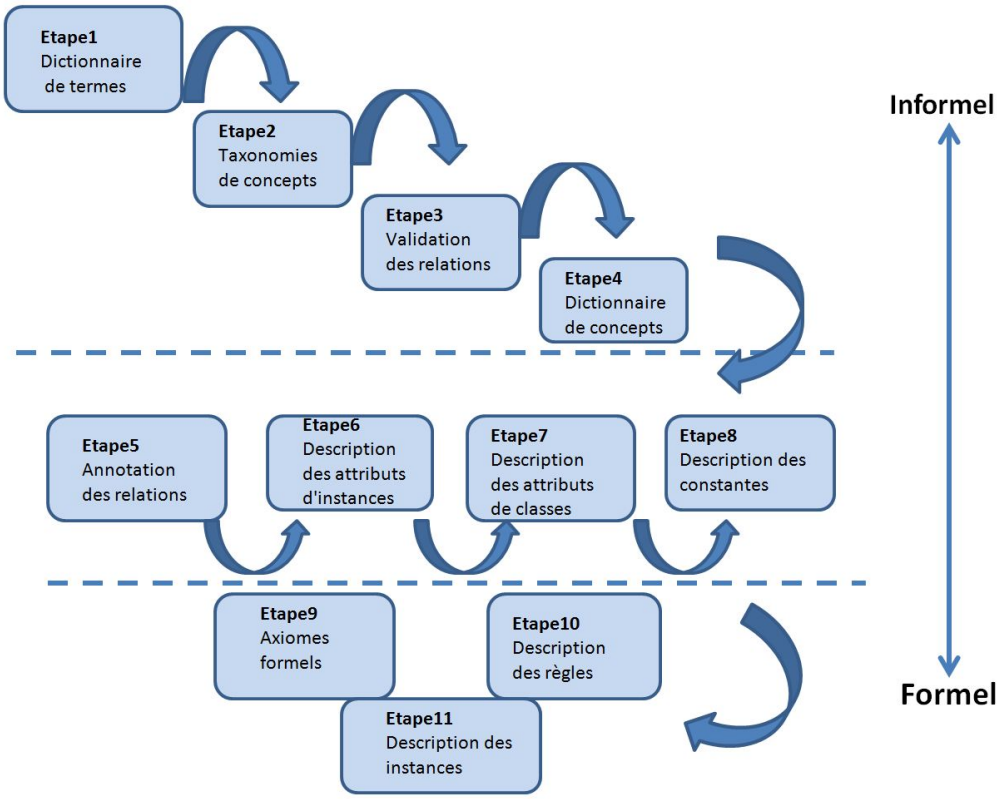

FIGURE 3.1 – Le processus de conception des ontologies

générale, il n'existe pas de primitive générale d'un domaine. En prenant l'exemple en médecine, il est difficile, voir même impossible, de présenter les notions à partir desquelles toutes les connaissances seront construites. Ce qui est important à ce niveau n'est pas la caractérisation ou la détermination de primitives déjà existantes dans un domaine, mais la modélisation ou la construction de primitives pour la résolution du problème. Dans l'ingénierie ontologique, la définition d'une ontologie consiste à la normalisation des connaissances qui permet de les formaliser selon les exigences du domaine à représenter.

L'étape de formalisation des connaissances consiste à bien formaliser les connaissances à représenter à travers une formalisation de l'ensemble des relations susceptibles d'exister entre les concepts tout en définissant l'ensemble des extensions qu'elles relient. En effet, elle consiste à désambiguïser les notions de l'ontologie référentielle obtenue dans l'étape de normalisation sémantique, à travers la définition des concepts et leur signification et non plus des termes, selon la sémantique formelle déjà définie. Une sémantique formelle et extensionnelle nécessite la création de nouveaux liens permettant de les relier à un ensemble de référents des objets du domaine (les extensions des concepts). Cela nécessite la création de nouveaux concepts formels, des propriétés et des axiomes à travers de nouvelles opérations ensemblistes telles que l'union, l'insertion, ... Une structure hiérarchique sous formes de graphes conceptuels est créée à travers les extensions générées des concepts, ainsi que de l'héritage multiple.

Pour finir, la dernière étape de développement de l'ontologie consiste en l'opérationnalisation des connaissances dans laquelle le système informatique utilise un langage opérationnel de représentation de connaissances (RDF, OWL) et permet ainsi de manipuler les concepts en fonction de leur interprétation sémantique (en suivant les règles et les opérations associées). Finalement une sémantique computationnelle sera générée pour chaque concept défini : dans laquelle chaque concept est défini comme la spécification d'un calcul. Ainsi est générée une ontologie computationnelle.

3.2.2/ LA DÉFINITION DES ONTOLOGIES GÉNÉRALES POUR MODÉLISER LA SENSIBILITÉ CONTEXTUELLE

Un système est dit sensible au contexte s'il peut récupérer, gérer, interpréter et utiliser des informations issues du contexte et adapter sa réponse en fonction du contexte d'utilisation. En effet, ces informations sont explorées sous forme d'un vocabulaire caractérisé par une variété de situations de contexte à traiter. En représentation des connaissances dans notre domaine de la sensibilité au contexte, un module est constitué d'un vocabulaire spécifique utilisé pour décrire un modèle du monde réel : vocabulaire auquel est associé un ensemble d'hypothèses (des assertions) sur le sens qui doit être attaché aux éléments de la base de connaissance. Devant l'ambiguïté qui peut être générée au niveau du traitement automatique de l'information de contexte, un niveau de standardisation du langage est nécessaire à travers une spécification explicite fournie par les ontologies. Cette spécification correspond à un modèle abstrait d'une partie du monde réel qui se présente comme un ensemble de définitions de concepts munis de propriétés et de relations entre ces concepts, ce qui fait l'intérêt de définir nos ontologies pour faire face au sensibilité contextuelle.

Bien au-delà, une telle structure hiérarchique donnée par l'ontologie nous permet de raisonner sur les informations représentées formellement (à travers un langage approprié : OWL) dans ces ontologies afin d'adapter les applications interactives pervasives.

À ce stade, la conceptualisation de notre ontologie de contexte ***OC*** peut être décrite comme étant un système formel qui est constitué de :

- Un ensemble ***CC*** de Concepts de Contexte organisés en une hiérarchie ***H*** ;
  Dans laquelle, les Concepts de Contexte sont reliés par des relations de spécialisation (subsomption) $\sqsubseteq$ (un ordre partiel),
  Et, $CC_1 \sqsubseteq CC_2$ signifie que CC_1 est un sous-concept de CC_2.

- Un ensemble ***RC*** de Relations de Contexte ;

- Un ensemble ***AC*** d'Axiomes de Contexte permet d'axiomatiser les propriétés et les relations.

Pour mettre à disposition des services utiles et sensibles au contexte d'utilisation de l'utilisateur, l'adaptation dépend de paramètres récupérés selon ses tâches, ses activités courantes, sa localisation et ses terminaux dans une situation contextuelle donnée. Avec cette définition, toute application qui repond aux paramètres (variables) de contexte est une application sensible à ce contexte En effet, la définition de nos différentes ontologies, celles de l'utilisateur (UO), du terminal ou device (DO), du Service (SO) et de l'environnent(EO) ont pour objectif de fournir un environnement pour l'intégration des ressources de contexte. La base de connaissance sera utilisée par la suite pour son in-

terprétation selon les différents paramètres qui décrivent les différentes situations contextuelles susceptibles d'être présentes dans une application pervasive.

La tâche de conceptualisation et d'analyse des données utilisées dans l'ontologie demande un effort et une concentration double pour les éditer (via un éditeur d'ontologies comme Protégé). La construction d'une base de connaissance est faite à travers un langage dédié à la spécification de tous les vocabulaires, des termologies et surtout des sémantiques associées à ces différentes connaissances. En effet, l'obtention d'une telle base est faite à travers une source d'informations et de connaissances (comme états de l'art, documents, portails destinés aux utilisateurs, ...).

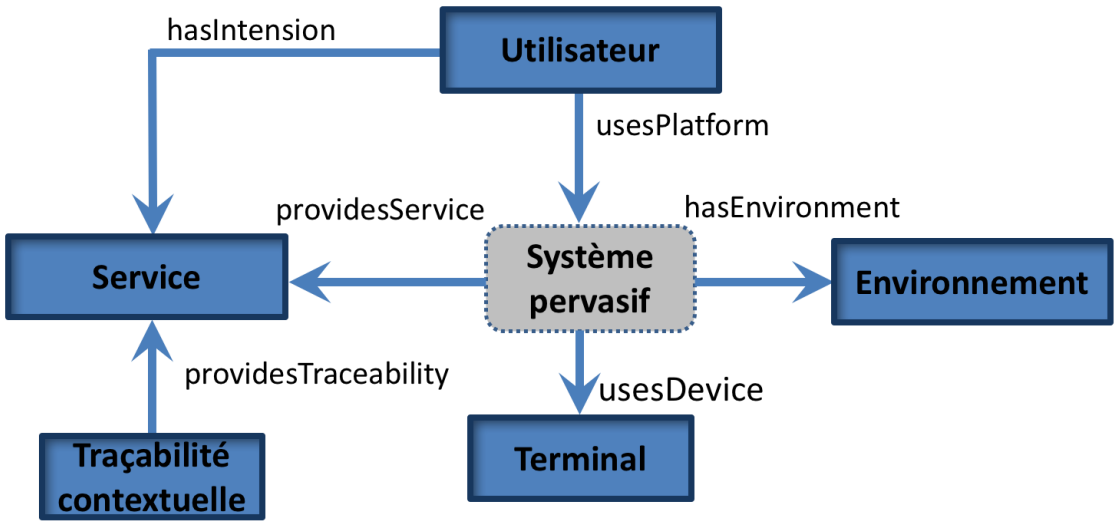

FIGURE 3.2 – Ontologie Conceptuelle de Contexte

Dans ce contexte, nous intégrons le modèle de contexte de R. Hervas dans une partie de notre plateforme comme nous l'avons argumenté précédemment. Cette intégration nous a permis de valider la couche représentation et gestion de la sensibilité de contexte. En effet, nous exprimons la représentation de contexte sous la forme d'une ontologie à deux niveaux à savoir : un niveau de base et un deuxième niveau spécifique.

Le niveau de base contient tous les concepts généraux du contexte d'utilisateur (terminal, environnement, ...). Ce niveau de base permet de préparer la phase d'adaptabilité au contexte qui vient dans l'étape suivante du processus de la sensibilité au contexte.

Concernant le niveau spécifique, d'une manière générale, ce niveau contient toutes les sous-classes des concepts déjà définis dans le niveau de base(cf figure 3.2).

Par exemple, pour le concept *Utilisateur*, nous avons défini un ensemble de sous classes permettant de définir un staff hospitalier (médecin, infirmier, résident,...). Dans le domaine de la sensibilité au contexte, l'ensemble des concepts de cette ontologie à deux niveaux est renforcé par une liaison entre ces différents concepts à travers la définition de l'ensemble des relations représentées entre les concepts par des prédicats : pour désigner par exemple l'emplacement d'un utilisateur :

*Est*Localisé*(Utilisateur, Localisation)*
*Est*Localisé*(Médecin, Salle opération)*

L'ONTOLOGIE D'UTILISATEUR

Généralement, dans le développement des applications interactives pour les environnements pervasifs, le focus est mis sur l'utilisateur et ses attendus. Il s'agit d'une démarche centrée utilisateur qui doit prendre en compte ces exigences lors de la conception d'un tel système. Ces exigences peuvent se présenter sous différentes formes, en fonction de l'environnement et de ses composants. Pour décrire notre modèle Utilisateur, nous répondons aux différentes questions posées dans la description du modèle d'utilisateur PIVon : modèle mentionné au chapitre précèdent, PIVon d'Hervas est notre modèle retenu comme référence.

Tout d'abord, nous décrivons les caractéristiques statiques d'un utilisateur dans son profil afin que nous puissions répondre à la première question « *Quelles sont les caractéristiques de l'utilisateur ?* ». Ce niveau comprend des données personnelles, des intérêts, des affiliations, . . . Toutes ces informations sont dynamiques et peuvent être soumises à des changements fréquents avec l'évolution des exigences et des préférences des utilisateurs. Nous répondons à la deuxième question « *Qu'est-ce que l'utilisateur veut faire ?* », tout en planifiant toutes les activités susceptibles d'être exécutées par l'utilisateur dans une situation contextuelle donnée dans un environnement pervasif. En effet, la notion de *situation contextuelle paramétrée*, nous est utile dans le processus de développement lors de la récupération des paramètres de contexte sous forme des traces de contexte « *tracability* » que nous allons utiliser. Nous constatons, à ce niveau, que le fait de répondre à ces différentes questions ne permet pas d'avoir une définition universelle de la situation de l'utilisateur incluant les questions de dynamicité, d'évolutivité et de mobilité. La diversités des exigences des utilisateurs face à ces systèmes pervasifs nécessite une forte corrélation entre les éléments constituant le modèle général de contexte .

Dans notre modèle descriptif d'utilisateur (cf figure 3.3), nous proposons de spécifier un ensemble de fonctionnalités considérées comme pertinentes pour adapter les services offerts des systèmes pervasifs. Nous considérons que cette spécification est en fait l'objet d'une corrélation **[fonctionnalité/service]**. Quant à la fonctionnalité, elle est définie en fonction des paramètres récupérés suite à une situation contextuelle courante. Elle est définie en fonction du triplet **[Rôle ⇒ Tâche ⇒Activité]**. Cette fonctionnalité est exécutée en bloc de ces trois derniers éléments. Nous considérons que cette fonctionnalité d'utilisateur face à son système pervasif représente *ses Intentions*. Ces intentions représentent les attendus de l'utilisateur en termes de services offerts dans une situation contextuelle donnée. Un utilisateur peut avoir n intentions et cela évolue avec le changement de sa situation contextuelle par rapport à son environnement courant et à son système pervasif. Chaque fonctionnalité se définit en fonction des paramètres d'entrée (traces d'entrées dans le contexte d'utilisation) afin de définir les sorties adaptées aux attendus des utilisateurs pour le contexte en cours.

Une fonctionnalité nous permet d'informer sur un ensemble de paramètres de contexte d'utilisation. Les éléments constituant le bloc de la fonctionnalité sont sémantiquement définis dans notre modèle à base d'ontologie avec le reste des éléments de contexte. Dans notre ontologie d'utilisateur, une relation de subsomption est réalisée entre les préférences d'utilisateur (comme par exemple une préférence pour l'utilisation de polices d'affichage), et ses intentions face à l'application pervasive. En effet, lorsqu'un utilisateur se connecte en définissant son rôle dans l'application, il produit une demande pour effectuer une tâche, ce qui peut être subdivisé en plusieurs activités.

Dans ce cas-là, une récupération des traces du contexte en cours est réalisée. Ces paramètres récupérés seront intégrés dans la chaîne d'adaptation de l'information au contexte d'utilisation, afin de mieux répondre à différents besoins spécifiques des utilisateurs.

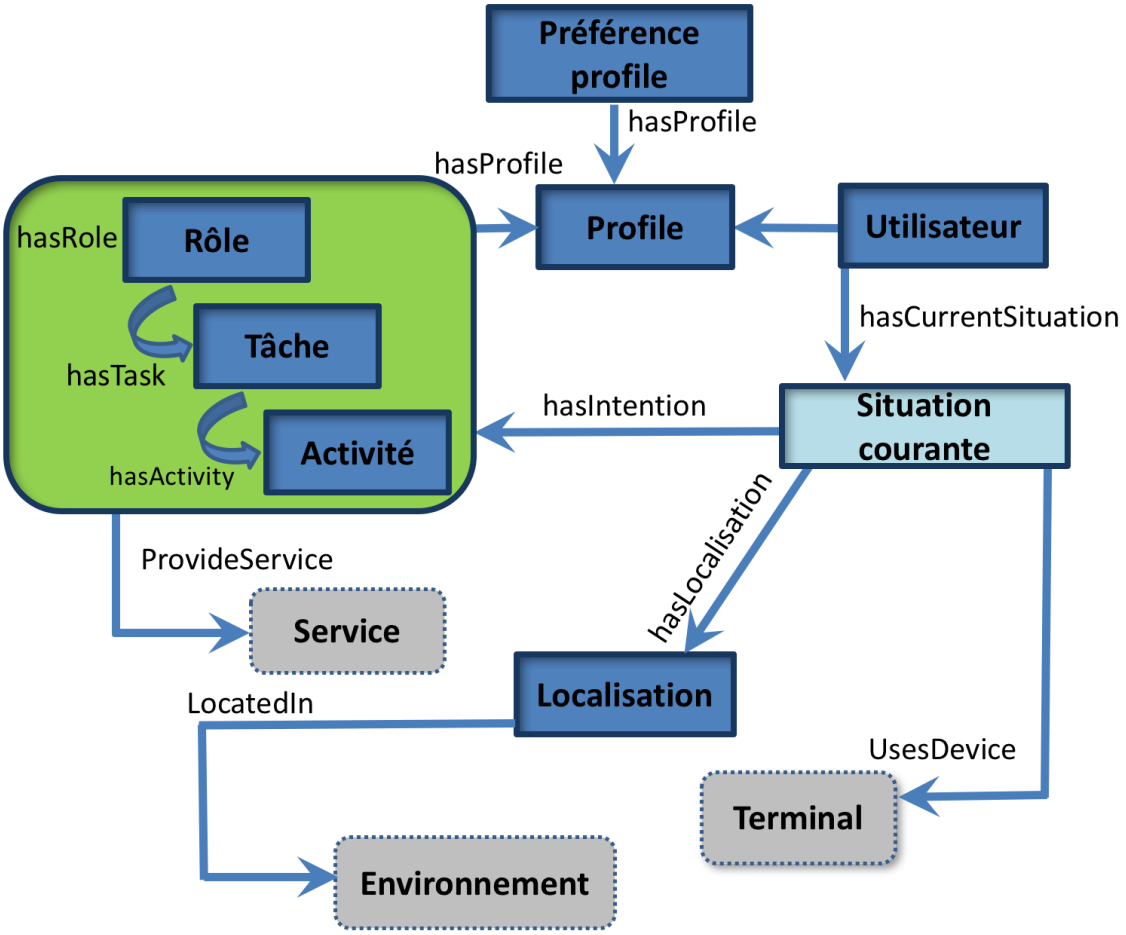

FIGURE 3.3 – Ontologie Conceptuelle d'Utilisateur

L'ONTOLOGIE DE TERMINAL

Les smartphones, les tablettes, les nouveaux réseaux 3G, 4G, ...et toutes les autres technologies qui apparaissent ces dernières décennies ont bouleversé nos habitudes et donc les usages de ces différents systèmes qui prennent place dans notre quotidien. Avec l'utilisation de ces diverses technologies, l'information pervasive est devenue très importante dans le sens où elle peut nous informer sur les ressources informatiques continuellement utilisées dans des environnements pervasifs. La mobilité apportée par ces nouvelles technologiques a étendu l'échelle de la sensibilité de contexte. En effet, comme nous l'avons mentionné précédemment, l'utilisateur joue un rôle important dans le développement des applications interactives dans différents environnements pervasifs. Dans ce contexte, une relation centrale explicite lie fortement cet utilisateur à son terminal. Cette relation peut être exprimée à travers l'évolution de contexte d'utilisateur face

à son environnement afin de gérer la mobilité à cet environnement Cette forte relation permet d'adapter les services, les terminaux au sein de son environnement avec des technologies variées.

Dans notre domaine, il est primordial que notre application puisse disposer des informations sur le contexte dans lequel elle opère. Ainsi, divers types d'informations devraient être assemblés pour former une représentation du contexte de l'appareil sur lequel cette application fonctionne. Il est nécessaire que la terminologie de contexte soit communément comprise par tous les appareils participants. Dans ce contexte, nous proposons une ontologie de terminal de contexte. Cette ontologie de l'appareil vise à être suffisamment générale en évitant les contraintes restrictives et en fournissant une organisation taxonomique pour aider à la spécialisation de notre application sensible au contexte. Nous considérons que l'information sur les terminaux peut être définie dans un modèle descriptif à base d'ontologie. En référence à la méthodologie de conception du modèle de R. Hervas [Herv10] [Herv11] les points suivants sont nécessaires :

- La conceptualisation d'un profil détaillé de terminal : c'est une étape indispensable qui permet de décrire d'une manière générale et détaillée les terminaux susceptibles d'être utilisés. Ainsi, une définition des propriétés générales est importante à ce stade (par exemple la disponibilité de l'appareil et son emplacement dans l'environnement) et les propriétés spécifiques (en cas d'un fonctionnement particulier de l'appareil) ;
- La définition dans notre modèle propose des relations de dépendance entre les appareils au niveau de la compatibilité, au niveau des associations et des liaisons entre les appareils. . . ;
- La définition de l'organisation taxonomique créée différentes familles de dispositifs.

Dans notre ontologie de terminal (cf figure 3.4), nous décrivons toutes les caractéristiques matérielles et logicielles nécessaires à l'adaptation au niveau du service offert. La description des caractéristiques du dispositif est associée à la sémantique correspondante. Nous réalisons la conceptualisation hiérarchique qui facilite la réutilisation et le partage des informations du terminal avec les autres composants du modèle de contexte. Comme l'environnement courant de l'application doit prendre en charge l'acquisition des caractéristiques des différents terminaux susceptibles d'êtres présents, la mise à niveau de l'information globale du terminal est nécessaire. Nous créons, par exemple, des relations entre le modèle de l'appareil et celui de service afin d'offrir un service en fonction des caractéristiques logicielles de cet appareil. Ainsi, dans notre modèle ontologique de terminal, nous définirons l'état courant de l'appareil, à travers la définition de ses propriétés générales (par exemple la disponibilité de l'appareil et son emplacement sur la base du modèle de l'environnement) afin de permettre la spécialisation de l'état de fonctionnement de l'appareil.

Nous définissons ainsi les relations de dépendances des terminaux les uns par rapport aux autres tout en exprimant le niveau de compatibilité, les relations de liaison, . . . Nous proposons ainsi une organisation taxonomique qui est faite à travers le classement des terminaux selon leurs types (dispositifsAutonomes, dispositifsActionnaires, . . .).

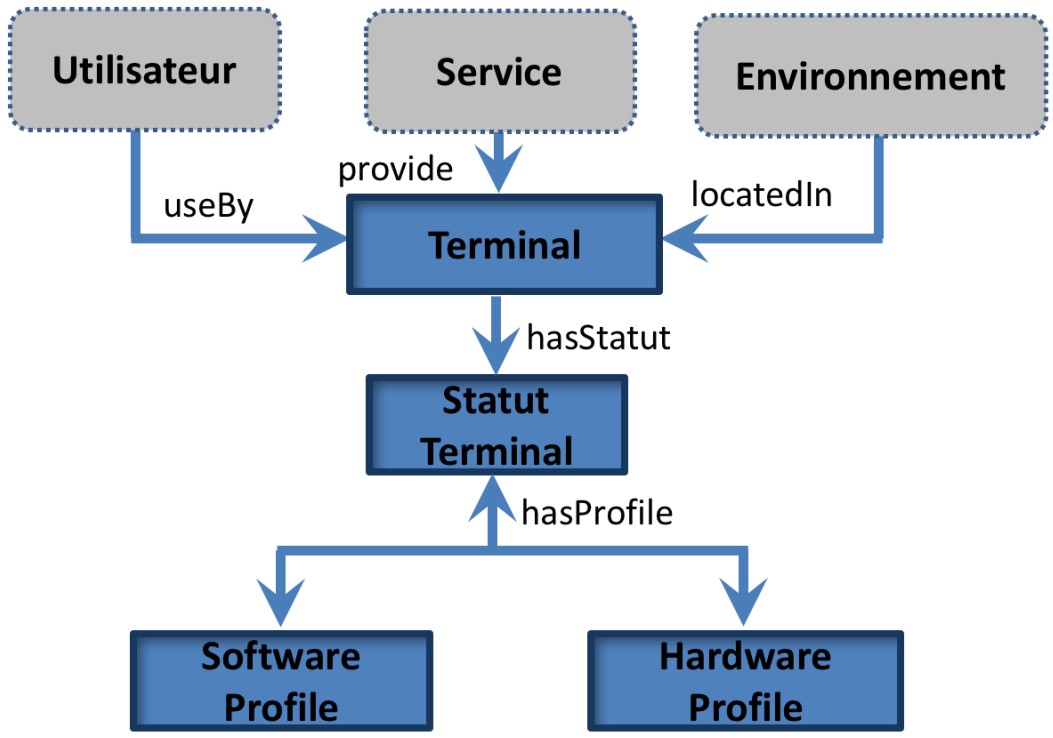

FIGURE 3.4 – Ontologie Conceptuelle de Terminal

L'ONTOLOGIE DE L'ENVIRONNEMENT

Le changement des situations contextuelles ne se limite pas au changement du type de terminal ou des intentions de l'utilisateur mais aussi à tout changement de l'environnement de l'utilisation de l'application pervasive. Ce changement influence la définition des vues et des services d'adaptation attendus. Au sein d'un environnement pervasif, la mobilité est un aspect clé et l'utilisateur interagit par le biais de divers dispositifs, avec divers intentions, afin d'avoir des services satisfaisant ses besoins (cf figure 3.5).

Cet environnement peut se caractériser par certaines conditions environnementales, ce qui permet aux systèmes de fournir des informations de contexte suite à son comportement. L'emplacement se présente également comme une dimension importante dans les environnements d'interaction. Par exemple, en raison d'un trouble météo et par exemple de fortes pluies et un ciel sombre, le système pervasif pourrait décider d'allumer la lumière. En revanche, avec le changement de la situation et de la localisation de l'utilisateur, cela peut ne pas être nécessaire d'allumer la lumière au milieu de la nuit. Nous exprimons toutes les propriétés sémantiquement tout en proposant une organisation taxonomique en fonction des espaces et des zones de localisation ainsi qu'en mettant cette présente ontologie en relation avec le reste du modèle de contexte (ontologie de service, ontologie de terminal, ...).

L'ONTOLOGIE DE SERVICE

Pour les applications pervasives, les services sont généralement considérés comme des extensions des Services Web existants (nous notons, à titre d'exemple, la messagerie,

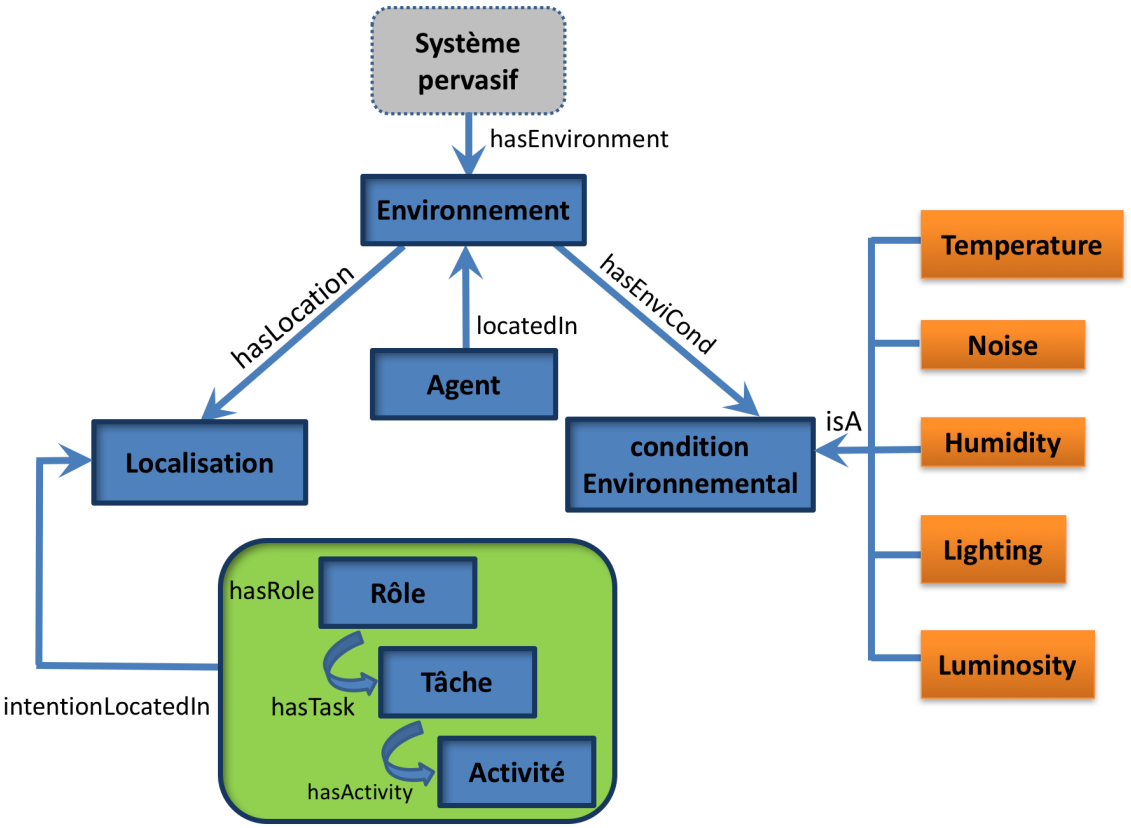

FIGURE 3.5 – Ontologie Conceptuelle de l'Environnement

les applications multimédias, . . .) tout en reprenant les principes de base avec une grande attention sur comment les rendre adaptables au contexte d'utilisation.

Comme nous l'a montré l'étude de la littérature, la sensibilité au contexte est devenue un élément central pour la conception et la mise en place de services adaptatifs. Nous décrivons notre ontologie de service tout en le référant à une vision centrée utilisateur. En effet, les services offerts aux divers utilisateurs fournissent des fonctionnalités qui par leur utilisation facilitent ou aident l'utilisateur dans l'accomplissement de ses tâches. L'adaptation de tels services nécessite à la base la prise en compte des habitudes utilisateurs, des caractéristiques des dispositifs qu'ils utilisent, des changements de l'environnement dans lequel évoluent ces utilisateurs.

En se basant sur la définition de [Wino01], le contexte est défini comme étant l'ensemble des paramètres externes de l'application qui peuvent influencer sur son comportement en définissant de nouvelles vues sur ses données et ses fonctionnalités. Ces paramètres peuvent être dynamiques et peuvent donc changer durant l'utilisation de l'application. Compte tenu de l'importance de l'adaptation de contexte surtout au niveau de la personnalisation des services offerts et au niveau de l'adaptation de l'interface au contexte d'utilisation, nous considérons que le contexte peut être défini selon les différentes « traces externes » de la situation courante de contexte d'utilisation de l'application pervasive.

Or une situation contextuelle est caractérisée par une suite de paramètres qui peuvent être récupérés comme des traces utiles pour l'adaptation en temps réel et d'une manière

dynamique aux changements successifs de contexte d'utilisation. Ce type de systèmes de traçabilité est mis en place afin de récupérer tous les paramètres sous la forme de traces de l'environnement extérieur liées à une situation contextuelle courante instantanée : il permet le suivi de l'évolution des besoins des utilisateurs en termes des services souhaités (cf figure 3.6).

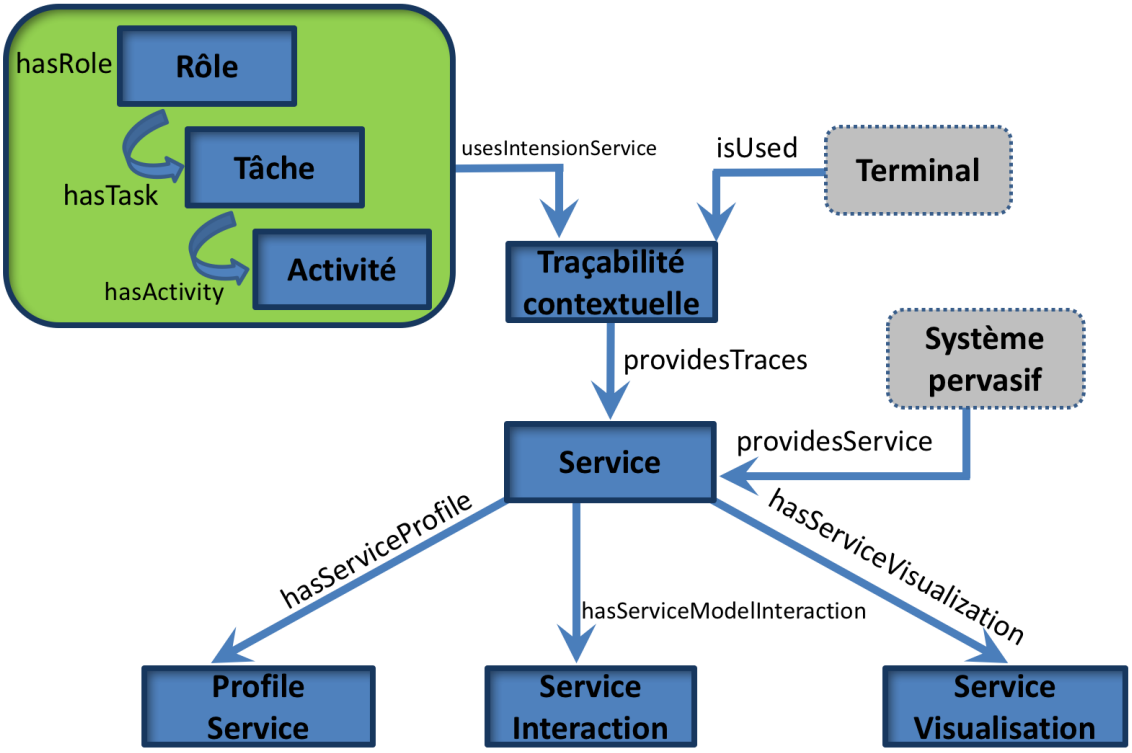

FIGURE 3.6 – Ontologie Conceptuelle de Service

Pour la construction de notre ontologie de service, nous organisons notre modèle afin :

- de donner une grande importance aux relations entre les éléments retenus du contexte et la représentation visuelle des données. Un service est offert à un ou plusieurs utilisateurs avec des caractéristiques spécifiques, au sein de l'environnement pervasif, et présentés au travers de dispositifs possédant des caractéristiques différentes au niveau des fonctionnalités. Pour faire face aux différentes contraintes, nous proposons un module qui permet de gérer la traçabilité contextuelle pour toutes les situations courantes des utilisateurs à un instant t d'utilisation.
  Ce fragment est défini comme un ensemble de classes de traçabilité (cf figure 3.7) :
  - La Classe Entité *(Entity class)* : cette classe définit les paramètres de connexion de l'entité. Une entité peut être soit l'utilisateur (Agent humain) soit le terminal (Agent technique) en fonction de leurs profils, dans la situation contextuelle courante.
  - La Classe Fonctionnalité *(Feature class)* : cette classe peut être également définie comme l'ensemble des intentions des utilisateurs face à son système pervasifs (tâche, activité, ...).

- La Classe Localisation *(Location class)* : cette classe définit les paramètres de localisation des entités dans l'environnement pervasif.

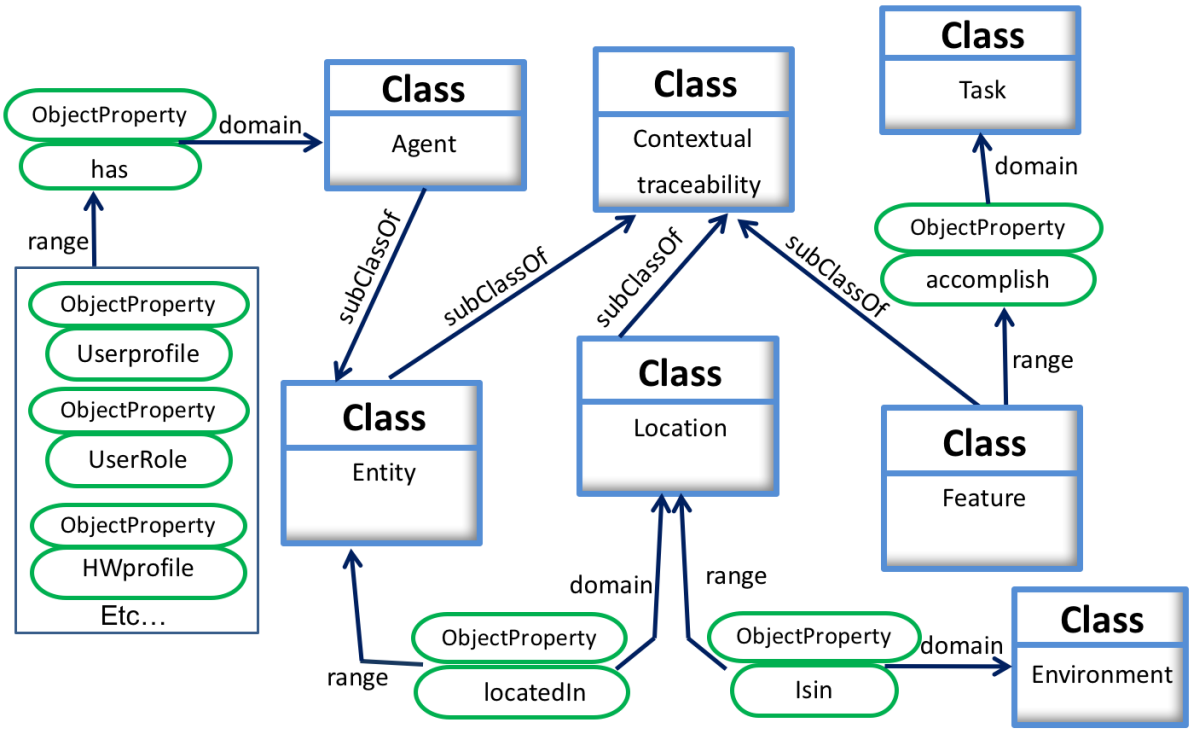

FIGURE 3.7 – Ontologie Conceptuelle de Traçabilité

En effet, la gestion de la traçabilité est réalisée à l'aide d'un mécanisme, que nous appelons Carte de Visite COntextuelle (*CVCO* que nous présentons dans la prochaine section), qui permet la récupération des traces instantanées de contexte à un instant donné. Ce mécanisme permet de nous informer sur le contexte courant de l'utilisateur dans son environnement pervasif.

- de traiter les questions sociales, de localisation et les caractéristiques techniques : les services face aux environnements pervasifs peuvent être optimisés en fonction de la définition instantanée des traces sociales, des localisations et des caractéristiques techniques. À ce stade nous considérons une forte corrélation entre la totalité du modèle de contexte et le modèle de services pour des fins d'adaptation. L'ontologie, que nous proposons, permet de spécifier un modèle, à l'aide de la traçabilité de contexte dans des situations d'utilisations instantanées, dédiées à l'adaptation de l'ensemble des services qui répondent le mieux aux besoins des utilisateurs des applications pervasives.

Il nous semble nécessaire de préciser que notre motivation n'est pas de proposer de nouvelles ontologies de modélisation du contexte, mais d'utiliser des standards d'ontologies de contexte, qu'ils sont déjà définis, et de les étendre et les améliorer afin qu'ils permettent de décrire la gestion de la sensibilité au contexte dans un environnement sensible au contexte qui réponds mieux à nos objectifs. En effet, pour la modélisation des principaux éléments de contexte qui aident à le sensibilisé, nous proposons une extension pour les différentes ontologies de base voire importante pour la modélisation

de éléments de contexte à savoir : l'ontologie d'utilisateur, de terminal, de l'environnement et de service, tout en les améliorant la plupart de ces ontologies et intégré une nouvelle ontologie de traçabilité qui sera alimentée par la CVCO. Ces différentes ontologies seront en communication et en interopérabilité entre eux afin de de raisonner sur les données décrites et donc de répond aux différents exigeants de système dans la plateforme COALA pour des raisons d'adaptation.

3.2.3/ GÉNÉRATION ET INTÉGRATION D'UNE ONTOLOGIE DE TRAÇABILITÉ CONTEXTUELLE POUR L'ADAPTATION DES SYSTÈMES PERVASIFS

Pour s'adapter au contexte, les applications pervasives doivent pouvoir prendre en compte le changement dynamique des situations d'utilisation dans l'environnement, de plus les informations de contexte en provenance de l'environnement doivent pouvoir parvenir aux applications entrantes. Ces flux d'informations ne peuvent pas être déterminés à l'avance et doivent se construire pendant l'exécution. Les modèles de gestion de l'information de contexte existant ne traitent pas ou peu cet aspect dynamique. Dans le domaine de la sensibilité au contexte, les applications interactives doivent êtres créées et développées d'une façon indépendante du contexte. Il est non seulement primordial de placer une séparation entre les données propres à l'application et les données de contexte mais également de récupérer les valeurs et d'identifier les paramètres du contexte dans leur évolution au cours de différentes situations d'utilisation. En effet, les caractéristiques de contexte externes jouent un rôle important dans la production des réactions d'adaptations fournies par les systèmes pervasives.

PROPOSITION D'UNE CARTE DE VISITE CONTEXTUELLE

Pratiquement, lors de l'utilisation de l'application, l'utilisateur se connecte à sa session en envoyant automatiquement et implicitement les informations et les propriétés de son contexte d'utilisation. À ce stade, l'utilisation d'un mécanisme qui permet de fournir les traces instantanées des situations contextuelles, sous formes de paramètres de contexte, sera utile non seulement pour l'adaptation des applications pervasives mais aussi pour leur amélioration.

Par ailleurs, nous constatons que l'utilisation d'un mécanisme qui permet de fournir la traçabilité des paramètres des situations contextuelles de l'utilisateur peut être une solution intéressante pour améliorer les résultats d'adaptation. Ainsi, nous proposons un mécanisme de récupération en temps-réel des différents paramètres externes (traces) liés à une situation contextuelle courante.

Ce mécanisme est considéré comme une Carte de Visite COntextuelle (CVCO) qui permet de fournir la traçabilité de contexte au cours de l'utilisation. Dans la carte CVCO, nous proposons d'utiliser des représentations XML simples et faciles à interpréter, afin de stocker et d'échanger les valeurs des paramètres d'une situation contextuelle donnée. Comme nous avons défini un ensemble de blocs pour modéliser le contexte (localisation, environnement, terminal, ...), nous définissons pour chaque bloc un élément XML qui permet de récupérer les paramètres courants lors de l'utilisation de contexte. Les paramètres constituant chaque élément ont été définis selon leur structure. Ceci permet d'une façon automatique de récupérer les traces de contexte d'une situation instantanée sous formes de paramètres de contexte et donc permet de proposer des nouveaux ser-

vices adéquats. À ce stade, l'intégration de la carte de visite de traçabilité contextuelle permet de fournir la traçabilité, dans notre modèle de contexte et nous permet de réaliser une stratégie de gestion et d'adaptation des applications sensibles au contexte à travers la fourniture de services dynamiques et personnalisés aux utilisateurs en temps-réel.

Ce mécanisme nous permet d'adapter automatiquement les interfaces utilisateurs des applications interactives pervasives. Le modèle de contexte à base de traçabilité sera intégré par la suite dans notre plateforme pour adapter les services au contexte.

Le fonctionnement de la carte CVCO, comme le montre la figure 3.8, permet de récupérer automatiquement les informations dynamiques de contexte lorsque les utilisateurs seront connectés à leur session de travail. Ces informations seront envoyées périodiquement en prenant en compte toutes les modifications au niveau du contexte d'utilisation. Il faut noter à ce niveau que, lors d'une session utilisateur, l'interaction entre cet utilisateur et son application est déterminée en fonction des paramètres de son contexte (la localisation par exemple, avec service à n'afficher qu'en fonction d'une certaine localisation). La carte CVCO permet donc de fournir toutes les informations utiles sur les propriétés matérielles et logicielles, l'emplacement courant pour un utilisateur courant.

Toutes ces informations et traces récupérées sont nécessaires pour gérer dans un premier temps et adapter des services en fonction de l'environnement de l'utilisateur. Les informations contenues dans la carte sont les contraintes et les traces de l'environnement, des utilisateurs, des rôles, des préférences, des terminaux, ... Grâce à ces traces, nous pouvons créer des profils de contexte dynamiques avec mise à jour en temps réel qui permet aux utilisateurs de disposer des services adaptés.

Ce concept de carte CVCO permet de prendre en compte un grand nombre d'informations (traces) nécessaires à l'alimentation de nos ontologies de contexte ainsi qu'au fonctionnement des applications interactives pervasives ayant besoin de mettre en œuvre l'adaptabilité. Le flux d'échange sera entre l'ontologie de service (où se situe l'ontologie de traçabilité) et le reste du modèle de contexte (User Ontologie, Device Ontologie, ...).

GÉNÉRATION AUTOMATIQUE D'UNE ONTOLOGIE DE TRAÇABILITÉ À PARTIR DE LA CVCO

Plusieurs stratégies de correspondance et de cartographie ont été proposées dans [Deck00], [Bohri05] [Rod06] [Ghaw09]. En effet, Certains des auteurs se sont concentrés principalement sur une correspondance entre XML et RDF, d'autres visent à faire la cartographie des données sous leur format *XML Schema* avec une correspondance OWL (en référant des instances de données XML). Concrètement, OWL est un langage plus fort avec une plus grande intelligibilité que RDF. Ainsi, OWL est livré avec un plus grand vocabulaire et une syntaxe plus forte que RDF, ce qui peut diminuer la probabilité de la perte de sémantique des données au niveau du processus de cartographie. Pour ces différentes raisons, nous choisissons d'utiliser une partie de fonctionnement de mapping XML (après la validation XSD) avec OWL proposé dans [Ghaw09] : compte tenu de son importance et de sa performance au niveau mapping.

Dans [Ghaw09], La méthode de mappage proposée, et appelée X2OWL, vise à la construction d'une ontologie OWL à partir des données XML. Cette méthode est basée sur le *XML Schema* qui permet de générer automatiquement la structure de l'ontologie, tout en définissant l'ensemble des règles de cartographie entre les éléments des données sources XML et l'ontologie à créer (cf table 3.1). Les règles de cartographie contribuent à

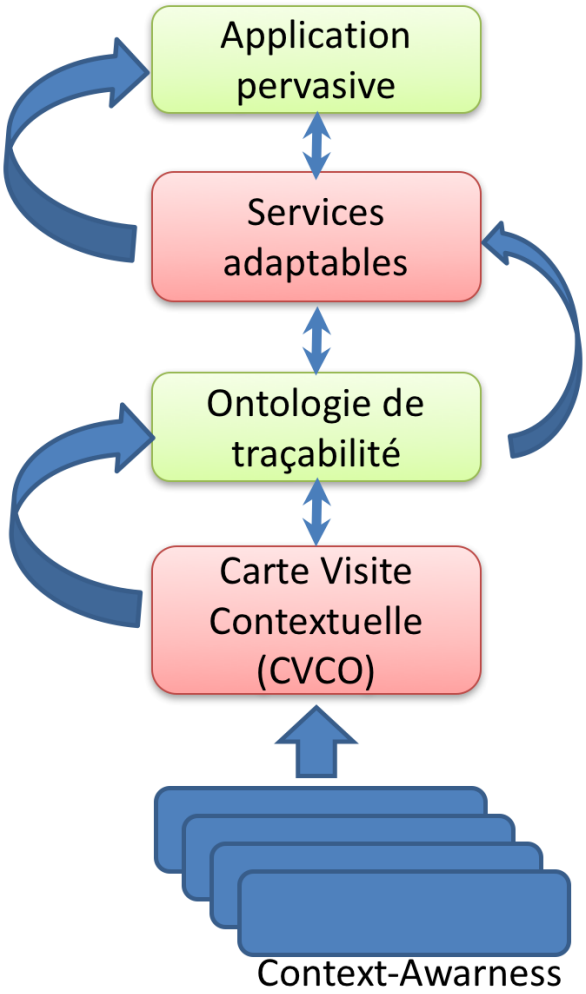

FIGURE 3.8 – Fonctionnement de la Carte de Visite COntextuelle

la traduction de la requête entre OWL et XML. Cette approche traite les éléments simples et les éléments complexes pour créer le schéma d'XML. Les *XML Schema* peuvent être modélisés à l'aide de différents styles, certains d'entre eux utilisent un seul élément global (élément racine), et d'autres utilisent plusieurs éléments globaux. Dans notre travail, nous prendrons en considération la façon de construire une ontologie OWL à partir de plusieurs sources de données XML.

Nous adoptons la cartographie indiquée dans la table 3.1 tout en suivant les règles de transformations XSD à OWL. En effet, la construction des entités OWL à travers le mapping des données XML est basée sur la cartographie de trois éléments de base comme :

- Les classes OWL : comme il est déjà indiqué dans la table précédente, la cartographie des classes OWL est faite au départ à travers les schémas XML(XSD) tout en respectant l'ensemble des règles de transformation. Pour la cartographie de type complexe par exemple, deux types sont distingués à ce niveau : un type complexe global et un autre local. Ces deux différents types sont cartographiés à l'aide des classes OWL. En effet, la classe générée à travers le type global prend le nom de ce type alors la classe générée à travers le type local prend le nom de l'élément dont il dépend. Les attributs sont également cartographiés à l'aide des

XSD	OWL
xsd:element (global) with complex type	owl:Class and subclass the class generated from the referenced complex type
xsd:element (global) with simple type	owl:Datatype
xsd:element (local to a type)	owl:DatatypeProperty or owl:ObjectProperty according to the element type.
xsd:attributeGroup	owl:Class
xsd:complexType with xs:simpleContent	owl:Class
xsd:complexType with xsd:complexContent	owl:Class
xsd:minOccurs, xsd:maxOccurs	owl:minCardinality, owl:maxCardinality
xsd:sequence, xsd:all	owl:intersectionOf
xsd:choice	combinaison of owl:intersectionOf, owl:unionOf and owl:complementOf

TABLE 3.1 – La cartographie XSD/OWL

classes OWL. Comme le schèma XML donne la possibilité d'exprimer la notion d'héritage, deux types d'héritage sont mis en places à ce niveau : l'extension et la restriction. Ces deux types sont transformés à l'aide de classes d'héritage OWL (en utilisant le rdfs :subclassOF).

- La propriété d'objet (*Object properties*) : les liens et les relations entre les éléments exprimés dans le schéma XML sont traduits comme étant des propriétés des objets dans l'ontologie à générer. Par exemple, lorsqu'un élément est d'un type complexe, ce type est déjà cartographié à l'aide d'une classe OWL. Par conséquent, une nouvelle propriété d'objet est ajoutée. Elle dépend de cette classe et a comme "domaine" la classe correspondant à ce type complexe et comme "rang" la classe correspondant au type de cet élément.
- La propriété de type de données(*Datatype properties*) : les éléments de type simple sont cartographiés à l'aide des propriétés de type de données. Lorsqu'un type complexe (global ou bien local) contient un élément de type simple qui a comme "domaine" la classe correspondant au type complexe. Dans ce cas, le type de cet élément est défini comme un type simple (xsd :integer, xsd :string), alors le "rang" de la propriété est le type lui-même.

LE MAPPING DE LA CVCO (XML) VERS UN MODÈLE DE TRAÇABILITÉ OWL :XSD2OWL

Chaque déclenchement d'une session utilisateur sur son application pervasive est considéré comme une situation courante d'utilisation de contexte. Chaque situation cou-

rante de contexte est considérée comme une instance des paramètres de ce contexte. Le changement des situations de contexte engendre automatiquement le changement des paramètres de celui-ci et donc le changement des services et des manières de présenter l'information sur les interfaces des utilisateurs. L'application sensible au contexte peut s'adapter au changement de toutes les situations d'utilisation tout en suivant et répondant aux traces de celles-ci.

Nous avons modélisé précédemment notre contexte d'utilisation en utilisant les ontologies de contexte. Ainsi, nous avons défini la classe principale nommée « traçabilité de contexte » (avec les sous-ensembles liés) par laquelle se réalise la correspondance et la cartographie des données récupérées à partir de la carte CVCO sous leur format XML à l'aide de celles en OWL. Nous appliquons le processus de mapping XML/OWL sur l'ensemble des données recupérées dans la carte CVCO afin de genérer les classes de l'ontologie de tracabilité. Cette classe de cartographie de la traçabilité d'une situation courate instantanée est placée dans l'ontologie de service afin de bien servir dans le processus d'adaptation des services en fonction des traces courantes d'utilisation de contexte : corrélation (traces/services).

Nous proposons à ce stade un processus de cartographie (mapping) qui permet de transformer et de faire la correspondance entre les éléments (traces récupérées dans la carte CVCO) sous leur forme XML et les ressources OWL de l'ontologie de traçabilité. Cette génération cartographique nous amène à spécifier des correspondances entre XML et OWL.

Le processus de génération de sources d'ontologies à travers des sources de données XML (récupéré à travers la carte CVCO) commence par la création du schéma XML correspond. Il doit être généré automatiquement à partir de la source du document XML (cf figure 3.9). Pour réaliser la transformation des données récupérées sous format XML des données XSD, qui présentent le schéma XML, nous utilisons l'API java « **Trang** »[1]. Cette API permet de prendre en entrée des données XML et de produire en sortie un schéma écrit en XML-Schema.

Nous utilisons également, dans un second temps, la bibliothèque Java XSOM (XML Schéma Object Model)[2] afin d'analyser le contenu XSD et de lier les informations entre elles. Ce qui correspond à ce que tous les éléments soient définis, puis référencés. La sortie de XSOM est utilisée ensuite comme une entrée pour réaliser la cartographie java XSD2OWL de traçabilité et pour décrire le schéma de l'ensemble des sommets qui contient tous les éléments, attributs, les types primitifs, les groupes d'éléments, les groupes d'attributs (XSshema), . . .

Suivant l'enchaînement de l'ensemble des étapes dans le processus de génération de l'ontologie de traçabilité, nous utilisons l'API Jena[3] pour les sorties de SXD2OWLMapping avec le serveur Apache Maven[4] pour la production de sémantique supplémentaire des données RDF.

L'étape suivante consiste à faire la correspondance des ressources OWL. En effet,

1. Produces as output a W3C *XML Schema* : http ://www.thaiopensource.com/relaxng/trang.html
2. *XML Schema* Object Model (XSOM) is a Java library that allows to parse *XML Schema* documents : https ://xsom.dev.java.net
3. Work with models, RDFS and the Web Ontology Language (OWL) to add extra semantics to your RDF data : https ://jena.apache.org/download/index.cgi
4. Apache Maven is a software project management and comprehension tool : http ://maven.apache.org/download.cgi

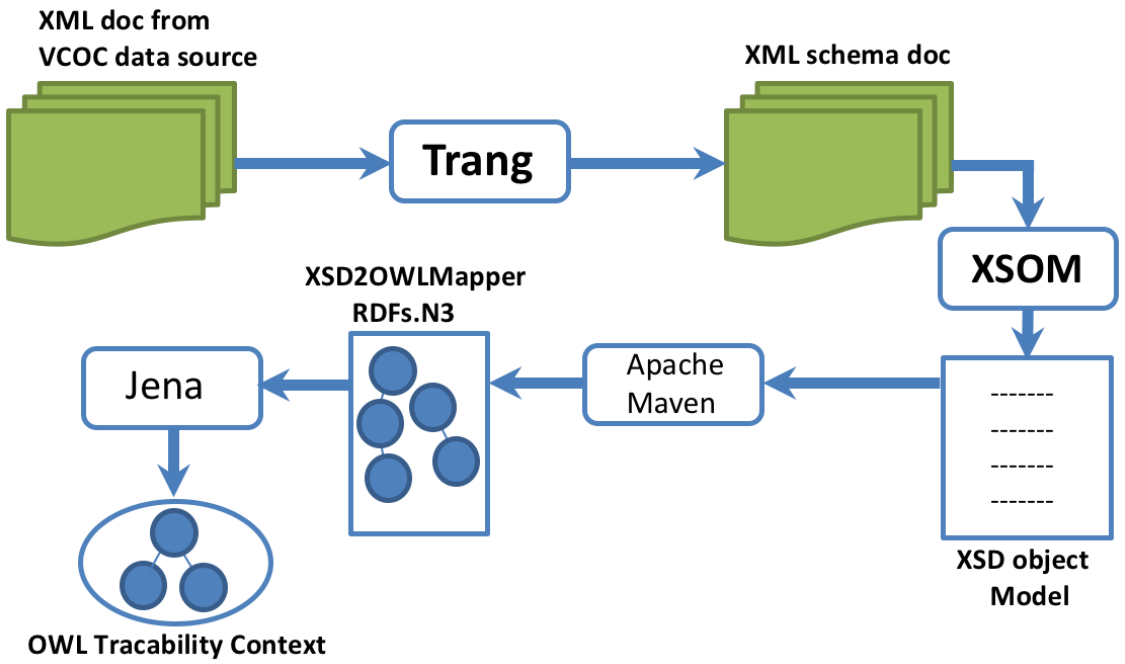

FIGURE 3.9 – Le processus de génération de l'ontologie de traçabilité (OWL) à l'aide de la CVCO(XML)

comme notre ontologie de traçabilité est principalement définie par les classes et les sous-classes, les propriétés des types de données et des objets, nous suivons la spécification des correspondances pour générer l'ontologie OWL de la traçabilité de contexte, une cartographie en fonction de trois types de correspondance (cf table 3.2) :

- correspondance de classe : pour faire correspondre les concepts OWL ;
- Correspondance de la propriété de type de données (*Datatype property*) : pour faire correspondre les propriétés de types de données OWL (Datatype property) ;
- Correspondance de la propriété des objets (*Object property*) : pour relier les classes cartographiées via une propriété d'objet OWL.

Cartographies	Notation
Class	(OWL Class URI, XPath expression, OWL Class URI, XPath expression, ID XPath expression)
Datatype Property	(OWL Datatype Property URI, Domain Class Mapping, Xpath Expression)
Object Property	(OWL Object Property URI, Domain Class Mapping, Range Class Mapping)

TABLE 3.2 – Les notations sous-jacentes de la cartographie OWL

Pour le fonctionnement, les ressources OWL (les classes, les propriétés des objets et des types de données) sont référencées par leurs références URI. Les expressions XPath [Clar99] sont utilisées pour désigner parmi les classes celles qui sont cartographiées. En effet, l'utilisation des expressions XPath permet la distinction entre les classes qui ont les mêmes noms mais avec des sous-nœuds différents. Elle permet donc de fournir la correspondance et de les associer à leur concept OWL correspondant. En se référant à la table 3.2, nous constatons que les classes cartographiées ont été définies à travers l'utilisation de paires (URI Référence, XPath Expression). Cette paire permet d'informer qu'une nouvelle instance de la classe OWL est créée pour chaque nœud XML et qu'elle est identifiée par son *XPath Expression*. En se référant toujours aux notations de la table 3.2, il est également possible de créer des correspondances de classe avec des triplets à l'aide de l'utilisation des ID.

L'INTÉGRATION DE MODÈLE DE TRAÇABILITÉ OWL DANS LE MODÈLE DE BASE DE CONTEXTE

Lors de l'étude des différents travaux de recherche évoqués dans le domaine de la sensibilité au contexte présentée précédemment, nous constatons que l'utilisation des modèles de contexte à base d'ontologies permet la corrélation : utilisateur/servicesPertinents pour l'adaptation de contexte d'utilisation. Notre contribution spécifie cette corrélation tout en proposant un mécanisme qui permet de fournir la traçabilité de contexte qui permet d'adapter les services en fonction des traces de contexte récupérées instantanément.

Nous proposons, pour ce faire, d'intégrer notre modèle de traçabilité de contexte dans le modèle général de contexte. Notre ontologie de service a adopté la classe de traçabilité contextuelle (avec les sous-classes et les propriétés correspondantes). Cette classe est alimentée par les données récupérées instantanément via la carte de visite contextuelle. Elle permet de fournir les traces de toutes les situations contextuelles courantes,

tout en prenant en compte les changements dynamiques de celles-ci. Ces paramètres présentent les instances des caractéristiques des situations courantes. L'avantage de l'intégration de mécanisme de traçabilité est la fourniture en amont des paramètres et des caractéristiques de la situation contextuelle courante. Ainsi, comme les éléments de contexte, dans notre modèle général, sont complémentaires les uns aux autres, la réaction d'adaptation de l'ensemble des services est faite suite à une communication entre ces différents éléments (cf figure3.10).

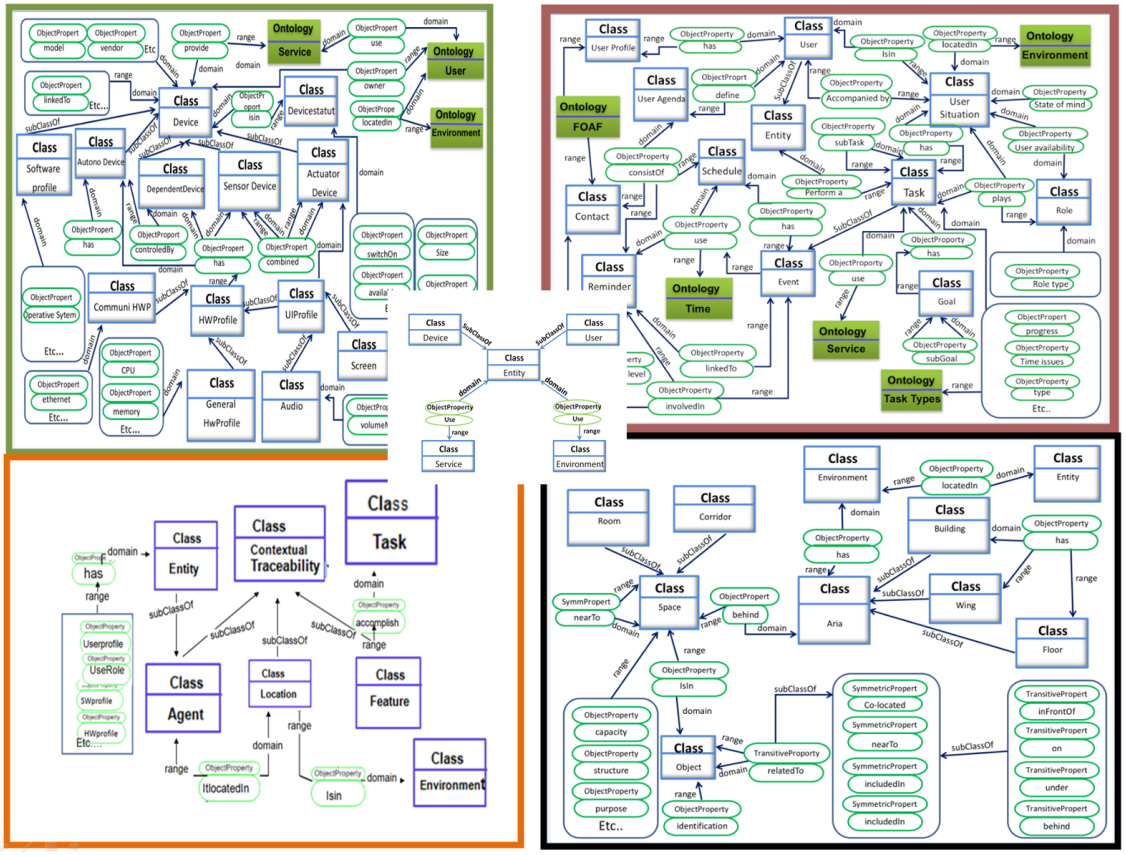

FIGURE 3.10 – Modèle générale de contexte à base d'ontologie de traçabilité

3.2.4/ LES COMPOSANTS DE DIFFÉRENTES ONTOLOGIES DE NOTRE DOMAINE

Dans le domaine de sensibilité au contexte, les éléments du monde réel sont représentés généralement par des descriptions qui peuvent être des concepts, des rôles, ... La modélisation des connaissances dans ce domaine avec les logiques de description (LD) permet de décrire les individus de ce domaine. Cette modélisation se réalise via deux niveaux qui permettent de séparer la représentation des connaissances intentionnelles (y compris l'ensemble des intensions) de celle de l'extensionnelle (y compris l'ensemble des extensions).

En effet, les connaissances générales de domaine de sensibilité au contexte sont décrites dans un premier niveau générique conceptuel terminologique (la *TBox*), puis dans un niveau factuel, ces connaissances seront représentées selon une configuration précise

sous forme des assertions (la *Abox*). Avec les logiques descriptives, la Tbox est située dans le niveau conceptuel, celui de l'introduction des différentes primitives conceptuelles. Ce niveau permet de décrire les connaissances générales tout en mettant l'accent sur l'ensemble des déclarations liées aux différentes primitives. Ces déclarations décrivent les propriétés et les relations entre concepts, ce qui permet la définition intentionnelle des connaissances de domaine. Dans le niveau factuel, les connaissances factuelles de domaine seront décrites tout en respectant une configuration précise donnée par les assertions ABox. Une ABox contient l'ensemble des déclarations des instances de concepts qui ont été déjà définies dans la TBox [Baad03]. Comme nous l'avons abordé, les concepts sont considérés comme étant les points de départ et l'objectif pour toutes ontologies, ces concepts peuvent être divisés en deux catégories :

- les concepts primitifs : ils sont les concepts insérés dans l'arborescence de l'ontologie qui ont des conditions nécessaires pour la relation de membre de classe. Notons ici l'exemple suivant : nous affirmons dans l'ontologie utilisateur que le médecin est une personne, alors nous pouvons affirmer que la relation inverse n'existe pas.
- les concepts définis : un concept est dit défini, si la connaissance construite correspond à un concept dans le monde modélisé. Ces concepts possèdent généralement une description nécessaire et suffisante pour qu'un terme soit un membre de la classe mère. Par exemple si l'on définit un agent dans notre ontologie comme suit :
  - Agent *est-Un* Object ou Entité : tout concept qui remplit la condition après la relation binaire *a-Un* implique qu'il soit membre du concept «Agent». Une ontologie dans le domaine de la sensibilisé au contexte est non seulement le repérage et la classification des concepts mais aussi les relations qui sont liées entre ces différents concepts. Deux types de relations peuvent être décrit à ce niveau : les relations de subsomption qui définissent un lien de généralisation «**is-a**» ou «**is-a–kind-of**» (une entité is-a-kind-of agent). Cette relation permet formellement l'héritage à travers les liens de spécification et de généralisation. Pour les autres types de relations, nous prenons les relations associatives, elles permettent non seulement de faire la liaison entre les concepts mais aussi de représenter les propriétés susceptibles d'être attribuées à ces différents concepts : comme par exemple *DispositifAutonome* caracterisé-Par *HWProfile*. Dans cette même classe des relations associatives, nous trouvons les relations nominatives qui définissent les noms des concepts (nous prenons l'exemple : a-Un-Nom qui permet de définir le nom de la spécialité du médecin :
    - SpécialitéMédecin *a-Un-Nom* NeuroChirurgien).
- Finissons par les relations de localisation qui permettent de définir généralement la position des concepts les uns par rapport aux autres.

3.2.5/ Formalisme à base de Logiques de Description pour la représentation des connaissances

Les ontologies fournissent un vocabulaire commun de domaine et définissent la signification des termes et des relations entre elles. Les connaissances dans les ontologies sont principalement formalisées tout en utilisant les différents composants : concepts (classes), relations (propriétés), fonctions, axiomes (règles) et instances (individus)(application des taches numéro : 9, 10, 11 du processus de développement d'ontologie).

Ces connaissances doivent êtres exprimées à l'aide d'un langage formel de description des connaissances dans un premier temps, pour savoir les manipuler, dans un second temps à l'aide de mécanismes bien définis opérant sur les différents éléments de la représentation. La syntaxe de logiques de description (LD) peut être considérée comme étant une famille de langages de représentation de connaissances qui peuvent être utilisées pour représenter la connaissance terminologique d'un domaine d'application d'une manière formelle et structurée. Une sémantique de domaine sémantique peut être fournie à travers une transcription en logique des prédicats du premier ordre.

Expression	Nom	Syntaxe	Interprète	Description
⊑	SubSomption de concepts	C ⊆D	Ttes instances de C dans D	« Inclusion »
≡	Concept d'équivalence	C ≡ D	C est équivalent à D	« Egale »
⊤	Concept universel	⊤	Concept plus générale	« Tout »
⊥	Concept vide	⊥	Concept moins général	« Rien »
⊓	Intersection des concepts	C⊓D	C et D	« Et »
∃	Restriction existentielle	∃R.C	R-successeur existe dans C	« ILExiste »
∀	Restriction universelle	∀R.C	Tous R-success sont dans C	« Tous »
¬	Négation	¬C	Négation ou complnt des concepts	« Non »
⊔	Union	C⊔D	union ou disjtion des concepts	« Ou »

TABLE 3.3 – Règles de syntaxe pour la logique descriptive

Pour le traitement de la sémantique dans un domaine, les logiques de description utilisent les notions de concept, de rôle et d'individu. Or, un concept est défini comme étant une classe d'éléments et est interprété comme un ensemble. Les rôles correspondent aux liens entre les éléments et sont interprétés comme des relations binaires sur un univers donné. Les individus correspondent aux éléments dans le domaine. De ce fait, les logiques de description permettent d'extraire des connaissances implicites de la connaissance explicite à travers des relations de subsomption dans une hiérarchie des concepts et des rôles afin de construire la base de connaissance. Pour ce faire, la logique descriptive sera appliquée sur l'ensemble des concepts (nous prenons l'exemple de deux concepts C et D) selon le rôle R. Nous donnons le tableau qui présente les règles de syntaxe de base de logique descriptive LD (cf table 3.3)

Nous pouvons définir les règles de base pour la logique descriptive comme suit :

$CON = \{ C_1, C_2 \}$ un ensemble fini de concepts atomiques ;

$ROL = \{ R_1, R_2 \}$ un ensemble fini de rôles atomiques et ;
$IND = \{ a_1, a_2 \}$ un ensemble fini d'individus.

Pour les concepts atomiques, ils définissent un ensemble d'individus, comme par exemple :

- Les concepts atomiques primitifs : terminal, environnement, . . . ;
- Les concepts atomiques non préemptifs : disposotifAutonone, Luminosité, . . . ;

Les rôles permettent de dénoter les ensembles d'individus, et se distinguent en deux niveaux :

- Les rôles atomiques : partieDe, HWPrifileDe . . . ;
- les rôles complexes : AgentHWPrifileDe ⊓ DispositifAutonomeHWPrifileDe . . .

Les concepts et les rôles atomiques peuvent être combinés au moyen de constructeurs pour former respectivement des concepts et des rôles composés. Si nous prenons l'exemple qui suit :

Le concept composé *Terminal* ⊓ *DispositifAutonome* résulte de l'application du constructeur sur les concepts atomiques *Terminal* et *DispositifAutonome*.

Le concept *Terminal* ⊓ *DispositifAutonome* s'interprète comme l'ensemble des individus qui appartiennent aux concepts *Terminal* ⊓ *DispositifAutonome* . Les différentes LD se distinguent par les constructeurs qu'elles proposent. Plus les LD sont expressives, plus la probabilité des contradictions au sein de l'ontologie (la vérification de cohérence) est réduite.

Nous prenons un autre exemple descriptif, si nous considérons que la définition du concept *DispositifAutonome* est un terminal qui a un ensemble spécifique de caractéristiques de *HWProfile*, selon les règles de logique descriptive définies dans la table 3.3, alors la syntaxe LD de la présente déclaration est la suivante :

$$DispositifAutonome \equiv Terminal \sqcap \exists AUnHWProfile.T$$

Prenons un autre exemple, si nous déterminons que chaque médecin a nécessairement sa spécialité, la syntaxe de LD pour cette déclaration devient :

$$Médecin \equiv ASpecialité.T.$$

A coté d'une spécification explicite des membres d'un concept, ces membres peuvent également être définis selon les axiomes d'une manière plus générale. En effet, dans ces axiomes, il faut préciser intialement l'ensemble des conditions nécessaires ou suffisantes pour que ces individus soient des membres d'un concept.

$$Terminal \equiv TechniqueAgent \sqcap Entité$$

$$DispositifAutonome \sqsubseteq TechniqueAgent$$

$$DispositifAutonome(A) \equiv Terminal \sqcap \exists\ a\text{-}Un\text{-}HWProfile(A.B)$$

$$HWProfile(B))$$

$$HWProfile \sqsubseteq TechniqueAgent$$

Le concept *Terminal* est défini, dans cette déclaration à base de la syntaxe DL, comme étant tous les appareils techniques (les agents techniques) et toutes les entités (les entités humaines) susceptibles d'être présents dans une application interactive sensible au contexte. Cette déclaration affirme que le *DispositifAutonome* est une sous classe du concept principal*DispositifAutonome*. Nous ajoutons une règle pour définir l'ensemble de couples d'individus comme suit :
$\exists$ *a-Un-HWProfile(A.B)*, dans laquelle une restriction existentielle a été appliquée. En se référant à la table 3.3, il existe d'autres types de restriction des propriétés. Le quantificateur de restriction universelle, par exemple, permet de lier toutes les valeurs d'une propriété introduite dans la restriction de classe. L'idée d'utilisation d'axiomes pour la définition des classes ainsi que des propriétés permet de vérifier toutes les cohérences dans l'ontologie. Compte tenu de l'expressivité riche et de la nature combinatoire de OWL-DL (à travers les services de raisonnement offerts par *OWL reasoners*), un ensemble de nouveaux concepts sera créé et extrait à travers une combinaison de concepts existants qui seront automatiquement placés dans la hiérarchie de l'ontologie.

Le domaine de la sensibilité au contexte est riche au niveau de la variété des paramètres et des informations liées aux situations contextuelles qui permettent de décrire le contexte des utilisateurs face aux systèmes sensibles au contexte. De ce fait, la création d'une base de connaissance de contexte d'utilisation des applications interactives nécessite des efforts au niveau du traitement de la terminologie en premier lieu *Terminological box*, par l'axiomatisation des propriétés et des relations ainsi que par les déclarations qui permettent de décrire la structure de domaine en cours. En deuxième lieu, les efforts portent sur le traitement des assertions et les déclarations des éléments et des objets *Assertion box*. Pratiquement, la *Terminological box* est constituée du modèle de base de déclarations et des axiomes des concepts de la base de connaissance qui seront utilisés pour les assertions Box et les inclusions entre concepts dans cette même base.

A ce stade, la base de connaissance dans notre domaine de recherche peut être définie comme suit :

Une base de connaissances de contexte **KC** est une paire **T,A**$_i$, où :

T est un ensemble d'axiomes Terminologiques (TBox) ;
A est un ensemble d'axiomes assertionnels (ABox) ;
Les axiomes de TBox sont sous la forme :

$$A,\ C \sqsubseteq D,\ C \equiv D,\ R \sqsubseteq S,\ R \equiv S \text{ and } R+ \sqsubseteq R, \text{ où :}$$

A est un concept atomique ;
C, D sont des concepts ;
R, S sont des rôles et ;

R+ est l'ensemble des rôles transitifs.

Les composantes de base de Tbox et Abox présentent l'ensemble d'opérateurs de LD qui ont été déjà décrits dans la table 3.3. Nous finissons avec la déclaration suivante pour montrer l'utilisation de la logique de description comme un outil pour exprimer les concepts complexes ainsi que les rôles utilisés dans la Tbox et l'Abox dans notre base de connaissance :

$$\textit{Utilisateur} \sqcap \exists\ \textit{a-Un-Role.T};$$
$$\textit{Utilisateur} \sqcap \exists\ \textit{a-Un-Role.T} \sqcap \forall\ \textit{a-Un-Role.Medecin}$$

3.3/ DÉVELOPPEMENT ET IMPLÉMENTATION DE NOTRE MODÈLE À BASE DE L'ONTOLOGIE DE TRAÇABILITÉ

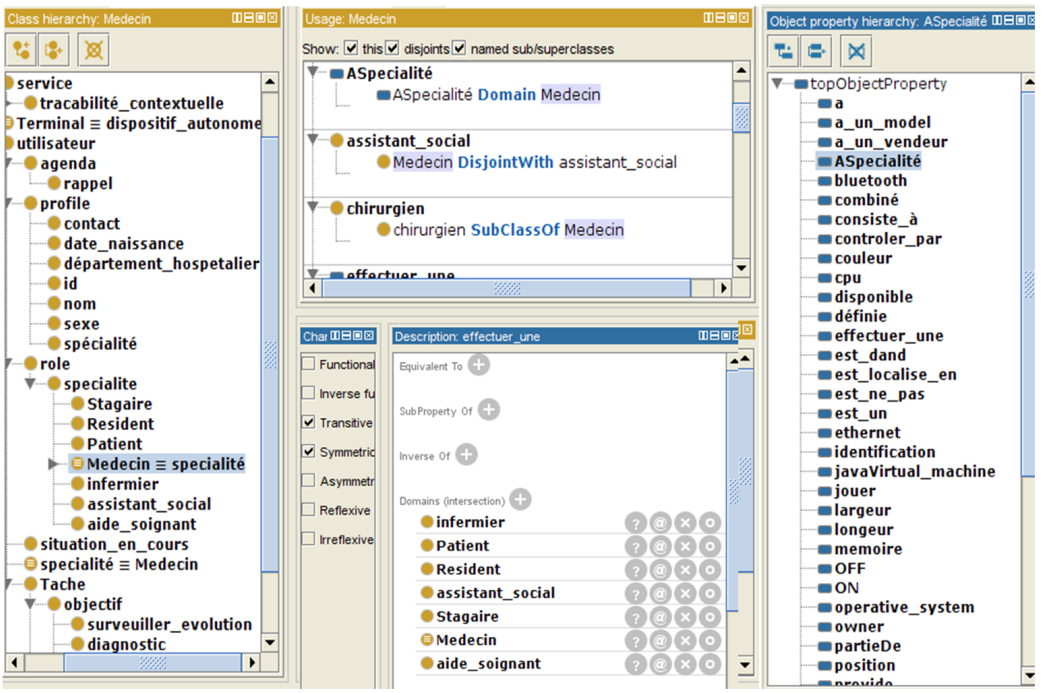

FIGURE 3.11 – Implémentation dans Protégé de l'ontologie de contexte

Dans le domaine de la sensibilité au contexte, le développement des ontologies nécessite un effort de conceptualisation ainsi qu'un long processus d'édition par le biais des éditeurs d'ontologies tels que **Protégé4.3**, qui possèdent des terminologies standards. Notre ontologie est développée sous Protégé (cf figure 3.11). Elle décrit de manière générique les données décrivant le contexte utilisateur en face de son environnement. Par conséquent, notre choix s'appuye sur les ressources existantes pour la description du contexte utilisateur (modèles de la littérature de description et de gestion des dimensions de contexte à base d'ontologies : PIVon, Coiva, OCAAR).

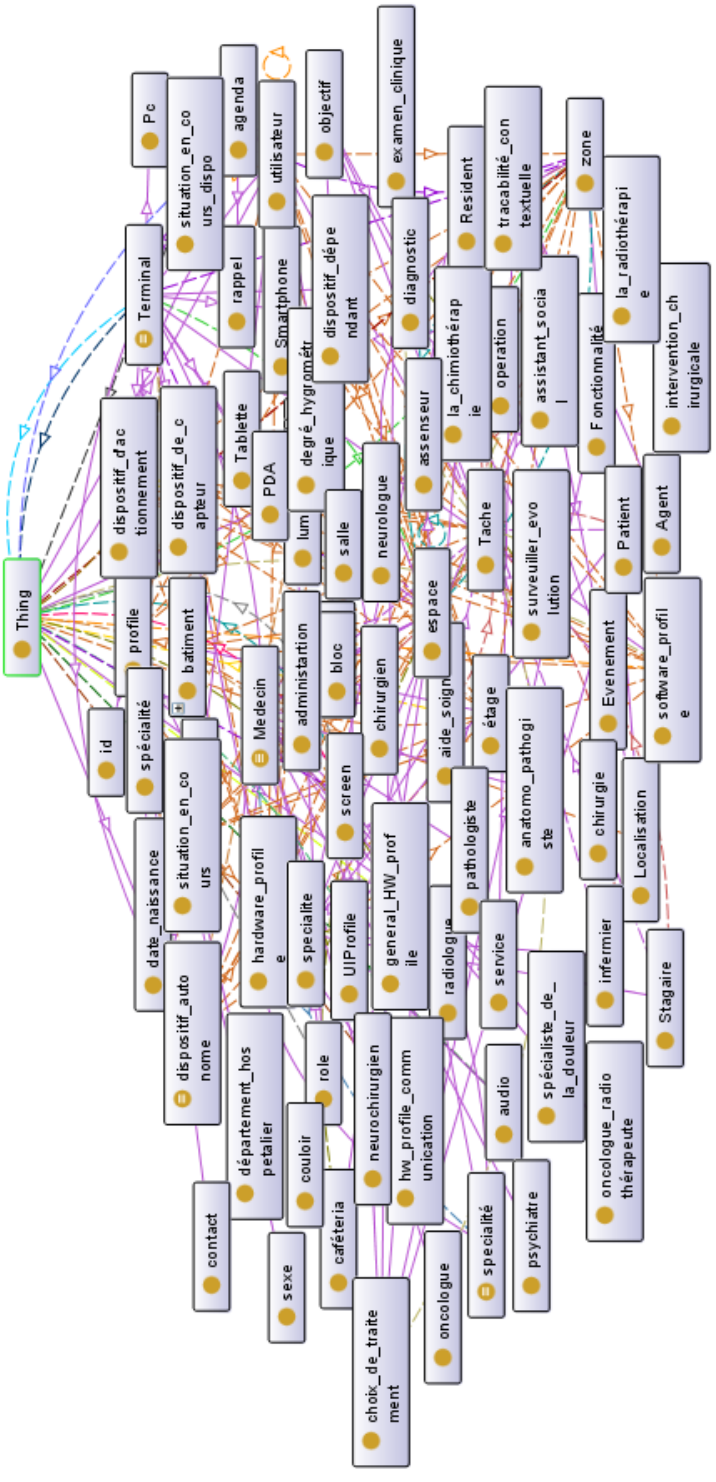

FIGURE 3.12 – Ontologie globale de contexte

Nous avons choisi d'appliquer nos ontologie dans le domaine de la chirurgie des tumeurs du cerveau allant de la phase d'observation et d'analyse jusqu'à la phase d'évaluation de cycle d'évolution de ce tumeur. Nous utilisons pour cette tâche(la tâche numéro 1 de processus de développement de notre ontologie), des ressources documentaires archivées

sur le portail officiel de l'Instit National du Cancer de Paris (INCa) comme étant l'agence d'expertise sanitaire et scientifique en cancérologie de l'État chargée de coordonner les actions de lutte contre le cancer. Le corpus INCa est un regroupement sémantique des concepts implicites. Les concepts utilisés depuis ce thésaurus ne sont pas explicitement exprimées, mais il y a des descriptions et des définitions implicitement liées qui sur lesquelles nous avons travaillons (tache numéro 2, 3 et 4 du processus de développement de notre ontologie). Ce portail est très riche au niveau de la fourniture des informations dans le domaine en termes de concepts et de relations. Notre ontologie a été créée en grande partie avec les concepts du modèles de R. Hervas en choisissant en plus des concepts sur le domaine de la chirurgie des tumeurs du cerveau (cf figure 3.11) .

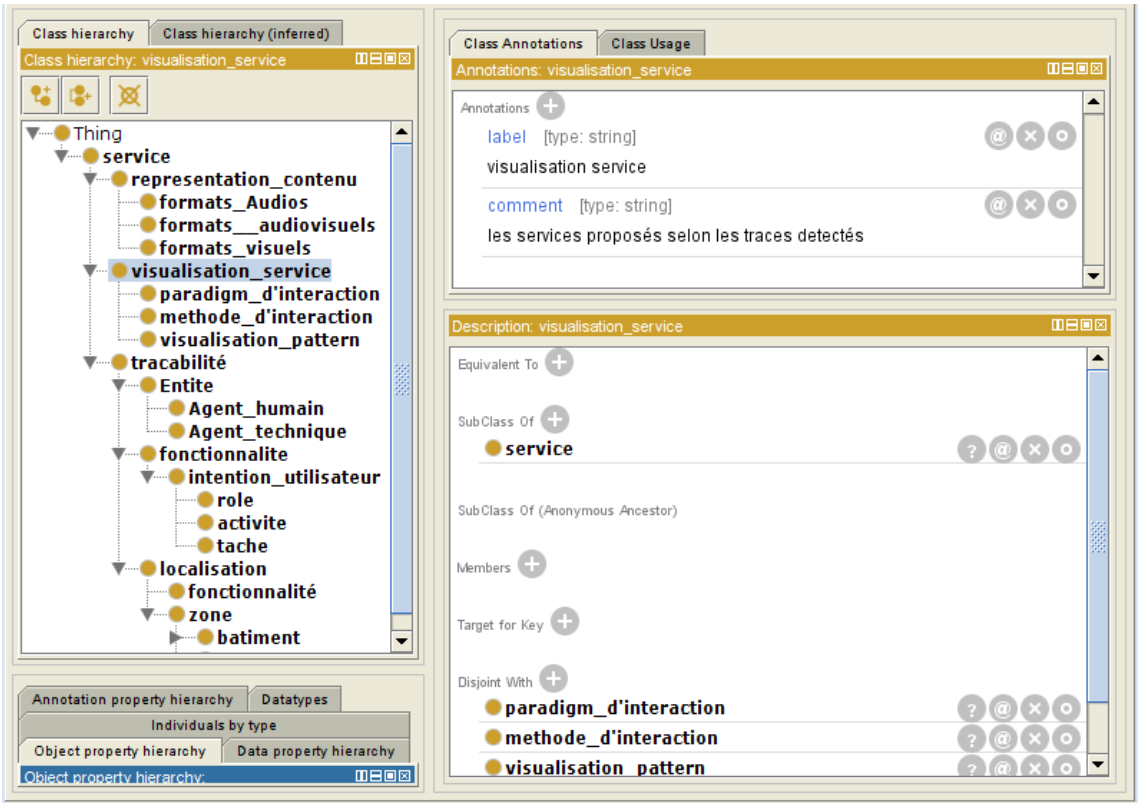

FIGURE 3.13 – Implémentation dans Protégé de l'ontologie de traçabilité

En effet, cette ontologie a été créée pour la modélisation des connaissances et le raisonnement en s'appuyant entre temps sur l'analyse du corpus INCa. L'ontologie complète est présentée sur la figure 3.12. Cette ontologie en réalité se compose de multiples ontologies : une ontologie de terminal, une ontologie d'environnement, et une ontologie utilisateur. Ces ontologies établissent de manière non-ambigüe la sémantique entre les éléments de contexte en réagissant avec leur environnement courant. En effet, comme nous avons dit précédemment, dans notre développement d'ontologies, nous avons utilisé Le modèle de connaissances sous-jacent à **Protégé 4.3** qu'il est issu du modèle des frames basé sur les classes et les propriétés. Nous avons appliqué par-là les taches numéro 5, 6 , 7 et 8 du processus de développement d'ontologie.

Une plateforme d'adaptation au contexte efficace et pertinente doit intégrer des mécanismes autonomes qui permettent de réagir en fonctions des situations courantes

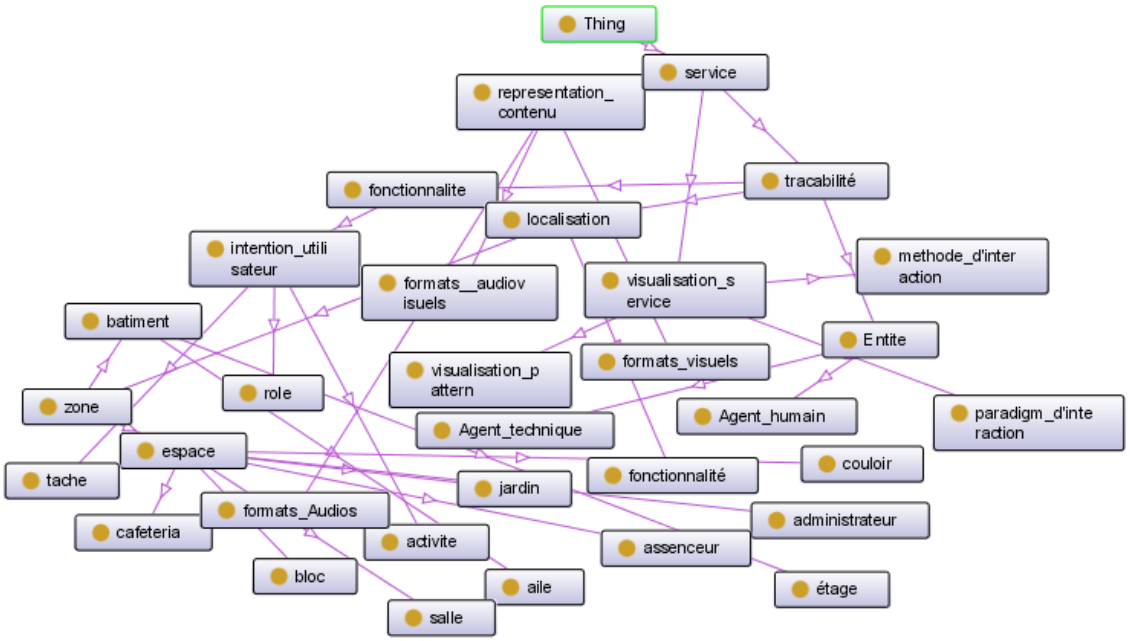

FIGURE 3.14 – Ontologie de traçabilité

des utilisateurs pour adapter leurs applications pervasives. Ce mécanisme de traçabilité des paramètres de contexte des différentes situations courantes est mise en place afin d'informer le système pervasif sur l'évolution d'utilisateur dans son contexte. L'intégration de ce mécanisme comme source d'information sur le contexte dans l'ontologie de contexte (ontologie de service) permet d'accéder facilement à l'information et d'aider la prise à de décisions par les systèmes pervasifs. En effet, l'ontologie de service qui est dotée de la traçabilité est également implantée en Protégé (cf figure 3.13). Elle permet de fournir des connaissances pertinentes sur le contexte instantané afin d'aider dans la phase d'adaptation. Une ontologie de traçabilité de contexte a l'avantage de combiner les informations liées au contexte courant ainsi qu'aux services à offrir aux utilisateurs. Elle fournit une thermologie pour les concepts de services en fonctions des traces de contexte.

L'ontologie complète de traçabilité de contexte (cf figure 3.14) correspond à la modélisation de service en fonction des traces d'une situation contextuelle courante.

Synthèse du chapitre

Nous avons présenté, dans ce présent chapitre, la démarche que nous avons suivie pour le développement de nos ontologies dans le domaine de la sensibilité au contexte. En effet, nous avons adopté un mécanisme appelé CVCO (notre *Carte de Visite COntextuelle*) qui permet de fournir la traçabilité (les traces de contexte) des situations contextuelles courantes instantanées et dynamiques dans les environnements pervasifs. L'intégration de ce mécanisme permet de nous aider dans le développement de notre ontologie de traçabilité de contexte (pour adapter les services dans les environnants pervasifs).
Nous avons développé, de ce fait, deux ontologies : une ontologie des éléments de contexte et une autre pour fournir la traçabilité des adaptations. Ces ontologies nous permettent d'effectuer la caractérisation sémantique des informations de contexte pour aider à l'adaptation. L'ontologie de contexte et l'ontologie de traçabilité ont pour but l'indexation des connaissances à travers un langage de spécification pour la sensibilisation contextuelle. Nous intégrons par la suite, nos ontologies dans notre nouvelle plateforme d'adaptation au contexte.

Le chapitre suivant est consacré à la définition de notre nouvelle plateforme d'adaptation au contexte qui se nomme *COALA : COntext Adaptation pLAtform*. En effet, COALA est une plateforme dédiée à la sensibilisé des applications interactives pervasives tout en les rendant adaptables au contexte d'utilisation. Cette nouvelle plateforme est basée sur trois couches principales : La première couche permet d'assurer la gestion de contexte à travers une modélisation rigoureuse de ses différents éléments. C'est une couche plutôt attachée au niveau utilisateur puisque l'objectif à ce niveau est de récupérer toutes les informations liées à une situation contextuelle courante. dans cette couche l'objectif est donc de regrouper les informations matérielles et logicielles relatives aux nœuds qui vont accueillir l'application à déployer via le niveau utilisateur qui permet de fournir les informations sur l'utilisateur de l'application comme ses préférences, sa localisation et son profil. La deuxième couche consiste du niveau application de la plateforme. Cette couche permet entre autres l'interprétation et l'intégration afin d'assurer l'adaptation du comportement de l'application aux traces de contexte récupéré. La troisième couche permet de proposer des nouveaux services adaptés aux applications de la plateforme.

4

PLATEFORME COALA : DÉFINITIONS ET EXPÉRIMENTATIONS

INTRODUCTION

Récemment, la prise en compte du contexte est un aspect essentiel dans l'informatique orientée service dans les environnements ambiants. Cette prise en compte concerne principalement les différents paramètres qui décrivent le contexte utilisateur à un moment donné (la localisation, les préférences de l'utilisateur, ...). Lors de changements significatifs de situations en cours d'exécution, la prise en compte du contexte peut déclencher une suite de réactions induites par différents changements. Dans le domaine de la sensibilité au contexte, l'architecture sensible au contexte (*context-aware*) permet d'exprimer la prise en compte du contexte et l'adaptation à travers deux couches principales : la couche de modélisation et de gestion de contexte et la couche d'adaptation de comportement de l'application au contexte.

Notre objectif est de proposer une plateforme qui permette l'adaptation de contexte tout en accordant la même importance aux couches de l'architecture de notre plateforme. Le recours à l'intégration d'un modèle de traçabilité de contexte à base d'ontologie est essentiel pour répondre aux besoins des utilisateurs au niveau dynamisme et personnalisation des services offerts. Dans ce chapitre, nous décrirons les différentes couches (modules) constituant notre nouvelle plateforme d'adaptation au contexte COALA (*COntext Adaptation Platform*), tout en détaillant dans chaque niveau les différents modules ainsi que les outils qui permettent de garantir l'adaptation au contexte. Nous décrirons également dans ce chapitre le principe de fonctionnement de COALA pour finir en exposant des exemples des scénarios expérimentaux.

4.1/ COALA : UNE NOUVELLE PLATEFORME D'ADAPTATION DE LA SENSIBILITÉ CONTEXTUELLE

COALA est une plateforme dédiée à l'adaptation au contexte des applications interactives pervasives. Cet environnement met en œuvre les avantages de l'ingénierie ontologique, du web semantique et du mapping OWL.

4.1.1/ PRÉSENTATION DE NOTRE NOUVELLE PLATEFORME D'ADAPTATION AU CONTEXTE

Dans le cadre de nos recherches, notre objectif est de proposer COALA qui permet d'offrir des services adaptables aux utilisateurs. En effet, COALA est une plateforme dédiée à la sensibilisé des applications interactives pervasives tout en les rendent adaptables au contexte d'utilisation. Dans un contexte défini, l'utilisateur se connecte à l'application interactive en envoyant automatiquement les informations et les propriétés de contexte qui l'entourent. L'enchaînement et la liaison entre les composants dans COALA sont définis ainsi (cf figure 4.1) :

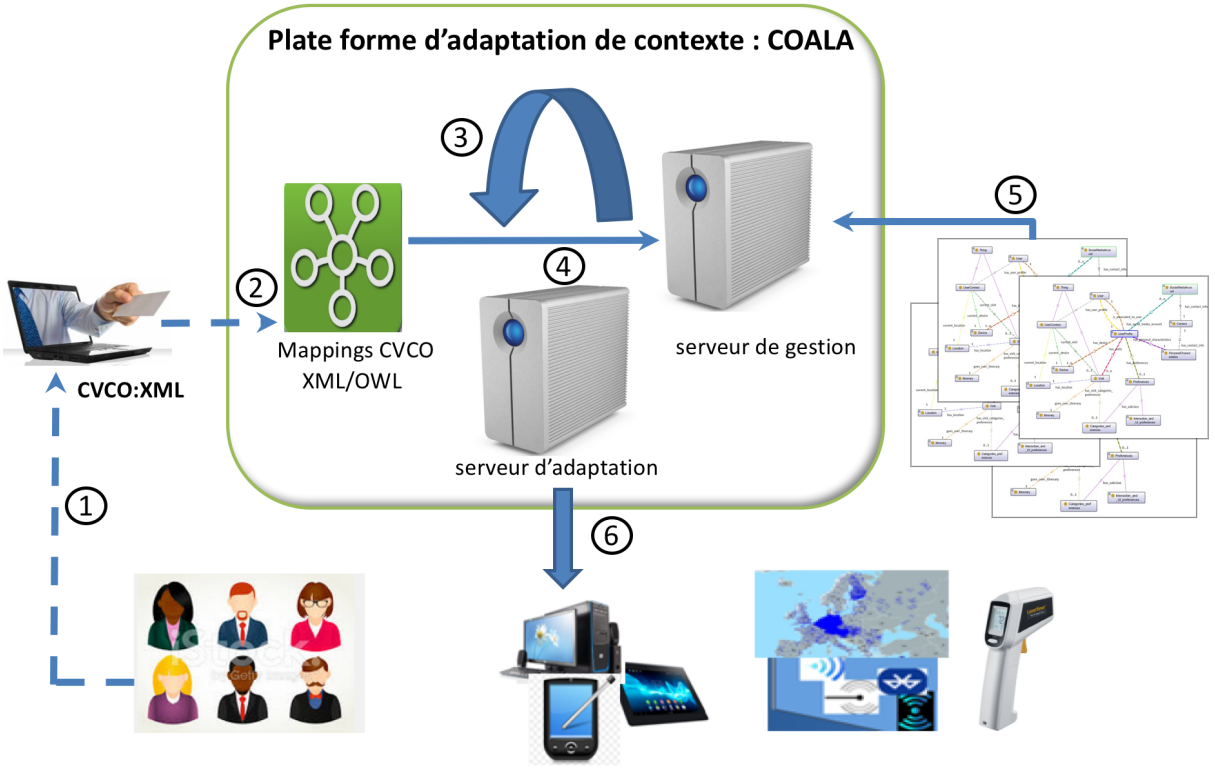

FIGURE 4.1 – Présentation de la plateforme COALA

❶ La génération d'une carte de visite contextuelle à partir les éléments de contexte,
❷ La cartographie XML/OWL des informations récupérées des situations courantes de contexte,
❸ L'envoi des cartographies au serveur de gestion pour l'intégration à l'ontologie de traçabilité,
❹ La communication entre l'ontologie et le serveur pour le choix des méthodes d'adaptation et la préparation des données de services,
❺ L'envoi des décisions aux serveurs d'adaptation et la détermination des services adaptables,
❻ La mise à jour et l'affichage des services sur les interfaces interactives selon les désirs et les besoins des utilisateurs.

Cet environnent combine les avantages de la représentation des connaissances et de la traçabilité de sensibilité de contexte. Il permet à tout utilisateur (staff hospitalier par

exemple) l'accès au contenu de l'application, la traçabilité, la prise de décision, ... L'objectif de cette plateforme est d'offrir la corrélation entre les informations générales de contexte à travers la création de connaissances permettant l'adaptation des besoins des utilisateurs aux systèmes pervasifs.

Les connaissances créées durant la connexion des utilisateurs à leur session, les décisions au niveau des services adaptables doivent être prises pour chaque utilisateur selon sa situation contextuelle en courante. COALA interprète la requête de l'utilisateur et affiche le service demandé adaptable avec un haut degré de personnalisation.

4.1.2/ Stratégies d'adaptation de services au contexte dans COALA

Dans COALA, nous avons défini l'ensemble des stratégies d'adaptation fonctionnelle d'une application interactive sensible au contexte selon son fonctionnement. En effet, ces stratégies sont basées sur la définition de l'ensemble des relations permettant d'associer une situation contextuelle en cours à l'ensemble des actions d'adaptation de services. Une situation contextuelle est définie par un ensemble de paramètres de contexte courant récupérés. Ces paramètres seront utilisés pour les actions d'adaptation afin de fournir des services adéquats et adaptables. Les actions d'adaptation sont exécutées par le gestionnaire d'adaptation de services dans COALA. Les opérations d'adaptation permettent de garantir l'ensemble des modifications sur la liste des services qui répond aux paramètres de contexte récupérés.

À ce stade nous pouvons définir deux types d'adaptation de services dans COALA à savoir : l'adaptation de contenu et l'adaptation de présentation de service.

L'adaptation de contenu

Le gestionnaire d'adaptation de contenu des services dans COALA consiste à fournir l'ensemble des données multimédia (images, vidéos, son, texte, ...) selon le contexte de l'utilisateur. Ces données doivent être exploitables selon toutes les exigences des situations contextuelles particulières.

Par exemple, un médecin se trouve dans un bloc opératoire, il est doté d'un terminal de type smartphone, et il a pas l'opportunité d'utiliser des services comme la manipulation vidéo. Dans ce cas, le médecin invoque, au niveau du service fourni, une donnée nécessaire à l'adaptation de format. En effet, plusieurs autres types d'adaptation peuvent être indispensables pour fournir des données directement exploitables par les utilisateurs d'applications interactives pervasives.

À ce niveau, nous définissons l'adaptation de contenu des services aux situations contextuelles courantes d'une application pervasive par l'ensemble des opérations indispensables qui permettent de fournir un contenu de données répondant exactement aux besoins des applications dans un contexte d'utilisation précis. Le but de l'adaptation dans COALA est de fournir la meilleure information possible à l'utilisateur en fonction de l'exigence de sa situation contextuelle. La réussite et l'efficacité de l'adaptation de contenu dépendent fortement de la qualité des connaissances générées et récupérées sur le contexte : les connaissances dans nos ontologies de contexte. Pour la tâche d'adaptation de contenu dans COALA, nous avons conçu et développé un module d'adaptation fondé sur les notions d'adaptation de données rencontrées dans [Berhe05]). Les auteurs

ont fondé leur travail sur un moteur qui permet d'assurer la sélection de l'ensemble des opérations d'adaptation de contenu afin de fournir des données adaptées au contexte d'utilisation des situations courantes. En effet, pour chaque service du modèle fonctionnel de l'application pervasive, le module d'adaptation de contenu sélectionne les opérations d'adaptation de données nécessaires pour adapter les sorties qui répondent au mieux à leur contexte d'utilisation.

Ainsi, dans un même ordre idée, pour construire le graphe d'adaptation, nous avons élaboré un algorithme qui permet de proposer toutes les possibilités d'adaptation qui permettent d'aboutir à un résultat adapté au contexte d'utilisation tout en sélectionnant le média qui répond au mieux à la situation de contexte en cours (cf figure 4.3). Le graphe, produit par cet algorithme, permet de présenter plusieurs chemins possibles pour réaliser l'adaptation de contexte selon le média correspondant. Pour réaliser cet algorithme, nous sommes partis de deux algorithme issus de [Berhe05] que nous avons précédemment décrits dans la section 1.4.2 de l'état de l'art (cf figure 4.2). En effet, un premier algorithme dédié à la préparation de l'adaptation de contenu qui se situe dans le gestionnaire d'adaptation de COALA (cf figure 1.16). Dès que le service adapté est choisi, le gestionnaire d'adaptation commence à préparer le processus d'adaptation de contenu.

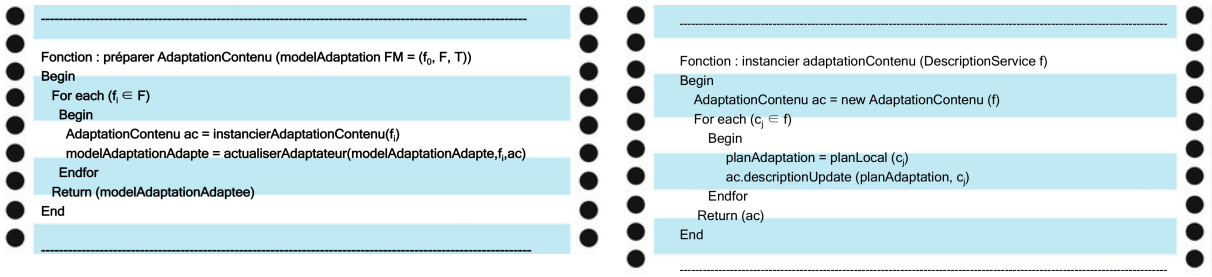

FIGURE 4.2 – Rappel des algorithmes utilisés dans l'adaptateur de contenu de COALA (issus de [Berhe05])

Au fur et à mesure, le gestionnaire d'adaptation de contenu instancie l'adaptateur de contenu pour chaque service de modèle d'adaptation fourni à la couche d'adaptation de contenu (cf figure 1.17).

Notre stratégie d'adaptation garantit l'adaptation des données d'entrée et de sortie des services. En effet, l'adaptateur dans COALA permet de garantir, pour chaque donnée échangée avec l'utilisateur, une adaptation de contenu à la situation contextuelle courante. Les opérations d'adaptation sont instanciées et déployées dynamiquement par le gestionnaire d'adaptation COALA pour chaque service dans l'application. Pour conclure, le processus d'adaptation de contenu produit un nouveau modèle fonctionnel avec des services fournissant des données adaptées à leur contexte d'utilisation.

L'ADAPTATION DE PRÉSENTATION

L'adaptation de contexte dans les environnements pervasifs peut être assez facilement réalisée à travers l'utilisation de techniques d'adaptation au contexte d'utilisation : adaptation de contenu, adaptation de présentation. . . Il apparaît nécessaire d'effectuer ces actions de manière automatisée pour éviter à l'utilisateur d'avoir à faire des manipulations

pour lesquelles il n'est pas compétent ou pour lesquelles il n'a pas envie de perdre son temps dans des menus de configuration et des interfaces de présentation des données.

Le fait de mettre en place une politique ou une stratégie d'adaptabilité de présentation permet à l'utilisateur d'oublier totalement la partie configuration afin que l'adaptation de la présentation des services de son application soit transparente. Les applications interactives nécessitent la présentation des données sur des interfaces adaptables. En effet, les interfaces d'interactions assurent les échanges de données entre l'utilisateur final de l'application et les différents services de l'application pervasive. La qualité de ces interfaces réside dans leur fonctionnement adapté à leur contexte d'utilisation. La bonne exécution de ces interfaces est conditionnée par leur capacité à s'adapter au terminal utilisé (les caractéristiques logicielles et matérielles) ainsi qu' aux préférences des utilisateurs.

Dans COALA, l'adaptation de présentation dépend principalement des caractéristiques du terminal utilisé lors de l'utilisation de contexte (situation contextuelle courante), des préférences de l'utilisateur, et surtout de sa localisation et de ses actions sur l'application interactive pervasive. L'adaptation de présentation dans notre plateforme consiste à bien gérer les interfaces graphiques, qui permettent aux utilisateurs d'interagir avec ses applications, selon les situations contextuelles courantes d'utilisation de contexte. En effet, pour chaque service de l'application, le gestionnaire d'adaptation dans COALA génère l'interface adéquate qui permet de fournir des facilités de navigation pour que l'utilisateur puisse interagir facilement avec les services de son application et donc atteindre ses objectifs.

```
-----------------------------------------------------------------------------------------------------------

Procedure : selectRessource (Boolean connex, String loc)
Begin
   if (connex==false) then
         activerRessource (" msg-vocal ")
   Elseif
         if (loc == " outside office ")
            Begin
              activerRessource (" audio ")
              activerRessource (" vidéo ")
              activerRessource (" msg-instantane ")
              activerRessource (" notif-mail ")
            Endif
   Endif
End

-----------------------------------------------------------------------------------------------------------
```

FIGURE 4.3 – Algorithme général de la sélection de média d'affichage de services dans COALA

4.1.3/ L'ARCHITECTURE DE LA PLATEFORME COALA

COALA peut être représentée sous la forme d'un schéma en couches qui est constitué de trois niveaux principaux (cf figure 4.4). Chaque niveau dépend des services du niveau inférieur. Notons que le niveau le plus bas de notre plateforme (niveau "utilisateurs") contient les concepts qui définissent les concepts généraux du contexte (utilisateur, terminal, localisation, ...). Ce niveau de base correspond aux différents blocs du contexte

qui définissent les traces de contexte dans notre modèle (cf chapitre 3). Le niveau 2 (niveau ”application”) contient la cartographie XML/OWL de la carte de visite contextuelle (CVCO) récupérée, ainsi que la spécification des concepts, définis au niveau bas, dans l'ontologie de contexte. La couche supérieure d'adaptation consiste à l'envoi des décisions aux serveurs d'adaptation et à la détermination des services adaptables ainsi qu'à la mise à jour et à l'affichage de ces services sur les interfaces interactives.

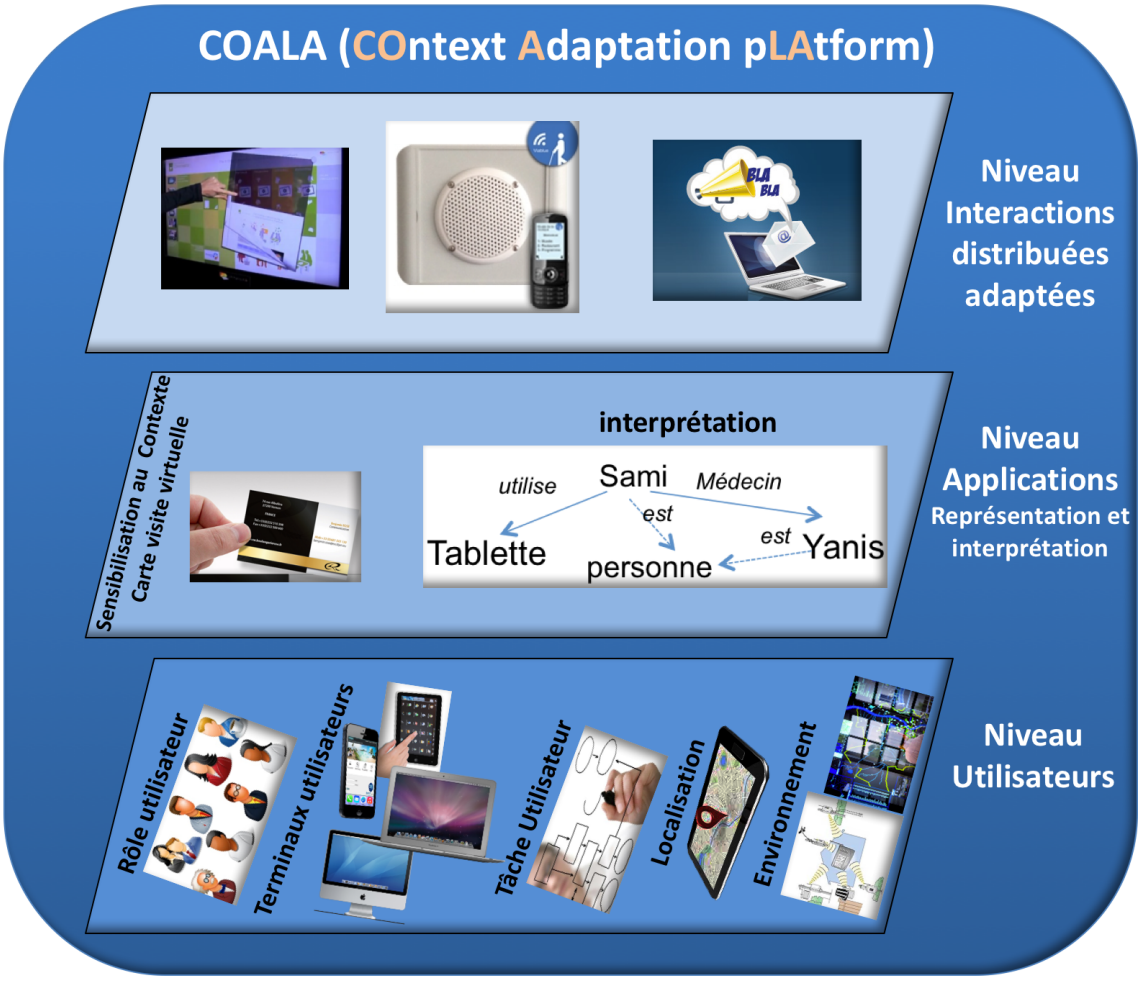

FIGURE 4.4 – L'architecture de la plateforme COALA

- Le niveau ”Modélisation” : c'est à ce niveau que la sensibilité au contexte est définie à travers les différentes dimensions essentielles de contexte à savoir (le profil de l'utilisateur, le dispositif technique, la tâche d'utilisateur, son environnement, ...). À ce stade, COALA décrit le contexte d'un utilisateur par le n-uplet suivant :

$$< \textit{Rôle, Dispositif, Tâche, Environnement, Localisation} >$$

Dans COALA, un contexte C_i peut être défini comme suit :

$$C_i =< R_i, D_i, T_i, E_i, Li >$$

Chacune de ces dimensions dans l'ontologie a son importance dans COALA puisqu'elle perme de définir le contexte de l'utilisateur d'une manière générale. Dans notre plateforme, le modèle du contexte de l'utilisateur (cf Figure 4.5) est situé au niveau bas de l'architecture comme base de COALA. Le modèle est illustré dans la figure suivante ou chaque dimension est représentée par un rectangle.

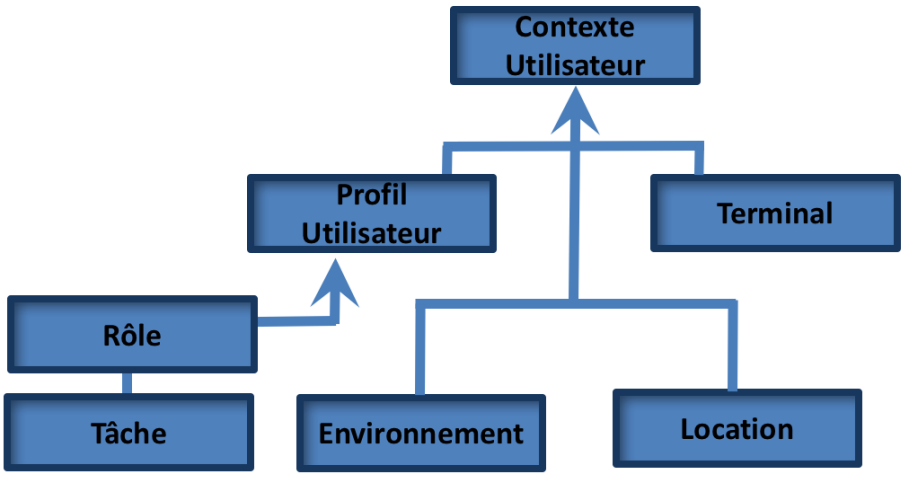

FIGURE 4.5 – Modèle du contexte de l'utilisateur dans COALA

Ces différents dimensions sont décrites ainsi :

- **Le profil utilisateur** : il est composé de toutes les informations, concernant l'utilisateur, qui seront utiles dans la modélisation de l'entité utilisateur. La qualité de la définition du profil de l'utilisateur est importante pour les informations délivrées au système. Dans COALA, le profil utilisateur, décrit dans le modèle proposé, prend en compte la situation professionnelle de l'utilisateur à travers la modélisation de son rôle, et de sa tâche. Ces informations seront regroupées dans la dimension de profil de l'utilisateur avec le reste des informations sur la situation de l'utilisateur (son état civil, ses centres d'intérêt, ses préférences, ...). Tous ces paramètres seront utiles dans la phase d'adaptation.
- **Le rôle** : c'est la description des différents rôles des utilisateurs qui utilisent un tel système d'interaction adapté.
- **La tâche** : la définition de la tâche (l'activité et les sous-activités) des usagers dans le modèle utilisateur est importante surtout dans la description des situations contextuelles courantes des utilisateurs. Cette dimension généralement est représentée à travers un modèle d'activité qui sera spécifique à une application donnée.
- **Le dispositif technique** : ce profil permet de décrire les capacités d'un dispositif mobile lors d'une situation d'utilisation de contexte. Dans les environnements mobiles, il existe une diversité des terminaux utilisés : de type PDA, Smartphone ou Laptop, ...Ces dispositifs sont caractérisés par leurs paramètres utilisés dans cet environnement tels que la largeur de l'écran, la résolution, la puissance de CPU, ...Ces informations sont importantes car, par exemple, les autonomies de deux types de terminaux peuvent être différentes et dans ce cas le système ne fournira pas ni les mêmes informations ni les mêmes moyens de communication à deux utilisateurs qui utilisent deux types différents de terminaux. Comme dans la dimension utilisateur, la dimension dispositif est représentée à l'aide d'un modèle de terminal qui prend en considération les

données suivantes : le type (Smartphone, Pc , tablette, . . .), l'autonomie de la batterie pour la mobilité, l'espace mémoire, la taille de l'ecran, l'espace modalité d'entrées/sorties (clavier, écran tactile, micro, . . .).
Un profil comprend les caractéristiques des réseaux utilisés par les terminaux comme pour leur bande passante, leur connectivité, leur qualité de service, . . . Dans les environnements ambiants, les réseaux de connections utilisés sont, par exemple, les réseaux sans fil tels que le WLAN et le Bluetooth dans les PAN. Par conséquent, les applications mobiles supportées par ce genre d'environnements doivent être adaptées à ces caractéristiques. Un dispositif peut également se retrouver connecté à différents réseaux en fonction de sa localisation, dans le cadre d'une utilisation mobile.

- **L'environnement** : cette dimension sert à enrichir le modèle de contexte par d'autres paramètres utiles selon le domaine d'application. Par exemple, nous pouvons trouver dans la dimension environnement la température ambiante, l'état actuel de l'utilisateur (conduire une voiture, occupé, . . .).
- **La localisation de l'utilisateur** : cette dimension est indispensable puisqu'elle permet d'indiquer l'endroit où se situe l'utilisateur d'un point de vue spatial par rapport aux autres objets (autres utilisateurs, réseaux disponibles, bande passante, . . .).

- Le niveau "Application" : dans COALA, c'est le niveau principal de la plateforme. Il constitue le noyau pour la gestion de contexte. En effet, il gère le moteur de l'ensemble des opérations contextuelles. L'architecture générale du niveau "Application" est illustrée sur la figure 4.4. Nous distinguons deux modules principaux dans le gestionnaire de COALA : un module permet de modéliser les situations contextuelles courantes d'utilisation à travers l'exploitation des informations contextuelles récupérées et placées dans la carte de visite contextuelle à l'instant t d'utilisation et un autre module d'adaptation du système permet d'interpréter et de raisonner sur les informations contextuelles traitées.
  En effet, pour la modélisation de la sensibilité au contexte, le gestionnaire poursuit tout un enchaînement d'étapes commençant par la phase de de récupération des informations qui composent le contexte. Cette collecte est faite lors de la première connexion de l'utilisateur sur sa session. Une phase de génération de la carte de visite contextuelle (*CVCO*) est par la suite produite. Or, dans une situation contextuelle S : un Dispositif D , une Tâche utilisateur T, un Rôle utilisateur R et à un localisation spatiale L, une carte CVCO sera définie et générée. Cette carte prend comme données l'ensemble des paramètres et des traces de contexte d'utilisation dans la situation contextuelle courante. Cette CVCO sera envoyée au moteur d'adaptation afin de réaliser l'interprétation et le raisonnement sur le contenu de celle-ci. Ensuite, le gestionnaire prend en considération la situation contextuelle courante récupérée et envoie des requêtes au serveur d'adaptation. Une information contextuelle adaptée selon les différents paramètres d'utilisation sera générée et sera envoyée pour la distribuer et la visualiser sur l'interface de l'utilisateur.
- Le niveau d'interaction adaptée : l'adaptation d'interface prend une dimension très importante surtout dans le cas où l'on veut ajouter la sensibilité au contexte à des applications ambiantes pour qu'elles soient adaptées au nouveau besoins dus aux changements de leur environnement d'utilisation. Dans COALA, nous essayons de rendre l'application interactive, ainsi les données sont adaptées dynamiquement aux caractéristiques du terminal utilisé et de l'utilisation en cours.

LA GESTION ET L'ADAPTATION DE CONTEXTE

Comme nous avons choisi d'appliquer notre plateforme à l'adaptation de contexte dans le domaine médicale, il est important de noter qu'il existe un grand nombre de tâches et activités médicales qui doivent être gérées rapidement dans une grande variété de contexte d'utilisation.

"Le qui", "le quand" et "le comment" de la manipulation des applications pervasives peut entraîner des problèmes au niveau de la gestion et de l'adaptation des interfaces et des services de ces applications. Pour résoudre ce problème, notre plateforme permet l'utilisation d'ontologies de traçabilité qui gère le contexte dans des situations courantes dynamiques de l'application pervasive. En effet, nos ontologies peuvent également être utilisées pour les systèmes sensibles au contexte afin de mieux interagir et comprendre les requêtes des utilisateurs auxquelles ils doivent répondre.

4.2/ FONCTIONNALITÉS DE COALA

Chaque déclenchement d'une session utilisateur d'une application pervasive est considérée comme une situation courante d'utilisation de contexte. Chaque situation courante de contexte est considérée comme une instance des paramètres de ce contexte. Le changement des situations de contexte engendre automatiquement le changement des paramètres et donc le changement des services et les manières de présenter l'information sur les interfaces des utilisateurs. Ainsi, l'application sensible au contexte dans COALA doit s'adapter aux changements de toutes les situations d'utilisation tout en tenant compte des besoins des utilisateurs courants de contexte à travers les traces récupérées.

Une trace de contexte est définie à travers quatre dimensions :

< **U***tilisateur[Rôle+Tâche,Activité],* **T***erminal,* **S***ervice,* **E***nvironnement[Localisation]* >

La plateforme COALA est dotée d'un système qui permet de fournir des applications interactives sensibles au contexte d'utilisation. Il permet de structurer et de représenter la connaissance de contexte utilisateur fondée sur un mécanisme qui permet de fournir la traçabilité des situations contextuelles au cours de l'utilisation.

Pour décrire le fonctionnement de COALA (cf figure 4.6), nous définissons une carte de visite contextuelle (*CVCO*) qui permet de fournir la traçabilité de contexte. Les paramètres liés à une situation contextuelle courante seront récupérés et stockés dans cette carte contextuelle. Le contenu de cette carte est représenté en XML. La représentation XML des traces récupérées nous permet d'organiser les paramètres sous forme de blocs tout en offrant la possibilité de la structuration hiérarchique entre ces blocs.

À ce stade, nous avons identifié deux principaux composants du système de COALA : le gestionnaire de modélisation de la sensibilité au contexte et le gestionnaire de l'adaptation à ce contexte (cf figure 4.6) :

- le gestionnaire de sensibilité au contexte COALA (*COALA Awareness manager*) permet de gérer le module de modélisation de contexte,

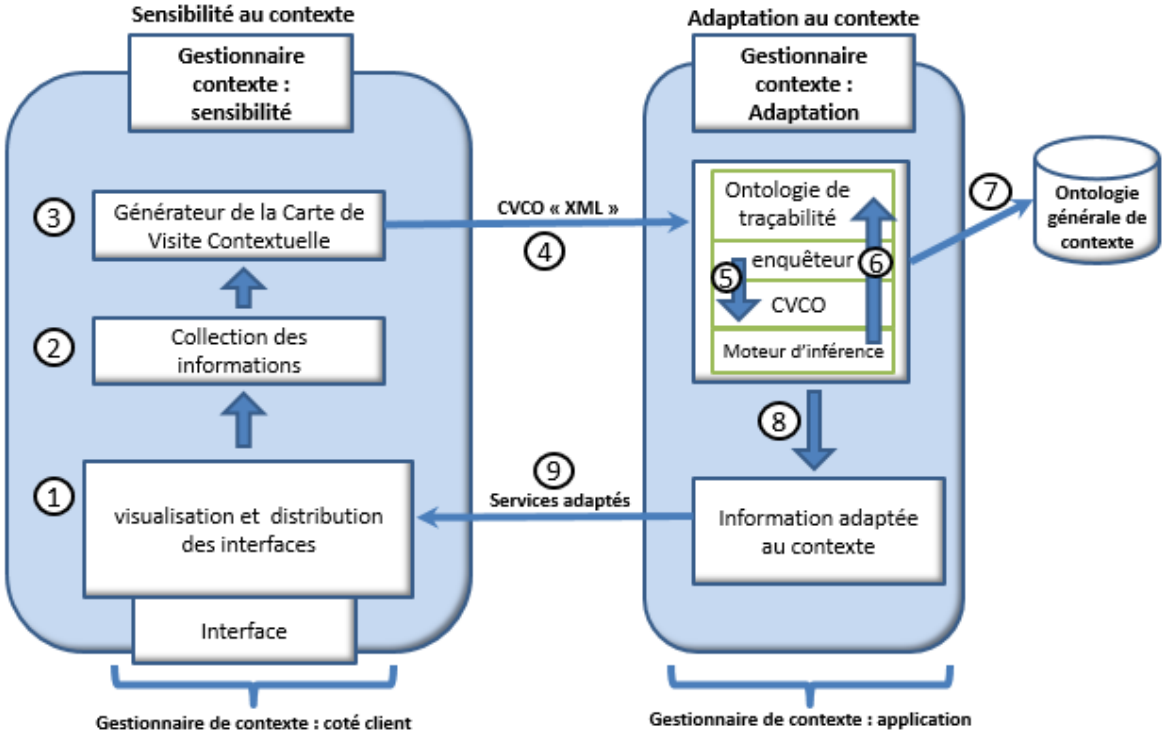

FIGURE 4.6 – Le fonctionnement de la plateforme COALA

- le gestionnaire de l'adaptation COALA est le module d'application qui permet l'adaptation des application interactives pervasives aux contextes d'utilisation de manière dynamique et instantanée.
  La figure 4.6 montre les principales fonctionnalités de COALA que l'on peut décrire de la maniéré suivante :
  - ❶ L'interface de l'application utilisateur qui permet à la fois la collecte des paramètres de contexte au cours d'une situation d'utilisation et la visualisation de la liste des services adaptées, suite à une requête de recherche d'un service.
  - ❷ Une fois l'utilisateur connecté sur sa session, le gestionnaire COALA collecte les paramètres et les traces de contexte des différentes situations courantes du contexte.
  - ❸ Cette étape permet de générer automatiquement la carte de visite de la situation contextuelle en cours.
  - ❹ La carte sous son format XML est envoyée puis utilisée par l'adaptateur COALA afin de permettre l'adaptation des besoins des utilisateurs par rapport à ses applications pervasives.
  - ❺ Au cours de cette étape, le noyau d'adaptation reçoit l'ensemble des traces liées à une situation d'utilisation de contexte, afin de les exploiter comme des paramètres contextuels récupérés par la CVCO.
  - ❻ L'adaptateur, à ce niveau, gère l'ontologie de la traçabilité pour l'exploiter dans la tâche d'adaptation tout en la mettant en communication avec l'ontologie générale de contexte.

❼ Dans cette étape, comme les connaissances générales de contexte seront utilisées dans la prise de décision des éléments d'adaptation, l'adaptateur met en relation les différentes connaissances situées dans l'ontologie de traçabilité à celles du contexte général.
❽ Au cours de cette étape, l'adaptateur interroge les ontologies en termes de connaissances selon les paramètres de contextes qu'il possède pour une situation courante, afin de générer les informations adaptées à ce contexte d'utilisation courant.
❾ À ce stade, le gestionnaire de COALA reçoit l'ensemble des informations d'adaptation qui seront affichées sous forme de services adaptés (service audio, service vidéo, ...) sensibles au contexte d'utilisation.

4.3/ DÉVELOPPEMENT ET UTILISATION DE COALA

Afin de tester les performances dans un contexte théorique, nous avons implémenté les différentes contributions dans des prototypes à travers des scénarios expérimentaux qui permettent de tester la fonctionnalité de notre plateforme COALA pour la modélisation et l'adaptation des situations contextuelles. Nous avons implémenté la gestion des informations des contextes à travers la définition de nos différentes ontologies dédiées à cette tâche. Nous avons également mis en place notre plateforme COALA d'adaptation qui s'appuye sur des mécanismes qui permettent de fournir les traces d'utilisation de contexte dans des situations courantes d'utilisation de ce contexte.

Nous avons implémenté pour ce faire une carte de visite contextuelle qui permet de fournir la traçabilité de contexte lors de d'utilisation instantanée tout en suivant le processus de Mapping qui permet de fournir au final une ontologie de traçabilité qui sera en communication avec l'ontologie générale de contexte et le processus d'adaptation dans COALA. Pour tester les différentes parties de la plateforme, nous avons implémenté des prototypes permettant d'évaluer les performances de nos contributions.

4.3.1/ CHOIX DES OUTILS D'IMPLÉMENTATION

Dans cette présente section, nous décrivons les choix des outils que nous avons utilisés pour développer notre prototype de COALA. Pour la réalisation de notre plateforme, nous avons utilisé différents outils. Ils permettent d'une part le développement de nos ontologies, et d'autre part leur manipulation à travers les requêtes et leur communication.

Nous avons choisi d'utiliser des outils qui permettent de réaliser le processus de Mapping du contenu XML récupéré dans la CVCO pour le transformer en une ontologie de traçabilité de contexte. Ces différents outils sont présentés ci-dessous :

- Protégé 4.3 est un éditeur d'ontologies et Framework de gestion des connaissances et de développement des ontologies. Il a été développé en open-source. Protégé permet de développer une ontologie dans un domaine donné (ontologie de domaine), de définir des formalités d'entrée des données et de les acquérir sous forme d'instances de cette ontologie. Pour créer de véritables applications à base de connaissances, il existe également une librairie Java qui peut être étendue pour cette tâche en utilisant un moteur d'inférence pour le raisonnement et la déduction de nouveaux faits par l'application de règles d'inférences aux

instances de l'ontologie de domaine.

- Jena est une API Java open source qui contient des classes et des interfaces pour la manipulation des modèles décrits en RDF, RDFS et OWL. Jena permet également de stocker les connaissances et d'appliquer certains mécanismes d'inférences.

- SPARQL est un langage de requêtes orienté données. Il permet d'interroger les informations détenues dans les ontologies. SPARQL prend la description de ce que l'application souhaite, sous forme de requêtes et retourne ces informations sous la forme d'un ensemble de liaisons ou d'un graphe RDF.

- Java est une plateforme open-source bien connue qui nous a permis, en particulier dans le cadre de cette thèse de manipuler de nombreux langages pour les ontologies (comme jena), et de réaliser nos interfaces de communication,. . .

- Trang est une API java open-source qui est construite autour d'un modèle d'objet RELAX conçu pour soutenir la transformation de schéma(XSD) d'un document XML. La sortie fournie par Trang donne un schéma compréhensif y compris les définitions, la façon dont le schéma a été divisé dans le fichier, les annotations et les commentaires.

- XSOM : Xml Schema Object Model est une bibliothèque Java qui permet aux applications d'analyser facilement des documents XML Schema et inspecte les informations contenues. Il permet de prendre schéma XML en entrée(input).

- La dernière version d'Apache Maven 3.3.9 qui utilise un paradigme connu sous le nom de Project Object Model (POM) ainsi que la distribution automatique de modules jar, a été utilisée.

Pour conclure, nous avons utilisé l'API du web sémantique Jena, le raisonneur Pellet et l'éditeur Protégé. Nous utilisons dans l'implémentation de nos ontologies le langage OWL et l'éditeur Protégé car il possède une interface utile pour définir l'ensemble des règles qui sont ensuite incluses dans le fichier de l'ontologie OWL. Quant à Jena, c' est une API Java qui contient des classes et des interfaces pour l'interaction avec les modèles RDF et OWL et les raisonneurs comme Pellet [Pars04]. Mapping XSD2OWL permet de générer automatiquement notre ontologie de traçabilité à partir du contenu des paramètres de contexte récupérés dans la CVCO. Nous avons suivi le processus de cartographie tout en utilisant l'ensemble des outils correspondants. En effet, nous avons commencé par l'utilisation de l'API java Trang. Cet API, comme nous l'avons expliqué précédemment, permet de valider le contenu sous son format XML : XML-Schéma (XSD). Nous utilisons la bibliothèque Java XSOM (XML Schéma Object Model) afin d'analyser par la suite le contenu XSD. La sortie générée par XSOM est utilisée comme une entrée pour réaliser la cartographie java XSD2OWL.

Suivant l'enchaînement de l'ensemble des étapes dans le processus de génération de l'ontologie de traçabilité, nous utilisons l'API Jena pour les sorties de SXD2OWLMapping avec le serveur Apache Maven pour la production de sémantique supplémentaire des données RDF.

4.3.2/ IMPLÉMENTATION DES ONTOLOGIES DANS COALA

L'implémentation de nos ontologies (ontologie générale de contexte et l'ontologie de traçabilité de ce contexte) dans COALA consiste tout d'abord à l'édition, sous Protégé 4.3, de ces ontologies avec le langage OWL. L'ontologie de traçabilité a été alimentée par les traces de contexte récupérés dans la CVCO. Nous avons également utilisé les requêtes SPARQL afin de répondre aux requêtes faites par les utilisateurs (staff hospitalier par exemple) afin d'adapter les formes d'affichage dans interfaces générées dans COALA. Cette organisation est représentée dans la figure 4.7.

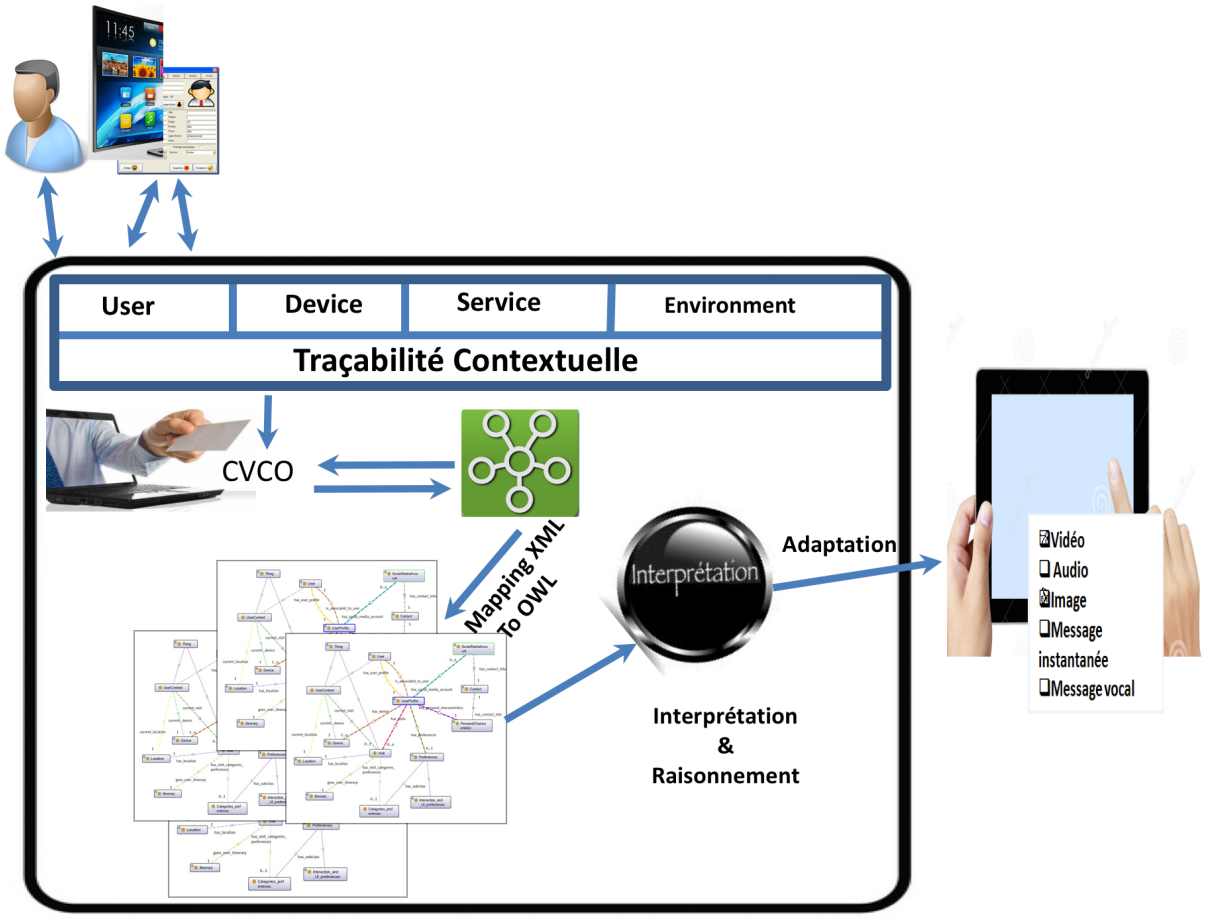

FIGURE 4.7 – les interactions dans la plateforme COALA

En ce qui concerne la relation entre nos ontologies, compte tenu de la diversité de nos deux ontologie (ontologie générale de contexte et ontologie de traçabilité) en termes de domaine représenté, de vocabulaire, de la représentation des connaissances, il est important d'établir un lien conceptuel entre elles afin de pouvoir les utiliser dans la même application pour adapter les besoins des utilisateurs (cf figure 4.8. Notre idée consiste à coupler les deux ontologies afin de faire en sorte les ulitiser comme une ontologie globale de contexte à base de traçabilité des situations courantes de contexte, et qui sera utilisée comme une base de connaissance générale pour le domaine de la sensibilité au contexte. En effet, le couplage entre deux ontologies veut dire qu'impérativement faut trouver les relations qui existent entre ces deux ontologies . Nous avons fait ce couplage

de façon manuelle. Comme les deux ontologies sont simple et ne nécessitent pas le recours a des méthodes de couplage automatique. or le couplage consiste à simple trouver une superclasse qui peut regrouper les différents profils attribués par les deux ontologies. Nous essayons de trouver des correspondances potentielles entre les classes de superclasses définies dans l'ontologie générale de contexte et l'ontologie de traçabilité des situations courantes. Par exemple, la classe « HWProfile » et « SWProfile » de l'ontologie de traçabilité peut entrer une sous classe de 'Terminal' de l'ontologie générale de contexte. Nous avons pensé à faire cette correspondance afin même d'éviter la redondance dans nos ontologies.

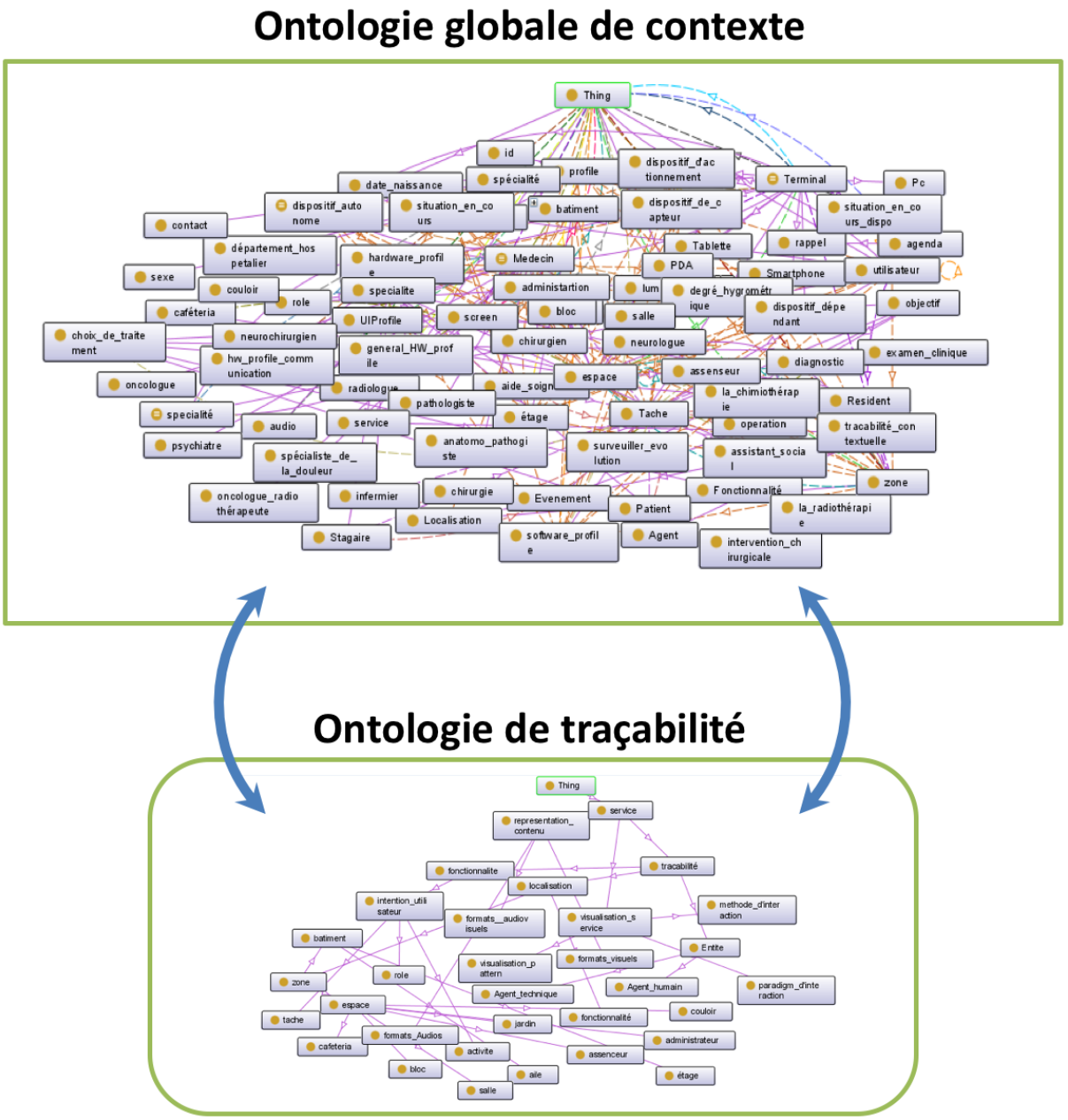

FIGURE 4.8 – L'interaction entre les ontologies dans la plateforme COALA

4.3.3/ IMPLÉMENTATION DE LA CARTE DE VISITE CONTEXTUELLE : CVCO

```
<contextTrace name="proccessorArchit">x86</contextTrace>
<contextTrace name="coreSpeed1">2.26</contextTrace>
<contextTrace name="memSlot">1024</contextTrace>
<contextTrace name="resolution">9 pce</contextTrace>
  <contextTrace name="hardDissize">600GB</contextTrace>
  <contextCategory name="SWProfileConfiguration">
<contextTrace name="systemBoot">android</contextTrace>
<contextCategory name=" connectionNetwork ">
    <contextTrace name="Hostname">SALZO</contextTrace>
    <contextTrace name="IPname">192.168.0.1</contextTrace>
    <contextTrace name="proxyserveur">proxyweb.univ-fcomte.fr</contextTrace>
    <contextTrace name="connectionType">speedWAN</contextTrace>
    <contextTrace name="Bandwidth">336.8kbps</contextTrace>
</contextCategory >
<contextBloc name="Location">
    <contextTrace name="batname">ARAGO</contextTrace>
    <contextTrace name="blocname">CHIR</contextTrace>
    <contextTrace name="étageN'">3</contextTrace>
    <contextTrace name="roomNumber">111</contextTrace>
</contextBloc>
<contextBloc name="EnvironmentPervasif">
  <contextTrace name="proccessorArchit">x86</contextTrace>
  <contextTrace name="coreSpeed1">2.26</contextTrace>
  <contextTrace name="memSlot">1024</contextTrace>
  <contextTrace name="resolution">9 pce</contextTrace>
  <contextTrace name="hardDissize">600GB</contextTrace>
  <contextCategory name="SWProfileConfiguration">
  <contextTrace name="systemBoot">android</contextTrace>
  <contextCategory name=" connectionNetwork ">
    <contextTrace name="Hostname">SALZO</contextTrace>
    <contextTrace name="IPname">192.168.0.1</contextTrace>
    <contextTrace name="proxyserveur">proxyweb.univ-fcomte.fr</contextTrace>
    <contextTrace name="connectionType">speedWAN</contextTrace>
    <contextTrace name="Bandwidth">336.8kbps</contextTrace>
  </contextCategory >
  <contextBloc name="Location">
  <contextTrace name="batname">ARAGO</contextTrace>
  <contextTrace name="blocname">CHIR</contextTrace>
  <contextTrace name="étageN'">3</contextTrace>
  <contextTrace name="roomNumber">111</contextTrace>
</contextBloc>
```

FIGURE 4.9 – Extrait XML de la CVCO d'une situation contextuelle courante.

Dans les environnements pervasifs sensibles au contexte, les applications doivent pouvoir prendre en compte dynamiquement le changement des situations d'utilisation dans l'environnement, ainsi que les informations de contexte en provenance de l'environnement. Ces flux d'informations ne peuvent pas être déterminés à l'avance et doivent se construire pendant l'exécution. AInsi pour adapter les flux, nous avons mis en place dans notre plateforme un gestionnaire qui permet de répondre aux besoins et aux préférences des utilisateurs en tenant compte de toutes les contraintes et changements dynamiques du contexte d'utilisation.

Ce gestionnaire permet, dans un premier temps, de récolter et de récupérer les infor-

mations à travers la mise en place du mécanisme de la carte de visite contextuelle. La récupération des CVCO au cours de l'exécution permettra la création de traces. Ces traces dynamiques de contexte courant permettront d'alimenter l'ontologie de traçabilité.

```
-----------------------------------------------------------------------------------------------------------------
Input: xml from CVCO data source
output: OWL ontologies Traceability
Begin
    //convert Xml from CVCO data source to Xml schema(XSD)
    Trace.xsd = trang(trace.xml)
    //convert Xml shema (XSD) to an xssschema file using XSOM
    Xssshema testxss=XSOM(trace.xsd)
    //mapping XSD2OWL Traceability
    Create an empty OWL traceability file
    While (testxss.nextElement != null)
    Begin
        //Find the type/the element/the attributs into testxss and convert them to OWL
        Switch (Extract testxss component)
            Case1: convert a simple type;
            Case2: convert a complex type;
            Case3: convert element;
            Case4: convert attributes;
        End Switch
        Insert new component in OWL ontology traceability file
    End
    Create ontology traceability file. N3(owl file)
    Reference generated classes to the existing OWL traceability classes
End

-----------------------------------------------------------------------------------------------------------------
```

FIGURE 4.10 – Mapping de la CVCO dans l'ontologie de traçabilité de contexte

La carte virtuelle de traçabilité est *extensible* en fonction des informations dont le système a besoin pour fonctionner : nous avons donc choisi de représenter les informations d'une manière simple et ordonnée en arborescence XML (cf figure 4.9). Dans COALA, un gestionnaire permet de récupérer les traces contextuelles contenues dans la carte de visite. Ces paramètres de contexte récupérés permettent de constituer une cartographie de l'environnement contextuel de l'utilisateur.

Pour implémenter les CVCO, nous avons créé un module java implémenté dans un client utilisant cette technologie pour récupérer les informations contextuelles. Le fichier XML est automatiquement récupéré, interprété et validé (XSD) en utilisant les outils correspondants (mappage XML2OWL). Nous implémentons notre algorithme de Mapping de la CVCO dans l'ontologie de traçabilité de contexte (cf figure 4.10). En effet, pour valider notre approche, nous avons développé un prototype de la plateforme COALA. Nous nous sommes plutôt intéressés à la couche de gestion et de modélisation ainsi qu'à la couche d'adaptation au service de l'architecture de COALA.

D'une manière générale, il existe plusieurs outils automatisés pour effectuer un raisonnement logique sur les ontologies. Dans notre travail de recherche nous avons employé le moteur de raisonnement Pellet pour vérifier la consistance de notre ontologie et pour déduire de nouvelles informations implicites du contexte. Nous présentons le code OWL de notre modèle de l'ontologie de traçabilité dans la figure 4.11.

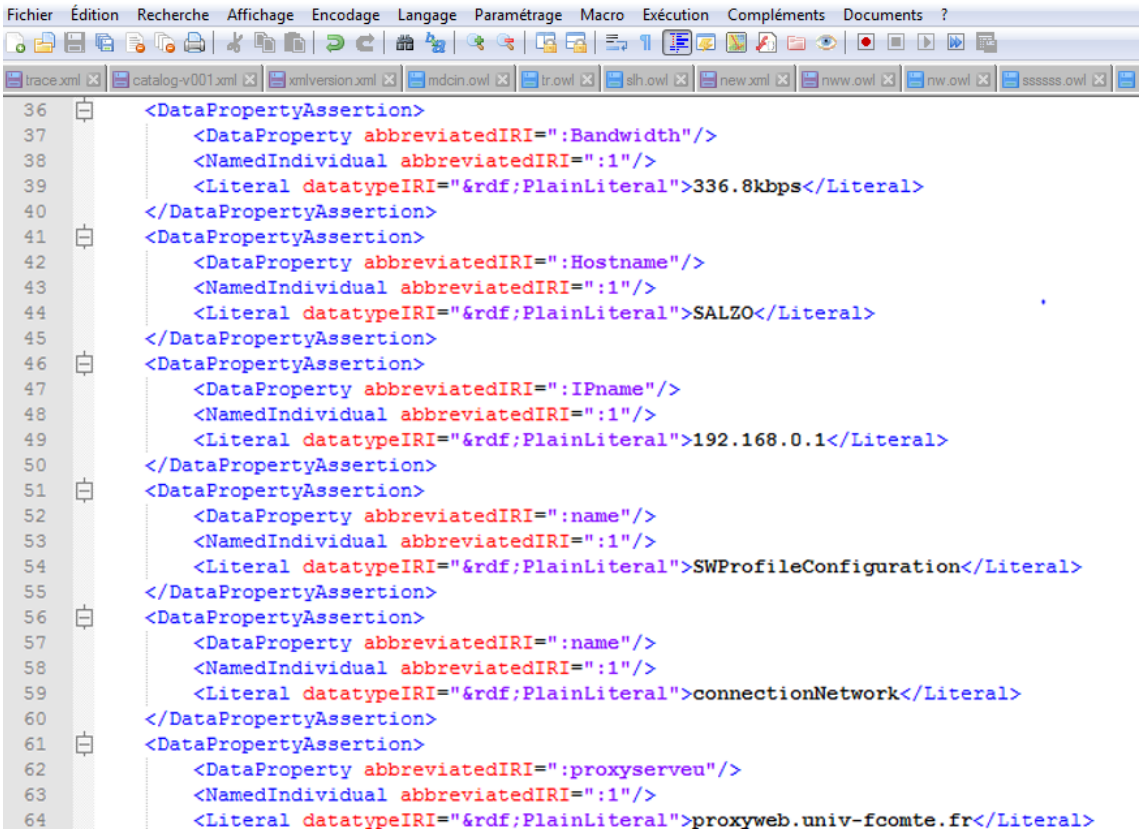

FIGURE 4.11 – Représentation OWL de la traçabilité du contexte dans la plateforme COALA

4.4/ EXEMPLE D'UN SCÉNARIO D'UTILISATION DE CONTEXTE POUR L'ADAPTATION

Nous avons choisi de tester une application médicale de suivi de l'évolution de tumeur du cerveau, afin de l'intégrer dans notre plateforme pour la modéliser et l'adapter aux situations instantanées de contexte. Cette intégration nous a permis de valider notre architecture et notre approche de la sensibilité au contexte. Nous consacrons ce chapitre à la présentation de la conception et de la réalisation de notre prototype tout exposant la manière dont nous l'avons déployé pour adapter l'application médicale de suivi de l'évolution de tumeur de cerveau.

4.4.1/ INTRODUCTION AU SCÉNARIO EXPÉRIMENTAL DE NOTRE PROTOTYPE

Dans cette section, nous présentons un test d'implémentation du cas d'utilisation : "le suivi de l'évolution de tumeur de cerveau chez Mme Josiane X" (cf figure 4.12). Afin de mettre en évidence la clarté, la cohérence, l'employabilité de notre ontologie de traçabilité, le déploiement de notre carte visite contextuelle et son rôle dans la fourniture de la sensibilité de contexte, nous allons présenter notre exemple de cas d'étude.

En effet, pour illustrer le processus de génération de l'ontologie de traçabilité intégré dans

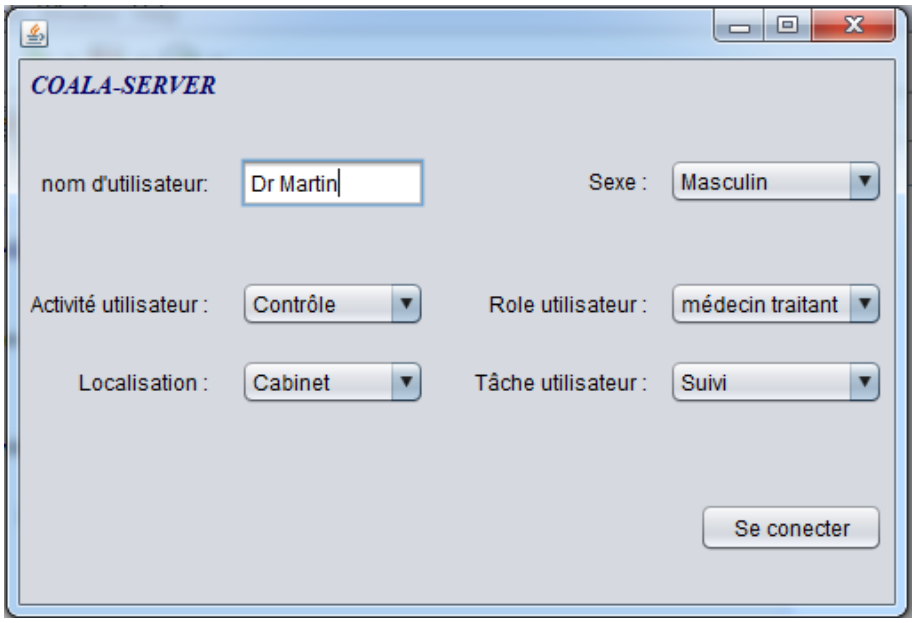

FIGURE 4.12 – Interface principale de connexion dans COALA.

notre plateforme COALA, nous proposons un prototype de l'application de télé-chirurgie de la tumeur du cerveau, inspiré d'un cas clinique réel. Cette proposition est vue d'un point de vue générique et peut être appliquée à plusieurs scénarios.

Comme nous l'avons indiqué, notre choix est de proposer un prototype pour l'application de suivi de la tumeur du cerveau. Nous commençons par décrire le centre hospitalier et la composition du personnel dans le domaine du traitement des tumeurs du cerveau : les médecins (professeur, résident, interne, urgence, ...), les personnels hospitaliers (infirmière, aide soignante, ...) et les patients. En effet, le milieu médical pour ce cas est un environnement critique qui touche la vie des êtres humains et qui nécessite l'intervention et la collaboration du staff complet.

Notre fil rouge est le cas suivant :

- Après l'apparition chez Josiane, âgée de 54 ans, de symptômes qui permettent de suspecter la présence d'une tumeur du cerveau, il a été procédé à un scanner cérébral.
- À ce stade, le médecin traitant, le Docteur Martin, joue un rôle important dans le diagnostic, car c'est souvent lui qui est consulté en premier. En cas de suspicion de tumeur, plusieurs médecins de spécialités différentes (réunions de concertation pluridisciplinaire ou *RCP*) se réunissent pour discuter des solutions de traitements possibles. Ils se basent pour cela sur des recommandations de bonne pratique. Il existe un grand nombre de tumeurs du cerveau différentes. Selon leur localisation, leur taille et leur agressivité, c'est-à-dire la vitesse à laquelle elles se développent, ces tumeurs n'entraînent pas les mêmes symptômes et n'ont pas la même gravité. Les symptômes provoqués par une tumeur cérébrale diffèrent en fonction de la taille de la tumeur et de sa localisation. Il peut s'agir notamment de maux de tête, de crises d'épilepsie ou de troubles fonctionnels (difficultés à parler, à coordonner ses mouvements, à se repérer dans l'espace, ...). La chirurgie est le traitement principal des tumeurs cérébrales. D'autres traitements comme la radiothérapie ou

la chimiothérapie sont utilisés en complément ou lorsque la chirurgie est impossible. Le choix des traitements est adapté à la situation du patient.

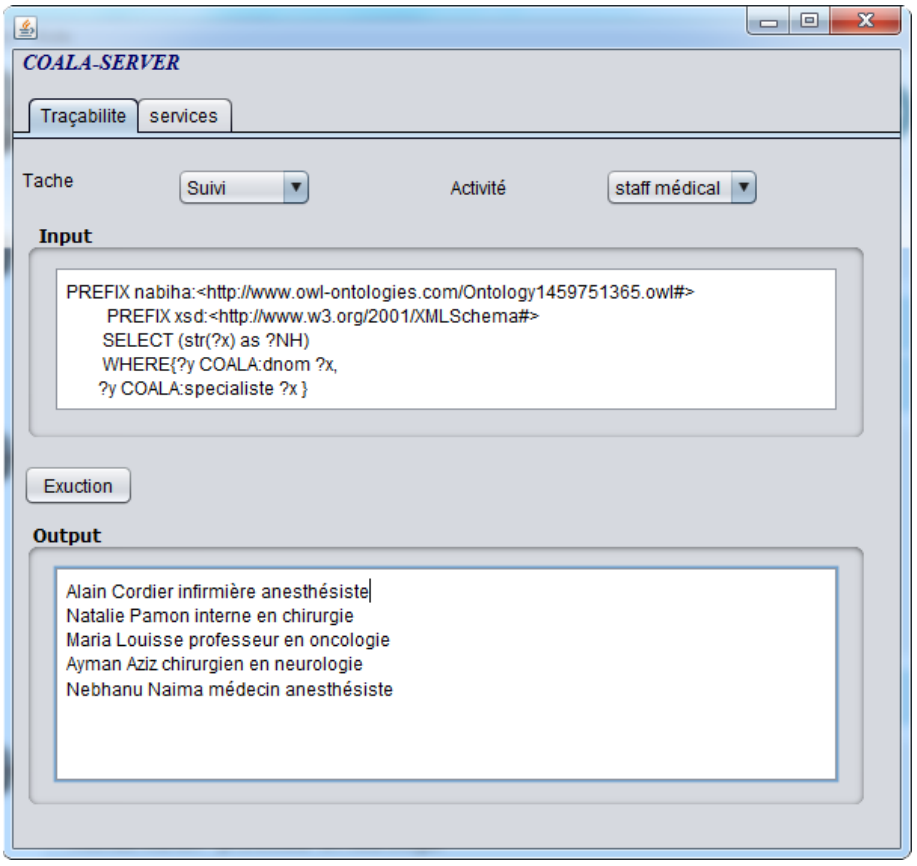

FIGURE 4.13 – Extraction de Rôle dans COALA pour l'adaptation à chacun des intervenants

- L'équipe de prise en charge est composée de médecins et de soignants de différentes spécialités (cf figure 4.13) : neurochirurgien, neurologue, oncologue, chirurgien, anatomopathologiste, ...

4.4.2/ CARACTÉRISATION ET MODÉLISATION DES SITUATIONS CONTEXTUELLES COURANTES

D'après ce que nous venons de caractériser, nous constatons que l'environnement médical est multidimensionnel. Nous pouvons définir différentes situations contextuelles (SC) susceptibles de décrire le contexte courant d'utilisation d'une application pervasive :$SC_1, SC_2, \ldots SC_n$. Chacune de ces situations nécessitera une adaptation Particulière en fonction de contexte courant.

Par exemple, on peut définir une situation Contextuelle SC_i comme suit :

- La situation présente un chirurgien situé dans son cabinet, qui utilise son application pervasive sur son Pc standard pour suivre l'état du patient. Il peut au cours d'un déplacement vers l'hôpital continuer à utiliser l'application sur son smartphone.
- Deux situations contextuelles sont définies à ce niveau (la situation initiale SC_i et

la nouvelle situation en mobilité) et donc une suite de services doit être disponible et adaptable à ces deux situations différentes :

- Les traces sur son profil (tâche, rôle, activité, ...) permettent de définir par exemple le service de droit d'accès aux données,
- La localisation avec la corrélation de l'environnement (du bureau à l'hôpital en passant par la voiture, les ascenseurs, les couloirs, ...), modifie la liste des services qui doivent être adaptés au déplacement de l'utilisateur dans l'environnement (service d'affichage, service de connectivité, ...),
- Les traces liées à son terminal (Pc vers Smartphone) permettent de renseigner sur les services nécessaires à l'adaptation au terminal : sur son Pc l'application est utilisée de manière *complète*, alors que sur son Smartphone cette dernière doit être décomposée en une suite softs adaptées à la faible capacité du Smartphone (réduction au niveau taille d'affichage par exemple ou absence de certaines modalités comme la représentation vidéo par exemple).

Reprenant le cas clinique de la patiente "Josiane", la majorité des tumeurs ne peuvent être retirées que par une chirurgie. Une interversion chirurgicale pour un tel cancer nécessite une coopération entre plusieurs médecins qui auront chacun un rôle spécifique, une compétence spécifique, un contexte d'utilisation spécifique...

Nous allons présenter dans ce qui suit un exemple d'un scenario d'utilisation de contexte d'un médecin Neurologue pour une adaptation de formes d'affichages générées via des interfaces de l'application.

Reprenant le cas clinique de la patiente "Josiane", la majorité des tumeurs ne peuvent être retirées que par une chirurgie. Une interversion chirurgicale pour un tel cancer nécessite un suivi d'un médecin traitant en premier lieu, et une coopération, après, entre plusieurs médecins qui auront chacun un rôle spécifique, une compétence spécifique, un contexte d'utilisation spécifique....

< Rôle="médecin Neurologue", tâche="stimule et analyse l'état actuel", localisation="Bureau 433, service neurologie", terminal="Pc fixe", type de réseau="WAN" >

À partir de ce scénario, nous pouvons en extraire une suite des situations contextuelle à adapter. Étant donné le rôle, la tâche, la localisation, l'environnement où se trouve le médecin traitant (dans notre cas un Neurologue, une situation contextuelle est Créée selon les paramètres Récupérés via la carte de visite contextuelle : on suppose ici que le contenu de fichier XML(figure) est l'ensembles des traces de contexte pour une situation donnée récupérées via la CVCO, qui a été déjà récupérée par le système de COALA. Un extrait de contenu de la CVCO sous son format XML est présenté dans la figure 4.14

Tout d'abord, le contenu CVCO (la source de données) est récupéré sous la forme d'un document XML, puis est transformé en un XML-Schema (validation XSD) en utilisant l'API Trang pour java. En fait, Trang prend en entrée un schéma écrit en XML et produit en sortie un schéma écrit en XML-Schema (XSD), comme dans la figure 4.15.

À ce stade, le schéma XML est analysé en utilisant le XML-Schema Object Model (XSOM est un bibliothèque Java). Il permet d'analyser facilement les schémas XML. La sortie

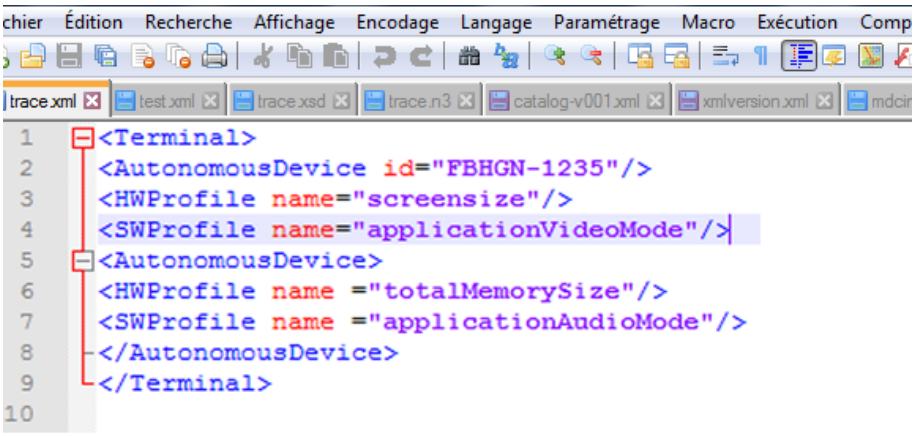

FIGURE 4.14 – Extrait d'une représentation XML de contexte

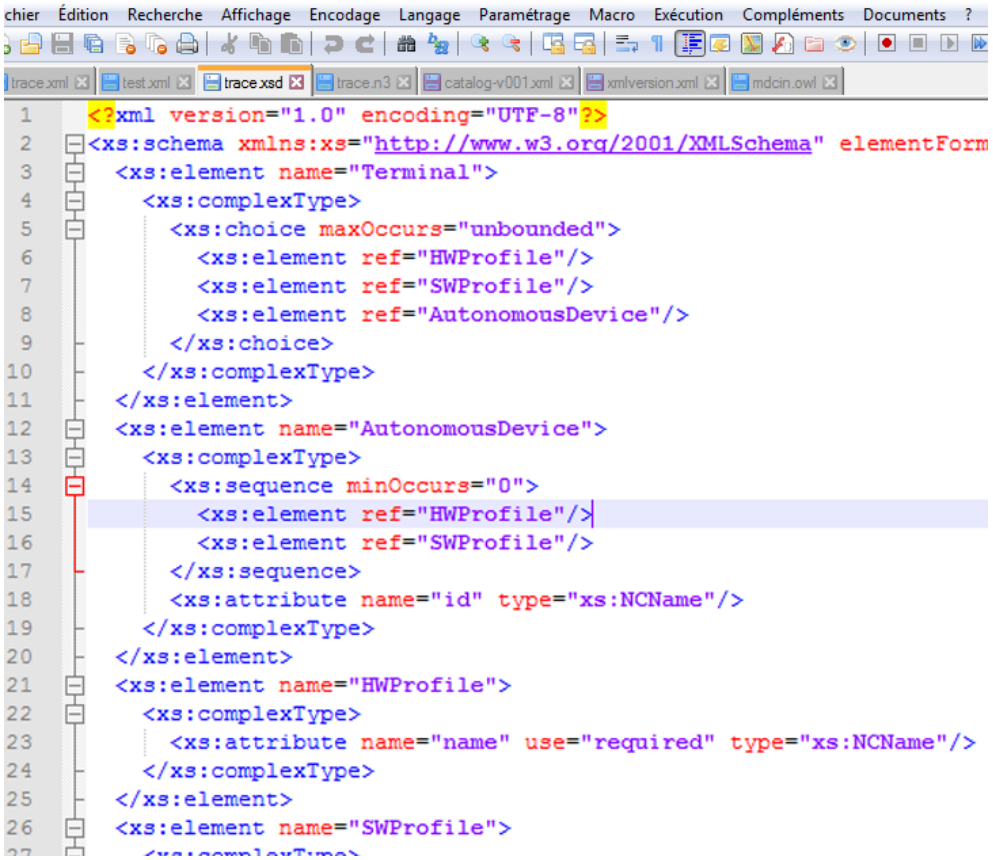

FIGURE 4.15 – Extrait d'une représentation XML-Schema

générée par XSOM est utilisée comme entrée pour le mappage XSD2OWL(RDFs/OWL) (cf figure4.16). L'étape suivante consiste à utiliser l'API Jena ainsi que apache Maven (qui est basé sur le principe de modèle des objets : POM) afin de générer les entités OWL. Le fichier OWL, produit comme trace, peut évidemment être chargé dans un éditeur OWL tels que Protégé-OWL, qui permet de nous donner la représentation OWL/RDF de la traçabilité du contexte pour une situation donnée d'utilisation de contexte (cf figure 4.17.

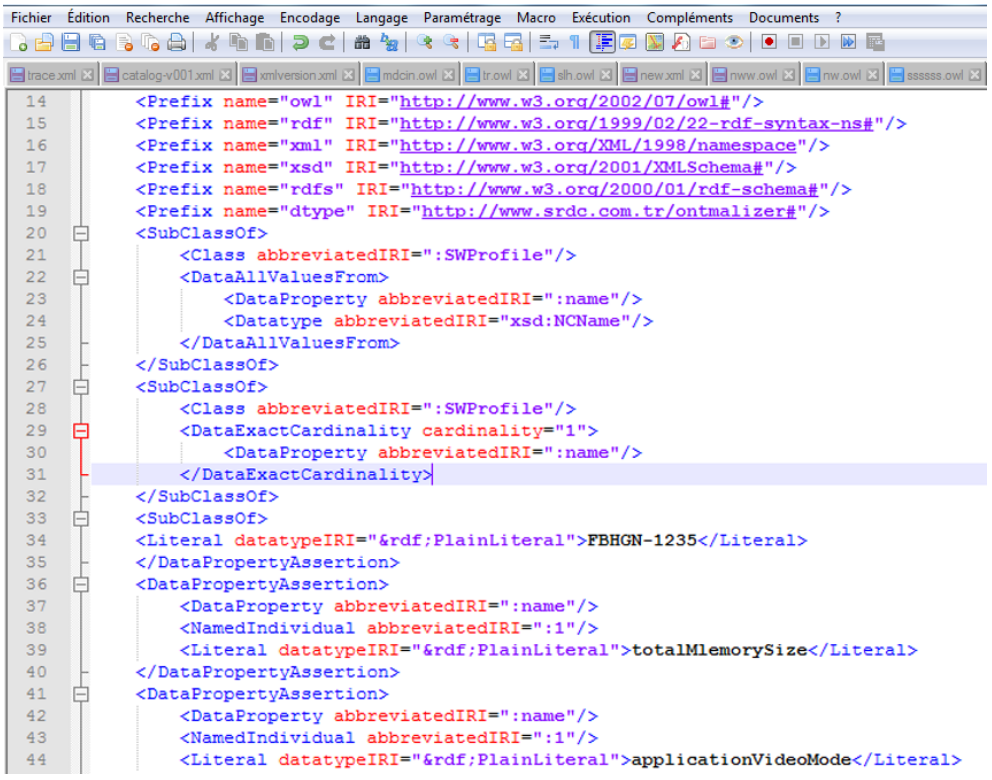

FIGURE 4.16 – Extrait d'un code OWL

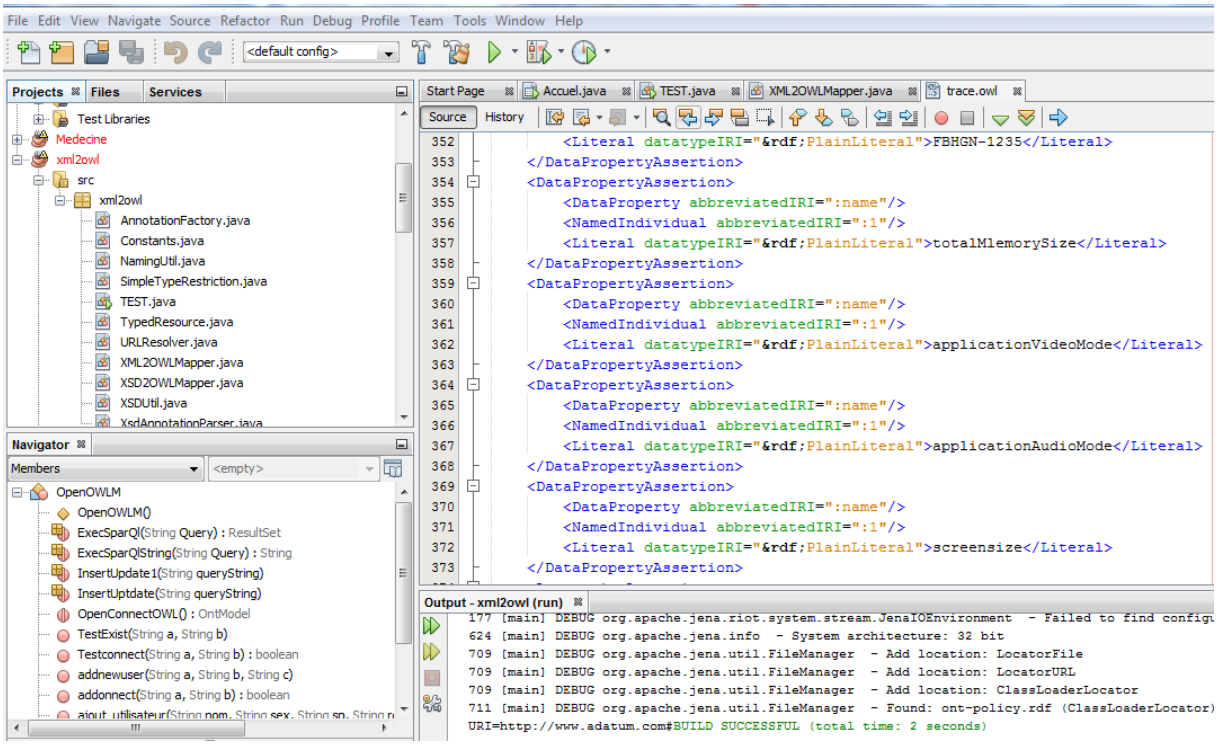

FIGURE 4.17 – Édition dans Protégé de la traçabilité de contexte dans COALA

Pour décrire le processus de modélisation de la sensibilité au contexte de notre cas d'étude, nous avons choisi de prendre en exemple, les deux classes OWL liées au terminal pour décrire sa modélisation et son utilisation dans le processus d'adaptation :

$traceability : HWProfile$ et $traceability : SWProfile$ qui ont la propriété $object : hasAProfile$.

4.4.3/ ADAPTATION DE FORMES D'AFFICHAGES PAR LA GÉNÉRATION DES INTERFACES DES SERVICES ADAPTÉS DANS COALA

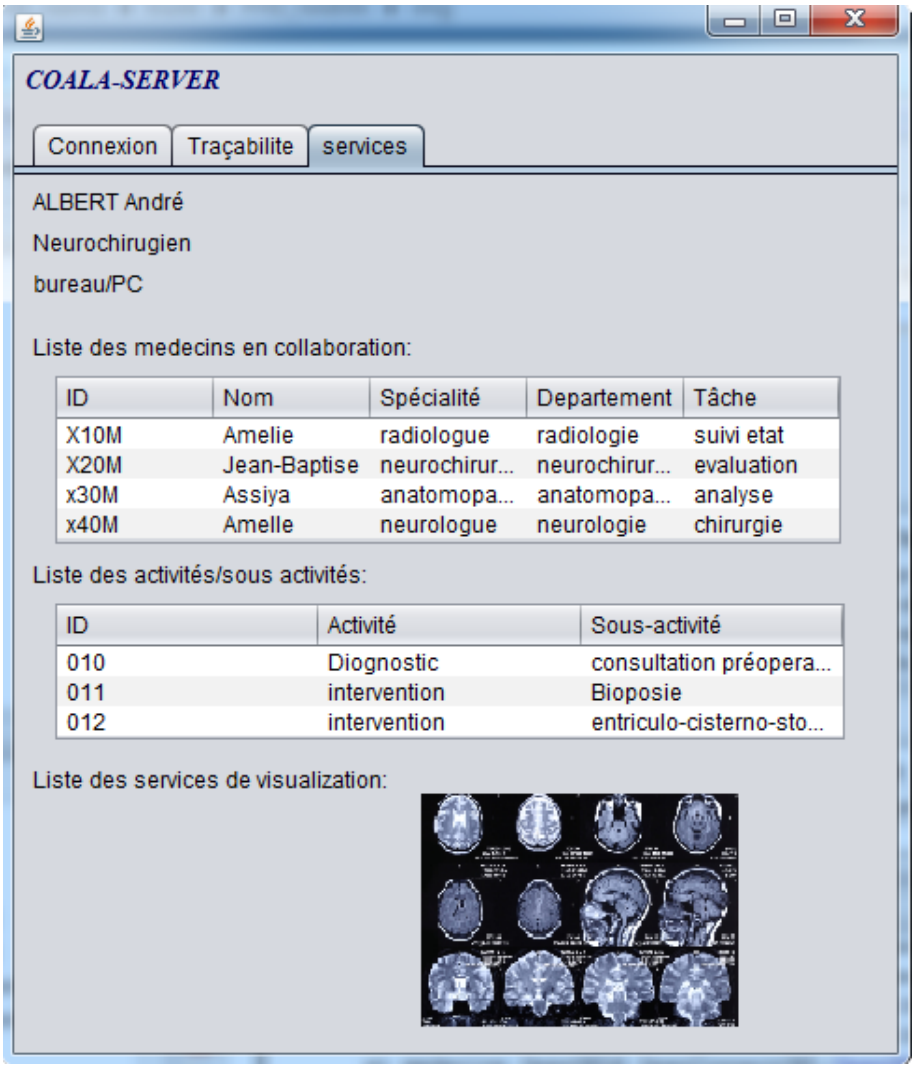

FIGURE 4.18 – La fenêtre coala d'affichage des services adaptés

Pour l'adaptation, notre étude de cas nous a permis de définir les besoins de médecin, en termes de formes d'affichage, de son dispositif utilisé et selon ses différentes situations contextuelles, lors de suivi de dossier de son patiente Josiane : SC_i : En effet, les services utiles au médecin traitant pour son suivi comportent entre autres la liste suivante :

- Un dossier d'état de patient ;
- La description du traitement prescrit ;

- La description de chaque acte effectué par un spécialiste ;
- liste des images que permet de voir l'évolution de la tumeur depuis son apparition ;
- Des graphiques de suivi du dossier (températures, poids,...).

Nous supposons que le médecin peut revenir sur le dossier de josiane à tout moment et dans divers déplacement (bureau, salle de réunion, couloir, ...) et via divers dispositifs (PC, smartphone, ...)

Une adaptation de comportement de l'application sur laquelle il travaille ce médecin, concerne la sélection des services et des formes d'affichages, doit être lui fournit en fonction de ses situation contextuelles courantes. L'adaptation devra être automatiquement réalisée grâce aux traces qui ont été récupérés dans la CVCO en fonction de son déplacement, de sa localisation et en fonction de terminal utilisé.

La taille de l'écran utilisé par le médecin à un moment donnée et sa localisation récupérée dans la CVCO est également présente la source de l'adaptation (formes d'affichage sur l'interface de l'application) ainsi que la source de la modification des formes d'adaptation en fonction de son déplacement. Au niveau de présentation du dossier de josiane, le dossier peut être afficher dans une fenêtre avec une présentation graphique, textuelle et vidéo le moment que le médecin est se connecter depuis son PC (caractéristique de terminal : grand écran), ainsi qu'il est installé dans son bureau (une forte bande passante) (cf figure 4.18).. Comme le médecin est mobile, il va suivre le dossier de patiente même en dehors de son pc (couloir par exemple), donc dans ce cas-là, il doit utiliser son smartphone : une nouvelle situation de contexte est générée alors : le médecin utilise un petit écran de son smartphone, et en dehors de son office (nouvelle localisation : couloir). L'application pour lui adapté, elle va lui fournir une autre forme d'affichage pour la présentation de dossier de patiente.

Afin de valider notre plateforme d'adaptation au contexte d'utilisation, nous avons développé notre prototype des interfaces d'interaction pour le domaine médical de tumeur de cerveau.

Lors de l'utilisation de l'application, le médecin va se connecter à l'application en envoyant automatiquement les informations qu'ils lui concernent ainsi que les propriétés du système qu'il l'utilise (hétérogénéité des terminaux). Dans un premier lieu, une carte de visite contextuelle a été générée qu'elle contienne l'ensemble des informations collectées suite à sa dernière connexion. Toutes les informations dynamiques liées à ses situations courantes vont être récupérées pour maintenir la cohérence et forcer l'adaptation. La carte permet donc de récupérer et fournir toutes les informations nécessaires pour adapter les formes d'affichage correspond. Des nouvelles situations contextuelles seront intégrées dans le modèle sémantique de contexte déjà définis dans l'ontologie de traçabilité de contexte. Des requêtes sparql seront appliquées afin de permet l'adaptation.

Dans notre application interactive, afin de pouvoir gérer l'interface utilisateur, qui permet d'assurer l'interaction avec les services de l'application et l'adaptation à son contexte d'utilisation, nous avons proposé une modélisation de l'ensemble des interfaces susceptibles d'être utilisées par le système d'adaptation COALA au profil de staff du service chirurgical. En effet, afin que l'utilisateur (médecin, chirurgien, généraliste, ...) puisse accéder à des services adaptés à son contexte d'utilisation, la plateforme *COALA* associe un ensemble de services, de fenêtres graphiques permettant d'assurer l'interaction avec ces services. En effet, les données demandées par l'utilisateur de l'application pervasive sont présentées sous forme de listes dans une interface graphique qui dépend

tout d'abord de l'utilisateur : dans notre cas c'estle médecin neurologue (son rôle, sa tâche, ses activités et sous-activités, ...). À ce stade, COALA vise à adapter l'interface de médecin selon les exigences courantes d'utilisation de contexte.

Une fois l'utilisateur connecté sur sa session, l'étape suivante consiste à spécifier pour chaque élément du contexte, l'ensemble des valeurs qui constituent les traces de l'utilisateur de l'application lors de son entrée sur sa session (cf figure 4.12). Cette interface permet de gérer toutes les configurations possibles (comme l'ajout par exemple) du contexte en se basant sur différentes valeurs et traces des éléments de contexte (cf figure 4.18).

Nous avons élaboré les différentes interfaces de traçabilité comme étant des interfaces de configuration qui permettent de gérer la traçabilité de contexte fournie dès que l'utilisateur se connecte sur sa session (cf figure 4.13).

Les différents outils de communication et d'interaction utilisent la technologie de l'environnement mobile pervasive. L'utilisateur peut être connecté sur son poste à partir de différents terminaux mobiles ou non et est susceptible de se déplacer au sein de l'hôpital (son bureau, le couloir, la salle d'examens , ou encore le bloc opératoire, . . .). Nous avons créé pour cela des interfaces utilisateurs qui permettent l'interaction avec les services de l'application pervasive et qui s'adaptent à son contexte d'utilisation (cf figure 4.18).

De même, le médecin traitant qui souhaite suivre l'état du patient a besoin d'accéder à plusieurs fonctionnalités dans l'application (comme les dossiers d'analyse, la description du traitement prescrit, les images d'échographies, la température, le poids, ...). Et cet utilisateur souhaite que l'application lui offre des services adaptés ainsi que l'accès à des services de collaboration avec le reste des membres du staff médical(coopération).

Synthèse du chapitre

Dans ce quatrième chapitre de description et d'implémentation de notre plateforme COALA, nous avons achevé ce document en présentant le prototype réalisé afin de valider notre approche d'adaptation à une sensibilité au contexte.

Nous avons testé notre approche en déployant une application médicale pour un service hospitalier permettant le suivi de tumeur du cerveau. Notre travail d'implémentation, décrit dans ce chapitre, a été effectué sur un cas d'usage clinique en vue d'arriver au développement du système COALA.

De ce fait, nous avons détaillé la conception et la réalisation technique de ce prototype. Les services proposés par cette application font l'objet d'une exploitation à travers les différents terminaux utilisés dans une situation contextuelle donnée. Ainsi, notre application est accessible depuis les différentes localisations annoncées par l'utilisateurs.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

CONCLUSION GÉNÉRALE

Les travaux de cette Thèse s'inscrivent dans le domaine de la sensibilité au contexte et plus particulièrement des nouvelles attentes en termes de personnalisation contextuelle des applications interactives pervasives. Plus précisément, le contexte n'est plus un modèle prédéfini au moment de la conception du système des application mais plutôt une description de situations contextuelles courantes susceptibles d'être changées au fur et à mesure des changements de contexte d'utilisations des utilisateurs.

Aujourd'hui, concevoir des applications interactives offrant des services sensibles aux situations contextuelles d'utilisation exige un système puissant doté de capacités permettant de choisir, dans une situation contextuelle courante et dynamique, les meilleures actions d'adaptation. Ces choix seront guidés par la détermination des paramètres de contexte qu'il faut prendre en compte pour une situation contextuelle particulière, pour la conception des services adaptatifs.

Ces réflexions ont conduit les recherches dans ce domaine, et ainsi nous ont amenés à proposer un ensemble de techniques et de mécanismes permettant l'analyse et l'exploitation des situations contextuelles courantes et dynamiques pour la modélisation et l'adaptation des services offerts par les application interactives pervasives.

La première partie de ce document a permis de présenter notre état d'art sur le domaine de la sensibilité au contexte. En effet, après notre étude des travaux existants dans le domaine de la sensibilité au contexte, nous avons constaté que la prise en compte du contexte utilisateur a un grand intérêt dans l'utilisation des ontologies pour la modélisation et la gestion du contexte. En revanche, nous avons observé l'absence d'une approche générique et complète qui permette de gérer et de modéliser le contexte d'utilisation afin d'adapter les services fournis par les applications pervasives. La plupart des recherches et des contributions existantes présentent des approches de sensibilité et d'adaptation de contexte statiques et spécifiques à un élément de contexte particulier (la sensibilité au terminal de l'utilisateur par exemple).

Ainsi, le premier chapitre de l'état de l'art a permis de montrer une étude générale de l'informatique sensible au contexte. Nous avons abordé les différentes approches de la modélisation et de l'adaptation dans le domaine de la sensibilité au contexte afin de mettre en évidence les défis et les limites de ces différents travaux. Le chapitre suivant de l'état de l'art a permis de décrire notre classification des différentes approches citées au chapitre précédent (chapitre1) à travers une étude comparative entre ces approches. Cette dernière permet de mettre en valeur le modèle de sensibilité au contexte(PIVon de R. HERVAS) à base d'ontologie, qui a été retenu comme référence pour notre travail.

La deuxième partie du document constitue notre contribution : la proposition d'un mécanisme qui permet de fournir les traces de contexte dans une situation courante d'utilisation de contexte (carte visite contextuelle : *CVCO*). Ce mécanisme permet d'alimenter notre ontologie de traçabilité. Cette ontologie, avec l'ontologie générale de contexte sont au cœur de notre nouvelle plateforme *COALA* (COntext Adaptation Platform) ainsi que de son implémentation.

La démarche que nous avons suivie pour cela commence tout d'abord par le développement de nos ontologies générales de contexte qui définissent les concepts manipulés par les sources de contexte et leurs relations. Cette proposition permet de développer un modèle sémantique général de contexte pour exprimer les éléments de contexte. Ce choix est motivé par l'intention de rassembler les différentes informations liées aux éléments de contexte et de faciliter leur partage. Nous avons développé par la suite notre ontologie de traçabilité de contexte. Nous avons adopté, dans ce cadre, un mécanisme appelé la carte de visite contextuelle (*CVCO*) qui permet de fournir la traçabilité (les paramètres, collectés en trace, d'une situation contextuelle au cours de l'utilisation).

Une proposition pour résoudre une partie des problèmes traités dans l'état de l'art a été mise en place dans notre travail de recherche. En effet, le chapitre suivant est consacré à la présentation de notre nouvelle plateforme d'adaptation au contexte *COALA*. Les situations contextuelles sont instantanées et dynamiques dans les environnements pervasifs. Le développement de notre ontologie de traçabilité, en récupérant les traces d'une situation d'utilisation de contexte, a permis le développement de notre nouvelle plateforme d'adaptation en fonction de la traçabilité de contexte. Une évaluation de la proposition, grâce à des expérimentations et implémentations des scénarios d'utilisation, nous donne une suite de résultats qui sont très encourageants. En effet, *COALA* est très novatrice car elle fournit un traitement de contexte d'utilisation se basant sur la traçabilité des situations d'utilisation tout en respectant les différentes contraintes liées aux changements dynamiques des situation d'utilisation. La nouveauté de *COALA* est déterminante au niveau de :

- la définition d'une stratégie d'adaptation générique de contexte qui permet de mettre en corrélation la traçabilité de situation courante de contexte et les services adaptés,
- l'offre de services adaptés selon les traces courantes des situations d'utilisations,
- l'évolutivité de la stratégie d'adaptation puisqu'elle permet de prendre en considération l'aspect dynamique de changement de contexte et donc la proposition des nouvelles actions d'adaptation.

PERSPECTIVES

Les travaux de cette thèse nous ont permis de proposer la nouvelle plateforme *COALA). Plusieurs questions concernant ces travaux ont été soulevées dans ce manuscrit. Aussi, de nombreux travaux, qui mériteraient une étude approfondie, nous paraissent intéressants afin de poursuivre dans cette voix.*

COMPOSANT INTELLIGENT D'*awareness*

En effet, à très court terme, notre objectif a été orienté plutôt vers la récupération des situations contextuelles courantes d'utilisation de contexte, à travers des représentations des données XML simples, puis vers la cartographie pour extraire des connaissances utiles dans la phase d'adaptation.

À plus long terme, nous souhaitons mettre à disposition de notre plateforme, un composant intelligent dont le fonctionnement doit être conscient des changements des situations de contexte qui se produisent dans l'environnement pervasif. Il conviendrait donc d'introduire ce nouveau composant permettant une exploration intelligente et continue des différentes situations contextuelles dynamiques. Par ailleurs, si une carte de visite venait à être récupérée, et qu'elle puisse contenir des situations similaires, il serait alors nécessaire pour ce composant intelligent d'effectuer un filtrage lors du changement de contexte, afin de limiter tous les problèmes de redondances.

VERS ADAPTATION RÉELLEMENT PERSONNALISÉE

Nous souhaitons également rendre notre prototype générique pour s'adapter aux différents scénarios d'utilisation. Nous envisageons par cela d'intégrer l'utilisateur dans le processus d'adaptation pour qu'il puisse modifier des décisions déjà prises par le système de manière automatique par notre plateforme. Un modèle de traçabilité locale plus personnalisé pourrait également être proposés en se basant sur le vécu et les intentions des utilisateurs en termes d'adaptation.

Ce modèle sera susceptible d'offrir un rapport entre la situation courante d'utilisation de contexte et le meilleur service d'adaptation d'utilisateur confirmé, par exemple, par le calcul d'un taux de satisfaction *de l'utilisateur.*

INTEROPÉRABILITÉ DE NOS ONTOLOGIES

La base de connaissance fournie dans COALA *est indispensable dans le processus d'adaptation aux éléments de contexte. En effet, ces ontologies fournissent les informations nécessaires qui permettent une meilleure adaptation au contexte d'utilisation. Une future recherche pourrait consister à développer un module dans notre plateforme permettant de mettre en communication et en interaction les différentes ontologies développées dans ce travail. Il s'agira donc de gérer la communication entre l'ontologie de traçabilité (ontologie pour les situations contextuelles courantes) et l'ontologie de contexte (ontologie générale de contexte).*

MA BIBLIOGRAPHIE PERSONNELLE

ARTICLE EN REVUE RÉFÉRENCÉE

[Amr16] *Amri, Hédi and Khalfallah, Ali and Gargouri, Malek and Nebhani, Naima and Lapayre, Jean-Christophe and Bouhlel, MS. Medical Image compression approach based on image resizing, digital watermarking and lossless compression. Journal of Signal Processing Systems (J SIGNAL PROCESS SYS, ISIWEB Impact Factor : 0,6). Accepted, to appear.*
quatrième auteur : issu d'une collaboration dans le domaine de la Télémedicine

PUBLICATIONS EN CONFÉRENCE AVEC COMITÉ DE LECTURE D'AUDIENCE INTERNATIONALE, DONT LES ACTES SONT PUBLIÉS

[Neb16] *Nebhani, Naima and Lapayre, Jean-Christophe and Bouhlel, MS. Ontology Traceability for the Adaptation of Services in Pervasive Environment. Proceedings of the IEEE SMC 2016, The 2016 IEEE International Conference on Systems, Man, and Cybernetics (classée B dans Core). Pages 182-261 , Budapest, Hongrie, Octobre 2016.*
Auteur principal : contribution de Thèse

PUBLICATIONS EN CONFÉRENCE AVEC COMITÉ DE LECTURE D'AUDIENCE NATIONALE, DONT LES ACTES SONT PUBLIÉS

[Neb14] *Nebhani, Naima and Lapayre, Jean-Christophe and Bouhlel, MS. The paradigm Model and Modality for Ambient intelligence interaction, International Workshop Interaction Humain-Machine and Imagerie. Le second Workshop sur la thématique de l'Interaction Homme-Machine et Imagerie (IHMIM). (IHMIM'14), May 3-6, 2014, Hammamet, Tunisia.*
auteur principal

[Neb13] *Nebhani, Naima and Lapayre, Jean-Christophe and Bouhlel, MS. AmUI : Un nouveau Modèle d'Interaction Ambiante à base de KUP. The 6th International Conferences : Sciences of Electronics, Technologies of Information and Telecommunications SETIT 2012. 21-24 Mars 2012, Sousse, TUNISIE.*
auteur principal

BIBLIOGRAPHIE

[Abo99] *Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In* Handheld and ubiquitous computing*, pages 304–307. Springer, 1999*.

[Ach12] *Achour Fatima, Jedidi Anis, and Gagouri Faiez. The generic model for pervasive information system. In* Journal of Emerging Technologies in Web Intelligence*, pages 343–351, Nov 2012.*

[Agost00] *Thomas C Agoston, Tatsuro Ueda, and Yukari Nishimura. Pervasive computing in a networked world. In* Proceedings of INET*, pages 3–5, 2000.*

[Agre01] *Philip E Agre. Changing places : contexts of awareness in computing.* Human-computer interaction*, 16(2) :177–192, 2001.*

[Amar06] *IEEE.* Modélisation d'informations contextuelles pour des agents mobiles sensibles au contexte*, volume 5, 2006.*

[Baad03] *Franz Baader.* The description logic handbook : Theory, implementation and applications*. Cambridge university press, 2003.*

[Bal07] *Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems.* International Journal of Ad Hoc and Ubiquitous Computing*, 2(4) :263–277, 2007.*

[Bau03] *J. Bauer. Identification and modeling of context for different information in air t raffic. Master's thesis, Université d'eléctronique et d'informatique de Berlin, Mars 2003*.

[Berhe05] *Girma Berhe, Lionel Brunie, and J-M Pierson. Distributed content adaptation for pervasive systems. In* International Conference on Information Technology : Coding and Computing (ITCC'05)-Volume II*, volume 2, pages 234–241. IEEE, 2005.*

[Birn97] *Joel Birnbaum. Pervasive information systems.* Communications of the ACM*, 40(2) :40–42, 1997.*

[Bohri05] *Hannes Bohring and Sören Auer. Mapping xml to owl ontologies.* Leipziger Informatik-Tage*, 72 :147–156, 2005.*

[Bors97] *Willem Nico Borst.* Construction of engineering ontologies for knowledge sharing and reuse*. Universiteit Twente, 1997.*

[Bouz97] *Bruno Bouzy and Tristan Cazenave. Using the object oriented paradigm to model context in computer go. In* Proceedings of the first international and interdisciplinary conference on modeling and using context*, pages 279–289, 1997.*

[Bouas09] *Ismael Bouassida Rodriguez, German Sancho, Thierry Villemur, Said Tazi, and Khalil Drira. A model-driven adaptive approach for collaborative ubiquitous sys-*

tems. In Proceedings of the 3rd workshop on Agent-oriented software engineering challenges for ubiquitous and pervasive computing*, pages 15–20. ACM, 2009.*

[Brac85] Ronald J Brachman and James G Schmolze. An overview of the kl-one knowledge representation system. Cognitive science*, 9(2) :171–216, 1985.*

[Brad02] Neil Bradley. The XML companion*. Addison-Wesley Professional, 2002.*

[Bro97] Peter J Brown, John D Bovey, and Xian Chen. Context-aware applications : from the laboratory to the marketplace. Personal Communications, IEEE*, 4(5) :58–64, 1997.*

[Broo03] Kevin Brooks. The context quintet : narrative elements applied to context awareness. In Human Computer Interaction International Proceedings*, volume 2003. Citeseer, 2003.*

[Buch03] Thomas Buchholz, Axel Küpper, and Michael Schiffers. Quality of context : What it is and why we need it. In Proceedings of the workshop of the HP OpenView University Association*, volume 3, 2003.*

[Cassa09] Cyril Cassagnes, Philippe Roose, and Marc Dalmau. Kalimucho : software architecture for limited mobile devices. ACM SIGBED Review*, 6(3) :12, 2009.*

[Chen04] Harry Chen, Tim Finin, and Anupam Joshi. A context broker for building smart meeting rooms*. Defense Technical Information Center, 2004.*

[Cha04] Dan Chalmers, Naranker Dulay, and Morris Sloman. Towards reasoning about context in the presence of uncertainty. In Proceedings of Workshop on Advanced Context Modelling, Reasoning And Management at UbiComp 2004*, pages 1–33, 2004.*

[Cha07] Tarak Chaari. Adaptation d'applications pervasives dans des environnements multi-contextes*. PhD thesis, institut national des sciences appliquées de Lyon, 2007.*

[Chan11] Sumanth Channabasappa. A framework for session initiation protocol user agent profile delivery. International Journal of Ad Hoc and Ubiquitous Computing*, 2011.*

[Chen03] Harry Chen, Tim Finin, and Anupam Joshi. An ontology for context-aware pervasive computing environments. The Knowledge Engineering Review*, 18(03) :197–207, 2003.*

[Chen05] Harry Chen, Filip Perich, Tim Finin, and Anupam Joshi. Soupa : Standard ontology for ubiquitous and pervasive applications. In Mobile and Ubiquitous Systems : Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*, pages 258–267. IEEE, 2005.*

[Clar99] James Clark, Steve DeRose, et al. Xml path language (xpath) version 1.0, 1999.

[Corc03] Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Methodologies, tools and languages for building ontologies. where is their meeting point ? Data & knowledge engineering*, 46(1) :41–64, 2003.*

[Cout87] Joelle Coutaz. Pac, an object oriented model for dialog design. In Proceedings Interact*, volume 87, pages 431–436, 1987.*

[Cout95] Joëlle Coutaz, James L Crowley, Simon Dobson, and David Garlan. Context is key. Communications of the ACM*, 48(3) :49–53, 2005.*

[Davi97] *Randall Davis. Interactive transfer of expertise : Acquisition of new inference rules.* Artificial intelligence*, 12(2) :121–157, 1979.*

[Davi05] *Pierre-Charles David.* Développement de composants Fractal adaptatifs : un langage dédié à l'aspect d'adaptation*. PhD thesis, Université de Nantes, 2005.*

[De06] *Diego López De Ipiña, Juan Ignacio Vázquez, Daniel García, Javier Fernández, Iván García, David Sainz, and Aitor Almeida. Emi2lets : A reflective framework for enabling ami.* J. UCS*, 12(3) :297–314, 2006.*

[Deck00] *S. Decker. The semantic web : The roles of xml and rdf.* Internet Computing, IEEE*, 4(5) :63–73, 2000.*

[Dey01] *Anind K Dey, Gregory D Abowd, and Daniel Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications.* Human-computer interaction*, 16(2) :97–166, 2001.*

[Dey05] *Anind K Dey and Jennifer Mankoff. Designing mediation for context-aware applications.* ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1) :53–80, 2005.*

[El11] *Yassine El Ghayam and Mohammed Erradi. Distributed context management in collaborative environment. In* New Technologies of Distributed Systems (NOTERE), 2011 11th Annual International Conference on*, pages 1–8. IEEE, 2011.*

[Fan09] *C. L. Fankam, D. Bellatreche, Y. A. Hondjack, and G. Pierra Ameur. Conception de bases de données à partir d'ontologies de domaine. In* Technique et Science Informatique, TSI*, volume 28, page 1233–1261, 2009.*

[Fuch05] *Florian Fuchs, Iris Hochstatter, Michael Krause, and Michael Berger. A metamodel approach to context information. In* null*, pages 8–14. IEEE, 2005.*

[Garc08] *Manuel García-Herranz, Pablo A Haya, Abraham Esquivel, Germán Montoro, and Xavier Alamán. Easing the smart home : Semi-automatic adaptation in perceptive environments.* J. UCS*, 14(9) :1529–1544, 2008.*

[Ghaw09] *Raji Ghawi and Nadine Cullot. Building ontologies from xml data sources. In* DEXA Workshops*, pages 480–484, 2009.*

[Grub93] *Thomas R Gruber. A translation approach to portable ontology specifications.* Knowledge acquisition*, 5(2) :199–220, 1993.*

[Gu04] *Tao Gu, Xiao Hang Wang, Hung Keng Pung, and Da Qing Zhang. An ontology-based context model in intelligent environments. In* Proceedings of communication networks and distributed systems modeling and simulation conference*, volume 2004, pages 270–275, 2004.*

[Guar95] *N. Guarino. Formal ontology, conceptual analysis and knowledge representation. In* International Journal of Human-Computer Studies*, volume 43, page 625–640, 2000.*

[Guar00] *Nicola Guarino and Christopher Welty. Identity, unity, and individuality : Towards a formal toolkit for ontological analysis. In* ECAI*, volume 2000, pages 219–223. Citeseer, 2000.*

[Gui12] *Valérian Guivarch, Valérie Camps, and André Péninou. Context awareness in ambient systems by an adaptive multi-agent approach. In* Ambient intelligence*, pages 129–144. Springer, 2012.*

[Gui13] *Valérian Guivarch, Valérie Camps, André Péninou, and Simon Stuker. Dynamic filtering of useless data in an adaptive multi-agent system : Evaluation in the ambient domain. In* Advances on Practical Applications of Agents and Multi-Agent Systems*, pages 110–121. Springer, 2013.*

[Har89] *H Rex Hartson and Deborah Hix. Toward empirically derived methodologies and tools for human-computer interface development.* International Journal of Man-Machine Studies*, 31(4) :477–494, 1989.*

[Harr06] *Hamid Harroud and Ahmed Karmouch. Implicit context-sensitive mobile computing using semantic policies. In* Autonomic Networking*, pages 188–200. Springer, 2006.*

[Hef04] *Jeff Heflin. Owl web ontology language-use cases and requirements.* W3C Recommendation*, 10(10) :1–12, 2004.*

[Held02] *Albert Held, Sven Buchholz, and Alexander Schill. Modeling of context information for pervasive computing applications. In* Procceding of the World Multiconference on Systemics, Cybernetics and Informatics*, pages 102–120. Springer, 2002.*

[Herv08] *Ramón Hervás, Salvador W Nava, Gabriel Chavira, Vladimir Villarreal, and José Bravo. Pivita : taxonomy for displaying information in pervasive and collaborative environments. In* 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, pages 293–301. Springer, 2009.*

[Herv09] *Ramón Hervás, Salvador W Nava, Jesús Fontecha, Gregorio Casero, Javier Laguna, and José Bravo. Exploring context semantics for proactive cooperative visualization. In* Cooperative Design, Visualization, and Engineering*, pages 52–55. Springer, 2009.*

[Herv10] *Ramón Hervás, José Bravo, and Jesús Fontecha. A context model based on ontological languages : a proposal for information visualization.* J. UCS*, 16(12) :1539–1555, 2010.*

[Herv11] *Ramón Hervás and José Bravo. Coiva : context-aware and ontology-powered information visualization architecture.* Software : Practice and Experience*, 41(4) :403–426, 2011.*

[Herv13] *Ramón Hervás, José Bravo, Jesús Fontecha, and Vladimir Villarreal. Achieving adaptive augmented reality through ontological context-awareness applied to aal scenarios.* J. UCS*, 19(9) :1334–1349, 2013.*

[Hofe03] *Thomas Hofer, Wieland Schwinger, Mario Pichler, Gerhard Leonhartsberger, Josef Altmann, and Werner Retschitzegger. Context-awareness on mobile devices-the hydrogen approach. In* System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2003.*

[Ind03] *Jadwiga Indulska, Ricky Robinson, Andry Rakotonirainy, and Karen Henricksen. Experiences in using cc/pp in context-aware systems. In* Mobile Data Management*, pages 247–261. Springer, 2003.*

[Jac06] *Christophe Jacquet, Yacine Bellik, and Yolaine Bourda. Kup, un modèle pour la présentation multimodale et opportuniste d'informations en situation de mobilité.* Ingénierie des systèmes d'information*, 11(5) :115–139, 2006.*

[Jac07] *Christophe Jacquet, Yacine Bellik, and Yolaine Bourda. Kup : un modèle pour la présentation opportuniste et multimodale d'informations à des utilisateurs mobiles. In* Proceedings of the 19th International Conference of the Association Francophone d'Interaction Homme-Machine*, pages 43–50. ACM, 2007.*

[Karp00] *Peter D Karp. An ontology for biological function based on molecular interactions.* Bioinformatics*, 16(3) :269–285, 2000.*

[Kays97] *D Kayser. La représentation des connaissances, hermes.* Collection Informatique*, pages 104–112, 1997.*

[Korp03] *Panu Korpipää, M Jani, Juha Kela, Esko-Juhani Malm, et al. Managing context information in mobile devices.* IEEE pervasive computing*, 3 :42–51, 2003.*

[Kras88] *Glenn E Krasner, Stephen T Pope, et al. A description of the model-view-controller user interface paradigm in the smalltalk-80 system.* Journal of object oriented programming*, 1(3) :26–49, 1988.*

[Lass99] *Ora Lassila and Ralph R Swick. Resource description framework (rdf) model and syntax specification.* Studies in health technology and informatics*, 107(0 1) :459, 1999.*

[Lind08] *DAB Lindberg, BL Humphreys, AT McCray, et al. The unified medical languagesystem-methods inf med 1993 aug ; 32 (4) : 281-91.* IMIA Yearbook*, pages 179–179, 2008.*

[Loub11] *Christine Louberry, Philippe Roose, and Marc Dalmau. Kalimucho : Plateforme d'adaptation des applications mobiles. In* NOTERE 201-Conférence Internationale sur les NOuvelles Technologies de la REpartition*, pages 83–90, 2011.*

[Maed04] *Alexander Maedche and Steffen Staab. Ontology learning. In* Handbook on ontologies*, pages 173–190. Springer, 2004.*

[Mcca93] *John McCarthy. Notes on formalizing context.* Proc. 13th Int. Joint Conf. on Artificial Intelligence*, pages 555–560, 1993.*

[Mins00] *Marvin Minsky. Commonsense-based interfaces.* Communications of the ACM*, 43(8) :66–73, 2000.*

[Mink10] *Wolfgang Minker and Tobias Heinroth. Advanced spoken dialogue management in adaptive and trusted ambient ecologies. In* Intelligent Environments (Workshops)*, pages 178–184, 2010.*

[Mira08] *Moeiz Miraoui, Chakib Tadj, and Chokri ben Amar. Context modeling and context-aware service adaptation for pervasive computing systems.* International Journal of Computer and Information Science and Engineering*, 2(3) :148–157, 2008.*

[Mora07] *Pavlos Moraitis and Nikolaos Spanoudakis. Argumentation-based agent interaction in an ambient-intelligence context.* Intelligent Systems, IEEE*, 22(6) :84–93, 2007.*

[Most04] *Ghita Kouadri Mostefaoui, Jacques Pasquier-Rocha, and Patrick Brezillon. Context-aware computing : a guide for the pervasive computing community. In* Pervasive Services, 2004. ICPS 2004. IEEE/ACS International Conference on*, pages 39–48. IEEE, 2004.*

[Nar00] *Dushyanth Narayanan, Jason Flinn, and Mahadev Satyanarayanan. Using history to improve mobile application adaptation. In* Mobile Computing Systems and Applications, 2000 Third IEEE Workshop on.*, pages 31–40. IEEE, 2000.*

[Pars04] *Bijan Parsia and Evren Sirin. Pellet : An owl dl reasoner. In* Third International Semantic Web Conference-Poster*, volume 18, 2004.*

[Pasc97] *Jason Pascoe. The stick-e note architecture : extending the interface beyond the user. In* Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 261–264. ACM, 1997.*

[Pfa12] *Günther E Pfaff.* User Interface Management Systems : Proceedings of the Workshop on User Interface Management Systems held in Seeheim, FRG, November 1–3, 1983*. Springer Science & Business Media, 2012.*

[Pito94] *Evaggelia Pitoura and Bharat Bhargava. Building information systems for mobile environments. In* Proceedings of the third international conference on Information and knowledge management*, pages 371–378. ACM, 1994.*

[Raib08] *Claudia Raibulet. Facets of adaptivity. In* Software Architecture*, pages 342–345. Springer, 2008.*

[Ree79] *Trygve Reenskaug. Models-views-controllers.* Technical note, Xerox PARC*, 32(55) :6–2, 1979.*

[Rod06] *Toni Rodrigues, Pedro Rosa, and Jorge Cardoso. Mapping xml to existing owl ontologies. In* International Conference WWW/Internet*, pages 72–77, 2006.*

[Ryan98] *Nick Ryan. Contextml : Exchanging contextual information between a mobile client and the fieldnote server.* Retrieved October*, 10 :2006–2008, 1999.*

[Ryan99] *Nick Ryan, Jason Pascoe, and David Morse. Enhanced reality fieldwork : the context aware archaeological assistant.* Bar International Series*, 750 :269–274, 1999.*

[Samu02] *Michael Samulowitz, Florian Michahelles, and Claudia Linnhoff-Popien. Capeus : An architecture for context-aware selection and execution of services. In* New developments in distributed applications and interoperable systems*, pages 23–39. Springer, 2002.*

[Schi93] *Bill N Schilit, Marvin M Theimer, and Brent B Welch. Customizing mobile applications. In* Proceedings USENIX Symposium on Mobile & Location-indendent Computing*, volume 9, pages 20–25, 1993.*

[Schi94] *Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In* Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pages 85–90. IEEE, 1994.*

[Schm99] *Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In* Handheld and ubiquitous computing*, pages 89–101. Springer, 1999.*

[Shen05] *Quan Z Sheng and Boualem Benatallah. Contextuml : a uml-based modeling language for model-driven development of context-aware web services. In* Mobile Business, 2005. ICMB 2005. International Conference on*, pages 206–212. IEEE, 2005.*

[Sowa00] *John F Sowa. Knowledge representation : logical, philosophical, and computational foundations.* Pacific Grove, CA, Brooks/Cole*, 594 :68–75, 2000.*

[Span06] *Nikolaos I Spanoudakis and Pavlos Moraitis. Agent-based architecture in an ambient intelligence context. In* Proc. 4th European Workshop Multi-Agent Systems (EUMAS06)*, pages 163–174, 2006.*

[Stud98] *Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering : principles and methods.* Data & knowledge engineering*, 25(1) :161–197, 1998.*

[Tsel04] *TH Tse and Stephen S Yau. Testing context-sensitive middleware-based software applications. In* Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, pages 458–466. IEEE, 2004.*

[Urbi09] *Aitor Urbieta, Ekain Azketa, Inma Gomez, Jorge Parra, and Nestor Arana. Bridging the gap between services and context in ubiquitous computing environments using an effect-and condition-based model. In* 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, pages 149–158. Springer, 2009.*

[Van97] *Gertjan Van Heijst, A Th Schreiber, and Bob J Wielinga. Using explicit ontologies in kbs development.* International journal of human-computer studies*, 46(2) :183–292, 1997.*

[Van02] *Eric Van der Vlist.* XML schema*. O'Reilly Media, Inc., 2002.*

[Vill14] *Vladimir Villarreal, Jesus Fontecha, Ramon Hervas, and Jose Bravo. Mobile and ubiquitous architecture for the medical control of chronic diseases through the use of intelligent devices : Using the architecture for patients with diabetes.* Future generation computer systems*, 34 :161–175, 2014.*

[Wang02] *Xiaohang Wang, Jin Song Dong, C Chin, Sanka Ravipriya Hettiarachchi, and Daqing Zhang. Semantic space : An infrastructure for smart spaces.* Computing*, 1(2) :67–74, 2002.*

[Wang04] *Xiao Hang Wang, Da Qing Zhang, Tao Gu, and Hung Keng Pung. Ontology based context modeling and reasoning using owl. In* Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 18–22. Ieee, 2004.*

[Weise93] *Mark Weiser. Some computer science issues in ubiquitous computing.* Communications of the ACM*, 36(7) :75–84, 1993.*

[Wino01] *T Winograd. Architectures for context, human-computer interaction.* L. Erlbaum Associates Inc. Hillsdale, NJ, USA*, 16 :402–419, 2001.*

[Yau01] *Stephen S Yau and Fariaz Karim. Context-sensitive middleware for real-time software in ubiquitous computing environments. In* Object-Oriented Real-Time Distributed Computing, 2001. ISORC-2001. Proceedings. Fourth IEEE International Symposium on*, pages 163–170. IEEE, 2001.*

TABLE DES FIGURES

LISTE DES TABLES

Résumé :

*L'étude de la littérature montre que la sensibilité au contexte est devenue un élément primordial pour la mise en place des services adaptatifs dans les applications interactives pervasives. Le contexte n'est plus un modèle préétabli et prédéfini au moment de la conception des systèmes des applications interactives mais plutôt une description dynamique des situations courantes qui peuvent être découvertes dans les données de contexte et qui peuvent changer dynamiquement en fonction des changements des exigences et des préférences des utilisateurs. Les applications pervasives doivent pouvoir s'exécuter dans différents contextes d'utilisation selon l'environnement de l'utilisateur, son profil, le terminal qu'il utilise, sa localisation, . . . Afin de répondre aux différentes exigences d'adaptation aux changements dynamiques des situations contextuelles, nous proposons des mécanismes permettant de fournir la traçabilité de contexte sous la forme de Cartes de Visite COntextuelles (*CVCO*). Ces mécanismes s'appuient sur la technique de Mapping OWL afin de donner des modèles ontologiques de traçabilité de contexte. Les différentes ontologies créées pendant ce travail sont intégrées au sein de l'architecture de notre nouvelle plateforme d'adaptation de contexte COALA (*COntext Adaptation Platform*), afin de permettre l'adaptation automatique des services offerts par les application interactives pervasives. Ces travaux ont été réalisés et appliqués dans le cadre d'une thèse en co-tutelle entre la Tunisie et France.*

Mots-clés : *Sensibilité au contexte, Adaptation des services, Traçabilité de contexte, Environnement pervasif, Applications interactives*

Abstract:

The study of the literature proves that context-awareness has become a key element for the implementation of adaptive services in pervasive interactive applications. In fact, the present work focuses on the context of modeling issues for the adaptation of services provided by the invective pervasive applications to the sensitivity of context situations. The context is no longer a pre-established and predefined template when designing interactive applications systems but rather a dynamic description of the common situations that can be discovered in the context data and can dynamically be changed when changing user's requirements and preferences. The systems of pervasive applications must be available in different contexts of use according to the user's environment, the terminal being used, location . . .. The major challenge of these systems accordingly relates to the adaptation of the services offered by interactive applications to the user context. In Order to respond to different requirements of dynamic adaptation to changes in contextual situations, following the changes in usage patterns as well as the dynamics of the pervasive environment and the heterogeneity of context data sources, we propose mechanisms used to provide the traceability of the context as Virtual contextual Cards (VCOC). These works were carried out and applied as part a PhD under joint supervision between Tunisia and France.

Keywords: *Context-Awareness, Pervasie Environment, Service Adaptation*

SPIM

■ École doctorale SPIM 16 route de Gray F - 25030 Besançon cedex

■ tél. +33 (0)3 81 66 66 02 ■ **ed-spim@univ-fcomte.fr** ■ www.ed-spim.univ-fcomte.fr

UFC
UNIVERSITÉ
DE FRANCHE-COMTÉ