



**HAL**  
open science

# Déploiement d'applications d'assistance dans un environnement d'aide au handicap

Cédric Seguin

► **To cite this version:**

Cédric Seguin. Déploiement d'applications d'assistance dans un environnement d'aide au handicap. Informatique ubiquitaire. Université de Bretagne Sud, 2015. Français. NNT : 2015LORIS367 . tel-01292592

**HAL Id: tel-01292592**

**<https://theses.hal.science/tel-01292592>**

Submitted on 23 Mar 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**THESE / UNIVERSITE DE BRETAGNE-SUD**  
*sous le sceau de l'Université européenne de Bretagne*  
pour obtenir le titre de  
**DOCTEUR DE L'UNIVERSITE DE BRETAGNE-SUD**  
*Mention : STICC*  
Ecole doctorale SICMA

présentée par

**Cédric SEGUIN**  
Préparée au Laboratoire Lab-STICC,  
Lorient, France

# Déploiement de services d'assistance dans un environnement d'aide au handicap

**Thèse soutenue le : 30 avril 2015**  
devant le jury composé de :

**Sylvain Giroux**  
Professeur, Université de Sherbrooke / Rapporteur

**Jean-Yves Antoine**  
Professeur, Université François-Rabelais / Rapporteur

**André Thépaut**  
Directeur d'études, Télécom Bretagne / Examineur

**Alexandre Abellard**  
Maître de conférences, Université de Toulon-Var / Examineur

**Willy Allègre**  
Docteur, Ingénieur, CMRRF de Kerpape / Membre invité

**Florent De Lamotte**  
Maître de conférences, Université de Bretagne Sud  
/ Co-directeur de thèse

**Jean-Luc Philippe**  
Professeur, Université de Bretagne Sud  
/ Directeur de thèse



À mes parents, À ma sœur.



## Remerciements

Il m'est impossible de commencer ces remerciements sans citer mes encadrants, mon Directeur de thèse Jean-Luc Philippe et mon encadrant de thèse Florent Frizon de Lamotte. Jean-Luc, Florent, cette thèse n'a pas été un long fleuve tranquille, mais aurais-je autant appris si cela avait été le cas ? Grâce à votre soutien et vos conseils nous avons formé une équipe qui a porté ce projet jusqu'au bout. Je tiens sincèrement à vous remercier de m'avoir accompagné pendant ces quelques années, d'avoir dispensés vos conseils pendant nos réunions et d'avoir toujours cru en moi.

Parmi les personnes importantes qui ont contribué à la réussite de mes travaux, il m'apparaît comme une évidence de remercier Sylvain Giroux. Sylvain, tu m'as accueilli pendant 8 mois dans ton laboratoire et j'ai pendant cette période découvert le Professeur et l'homme. Cette expérience au Domus a sonné comme une révélation, je t'en remercie profondément. Je profite de ce moment pour bien évidemment étendre mes remerciements à tous les membres du Domus que j'ai pu rencontrer, une équipe accueillante et dynamique.

Je tiens à remercier les membres de mon jury d'avoir acceptés de rapporter et examiner mes travaux de thèse. Merci aux Professeurs Jean-Yves Antoine et Sylvain Giroux, les rapporteurs qui ont su trouver la justesse dans l'analyse de mes travaux. Merci aux examinateurs André Thépaut, qui a accepté la charge de président du jury, et Alexandre Abellard, mais aussi Willy Allègre, membre invité à la soutenance. Merci pour ces échanges riches et pertinents que nous avons eu pendant la soutenance.

Enfin, il est évident que la thèse bien que le travail d'une personne, d'une équipe, s'inscrit dans un contexte plus large, celui de son laboratoire. À ce titre, je voudrais particulièrement remercier Guy Gogniat qui m'a accueilli au sein du lab-STICC, tous les collègues permanents ou non, toujours ici ou ailleurs, les collègues du bureau 204 au centre de recherche et de la salle 305 à l'ENSIBS, vous êtes bien trop nombreux mais j'ai une pensée pour chacun d'entre vous. Je ne pourrais pas conclure les remerciements des membres du Lab-STICC sans avoir un mot pour nos secrétaires. Promis ce n'est pas sous la torture, mais avec toute la sincérité qu'il se doit que je tiens à remercier Virginie et Florence, qui auront presque été des mamans. Certes, de très jeunes mamans mais d'incroyables soutiens aussi ! Merci les Thérèse. Une pensée toute particulière pour mon ami Vianney. Ça fait un moment qu'on se suit, se soutient, se supporte. Certains diront qu'on est un vrai petit couple, je répondrai juste : jaloux. Merci, mon pote ! Enfin, une dernière pensée pour Benoît, mon compagnon de travail, de galère, avec qui j'ai passé la plupart de mes soirées, weekend et vacances dans les couloirs vides de l'ENSIBS. On s'est quand même bien marré !



# Table des matières

<b>Introduction générale</b>	<b>5</b>
<b>1 La technologie au service du handicap</b>	<b>9</b>
1.1 Le handicap, un challenge social et économique . . . . .	11
1.1.1 Introduction générale au handicap . . . . .	11
1.1.1.1 Le modèle de développement humain . . . . .	11
1.1.1.2 Le processus de production du handicap . . . . .	11
1.1.1.3 Les grandes catégories d'incapacités . . . . .	14
1.1.2 La situation des personnes handicapées dans le monde . . . . .	14
1.1.3 Le cas des personnes âgées . . . . .	16
1.1.3.1 Démographie et géographie . . . . .	16
1.1.3.2 Une population fragile . . . . .	16
1.1.4 De l'incapacité à la dépendance, de l'hospitalisation au maintien à domicile . . . . .	17
1.2 La technologie au quotidien . . . . .	18
1.2.1 La démocratisation de la technologie . . . . .	18
1.2.2 La domotique : le paradigme du grand public . . . . .	19
1.2.3 La domotique : les spécifications de l'aide au handicap . . . . .	20
1.2.4 Les solutions domotiques d'assistance . . . . .	20
1.3 Revue des principaux habitats intelligents et de leurs solutions d'aide au handicap . . . . .	21
1.3.1 Les habitats intelligents à travers le monde . . . . .	22
1.3.2 Les solutions développées au Lab-STICC . . . . .	23
1.3.2.1 Projets sur l'habitat intelligent de l'action HAAL . . . . .	23
1.3.2.2 Projet sur l'habitat intelligent de l'équipe MOCS . . . . .	24
1.4 Positionnement et motivation . . . . .	27
<b>2 Modélisation d'environnement intelligent</b>	<b>33</b>
2.1 Introduction à la conception . . . . .	35
2.2 Modélisation de l'informatique ubiquitaire . . . . .	39
2.2.1 Du service d'aide au handicap à la tâche . . . . .	39
2.2.1.1 Définition d'un service . . . . .	39
2.2.1.2 La naissance de la notion de tâche . . . . .	40
2.2.1.3 L'offre de service d'aide au handicap . . . . .	42
2.2.1.4 Modélisation d'une tâche d'aide au handicap . . . . .	43
2.2.2 De l'équipement domotique à la ressource . . . . .	44
2.2.2.1 Définition d'une ressource . . . . .	45
2.2.2.2 L'appareillage domotique . . . . .	45
2.2.2.3 Associations de ressources . . . . .	46

2.2.2.4	Modélisation d'une ressource . . . . .	47
2.2.3	La capacité, lien entre la tâche et la ressource . . . . .	48
2.3	Ontologie de l'environnement domotique . . . . .	50
2.3.1	État de l'art et positionnement . . . . .	51
2.3.2	Introduction aux ontologies . . . . .	52
2.3.3	Notre ontologie d'environnement . . . . .	55
2.3.3.1	Construction d'un environnement domotique . . . . .	55
2.3.3.2	Héritage de capacité . . . . .	57
2.3.3.3	Les tâches pour compléter l'ontologie . . . . .	59
2.3.3.4	La recherche de ressource pour exécuter une tâche . . . . .	59
<b>3</b>	<b>Déploiement de services d'aide au handicap</b>	<b>67</b>
3.1	L'infrastructure informatique . . . . .	69
3.1.1	Le système d'exploitation : Linux . . . . .	74
3.1.2	La machine virtuelle Java . . . . .	75
3.1.3	Le Framework OSGi . . . . .	76
3.1.4	Diagrammes d'état et cycle de vie . . . . .	77
3.1.4.1	Les processus Unix . . . . .	77
3.1.4.2	Les bundles OSGi . . . . .	78
3.2	Le flot de déploiement de tâches . . . . .	79
3.2.1	Le manager d'ontologie . . . . .	82
3.2.2	Le gestionnaire de tâches . . . . .	83
3.2.2.1	Phase de décision . . . . .	84
3.2.2.2	Phase d'ordonnancement . . . . .	85
3.2.2.3	Phase d'exécution . . . . .	85
3.2.3	Le moniteur de ressources . . . . .	86
<b>4</b>	<b>Preuve de concept et résultats</b>	<b>91</b>
4.1	L'environnement d'expérimentation . . . . .	93
4.1.1	Les appartements tremplins du CMRRF de Kerpape . . . . .	94
4.1.2	La maison Domus de l'Université de Sherbrooke . . . . .	95
4.1.3	La plateforme d'expérimentation du labSTICC de Lorient . . . . .	98
4.1.3.1	La plateforme d'expérimentation originale . . . . .	98
4.1.3.2	La plateforme d'expérimentation évoluée . . . . .	100
4.2	Mise en œuvre du système d'assistance domotique . . . . .	102
4.2.1	Cas d'étude d'un patient simulé dans un appartement tremplin . . . . .	102
4.2.1.1	Le profil du patient expérimental . . . . .	102
4.2.1.2	Scénario d'étude . . . . .	103
4.2.1.3	Équipements possédés et proposés . . . . .	103
4.2.1.4	L'offre personnalisée de services . . . . .	104
4.2.2	Conception : Modélisation de l'environnement . . . . .	104
4.2.2.1	Les éléments matériels . . . . .	105
4.2.2.2	Les éléments logiciels . . . . .	106
4.2.3	Exécution : Déploiement de tâches d'aide au handicap . . . . .	111
4.2.3.1	Initialisation du système d'assistance . . . . .	111
4.2.3.2	Fonctionnement courant du système . . . . .	113
4.3	Résultats et limites de la solution proposée . . . . .	118
4.3.1	Intégration des recommandations de l'aide au handicap . . . . .	118
4.3.2	Stratégies de déploiement . . . . .	119
4.3.3	Mesure des temps de réaction . . . . .	119

4.3.4	Limites du système de déploiement . . . . .	120
	<b>Conclusion</b>	<b>121</b>
	<b>Bibliographie</b>	<b>125</b>



# Introduction

Utilisé dans la langue française « handicap » vient du terme anglais « hand in cap » (la main dans le chapeau) et tire ses origines d'un jeu pratiqué au 16<sup>e</sup> siècle en Grande-Bretagne. La règle était d'échanger à l'aveugle des biens dont la valeur était contrôlée par un arbitre qui assurait l'égalité des échanges. Le handicap est un concept qui a évolué au fil du temps, de sa compréhension et de son appréhension. Si l'on prêle à l'antiquité les premiers écrits relatant des cas d'individus souffrant de handicap, ils étaient, à l'époque, exclus de la société, considérés comme des victimes d'une malédiction divine, leur sort était souvent lié à la mort. Au moyen âge, le handicap est interprété comme une différence et suscite la peur auprès de la population qui les enferme dans des hospices et des « hôtels-Dieu ». Avec le temps, la médecine et les nouveaux courants de pensée prennent le dessus sur les croyances populaires et c'est au 18<sup>e</sup> siècle, sous l'impulsion de grands penseurs et philosophes des lumières, que la raison, la science et le respect de l'humanité font évoluer le concept du handicap. Au 19<sup>e</sup> siècle, les premières lois, faisant appel au terme « handicapé », apparaissent et des associations au niveau national se créent. La loi handicap du 11 février 2005, aussi appelée la loi « pour l'égalité des chances, la participation et la citoyenneté des personnes handicapées », ou loi handicap, est en France, la seule définition légale et officielle du handicap. Elle spécifie le contexte et l'importance de l'intégration du handicap dans la société et définit le cadre social et juridique, et les orientations politiques à prendre en France, sur les sujets du handicap.

Art. L. 114. - Constitue un handicap, au sens de la présente loi, toute limitation d'activité ou restriction de participation à la vie en société subie dans son environnement par une personne en raison d'une altération substantielle, durable ou définitive d'une ou plusieurs fonctions physiques, sensorielles, mentales, cognitives ou psychiques, d'un polyhandicap ou d'un trouble de santé invalidant.

La prise en charge des personnes en situation de handicap est une problématique importante pour notre société, puisqu'elle implique des enjeux humains et financiers. Nous pouvons l'observer spécialement en France, ces dernières années, où le soutien aux personnes dépendantes et en particulier celui aux personnes âgées a été l'une des promesses politiques de notre gouvernement. À ce titre, de nombreux financements ont été débloqués à l'échelle nationale pour promouvoir le maintien à domicile, notamment. La dépendance réside dans l'incapacité d'une personne à réaliser, seule, une action élémentaire ou complexe. La perte d'autonomie, exprimée par cette incapacité, introduit le besoin d'une assistance quotidienne adaptée pour compenser les situations de handicap. Or, l'assistance se traduit souvent par la présence d'une tierce personne, aide à domicile, infirmière, etc. voire une prise en charge complète en centre spécialisé, si la perte d'autonomie est importante. Cette situation peut être vécue comme une détérioration de dignité dès lors que des personnes s'introduisent dans leur espace, malgré le besoin réel. Toutes les solutions mises

en place pour combattre les dépendances ont un coût. Que ce soit le coût du traitement, de l'assistance personnelle ou des équipements pour compenser le handicap, il est pris en charge par le patient lui-même, mais aussi en partie financé par la société. On estime à 5 millions le nombre de personnes handicapées en France et le pays n'échappe pas à la règle générale qui s'applique à la situation mondiale, qui tend à montrer que ce chiffre n'a de cesse d'augmenter. Comme pour tous les pays, mais c'est particulièrement vrai dans les pays développés, les progrès de la médecine et les avancées technologiques ont pour conséquence une population vieillissante et plus soumise aux déficiences entraînant des incapacités.

L'enjeu social est bien présent, mais il n'est pas le seul. Ainsi, on estime que la perte économique mondiale liée au handicap s'élève à plusieurs milliards de dollars. Ajoutons à cela, le coût de la prise en charge des traitements, des actions d'assistance et de l'hospitalisation en institution, etc. Autant de coûts importants qui, pour une population dont la pauvreté est un facteur indéniable, ne peuvent être assumés seuls et sont en partie pris en charge par la société. Si l'on associe les facteurs de coût aux tendances démographiques, nous notons l'importance de l'enjeu financier et l'urgence de prendre des décisions pour améliorer une situation qui n'aura de cesse d'empirer dans le futur.

Depuis plusieurs années, la technologie est entrée dans nos vies, nous possédons tous un ordinateur, voire plusieurs, une télévision, un téléphone intelligent, etc. Tous ces équipements nous facilitent la vie au quotidien. Un téléphone intelligent, outil de communication, nous sert d'agenda et permet de programmer des rendez-vous qui nous seront rappelés par des alertes audio, visuelles et sensibles. Nous consommons les informations rapidement grâce à la télévision, à internet sur notre ordinateur ou notre téléphone, assistons à des visio-conférences, à travers le monde, etc. Dernièrement, les bracelets et autres montres connectées ont fait leur apparition. Ils nous permettent de « mieux » nous connaître. D'ailleurs, parlant des objets connectés et de leur évolution, il ne passe pas un jour sans qu'un nouvel équipement ne s'ajoute à cette longue liste, télévision, réfrigérateur, pèse personne, ampoules, alarmes, etc. Autant de ressources qui nous permettent de nous analyser. Autant de ressources dont les capacités ne sont pas encore pleinement exploitées. Pour recentrer le sujet sur l'aide au handicap, ces informations glanées au fil des capteurs nous permettent d'imaginer des algorithmes pour modéliser, comprendre et anticiper nos actions. Des projets de recherches ont déjà montré des résultats probants sur l'analyse d'activité de la vie quotidienne, en isolant des schémas réguliers se répétant. Dans ce cas, des anomalies de comportement peuvent être observées et compensées par d'autres actions.

Dans les progrès technologiques importants de ces dernières années, n'oublions pas la domotique qui en terme de contrôle, gestion et sécurité de son environnement a fait un bond spectaculaire. A quelques fantasmes visionnaires près, personne n'imaginait il y a 30 ans pouvoir piloter et surveiller sa maison à des milliers de kilomètres. Les premiers projets émergents pendant les années 50, portés par la révolution de la miniaturisation des composants électroniques. Cependant, la vision d'un habitat intelligent relève de la science-fiction puisqu'à l'époque très peu sont ces familles ayant un accès à la technologie, télévision, réfrigérateur, etc. En 1965, seuls 40% de la population française possède un écran de télévision et suit en direct l'élection présidentielle. C'est dans les années 80 que la domotique prend son sens et devient réalité. Même si en ce temps, elle est un signe de richesse abordable pour peu de personnes, on commence à parler de maisons motorisées pour la raison qu'on peut désormais piloter un portail, un garage, etc. Aujourd'hui, la maison domotique est programmée, automatisée, et pilotable à distance. L'informatique s'est répandue et diffusée dans tous nos appareils : du réfrigérateur à nos chauffages. La domotique a évolué sur de nombreux points les capteurs deviennent diffus et les action-

neurs toujours plus variés. De plus, grâce aux travaux sur l'analyse des données et de l'intelligence artificielle, nous arrivons au stade de l'habitat autonome, autogéré.

A ce stade, deux questions nous intéressent : Quel peut être l'impact de la technologie dans les situations de handicap ? Et comment exploiter au mieux ces ressources hétérogènes et complexes pour assister les personnes souffrant de handicap ? Si l'Organisation Mondiale de la Santé, dans sa dernière définition du handicap, permet de répondre à la première des questions, la seconde reste à traiter. En effet, les travaux sur les systèmes domotiques et les environnements intelligents, dans le monde de l'aide à la personne, nous ouvrent la voie vers un univers qui reste à explorer.

La notion d'un système d'assistance au handicap se définit à deux niveaux : à la conception et à l'exécution. La première partie, la conception du système, se consacre à l'analyse de la situation du patient et de son environnement. Le patient, futur utilisateur, possède des besoins spécifiques. L'environnement du patient est constitué d'équipements qui composent l'infrastructure matérielle. Les besoins orientent et dimensionnent cet ensemble d'équipements et peuvent mener à une évolution de l'infrastructure. La combinaison des besoins et de l'infrastructure permettent de former l'offre de service qui sera proposée au patient pour compenser ses incapacités. La phase de conception a donc pour objectif, à partir du profil du patient et de son environnement de faciliter, voire dans certains cas d'automatiser, la réalisation du système d'assistance. Ce système d'assistance comprend un programme qui permet l'exécution des services d'aide au handicap pour le patient dans son environnement. La seconde partie traite l'exécution du système d'assistance. Cette étape se focalise sur le fonctionnement du système et son comportement en utilisation, notamment pour le déploiement des services. Le patient interagit avec le système grâce à l'interface homme machine dont il dispose. Le cœur du système, un organe de déploiement contenant les informations d'activation et protocoles de communication, est alors chargé d'exécuter le service demandé. L'exécution du service sur l'infrastructure matérielle génère un effet capable de satisfaire le besoin du patient.

Nous remarquons qu'il existe deux grandes tendances complémentaires dans l'assistance aux personnes et l'aide au handicap : le contrôle d'environnement et l'analyse comportementale. Parce que les besoins des patients sont uniques et spécifiques à leur pathologie, les solutions proposées sont tout aussi singulières. Ainsi, l'infrastructure support (matérielle et logicielle) se trouve fortement impactée puisque le contrôle d'environnement demandera beaucoup d'équipements de type actionneur et de communication alors que l'analyse requerra de nombreux capteurs et des capacités de traitement plus grandes. La tendance générale vise à se consacrer soit sur les travaux de contrôle d'environnement, soit sur l'analyse du comportement, or, nous tentons d'appréhender les deux mondes conjointement en unifiant les solutions dans un même système. Les projets d'aide au handicap sont toujours plus ambitieux et proposent des algorithmes réalisant des actions complexes. Par exemple, les travaux sur la fouille de donnée permettent de reconnaître des activités réalisées par le patient. Remis dans le contexte du temps réel, le besoin instantané de capacités de traitement requis peut être conséquent. De manière générale, les applications développées pour l'assistance aux personnes sont toujours plus performantes, autant dans la qualité du service fourni que dans la précision de la réponse apportée. Ceci semble conduire à la nécessité de rajouter des ressources de traitement toujours plus puissantes et adaptées. Pourtant, l'infrastructure matérielle disponible dans l'environnement du patient dispose déjà de nombreux équipements alors pourquoi rajouter des ressources dédiées et coûteuses ? Nous ne pouvons que constater que malgré des équipements toujours plus nombreux, performants et polyvalents, etc. on observe un manque dans l'exploitation de ces

ressources. Si l'on sait optimiser l'utilisation d'un équipement de manière individuelle, on maîtrise beaucoup moins l'optimisation conjointe de plusieurs appareils. Or l'association de toutes les ressources domotiques de l'environnement nous permet de créer une entité ubiquiste surpuissante.

Notre objectif est de venir corriger cette faiblesse en créant un outil de déploiement d'application d'assistance, plaçant l'infrastructure et son comportement au centre des préoccupations. Autrement dit, avant de pouvoir satisfaire les besoins du patient, nous devons nous assurer que le système pourra exécuter le service en utilisant conjointement l'ensemble de ses capacités. L'exploitation conjointe des composants de l'environnement commence dès la phase de conception avec un raffinement de la description des équipements domotiques. Nous intégrons les capacités et les propriétés intrinsèques de chaque ressource, en temps réel, dans une base de connaissances. Cet organe d'information est ensuite interrogé pour déployer de manière optimale les services sur des ressources hétérogènes. L'analyse de l'utilisation des capacités est assurée par un superviseur qui se charge de mettre à jour régulièrement la base de connaissances. Ainsi, en prenant connaissance du contexte de l'infrastructure, de son utilisation courante et en répartissant l'exploitation de ses capacités, nous tentons de garantir une exécution des services en toute condition. En outre, lorsque l'organisation des applications sur l'ensemble des ressources ne suffit plus, nous mettons en place des mécanismes internes de gestion des services entre eux. Ainsi, des services peuvent être suspendus temporairement pour libérer des capacités et des versions dégradées et alternatives de ces services sont prévues pour affiner l'optimisation.

Dans ce document, nous nous consacrons au déploiement optimisé de services d'aide au handicap dans un environnement domotique respectant les spécifications de l'assistance aux personnes. Nous étudierons dans le premier chapitre le handicap, de sa production jusqu'aux situations d'incapacités. Nous reprendrons l'évolution de la technologie et de la domotique et observerons ses applications à travers le monde pour compenser les situations de handicap. Enfin dans ce chapitre nous positionnerons nos travaux et exposerons nos motivations. Dans la seconde partie du document, nous introduirons le besoin d'une conception adaptée au patient et à son environnement et la nécessité de formaliser cette description dans l'optique d'un déploiement optimal. Notre flot de déploiement et ses constituants seront présentés dans le troisième chapitre. Enfin, le quatrième chapitre se focalisera sur la présentation du démonstrateur, la justification des choix faits pendant le projet ainsi que sur une discussion autour des résultats.

# Chapitre 1

## La technologie au service du handicap

### Sommaire

---

<b>1.1</b>	<b>Le handicap, un challenge social et économique . . . . .</b>	<b>11</b>
1.1.1	Introduction générale au handicap . . . . .	11
1.1.1.1	Le modèle de développement humain . . . . .	11
1.1.1.2	Le processus de production du handicap . . . . .	11
1.1.1.3	Les grandes catégories d'incapacités . . . . .	14
1.1.2	La situation des personnes handicapées dans le monde . . . . .	14
1.1.3	Le cas des personnes âgées . . . . .	16
1.1.3.1	Démographie et géographie . . . . .	16
1.1.3.2	Une population fragile . . . . .	16
1.1.4	De l'incapacité à la dépendance, de l'hospitalisation au maintien à domicile . . . . .	17
<b>1.2</b>	<b>La technologie au quotidien . . . . .</b>	<b>18</b>
1.2.1	La démocratisation de la technologie . . . . .	18
1.2.2	La domotique : le paradigme du grand public . . . . .	19
1.2.3	La domotique : les spécifications de l'aide au handicap . . . . .	20
1.2.4	Les solutions domotiques d'assistance . . . . .	20
<b>1.3</b>	<b>Revue des principaux habitats intelligents et de leurs solutions d'aide au handicap . . . . .</b>	<b>21</b>
1.3.1	Les habitats intelligents à travers le monde . . . . .	22
1.3.2	Les solutions développées au Lab-STICC . . . . .	23
1.3.2.1	Projets sur l'habitat intelligent de l'action HAAL . . . . .	23
1.3.2.2	Projet sur l'habitat intelligent de l'équipe MOCS . . . . .	24
<b>1.4</b>	<b>Positionnement et motivation . . . . .</b>	<b>27</b>

---



## 1.1 Le handicap, un challenge social et économique

Ce chapitre présente, dans sa première partie, le handicap de sa production à sa situation cruciale pour la société. La France, comme tous les pays développés et en voie de développement, connaît une augmentation de la proportion de personnes souffrant d'une situation de handicap. C'est une règle globale entraînée notamment par le vieillissement de la population mondiale et par conséquent plus sujette à des déficiences. L'introduction générale a montré l'importance de considérer et d'adapter des solutions pour compenser les situations de handicap. Dans la suite de cette partie, la section 1.1.1 introduit la définition du handicap et son modèle tandis que la section 1.1.2 dresse la situation du handicap dans le monde. Le cas des personnes âgées, une population plus sujette au handicap, est abordé dans la section 1.1.3. Enfin, le lien entre incapacité et dépendance ainsi que les conséquences sont établis dans la section 1.1.4.

### 1.1.1 Introduction générale au handicap

Le handicap est un concept qui a évolué dans le temps, particulièrement ces dernières décennies. Toutefois, les interprétations et les réponses apportées aux questions du handicap ont largement été influencées par les époques et la perception de la société. Longtemps attribué à un problème individuel, de par sa compréhension purement médicale, le modèle individuel généré par cette interprétation de la situation a placé l'individu ayant une déficience comme le problème à résoudre. Ainsi, la responsabilité du handicap appartient à la personne qui doit être « réparée ». L'exclusion sociale étant souvent la seule réponse apportée à cette compréhension du handicap. En réaction au modèle individuel est apparu, dans les années 70, aux États-Unis et au Royaume-Uni, un modèle social du handicap. Partant du principe que la déficience fait partie intégrante de la vie, elle demande des réponses adaptées. On ne place plus l'individu au centre du problème, mais les barrières externes et environnementales. Ici, les personnes handicapées ne sont alors plus considérées comme désavantagées en raison de leurs caractéristiques individuelles, mais limitées par leur société. Le handicap devient la conséquence de l'organisation de la société et non plus la cause. En gardant conscience de l'aspect individuel déterminé par les soins médicaux, les réponses apportées par le modèle social, aux questions du handicap, portent sur l'élimination des barrières qui empêchent l'intégration des personnes handicapées à la vie commune.

#### 1.1.1.1 Le modèle de développement humain

Le « Modèle du développement humain », schématisé sur la figure 1.1, a été proposé en 1996 par le Réseau International sur le Processus de Production de Handicap (RIPPH). Il met en évidence les éléments produisant des situations de handicap. Le développement humain résulte des interactions entre les facteurs personnels (intrinsèques à l'individu) et les facteurs environnementaux (extrinsèques à l'individu) qui vont influencer le degré de participation sociale.

#### 1.1.1.2 Le processus de production du handicap

Transporté dans le domaine du handicap, le modèle du développement humain, est adapté en 1998 pour devenir le Processus de Production du Handicap [1]. La notion de participation sociale devient une habitude de vie, et est intégrée comme une interaction

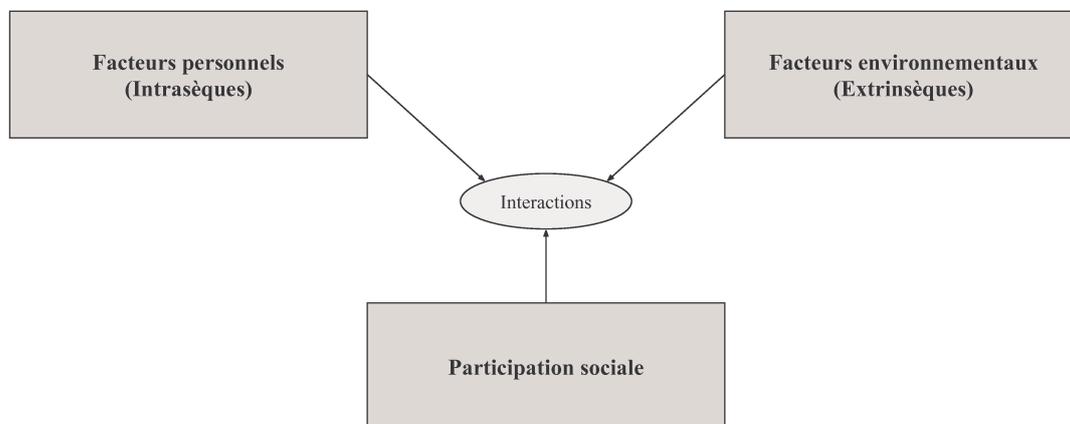


FIGURE 1.1 – Modèle du développement humain par le RIPPH

entre des facteurs personnels et des facteurs environnementaux. Elle est la variable d'évaluation de la situation de handicap. Les facteurs personnels deviennent la combinaison de l'état de systèmes organiques, qui varient selon leur degré d'intégrité ou de déficience, et d'aptitudes, qui eux-mêmes varient selon la capacité ou l'incapacité d'un individu. Les facteurs environnementaux sont considérés comme des facilitateurs ou des obstacles aux habitudes de vie d'une personne. Les interactions de ces facteurs renforcent la participation sociale de la personne ou sa situation de handicap. Les facteurs personnels peuvent être limités par les facteurs de risques. Ces derniers appartiennent à l'individu ou proviennent de l'environnement et peuvent engendrer une maladie, un traumatisme, porter atteinte à l'intégrité d'une personne ou ralentir son développement.

La figure 1.2 illustre le processus du développement du handicap. Pour bien comprendre les interactions entre les facteurs personnels, environnementaux et habitudes de vie, le RIPPH propose les définitions suivantes.

Définition des facteurs personnels :

- Un facteur personnel : est une caractéristique intrinsèque de l'individu comme son âge, son sexe, sa catégorie socioprofessionnelle, son identité socioculturelle, etc., mais aussi un système organique ou encore une aptitude.
- Un système organique : est un ensemble d'organes dont le fonctionnement contribue à une tâche complexe commune, par exemple, le système cardiovasculaire, le système respiratoire, le système musculaire, etc.
- Une aptitude : est la faculté qu'un individu a, à accomplir une activité physique ou mentale. On peut notamment citer les aptitudes liées aux capacités intellectuelles, les aptitudes liées aux activités motrices, les aptitudes liées à la respiration, etc.

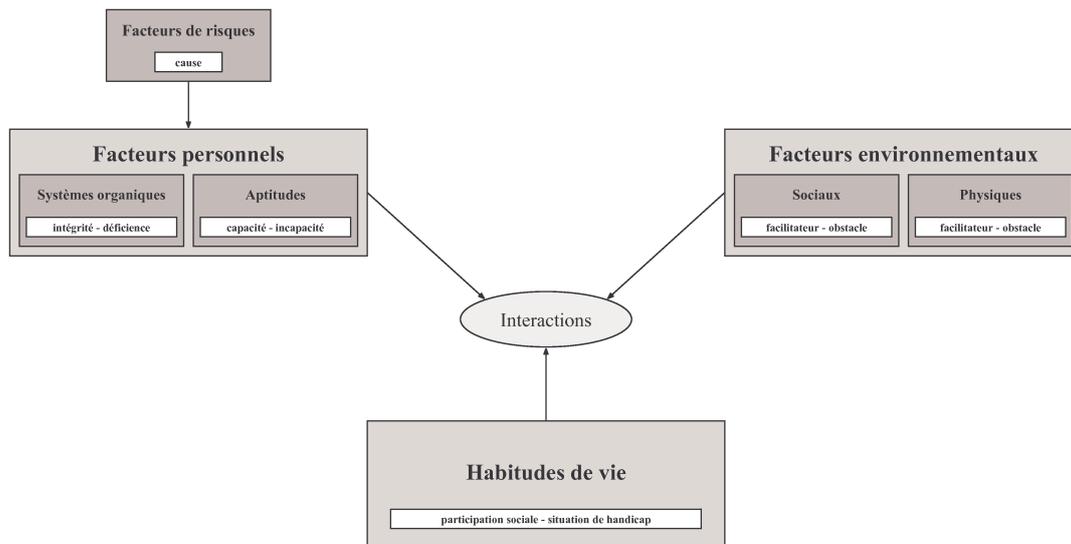


FIGURE 1.2 – Modèle du processus de production du handicap par le RIPPH

Définition des facteurs environnementaux :

- Un facteur environnemental : est une dimension appartenant directement à la société. On distingue deux grandes catégories que sont les facteurs physiques, natures et aménagements; et les facteurs sociaux, politico-économiques et socio-culturelles. Par exemple, un bâtiment de plusieurs étages sans ascenseur devient un obstacle pour certaines personnes.
- Un facilitateur : est un facteur environnemental qui favorise les habitudes de vie.
- Un obstacle : est un facteur environnemental qui empêche les habitudes de vie.

Enfin, le résultat des interactions entre les habitudes de vie, les facteurs personnels et les facteurs environnementaux définissent la situation de handicap d'une personne. Par exemple, une personne ayant une déficience du système oculaire, possédant une canne blanche ou un chien guide, et se promenant dans une rue aménagée de signaux sonores ne constitue pas une personne en situation de handicap.

Le modèle de processus de production du handicap s'accorde parfaitement avec le modèle de Classification Internationale du Fonctionnement, du Handicap et de la Santé (CIF) proposé par l'Organisation Mondiale de la Santé (OMS), en anglais *World Health Organisation* (WHO), et la définition du handicap proposée par la Convention relative aux Droits des Personnes Handicapées des Nations Unie (CDPH). Une compréhension claire des notions de déficience, handicap et incapacité, nous permet de définir le handicap comme une situation et non comme une caractéristique individuelle. Le handicap est donc une notion contextuelle et évolutive, qui correspond à la capacité ou l'incapacité de l'individu à réaliser une habitude de vie.

### 1.1.1.3 Les grandes catégories d'incapacités

Il existe 5 grandes catégories d'incapacités :

- L'incapacité motrice : se traduit par la difficulté d'un individu à se déplacer, réaliser des gestes ou bouger ses membres. L'incapacité peut être partielle ou totale, temporaire ou incurable. Exemples : paralysies, amputations, infirmité motrice cérébrale, myopathie, etc.
- L'incapacité sensorielle : exprime les difficultés en lien avec les organes sensoriels. Cette incapacité regroupe plus communément le sens visuel et le sens auditif. Exemples : cécité, malvoyance, surdité, etc.
- L'incapacité intellectuelle : traduit les déficiences des fonctions mentales et intellectuelles, entraînant des difficultés de réflexion, de compréhension et même de conceptualisation. Exemples : Alzheimer, autisme, polyhandicap, etc.
- L'incapacité psychique : caractérise une pathologie entraînant des troubles mentaux, affectifs et émotionnels sans pour autant influencer sur les capacités intellectuelles. Exemples : bipolarité, schizophrénie, etc.
- Les maladies invalidantes : sont des maladies pouvant provoquer une incapacité et évoluer dans le temps. On peut citer les maladies dégénératives, respiratoires, infectieuses, etc. Exemples : sclérose latérale amyotrophique, épilepsie, etc.

### 1.1.2 La situation des personnes handicapées dans le monde

La référence lorsque l'on souhaite observer l'état du handicap dans le monde, est le rapport proposé par l'Organisation Mondiale de la Santé et la Banque Mondiale [2]. Ce rapport mondial a pour objectif de regrouper les faits scientifiques avérés les plus probants, disponibles sur la question du handicap. Les chiffres importants à retenir sont que plus d'un milliard de personnes vivent avec un handicap, sous une forme ou sous une autre, soit environ 15% de la population mondiale et dont 110 millions éprouvent de fortes difficultés fonctionnelles. Il existe aussi un lien très étroit entre le handicap et la pauvreté, puisqu'on estime à 80% la part de personnes handicapées vivant dans un pays en développement. Le rapport sur le handicap montre aussi que le nombre de personnes handicapées est en constante croissance, due à l'allongement de l'espérance de vie et à l'explosion des maladies chroniques comme le diabète et les maladies cardio-vasculaires, mais aussi de facteurs environnementaux comme les catastrophes naturelles ou les accidents de la route.

Le lien entre pauvreté et handicap ne doit pas être négligé, puisqu'il génère un cercle vicieux menant du handicap à la pauvreté et de la pauvreté au handicap. Pour illustrer cette situation, les étapes du lien entre handicap et pauvreté sont schématisées, Figure 1.3. Nous remarquons un effet cyclique qui se renforce au fil du temps. Le handicap entraîne une exclusion sociale et culturelle poussant à son tour vers une privation des opportunités pour le développement de ces personnes, dans la société. Le manque de droit observé participe à l'appauvrissement individuel et favorise leur vulnérabilité. L'affaiblissement issu de cette condition accentue les possibilités de présenter un handicap et ainsi de suite. Ce cercle vicieux fait ressortir la conséquence qui mène la personne en situation de handicap vers une situation de dépendance.

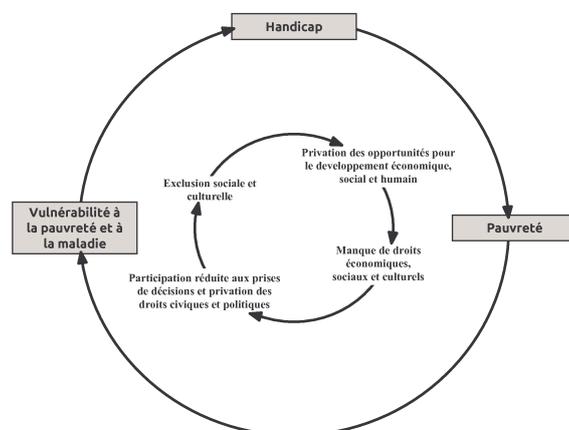


FIGURE 1.3 – Schéma circulaire du handicap

C'est dans ce contexte, révélé par le *Department for International Development* (DFID) et son rapport [3], qu'il est défini que le handicap est à la fois une cause et une conséquence de la pauvreté. Les principaux liens entre pauvreté et handicap sont :

- Les conditions de vie insalubres et dangereuses
- L'absence ou l'inaccessibilité aux soins médicaux ou de réadaptation
- L'accès restreint à l'éducation et à l'emploi
- L'exclusion de la vie sociale

L'enjeu économique sous-jacent a été établi par la Banque Mondiale en 2004 [4] où la perte mondiale de produit intérieur brut causé par le handicap est estimée à environ 2000 milliards de dollars annuels. D'un point de vue individuel, une personne subit doublement le coût du handicap puisqu'elle doit assumer les frais médicaux et en réadaptation, une assistance médicale onéreuse, et qu'elle souffre d'une perte de revenu conséquente, liée à son handicap. Le handicap favorise la paupérisation des personnes touchées qui ne peuvent seules subvenir à l'ensemble des dépenses engendrées par leur situation. Dans un contexte plus global, on prend conscience que les enjeux sont démographiques. 15% de la population est touché par un handicap et ce chiffre continue d'augmenter au fil du temps, puisque le vieillissement de la population mondiale favorise l'acquisition d'une déficience pouvant entraîner une incapacité. Cependant, les enjeux sont aussi sociaux et juridiques, car le handicap touche les aspects essentiels de la vie comme l'accès aux services de base tels que la santé, l'éducation, l'emploi, etc. Le sentiment de discrimination et l'exclusion de la vie sociale sont autant de paramètres à prendre en compte pour le développement sain d'une société. En résumé, une personne souffrant de handicap est avant tout une personne et doit pouvoir bénéficier des mêmes droits, devoirs et obligations que tout autre individu, comme il est défini dans la Convention des Nations Unies relative aux Droits des Personnes Handicapées (CDPH).

### 1.1.3 Le cas des personnes âgées

Les personnes âgées sont souvent prises en exemple pour exprimer les situations de dépendance. D'une part cette situation nous concernera tous un jour ou l'autre, et d'autre part cette population est plus fragile et souffre, par conséquent, plus facilement d'incapacités, de handicap et de dépendances.

#### 1.1.3.1 Démographie et géographie

Tout part d'un constat simple, au 20<sup>e</sup> siècle, la population mondiale est passée de 1,6 milliard à 6,1 milliards. Nous avons même accueilli la naissance du 7 milliardième humain en 2011 et ce n'est pas fini puisque les chiffres affirment que nous sommes déjà plus de 7,2 milliards d'habitants de la planète terre en 2014. Or, on estime que la proportion de la population mondiale de plus de 60 ans va doubler, entre 2000 et 2050, pour passer de 11% à 22%. L'OMS chiffre que les personnes âgées devraient passer de 65 millions à deux milliards au cours de cette période. Bien entendu, les premiers pays touchés seront les pays développés suivis des pays en développement. Cependant, il ne faut pas négliger les pays à revenu faible et intermédiaire, puisque, l'OMS estime que l'évolution de la proportion des personnes âgées leur prendra moitié moins de temps pour arriver au même stade. Si en France, un siècle a été nécessaire pour que le taux de personnes âgées de plus de 65 double, passant de 7% à 14%, les estimations tablent sur 25 ans pour que la population des pays comme le Brésil et la Chine atteignent la même croissance. Les graphiques, issus des sites de l'OMS et du *Population Reference Bureau* (PRB), sur la figure 1.4, montre que la population des personnes âgées va considérablement augmenter sur la grande majorité de la surface du globe.

Porté par les progrès de la science en particulier dans le domaine de la médecine, mais aussi par les avancées technologiques, et l'amélioration du confort et de la qualité de vie, le nombre de personnes âgées de plus de 80 ans ou plus aura été multiplié par presque quatre pour atteindre les 390 millions entre 2000 et 2050. C'est une première dans l'histoire de l'humanité puisqu'on pense qu'une majorité d'adultes auront leurs parents toujours en vie et que leurs enfants rencontreront davantage leurs grands-parents et éventuellement leurs arrière-grands-parents.

En France métropolitaine, si les tendances démographiques estimées ces dernières années se maintiennent au même niveau, la population du pays devrait compter environ 70 millions d'habitants, à l'aube de 2050 dont un tiers aura plus de 60 ans. La France n'est pas un cas isolé, puisque les projections sont encore plus marquées pour d'autres pays européens comme l'Allemagne et le Royaume-Uni.

#### 1.1.3.2 Une population fragile

Sauf pandémies ou catastrophes naturelles imprévisibles qui rendraient les estimations des projections mathématiques fausses, le taux de natalité dans les pays développés est faible et en diminution alors que l'espérance de vie augmente au fil des décennies, des découvertes scientifiques et des progrès technologiques, le vieillissement de la population est donc inéluctable. Or cette population dispose d'une sensibilité aux déficiences plus fortes que toutes les autres, puisque comme l'indique la définition du vieillissement « Le vieillissement est un affaiblissement naturel des facultés physiques et psychiques dû à l'âge ». En effet, nos capacités fonctionnelles issues de nos systèmes organiques se développent pendant les premières années de notre vie pour décliner au fur et à mesure des années

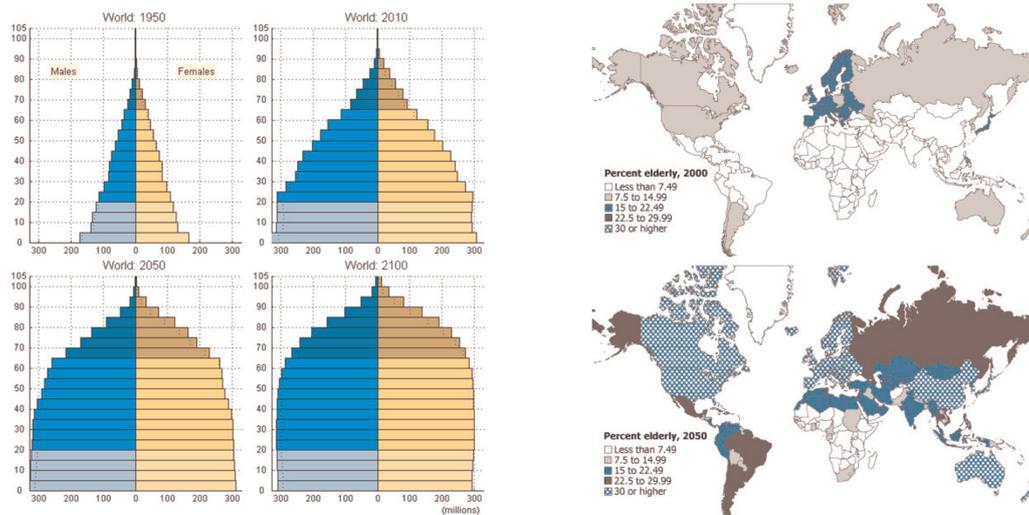


FIGURE 1.4 – Projection des personnes âgées dans le monde

passées dans notre enveloppe charnelle. Même si nous sommes tous inégaux vis-à-vis du déclin de nos fonctions vitales, nous sommes tous soumis à cette dégénérescence naturelle.

#### 1.1.4 De l'incapacité à la dépendance, de l'hospitalisation au maintien à domicile

Les personnes en situation de handicap, et plus particulièrement les personnes âgées, sont une population fragile autant d'un point de vue de la santé que d'un point de vue économique. Le handicap appauvrit ces personnes et leur situation précaire favorise ainsi le développement d'un handicap et ainsi de suite. Nous sommes en présence d'une catégorie de personne dont la condition requiert une attention toute particulière. En effet, la situation de handicap traduit une incapacité qui si elle n'est pas corrigée entraîne une situation de dépendance. Or d'ici 2050, dans les pays en développement, l'OMS prédit que le nombre de personnes âgées qui auront perdu leur autonomie et ne pourront plus vivre seules, devrait être multiplié par quatre.

La perte d'autonomie est liée au fait que les personnes handicapées ne peuvent plus réaliser des tâches de la vie quotidienne à cause d'incapacités motrices ou intellectuelles. Elles éprouvent des difficultés à se déplacer, perdent la mémoire ou encore la notion du temps, mais ont aussi une santé plus fragile et requièrent un suivi médical poussé. Cette prise en charge adaptée passe par une assistance à domicile : aides-soignants pour les traitements médicaux ou aides à domicile pour les tâches de la vie quotidienne. L'incapacité évolue dans le temps et peut considérablement s'aggraver. Aussi, pour des cas plus graves, où une prise en charge classique à domicile ne suffit plus à compenser la dépendance, l'hospitalisation en établissement spécialisé devient la seule solution envisageable. Par ailleurs, l'OMS prévoit une augmentation spectaculaire du nombre de personnes atteintes de démences telles que la maladie d'Alzheimer, dans les prochaines années.

L'hospitalisation est une phase déterminante puisqu'elle affecte énormément les personnes en situation de handicap. De plus, l'assistance et les traitements plus lourds ont aussi un coût supplémentaire. C'est en partie la conséquence de ces constats qui mène les politiques actuelles à s'intéresser, de plus en plus, au maintien à domicile. Toutefois, il est légitime de se poser la question de comment repousser l'hospitalisation lorsque les services à domicile ne suffisent plus. Nous avons vu dans ce chapitre que le handicap était une situation qui variait au gré des interactions avec les facteurs environnementaux. Les technologies actuelles, informatiques et domotiques, sont vues comme des facilitateurs et peuvent ainsi servir à combattre la dépendance. Enfin, avec un aménagement du domicile du patient et des services d'aide au handicap adaptés, la technologie offre à la personne la possibilité de retrouver son autonomie et par conséquent son indépendance.

## 1.2 La technologie au quotidien

L'OMS dans sa définition du handicap de 2001, reprise notamment dans la législation française, cadre le contexte et montre que la situation de handicap peut être compensée par une transformation des facteurs environnementaux du patient. Pour prendre un exemple simple, un patient atteint d'une déficience physique est en situation de handicap si ce dernier ne peut pas se déplacer seul dans son logement à cause de l'incapacité à ouvrir des portes. L'ajout d'un système d'ouverture de porte automatisé corrige cette incapacité et le patient n'est dès lors plus considéré en situation de handicap.

Il faut bien tenir compte du fait que la prise en compte des besoins du patient n'est pas toujours aussi simple. Il existe autant de pathologie que de patient et la situation de handicap peut être la somme d'une multitude de facteurs. C'est à ce niveau qu'interviennent les travaux de spécifications et d'expressions des besoins du patient, dans le but de lui offrir un système compensant le plus précisément possible ses situations d'incapacités, comme ceux d'IntelHome [5]. A travers les modèles d'environnement et d'interaction, la technologie apparaît une première fois, grâce à la domotique notamment qui offre suffisamment de solutions pour permettre au patient de garder son autonomie et de repousser sa dépendance le plus longtemps possible.

Cependant, la domotique n'a pas pour vocation de traiter le handicap puisqu'elle a d'abord été créée dans le cadre du grand public, une catégorie vaste de personne très éloignée des besoins particuliers des personnes que nous ciblons, dans nos travaux. Dans une première section 1.2.1, nous analysons la place qu'a prise la technologie dans nos vies, au fil de ces dernières années. Puis nous confrontons les visions des solutions grand-public et spécifiques à l'aide au handicap dans les sections 1.2.2 et 1.2.3.

### 1.2.1 La démocratisation de la technologie

Si à l'heure actuelle la génération de nos aînées n'est pas toujours très à l'aise avec la technologie, on peut aisément prévoir une inversion du phénomène avec les générations futures. C'est seulement avec la seconde révolution industrielle, il y a de cela près d'un siècle, que la technologie a commencé à se diffuser dans le quotidien avec l'invention de l'électricité. Le bouleversement a été gigantesque puisqu'il a impacté aussi bien l'industrie que le particulier. La suite de l'évolution a, très rapidement, donné naissance aux transistors et continué jusqu'à l'avènement de l'informatique. La technologie a continué sa diffusion et s'est miniaturisé au fil du temps.

Si bien qu'aujourd'hui, notre génération entretient un lien étroit avec la technologie. On parle désormais de dépendance à cette dernière et l'on découvre de nouvelles maladies

liées à l'addiction à internet ou à notre téléphone mobile (nomophobie). Sans tomber dans le débat du rapport qui nous lie à la technologie, nous ne pouvons nier sa démocratisation, tant d'un point de vue de l'accessibilité que d'un point de vue de l'interactivité. La technologie s'est adaptée à nous autant que nous nous sommes adaptés à elle.

La technologie s'est imposée au fil des années comme un outil du quotidien, dans le cadre de notre travail comme dans le cadre personnel, et nous sommes au cœur de son processus de développement. Ce constat nous mène à penser que notre compréhension de la technologie n'aura de cesse d'augmenter tandis que notre appréhension disparaîtra.

La domotique n'échappe pas à cette règle, elle est et sera de plus en plus présente au cœur de nos maisons. On trouve parfois dans notre espace de vie plusieurs ordinateurs personnels, fixes ou mobiles, les télévisions sont connectées et intelligentes, nos appareils électroménagers s'intègrent aussi dans la domotique, etc.

### 1.2.2 La domotique : le paradigme du grand public

Développée pour faciliter les actions quotidiennes de l'utilisateur, la domotique pour le grand public suit une vision simple de confort et praticité, dont les 4 grands axes d'une vision grand public sont présentés ci-dessous :

- Confort domestique : Nombreuses sont les solutions pour contrôler les portes de garage et portails électriques. Désormais, on peut contrôler les éléments du mobilier tels que les portes à l'intérieur de son domicile, les volets, voire même les fenêtres ou encore la hauteur des placards et plans de travail. Le confort domestique comprend aussi le pilotage des appareils électrodomestiques comme un four ou un aspirateur. Ce sont aussi des solutions de confort et de praticité.
- Gestion énergétique : La gestion thermique et électrique est une notion importante de la domotique grand public. Pouvoir faire des économies tout en assurant son confort est l'objectif proposé par cet axe. Ainsi, on retrouve des solutions d'automatisation de l'éclairage ou la recharge d'équipements, voiture électrique par exemple, mais aussi une gestion autonome de son chauffage, de sa climatisation, etc.
- Sécurité et alarmes : C'est l'assurance de se protéger à l'intérieur de son logement avec des détecteurs de fumée et autres capteurs d'alarme, mais aussi de bénéficier d'une protection en cas d'absence avec des solutions prévenant et alertant des intrusions. Il est même possible de gérer l'ensemble de sa maison à distance.
- Applications multimédias : On le remarque encore aujourd'hui avec les très récentes offres de films et séries accessibles par internet, cet axe de développement est en plein essor. C'est le confort des installations cinéma avec l'aspect pratique d'être à domicile. On peut aisément en y mettant le prix obtenir un système multimédia complet.

Dans un cadre commercial, le but est de toucher le plus grand nombre de clients avec une même offre pour limiter les coûts de développements. Ici, on ne prend pas en compte les besoins individuels des utilisateurs tant au niveau de l'offre de service que des interactions entre l'homme et le système. Pour ce qui nous concerne, suivant la pathologie du patient, le médecin peut proposer des services adaptés, la nécessité d'une conception unique est de surcroît évidente. De plus, les moyens d'interactions des patients peuvent

être complètement différents d'un cas à l'autre, il peut s'agir d'une préférence ou tout simplement d'une condition physique de l'utilisateur.

### 1.2.3 La domotique : les spécifications de l'aide au handicap

Les solutions domotiques pour l'aide au handicap répondent aux besoins spécifiques du patient. Leur conception est donc unique et dépendante du cas du patient. Ces solutions pour favoriser leur intégration et leur acceptation doivent respecter plusieurs critères :

- Individualisation : Chaque patient possède des besoins uniques dépendant de sa pathologie. La solution pour compenser son handicap doit être adaptée et non générale. L'individualisation de la solution est donc un challenge important de la conception.
- Adaptabilité : Chaque environnement est différent et évolue. Une solution doit être évolutive pour s'adapter à un changement d'ordre de grandeur, tout en maintenant son fonctionnement global. Un système peut être utilisé à l'échelle d'une pièce, d'un appartement ou encore d'un centre hospitalier, mais doit pouvoir intégrer de nouveaux équipements à la volée.
- Interopérabilité : Chaque technologie possède sa propre interface et son protocole de communication propriétaire. Bien que plusieurs initiatives cherchent à inverser la tendance, l'interopérabilité reste un sujet préoccupant.
- Disponibilité : La disponibilité est le critère de sûreté de fonctionnement. Dans le cadre de l'aide au handicap, la solution ne peut pas se permettre de ne pas répondre aux besoins du patient. Des mécanismes de fonctionnement dégradé ou alternatif doivent être mis en place pour assurer la sécurité du patient.
- Coût : Lorsqu'on parle de solutions grand public, il faut préciser que le choix du système domotique appartient à l'utilisateur. Dans le monde de l'aide au handicap, ce choix est souvent une nécessité, le dernier rempart avant l'hospitalisation. Le critère de coût est d'autant plus important pour une population où la pauvreté est un enjeu important.
- Intrusivité : Le critère intrusif est là encore un enjeu important, puisqu'une personne handicapée est avant tout une personne et qu'en conséquence sa vie privée doit être respectée.

### 1.2.4 Les solutions domotiques d'assistance

Les solutions domotiques d'aide au handicap visent à compenser les incapacités des patients. Parmi les grandes incapacités, nous identifions deux principaux handicaps que les maisons intelligentes tendent à résoudre.

Le handicap physique traduit une situation d'incapacité à se déplacer et à interagir avec les éléments présents dans l'environnement du patient. Certaines tâches quotidiennes comme ouvrir ses volets ou allumer la lumière sont difficiles, voire impossibles à réaliser sans l'aide d'une tierce personne. Certaines solutions d'assistance associées aux maisons intelligentes sont focalisées sur l'augmentation des capacités physiques du patient, et présentent des caractéristiques particulières. Outre la présence de quelques capteurs pour permettre d'automatiser certaines actions, l'environnement domotique dispose surtout d'un ensemble considérable d'actionneurs pour donner au patient la possibilité de contrôler le plus d'appareils domotiques possible. Les solutions de contrôle environnemental disposent

souvent d'une interface adaptée aux capacités du patient [6] (tactile [7], oculométrie [8], etc.), mais aussi d'une conception conforme aux besoins du patient comme le proposent les projets HomeDSL [9], [10] et IntelHome [5]. Cependant, le contrôle environnemental est influencé par la domotique grand public et toutes les solutions techniques sont issues des grands constructeurs mondiaux. Nous pouvons citer parmi les leaders français la société Legrand ou encore Delta Dore et en Europe le standard Konnex (KNX) avec son bus domotique et son protocole de communication ouvert.

Enfin, pour compenser un handicap mental, la solution domotique va se focaliser sur l'analyse du comportement du patient, pour l'assister au quotidien dans ses tâches. Contrairement au contrôle environnemental, la majorité des éléments du système domotique sont alors des capteurs pour agréger l'information qui est ensuite traitée par des centrales de calcul. Les actionneurs présents n'ont souvent qu'un rôle d'avertisseur et non d'exécuteur. Pour illustrer les solutions d'analyse comportementale, on peut citer certaines applications proposées par le laboratoire Domus avec des projets pour lutter contre la maladie d'Alzheimer : un calendrier interactif [11, 12], Archipel un assistant aux recettes de cuisine [13]. La compensation du handicap mental tient donc de l'analyse comportementale des patients. La fusion des données issues des capteurs peut requérir de grosses capacités de calcul pour le traitement de l'information. Ces unités de traitement ont un coût non négligeable pour le patient et la notion de scalabilité nous permet de prendre en compte les ressources déjà présentes dans l'environnement.

Les maisons intelligentes et autres living lab s'inscrivent dans un domaine de recherche particulier : L'Ambient Assisted Living (AAL), en français l'assistance ambiante aux personnes dépendantes. L'AAL a pour objectif d'augmenter la qualité de vie des personnes dépendantes en utilisant les technologies de l'information et de la communication. L'AAL est un vaste sujet de recherche se découpant en deux axes. Le premier axe issu de l'AAL concerne les outils et techniques qui soutiennent l'AAL et lui permettent d'exister. Il s'agit en d'autres termes de la partie support et infrastructure qui comprend les maisons intelligentes, les capteurs mobiles et portables [14, 15, 16], ainsi que l'assistance robotique [17, 18, 19, 20]. Le second axe de recherche concerne les algorithmes et applications dédiées à l'aide au handicap et traite des sujets comme la reconnaissance d'activité [21, 22, 23, 24, 25], la détection d'anomalie [26, 27, 28, 29, 30], la localisation et l'identification d'identité [31, 32, 33], etc.

Nous observons que les solutions d'assistance ambiante aux personnes dépendantes représentent un vaste sujet, c'est pourquoi nous nous proposons de faire une revue rapide, dans la section 2.4, des solutions AAL et particulièrement des habitats intelligents les plus connus et ceux qui ont fortement influencés ces travaux.

### **1.3 Revue des principaux habitats intelligents et de leurs solutions d'aide au handicap**

Nous avons vu dans la section précédente que l'AAL est un domaine de recherche immense. Plusieurs auteurs ont tenté d'établir une cartographie des nombreux projets [34, 35, 36]. Les projets d'AAL sont souvent classés suivant leur axe de recherche : habitats intelligents, capteurs mobiles et portables ou assistance robotique. Les habitats intelligents représentent un support indispensable à la recherche et au développement d'applications dans le domaine de l'AAL. Aussi nous nous consacrons, sans être exhaustifs dans cette partie, à l'étude de ces laboratoires maison et de leurs algorithmes qui ont eu un impact important dans la littérature et sur la maison de demain pour l'assistance des personnes

âgées et des personnes dépendantes. Dans un premier temps, nous étudions les projets les plus influents à travers le monde que nous regrouperons en trois zones géographiques. Puis nous présentons les solutions proposées au sein du laboratoire Lab-STICC et pour finir les solutions proposées par notre équipe de recherche ces dernières années, qui ont inspiré nos travaux.

### 1.3.1 Les habitats intelligents à travers le monde

Le continent nord-américain est l'un des plus actifs dans le domaine et regroupe les habitats intelligents parmi les plus connus. House\_n au MIT (*Massachusetts Institute of Technology*) est annoncée comme «The House of the future». Les recherches menées par l'équipe de House\_n sont pluridisciplinaires et tentent de décrire la maison de demain [37, 38]. Utilisant des centaines de capteurs distribués dans toutes les parties de la maison, ils tentent de mesurer le comportement des patients au sein de leur environnement et comment la technologie influe sur leur comportement [39]. « The GatorTech Smart House » de l'Université de Floride est un autre laboratoire d'expérimentation [40] exploitant les capacités d'appareils intelligents. La porte est ainsi équipée pour lire des tags RFID et les sols de la maison cachent des capteurs de pression qui permettent l'identification et la localisation des personnes. La reconnaissance d'activité [41, 42] et la gestion des équipements [43] sont les principaux axes de recherche. Le projet MavHome de l'Université du Texas cherche à maximiser le confort de ses habitants en minimisant la charge cognitive par l'intermédiaire d'agents logiciels [44]. Les agents logiciels développés tentent de mesurer et de prédire les habitudes des utilisateurs ainsi que l'utilisation des éléments électriques pour favoriser l'automatisation des tâches [45]. MavHome utilise conjointement plusieurs technologies : les bases de données, l'intelligence artificielle, les environnements pervasifs et la robotique. « Aware Home » est un projet du Georgia Institute of Technology qui utilise un ensemble de capteurs, mais aussi les robots d'assistance pour surveiller et aider les personnes âgées [46, 47]. Au Canada, le laboratoire Domus focalise ses recherches sur l'assistance des patients atteints de déficiences cognitives. L'analyse des activités de vie quotidienne [48, 49] est une partie importante des recherches menées par le Domus puisqu'elle permet ensuite le développement de projets adaptés à ces activités comme un calendrier interactif [11, 12] ou un assistant à la cuisine [50].

Le continent asiatique est aussi très dynamique dans les habitats intelligents. Au Japon, on ne compte pas moins de 15 habitats intelligents en développement. Ces projets cherchent à maximiser l'utilisation des technologies d'assistance pour permettre à la population âgée de vivre le plus longtemps à domicile en créant un environnement confortable et bienveillant. L'un des projets les plus populaires est celui initié par le ministère de l'Industrie et du Commerce japonais qui a ordonné la construction de plusieurs « Welfare Techno House ». Des environnements où cohabitent les patients et aides-soignants et où sont mesurés notamment l'activité cardiaque, le poids et d'autres variables vitales pour s'assurer de l'état de santé [51, 52]. Ubiquitous Home Project est un autre projet japonais d'habitat intelligent dans lequel sont utilisés des capteurs infrarouges passifs, des caméras, des microphones, ou encore des tags RFID afin de mesurer l'activité et surveiller les personnes âgées [53, 54]. Enfin, le projet le plus connu du Japon reste celui initié par le Professeur Matsuoka à Osaka où le but est de détecter les anomalies de comportement ou accidents au travers d'une centaine de capteurs présents dans l'environnement. Le principe est de multiplier le nombre de capteurs simples et de fusionner les informations pour mettre en évidence des schémas d'activité [55].

En Europe, l'objectif principal des habitats intelligents est de favoriser les conditions de vie des personnes âgées à domicile. « CarerNet » tente de déployer « l'hôpital à la maison » [56]. L'équipe de recherche a mis en place des services télémédicaux, c'est-à-dire qu'ils proposent des mesures et des tests à domicile dont les résultats sont transmis aux médecins compétents sans que ces derniers se déplacent. Le projet « ENABLE » se concentre sur des sujets souffrants de démences modérées. Ils cherchent à démontrer l'impact positif des technologies d'assistance sur la qualité de vie des patients [57, 58]. En France, nous retrouvons le projet PROSAFE [59], dirigé par le LAAS de Toulouse, qui fait appel à un apprentissage par réseau de neurones pour évaluer l'environnement et déclencher des alertes automatiquement en cas d'anomalie. Nous pouvons aussi citer, entre autres, les projets GERHOME [60] un projet de conception, d'expérimentation et de certification des solutions techniques pour le maintien à domicile des personnes âgées ou encore ERGDOM [61] qui met en place des outils favorisant le confort et diminuant les dépenses énergétiques des patients en mesurant leurs habitudes.

### 1.3.2 Les solutions développées au Lab-STICC

Le Lab-STICC est un laboratoire multisites qui regroupe 450 chercheurs dispersés sur plusieurs établissements comme Télécom Bretagne et l'Université de Bretagne Occidentale à Brest, ainsi que l'Ecole Nationale Supérieure d'Ingénieurs de Bretagne-Sud et l'Université de Bretagne-Sud, dans les villes de Vannes et Lorient. Le Lab-STICC est organisé en trois pôles de recherche MOM (Micro-Ondes et Matériaux), CACS (Communications, Algorithmes, Circuits et Systèmes) et CID (Connaissance, Information, Décision). En 2011, sous l'impulsion d'André Thépaut et Jean-Luc Philippe, une action interpôle est créée. Cette action nommée HAAL pour Human Ambient Assisted Living, regroupe toutes les équipes des pôles de recherche du Lab-STICC qui ont pour vocation l'assistance aux personnes dépendantes :

- Adaptation des services à l'environnement (Équipes IHSEV, DECIDE, SFIIS)
- Interaction humain système (équipe IHSEV)
- Électronique pour l'aide aux personnes (Équipe MOCS)

#### 1.3.2.1 Projets sur l'habitat intelligent de l'action HAAL

Si nous traitons à part entière dans la section suivante les projets de recherche qui se sont tenus au Lab-STICC de Lorient dans le cadre de l'équipe MOCS, il convient de rappeler les projets importants qui ont lieu ou ont été développés au sein du Lab-STICC à l'échelle régionale dans les équipes IHSEV du pôle CID.

Les équipes HAAL de Telecom Bretagne déploient, en ce moment à Kerpape, un bus domotique nommé xAAL [62, 63]. Chaque vendeur de solutions domotique (LeGrand, KNX, EnOcean, Zwave, etc.) possède un protocole de communication spécifique qui lui est propre. En effet, il n'est pas possible d'utiliser un interrupteur LeGrand pour piloter un volet DeltaDore. De ce constat résulte un problème d'interopérabilité entre les éléments domotiques que les équipes HAAL de Télécom Bretagne cherchent à résoudre. S'il existe des protocoles fonctionnels comme UPnP, xAP ou encore xPL, ces derniers proposent des mécanismes d'interopérabilité complexes ou insuffisants (découverte de nouveaux appareils non disponible). Or le protocole xAAL, dont l'architecture est décrite sur la figure 1.5,

se veut un langage de communication universel simple et non un nouveau protocole de plus haut niveau apportant plus de fonctionnalité que la communication simple entre les différents éléments domotiques.

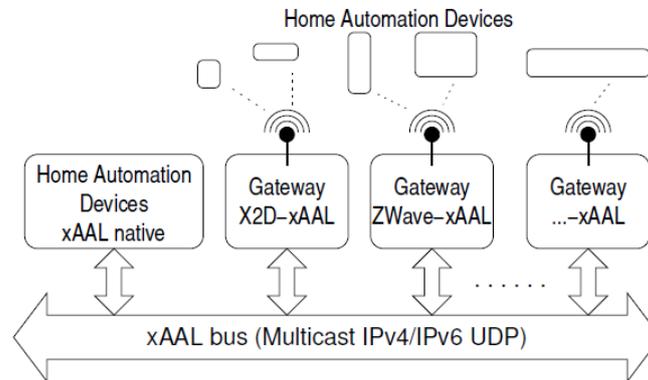


FIGURE 1.5 – Architecture du bus xAAL

Le système xAAL est basé sur un bus IP (Internet Protocole) et permet, notamment, les communications natives entre certains appareils domotiques, mais aussi entre les passerelles (gateway) de l'environnement domotique. Les passerelles se chargent ensuite de transmettre l'information, dans leur propre langage, aux composants qu'elles dirigent.

Parmi les autres travaux de recherche nous pouvons citer à Brest, Le projet SIGAAL [64, 65] cible la population des personnes âgées, et plus particulièrement, les personnes âgées en situation d'isolement. Le projet SIGAAL a pour but de limiter la solitude des personnes âgées en leur apportant les moyens technologiques pour maintenir ou renforcer le lien social mais aussi de prévenir des fragilités ou des pathologies naissantes en repérant des ruptures dans les habitudes de vie. Par l'intermédiaire des téléviseurs intelligents, de plus en plus répandus, le projet SIGAAL propose d'apporter des services :

- Informationnels comme l'accès aux flux d'actualités ciblés et personnalisés, favorisant le maintien de la personne dans son tissu social local.
- Relationnels comme la vocalisation sur l'écran de télévision pour faciliter la rédaction de SMS ou de courriels, entretenant et renforçant les liens familiaux.
- Ambiants comme la gestion des automatismes et des équipements du domicile.

### 1.3.2.2 Projet sur l'habitat intelligent de l'équipe MOCS

L'équipe MOCS, Méthodes, *Outils pour Circuits et Systèmes*, est une équipe de recherche se trouvant dans le pôle de recherche CACS et dont nous faisons partie. L'équipe MOCS dont les travaux de recherche sont pluridisciplinaires possède un axe nommé Systèmes Pervasifs. Le groupe de travail *Systèmes Pervasifs* se consacre à l'étude de systèmes ambiants notamment dans l'habitat pour favoriser le maintien à domicile. Les sujets abordés y sont variés et traitent aussi bien de flot de conception, pour favoriser le déploiement

de systèmes d'aide au handicap, que de l'application de solution domotique pour le contrôle environnemental du patient.

Notre équipe de recherche s'inscrit dans un environnement propice, puisque comme nous l'avons décrit dans la section précédente, la Bretagne est une région riche en recherche sur la thématique de l'aide au handicap, mais plus particulièrement parce que le département du Morbihan possède l'un des plus grands centres pour les personnes souffrant de handicap physique, le Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelle (CMRRF) de Kerpape. Le CMRRF de Kerpape possède une culture de la recherche fortement développée, menée par son laboratoire d'électronique, et propose constamment des collaborations avec le laboratoire LabSTICC, mais aussi des projets étudiants avec l'Ecole Nationale Supérieure d'Ingénieur de Bretagne-Sud et la faculté des sciences de l'Université de Bretagne-Sud. La région Bretagne et le Conseil Général du Morbihan soutiennent activement les problématiques d'aide au handicap et de maintien à domicile en participant au financement depuis plusieurs années des projets de l'équipe MOCS et notamment ces travaux de recherche.

Depuis 2003, pas moins de quatre thèses sur la thématique de l'aide au handicap ont été lancées au LabSTICC de Lorient. Les travaux d'Aïssam Belabbas (COHAND) [66] ont amorcé cet axe de recherche. Aïssam Belabbas a énoncé les premières définitions des systèmes domotiques pour l'aide au handicap. Quelques années plus tard, Saïd Lankri (DANAH) a conçu une plateforme d'expérimentation domotique DANAH [67], en reprenant les évolutions des projets COHAND et QuatrA, et en développant un middleware qui a été le support des travaux de recherche de Thanh Truong, entre autres. Thanh Truong (AIDOM) [68] s'est donc appuyée sur la plateforme domotique du laboratoire pour mesurer de manière passive l'activité du patient au travers des informations domotiques. Enfin, IntelHome est le nom de la thèse menée par Willy Allègre [69], et dont les recherches s'intègrent dans la continuité des travaux de DANAH et AIDOM.

Nous décrivons, dans la suite, les projets marquants de l'équipe MOCS du LabSTICC, qui ont donné lieu à nos travaux de recherche.

**COHAND** Le projet COHAND [70, 71], est la première solution d'aide au handicap étudiée au lab-STICC en collaboration avec le CMRRF de Kerpape. Ce projet, qui s'établit au sein du complexe de Kerpape, cible une population présentant des handicaps physiques et dont les incapacités motrices rendent délicats les déplacements et l'utilisation d'une partie des éléments qui les entourent. COHAND offre la possibilité aux patients de profiter du contrôle environnemental et de la navigation automatisée. Ainsi, le patient peut interagir directement avec un élément de son environnement, COHAND lui proposant une offre de service couplée à sa localisation. La navigation automatisée, basée sur des modèles topologiques de l'environnement, offre au patient la possibilité d'atteindre sa destination de manière autonome en prenant en compte les aléas (obstacles ou échecs de l'exécution de service) que peut rencontrer le système tout au long du déplacement et en proposant des mécanismes de reconfiguration. COHAND a posé les bases d'une solution d'aide au handicap, en apportant les premières définitions d'un système domotique et en abordant les notions de sûreté de fonctionnement, de performances temporelles et d'ergonomie. Une plateforme de simulation a été proposée pour preuve de concept, mais aucune implémentation n'a été mise en œuvre.

**QuatrA** Le projet QuatrA, pour Aide Ambiante Ajustée Automatiquement, peut être vu comme l'application concrète des premières recherches mises en avant avec le projet COHAND. Toujours en collaboration avec le CMRRF de Kerpape, QuatrA qui vise aussi une population handicapée physique s'est focalisé sur le contrôle environnemental, avec pour objectif une plateforme expérimentale déployée au centre de Kerpape [72]. L'offre de service proposée par QuatrA regroupe le contrôle des appareils électriques comme une lampe, une porte, un volet ou un lit médicalisé, mais permet aussi d'interagir avec les équipements multimédias tels qu'une télévision, un téléphone et même un ordinateur. Cette solution originale s'appuie sur une architecture composée d'un ensemble de bornes domotiques fixes dispersées dans l'environnement qui communiquent avec la borne domotique mobile embarquée sur le fauteuil roulant électrique du patient. Le patient utilise un appareil mobile qui possède une interface graphique adaptée en contact sans fils avec la borne du fauteuil roulant. QuatrA propose, grâce à sa plateforme d'expérimentation, une première architecture électronique composée d'éléments de commandes domotiques fixes et mobiles qui sera reprise et simplifiée dans les projets futurs. QuatrA définit aussi les notions de service élémentaire et de scénario qui est une succession fonctionnelle ou temporelle de services élémentaires.

**DANAH** DANAH, qui signifie Domotique, Aide à la Navigation et Assistance au Handicap, aborde l'ensemble des problématiques scientifiques adressées par les projets précédents. DANAH se veut la pierre angulaire de tous les projets du labSTICC et réintroduit les notions de navigation, de qualité de service et de reconfiguration proposée par COHAND. Tout en s'appuyant sur l'architecture domotique, auparavant simplifiée, proposée dans le projet QuatrA, une application middleware pour l'aide au handicap a été développée. Partant du constat que les appareils domotiques utilisent des protocoles de communications différents, DANAH intègre dans son application les mécanismes mettant en œuvre l'interopérabilité des équipements contrôlés. Ainsi, DANAH rassemble autour d'une interface unique une offre de service hétérogène. Le projet DANAH met aussi en évidence les besoins d'une conception facilitée et adaptée au patient. Un premier flot de conception, basé sur le travail des interfaces de communication, est alors introduit. Les modélisations de l'environnement fournies par le flot de conception ont pour objectif de générer automatiquement la configuration personnalisée du système DANAH. Enfin, l'application DANAH met en œuvre les prémices de la reconfiguration de service statique et dynamique [73]. Si tous les aspects de recherche abordés n'ont pas eu la même réussite, DANAH pose des bases importantes autant en matière d'infrastructure électronique que de développement informatique. L'application middleware proposée a été le support de plusieurs projets au LabSTICC et l'architecture domotique, composée de serveurs et clients mobiles, est encore en place aujourd'hui.

**AIDOM** AIDOM est un projet qui s'inscrit en parallèle des travaux sur le contrôle environnemental tenus au LabSTICC. Thanh Truong s'est concentrée sur l'agrégation de données issues de la domotique pour proposer un algorithme pour la reconnaissance d'activité [74] et la proposition automatique de scénario [75]. L'originalité des travaux tient dans le fait que ses algorithmes s'appuient sur les commandes du système domotique, autrement dit les requêtes de service. Le système est ainsi capable de repérer des schémas d'activités sans ajouts d'équipements supplémentaires comme des capteurs particuliers et autres caméras. Les solutions proposées sont dites non intrusives. Un second axe de recherche s'est focalisé sur la détection d'anomalies à partir des habitudes de vie des patients identifiées au préalable [30].

**IntelHome** Willy Allègre a proposé deux solutions originales : la création d'un flot de conception assisté [76] et la supervision non-intrusive du patient [77]. Un DSL (Domain Specific Language) graphique a été développé pour permettre aux personnes, non spécialistes des outils informatiques, mais possédant des compétences indéniables dans le domaine de la prise en charge, de concevoir facilement un système domotique. Un modèle d'environnement et un modèle d'interaction sont ainsi décrits et grâce aux techniques de l'IDM (Ingénierie Dirigée par les Modèles), le flot de conception génère le fichier de configuration du système domotique. La supervision non-intrusive reprend une partie des éléments définis dans les travaux de Thanh Truong. Ainsi, en créant une représentation de l'environnement sous la forme d'une base de connaissances, il est possible d'inférer de nouvelles informations sur le patient, sans avoir recours à l'ajout de capteurs spécifiques considérés comme intrusifs. Les travaux, menés par Willy Allègre, ont donné suite à un projet ASIM [78] soutenu par Vity Technologies une entreprise du bassin Lorientais. ASIM (Assistance par Serveur Intelligent Multi-protocoles) est un projet qui vise la réalisation d'un serveur domotique multiprotocoles pour assister les personnes fragiles et les maintenir à domicile.

## 1.4 Positionnement et motivation

Dans cette section, nous étudions l'évolution de l'informatique ubiquitaire pour ensuite positionner nos travaux. Waldner [79] en 2007, montre les grandes étapes de l'évolution de l'informatique, reprise sur la figure 1.6. Il part de la naissance de l'informatique en 1960 pour mettre en exergue l'évolution rapide de l'informatique au sein de notre société avec l'apparition des communications entre appareils, de la mobilité des équipements et de leur présence qui se fait ubiquiste. Avec l'émergence de la technologie qui rend les appareils numériques présents physiquement partout, mais invisibles ou tout du moins transparents pour les utilisateurs, de nouveaux champs de recherche ont émergé. L'informatique ubiquitaire (ubiquitous computing, en anglais), aussi connu sous les noms d'informatique omniprésente (pervasive computing) ou encore d'intelligence diffuse (ambient intelligence), consiste à faire collaborer une multitude d'appareils indépendants et complexes comme des ordinateurs, des capteurs, des actionneurs et des dispositifs communicants. Le concept a été introduit par Weiser [80] en 1991. Ce dernier entrevoit l'informatique ubiquitaire comme un système combiné d'applications et de données, qui serait en mouvement avec l'utilisateur. Une première vision de l'informatique ubiquitaire s'est consacrée à la migration et l'adaptation automatique des applications à différents contextes et ressources. Cependant, la complexité des systèmes informatiques actuels rend difficiles et coûteux leur gestion et leur déploiement. Pour pallier ces difficultés, les notions d'informatique autonome (autonomic computing) ont été initiées par IBM en 2001 et reprises par Kephart en 2003 [81]. Les concepts de l'informatique autonome tentent de mettre en œuvre des mécanismes afin de faciliter la gestion des applications tout en restant invisibles pour l'utilisateur. Ces concepts sont l'auto-configuration qui par des actions simples permet le déploiement des applications fonction de leur environnement, l'auto-réparation et l'auto-protection qui rendent les systèmes plus sécurisés et fiables ainsi que l'auto-optimisation qui cherche à améliorer les performances. Cette approche de l'informatique autonome a été introduite au sein de systèmes relativement homogènes et dont les capacités de traitement de l'information sont importantes. Or cette vision des choses est à l'extrême opposée des spécifications requises par le domaine de l'AAL et des habitats intelligents. Plusieurs travaux de recherche tentent d'adapter l'informatique autonome au domaine des habitats intelligents.

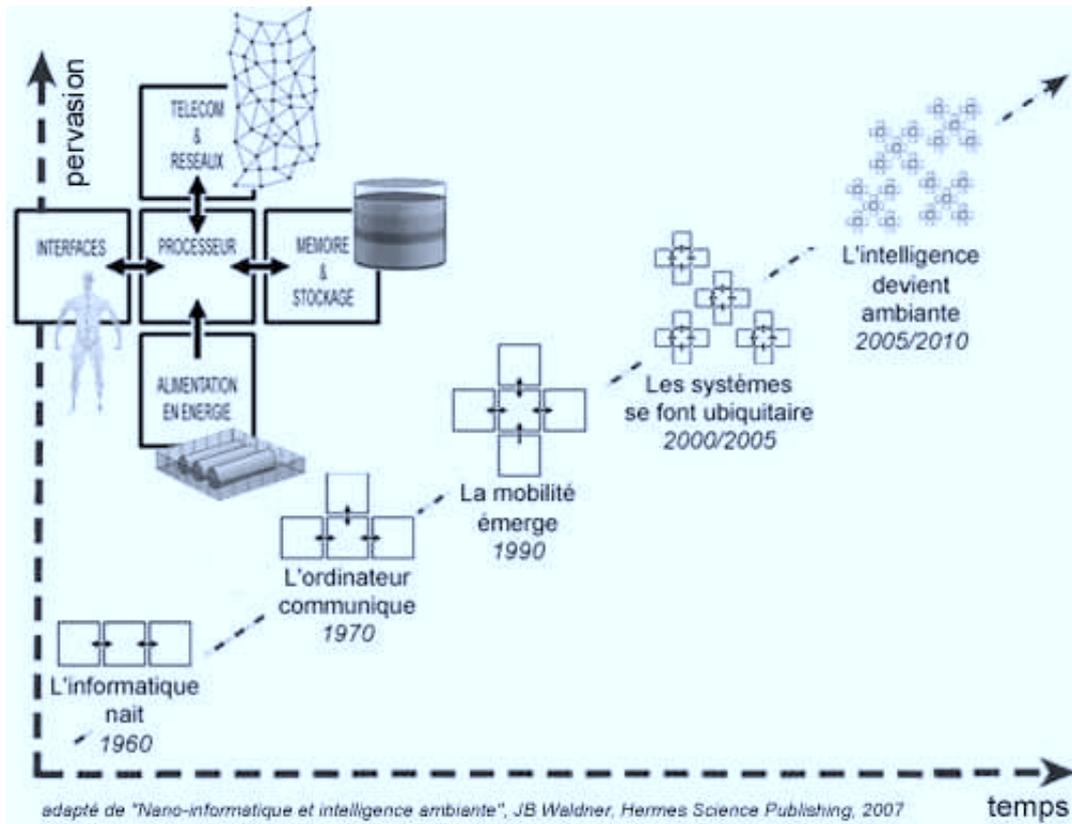


FIGURE 1.6 – Les étapes de l'évolution de l'informatique par JB Waldner

Ranganathan et al. [82], avec le projet Gaia propose des mécanismes d'auto-configuration d'application. Le principe de base est que les applications peuvent changer leur structure pour s'adapter aux différents environnements et récupérer des fautes. Une application est composée d'une entrée (controller), d'une sortie (view) et de composants logiques (model). Par exemple, si l'utilisateur souhaite écouter de la musique, l'application est traduite sous la forme suivante : « windows media player joue le fichier file.wmv sur le haut-parleur de l'ordinateur portable ». Toutefois, des variantes existent comme la forme : "Itunes" joue le fichier file.aac sur le « smartphone ». Cette notion est appelée le polymorphisme d'applications et est basée sur une ontologie et la comparaison sémantique des concepts. Ainsi, si l'utilisateur souhaite exécuter une action, cette dernière est traduite par le système qui liste les concepts susceptibles de répondre aux besoins. Une comparaison sémantique est alors effectuée pour faire émerger une configuration de l'habitat intelligent.

Aly Syed et al. [83] présente un Framework d'auto-configuration de service. Cette solution utilise une base de connaissances où les appareils sont vus comme un ensemble éparpillé physiquement dans l'environnement. Chaque appareil peut réaliser une ou plusieurs sous-fonctions : les services. Pour répondre à la requête d'un utilisateur, le système associe les appareils pour former un équipement virtuel qui peut exécuter l'ensemble de la fonction demandée. Un équipement virtuel est composé de plusieurs appareils pour fournir toutes les sous-fonctions requises. Un appareil peut ainsi faire partie d'un ou plusieurs nœuds virtuels. Le système s'appuie sur une base de connaissances qui regroupe

les recettes, les capacités, des règles et des propriétés. Alors que les capacités permettent de définir les sous-fonctions des appareils participants au système, les recettes décrivent la suite logique de fonctions à réaliser, et les règles définissent le contexte et régissent le système. Les propriétés sont liées aux capacités et définissent la présence de périphériques et de fonctionnalités. Par exemple, pour lire une musique au format mp3, le système compare le contexte de la requête avec les contextes de la base de recettes. Si les conditions sont respectées alors le déploiement est mis en place. Chaque équipement virtuel possède sa base de connaissances. Les principes de construction de la base de données ainsi que le chronogramme d'un exemple sont décrits en Figure 1.7.

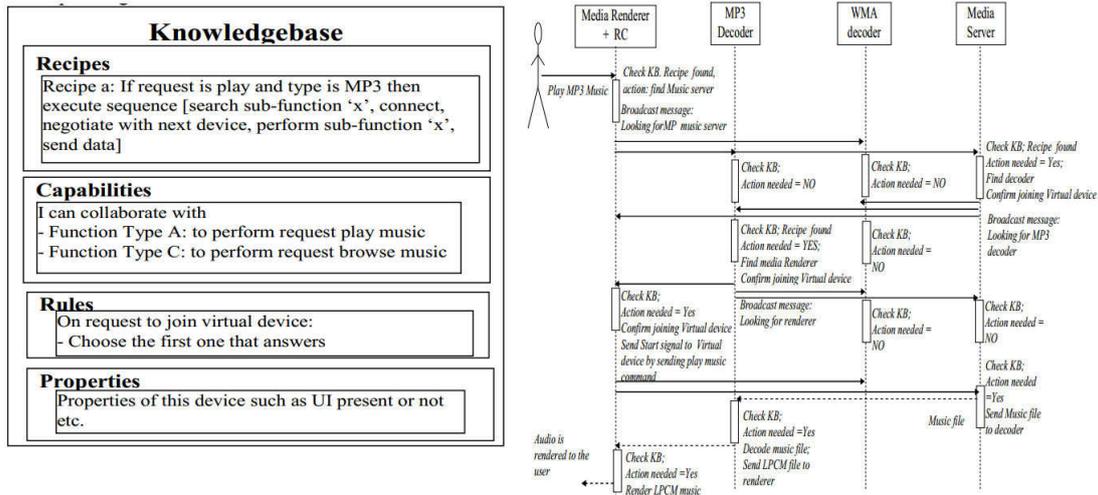


FIGURE 1.7 – Modèle de base de données et diagramme d'exécution d'une application

Gouin-Vallerand et al. [84, 85], propose un projet Tyche, pour l'auto-configuration et le déploiement d'applications dans les espaces ubiquitaires. Ce middleware est dit « context-aware » et s'appuie sur une description ontologique de l'environnement. Dans sa dernière version, les capacités d'interactions ainsi que les préférences d'utilisation sont prises en compte dans la méthode de déploiement de l'application. Pour déployer le service, Tyche requiert quatre éléments contextuels principaux : les profils utilisateurs, les profils des appareils, la topologie de l'environnement et les profils d'application. La modélisation contextuelle exprimée en OWL est associée à des mécanismes de raisonnement par logique floue afin de faire correspondre les besoins des services à l'environnement pervasif. La contribution est nommée FLORE, pour Fuzzy Logic Organization Reasoning Engine, et contient deux niveaux de raisonnement : micro-context et macro-context. Aussi pour déterminer la ressource idéale, un quotient de capacité de l'équipement (DCQ, Device Capabilities Quotients) est calculé à partir des résultats de raisonnement de la logique floue. Enfin, Tyche utilise les techniques des architectures orientées services et notamment la technologie OSGi pour déployer son service dans l'environnement. Pour illustrer ces méthodes, prenons l'exemple d'un patient atteint de déficiences cognitives et dont l'objectif est de se préparer un repas. Le profil du patient indique notamment une faible acuité visuelle, une vitesse moyenne et des préférences pour les interfaces tactiles. Le profil de l'application indique que son espace de déploiement devrait être la cuisine et qu'elle nécessite un écran et une interface d'entrée pour passer les étapes. Par rapport à la topologie de l'espace intelligent, les spécifications de l'application et la position du patient le middleware détecte

quatre appareils disponibles, comme le montre la figure 1.8. Une fois les informations contextuelles soumissionnées, FLORE calcule le DCQ correspondant à chaque appareil. Pour cette application, la tablette tactile de la cuisine est choisie devant l'ordinateur portable en raison de sa proximité avec le patient et de son interface préférentielle.

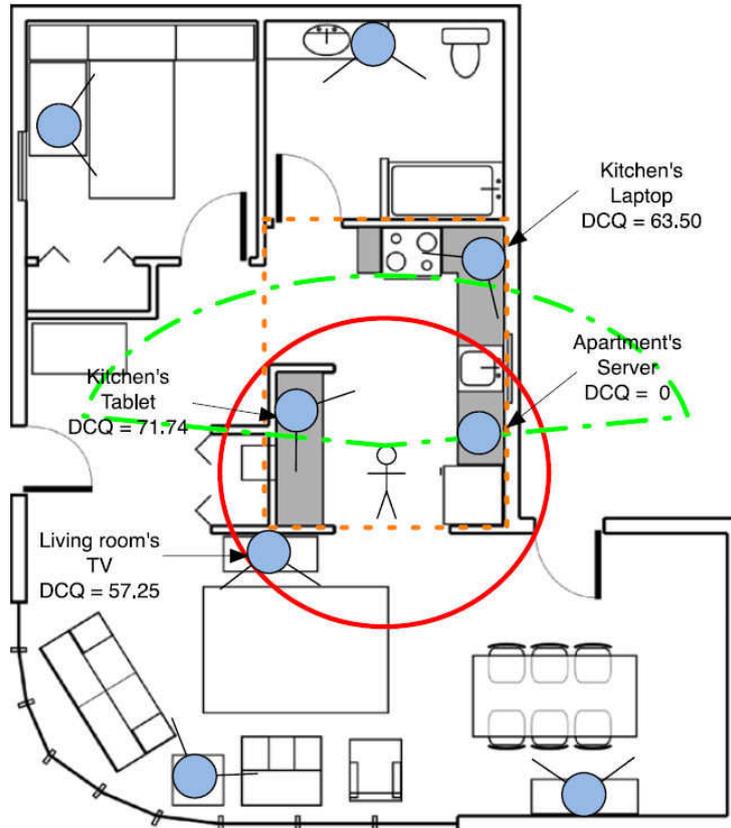


FIGURE 1.8 – Exemple d'un cas de déploiement de service utilisant FLORE.

Parmi les autres projets en rapport avec le déploiement de services dans un espace pervasif, nous pouvons citer Valée et al. [86] qui propose des mécanismes de composition de services. Ces travaux s'inscrivent dans le projet européen AMIGO qui a pour but de développer un Framework d'assistance aux utilisateurs dans les habitats intelligents. Le but recherché par Valée et al. est d'implémenter des techniques pour dynamiquement construire des services basiques dans un environnement ouvert et en évolution permanente. L'utilisateur ne fait appel qu'à des services de haut niveau qui répondent à ses besoins pendant que le système se charge de composer les fonctions correspondantes à ce service, en utilisant une ontologie décrivant le contexte. Ghorbel et al. [87] offre une solution pour fournir à l'utilisateur des services spécifiques au travers de son smartphone, lorsque ce dernier entre dans un environnement pervasif. Notons aussi le projet iPOJO de [88], de l'Université de Grenoble, basé sur les mécanismes d'OSGi qui propose une application de déploiement de service. Ces derniers sont définis en utilisant le modèle de composant orienté service iPOJO.

La littérature nous montre que beaucoup de chercheurs se sont interrogés sur la distribution de services par rapport aux profils et aux préférences des patients. Toutefois,

peu de recherches se sont penchées sur les aspects de distribution de services en fonction des capacités matérielles. Nous positionnons, nos travaux à un niveau de description de l'environnement inférieur à ce que nous avons étudié précédemment. Sans pour autant négliger les besoins du patient, nous cherchons à déployer le service sur un ensemble de machines pour soulager le système, cela même si l'opération n'est pas la plus optimale pour l'utilisateur. Alors que Ranganathan et al. et Syed et al. déterminent la meilleure ressource par des comparaisons sémantiques, aucun mécanisme de mesure du taux d'utilisation de l'appareil n'est pris en compte. À l'inverse, Gouin-Vallerand et al. calculent un quotient des capacités de l'équipement pour déterminer la meilleure ressource en prenant compte de son taux d'utilisation. Toutefois, la priorité est donnée aux profils et aux préférences du patient, ce qui « fausse » le calcul de charge et minimise ainsi l'impact du taux d'utilisation des ressources. Notre approche est basée sur une description précise et fidèle des composants. De plus, aucun des travaux présentés n'intègre la gestion continue des services entre eux. En effet, chaque service est considéré comme une entité indépendante de la précédente. Dans notre cas, nous considérons notre middleware comme une sorte de système d'exploitation qui offre la concurrence des services d'aide au handicap, sur plusieurs équipements informatiques. Aussi, le taux d'utilisation des capacités est pris en compte au même titre que les capacités à utilisation « unique ». Si un service, comme la lecture d'un film, occupe déjà l'équipement choisit pour installer la nouvelle tâche des mécanismes de gestion ont été développés pour analyser la priorité et la préemption des dits services et exploiter des techniques d'ordonnancement ou de services alternatifs pour favoriser le déploiement. Notre approche s'appuie sur deux phases complémentaires, représentées sur la figure 1.9. La partie de modélisation du contexte est étudiée dans le chapitre 2 et la partie application middleware est décrite dans le chapitre 3.

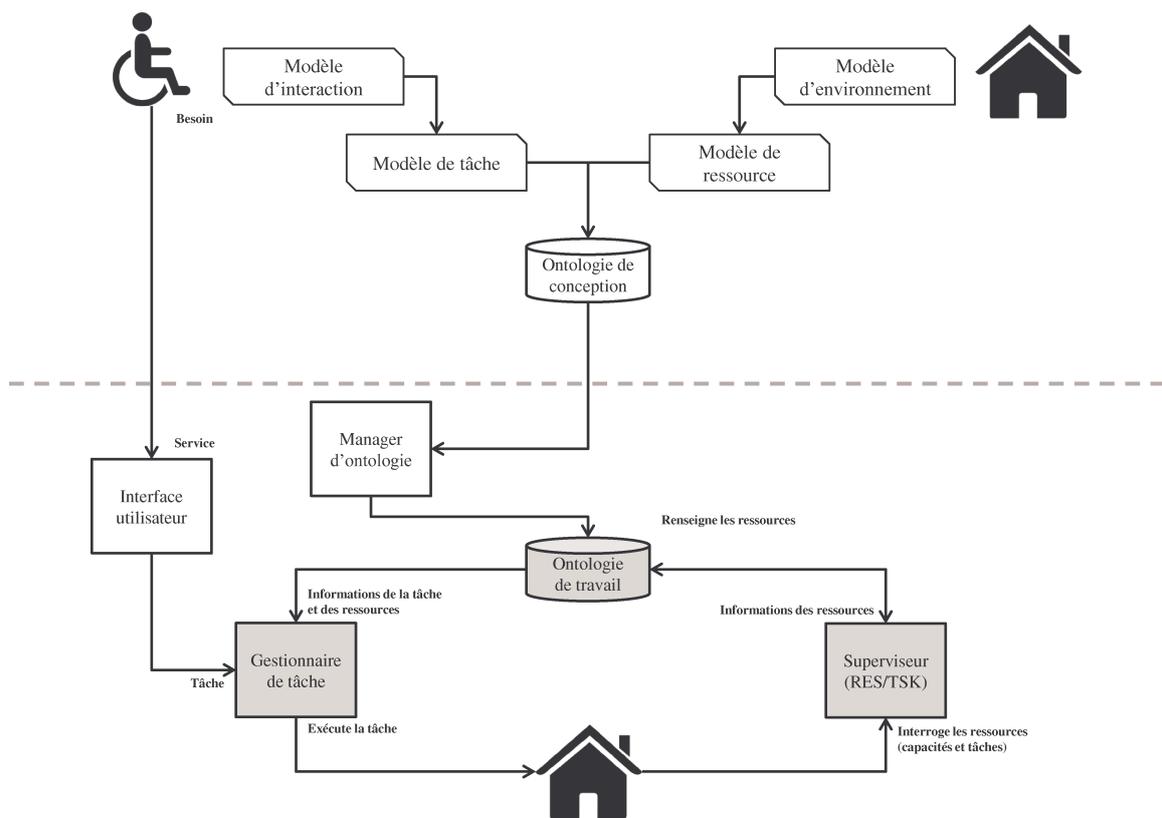


FIGURE 1.9 – Schéma global des travaux : conception et exécution

## Chapitre 2

# Modélisation d'environnement intelligent

### Sommaire

---

<b>2.1</b>	<b>Introduction à la conception . . . . .</b>	<b>35</b>
<b>2.2</b>	<b>Modélisation de l'informatique ubiquitaire . . . . .</b>	<b>39</b>
2.2.1	Du service d'aide au handicap à la tâche . . . . .	39
2.2.1.1	Définition d'un service . . . . .	39
2.2.1.2	La naissance de la notion de tâche . . . . .	40
2.2.1.3	L'offre de service d'aide au handicap . . . . .	42
2.2.1.4	Modélisation d'une tâche d'aide au handicap . . . . .	43
2.2.2	De l'équipement domotique à la ressource . . . . .	44
2.2.2.1	Définition d'une ressource . . . . .	45
2.2.2.2	L'appareillage domotique . . . . .	45
2.2.2.3	Associations de ressources . . . . .	46
2.2.2.4	Modélisation d'une ressource . . . . .	47
2.2.3	La capacité, lien entre la tâche et la ressource . . . . .	48
<b>2.3</b>	<b>Ontologie de l'environnement domotique . . . . .</b>	<b>50</b>
2.3.1	État de l'art et positionnement . . . . .	51
2.3.2	Introduction aux ontologies . . . . .	52
2.3.3	Notre ontologie d'environnement . . . . .	55
2.3.3.1	Construction d'un environnement domotique . . . . .	55
2.3.3.2	Héritage de capacité . . . . .	57
2.3.3.3	Les tâches pour compléter l'ontologie . . . . .	59
2.3.3.4	La recherche de ressource pour exécuter une tâche . . . . .	59

---



Nos travaux se divisent en deux parties distinctes : la conception du contexte sous forme d'une ontologie le représentant et l'exécution des services domotiques, dans l'environnement, au travers d'une application middleware. Si les travaux sur l'exécution représentent la partie visible de l'iceberg, il convient de prendre le temps d'étudier la partie conception, dont l'intérêt tient dans la représentation originale que nous proposons de l'environnement domotique. Dans ce chapitre, nous allons étudier la première partie de nos travaux : la modélisation de contexte (context modeling, en anglais). En effet, la connaissance de son contexte (context awareness, en anglais) permet aux systèmes diffus de s'adapter à leur environnement. Par exemple, connaître le profil du patient, ses capacités et incapacités ou encore ses préférences donnent la possibilité de détecter des habitudes de vie ou des anomalies de comportement. Une connaissance de la topologie de l'environnement permet d'offrir des services en lien avec la localisation du patient. Enfin, connaître les appareils domotiques et leurs applications facilite une configuration adéquate du système. Il existe autant de contextes spécifiques que d'applications dédiées. Dans ce chapitre, nous abordons la partie conception et modèles qui nous permet de décrire l'environnement domotique. Dans un premier temps, nous introduisons la notion de conception. Puis, nous présentons les différents éléments de l'informatique ubiquitaire que nous retenons pour notre modélisation. Enfin, nous présentons notre modélisation ontologique du contexte dans la troisième partie, de ce chapitre.

## 2.1 Introduction à la conception

Il faut bien différencier un système domotique pour utilisateur technophile, grand public, d'une solution pour un patient en situation de handicap. Si les solutions de domotique grand public ont pour objectif de faciliter le quotidien de leurs utilisateurs, les besoins de ces derniers ne sont pas spécifiques, puisqu'ils ne possèdent pas d'incapacités, et que les solutions proposées sont adaptées pour couvrir les exigences générales. C'est l'effet de marché ou l'objectif est d'atteindre un maximum de clientèle en minimisant les coûts de développement. Or dans le monde du handicap, les priorités sont toutes autres. Ici, l'objectif principal n'est pas de toucher le plus grand nombre de personnes, mais de fournir des réponses adaptées à leurs demandes. Que le handicap soit d'ordre physique, mental, ou encore une combinaison des deux pour les personnes plurihandicapées, aucune situation n'est identique et le système domotique pour compenser les incapacités doit pouvoir être ajustable. L'adaptation est effectuée tant du côté des spécifications des besoins du patient que du côté de son espace de vie et des équipements présents, dont la pluralité et la disposition, là aussi, sont uniques d'un individu à un autre. On parlera d'offre de service pour décrire les mécanismes apportés en réponse aux besoins du patient. Les services sont une combinaison d'une opération comme allumer ou éteindre et de l'équipement sur lequel il repose, ici, une lampe. Nous venons de l'aborder rapidement, mais l'interopérabilité des équipements est un second verrou des espaces pervasifs. Il existe une multitude d'appareils domotiques, informatiques ou encore dits intelligents et tout autant d'interfaces et de protocoles de communication. La notion d'interopérabilité tient donc dans la capacité d'un système à gérer un ensemble de ressources hétérogènes, mais aussi d'apporter une solution unifiée pour la commande de ces équipements. La mise en place de descriptions de haut niveau permet de répondre en partie au problème. En outre, il faut aussi noter les aspects de confort d'utilisation bien souvent évalués par la qualité de service et le temps de réaction du système. Tous ces aspects peuvent conditionner l'acceptation ou le rejet d'une solution domotique d'aide au handicap.

Grâce aux collaborations avec le CMRRF<sup>1</sup> de Kerpape et les échanges avec l'équipe médicale, notamment leurs ergothérapeutes et leurs ingénieurs, la notion de la conception d'une solution domotique pour l'aide au handicap appropriée est apparue comme une évidence. À titre d'exemple, on peut citer les travaux réalisés dans le projet DANAH dont la partie sur la conception a mis en exergue le besoin d'un flot de conception et proposé une première approche. Nous pouvons surtout mentionner plus récemment les travaux de Willy Allègre dans le projet IntelHome qui a apporté les outils nécessaires à la conception du système domotique désiré. Basée sur l'ingénierie des modèles, l'approche proposée dans IntelHome permet d'offrir une simplicité de conception à des membres de l'équipe médicale, non-ingénieurs/développeurs, mais qui possèdent de vraies compétences dans la prise en charge des personnes dépendantes. Le flot de conception proposé dans IntelHome est basé sur la spécification des besoins du patient et génère le fichier de conception spécifique à la plateforme domotique ainsi qu'une ontologie de service décrivant l'offre de disponible pour le patient. L'offre de service proposée est l'un des points de départ de nos travaux. Nous ne prétendons pas modifier les aspects d'expressions des besoins des patients et de leur offre de services domotiques mais souhaitons tirer profit de l'exécution de ces services sur l'infrastructure matérielle. Nos travaux sont donc une reprise de ceux réalisés dans IntelHome en ajoutant de nouveaux modèles pour optimiser le déploiement de ces services à bas niveau. En d'autres termes, Si l'individualisation, l'adaptabilité et l'interopérabilité notamment, sont parfaitement abordées et traitées dans IntelHome nous nous intéressons plus particulièrement aux critères de fiabilité, de réactivité, d'intrusivité et de coût de la solution domotique d'aide au handicap.

- Critère du coût : Le coût de la solution domotique prend encore plus d'importance qu'auprès du grand public puisque ce n'est pas par choix, mais par nécessité que le patient doit retenir un système domotique. De plus, comme nous l'avons vu dans le Chapitre 1, les personnes en situation de handicap accusent doublement le coût de ce dernier puisqu'ils assument les frais médicaux et de réadaptation, mais aussi peuvent subir une perte substantielle de revenu qui les place dans une situation précaire. La solution domotique doit donc être en accord avec ce principe et par conséquent minimiser le coût de son installation. Pourtant, toute installation d'un système domotique que ce soit en termes de contrôle de l'environnement ou en termes de traitements et analyse d'information possède des coûts non négligeables. Une première approche pourrait être la réutilisation du matériel informatique et domotique déjà présent dans l'espace de vie, avant d'ajouter des appareils dédiés onéreux.
- Critère d'intrusivité : Si le but d'une solution d'aide au handicap est de proposer au patient l'autonomie et l'indépendance qu'il avait perdues, ce n'est pas en échange d'un système domotique intrusif. Par exemple, bien que permettant de nombreux traitements d'image, une caméra sera difficile à installer chez le patient. Toutefois, on ne peut pas rendre le système complètement non intrusif. Des données sur le patient sont acquises en permanence, par un ensemble de capteurs éparpillés dans le logement. Toujours présent, l'aspect intrusif de la solution n'est pas pour autant à négliger et il faudra en mesurer l'impact chez le patient. Une solution pour limiter l'intrusivité serait de traiter le plus de données localement pour ne pas que des informations personnelles ne puissent être interceptées. Cependant, certaines informations pourront être

---

1. Centre Mutualiste de Rééducation et de Réadaptation Fonctionnelles

envoyées aux personnes compétentes, comme un médecin pour la vérification des fonctions vitales et des mécanismes de sécurité devront être mis en place, mais ce n'est pas le sujet de ces travaux.

- Critère de réactivité : Avec les progrès scientifiques et techniques dans le domaine du handicap, nous observons l'arrivée de nouveaux services, souvent associés à l'analyse comportementale. Nous pouvons citer les travaux sur la reconnaissance d'activités quotidiennes (ADL, Activity of Daily Living). Domaine très riche, il existe de nombreux projets abordant le sujet, comme nous l'avons résumé dans le chapitre 1. Nous prendrons, pour exemple, les algorithmes de détection de chute [89, 90, 91], pour ne parler que d'eux dont les résultats sont probants, mais dont les traitements sont parmi les plus consommateurs en ressources de calcul. Tous ces projets ambitieux de par leur recherche ont un point commun : un besoin conséquent en capacité de traitement. Or, si le patient requiert un service et que la surcharge de calcul introduite par d'autres traitements retarde son exécution, la solution risque de ne pas convenir au patient.
- Critère de fiabilité : Le critère de fiabilité est crucial pour l'acceptation ou le rejet d'une solution domotique. Dans un premier temps, il évalue la qualité du service proposé. En effet, le patient possède une certaine attente vis-à-vis du bon fonctionnement de son système domotique. S'il requiert l'ouverture d'une porte et que cette dernière reste bloquée, la situation résultante peut être dangereuse et même dramatique. Enfin, le système domotique permet de s'assurer de l'intégrité de l'exécution des services, fonction des aléas rencontrés, en proposant des états dégradés du service ou des fonctions alternatives. Il doit donc garantir la sûreté de fonctionnement en toutes conditions. L'évaluation de ces deux points influe grandement sur l'acceptation ou le rejet d'une solution domotique d'aide au handicap.

En nous inspirant des travaux sur la répartition de la charge de calcul et les bases de connaissances nous souhaitons proposer des mécanismes de déploiement optimisés. Pour faire le lien avec les travaux apportés par le projet IntelHome, nous partons de l'offre de service construite par les personnes compétentes et proposons de gérer leur exécution de manière fine en observant l'intégralité du système domotique et de la disponibilité de ses équipements.

La notion de service est ici remplacée par la notion de tâche. Pour résumer, le service est alors décomposé comme un ensemble de fonctions informatiques avec des besoins matériels et logiciels qui lui sont propres. De la même manière nous utilisons la notion de ressource pour exprimer l'idée d'un équipement domotique dont les capacités matérielles et logicielles sont mises en avant. Le but final étant de pouvoir faire coïncider les besoins de tâches avec les capacités des ressources pour faciliter le déploiement des services. Tout en gardant à l'esprit les besoins et spécifications des patients, nous accordons donc une attention particulière aux éléments matériels et à l'exécution des tâches d'aide au handicap sur l'infrastructure domotique.

La figure 2.1 montre les interactions entre les modèles d'environnement et d'interactions proposées dans IntelHome et nos modèles de tâches et de ressources. En d'autres termes, le point de départ de nos travaux est la modélisation proposée par IntelHome, représentée sur la partie supérieure de la figure 1. Les besoins des patients sont décrits à l'aide du modèle d'interaction tandis que les infrastructures sont représentées grâce au

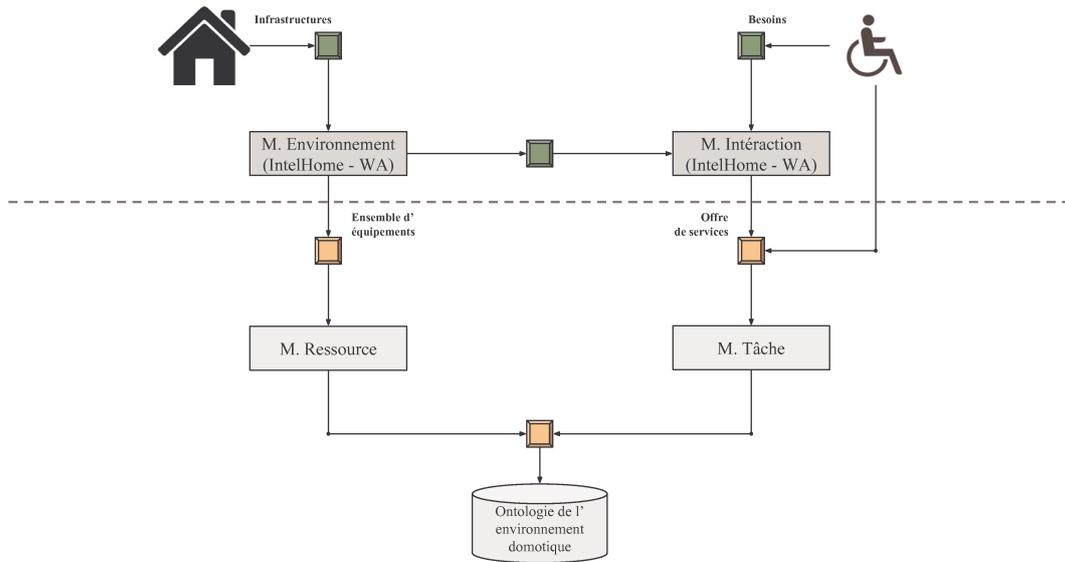


FIGURE 2.1 – Positionnement par rapport aux travaux précédents

modèle d'environnement. Un raffinement est effectué au niveau de la description des ressources pour restituer plus fidèlement leurs propriétés et capacités physiques intrinsèques. Les interactions alors considérées comme des opérations sont ainsi exprimées, dans notre modélisation, comme des tâches informatiques. Chaque application possède des besoins informatiques propres pour être exécutée. Les besoins requis par ces tâches informatiques peuvent ainsi être comblés par les capacités des ressources. Toutes ces informations sont contenues dans la base de connaissances représentée sous la forme d'une ontologie.

Prenons l'exemple d'un service « allumer la lumière » décrit sur la figure 2.2. Une fois le service modélisé sous forme d'une tâche, nous mettons en avant les besoins de cette dernière. Pour faciliter l'explication, nous faisons abstraction des besoins logiciels et nous nous concentrons sur les besoins matériels de ces tâches. De plus, les besoins et capacités sont donnés ici à titre d'exemple, nous y reviendrons plus précisément dans la suite du document. Nous pouvons noter que pour être exécuter cette tâche nécessitera des besoins de traitement, de commande et d'actionneur. Chacun de ces besoins devra alors être comblé par une ou plusieurs ressources de l'infrastructure. Dans le cas contraire, notre infrastructure où ne proposerait pas les capacités suffisante, la tâche, et à plus haut niveau le service, ne serait pas en mesure d'être réalisé.

Nous montrons ici le besoin, d'avoir une connaissance de l'environnement domotique et de ses éléments suffisamment bas niveau, pour pouvoir obtenir un déploiement fiable des services. Or le système va évoluer avec le temps, que se soit d'un point de vue charge ou disponibilité de ces équipements nous devons donc mettre en place une modélisation évolutive et intelligente. C'est ce que nous cherchons à faire avec les ontologies, que nous introduisons dans la section 2.3, en modélisant ainsi notre environnement domotique.

Dans la suite de ce chapitre, nous analyserons, en section 2.2, les modèles qui nous permettent de représenter les ressources, les services et les capacités de l'espace personnel du patient. Puis dans la section 2.3, nous introduirons les ontologies, une représentation interprétable informatiquement et capable de réaliser des traitements facilitant les interactions

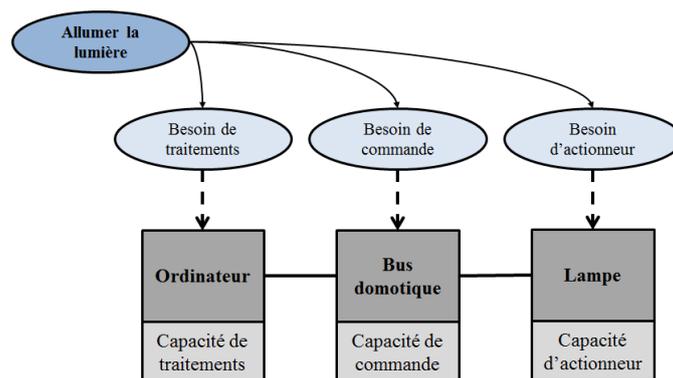


FIGURE 2.2 – Recherche d’un couple tâche - ressource

entre les éléments qui la compose.

## 2.2 Modélisation de l’informatique ubiquitaire

### 2.2.1 Du service d’aide au handicap à la tâche

Si dans la littérature ce sont les interactions avec les équipements qui forment l’offre de service au patient, nous souhaitons mettre en évidence l’indépendance des tâches vis-à-vis des ressources. Par exemple, le service « allumer la lumière » correspond à la fonction de passer à l’état 1 l’équipement lampe. Nous considérons que la tâche « allumer la lumière » est une application qui se charge, en priorité, de passer la lampe correspondante à l’état actif, mais qui peut aussi en cas d’aléas ou de besoin passer n’importe quelle autre ressource lampe à l’état 1. La tâche est donc une notion héritée du service. Alors que le service domotique renvoie à la description spécifique d’un besoin du patient, la tâche est l’application informatique directe de ce service. Avant d’aller plus loin dans la description, définissons ce qu’est un service d’aide au handicap, appelé parfois service domotique par abus de langage.

#### 2.2.1.1 Définition d’un service

*Au sens commun du terme* : Le sens primaire d’un service est l’idée d’obtenir, par une tierce personne, une action, un travail, dont nous n’avons pas le temps ou les connaissances pour le faire nous-mêmes. D’ailleurs, la locution « Peux-tu me rendre un service ? » renvoie à la notion de faire réaliser une opération désirée à notre interlocuteur. Le service public est une administration française destinée à satisfaire un besoin d’intérêt général, comme l’enseignement, la santé, la culture dans un contexte non marchand, par exemple. Il se peut que le service même public soit assuré dans un contexte commercial. Nous pouvons entre autres citer les services liés à l’énergie, au transport ou encore les communications. Toutefois, le service prend un autre sens dans le contexte mercantile ou il sera rendu en échange d’une contrepartie. C’est le cas d’une compagnie de taxi qui proposera le service de transporter des personnes, mais aussi les banques qui proposeront de gérer vos économies.

*Au sens, aide au handicap* : La définition du service reste la même dans le contexte de l’aide au handicap, il pourra être bénévole ou rétribué. Cette définition est particulière puisqu’elle répond à des besoins spécifiques qui lui sont propres. Dans le monde de l’aide

au handicap, comme son intitulé le sous-entend, le service est réalisé auprès de personnes dont les dépendances sont extrêmement variées. Ici, plus que la notion de facilité, c'est celle du besoin qui prédomine. En effet, les personnes dépendantes éprouvent des difficultés voire des incapacités à effectuer des actions de la vie quotidienne. Ces personnes voient leur autonomie diminuée et n'ont d'autres choix que de recourir à une personne ou un système tiers pour les soutenir.

*Au sens technologie pour l'aide au handicap* : l'OMS<sup>2</sup> considère l'environnement comme permettant la compensation du handicap. Nous considérons donc un système domotique comme pouvant réaliser des services à la personne dans le but de la rendre plus autonome. Dès lors De Lamotte, dans [72], propose une définition des services prenant en compte la diversité des besoins exprimés par les personnes à mobilité réduite. Le service est présenté comme « l'expression d'une requête sur une ressource domotique, indépendamment de l'interface de communication, et dont l'exécution est automatisée ». Pendant ses travaux de thèse, Allègre [69] reprend cette définition tout en l'enrichissant de nouvelles notions. Un service s'exprime alors comme « une fonction que propose le système domotique à l'utilisateur dans le but de satisfaire ses besoins ». Les services y sont définis d'un point de vue de l'utilisateur et s'expriment selon différents niveaux sémantiques :

- N1 : services élémentaires associés à une opération. Par exemple « Allumer une lampe du salon » qui correspond à l'opération « ON » sur l'équipement domotique « lampe du salon ».
- N2 : scénario composé d'un ensemble de services N1. « Sortir de chez soi » est un scénario qui allume les lampes, ferme les volets et ouvre la porte principale. Après une temporisation ou un autre événement, ferme la porte et éteint toutes les lumières.
- N3 : services de haut niveau sémantique associé à une fonction. « Allumer la lumière » reprend l'idée d'une fonction ayant pour effet la luminosité et qui est exécutée par un équipement sans le définir précisément. Ainsi, pour deux ressources proposant la fonction identique, les informations contextuelles de l'environnement, ou du patient (préférences, localisation) permettent la configuration du service.

### 2.2.1.2 La naissance de la notion de tâche

Une tâche ou un ensemble de tâches est l'expression informatique d'un service. Pour un service simple comme allumer la lumière, sa tâche associée sera l'ensemble des ses paramètres informatiques intrinsèques et sa commande d'exécution. En d'autres termes, si le service s'exprime par « l'exécution d'une opération d'aide au handicap, plus ou moins complexe, sur un équipement domotique particulier ». Nous interprétons la tâche comme « l'exécution d'un programme informatique réalisant une fonction ou plusieurs fonctions d'aide au handicap sur une ressource ou plusieurs ressources quelconques ». Nous pouvons apercevoir le lien qui unit le service et la tâche. Si le service correspond à l'expression des besoins du patient, la tâche quant-à elle représente l'expression des besoins d'un service pour s'exécuter sur l'infrastructure matérielle.

Pour résumer, la notion de tâche est la représentation modifiée d'un service d'aide au handicap auquel est ajoutée sa description informatique, comme nous le représentons

---

2. Organisation Mondiale de la Santé

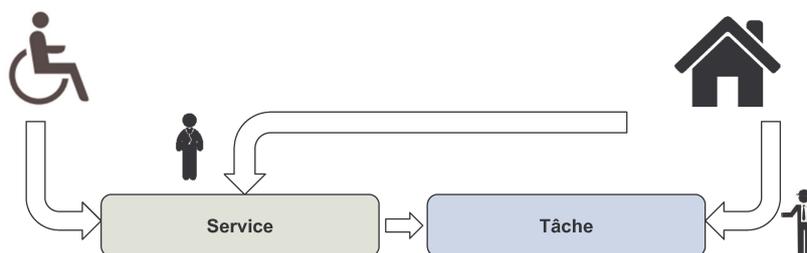


FIGURE 2.3 – Héritage Service-Tâche

sur la figure 2.3. Dans un premier temps, les services sont décrits par le corps médical, et relèvent du niveau fonctionnel et de l'expression des besoins utilisateurs. Des travaux traitent déjà ce sujet c'est le cas par exemple du projet IntelHome, où les services sont définis comme des opérations sur des équipements, et spécifiés par un membre de l'équipe médicale (médecin, ergothérapeute, etc.). Dans la définition d'une tâche, cette description du service est enrichie, par un technicien, d'informations spécifiques : les besoins matériels et logiciels pour l'exécution, etc. Dès lors, l'opération est interprétée comme une tâche informatique représentant un service d'aide au handicap.

Notons aussi qu'un service peut être représenté par un ensemble de tâche indépendantes, comme un service périodique qui serait composé d'une première tâche timer et d'une seconde réalisant les traitements. Prenons aussi pour exemple l'algorithme de management de [30], qui peut être décomposé en plusieurs tâches indépendantes comme montré sur la figure 2.4.



FIGURE 2.4 – Tâches du service de management d'alerte

Nous considérons deux types de tâches : les tâches utilisateurs et les tâches systèmes. Les tâches utilisateurs, comme leur intitulé le suggère, sont des services demandés par l'utilisateur. L'interaction avec ces tâches est directe puisque c'est le patient qui en fait la demande, par l'intermédiaire de son interface client. À l'inverse les interactions avec les tâches systèmes sont transparentes pour l'utilisateur. Elles peuvent être commandées par le système domotique, par exemple pour vérifier l'état des différentes ressources ou encore par un membre de l'équipe médicale à distance pour surveiller les constantes vitales du patient. Nous parlons aussi de tâches systèmes si une tierce personne, comme un médecin, lance un service à distance. Une représentation schématique est proposée sur la figure 2.5.

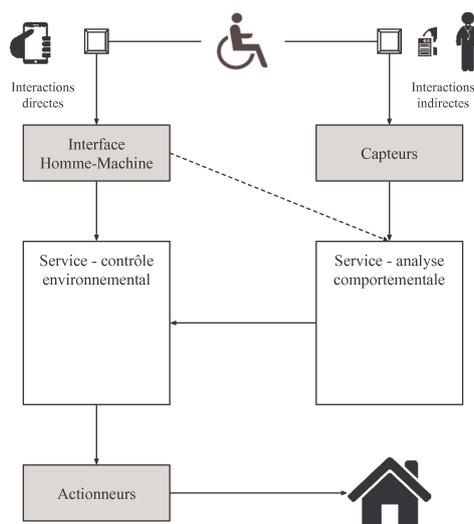


FIGURE 2.5 – Interactions patient-système

### 2.2.1.3 L'offre de service d'aide au handicap

Pour répondre aux besoins des patients présentant un handicap, un ensemble de services leur est proposé. Il est courant de différencier les populations présentant des incapacités physiques et des incapacités cognitives, même si certaines populations comme les personnes âgées sont susceptibles de présenter un plurihandicap. De la même manière, les solutions domotiques ne ciblent souvent qu'une seule de ces deux populations, car la gestion du plurihandicap est un exercice difficile tant la gravité de ce cas ne peut pas se soustraire à assistance humaine. Néanmoins, il est quand même possible de soulager cette surveillance en déléguant une partie de la charge de travail au système domotique. Enfin, les services peuvent être activés par l'utilisateur, on parle d'interactions directes ou de tâches utilisateurs, ou programmés à distance, de manière périodique ou événementielle, on parle dans ce cas d'interactions indirectes ou de tâches systèmes. Ces tâches systèmes sont transparentes pour l'utilisateur qui ne les a pas spécifiquement requises. L'offre de service s'articule autour de ces quatre catégories et est représentée, sur la figure 2.6. Par exemple, un service élémentaire comme ouvrir une porte est considéré comme une tâche informatique utilisateur de contrôle d'environnement. À l'inverse, un algorithme de reconnaissance d'activité ou un message alerte émis à la suite d'un événement sont considérés comme des tâches informatiques systèmes d'analyse comportementale.

**Les services de contrôle d'environnement** Les travaux menés au LabSTICC de Lorient sur l'aide au handicap sont, pour la plupart, teintés du projet initial Cohand qui avait pour but d'améliorer le maintien à domicile des patients présentant une déficience physique. Pour pallier ces déficiences, en se basant sur les technologies domotiques émergentes, des services de contrôle ont été proposés. Deux technologies ont principalement été soutenues par ce projet. Pour la gestion de la lumière et autres appareils sur le réseau électrique, l'ouverture et la fermeture des portes motorisées, ou encore des volets, les commandes utilisent un bus domotique. La seconde technologie utilisée pour le contrôle environnement est l'infrarouge pour le support d'autres appareils comme une télévision, un téléphone, le lit médicalisé, certains éléments de mobilier, et même un ordinateur.

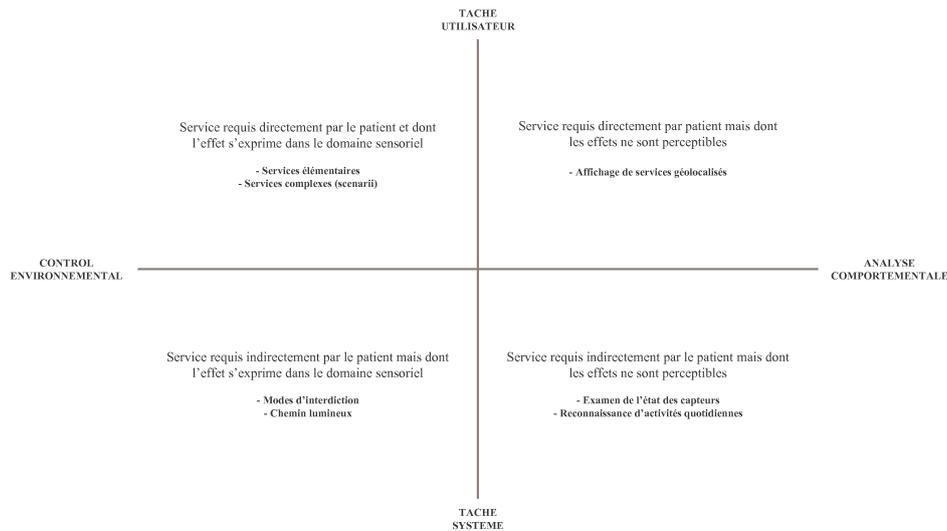


FIGURE 2.6 – Catégories des tâches d'aide au handicap

Une tâche de contrôle d'environnement aura un sens dans l'environnement domotique du patient.

**Les services d'analyse comportementale** Avec les évolutions des besoins des populations dépendantes sont apparus des projets d'assistances et d'automatisations des tâches de la vie quotidienne. Du scénario de contrôle environnemental, basé sur l'association de services élémentaires, aux applications complexes d'assistance, toutes ces tâches ont pour but de limiter le besoin cognitif des patients. Bien que présentes dans de nombreuses solutions d'aide au handicap, nous n'avions, au laboratoire, jamais adressé ces problématiques. Cependant, pour plus de cohérence avec l'ensemble de nos travaux, il paraissait indispensable d'intégrer de tels services. Plus que pour le contrôle des appareils domotiques, les services d'assistance mettent en évidence le besoin de déployer de manière réfléchie les tâches sous peine de voir l'intégrité de son système domotique être remise en cause. Il est facile avec plusieurs tâches d'assistance de surcharger son serveur domotique et de le rendre lent, voire instable. Un cas de figure inconcevable avec une population dont l'autonomie et l'indépendance tiennent au bon fonctionnement de leur solution d'aide au handicap.

#### 2.2.1.4 Modélisation d'une tâche d'aide au handicap

Une tâche s'exprime à deux niveaux : au niveau logiciel et au niveau matériel. Le niveau logiciel traduit les dépendances logicielles que doit résoudre un système pour exécuter une tâche. Dans le cas d'une tâche de consultation de la météo, nous avons défini la tâche comme la consultation de la page internet de météo France. Aussi, la tâche doit pouvoir lancer un navigateur web et posséder un accès à internet pour résoudre toutes ses dépendances logicielles. Le niveau matériel traite les besoins en composants matériels. Une ressource de calcul et de mémoire pour le traitement de données, par exemple, mais aussi un dispositif vidéo pour afficher le résultat de la page web, si on reprend notre exemple de la tâche de consultation de la météo. Toutes ces informations sont renseignées par le concepteur qui doit posséder une connaissance forte de l'offre de tâche.

Sur la figure 2.7, qui représente le modèle d'une tâche, la classe « besoin » exprime

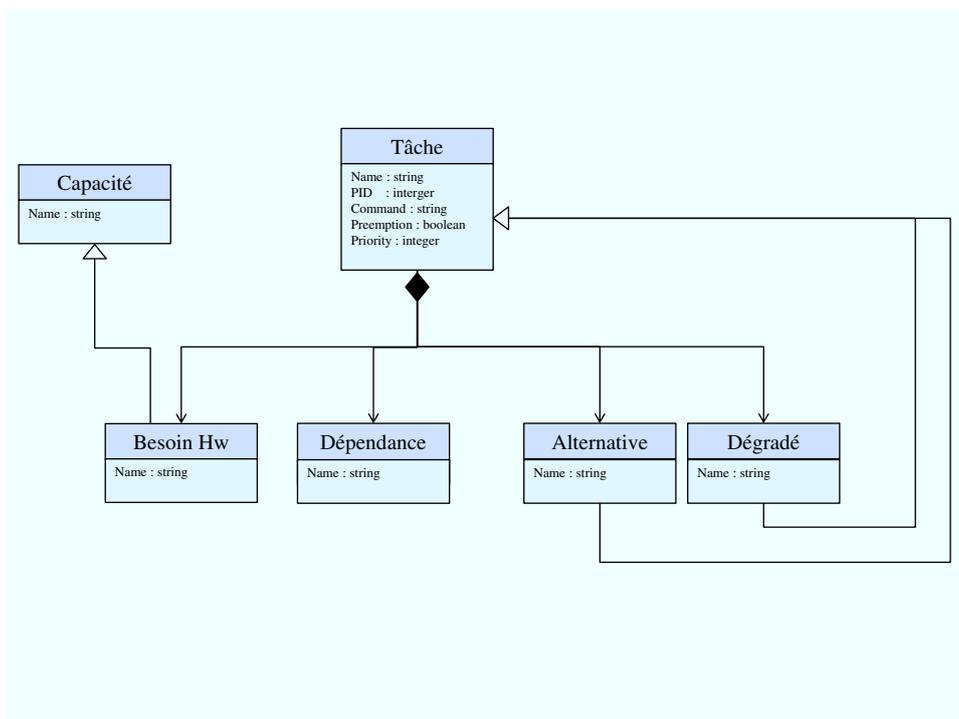


FIGURE 2.7 – modèle de tâche

toutes les contraintes matérielles requises par l'application. Considérons, un système domotique composé d'un serveur et d'un bus domotique sur lequel est branché un ouvreuse de porte électrique. Le service « ouvrir une porte » nécessitera des besoins de traitement et de communication domotique pour s'exécuter. Les capacités de traitements sont assurées par les unités de calcul et la mémoire vive du serveur et la capacité de communication est offerte par l'interface réseau. La classe dépendance reprend les besoins logiciels de la tâche. Reprenons le service « ouvrir une porte », la tâche associée est une application développée en Java et dont les dépendances logicielles sont la machine virtuelle et le fichier source. Parmi les autres classes caractéristiques de notre tâche, nous retrouvons les concepts alternatif et dégradé qui permettent en cas de non-possibilité d'exécution de la tâche désirée de se replier sur une solution alternative ou alors une version dégradée du service. Enfin, il faut noter que chaque tâche possède des propriétés propres comme son nom qui est un identifiant unique, son PID défini par le système d'exploitation d'une ressource informatique si la tâche est exécutée, sa commande d'activation pour exécuter le programme associé, un niveau de propriété et une information sur la préemption de la tâche. Notons que nous définissons généralement 3 niveaux de priorité : le niveau 0 correspond au niveau le plus bas et est utilisé pour les tâches de faibles importances comme certaines tâches systèmes, le niveau 1 est lui associé aux tâches utilisateurs, et le niveau 3 est la priorité la plus importante et est attribué principalement aux alertes.

### 2.2.2 De l'équipement domotique à la ressource

Les équipements domotiques sont l'ensemble des éléments physiques de l'environnement qui proposent une interaction avec le patient. Dans cette section, nous allons prendre le temps de décrire les notions associées aux ressources informatiques.

### 2.2.2.1 Définition d'une ressource

L'équipement domotique dans la littérature est associé à la notion d'un appareil particulier. Par exemple pour réaliser le service « allumer la lumière », nous avons vu que la définition retenue dans les précédents travaux tenus au Lab-STICC était « l'exécution d'une opération ou fonction sur un équipement ». En d'autres termes, pour pouvoir fournir le service de la lumière il faut une lampe dans l'environnement. Cette définition ne prend pas en considération la chaîne complète d'exécution du service et surtout ne met pas en évidence la notion de capacité des équipements. Nous proposons donc la définition d'une ressource matérielle comme étant la description d'un équipement et des ses capacités intrinsèques. Dans l'exemple schématisé, sur la figure 2.8, la lampe est une ressource avec une capacité « d'actionneur lumineux », le bus domotique offre « la capacité de commande domotique », tandis que l'ordinateur possède des « capacités de traitement ». De plus, grâce à la description des liens entre les différents éléments nous proposons la composition de ressource pour créer une ressource plus complexe. En effet, l'ensemble de la chaîne d'exécution forme une « super » ressource combinant toutes les capacités de chaque élément. Nous reviendrons plus tard sur ces notions de capacités, de composition et d'héritage, dans la suite de cette section.

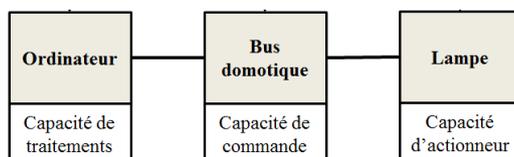


FIGURE 2.8 – Composition d'une ressource domotique

Le terme de ressource est défini pour mettre en avant le fait que les équipements domotiques possèdent des capacités spéciales. Il convient désormais de comprendre ces éléments qui constituent l'environnement domotique.

### 2.2.2.2 L'appareillage domotique

C'est l'environnement et le patient qui déterminent l'appareillage domotique. En effet, l'appareillage domotique est un terme générique puisqu'il diffère d'un patient à un autre, fonction de ses besoins et de sa pathologie. Pour des patients atteints d'un handicap physique, l'appareillage domotique sera principalement composé d'actionneurs facilitant les interactions entre le patient et son environnement. Lorsque le handicap est mental ou intellectuel, l'offre d'équipement sera surtout composée de capteurs et d'unités de calcul pour analyser son comportement et proposer des informations adéquates. Nous souhaitons fournir une solution domotique proposant le contrôle de son environnement et l'analyse du comportement du patient, dans le même environnement domotique. Nous distinguons les équipements en trois catégories de ressources : les ressources informatiques qui représentent les organes de décision et de commande, les ressources passerelles qui transmettent des informations et les ressources périphériques dépendantes d'autres ressources.

**Les ressources informatiques** Nous parlons de ressource informatique pour représenter une ressource de commande ou de traitement. Ces équipements sont bien souvent le point névralgique de toutes les installations d'aide au handicap. Parmi ces éléments,

on retrouve aussi bien un serveur de calcul dédié au traitement d'applications d'analyse comportementale qu'un serveur domotique consacré à l'exécution de commande pour activer des actionneurs. Ces ressources complexes présentent bien souvent des capacités de traitement, mais aussi de commande. Nous pouvons retrouver dans cette catégorie des ordinateurs spécialisés, des ordinateurs personnels fixes ou portables ou encore des cartes embarquées basse consommation ou des appareils mobiles comme un smartphone ou une tablette. Notons que nous pouvons connecter des ressources périphériques aux ressources informatiques dans le but d'apporter à ces dernières de nouvelles capacités, mais nous y reviendrons par la suite.

**Les ressources passerelles** Les ressources passerelles ne sont pas des ressources informatiques. Dans la description, elles se rapprochent plus des ressources périphériques puisqu'elles possèdent parfois des capacités qu'elles cherchent à transmettre. Prenons le cas d'une box adsl classique. Cette ressource passerelle présentera une capacité de « commande Internet ». Autrement dit, elle constitue un pont entre le réseau Ethernet et le réseau internet et offrira la capacité d'accéder à internet à toutes les ressources auxquelles elle sera connectée.

**Les ressources périphériques** On parle de périphérique pour définir un organe extérieur connecté à notre ordinateur : un clavier, un écran, une imprimante, etc. De la même manière, une ressource périphérique est un élément connecté directement, ou en d'autres termes en point à point, à une autre ressource. Comme nous l'avons cité, un écran connecté à une ressource informatique pourra lui transmettre sa capacité « d'actionneur vidéo ». Aussi, un capteur de mouvement branché sur le bus domotique, c'est-à-dire connecté sur une passerelle de bus domotique, offrira sa capacité de détection de présence. Les ressources périphériques se décomposent encore en trois catégories que sont :

- Les périphériques actionneurs : agissent sur l'environnement. C'est le cas d'un écran ou d'un haut-parleur, d'une lampe, etc.
- Les périphériques capteurs : renseignent sur l'environnement. On retrouve dans cette catégorie des détecteurs de présence, de température, des interrupteurs ou encore des interfaces homme-machine.
- Les périphériques de commandes : offre le contrôle d'autres équipements par leur intermédiaire. Assez proches du comportement des ressources passerelles, ils se différencient par le type de connexion. On parle d'émetteur récepteur infrarouge, de commande Enocean ou encore Zwave.

### 2.2.2.3 Associations de ressources

Comme expliqué précédemment, il est possible de créer des associations de composants. Pour illustrer ce concept, nous partons d'un exemple simple : un ordinateur de bureau. Et par ordinateur de bureau, nous n'entendons que le bloc principal sans périphériques. Cependant, un ordinateur de bureau peut être connecté à des ressources périphériques : un écran tactile, une souris, des haut-parleurs, etc. Dans ce cas-là, nous parlerons de ressources associées grâce à un lien de type périphérique. À l'inverse, nous avons vu la ressource box adsl qui est une passerelle autorisant aussi la connexion entre ressources. Dans le cas des ressources passerelles, nous parlerons d'association grâce à un lien de type réseau. Il est important d'intégrer une notion : la distance d'une ressource et sa notion

opposée la proximité de ressource.

Pour faire simple, des ressources ne peuvent s'associer et s'échanger des capacités que si ces dernières sont des ressources de proximité, c'est-à-dire directement connectées. C'est le cas des ressources périphériques, par exemple. Dans certains cas, les ressources peuvent être plus éloignées. Cela serait le cas si nous rajoutions une passerelle Ethernet (un switch Ethernet) entre la box adsl et l'ordinateur, dans l'exemple de tout à l'heure. Néanmoins, nous avons défini que les ressources liées par des liens uniquement réseaux étaient considérées comme des ressources de proximité. Ainsi, nous pourrions placer une douzaine de passerelles Ethernet entre la box adsl et notre ordinateur, la box transmettrait toujours sa capacité. C'est aussi le cas d'un capteur de mouvement (ressource périphérique) sur le bus de réseau séparé du serveur domotique (ressource informatique) par un convertisseur de bus réseau/Ethernet (ressource passerelle). Avec notre règle de proximité de ressource, la capacité du capteur de mouvement sera quand même transmise à l'ordinateur. Des exemples illustreront ces principes dans la suite du document.

#### 2.2.2.4 Modélisation d'une ressource

Le modèle proposé est décrit sur la figure 2.9. Une ressource, quel que soit son type (informatique, périphérique ou passerelle), est définie par :

- Son nom : l'identifiant unique de la ressource
- Son nœud d'activation : son adresse IP (Internet Protocol), numéro d'identification réseau
- Son utilisateur : le profil inscrit dans le système d'exploitation
- Son statut : renseigne son état (en fonctionnement, inaccessible, en panne, etc.)

Une ressource peut posséder des capacités et des applications. Ces deux éléments sont d'une importance capitale puisqu'ils renseignent sur la possibilité d'une ressource à exécuter ou non une tâche informatique. Notons que, chaque capacité associée à une ressource informatique génère une valeur image de son utilisation. Par exemple, un ordinateur possède un processeur. Une valeur associée à sa capacité de calcul est automatiquement déclarée, permettant de refléter avec précision l'état actuel de cette ressource. Il existe deux types de valeurs issues de deux types de capacités différentes. Dans le cas d'une capacité partageable, à usage variable (calcul, mémoire, etc.) la valeur sera une variable de type entière image du taux d'utilisation de l'unité. Si la capacité suggère un usage unique, non partageable (lampe, écran, haut-parleurs, etc.) la valeur sera dans ce cas un booléen renseignant sa disponibilité. Enfin, nous avons vu que les ressources peuvent être connectées les unes aux autres. La classe connexion précise les liens qui peuvent être de type connexion proche « isConnectedTo » entre un périphérique et une ressource informatique ou de type réseau « hasConnectionTo » comme le proposent les passerelles.

Nous pouvons représenter une ressource simplement comme sur la figure 2.10. Cette description est incomplète, mais elle nous permet de commencer à aborder la notion de capacité. Prenons le cas de l'ordinateur. Il possède des capacités matérielles et des applications logicielles. Ces deux concepts confrontés aux besoins de nos services domotiques informatiques nous permettent, comme nous venons de l'expliquer, de décider quelle ressource peut réaliser l'application.

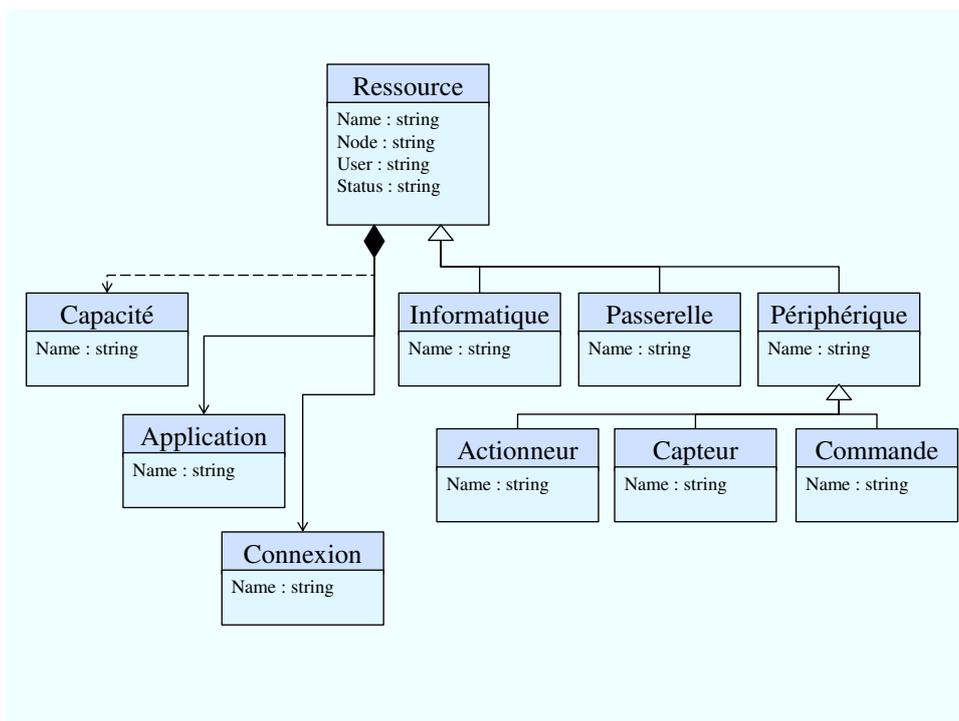


FIGURE 2.9 – modèle de ressource

L'exemple de la smart-caméra est bien plus intéressant puisqu'il allie des capacités hétérogènes. C'est un capteur avec des capacités de calcul. En soit une ressource parfaite pour répartir la charge de calcul d'un système domotique tout en proposant des algorithmes de traitement d'image. Cet équipement peut même devenir une ressource informatique si un système d'exploitation est installé. En d'autres termes, un équipement parfait pour un tel projet. Cependant, une smart-caméra est avant tout une caméra et pour respecter le critère d'intrusivité, il est impossible d'installer un tel équipement, dans un environnement domotique.

### 2.2.3 La capacité, lien entre la tâche et la ressource

Un environnement est un ensemble d'éléments entourant un individu. Quand on parle d'environnement dans le domaine de l'habitat, nous souhaitons décrire l'espace qui entoure l'utilisateur. Il est formé de tous les éléments physiques présents. Prenons l'exemple d'une cuisine, dont la notion d'environnement est déterminée par les dimensions de la pièce et ses éléments tels que les placards, les plans de travail, les chaises, etc. Toutefois, cette définition d'un environnement est très insuffisante pour représenter les aspects domotiques dans le domaine de l'aide au handicap. Nous considérons l'environnement domotique comme étant l'espace utilisé et utilisable par le patient. L'hypothèse de nos travaux est que le patient présente un handicap physique, sensoriel, intellectuel, etc. À partir de ce constat, nous limitons les éléments physiques à ceux proposant des interactions avec le patient. Ainsi, une lampe qui ne bénéficie pas d'une interface domotique ne peut être considérée dans notre définition de l'environnement. C'est la grande différence entre une planche de bois et une porte avec un moteur pilotable électroniquement. Nous distinguons, comme nous l'avons étudié dans ce chapitre, trois types d'équipements domotiques : les

Ordinateur de bureau	Smart-Caméra
<b>Capacités matérielles</b> <ul style="list-style-type: none"> <li>- Traitement Calcul</li> <li>- Traitement Mémoire</li> <li>- Traitement Stockage</li> <li>- Commande Ethernet</li> <li>- Actionneur Vidéo</li> <li>- Actionneur Audio</li> <li>- Capteur Clavier</li> </ul>	<b>Capacités matérielles</b> <ul style="list-style-type: none"> <li>- Traitement Cpu</li> <li>- Traitement Mem</li> <li>- Commande Ethernet</li> <li>- Capteur Vidéo</li> </ul>
<b>Offre logicielle</b> <ul style="list-style-type: none"> <li>- Java JRE</li> <li>- Navigateur Web</li> <li>- Démon EIBd</li> <li>- Lecteur Vidéo</li> <li>- Etc...</li> </ul>	<b>Offre logicielle</b> <ul style="list-style-type: none"> <li>- Aucune</li> </ul>
<b>Connexions</b> <ul style="list-style-type: none"> <li>- Passerelle réseau</li> <li>- Périphérique écran</li> <li>- Périphérique haut-parleurs</li> <li>- Périphérique clavier</li> <li>- Périphérique souris</li> </ul>	<b>Connexions</b> <ul style="list-style-type: none"> <li>- Passerelle réseau</li> </ul>

FIGURE 2.10 – Exemples de description de ressource

ressources informatiques, les ressources passerelles et les ressources périphériques. Les ressources informatiques sont composées d'organes de calcul et de mémoire, notamment. Nous considérons ces ressources comme le centre névralgique du système puisqu'elles peuvent exécuter des services. Les autres ressources ne possèdent pas d'organe de traitement, mais disposent de capacités qui leurs sont propres. Dans la conception de notre ontologie, nous souhaitons mettre en évidence les liens entre les éléments de l'espace domotique. Le but premier est d'exprimer les interactions entre les ressources et éventuellement de partager leurs capacités.

La notion de capacité est cruciale dans la construction de l'ontologie d'environnement. Les capacités sont le pivot entre ressource et tâche et nous permettent de lier les deux concepts entre eux. La capacité est une traduction conceptuelle d'une unité matérielle. Un processeur est une unité de calcul et la ressource qui le possède bénéficie d'une capacité de traitement de calcul. Aussi, une carte Ethernet représente une unité de communication réseau et donnera à la ressource la capacité de communiquer.

Notre objectif est le déploiement d'une tâche sur une ressource. Nous devons résoudre la problématique de quelle ressource peut supporter une tâche donnée. Une tâche est la représentation, sous la forme d'un programme, d'un service. Un programme informatique est une séquence d'instructions formant des opérations interprétables par une machine, à l'aide de ses unités matérielles. Outre le fait que pour réaliser une tâche, la ressource doit satisfaire toutes les dépendances logicielles, nous nous intéressons dans ce cas précis aux unités matérielles qui vont limiter les possibilités des ressources et nous permettre de les discriminer. D'un point de vue informatique, il est aisé de dire qu'une ressource ne propose pas les dépendances logicielles requises et par conséquent, qu'elle ne peut exécuter la tâche. D'un point de vue matériel, nous devons assurer toutes les capacités demandées. Par exemple, un serveur domotique peut résoudre toutes les dépendances logicielles requises par une tâche de diffusion audio, mais ne pas proposer les unités matérielles nécessaires, en l'occurrence l'actionneur sonore : un haut-parleur. Pire, cela ne va pas gêner l'exécution de la tâche, mais aucun effet ne sera perçu par l'utilisateur.

Nous regroupons les capacités en 6 catégories :

- Capacité de calcul : Cette capacité représente l'unité de calcul de la ressource. Nous évaluons la valeur du taux d'utilisation, correspondant à la charge de l'ensemble des processeurs.
- Capacité de mémoire : À l'instar de la capacité de calcul, la capacité de mémoire est la seconde caractéristique de traitement de données. Elle est évaluée en mesurant le taux de mémoire vive restant.
- Capacité de stockage : Elle permet de déterminer quel espace disque est disponible sur une ressource. Dès lors si la tâche requiert le stockage de donnée cette capacité permet de le quantifier.
- Capacité de commande : Les éléments qui offrent des mécanismes de communication comme une passerelle, un émetteur infrarouge ou encore un module Bluetooth, par exemple, offrent des capacités de commande. En effet, certaines ressources sont activables uniquement grâce à ces propriétés.
- Capacité actionneur : Un actionneur est un organe qui produit une opération dans l'environnement. Sous cette catégorie, nous pouvons retrouver des moteurs commandés des dispositifs vidéo ou encore audio.
- Capacité capteur : Cette unité nous renseigne en prenant des informations dans l'environnement. Nous considérons les capteurs au sens large du terme. En effet, dans cette catégorie nous regroupons sous un même terme : un capteur de mouvement, un thermomètre ou encore un écran tactile, qui de par son interface « capte » les interactions de l'utilisateur.

Parmi les capacités que nous avons citées, et qui sont représentées figure 2.11, nous en différencions deux types : les capacités à charge variable et les capacités à usage unique. En effet, l'état des actionneurs et d'une partie des capteurs peut s'exprimer sous forme d'un booléen. Autrement dit, la capacité est utilisée ou disponible. Prenons l'exemple d'un ordinateur portable, qui possède entre autres la capacité d'actionneur visuel, sur laquelle une tâche vidéo est en cours. La capacité visuelle de cette ressource ne peut pas être utilisée pour une autre tâche. Sauf si la nouvelle tâche en question est une alerte et possède un niveau de priorité supérieur, mais nous étudierons cette notion plus tard. Les capacités à charge variable représentent des unités matérielles dont le taux d'utilisation varie en fonction de leur charge. Ces évolutions au cours du temps vont nous offrir la possibilité avec une heuristique de choisir la meilleure ressource. Si on considère maintenant deux serveurs domotiques identiques, à l'instant « t », ils ne possèdent certainement pas la même charge de traitement et l'impact de l'exécution d'une nouvelle tâche ne sera pas le même.

Ces modèles des éléments composant notre système domotique sont formalisés dans une base de connaissance, une ontologie. Les ontologies sont des outils pratiques et performants pour décrire un ensemble de concepts hétérogènes et en déduire des interactions entre ces concepts. C'est ce que nous allons étudier dans la section suivante.

## 2.3 Ontologie de l'environnement domotique

À partir de la modélisation, semi-formelle, de nos tâches et de nos ressources, nous proposons une modélisation sémantique de l'environnement, sous la forme d'une ontolo-

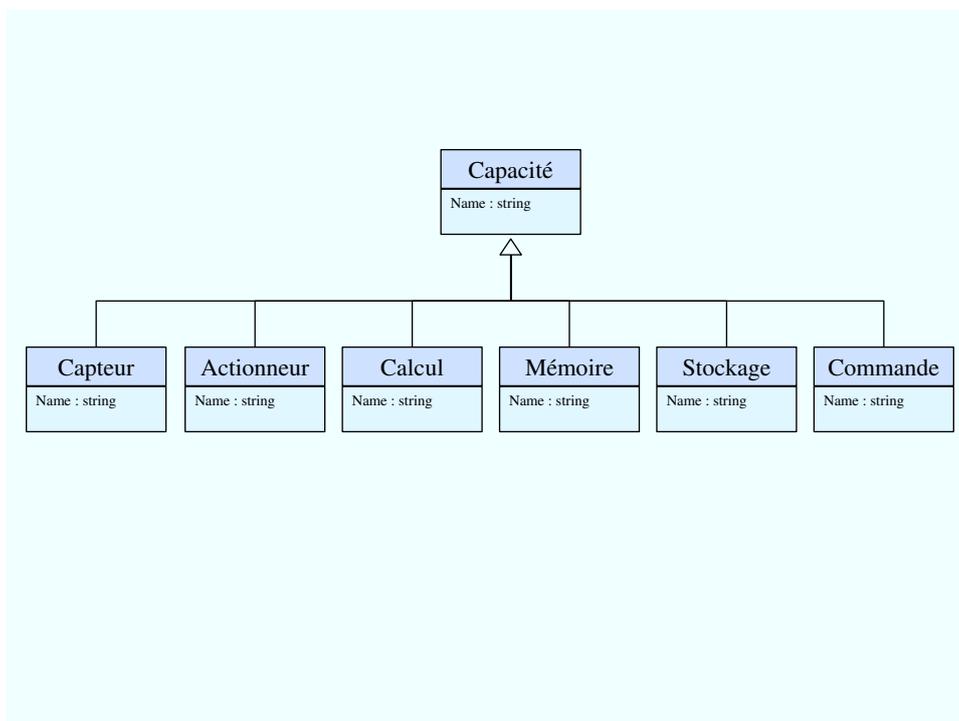


FIGURE 2.11 – modèle de capacité

gie écrite en langage RDF/XML, qui sera l’image informatique de notre environnement domotique. Les bases de connaissances et particulièrement les ontologies sont adaptées aux environnements pervasifs puisqu’elles permettent la représentation de concepts domotiques, concepts prodigieusement complexes et riches en informations, en se détachant de leurs spécificités technologiques, par exemple. De plus, comme nous l’avons vu auparavant, les bases de connaissances peuvent, comme owl, pratiquer l’hypothèse d’un monde ouvert où, à l’inverse de certains domaines d’application, l’information si elle est n’est pas connue n’est pas forcément fausse, mais partielle ou incertaine. De cette notion, des mécanismes exploitent les raisonnements logiques, pour inférer de nouveaux liens sémantiques. Les bases de connaissances apportent le dynamisme, l’interopérabilité, des notions nécessaires pour la représentation d’espaces intelligents et de systèmes pervasifs. Enfin, les ontologies offrent des descriptions qui vont aider à favoriser le processus de consommation d’informations de contexte, mais aussi le processus de fournisseur d’informations de contexte comme on peut le voir avec les travaux sur la découverte de services contextuels [92, 93] et les systèmes de collecte de données [94, 95], notamment.

### 2.3.1 État de l’art et positionnement

Les modélisations sémantiques et ontologiques sont souvent utilisées dans des domaines très spécifiques de l’informatique : le web sémantique, le génie logiciel, l’informatique biomédicale, l’intelligence artificielle ou encore la science de l’information. On les retrouve, toutefois, de plus en plus dans le domaine des espaces intelligents et de l’informatique ubiquitaire, pour la modélisation de contexte [96, 97], ou les interactions homme-machine [98]. Un défaut majeur, que possèdent ces modèles, est leur aspect statique de la représentation qui n’offre pas la modification ou l’évolution des éléments, notamment. DogOnt [99]

propose une ontologie spécialisée pour les systèmes domotiques, cette description s'intéresse principalement aux problèmes d'interopérabilité. Ils garantissent les communications entre les équipements hétérogènes de l'environnement, mais ne prennent pas en compte les capacités spécifiques de ces appareils. Le projet Gaia [82], utilise une ontologie pour décrire les informations contextuelles dans les espaces intelligents. L'ontologie est composée d'un profil d'application et d'un profil d'équipement et permet la configuration des applications en utilisant uniquement des règles sémantiques de réflexion. De plus, l'ontologie utilisée dans Gaia se focalise principalement sur les systèmes multimédias dans les espaces intelligents comme une salle de conférence avec plusieurs dispositifs de diffusion et de contrôle du contenu. Même si les résultats de Gaia sont très intéressants, la projection de leur modèle dans un contexte plus général d'informatique diffuse ne serait pas si triviale. Dans le projet CoBrA (Context Broker Architecture), des agents logiciels récupèrent les informations de contexte pour proposer après traitement des services adaptés à la situation. CoBrA s'appuie sur l'ontologie SOUPA [100] pour décrire les systèmes pervasifs. Bien que reprenant beaucoup de concepts importants dans la description des espaces intelligents, comme les caractéristiques des utilisateurs ou encore les aspects temporels, la description des équipements domotiques, proposée par SOUPA, n'est pas suffisamment développée et n'intègre pas les propriétés dynamiques comme les capacités des ressources, etc. À l'inverse, FLORE [101] intègre les aspects capacités des ressources et des besoins des applications. Cependant, FLORE prend aussi en compte les profils des capacités et préférences du patient, ce qui influe sur la complexité du modèle. Nous avons souhaité mettre en avant uniquement les aspects matériels et infrastructure dans nos travaux. Outre la complexité du modèle d'ontologie proposé par FLORE, les descriptions matérielles ne sont pas suffisamment développées pour nous permettre de réaliser notre but. Toutefois, leurs approches macro et micro sont des idées que nous reprenons dans nos travaux. En effet, nous aurons une première vision de notre système à l'échelle multi-ressources de l'environnement (vue globale) et une vision à l'échelle d'une ressource et de son ordonnancement (vue locale).

Notre approche ne s'intéresse qu'aux interactions des services avec les équipements domotiques, ou autrement dit aux tâches et à leur placement sur des ressources informatiques. Les services au sens où nous l'entendons peuvent être requis par le patient pour combler un besoin, par un personnel médical en local ou distant ou encore par un autre utilisateur lambda. Il peut aussi être demandé par le système lui-même suite à l'analyse d'un comportement habituel ou anormal ou encore pour une intervention périodique. Pour toutes ces raisons, nous avons souhaité, à l'inverse d'autres travaux sur les systèmes pervasifs, faire abstraction du profil du patient pour nous concentrer uniquement sur les applications et leurs supports. Pour résumer, la notion de service correspond à l'exécution d'une opération, ou d'une fonction, par un équipement spécifique. La notion de tâche se rapporte à une application informatique qui ne définit pas de ressource particulière préalable, mais peut être déployée sur n'importe quel appareil comblant les besoins de cette dernière.

### 2.3.2 Introduction aux ontologies

Les ontologies [102, 103] sont utilisées pour représenter des connaissances. On peut citer parmi les plus connues « Gene Ontology » [104] dont le but est destiné à structurer la description des gènes et des produits géniques. Une ontologie permet de décrire un ensemble structuré de termes et de concepts, souvent appelés classes, appartenant à un même domaine. Elle permet de lier les concepts en exprimant leurs relations et en précisant leurs propriétés. Les classes, d'une ontologie, sont présentées suivant une description

hiérarchique. Chaque terme de l'ontologie énonce une définition qui sera héritée par les concepts subordonnés. Une ontologie composée de concepts et d'individus, instances de ces concepts, forme une base de connaissance. Les termes sont liés entre eux par des règles qui complètent l'ontologie.

Précisons maintenant l'idée de monde clos et de monde ouvert pour expliquer les différences entre une base de données et une base de connaissances. Pour cela, prenons pour exemple les types de fraise, suivants : Belrubi, Ciflorette, Gariguettes, Mara des bois ; et cherchons à savoir lesquelles sont cultivées en France. Si l'on se place du côté d'une base de données, à priori, la solution est 4 puisque les quatre variétés sont bien des fraises françaises. Si maintenant nous nous positionnons d'un point de vue base de connaissances, la réponse sera au moins 4 puisque rien ne nous indique qu'il n'en existe pas d'autres. La différence entre un monde ouvert et un monde clos est que dans le premier nous ne pouvons pas tirer de conclusion tant que l'information n'est pas déterminée. Ainsi, si une proposition n'est pas vraie on ne peut pas en déduire qu'elle est fautive. Définir qu'une information est indisponible sans pour autant rendre impossible la formalisation de cette dernière par la suite, peut être intéressant dans des descriptions évolutives.

L'idée des bases de connaissances est donc d'extraire de nouvelles informations implicites en utilisant des mécanismes de réflexion. Ces nouvelles informations sont mises en évidence par le moteur d'inférence. L'inférence est une action de mise en relation d'une connaissance à une autre en utilisant des raisonnements déductifs, inductifs ou encore abductifs. Pour illustrer cette définition, étudions les exemples présentés Figure 2.12.

OWL est l'un des langages informatiques, de représentation des ontologies, les plus répandus [105], toutefois, il existe d'autres comme le standard RDF ou encore SKOS. L'intérêt des ontologies tient dans le formalisme donné à ces concepts, mais aussi dans l'interprétation de ces concepts par des machines. L'interprétation des ontologies permet d'automatiser les processus de raisonnement de la base de connaissance grâce à ses liens sémantiques et au moteur d'inférence. Sur la Figure 2.12a nous avons défini plusieurs concepts : la classe « personne » et la classe « femme », qui est un sous concept de « personne ». Nous créons ensuite une instance de la classe femme : un individu « Laura ». Laura est une femme comme nous l'avons explicitement précisé. Implicitement, Laura est aussi une personne puisqu'elle hérite de la connaissance hiérarchique. La Figure 2.12b illustre certaines des relations sémantiques définies dans OWL (en dur sont les propriétés explicitement définies et en pointillé les connaissances inférées.

- La propriété symétrique, en orange, révèle que si Laura est de la famille de Lydie, Lydie est aussi de la famille de Laura.
- La propriété inverse, en bleu, montre que si Laura est l'enfant d'Olivier alors Olivier est un parent de Laura.
- La propriété de transitivité, en vert, témoigne du fait que si Laura est un enfant d'Olivier, lui-même enfant de Solange nous permet de déduire l'information que Solange est la grand-mère de Laura.

Cette courte introduction aux ontologies montre l'étendue des possibilités offertes par une telle modélisation.

Citons aussi plusieurs logiciels de créations et d'éditeurs d'ontologie comme Hozo, OntoEdit, ou encore Protégé que nous avons utilisé pour composer notre ontologie. Protégé est parmi les éditeurs d'ontologie les plus populaires. Développé en Java par l'Université de Stanford avec la collaboration de l'Université de Manchester, il est distribué sous licence

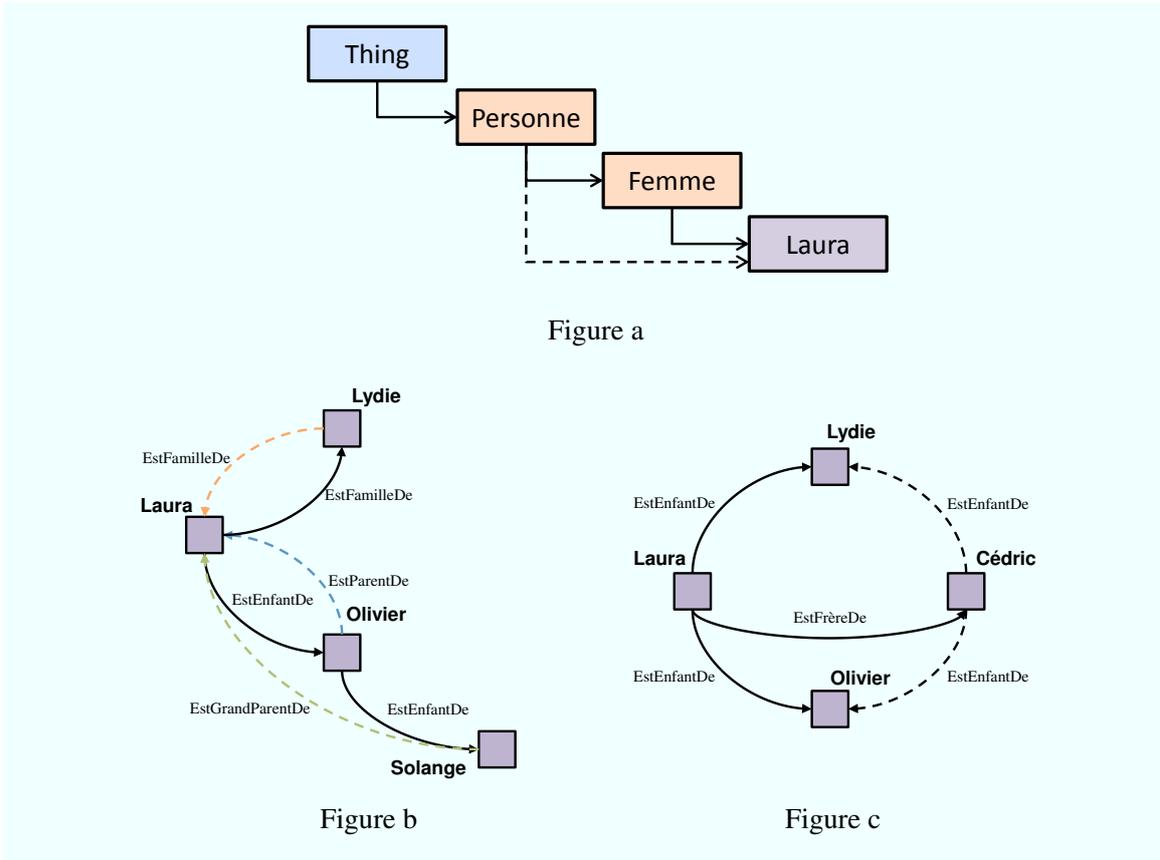


FIGURE 2.12 – Ontologies et propriétés sémantiques

open-source (Mozilla Public Licence 1.1), Protégé interprète plusieurs langages d'ontologie comme OWL, RDF/XML, RDFS, etc. Il bénéficie en outre de plusieurs implémentations de moteur d'inférences dont les plus connus sont Fact++, Hermit ou encore Pellet. Les moteurs d'inférences sont des outils informatiques qui permettent de déduire de nouvelles relations sémantiques. Le travail effectué par le moteur d'inférence sera plus ou moins long fonction de la complexité de la modélisation et du nombre d'instances définies dans l'ontologie. [106] a proposé des variables d'évaluation de l'impact des moteurs d'inférences sur des ontologies dites larges. Associé à l'un de ces moteurs d'inférence Protégé est un outil pratique et performant pour concevoir une ontologie.

Enfin, pour compléter les descriptions ontologiques nous pouvons utiliser le langage sémantique SWRL (Semantic Web Rule Language). SWRL est un langage de règle puissant associant le langage OWL-DL (OWL Description Logics) et le langage RuleML (Rule Markup Language). Ces règles sont interprétables par certains moteurs d'inférences comme Pellet, RacerPro, etc., et permettent d'enrichir la description. Un exemple bien connu est celui de la description des liens familiaux. Prenons 3 individus, une femme, son frère et son fils. Grâce au langage SWRL nous pouvons inférer que si le premier individu possède une relation maternelle avec le second individu et une relation fraternelle avec le troisième individu alors ces deux derniers entretiennent une relation avunculat. Cette règle s'exprime comme suit :  $\text{hasParent} (?x1, ?x2), \text{hasBrother} (?x2, ?x3) \rightarrow \text{hasUncle} (?x1, ?x3)$ . « x » représente les instances d'individus. L'un des problèmes majeurs de la détermination des

ontologies est ainsi accru. En utilisant de nouvelles inférences, qui plus est complexes, cela augmente de manière non négligeable les temps de calcul nécessaires à la résolution et à la vérification de la consistance d'une base de connaissance. Il est recommandé d'utiliser les règles SWRL pour des ontologies de taille petite et moyenne seulement, au vu de la complexité que cela apporte à la modélisation.

### 2.3.3 Notre ontologie d'environnement

Pour illustrer la force de description et de formalisme des ontologies, utilisons des exemples simples d'application de nos modèles. Nous allons dans un premier temps étudier les connexions entre les ressources en dissociant les liens avec les périphériques et les liens avec les passerelles dont les différences sont explicitées dans la section 2.2.2.3. Dans un second temps nous tâcherons d'expliquer comment et pourquoi les capacités peuvent être transmises entre ressources.

#### 2.3.3.1 Construction d'un environnement domotique

Dans cette partie qui ne concerne que les ressources, nous montrons les différentes interactions entre les éléments matériels. La figure 2.13 représente un petit environnement domotique composé de ressources avec chacune leurs capacités intrinsèques. Analysons élément par élément notre espace domotique :

- Les ressources informatiques : Un ordinateur de bureau (Desktop) et un ordinateur portable (laptop)
- Les ressources passerelles : Une passerelle Ethernet (Gateway Ethernet), une passerelle Internet (Gateway Internet) et une passerelle Knx/Eib (Gateway Ip/Knx) qui convertit les requêtes IP en protocole de communication Konnex
- Les ressources périphériques : Un clavier d'ordinateur (Keyboard), un écran d'ordinateur (Screen), un interrupteur Eib (Switch) et une lampe (Light) elle aussi connectée sur le bus Eib

Les connexions entre les passerelles et les ressources informatiques sont centralisées par la passerelle Ethernet et les liens de type réseau sont déclarés : « hasConnectionWith ». Les périphériques sont quant à eux connectés avec des liens, point à point, de type périphérique et déclarés avec : « isConnectedTo ». Le clavier et l'écran sont connectés à l'ordinateur de bureau tandis que l'interrupteur et la lampe sont branchés sur le bus domotique.

La figure 2.14 est une vue, légèrement modifiée, de l'environnement de développement Protégé. La colonne la plus à gauche, sur la figure, représente les concepts hiérarchisés que nous avons définis. Les choses (Things) sont la racine commune de tous les concepts. Ainsi nous avons défini les concepts de « Capacité » (Capacity), de « Tâche » (Task) et de « Ressource » (Resource) au rang 1. La colonne suivante sur la figure montre les individus instanciés dans notre ontologie. Nous retrouvons dans cet espace toutes les ressources définies précédemment : Ordinateurs, passerelles, etc. Enfin la partie suivante qui représente les deux tiers de l'espace, est la partie volontairement modifiée pour faire apparaître les résultats nous intéressants. Ainsi, apparaissent ici les trois ressources suivantes : l'ordinateur de bureau (`_computer_Desktop`), l'ordinateur portable (`_computer_Desktop`) et la passerelle Ethernet (`_gateway_Ethernet`). Pour chacune des ressources citées, nous pouvons observer les liens entre les individus (Object property). Dans cet exemple l'ordinateur de bureau possède une connexion réseau avec la passerelle Ethernet (`hasConnectionWith`)

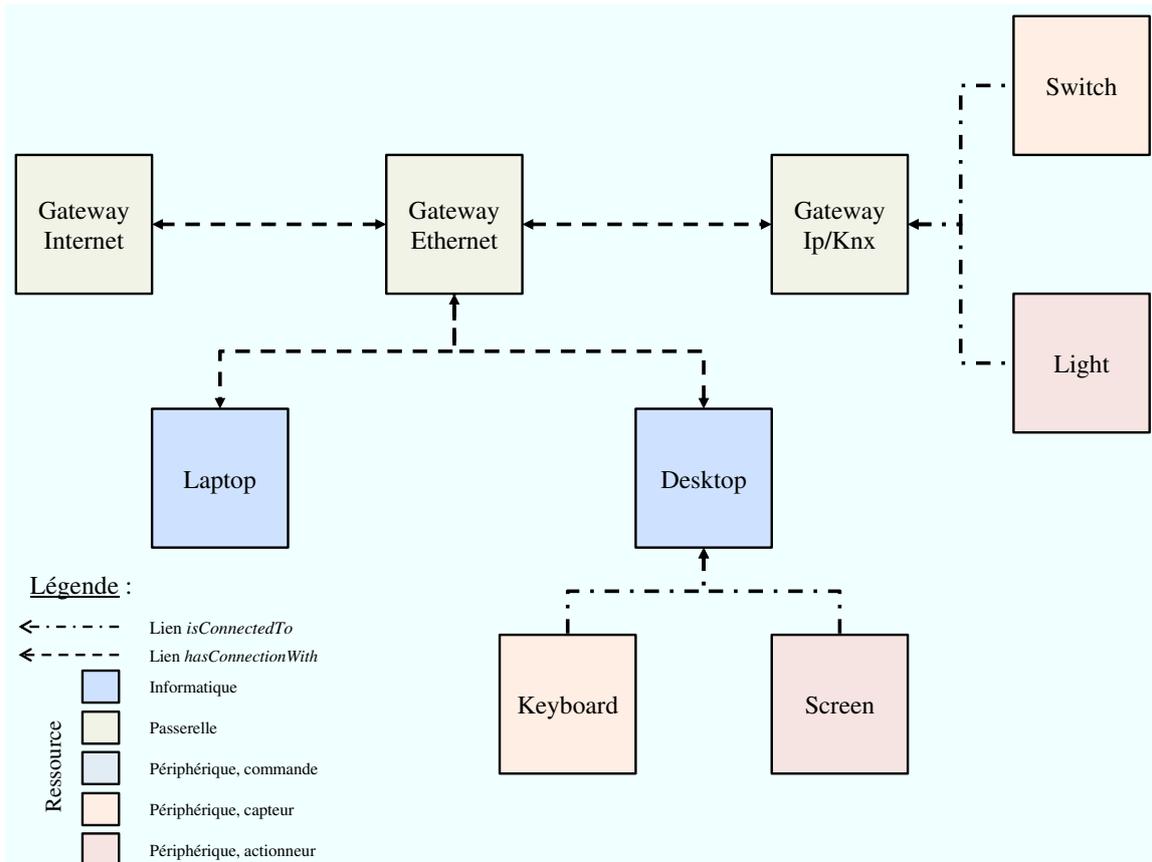


FIGURE 2.13 – Environnement domotique simple schématisé

`_gateway_Ethernet`) et deux connexions périphériques avec l'écran (`isConnectedTo _peripheral_Screen`) et le clavier l'écran (`isConnectedTo _peripheral_Keyboard`). L'ordinateur portable (`_computer_Laptop`) est connecté uniquement avec la passerelle Ethernet. Enfin, la passerelle Ethernet possède deux liens réseau déclarés, l'un avec la passerelle Internet (`_gateway_Internet`) et l'autre avec la Passerelle Eib (`_gateway_Eib`). Les connexions de la passerelle Ethernet avec les deux ordinateurs ne sont pas explicitement déclarées. En effet, `isConnectedTo` et `hasConnectionWith` possèdent une propriété de symétrie qui sera révélée par le moteur d'inférence. Les liens déclarés ci-dessus permettent de décrire la figure 10 sous la forme d'une ontologie.

Un des principes définis dans ce chapitre n'apparaît pas pour autant dans cette description. Nous avons spécifié, dans la partie 2.2.2.3, que les ressources bénéficiant d'un lien réseau étaient toutes considérées comme des ressources avec une proximité directe. Rappelons rapidement la notion de proximité directe : une ressource possède une proximité directe avec une autre ressource si le lien entre les deux n'est pas interrompu par une ressource tierce. C'est par exemple le cas du lien entre une ressource périphérique et une ressource informatique. Cette notion implique une seconde règle qui dit que les ressources possédant un lien réseau n'ont pas de notion de distance entre elles si aucune autre ressource n'interrompt ce lien. Reprenons notre exemple figure 2.13 pour illustrer ce propos. D'un point de vue typologique, l'ordinateur de bureau est à proximité de la passerelle Ethernet, mais possède une relation distante avec la passerelle Internet. Or les connexions

qui relie ces deux ressources sont toutes de type réseau (`hasConnectionWith`). Du point de vue de notre ontologie, un lien direct devrait apparaître entre la passerelle internet et l'ordinateur.

C'est ce que nous souhaitons démontrer sur la figure 2.15. Sur cette vue du logiciel Protégé, nous avons rajouté un espace appelé « Rule ». Ce domaine nous permet d'écrire des règles de logique sémantique, SWRL. Les axiomes R1, R2 et R3 représentent trois individus ressources. Le point d'interrogation indique qu'il n'y a pas de précision sur cet individu. En d'autres termes ?R1 sera aussi bien l'ordinateur portable que l'ordinateur de bureau que l'écran ou qu'une des nombreuses passerelles. Il est possible de nommer directement un individu en citant son nom à la place, mais la règle perd dans ce cas de sa généralité. La règle exprimée en langage courant dit que : si une ressource R1 a une connexion de type « réseau » avec une ressource R2 et que R2 possède aussi une connexion de type « réseau » avec une ressource R3, alors les ressources R1 et R3 bénéficient aussi d'une connexion de type « réseau ». Il faut bien entendu que R1 et R3 soient des ressources différentes pour ne pas générer de fausses relations.

Après avoir interrogé le moteur d'inférence, les résultats présentés en dessous de la règle montrent que de nouvelles relations ont été déduites. En effet, l'ordinateur de bureau (`_computer_Desktop`) a gagné de nouvelles connexions avec la passerelle Eib, la passerelle Internet et l'ordinateur portable. De la même manière, nous pouvons observer d'autres exemples de ressources ayant bénéficié de cette règle. La fenêtre présentée en dessous montre le raisonnement utilisé par le moteur pour inférer les nouvelles relations.

Sur la figure 2.16, nous mettons en place un second exemple où la ressource « Gateway Ip/Knx » n'est plus une ressource passerelle, mais une ressource périphérique de commande, comme nous pouvons l'observer sur le schéma topologique à gauche de la figure. À l'instar de toutes les ressources périphériques, la « Gateway Ip/Knx » possède désormais une connexion point à point déclarée « `isConnectedTo` ». Cela est vérifiable dans l'espace déclaratif du premier individu (`_computer_Desktop`) grâce à l'assertion : `isConnectedTo_peripheral_Eib`. Les résultats présentés sur la figure 2.16 sont obtenus après avoir effectué les inférences à l'aide du moteur. Dès lors, sur la seconde ressource (`_computer_Laptop`), nous pouvons noter qu'elle n'a aucun accès à la « Gateway Ip/Knx ».

Ces règles issues de la topologie de notre environnement domotique vont avoir un impact sur la transmission des capacités. Avoir cette notion de proximité au niveau des ressources a un objectif concret, celui de faciliter l'héritage des capacités, qui au contraire sera bloqué dans les autres cas. C'est ce que nous allons étudier dans la partie suivante.

### 2.3.3.2 Héritage de capacité

Nous définissons l'héritage de capacité comme la transmission de capacité d'une ressource à une autre ressource. L'exemple d'une ressource périphérique est le cas le plus simple pour exprimer cette notion. Cet exemple est illustré par la figure 2.17, où l'on retrouve 5 individus : un ordinateur portable, deux ordinateurs de bureau, un écran et un clavier. Le schéma topologique de cet environnement domotique est donné en bas à droite de la figure 2.17. Aucune des trois ressources informatiques ne présente de lien réseau et ce n'est pas le sujet de la démonstration. Par contre, le second ordinateur de bureau est lui connecté avec les deux ressources périphériques : clavier et écran.

Cet exemple est aussi le temps de faire un point sur les capacités. Les capacités re-

présentent les possibilités matérielles des ressources. Par exemple, la capacité de calcul (`_capacity_Process_Cpu`) correspond à la possibilité d'une ressource d'exécuter des instructions par son unité de calcul, son processeur. Il existe 6 types de capacité qui sont représentés dans la colonne la plus à gauche de l'interface de Protégé, sur la figure 2.17. Nous avons instancié, dans la colonne suivante, 6 individus capacités : actionneur audio (`_capacity_Actuator_Audio`), actionneur vidéo (`_capacity_Actuator_Video`), capteur clavier (`_capacity_Sensor_Keyboard`), traitement calcul (`_capacity_Process_Cpu`), traitement mémoire (`_capacity_Process_Cpu`) et traitement stockage (`_capacity_Process_Cpu`). Il est à noter que chaque individu capacité, lorsqu'il est associé à une ressource, attribue une propriété de donnée (Data Property) à cette ressource. Cette propriété de donnée est une variable entière s'il s'agit d'une capacité à taux d'usage (taux d'utilisation de la mémoire vive pour une capacité de traitement mémoire) ou un booléen s'il s'agit d'une capacité à usage unique (un écran ne peut « potentiellement » pas afficher deux images en même temps puisque l'une risquerait de masquer la seconde). Toutefois, cette propriété n'est pas représentée dans cet exemple.

Regardons de plus près l'espace des déclarations des ressources, sous la droite de la figure en dessous des règles, et comparons les deux ordinateurs de bureau (`_computer_Desktop`, `_computer_Desktop2`). Les capacités définies pour l'un et pour l'autre sont identiques, ce qui n'est pas anormal, les deux ressources étant de la même nature. Nous y retrouvons les capacités principales de traitement : le calcul, la mémoire et le stockage. À l'inverse de l'ordinateur portable (`_computer_Laptop`) ces deux premières ressources ne présentent pas de capacité d'affichage et de possibilité sonore ni de capacité d'interface clavier. Nous considérons l'ordinateur de bureau comme une ressource de base à laquelle nous pouvons, ou non, ajouter d'autres briques pour en faire une ressource plus complète. Dans notre cas nous pouvons observer deux connexions périphériques en plus pour l'ordinateur de bureau numéro 2, comme expliqué en début de cette section. Ces liens autorisent la transmission des capacités des ressources périphériques vers les ressources informatiques lorsqu'ils sont associés à la nouvelle règle SWRL :

*Computer(?R1), hasCapacity(?R2, ?C), isConnectedTo(?R1, ?R2) -> hasCapacity(?R1, ?C)*

Expliqué en termes humainement compréhensibles cette règle explique, que : si une ressource informatique est connectée à une autre ressource par un lien périphérique alors la capacité de cette seconde ressource est transmise à la première. Nous pouvons remarquer que nous limitons l'héritage de capacité aux ressources informatiques puisque les autres ressources n'en auraient pas l'utilité. Si l'on regarde la seconde partie de l'espace des déclarations des ressources, qui concerne l'ontologie après le passage du moteur d'inférence, l'ordinateur de bureau numéro 2 a bien hérité des capacités des ressources périphériques clavier et écran.

Dans le dernier exemple présenté sur la figure 2.18, on cherche à montrer la transmission de capacité lorsque des ressources sont connectées par un lien réseau. Le schéma topologique associé à l'exemple est présenté en bas de la figure. Il représente un environnement domotique avec deux ressources informatiques et deux ressources passerelles. La ressource passerelle « Gateway Ip/Knx » offre la capacité de commande du bus Eib (`_capacity_Command_KnxEib`). À première vue, la passerelle étant distante des deux ordinateurs, ces derniers ne devraient pas bénéficier de cette capacité. Or les liens de connexions sont de type réseau (`hasConnectionWith`) et nous avons défini plus tôt que

les liens de ce type ne créaient pas de distance si leur nature était conservée. La première règle d'inférence nous permet de remettre ces ressources à proximité. Associée à une nouvelle règle SWRL, nous pouvons transmettre des capacités à l'instar des ressources périphériques. La règle s'écrit :

$$\text{Computer}( ?R1), \text{hasConnectionWith}( ?R1, ?R2), \text{hasCapacity}( ?R2, ?C) \rightarrow \\ \text{hasCapacity}( ?R1, ?C)$$

Cette règle se traduit par : Si une ressource informatique à une connexion de type réseau avec une seconde ressource et si cette seconde ressource possède une capacité alors la ressource informatique hérite de cette capacité. Sur la figure 2.18, nous pouvons regarder les résultats avant et après exécution du moteur d'inférence sur l'ontologie. Notons que la capacité de commande du bus Eib est bien transmise aux ressources informatiques malgré la distance physique.

Pour terminer notre ontologie, il faut rajouter aux ressources leurs applications logicielles requises aussi par les services domotiques. Ces applications seront décrites sous la forme d'une propriété de donnée (Data Property) associée à une valeur texte. Par exemple, pour une ressource avec un navigateur web aura une propriété : `hasSoftware( « Browser Web » )`.

### 2.3.3.3 Les tâches pour compléter l'ontologie

Nous n'avons toujours pas défini les tâches domotiques ce qu'il nous reste à faire pour compléter notre ontologie. À la façon des ressources, les tâches sont instanciées comme des individus à part entière. En nous appuyant sur l'environnement domotique décrit sur la figure 2.13, nous pouvons définir deux services : « Allumer la lumière » et « Afficher la météo de Saint-Nazaire par internet ». Analysons les besoins de chacun des deux services, représentés sur la figure 2.19. Les besoins matériels, notion image des capacités des ressources, sont déclarés à l'aide des propriétés d'objet (Object Property) et les besoins logiciels, notion images des applications des ressources, sont déclarés grâce aux propriétés de données (Data Property).

### 2.3.3.4 La recherche de ressource pour exécuter une tâche

Pour illustrer l'importance de la capacité dans notre modélisation, étudions un exemple simple où nous reprenons l'environnement domotique simplifié défini sur la figure 2.8. Ces ressources possèdent chacune leurs capacités. Sur la figure 2.20, nous désirons étudier la possibilité d'exécuter le service « allumer la lumière ». La tâche informatique associée a besoin de la capacité de traitements de la capacité de commande domotique et de la capacité actionneur lumineux pour pouvoir être exécutée. L'association des ressources présente dans notre environnement simple satisfait toutes les recommandations de la tâche. Nous pouvons alors déployer ce service grâce aux ressources disponibles. Ici, nous ne présentons que 3 ressources, mais les possibilités offertes par des environnements plus complexes nous permettent d'imaginer des déploiements plus optimaux. Notons que sur cet exemple, nous ne représentons que les capacités matérielles.

Malgré la volonté d'obtenir le résultat du couple tâche/ressource directement grâce à l'ontologie et aux règles mises en place, nous ne sommes pas parvenus discriminer suffisamment les ressources pour obtenir une solution unique. C'est pourquoi nous avons décidé

de laisser la vérification de la faisabilité et la sélection de la solution au flot de déploiement.

Pour conclure ce chapitre, nous avons étudié le modèle de tous les éléments que nous pouvions retrouver dans un environnement domotique puis nous avons formalisé ces principes dans une base de connaissance. Combiné à des outils de conception assistée, l'environnement pourrait ainsi être facilement assemblé. Les blocs de ressources se liant les uns aux autres, offrant par conséquent leurs capacités pour former de « super » ressources. En outre, nous avons remarqué certaines associations fortes entre les concepts notamment grâce aux capacités véritables pivots de notre description. Nous avons pendant un temps tenté de faire résoudre par l'ontologie la question finale de notre projet « Qu'elle ressource est la plus adaptée pour exécuter telle tâche domotique ? », mais nous nous sommes heurtés à des problèmes de description sémantique complexe. Les ontologies sont un outil aussi performant que compliqué, mais aussi une technologie assez récente qui ne permet pas encore de décrire avec facilité tout ce que l'esprit peut imaginer. La solution pour répondre à notre question principale a été déplacée à un autre niveau : celui de l'exécution continue de notre application. Les choix cruciaux sont donc effectués au moment du déploiement du service domotique, ce que nous allons voir dans le prochain chapitre.

The screenshot displays the Protégé interface with the following components:

- Class Hierarchy:** Shows a tree structure starting with 'Thing', branching into 'Capacity', 'Task', and 'Resource'. 'Resource' further branches into 'Peripheral', 'Actuator', 'Command', and 'Sensor'. 'Peripheral' branches into 'Gateway' and 'Computer'.
- Object Properties:** Lists individuals for 'computer\_desktop':
  - Individuals: `_computer_desktop`, `_gateway_Elb`, `_gateway_Ethernet`, `_peripheral_EBPSwitch`, `_peripheral_Keyboard`, `_peripheral_Light`.
- Property Assertions:** Three panels show assertions for properties:
  - computer\_desktop:**
    - Object property assertions: `isConnectedTo_peripheral_Screen`, `isConnectedTo_peripheral_Keyboard`, `hasConnectionWith_gateway_Ethernet`.
    - Data property assertions: (empty)
    - Negative object property assertions: (empty)
    - Negative data property assertions: (empty)
  - computer\_Laptop:**
    - Object property assertions: `hasConnectionWith_gateway_Ethernet`.
    - Data property assertions: (empty)
    - Negative object property assertions: (empty)
    - Negative data property assertions: (empty)
  - gateway\_Ethernet:**
    - Object property assertions: (empty)
    - Data property assertions: (empty)
    - Negative object property assertions: (empty)
    - Negative data property assertions: (empty)

61  
FIGURE 2.14 – Déclaration des liens entre les ressources avec le logiciel Protégé

The screenshot displays the Protégé software interface, showing the 'Rules' tab and the 'Property assertions' tab for the class '\_computer\_Desktop'.

**Rules:**

- hasConnectionWith(?R1, ?R2), hasConnectionWith(?R2, ?R3), DifferentFrom (?R1, ?R3) -> hasConnectionWith(?R1, ?R3)

**Property assertions for \_computer\_Desktop:**

- Object property assertions:
  - isConnectedTo \_peripheral\_Screen
  - isConnectedTo \_peripheral\_Keyboard
  - hasConnectionWith \_gateway\_Ethernet
  - hasConnectionWith \_gateway\_Elb
  - hasConnectionWith \_computer\_Laptop
  - hasConnectionWith \_gateway\_Internet
- Data property assertions:
  - hasConnectionWith \_gateway\_Ethernet
  - hasConnectionWith \_gateway\_Elb
  - hasConnectionWith \_computer\_Laptop
  - hasConnectionWith \_gateway\_Internet

**Explanation for \_computer\_Desktop hasConnectionWith \_gateway\_Internet:**

Explanation for: \_computer\_Desktop hasConnectionWith \_gateway\_Internet

**Different individuals:** \_computer\_Desktop, \_gateway\_Elb, \_gateway\_Ethernet, \_peripheral\_ElBSwitch, \_peripheral\_Keyboard, \_peripheral\_Light, \_peripheral\_Screen

hasConnectionWith(?R1, ?R2), hasConnectionWith(?R2, ?R3), DifferentFrom (?R1, ?R3) -> hasConnectionWith(?R1, ?R3)

hasConnectionWith \_gateway\_Internet

hasConnectionWith \_gateway\_Ethernet

**Legend:**

- Show regular justifications:  All justifications
- Show iconic justifications:  Limit justifications to 2
- Display iconic explanation:

FIGURE 2.15 – Inférence des liens de proximité avec le logiciel Protégé

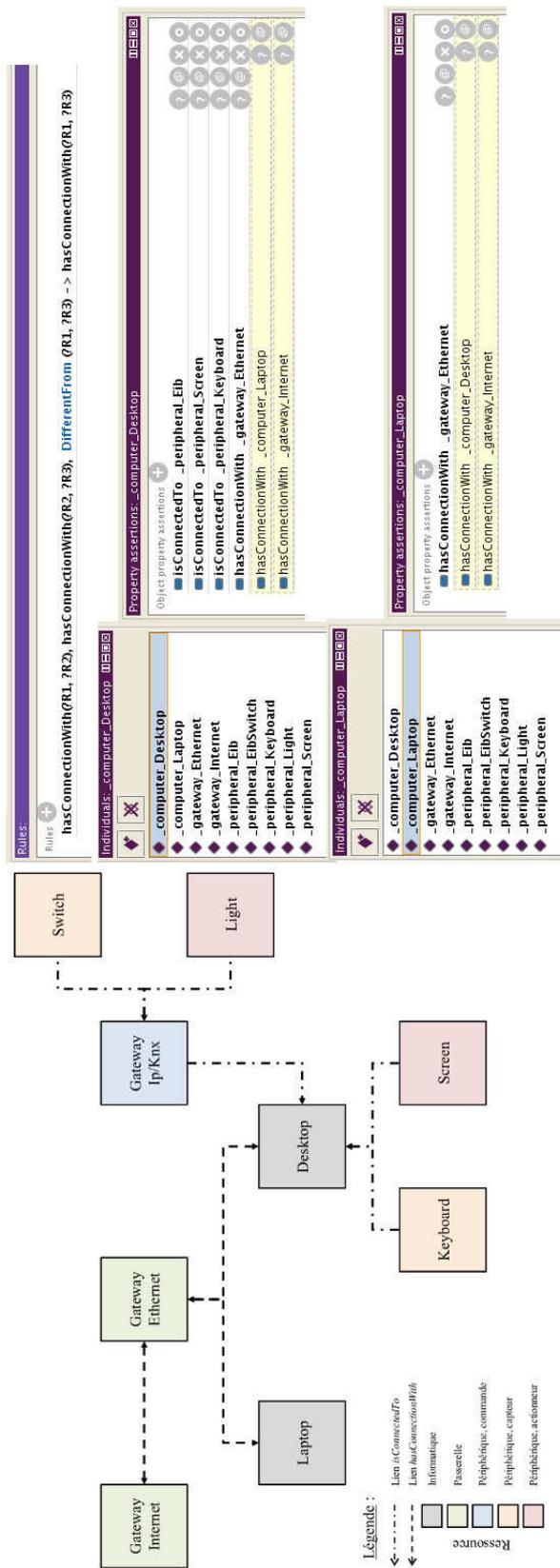


FIGURE 2.16 – Étude de l'impact du lien « connexion périphérique »

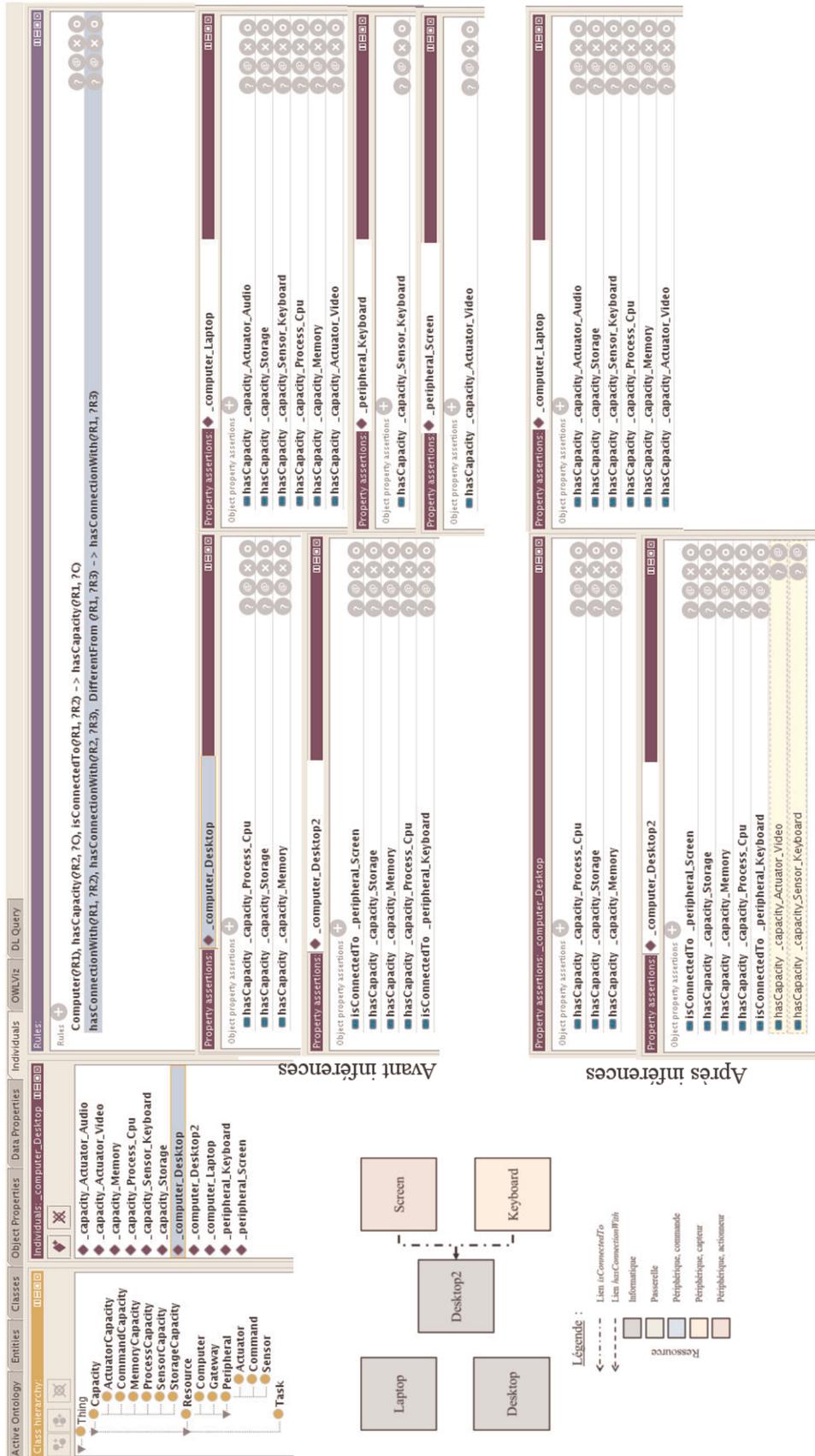


FIGURE 2.17 – Exemple d’héritage par ressource périphérique

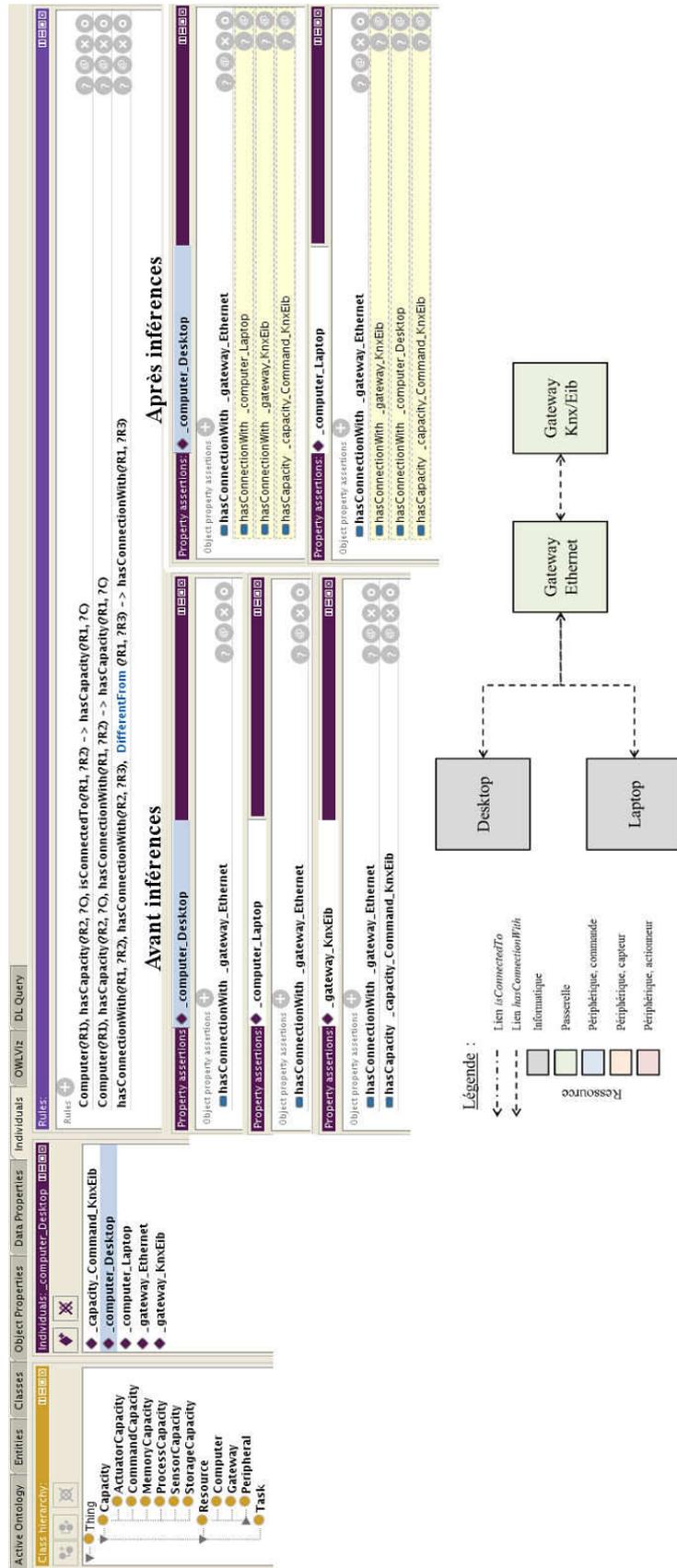


FIGURE 2.18 – Exemple d’héritage par ressource passerelle

Allumer la lumière EIB	Consulter la météo
Besoins matériels - Traitement Cpu - Traitement Mem - Commande Knx/Eib - Actionneur Lumineux	Besoins matériels - Traitement Cpu - Traitement Mem - Commande Internet - Actionneur Vidéo
Besoins logiciels - Démon EIBd	Besoins logiciels - Navigateur web - Java JRE

FIGURE 2.19 – Étude des besoins de tâches domotiques

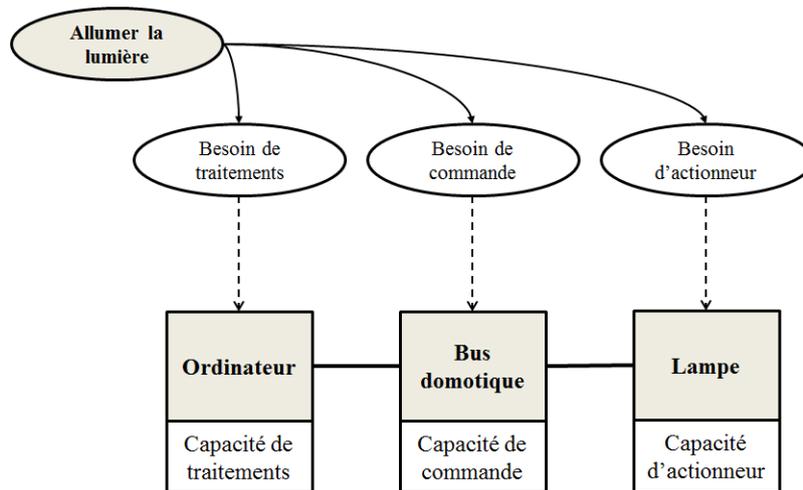


FIGURE 2.20 – Recherche du couple tâche/ressource

## Chapitre 3

# Déploiement de services d'aide au handicap

### Sommaire

---

<b>3.1</b>	<b>L'infrastructure informatique</b>	<b>69</b>
3.1.1	Le système d'exploitation : Linux	74
3.1.2	La machine virtuelle Java	75
3.1.3	Le Framework OSGi	76
3.1.4	Diagrammes d'état et cycle de vie	77
3.1.4.1	Les processus Unix	77
3.1.4.2	Les bundles OSGi	78
<b>3.2</b>	<b>Le flot de déploiement de tâches</b>	<b>79</b>
3.2.1	Le manager d'ontologie	82
3.2.2	Le gestionnaire de tâches	83
3.2.2.1	Phase de décision	84
3.2.2.2	Phase d'ordonnancement	85
3.2.2.3	Phase d'exécution	85
3.2.3	Le moniteur de ressources	86

---



Nous venons d'étudier la modélisation de notre environnement domotique. Les modèles de ressources et de tâches d'aide au handicap nous permettent de décrire les éléments qui composent cet espace domotique tant d'un point de vue physique (matériel) que logique (logiciel). Le modèle de capacité nous donne la possibilité de lier ces deux mondes autour d'un même concept, celui de la possibilité d'exécuter une tâche d'aide au handicap par une ressource, autrement dit de confronter les besoins de tâches aux capacités des ressources. Toutefois, l'ontologie ne nous renseigne pas suffisamment sur le couple tâche/ressource optimal. À cela, deux problèmes, nous n'avons pas réussi à discriminer suffisamment les résultats pour obtenir une liste restreinte de couples tâche/ressource, voire une unique solution, et nous ne pouvons pas explorer l'aspect temps réel d'une solution à cause des contraintes de traitement des ontologies. C'est-à-dire que nous ne souhaitons pas, d'une part, seulement discriminer les ressources sur leurs capacités, mais aussi sur le taux d'utilisation de ces dernières. D'autre part, nous désirons garantir une exécution continue d'un système domotique multitâches d'aide au handicap en temps réel, à la manière d'un système d'exploitation. Les mécanismes pour mettre en place la gestion de tâches concurrentielles rendraient la création d'une ontologie tellement complexe, que même s'il était possible de le faire les temps de traitement de cette dernière la rendraient inutilisable. Tous ces problèmes nous ont poussés à déplacer les aspects « décision du couple tâche/ressource » dans une application déployement de services d'aide au handicap.

Approfondissons maintenant le déployement d'un service d'aide au handicap dans un environnement domotique. Ce chapitre se consacre à l'étude des applications formant notre flot de déployement. Nous reprenons sous le terme flot de déployement, le cheminement que suit un service d'aide au handicap : de la requête du patient, jusqu'à l'exécution de la tâche dans l'environnement domotique.

### 3.1 L'infrastructure informatique

Prenons le temps de faire un point sur les termes : service d'assistance (aussi écrit service domotique) et tâche d'aide au handicap. Nous abusons souvent du terme « tâche d'aide au handicap » qui est la représentation informelle applicative « d'un service d'aide au handicap ». Comme précisé dans le chapitre précédent, nous souhaitons distinguer le terme un peu trop général de service et bien montrer la différence avec le concept informatique qu'il implique. Le service est le concept général, l'idée d'offrir une assistance, et la tâche d'aide au handicap est sa forme informatique, issue du modèle de tâche. La tâche d'aide au handicap contient plusieurs paramètres et propriétés, dont la commande d'activation du programme informatique, ses fichiers sources, etc., et réalise la fonction d'aide au handicap lorsqu'elle est exécutée. Toutefois, il ne faut pas confondre tâche d'aide au handicap, dont le programme complexe associé peut s'exécuter sur une ressource, avec une tâche, un processus ou un thread informatique qui sont des termes plus techniques encore. Par souci de clarté, nous précisons à chaque fois les termes « tâche d'aide au handicap » ou « tâche d'assistance » quand nous parlerons de la forme applicative d'un service, et nous parlerons de processus informatique pour parler de l'exécution du programme associé sur un ordinateur. Pour éviter toute ambiguïté avec les tâches du vocabulaire informatique, la figure 3.1 schématise les différents niveaux de concept que nous abordons tout au long de ce chapitre.

En avant-propos, nous pouvons présenter notre application de déployement comme un système d'exploitation multitâche préemptif où les services d'aide au handicap sont vus comme des applications cohabitant entre elles sur un groupe de ressources informatiques.

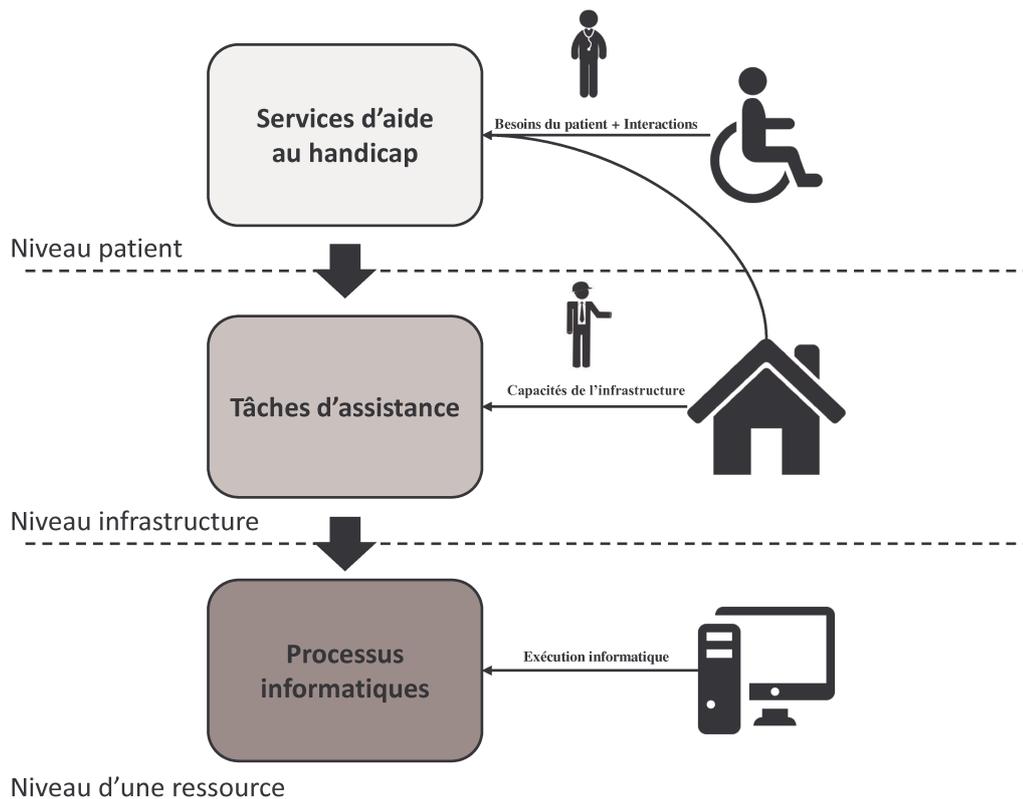


FIGURE 3.1 – Niveaux de concepts d'aide au handicap

L'annonce de ce principe impose d'expliquer certaines définitions utilisées : processus informatique, système d'exploitation, multitâche, préemption et priorité, etc.

Un processus est une application ou un programme en train de s'exécuter sur un système informatique. Il est défini comme un ensemble d'instruction à exécuter, un espace mémoire pour les données de travail et éventuellement d'autres ressources comme des descripteurs de fichiers, etc. Le processus dispose d'une caractéristique temporelle et séquentielle. Le processus informatique possède un cycle de vie donc un début et une fin, nous y revenons plus en détail dans la suite de ce chapitre. Dans un système d'exploitation multitâche, le processus est associé à un PID (Process IDentifier), un numéro unique d'identification. Nous précisons ce terme pour éviter l'amalgame entre la tâche d'aide au handicap que nous définissons comme une représentation applicative du service d'assistance et le processus qui est l'exécution de l'application sur une ressource informatique. En d'autres termes, le processus est interprété au niveau d'un ordinateur et de son système d'exploitation tandis que la tâche est interprétée par notre système de déploiement au niveau multi-ressources.

Un système d'exploitation [107] (operating system, en anglais) peut être résumé comme un gestionnaire de ressource. C'est un ensemble de programmes qui possèdent trois objectifs. Dans un premier temps, il fait l'interface entre les applications et le matériel en faisant abstraction des spécificités complexes de ce dernier. Il met à disposition les ressources matérielles de l'ordinateur et garantit une indépendance vis-à-vis du matériel.

Enfin, il favorise l'utilisation efficace des ressources matérielles. Par exemple, un système d'exploitation garantit un accès sûr à une imprimante en n'autorisant qu'une seule application à envoyer des données au même moment. Il facilite l'utilisation efficace des unités de calcul (CPU) en suspendant des processus de faible importance au profit d'autres applications plus productives. Il propose également des techniques d'abstractions pratiques qui dissimulent à l'utilisateur les détails complexes du matériel, comme l'utilisation de fichiers plutôt que d'emplacements mémoires. Le système d'exploitation peut-être mono ou multi-tâche, s'il partage le temps de calcul entre plusieurs programmes. Il peut aussi être temps réel, s'il assure des temps de réponse prédictibles, mono-utilisateur ou multi-utilisateurs, s'il permet la cohabitation de plusieurs utilisateurs, etc.

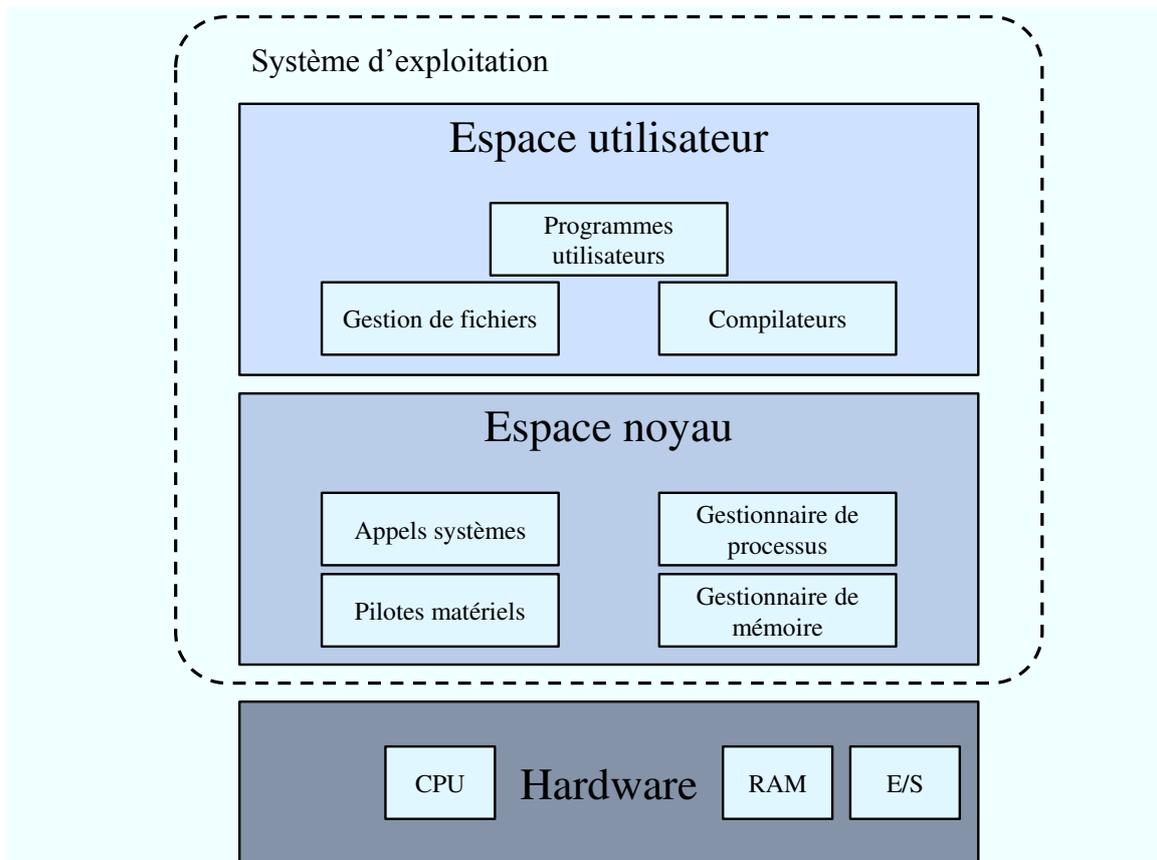


FIGURE 3.2 – Architecture d'un système d'exploitation [107]

Les systèmes d'exploitation les plus répandus présentent une architecture en couches, similaire à celle présentée sur la figure 3.2. Cette architecture se compose d'une couche noyau qui est le cœur du système d'exploitation et d'une couche utilisateur. La couche noyau est constituée des modules qui fournissent des mécanismes de bas niveau et interagissent directement avec le matériel, comme le gestionnaire de mémoire ou les pilotes matériels. L'espace noyau est le cœur du système. Il gère le matériel et autorise leur accès aux logiciels et bibliothèques situées dans la couche utilisateur. L'espace utilisateur possède les applications de plus haut niveau, et interagit avec les utilisateurs.

La notion qui nous intéresse est l'aspect multitâche que proposent la plupart des systèmes d'exploitation généralistes, et plus particulièrement le multitâche préemptif. Dans

un système multitâche, plusieurs programmes peuvent être en cours d'exécution en même temps sur un même ordinateur. L'ordonnanceur, qui est un composant du noyau, choisit l'ordre d'exécution des processus et distribue leur temps d'utilisation. Ainsi, plusieurs processus peuvent cohabiter en même temps de manière transparente pour l'utilisateur. Dans un système multitâche préemptif, l'ordonnanceur peut interrompre à tout moment un processus en cours d'exécution pour permettre à une autre de s'exécuter. La mise en œuvre de ces mécanismes implique que le processus doit être préemptible. Pour faciliter la préemption, les processus sont affectés d'une priorité qui représente leur importance dans le système d'exploitation. Ainsi, un processus considéré comme important ne sera pas gêné dans son exécution par un autre d'importance moindre.

En partant de ces notions, nous avons construit une application, de déploiement de services d'aide au handicap, qui se comporte comme un système d'exploitation multitâche préemptif. A l'instar, d'un système d'exploitation qui distribue ses processus sur ses différentes unités de calculs, nous distribuons les tâches d'aide au handicap sur différentes ressources informatiques. De fait, notre application est vue comme une couche d'abstraction supplémentaire et commune à toutes les ressources informatiques. Une vision schématique est représentée sur la figure 3.3.

Cependant, notre application de déploiement ne se substitue pas à un système d'exploitation pour plusieurs raisons. La première est que, le programme que nous avons développé est lui-même une application exécutée par une ressource informatique et donc gérée par un système d'exploitation. De plus, les tâches d'aide au handicap déployées par notre application requièrent elles aussi un système d'exploitation pour être exécutées. La seconde raison est que même si nous implémentons des mécanismes de préemption de tâche d'aide au handicap, cela ne perturbe pas et n'est pas non plus perturbé par les fonctions de préemption des systèmes d'exploitation présents sur les ressources. Par exemple, une tâche d'aide au handicap déployée sur un ordinateur sera considérée comme « en cours d'exécution » par notre application de déploiement et cela même si le système d'exploitation du système concerné la préempte naturellement. Par contre, cette même tâche d'aide au handicap pourra, si besoin, être préemptée directement par notre application de déploiement pour faciliter l'exécution d'un autre service. Enfin, il est important de préciser que les techniques de préemption que nous implémentons sont basées et utilisent les signaux d'interruption mis à disposition par les systèmes d'exploitation.

Pour résumer, nous pouvons représenter notre application de déploiement comme un système d'exploitation, ayant connaissance de son contexte d'environnement, et « régissant » les systèmes d'exploitation présents sur les ressources informatiques de l'espace domotique.

Les tâches d'aide au handicap, que nous utilisons, lorsqu'elles sont déployées sur des ressources, sont matérialisées informatiquement sous la forme de commandes et d'applications Unix/Linux, mais aussi d'applications Java. En d'autres termes, il est possible d'observer un service domotique à deux niveaux différents :

- Au niveau de l'application de déploiement : les services sont vus comme des tâches d'aide au handicap avec leur modèle et leurs propriétés. Ces services peuvent alors être déployés sur des ressources informatiques, mais resteront considérés par l'application de déploiement comme des tâches d'aide au handicap. Nous pourrions parler d'état global d'une tâche d'aide au handicap.
- Au niveau d'une ressource informatique : Lorsqu'une tâche d'aide au handicap est déployée sur une machine, l'exécution de la commande ou du pro-

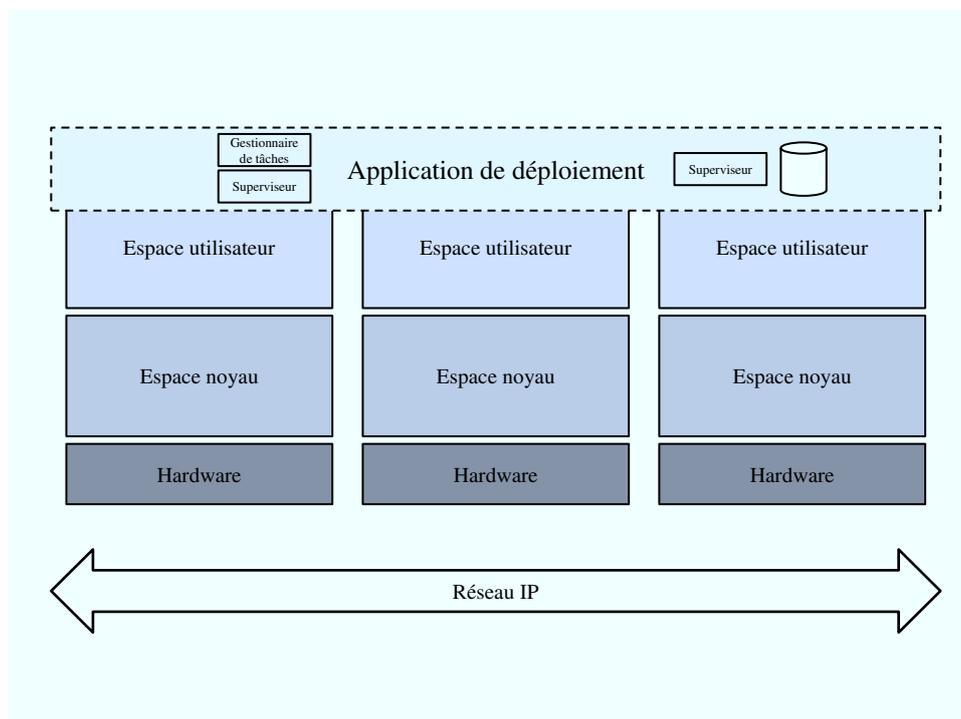


FIGURE 3.3 – Vision abstraite de l'application de déploiement

gramme associé crée un processus géré par le système d'exploitation de la ressource informatique. Aussi, il est possible de regarder l'état local du processus sur l'ordinateur qui l'exécute.

Nous venons d'analyser la matérialisation d'une tâche d'aide au handicap déployée sur une ressource informatique. Étudions maintenant les différentes formes qu'elles peuvent prendre. Comme nous l'avons précisé, au niveau d'une ressource informatique la forme de la tâche d'aide au handicap dépend de la couche d'application sur laquelle elle est exécutée. Observons, représentés sur la figure 3.4, les principales couches d'application utilisateurs que nous souhaitons adresser

- La première couche est la couche du système d'exploitation, dans notre cas Linux. À cet échelon, nous retrouvons les commandes Unix comme la fonction kill, utilisée pour stopper des processus, le démon EIB qui met en application des fonctions de communication avec le bus EIB, etc.

- La seconde couche s'appuie sur la première et est celle de la machine virtuelle Java (JVM). La JVM autorise l'exécution d'application développée en Java. Nous retrouvons à ce niveau par exemple notre application moniteur, l'interface utilisateur du système ou le Framework OSGi. Bien qu'utilisant une couche d'abstraction supplémentaire, ces applications Java sont traduites informatiquement sous la forme de processus au même titre que les programmes Linux.

- La dernière couche, que nous adressons, est le Framework OSGi. L'implémentation du Framework OSGi que nous utilisons est une application Java qui s'appuie donc sur la couche précédente. Ce Framework nous permet de lancer

des bundles, composants d'applications Java, autogérées. De nombreuses applications sont développées suivant les spécifications OSGi, pour des raisons que nous expliquons plus tard.

Les caractéristiques de chaque couche adressées sont décrites plus précisément, dans les sections suivantes.

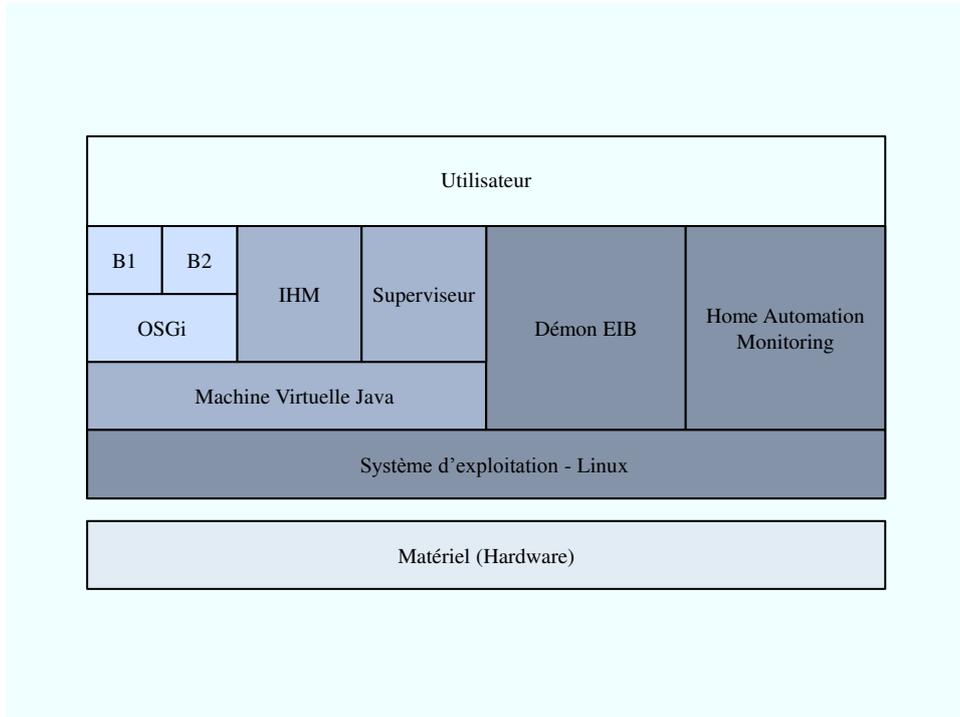


FIGURE 3.4 – Couches d'applications utilisateurs

### 3.1.1 Le système d'exploitation : Linux

Le système d'exploitation Linux [107] caractérise notre infrastructure informatique, puisqu'il en est la base. Son utilisation a été motivée par le fait qu'il est depuis des années à la base de la plupart des travaux, sur la thématique d'aide au handicap, au LabSTICC [108]. C'est un système d'exploitation libre et open-source. Comme la plupart des systèmes d'exploitation généralistes il présente une architecture similaire à celle étudiée figure 3.2. Linux est un système d'exploitation de type Unix. Il peut être exécuté sur différentes plateformes matérielles : x86, Sparc, PowerPC, Alpha, ARM, etc. Cet aspect portabilité nous permet de faire fonctionner Linux, sur des ressources diverses et variées telles que des ordinateurs, des assistants personnels mobiles, des consoles de jeux, etc. Dans l'objectif de mettre à profit toutes les ressources informatiques présentes dans l'environnement, Linux joue un rôle important pour la suite de nos travaux. En plus d'être un système multi-plateformes et multi-utilisateurs, Linux est aussi un système multitâche. Nous pouvons reprendre à notre compte les mécanismes de préemption qu'il implémente.

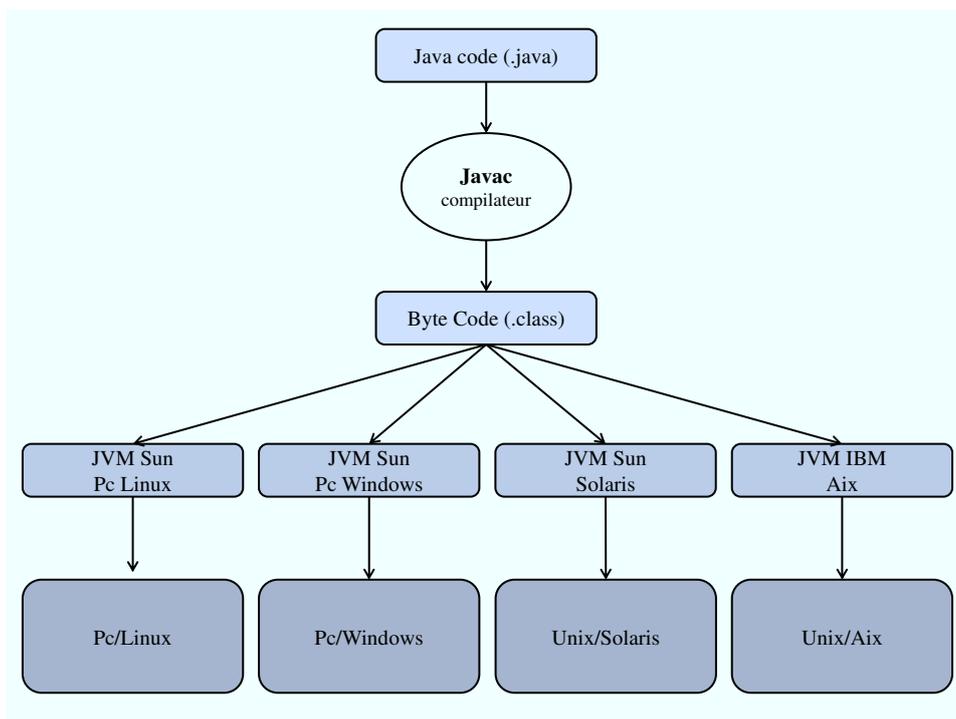


FIGURE 3.5 – Architecture Java

### 3.1.2 La machine virtuelle Java

Java [109] est un langage de programmation très répandu. De nombreuses applications développées au Lab-STICC ou au DOMUS, que nous utilisons, sont codées en Java. Java est un langage de programmation simple, par exemple la gestion de la mémoire est transparente pour l'utilisateur. Il est orienté objet, où chaque objet est une entité munie de propriétés et méthodes capables d'agir sur ses propriétés ou de réaliser certaines actions. Mais Java est surtout un langage indépendant des architectures matérielles. En effet, le compilateur génère un code universel appelé le « byte-code ». La machine virtuelle est un interpréteur privilégié spécifique à l'ordinateur qui permet l'exécution de programmes indépendamment du support matériel. La représentation des données étant indépendante de la machine qui exécute le code, les résultats des calculs sont donc eux aussi indépendants. Une machine virtuelle est un logiciel qui émule le fonctionnement d'un processeur physique. Elle constitue une couche intermédiaire entre le programme Java compilé et le système d'exploitation. La standardisation de Java et du fonctionnement de sa JVM ont offert des implémentations sur la plupart des systèmes d'exploitation et font de Java un langage de programmation facilement portable, comme le montre la figure 3.5. Cette dernière figure illustre le slogan bien connu du langage « Compile once, run everywhere » : Compiler une seule fois, exécuter n'importe où. Enfin, la machine virtuelle Java est elle-même multitâche et autorise l'exécution simultanée de plusieurs processus. Dans le cadre de nos travaux, nous utilisons une implémentation Linux de la JVM.

### 3.1.3 Le Framework OSGi

L'utilisation d'OSGi [110] nous a été inspirée lors d'une mobilité au laboratoire Domus. Une partie de leurs travaux de recherche sur l'analyse comportementale et l'assistance au handicap est codée pour être supportée par OSGi. La technologie OSGi, Open Service Gateway Initiative, est soutenue par l'OSGi Alliance [111]. Elle spécifie un Framework de déploiement et d'exécution de service Java. OSGi s'appuie sur les principes de la programmation orientée composants et des architectures orientées service (SOA, Service Oriented Architecture). La programmation par composant tend à corriger les limitations qu'apporte la programmation-objet. La programmation-objet ne propose pas la mise en relation des classes qu'elle définit. Aussi, elle n'offre aucun mécanisme natif pour la résolution des problèmes d'évolutivité de maintenance, lorsque l'application devient complexe. Pourtant les objets mis en relation peuvent éprouver des dépendances fortes et le maintien ou l'évolution de ce code ne pourra se faire qu'au prix d'un travail long et pénible. Pour diminuer l'importance des couplages entre les classes, OSGi repose sur les architectures orientées services. Cette approche apporte la notion de fournisseur de service et de consommateur de service aux composants et de ce fait un couplage faible, modulable et reconfigurable s'opère entre ces éléments. Ces deux technologies complémentaires sont associées dans OSGi. Sur la figure 3.6a, nous avons déclaré deux composants, fournissant chacun des services au travers d'interfaces. Ces composants, appelés 'Bundle' expriment, entre autres, un groupe de classes Java. Nous notons que ces deux composants sont liés entre eux par l'un des services. Cependant, les services ne sont pas les seuls mécanismes autorisant les liens entre composants puisque OSGi utilise aussi l'import et l'export des packages au travers de son canevas. Ces mécanismes d'échanges de dépendances sont gérés par le Framework OSGi. OSGi est donc employé dans le but de résoudre les restrictions de la programmation orientée objet, d'améliorer et faciliter l'évolution et la gestion des applications complexes et d'autoriser la mise en œuvre d'architectures orientées service.

Dans nos travaux, les applications OSGi déployées profitent de cette couche d'abstraction pour résoudre automatiquement certaines dépendances. La flexibilité apportée par le Framework autorise les applications à se modifier légèrement leur comportement, sans stopper leur fonctionnement, pour pallier des défauts et ainsi garantir une étape de fiabilité supplémentaire. Nous pouvons illustrer ce dernier exemple avec une application d'alerte qui affiche son message sur un écran. Si aucun écran n'est disponible et qu'aucune solution ne s'offre à nous, il suffit de stopper le bundle gérant l'affichage vidéo et de mettre à jour l'application en chargeant un nouveau bundle utilisant les capacités audio pour transmettre son information.

Sans rentrer dans les détails des caractéristiques du Framework OSGi, un bundle se décompose en plusieurs couches, voir figure 3.6b. Outre, la couche service qui introduit les mécanismes d'échanges entre les composants et la couche module qui permet la mise en œuvre des composants et gère leur interopérabilité, nous retrouvons une couche cycle de vie. Cette dernière couche correspond à l'état du bundle dans le conteneur. En effet, le conteneur OSGi à la responsabilité de gérer le cycle de vie des bundles. Toutefois, il propose une API standardisée que nous utilisons pour exécuter les bundles au travers de notre système de déploiement. Apache Felix est une implémentation, open source, de la plateforme OSGi conçue pour être légère et facile d'utilisation. C'est le conteneur que nous retenons pour gérer nos bundles dans notre environnement.

OSGi s'inscrit dans notre système au travers de l'environnement Java, comme schématisé sur la figure 3.4. La vie de ces applications s'intègre dans le cycle que nous spécifions

dans la section suivante.

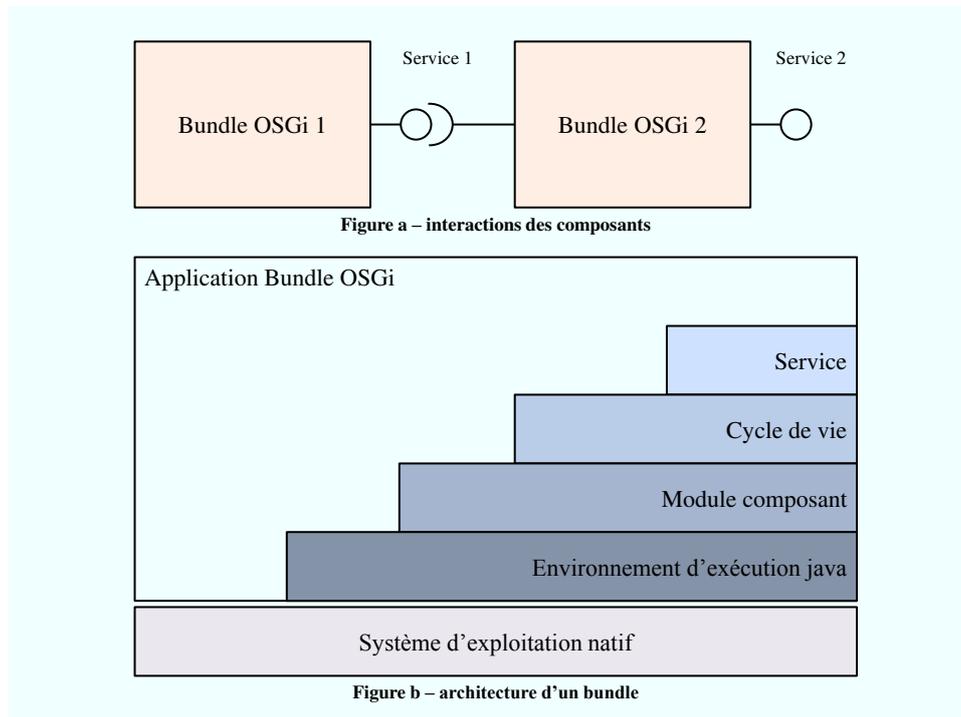


FIGURE 3.6 – Schémas de principe d'OSGi

### 3.1.4 Diagrammes d'état et cycle de vie

Notre système multitâche doit jongler avec les différents processus qui lui sont proposés. Dans notre application de déploiement, nous intégrons toutes les applications exécutables sur un système d'exploitation Linux ou les applications Java. Nous avons aussi vu l'intérêt des bundles OSGi qui apportent une flexibilité dans la gestion de ses composants et nous avons souhaité pouvoir gérer ce type d'application aussi. Notre système assure le déploiement de ces tâches d'aide au handicap, au niveau des différentes ressources de l'environnement. Comme nous l'avons vu dans la section 1, à l'échelle d'une ressource, la gestion des processus est garantie par le système d'exploitation, exception faite des bundles gérés indépendamment par le Framework OSGi.

#### 3.1.4.1 Les processus Unix

Une application d'aide au handicap, codée en Java par exemple, exécutée sur une ressource se transforme en processus informatique pendant l'exécution. Ce processus suit un cycle de vie dépendant de son système d'exploitation. Dans notre cas, nous utilisons Linux et les tâches suivent les états schématisés sur la figure 3.7. Un processus peut donc être dans l'un des états suivants :

- Exécution (R) : Le processus est élu et en cours d'exécution
- Sommeil (S) : Le processus est interrompu dans un multitâche préemptif au bout d'un quantum de temps

- Arrêt (T) : Le processus est temporairement arrêté par un signal. Il ne peut plus être appelé pour s'exécuter et ne réagira qu'à un signal de redémarrage
- Zombie (Z) : Le processus s'est terminé, mais son père n'a pas lu son code de retour

L'état qui nous intéresse le plus est celui d'arrêt (T). Le processus est placé en pause par un signal. Or Unix, nous offre les mécanismes pour gérer les signaux grâce à la commande « Kill ». La commande « Kill » et plus particulièrement « Sigstop » et « Sigcont » nous permettent de suspendre un processus pour libérer les ressources et le reprendre ensuite.

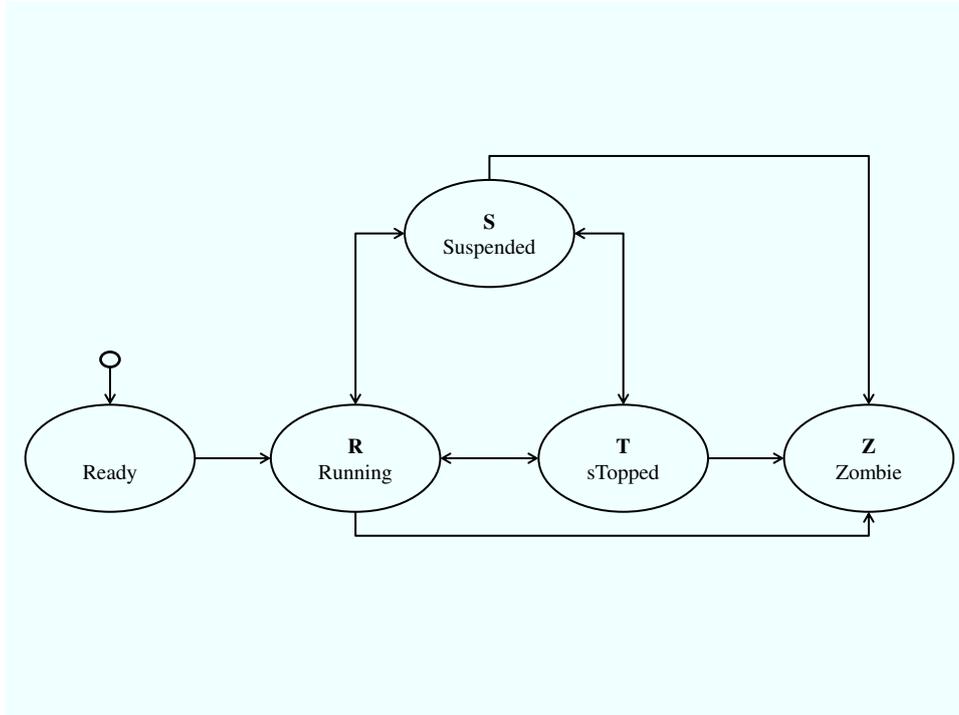


FIGURE 3.7 – Cycle de vie simplifié d'un processus Unix

### 3.1.4.2 Les bundles OSGi

Dans la section 3.1.3 de ce chapitre, nous avons introduit OSGi. OSGi est utilisé pour déployer des applications sur une plateforme gérée dynamiquement, en utilisant un modèle orienté composant. Ces composants peuvent ainsi être installés, démarrés, arrêtés, mis à jour, et désinstallés sans redémarrage du Framework OSGi. Le Framework s'occupe entièrement de la gestion de leur cycle de vie. En d'autres termes, le Framework OSGi offre la possibilité de déployer les bundles de manière continue. Les étapes du cycle de vie d'un bundle sont représentées sur la figure 3.8. Lors du déploiement sur la plateforme, le bundle est installé. Si toutes les dépendances requises sont satisfaites, le bundle passe dans l'état résolu. Cette étape permet de mettre à jour le bundle, par exemple, mais aussi de le démarrer. Il passe ensuite dans l'état actif et effectue ses traitements. Lorsqu'un bundle est activé, le Framework OSGi appelle une méthode qui signale qu'il est démarré. Il peut, par exemple, publier ses services, découvrir les services disponibles.

L'implémentation que nous utilisons est prévue pour facilement être lançable et embarquée grâce à son API standardisée. Une ou plusieurs instances du Framework peuvent être créées sur la machine virtuelle Java.

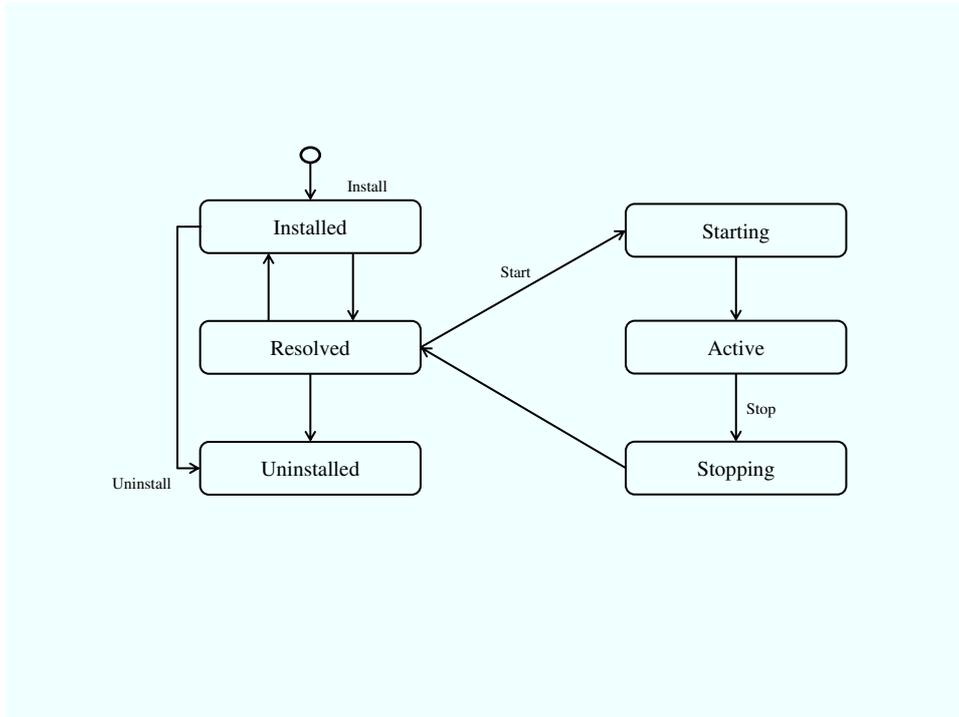


FIGURE 3.8 – Cycle de vie d'un bundle OSGi

## 3.2 Le flot de déploiement de tâches

Avant d'entrer dans le vif du sujet, reprenons simplement la problématique de notre projet. Nous considérons que les services proposés et mis à disposition du patient et de son système d'assistance sont de plus en plus complexes, et par conséquent requièrent de plus grandes capacités pour être réalisés. Paradoxalement, les systèmes d'assistances ne peuvent pas suivre cette tendance pour des raisons principalement économiques. Cependant, l'informatique est de plus en plus présente dans la vie de tous les jours, et plus particulièrement dans notre espace de vie. Pour profiter de cet avantage technologique, nous avons proposé des modèles adressant l'ensemble des éléments susceptibles de participer à l'exécution des services d'aide au handicap. Ainsi, nous proposons d'utiliser la plupart des équipements disponibles dans l'environnement de l'utilisateur. Toutefois, chaque ressource se gère indépendamment et n'a pas connaissance de son environnement. C'est pourquoi, nous proposons dans ce chapitre une application de déploiement qui permet dans contrôler de manière globale toutes les ressources présentes dans l'environnement et localement l'exécution des tâches sur chaque unité de traitement.

Le schéma du flot de déploiement est décrit sur la figure 3.9. La partie supérieure représente les étapes de conception du système.

Nous considérons la partie « Domaine de service » comme les entrées de notre système. En effet, pour compenser certaines situations de handicap l'utilisateur est amené

à exécuter un service d'assistance. Ces services composent l'offre de service proposée au patient. Elle est définie par le corps médical en fonction des besoins de l'utilisateur et de l'environnement dont il dispose. Cette phase de conception est considérée comme relevant du niveau fonctionnel et nous n'adresses pas ce point puisque d'autres travaux ont déjà apporté des résultats significatifs comme le projet IntelHome [78]. Nous nous intéressons principalement aux événements, les services, que l'utilisateur va souhaiter déployer.

La partie « Domaine de conception » reprend les modèles décrits dans les travaux d'IntelHome pour appuyer sur la modélisation des capacités matérielles et des besoins logiciels, en se focalisant sur une vision plus bas niveau, comme nous l'avons étudié dans le chapitre 2. Ces représentations sont compilées dans une base de connaissances, l'ontologie de conception, qui définit l'ensemble du système d'assistance.

Sur la partie inférieure de ce schéma, est représentée notre application de déploiement avec ses deux modules principaux le gestionnaire de tâches et le moniteur de ressources, dont nous reviendrons plus en détail dans la suite de ce document. Enfin, au centre de notre flot de déploiement se trouve l'ontologie de travail, une base de connaissances dynamique, qui définit en temps réel le système d'assistance.

Intéressons-nous aux trois composants principaux qui composent notre flot de déploiement :

- Le gestionnaire de tâches : Cette application assure le cycle de vie de la tâche, à partir du moment où elle est choisie par le patient jusqu'à son exécution dans l'environnement domotique. Pour un service d'aide au handicap désiré, la tâche informatique associée est étudiée. Les besoins matériels et logiciels exigés sont confrontés aux capacités et dépendances logicielles proposées par les ressources. Le but est d'obtenir l'ensemble des solutions « tâche/ressource » compatibles. La ressource optimale est sélectionnée grâce à la connaissance des taux d'utilisation de chaque ressource. Avant d'être déployée, le gestionnaire de tâche s'assure que l'exécution ne perturbera pas le fonctionnement et l'intégrité des tâches déjà en cours. Une fois toutes ces étapes validées, la tâche est ainsi déployée sur la ressource pour satisfaire les besoins du patient.
- Le moniteur de ressources : En parallèle de la gestion des tâches, l'application moniteur met-à jour les ressources informatiques présentes dans l'environnement domotique. Comme nous l'avons expliqué précédemment, l'application de déploiement tire sa force de la connaissance fidèle de son contexte. En effet, le gestionnaire de tâche sélectionne la ressource idéale sur les critères de charge de cette dernière. Or les ressources et leur utilisation évoluent au cours du temps et des différentes manipulations qu'il subit. Le moniteur a pour but d'interroger à tour de rôle les différentes ressources informatiques et de récupérer leur disponibilité et leurs taux d'utilisation. Toutes les informations sont inscrites dans l'ontologie de travail.
- Le manager d'ontologie : Cet élément crée l'ontologie de travail qui interagit, par la suite, avec le gestionnaire de tâche et le moniteur. Il peut générer l'ontologie de travail à partir de l'ontologie de conception si une modification est effectuée par les concepteurs ou à l'initialisation du système. Toutefois, il peut aussi être demandé par le moniteur. Dans ce cas, l'ontologie de travail est réalisée à partir de l'ontologie de travail précédente. Le manager d'ontologie infère les nouvelles relations sémantiques, grâce à son moteur d'inférence.

Ces trois entités sont développées plus précisément dans la suite de ce chapitre. Pour

finir la présentation globale de l'application, nous présentons rapidement les deux éléments secondaires la composant :

- L'interface utilisateur : Elle représente le dispositif de communication entre le système et l'utilisateur. Bien qu'il soit indispensable que cette application soit correctement adressée et étudiée, elle ne figure pas comme une partie essentielle de nos travaux de recherche. L'interface utilisateur est, en raison de son interface entre le patient et le système, le point de départ indispensable de notre flot de déploiement. Il est le traducteur informatique particulier des besoins de notre patient. Cependant, d'autres travaux gèrent très bien la conception de ce dispositif, c'est le cas du projet ASIM [78]. L'interface utilisée a pris plusieurs apparences dans nos travaux. Nous avons utilisé, au début de nos recherches, le middleware DANAH [67], qui permet au patient d'interagir avec son environnement. DANAH traite beaucoup d'aspects, en plus de l'interface utilisateur, que nous n'avons pas souhaité aborder. Cela rendait complexe l'intégration avec notre travail. Nous avons préféré nous concentrer sur une preuve de concept en nous focalisant sur d'autres aspects. Nous sommes donc revenus sur une application plus simple, certainement moins ergonomique et moins adaptée aux personnes handicapées aussi.
- L'ontologie de conception : Cette base de connaissances est créée à l'étape de conception (chapitre 2). Elle regroupe les informations sur le contexte du système avec la description des tâches et la modélisation des ressources. Bien qu'elle n'ait pas d'application directe d'un point de vue exécution, cette ontologie est à l'origine de l'ontologie de travail.

L'ontologie de travail est un cas particulier que nous ne décrivons pas au même niveau que les 3 éléments principaux, constituant le flot de déploiement. L'ontologie de travail est une copie de l'ontologie de conception, qui a longuement été traitée dans le chapitre 2. En outre, les évolutions de l'ontologie de travail sont reprises dans la description de chaque constituant principal puisqu'elle en est le point central et que tous les éléments s'y réfèrent, que ce soit pour l'interroger sur les individus de l'environnement ou les mettre à jour. C'est une base de connaissances dynamique qui exprime en temps réel les caractéristiques des ressources. Pour prendre un exemple concret, le pourcentage d'utilisation des processeurs d'une ressource, à l'instant « t », est indiqué dans l'ontologie. Cette représentation fidèle des caractéristiques participe au choix de la ressource optimale.

Suivons le déploiement d'un service, sur la figure 3.9, pour expliciter le flot de déploiement. Le patient se trouve dans une situation, qu'il ne peut résoudre seul, autrement dit il est en situation de handicap. Pour palier cette situation de handicap, le patient utilise l'interface utilisateur de son système d'aide au handicap. Rappelons que le dimensionnement du système et des besoins d'assistance du patient n'est pas adressé dans nos travaux mais sont des entrées de notre application que nous nous chargeons d'exploiter. Toutefois, il se peut qu'un service comme une alerte ou une tâche périodique ou encore le système lui même puisse demander le déploiement d'un service, indépendamment des besoins du patient. Précisons, là encore, que l'offre de service est construite et proposée par le corps médical. Nous sommes dans le domaine de service, celui des besoins et des envies de l'utilisateur. Le service sélectionné est interprété par le gestionnaire de tâche comme la tâche d'aide au handicap à exécuter. Puis, grâce notamment aux informations sur le contexte et a un algorithme de minimisation de la charge, le gestionnaire détermine la ressource la plus adaptée pour réaliser le service du patient. Cette solution tâche/ressource est alors

exécutée par le gestionnaire. Nous sommes dans le domaine de la tâche d'aide au handicap et du système domotique. Autrement dit, l'utilisateur n'intervient pas dans ces étapes qui lui sont totalement transparentes. Parallèlement aux traitements du gestionnaire, le moniteur interroge l'ensemble des ressources présentes dans l'environnement et met à jour l'ensemble des valeurs interrogées. Nous sommes dans le domaine des ressources et de celui de l'environnement puisque les ressources sont au centre de l'attention. Dans le coin supérieur gauche de la figure 3.9, le domaine de conception est hérité des étapes de modélisation que nous avons étudiées dans le chapitre 2. En effet, on y retrouve l'ontologie de conception manipulée par le manager d'ontologie qui générera, à partir de cette dernière, l'ontologie de travail. Pour finir, rappelons que l'ontologie de travail est au centre de tous les domaines puisqu'elle concerne quasiment toutes les étapes du flot à l'exception du domaine de service, qui comme nous l'avons n'est pas non plus adressé par nos recherches.

Dans la suite de cette partie, nous allons donc nous attarder sur la compréhension des éléments principaux qui constituent le flot de conception.

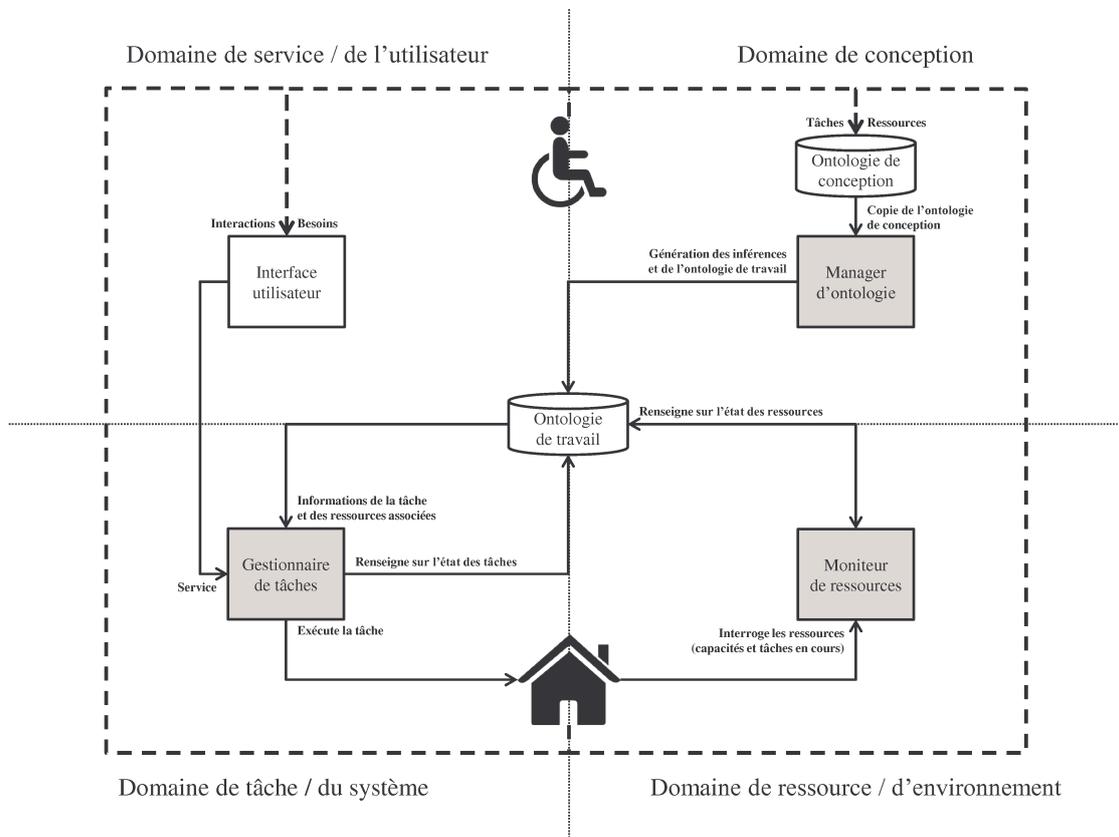


FIGURE 3.9 – Vue globale du flot de déploiement

### 3.2.1 Le manager d'ontologie

Le manager d'ontologie permet de garantir la cohérence entre l'ontologie de travail et l'ontologie de conception pour garder l'ontologie à jour au gré des modifications du système. Si un changement intervient dans la conception, il produira une nouvelle ontologie de travail cohérente par rapport à l'évolution des spécifications. Par exemple, si

l'état du patient se dégrade, l'ergothérapeute souhaite rajouter de nouveaux services dans le système. L'intégration de ces services entraîne une modification de la modélisation, de l'ontologie de conception. Par conséquent, l'ontologie de travail est mise à jour par le manager d'ontologie. Sur le même principe, lorsque le moniteur détecte une anomalie comme une ressource inaccessible ou défectueuse, il peut être amené à modifier directement l'ontologie de conception, en supprimant une ressource, si cette dernière ne revient pas dans un état de fonctionnement. La modification de l'ontologie de conception entraîne le manager à remettre à jour l'ontologie de travail pour prendre en compte la modification profonde du système. Le manager fait appel au moteur d'inférence pour prendre en compte les nouveaux éléments et ainsi supprimer les informations sémantiques qui ne sont plus d'actualité ou dans le cas contraire enrichir le modèle.

La figure 3.10a montre le manager d'ontologie et ses interactions avec les autres éléments. On retrouve donc l'ontologie de conception, qui est le modèle du système décrit dans le chapitre 2. Construit à partir des spécifications du patient et de l'environnement domotique, il est l'image initiale du système, conçu par les spécialistes. En cas de dysfonctionnement du système, il permet la réinitialisation de notre flot de déploiement dans son état d'origine. L'autre élément important qui appartient à l'univers du manager est l'ontologie de travail. C'est une copie exacte du fichier OWL de l'ontologie de conception qui a été enrichie des nouvelles informations sémantiques par le moteur d'inférence. Parmi les informations ajoutées, nous retrouvons, les capacités propres à chaque ressource, tout comme leurs propriétés d'utilisation (taux, usage unique, etc.).

L'appel au manager d'ontologie s'effectue à l'initialisation du flot de déploiement ou à la suite de la modification d'un individu de l'ontologie de conception, par une tierce personne ou par le moniteur. L'algorithme associé au manager d'ontologie, figure 3.10b, se découpe en deux branches, où l'on fait une copie de l'ontologie de conception ou de travail, en fonction du type d'appel. La procédure ensuite est commune, le manager utilise le moteur d'inférence pour enrichir la description et créer la nouvelle ontologie de travail.

### 3.2.2 Le gestionnaire de tâches

Le gestionnaire de tâches est l'organe principal de notre système de déploiement. Dès lors qu'un service est demandé par le patient, la tâche d'aide au handicap associée est alors analysée dans la base de connaissances. Les informations relatives à cette tâche sont récupérées et une recherche de la ressource optimale est lancée. Le gestionnaire assure le bon déroulement du déploiement jusqu'à l'exécution de la commande de la tâche d'aide au handicap. Le choix de la meilleure ressource est décidé à l'aide d'un algorithme de minimisation des taux d'utilisation des capacités. En d'autres termes, on va choisir la ressource qui dispose le plus de charge processeur et de mémoire vive, notamment. Le gestionnaire va aussi gérer l'ordonnancement des tâches, si besoin. Il tient à jour une table de tâche d'aide au handicap déployée, qui liste toutes les tâches en cours sur des ressources. Cette table est nommée « liste des tâches actives ».

Dans la liste des tâches actives, nous retrouvons, comme décrits sur la figure 3.11, des informations sur la ressource d'affectation. Ainsi, pour chaque ressource présente dans l'environnement, nous pouvons suivre toutes les applications déployées. Elles sont identifiées par leur PID, pour Process Identifier. Le PID est attribué par le système d'exploitation lorsqu'une tâche est instanciée sur un ordinateur. Cet identifiant, ainsi que l'état local de la tâche (PID Status), sont récupérés par le gestionnaire, juste après l'exécution de la

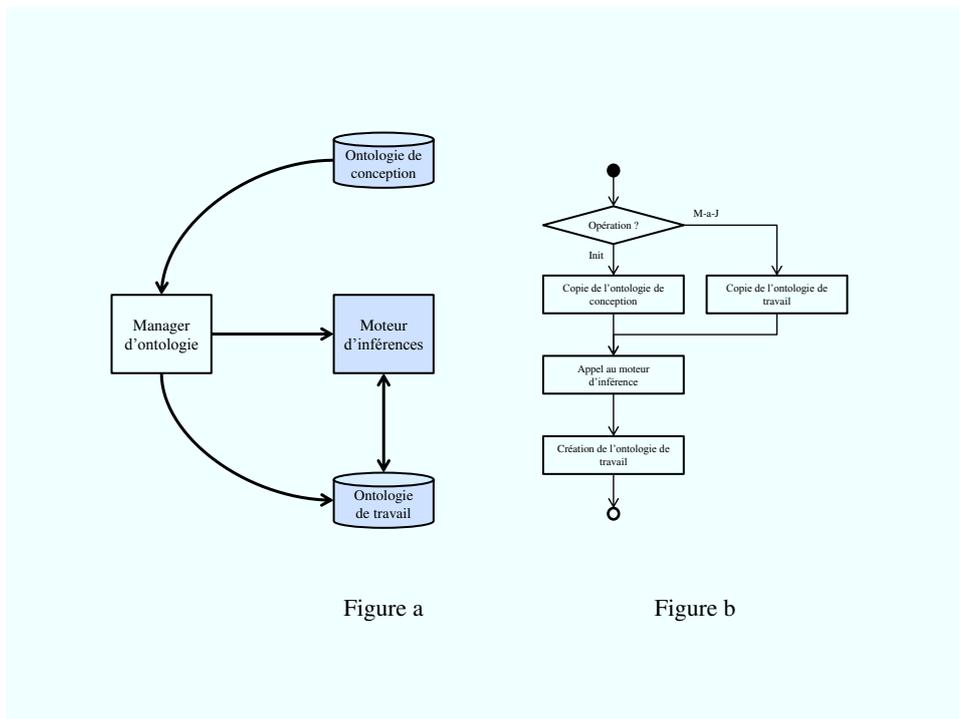


FIGURE 3.10 – Le manager d'ontologie

commande. Enfin, le paramètre SusPID, pour Suspended PID, est une information renseignée par le gestionnaire, si un ordonnancement a donné lieu à la suspension d'une tâche précédemment installée pour exécuter la nouvelle.

Remarque : La liste des tâches d'aide au handicap déployées est utilisée notamment par le moniteur pour vérifier l'état local des processus déployés sur chaque machine. Cet état est ensuite rapporté dans la table des tâches actives pour garantir le bon fonctionnement global du système.

Le gestionnaire de tâche se décompose en trois phases : la décision, l'ordonnancement et l'exécution. Le comportement du gestionnaire est représenté sur la figure 3.12 et est expliqué dans les sections suivantes : phase de décision 3.2.2.1, phase d'ordonnancement 3.2.2.2, phase d'exécution 3.2.2.3.

Ressource d'affectation [string]	Process Identifier (PID) [integer]	PID Status [string]	Suspended PID List [integer]
-------------------------------------	---------------------------------------	------------------------	---------------------------------

FIGURE 3.11 – Liste des tâches actives

### 3.2.2.1 Phase de décision

Suite à la requête du patient traduite en tâche, le gestionnaire effectue une recherche de la ressource optimale dans la phase de décision. Le gestionnaire, dans un premier temps,

effectue une recherche de toutes les ressources capables de réaliser la tâche d'aide au handicap. Les ressources obtenues sont listées dans « la table des ressources possibles ». Les besoins et dépendances des tâches sont confrontés aux capacités et applications logicielles de chaque ressource présente dans l'environnement. Pour accéder à ces informations, le gestionnaire interroge l'ontologie.

Si aucune ressource n'est capable de réaliser la tâche, en raison de la charge du système par exemple, le gestionnaire peut décider de passer à une version alternative du service, si possible. Les versions alternatives sont des tâches d'aide au handicap à part entière et devront donc passer à nouveau par une recherche de ressource depuis le début. S'il existe plusieurs solutions, alors une étape secondaire est lancée, pour choisir la ressource optimale, dans la liste des ressources possibles. À cet instant, le taux d'utilisations des capacités des ressources est analysé. Notre but est de minimiser la charge pour la répartir uniformément sur l'ensemble du système. Lorsque le choix est effectué, une seule solution d'un couple tâche ressource est proposé. Que la ressource choisie soit l'unique solution possible ou le résultat de notre fonction d'optimisation, nous devons nous assurer que le déploiement de la tâche ne crée pas un conflit avec d'autres tâches déjà déployées sur cette ressource. C'est pendant la phase d'ordonnancement que cette fonction est effectuée.

### **3.2.2.2 Phase d'ordonnancement**

Lorsque la ressource optimale est élue, une vérification sur la compatibilité avec les autres tâches déployées précédemment sur cette dernière est effectuée. Certaines tâches ne sont pas compatibles entre elles. On pense facilement à une alerte visuelle qui serait déployée sur une ressource projetant déjà un film. Le message pourrait ne pas être perçu par le patient. À l'inverse, un message visuel d'information sans importance pourrait s'afficher à la place du film, pendant un moment crucial et gâcher l'expérience du patient. En effet, nous avons vu cet exemple dans le chapitre 2, pour montrer que la capacité affichage est une capacité à usage unique et elle ne peut pas être partagée avec entre plusieurs tâches d'aide au handicap. Il existe des méthodes de partage d'écran ou d'incrustation d'image qui pourraient pallier ce problème, mais nous souhaitons mettre en avant le problème des capacités non partageable de certaines ressources.

En fonction des paramètres des tâches, comme leur préemption et leur priorité, le gestionnaire se charge de les ordonner. Dans la phase d'ordonnancement, il va tester la préemption de la tâche conflictuelle, déjà en place sur la machine. Si cette dernière peut être suspendue, le gestionnaire va regarder les priorités associées à chacune des deux tâches. Si la priorité de la nouvelle tâche est supérieure à la tâche en cours, le gestionnaire va suspendre cette dernière grâce aux mécanismes proposés par le système d'exploitation. Si l'un des tests échoue (préemption et priorité), la ressource est supprimée de la liste des solutions possibles pour l'exécution de cette tâche d'aide au handicap et un nouveau cycle de sélection de la ressource optimale est entamé.

Si le déploiement d'une nouvelle tâche a entraîné la suspension d'une tâche précédemment installée, le PID de cette dernière est associé au PID du nouveau processus, dans la table des tâches actives, sous le nom de SusPID.

### **3.2.2.3 Phase d'exécution**

L'exécution d'une tâche est différente si le gestionnaire a dû préempter une autre tâche ou non. Dans le cas, d'une exécution sans conflit, le gestionnaire interroge à nouveau l'ontologie pour construire la commande d'exécution de la tâche d'aide au handicap.

La commande d'exécution se compose de trois paramètres principaux, dont les deux premiers (l'adresse réseau de la ressource et l'identifiant de l'utilisateur associé à cette ressource) servent à créer une commande ssh, pour accéder au système d'exploitation de la ressource. Enfin, le dernier paramètre est la commande d'activation de la tâche d'aide au handicap désirée. Si par exemple, la tâche désirée est « Écouter FranceInfo » sur le Laptop la commande complète reconstruite sera :

```
ssh cedric@192.168.10.7 vlc http://95.81.155.3/franceinfo/all/franceinfo.mp3
```

```
ssh identifiant de l'utilisateur@adresse réseau + commande d'application
```

Toutes ces données sont assemblées entre elles et le gestionnaire ordonne l'exécution de la commande construite à travers le réseau. À chaque exécution d'une tâche, la table de gestion de tâches actives, ou autrement dit des tâches déployées, est mise à jour. Toutes les informations (PID, SusPID, etc.) sur les tâches déployées sont enregistrées pour faciliter le contrôle de ces dernières par le moniteur.

Dans le cas d'une exécution avec préemption, l'instance du gestionnaire de tâche créé pour l'occasion se met en attente de la fin de la tâche déployée ou de son acquittement. Dès lors, la tâche précédemment suspendue est reprise et les informations contextuelles sont mises à jour.

Pour garantir une ontologie, et donc la connaissance du contexte, la plus fidèle possible de l'environnement et l'état de charge de ses ressources sont interrogés, à intervalles réguliers, par le moniteur de ressource. Étudions son comportement dans la section suivante.

### 3.2.3 Le moniteur de ressources

Le moniteur est l'un des organes principaux du flot de déploiement. Il a pour mission de renseigner l'ontologie de travail. Un déploiement optimisé des tâches ne peut s'effectuer que si la connaissance de son environnement est suffisamment décrite, précise et fidèle. La notion de précision est associée ici aux aspects temporels de la représentation. En effet, nous mesurons en « temps réel », à intervalles réguliers, les taux d'utilisation de chaque capacité de l'ensemble des ressources. Le moniteur fonctionne en continu et interroge à tour de rôle toutes les ressources dont il a la charge. Il informe alors l'ontologie des évolutions de ses équipements informatiques pour garder une image fidèle de son contexte.

Le moniteur est aussi chargé d'étudier l'évolution des tâches déployées sur chaque machine. En effet, nous avons vu que le flot de déploiement s'appuyait sur les mécanismes (signaux) du système d'exploitation pour ordonnancer les processus relatifs aux tâches d'aide au handicap, à l'échelle d'une ressource. L'état de ces processus est conservé localement sur la machine dans la table des processus. Nous devons aussi garder cette information pour connaître l'état de nos tâches d'aide au handicap, à l'échelle de l'environnement pour notre flot de déploiement. Le moniteur peut donc interroger la table des processus d'un système d'exploitation pour obtenir certaines informations comme le « PID » ou « PID Status ». En outre, le système peut lui-même préempter un processus pour ses besoins ou alors terminer ce dernier, si tous ses traitements ont été effectués. Dans ce cas là encore, les informations sur les processus sont gardées à un niveau local, mais ils peuvent concerner des tâches d'aide au handicap et nous avons besoin de conserver ces informations au niveau de notre flot. En d'autres termes, le système d'exploitation ne renseigne pas notre application de ses actions. Si l'exécution d'une tâche d'aide au handicap s'est terminée

naturellement, c'est le moniteur qui doit confirmer cette information pour la répercuter dans l'ontologie.

La figure 3.13 schématise le comportement du moniteur. Une première étape consiste à vérifier que la ressource est accessible à travers le réseau. Une requête PING est effectuée pour s'assurer du fonctionnement de cette dernière. Si dans la théorie, une absence de réponse ne signifie pas que la ressource est défaillante, son inaccessibilité la rend inutilisable pour le système. D'un point de vue déploiement, cette ressource sera retirée de l'ontologie si l'impossibilité de communiquer se prolonge ou si une confirmation d'une défaillance de la ressource est obtenue. Lorsque la ressource répond une connexion ssh est établie avec cette dernière. Le moniteur effectue, ensuite, une surveillance matérielle et logicielle de la ressource. Pour les aspects matériels, il interroge la base de connaissances et récupère la liste des capacités de cette ressource. Chaque capacité est ainsi testée. S'il s'agit d'une capacité variable comme la charge processeur, le taux d'utilisation de la mémoire vive, l'espace de stockage disponible, etc. le moniteur consulte la valeur et l'enregistre dans l'ontologie puis passe à la capacité suivante. Les capacités à usage unique, comme l'utilisation d'un écran par une tâche, ne sont pas testées par le moniteur. C'est le gestionnaire de tâche qui se charge de gérer les conflits lorsqu'une nouvelle tâche d'aide au handicap entre en conflit avec une déjà déployée.

Le moniteur est aussi en charge de l'état des tâches d'aide au handicap déployée. Il cherche dans la liste des tâches actives, sur la ressource informatique qu'il interroge, toutes les tâches censées être en cours d'exécution ou préemptée. Il compare le PID de la tâche déployée avec les PIDs en cours sur la machine. La présence du PID confirme que l'exécution de la tâche n'est pas encore terminée. Pour garantir cette hypothèse, le moniteur regarde l'état du processus. Ce dernier peut être renseigné comme zombie par le système d'exploitation et indique qu'il est terminé malgré sa présence dans la table des processus. Aussi, si le processus est inconnu ou que son état est zombie, le moniteur étudie la liste des PIDs déployés pour voir si un SusPID est renseigné. La présence d'un SusPID associé à la tâche actuellement analysée révèle qu'elle avait remplacé une tâche préemptée sur cette ressource, pour des raisons de priorités notamment. Si un SusPID est connu, le moniteur ordonne la reprise de la tâche préemptée par le gestionnaire. Lorsque l'étude des capacités et des tâches déployées sur cette ressource est terminée, le moniteur passe à la ressource suivante et effectue les mêmes observations. Enfin, quand toutes les ressources ont été interrogées le moniteur s'arrête et il sera relancé quand son intervalle de temps sera écoulé ou lorsqu'une exécution de tâche d'aide au handicap est requise.

Dans un environnement domotique composé d'équipements hétérogènes, nous souhaitons fournir au patient en situation de handicap les services d'assistance dont il requiert. Dans le chapitre 2, nous avons modélisé les éléments constituant cet espace domotique, ressources et tâches d'aide au handicap, et formalisé sa description dans une base de connaissances. En plus de vouloir distribuer des services d'aide au handicap, notre objectif est de garantir un système domotique fonctionnel en toute condition. Autrement dit, une surveillance de la congestion du système et une gestion des tâches d'aide au handicap concurrentielles doivent être mises en place pour assurer un système fiable.

Un flot de déploiement a donc été présenté dans ce chapitre. Ce flot s'appuie sur la description de l'environnement, la connaissance de son contexte, pour proposer des mécanismes de déploiement des tâches d'aide au handicap sur les ressources que nous jugeons optimales. Au niveau d'un système multi-ressource, nous répartissons uniformément la charge de traitement. Toujours en utilisant la compréhension de l'environnement qui l'entoure, le flot de conception s'assure au niveau de chaque ressource que les tâches d'aide au handicap concurrentielles ne sont pas en conflit, entre elles. En cas de défaut, des mécanismes de préemption sont mis en place pour régler ce conflit. Pour s'assurer d'une connaissance du contexte fidèle, une analyse des ressources de l'environnement est effectuée. Ainsi, toutes les informations sur les capacités des ressources et leur état sont conservées dans la base de connaissances.

Dans le chapitre suivant, nous présenterons la mise en œuvre du système d'assistance pour éclairer les définitions présentées dans ce document, modélisation et flot de déploiement. Enfin, nous discuterons des résultats et analyserons l'acceptabilité de notre solution.

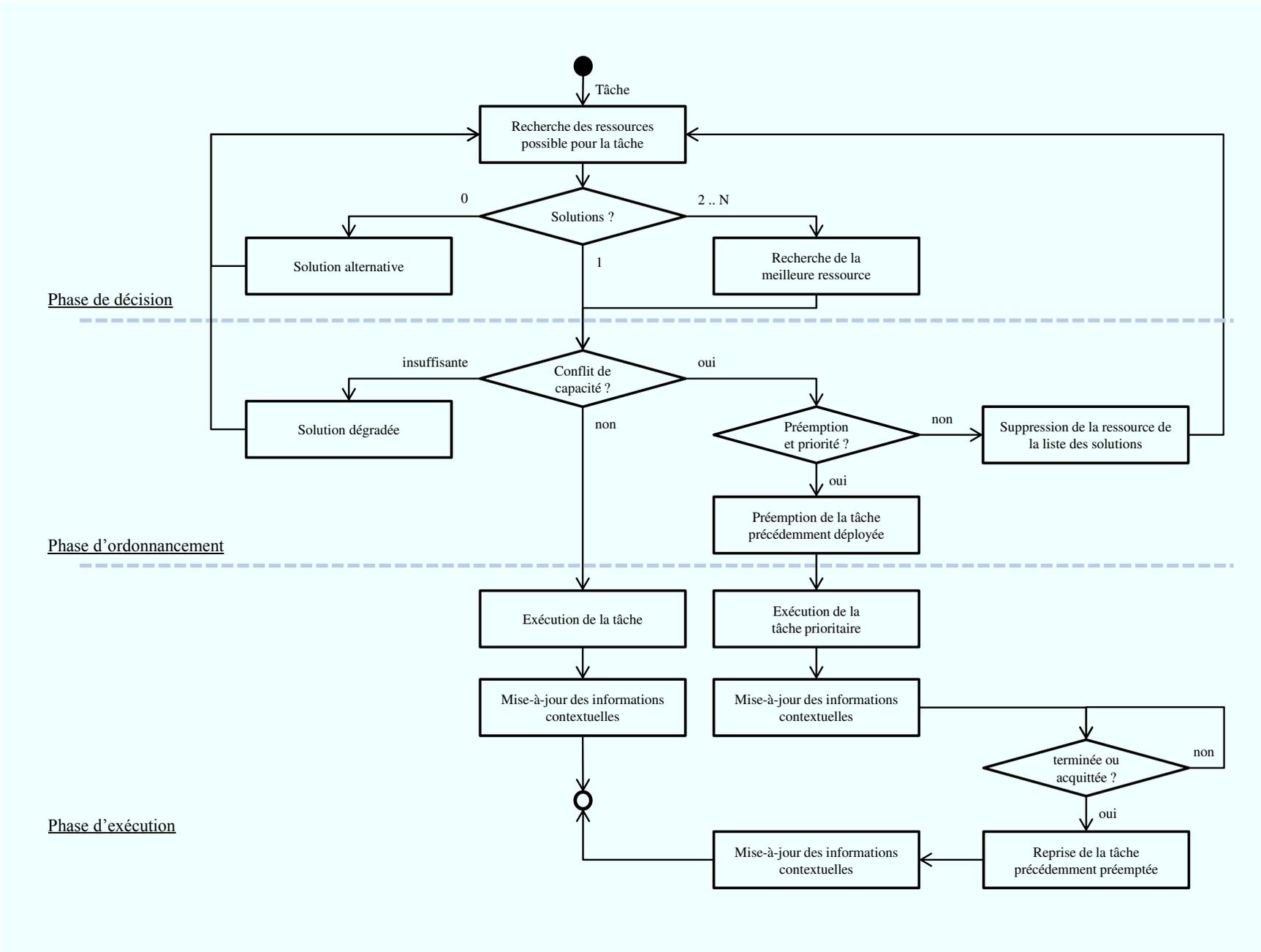


FIGURE 3.12 – Algorithme du gestionnaire de tâche

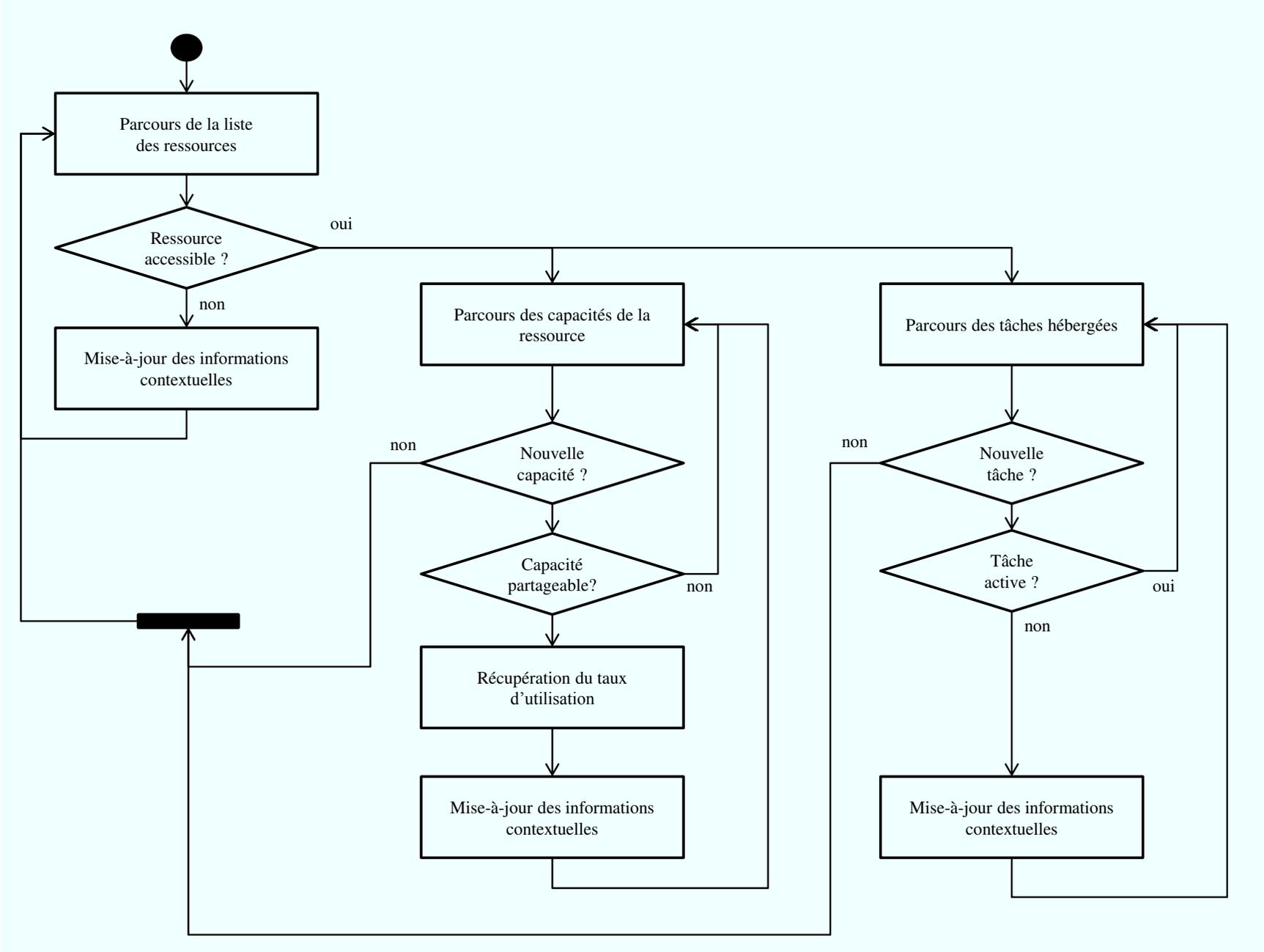


FIGURE 3.13 – Algorithme du moniteur de ressources

# Chapitre 4

## Preuve de concept et résultats

### Sommaire

---

<b>4.1</b>	<b>L'environnement d'expérimentation . . . . .</b>	<b>93</b>
4.1.1	Les appartements tremplins du CMRRF de Kerpape . . . . .	94
4.1.2	La maison Domus de l'Université de Sherbrooke . . . . .	95
4.1.3	La plateforme d'expérimentation du labSTICC de Lorient . . . . .	98
4.1.3.1	La plateforme d'expérimentation originale . . . . .	98
4.1.3.2	La plateforme d'expérimentation évoluée . . . . .	100
<b>4.2</b>	<b>Mise en œuvre du système d'assistance domotique . . . . .</b>	<b>102</b>
4.2.1	Cas d'étude d'un patient simulé dans un appartement tremplin . . . . .	102
4.2.1.1	Le profil du patient expérimental . . . . .	102
4.2.1.2	Scénario d'étude . . . . .	103
4.2.1.3	Équipements possédés et proposés . . . . .	103
4.2.1.4	L'offre personnalisée de services . . . . .	104
4.2.2	Conception : Modélisation de l'environnement . . . . .	104
4.2.2.1	Les éléments matériels . . . . .	105
4.2.2.2	Les éléments logiciels . . . . .	106
4.2.3	Exécution : Déploiement de tâches d'aide au handicap . . . . .	111
4.2.3.1	Initialisation du système d'assistance . . . . .	111
4.2.3.2	Fonctionnement courant du système . . . . .	113
<b>4.3</b>	<b>Résultats et limites de la solution proposée . . . . .</b>	<b>118</b>
4.3.1	Intégration des recommandations de l'aide au handicap . . . . .	118
4.3.2	Stratégies de déploiement . . . . .	119
4.3.3	Mesure des temps de réaction . . . . .	119
4.3.4	Limites du système de déploiement . . . . .	120

---



Pour des raisons économiques, mais aussi sociales et humaines, la prise en charge du handicap tend à prendre de l'importance dans les débats publics. Depuis des années, des recherches sont menées pour faciliter et favoriser l'intégration des personnes en situation de handicap dans la société. Par définition, nous avons vu que les situations de handicap résultaient des facteurs personnels et des facteurs environnementaux. Les facteurs environnementaux peuvent ainsi aggraver ou compenser une situation de handicap particulière. En d'autres termes, si l'environnement est adapté à la pathologie de la personne et compense son incapacité, la personne ne se trouve plus en situation de handicap.

Au cours de ces travaux de doctorat, nous avons pu étudier deux living lab<sup>1</sup> dont l'analyse des besoins et solutions de chaque environnement domotique nous a poussé à proposer une solution alliant les forces de chacun. Dans un premier temps, nous avons été invité au laboratoire Domus de l'Université de Sherbrooke. Cette visite de 7 mois a eu pour but d'étudier l'environnement technologique et les solutions mises en place, qu'ils proposent pour aider les personnes atteintes de troubles cognitifs. Le second living lab, est l'un des appartements tremplin au Centre Mutualiste de Rééducation et Réadaptation Formelle de Kerpape (CMRRF de Kerpape). L'équipe du laboratoire d'électronique de Kerpape, nous a autorisé à tester notre application de déploiement de service pendant une semaine au sein d'un appartement domotique.

Ce chapitre se consacre à l'étude d'un exemple concret afin d'illustrer les notions et concepts étudiés dans les précédents chapitres. Pour présenter les principes de modélisation, vus dans le chapitre 2, nous examinerons, dans la première partie, la plateforme d'expérimentation et ses origines. Dans la seconde partie, nous exposerons une preuve de concept et des scénarios spécifiques pour mettre en avant nos stratégies de déploiement. Enfin, nous discuterons des premiers résultats dans la dernière partie.

## 4.1 L'environnement d'expérimentation

Nous appelons environnement d'expérimentation la plateforme domotique de test que nous avons utilisée pour valider les concepts de notre système de déploiement de tâches d'aide au handicap. Suite aux projets qui se sont tenus au Laboratoire Lab-STICC de Lorient, une plateforme de contrôle d'environnement avait déjà été développée. En partant de cette base, nous l'avons adaptée et faite évoluer au cours des études de terrains effectuées tant à Kerpape qu'au Domus. Les évolutions menées ont été effectuées d'une part sur le matériel utilisé et d'autre part sur l'offre de services fournie par la plateforme.

Pour bien comprendre le cheminement de la construction de cette plateforme d'aide au handicap, nous présenterons rapidement les espaces domotiques mis en place à Kerpape et au Domus. Nous étudierons dans un premier temps les appartements tremplins domotisés du CRRMF de Kerpape qui ont mené à la plateforme de base. Puis dans un second temps, nous approfondirons l'environnement utilisé par laboratoire Domus qui reproduit une maison pour sénior et se consacre à l'analyse du comportement et des activités de la vie quotidienne. Enfin, nous passerons en revue notre plateforme d'expérimentation utilisée lors de nos travaux. Nous modéliserons certaines ressources et certaines tâches d'aide au handicap pour illustrer les concepts du chapitre 2.

---

1. Laboratoire d'étude et d'expérimentation

#### 4.1.1 Les appartements tremplins du CMRRF de Kerpape

Le CMRRF de Kerpape est un organisme de rééducation et de réadaptation fonctionnelles des patients atteints de déficience physique. Le centre de Kerpape a toujours été un grand promoteur de la technologie pour compenser le handicap et possède même son propre laboratoire d'électronique. Les soins, prodigués par le centre de Kerpape, de rééducation, de réadaptation et d'insertion sociale, professionnelle et/ou scolaire s'adressent aux adultes et enfants présentant des affections de l'appareil locomoteur, des affections neurologiques et neuromusculaires, des affections cardiovasculaires et respiratoires et aux grands brûlés. Plusieurs projets ont été menés en collaboration avec le lab-STICC dont le projet QuatrA [72] qui a ouvert la voie vers la domotique au sein des chambres médicalisées. Dernièrement, le CMRRF de Kerpape a fait construire et équiper en domotique deux appartements, nommés « appartements tremplins ». Ces logements intelligents ont pour vocation l'évaluation des patients du centre en habitat individuel, avant un retour dans leur domicile. Un second objectif est de démontrer l'expertise dans le domaine de l'aide au handicap et de sensibiliser les professionnels du bâtiment tant sur le plan de l'aménagement architectural du lieu de vie que des solutions domotiques. Ils favorisent la formation des artisans et apportent une certaine compréhension du monde du handicap et des systèmes d'assistance existants. Le troisième objectif des appartements tremplins est le support des travaux de recherche. En effet, ces espaces intelligents disposent de nombreux équipements domotiques et offrent un terrain d'expérimentation concret permettant de tester les travaux de recherche.

Les appartements tremplins regroupent un studio de 35m<sup>2</sup> et un appartement T2 de 75m<sup>2</sup>. Le plan des appartements tremplins est représenté sur la figure 4.1. L'entrée du hall, qui sépare les deux logements, est située sur la gauche du plan. La partie supérieure du plan décrit le studio composé d'un salon cuisine, d'une chambre et d'une salle de bain. Il est équipé d'un bus domotique propriétaire Legrand. L'appartement situé, sur la partie inférieure du plan, possède quatre pièces séparées : un grand salon cuisine, une chambre, une salle de bain et une pièce de rangement. Cet appartement est équipé différemment du studio puisqu'il bénéficie d'un bus domotique Knx/Eib ouvert.

Les deux appartements partagent le hall d'entrée et un local technique se trouvant derrière ce dernier. Dans le local technique, nous retrouvons un serveur avec un module IP pour la gestion du bus domotique Knx/Eib et un logiciel de supervision de l'appartement équipé en Knx. Des cartes d'acquisitions (audio et vidéo) sont aussi présentes pour gérer le portier du hall d'entrée.

Les équipements domotiques installés dans les appartements sont de type effecteur (capteurs et actionneur). La description de ces ressources, qui est schématisée sur la figure 4.1, est issue des travaux de Willy Allègre [69]. Parmi les équipements, on retrouve donc dans les appartements les actionneurs suivants (sigles carrés) :

- **L** : des actionneurs sur les lumières permettent les commandes de type Allumer/Eteindre et variateur.
- **V** : des volets électriques offrent les commandes de type Monter/Descendre.
- **P** : des portes équipées d'ouvre-porte électrique autorisent les commandes Ouvrir/Fermer.
- **L** : des lits électriques composés de deux sommiers individuels pilotés par infrarouge.

- **R** : des rails au plafond permettant de se déplacer entre le lit et la salle de bain commandés par infrarouge.
- **F** : des fenêtres motorisées contrôlées par infrarouge.
- **G** : des prises gigognes pilotées en infrarouge.
- **A** : un système d'appel activé par infrarouge.
- **H** : une hotte aspirante commandée en infrarouge.
- **M** : des équipements multimédias (télévisions et téléphones) en infrarouge.
- **P** : des plans de travail mobiles et éléments de cuisine amovibles, en infrarouge.
- **A** : un portier audio-vidéo connecté à la télévision.

Quelques capteurs sont aussi installés dans l'environnement domotique (sigles ronds) :

- **P** : des contacteurs pour indiquer l'état d'ouverture des portes.
- **A** : des détecteurs d'anomalies en cas d'incendie ou de problèmes d'alimentation électrique.
- **T** : des capteurs de température Knx.
- **L** : des capteurs de luminosité Knx.
- **D** : des détecteurs de présence associés aux capteurs de luminosité.

Nous pouvons remarquer que les capteurs sont moins nombreux que les actionneurs dans les deux appartements. Tous les éléments, ou presque de l'espace de vie, sont pilotés par les bus domotiques ou par infrarouge, tandis qu'on ne récupère que quelques informations comme la présence du patient, la luminosité des pièces, la températures, etc. En effet, les appartements ont été construits dans le but d'accueillir principalement présentant un handicap moteur, même si de nombreux patients atteints de troubles neurologique les occupent aussi. La priorité a été donnée au contrôle d'environnement même si rien n'empêche d'offrir des solutions d'analyse du comportement. Du côté des ressources proposées, on note la présence d'un serveur informatique, cependant, son but principal est de gérer les commandes pour activer les actionneurs et les traitements et de proposer des ihm et les protocoles de communication. Le besoin en ressource de calcul est donc largement limité par la nature des opérations à traiter et aucun algorithme d'analyse du comportement et des habitudes de vie n'est prévu dans ces logements, à court terme.

Pour résumer, ces appartements sont dédiés au contrôle de l'environnement. Ils proposent un ensemble de services lié aux besoins physiques de leurs patients et offrent la capacité de commander presque tous les équipements présents dans ces espaces de vie. Les besoins en actionneurs sont donc considérés comme une priorité, à l'inverse de capteurs qui sont présents pour assurer uniquement la sécurité du patient et du système. Intéressons-nous, maintenant, au laboratoire Domus de l'Université de Sherbrooke et de sa maison intelligente dont la priorité a été donné aux handicaps cognitifs.

#### 4.1.2 La maison Domus de l'Université de Sherbrooke

Le laboratoire Domus de l'Université de Sherbrooke au Québec est un leader dans l'analyse du comportement et des habitudes de vie quotidienne des personnes. Leur expertise a pour objectif de compenser les handicaps cognitifs en accompagnant les patients par

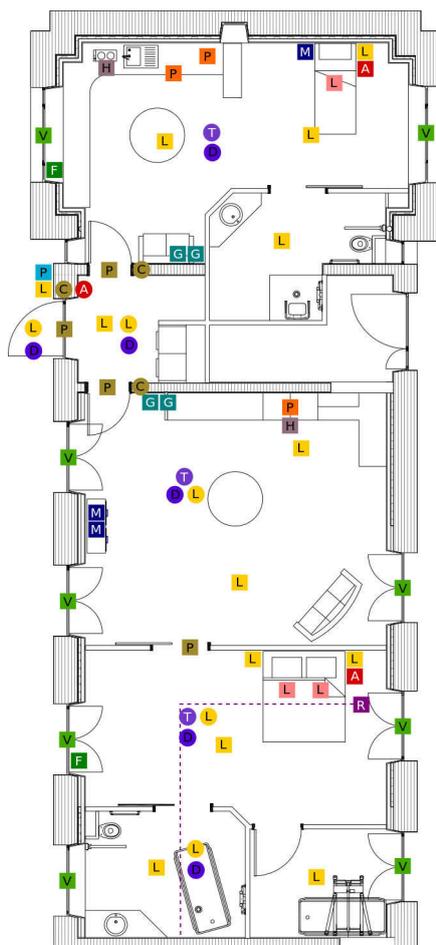


FIGURE 4.1 – Plan des appartements tremplins du CMRRF de Kerpape, issu de [69]

la technologie tout au long de la journée. Le public visé est différent de celui adressé au CMRRF de Kerpape puisque les patients ciblés ne présentent pas forcément d'incapacités physiques, mais plutôt des troubles cognitifs et intellectuels. Les personnes souffrant de la maladie d'Alzheimer sont donc un exemple des situations de handicap que souhaitent compenser, les membres du laboratoire Domus.

Le Domus appuie ses recherches sur une maison intelligente placée à l'intérieur même du laboratoire et de l'Université de Sherbrooke [112]. La maison intelligente Domus est un espace de vie complètement fonctionnel. Le plan de la maison Domus est schématisé sur la figure 4.2. La maison possède une chambre, une salle de bain, une cuisine, une salle à manger et un salon. Toutes ces pièces sont équipées de nombreux capteurs tels que :

- des détecteurs infrarouges sont présents dans les différentes pièces pour fournir l'information sur la présence, ou non, du patient dans une certaine zone. Ils peuvent couvrir un espace entier ou une partie en associations avec d'autres détecteurs infrarouges. Par exemple, un seul capteur couvre le salon et la salle à manger quand trois sont placés dans la seule cuisine.
- des détecteurs de pression sont installés dans la maison sous la forme de tapis. Nous pouvons en retrouver placés dans le hall principal à un endroit

stratégique pour détecter les déplacements entre la chambre et le salon, par exemple. Un second est installé sous le matelas du lit pour confirmer que le patient est bien couché sur son lit.

- des capteurs sur les interrupteurs de la lumière, qui envoient une information chaque fois qu'un occupant change l'état des lampes. Ces informations renseignent sur la présence de la personne puisqu'on peut déduire que si le patient éteint la lumière de la salle de bain, il quitte par conséquent les lieux.

- des contacteurs d'ouverture sont placés sur toutes les portes et les placards, mais aussi sur le réfrigérateur ou la porte du four. Un évènement est envoyé renseignant l'état d'ouverture des portes.

- des débitmètres, installés sur tous les robinets et la chasse d'eau des toilettes. Ils fournissent des statistiques sur l'utilisation des robinets d'eau et sont utilisés, associés à d'autres informations, pour déduire certaines activités. Si l'armoire à pharmacie a été ouverte et que le robinet du lavabo est utilisé, juste après, une déduction peut être émise sur l'activité « prise de médicaments » du patient, par exemple.

- des microphones et des capteurs électromagnétiques, utilisés dans les travaux de Thomas Camier [49], ont été installés pour permettre des détections d'activités basées sur des schémas sonores et sur l'analyse de la consommation électrique en temps réel.

Des actionneurs sont installés dans la maison bien qu'en nombre plus restreint. Par exemple, les lumières peuvent clignoter afin d'avertir le patient qu'une information n'a pas été prise en compte ou pour indiquer une zone de la maison en vue de lui faire effectuer une tâche dans cette zone. Nous pouvons aussi observer des diodes électroluminescentes pour indiquer de façon plus précise une zone par exemple un placard contenant l'ustensile désiré, etc. Des haut-parleurs sont disposés dans chaque pièce et peuvent émettre des messages vocaux et enfin des écrans et ordinateurs sont disposés dans la maison pour offrir des ihm ou alors afficher des informations visuelles.

Enfin, nous n'avons pas encore parlé de la salle des serveurs qui centralise et traite les informations issues de tous les capteurs. En effet, l'analyse du comportement par des algorithmes complexes est coûteuse en ressource de calcul et nécessite l'installation d'ordinateurs suffisamment puissants.

Pour résumer, à la vue des équipements présents dans l'environnement domotique de la maison intelligente Domus, les équipes de recherche travaillent plus particulièrement sur l'analyse du comportement des patients dans le but de les assister dans leurs activités quotidiennes. Il est intéressant d'étudier le nombre et la diversité des capteurs présents au Domus. La population visée est composée de personnes atteintes de troubles cognitifs. Peu de solutions ont été installées pour proposer le contrôle de la maison au patient, mais ce n'est pas l'objectif défini. Dans cet environnement, nous pouvons suivre les activités du patient et adapter des solutions en fonction de ses pathologies cognitives. Nous retrouvons, parmi d'autres, des tâches d'aide au handicap comme l'agenda interactif [11, 12] qui vérifie que le patient n'oublie pas les différents évènements inscrits dessus ou encore l'assistant à la cuisine [50], qui accompagne le patient dans la conception d'un repas en renseignant, validant et le conseillant sur les différentes étapes de la recette de cuisine.

L'étude de ces deux environnements nous a montré deux solutions distinctes permettant de compenser des handicaps pour différents types de population : les personnes souff-

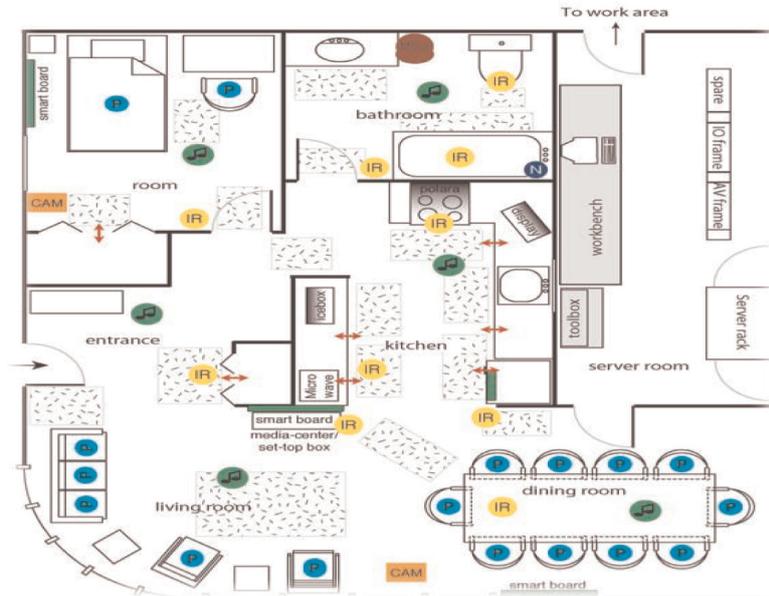


FIGURE 4.2 – Plan de la maison intelligente du Domus [112]

frant de handicaps physiques et les personnes atteintes de handicaps cognitifs. Toutefois, une population présente parfois un plurihandicap de type physique et cognitif, il peut s'agir des personnes âgées. Ces personnes peuvent éprouver des difficultés physiques ou souffrir des troubles cognitifs ou de la mémoire, en raison de la dégradation naturelle de leurs systèmes organiques. C'est donc motivé par cet objectif que nous avons décidé d'unifier les deux mondes et de former une plateforme d'expérimentation alliant le contrôle de l'environnement grâce aux actionneurs et autres bus domotiques ainsi que l'analyse comportementale en installant de nombreux capteurs et des ressources de calcul. Notre plateforme d'expérimentation est présentée dans la section suivante.

### 4.1.3 La plateforme d'expérimentation du labSTICC de Lorient

#### 4.1.3.1 La plateforme d'expérimentation originale

**Les équipements domotiques** Le Lab-STICC de Lorient, soutenu par l'ENSIBS a pour objectif de posséder son propre espace domotique. Pour cela, un appartement au sein de l'école est en préparation pour à terme effectuer des campagnes de mesure. Pour le moment, le Lab-STICC de Lorient possède une plateforme domotique héritée du projet QuatrA (collaboration entre le CMRRF de Kerpape et le Lab-STICC de Lorient). Nous nous sommes appuyés sur cette plateforme domotique uniquement dédiée au contrôle d'environnement. La photo de la figure 4.3 montre la plateforme dans sa première version. Sur cette figure, nous distinguons plusieurs éléments :

- Des lampes : Les lampes au premier plan, au nombre de deux, sont des ressources basiques, mais servent d'actionneur de manière générale. L'une est branchée sur le bus Knx/Eib, l'autre sur le bus LeGrand
- La télévision : La télévision est un autre actionneur cette fois pilotable en infrarouge



FIGURE 4.3 – Photo de la plateforme d’expérimentation, première version

- L’émetteur infrarouge : Il n’est pas bien visible à première vue, mais se trouve sur la tige de la lampe de droite. Il est branché sur le serveur domotique et permet de contrôler la télévision et le bus domotique LeGrand

- Une valise domotique LeGrand : Sur la gauche de la figure, la boîte blanche avec des prises et des interrupteurs cache en réalité un bus domotique propriétaire LeGrand. Nous pouvons retrouver un bus similaire dans le studio des appartements tremplins de Kerpape. Ce bus domotique étant propriétaire nous ne pouvons interagir avec qu’à l’aide de commandes infrarouges. Toutefois, nous pouvons commander chaque prise et chaque interrupteur de la valise

- Le tableau électrique Knx/Eib : Au centre de la photo et en arrière-plan, l’élément le plus imposant est un bus domotique ouvert Knx/Eib. Nous avons déjà évoqué cette technologie dans l’appartement tremplin de Kerpape. Ce tableau se décompose en deux parties, les prises de couleurs et leurs interrupteurs associés (occultés par la télévision) et la partie inférieure constituée des modules Eib dont la passerelle Ip/Eib et du bus domotique. Nous pouvons piloter le bus et toutes les ressources connectées dessus à l’aide d’un ordinateur relié à sa passerelle Ip/Eib

- Le serveur domotique : Sur le tableau Knx à gauche de la télévision nous pouvons noter la présence d’une carte électronique. Cette carte est une « Igep v2 », une carte basée sur une architecture ARM. L’application de contrôle d’aide au handicap est exécutée sur cette carte électronique, le patient utilisant son interface mobile connectée à cette ressource. Remarquons qu’à l’origine du projet QuatrA ce serveur domotique était un ordinateur de bureau compact dédié. Le passage de l’ordinateur dédié à une carte spécialisée faible encombrement et faible consommation, nous a permis d’améliorer la non-intrusivité de la solution et de diminuer les coûts liés à l’installation sans dégrader les performances de manière trop importante

**L'offre de services disponibles** Cette première version de la plateforme domotique était gérée par le middleware Danah [73], un ancien projet du Lab-STICC de Lorient que nous avons décrit dans le chapitre 1, notamment. Ce middleware implémente tous les protocoles de communications présents sur cette plateforme et bénéficie d'une interface ergonomique adaptée. Toutefois, en dehors du contrôle des ressources actionneurs le middleware Danah n'adressait pas d'autres ressources. L'offre de services proposée par cette solution était adaptée uniquement au contrôle de l'environnement et les tâches implémentées se limitaient au pilotage du bus Knx/Eib, du bus LeGrand et des équipements pilotés en infrarouge.

C'est pourquoi nous sommes partis sur une solution middleware plus simple et permettant le contrôle d'autres ressources singulières comme des ordinateurs ou des capteurs utiles aux applications d'analyse de comportement. L'offre de services a donc évolué et des tâches multimédias comme la consultation de pages internet ou la diffusion de film entre autres, ont été implémentées. Pour finaliser l'offre de service et réorienter la plateforme vers l'assistance au handicap, des applications d'analyse du comportement, de messages d'alerte, etc. ont aussi été ajoutées.

Notons, enfin, que des expériences ont aussi été menées pour intégrer à la plateforme des ressources atypiques comme une smartcam. Cet équipement, dont nous avons fait l'étude et la modélisation dans le chapitre 2, apporte des capacités très intéressantes à exploiter dans une même entité. L'association du capteur photographique à un processeur et un circuit programmable FPGA, permet, par exemple, de réaliser localement des opérations telles que des algorithmes de traitement d'image et donc de libérer des capacités de traitement pour les autres ressources de l'environnement. Maintenant, l'utilisation d'une caméra est en contradiction avec les spécifications de l'aide au handicap qui souhaitent mettre en œuvre une solution la moins intrusive possible.

Pour résumer, la plateforme de base offrait un contrôle domotique convenable qui permettait aux patients atteints de déficiences physiques de pouvoir évoluer dans leur environnement en contrôlant l'ensemble de leurs équipements. Bien que doté d'un ordinateur (serveur domotique), notre système domotique ne peut exécuter d'applications calculatoires comme celle concernant l'analyse des activités de la vie quotidienne. Ces algorithmes sont très consommateurs et les ressources disponibles limitées. Dans l'objectif, d'apporter le soutien nécessaire aux personnes en situation de handicap cognitif, nous avons souhaité faire évoluer la plateforme, en ajoutant des ressources et en raffinant leur description, pour unifier les offres de services de contrôle d'environnement et d'analyse comportementale.

#### 4.1.3.2 La plateforme d'expérimentation évoluée

**Les ressources domotiques** Nous venons de voir les limites de notre plateforme d'expérimentation dans sa version de base et les besoins d'unifier les offres de contrôle d'environnement et d'analyse du comportement, pour assister au quotidien les personnes souffrant de plurihandicap et les personnes âgées notamment. Le manque de ressources informatiques, pour exécuter des algorithmes calculatoires tels que la fouille de données, le traitement d'images ou d'événements temps réel, etc., ou encore le manque de capteurs pour créer de l'information est un obstacle à l'usage des tâches d'analyse comportementale. Pour toutes ces raisons, nous avons décidé de faire évoluer notre plateforme domotique. Dans sa dernière version, celle utilisée pour nos travaux, la plateforme d'expérimentation se compose des ressources schématisées sur la figure 4.4. On y retrouve :

- Des ressources informatiques : Un ordinateur de bureau avec ses périphériques d'entrées et de sorties, un ordinateur portable, et un serveur domotique sans interface homme-machine (pas de système d'affichage et pas de clavier ni de souris). En effet, ce serveur domotique est une ressource utilisable uniquement pour ses capacités de traitements et de commande puisqu'elle possède des modules radiofréquences, Z-Wave et Enocean, mais aucune interface d'entrée (clavier, souris, etc.) ni de sortie (écran, haut-parleurs, etc.). Notons que l'ordinateur de bureau est lui aussi doté d'un émetteur/récepteur infrarouge.
- Des ressources passerelles : Un routeur Ethernet qui nous permet de mettre plusieurs ressources à proximité, un routeur adsl nous offrant une connexion à Internet et une passerelle Ip/Knx qui fait la liaison entre le bus domotique Eib et le reste du réseau.
- Des périphériques (capteurs, actionneurs et commandes) divers et variés : Parmi les actionneurs notons, les interfaces écrans et haut-parleurs connectés à l'ordinateur de bureau, la télévision ou les lampes pilotées par infrarouge, ou le bus Knx/Eib. Nous retrouvons des interrupteurs que nous considérons comme des capteurs d'évènements, des capteurs température (Knx/Eib, Enocean), etc. Enfin, nous pouvons observer les périphériques de commande comme les modules radiofréquences, Z-Wave et Enocean.

**L'offre de tâches possibles** La proximité et les collaborations mises en place depuis des années avec le CMRRF de Kerpape ont favorisé le développement d'une plateforme d'expérimentation orientée contrôle d'environnement. L'ajout dans notre environnement domotique des ressources de traitements ainsi que des capteurs offre l'exécution des tâches d'analyse du comportement, comme le propose le Domus, en plus des tâches de contrôle d'environnement. Nous pouvons citer plusieurs services que nous déployons sur notre plateforme. La liste suivante n'est pas exhaustive et représente l'offre que nous mettons en place pour un but expérimental :

- Les tâches de contrôle d'environnement : Allumer et éteindre les différentes lampes disponibles, le contrôle de la télévision (allumer, éteindre, changer la chaîne, le volume, etc.), la diffusion d'un film à partir d'une bibliothèque numérique, l'écoute d'une radio à partir d'un flux internet, la consultation d'une page internet comme la météo ou les informations quotidiennes.
- Les tâches d'analyse comportementale : La reconnaissance d'activité et management d'alerte proposés par [30], une alerte hydratation qui détecte une forte température et affiche un message visuel, une alerte prise de médicament qui est un message périodique émis sous deux formes soit visuelle, soit auditive.
- Les tâches systèmes : Les tâches systèmes comprennent le flot de déploiement (manager d'ontologie, gestionnaire de tâche et superviseur de ressource), mais aussi l'interface graphique utilisateur.

Remarquons que les lampes possèdent toutes une activation manuelle. La lampe « Light Eib1 » est activable grâce à l'interrupteur « Interruptor Eib1 », la lampe « Light Eib2 » est activable grâce à l'interrupteur « Interruptor Eib2 » et la lampe « Light Zwave » est activable grâce à l'interrupteur « Interruptor Enocean ».

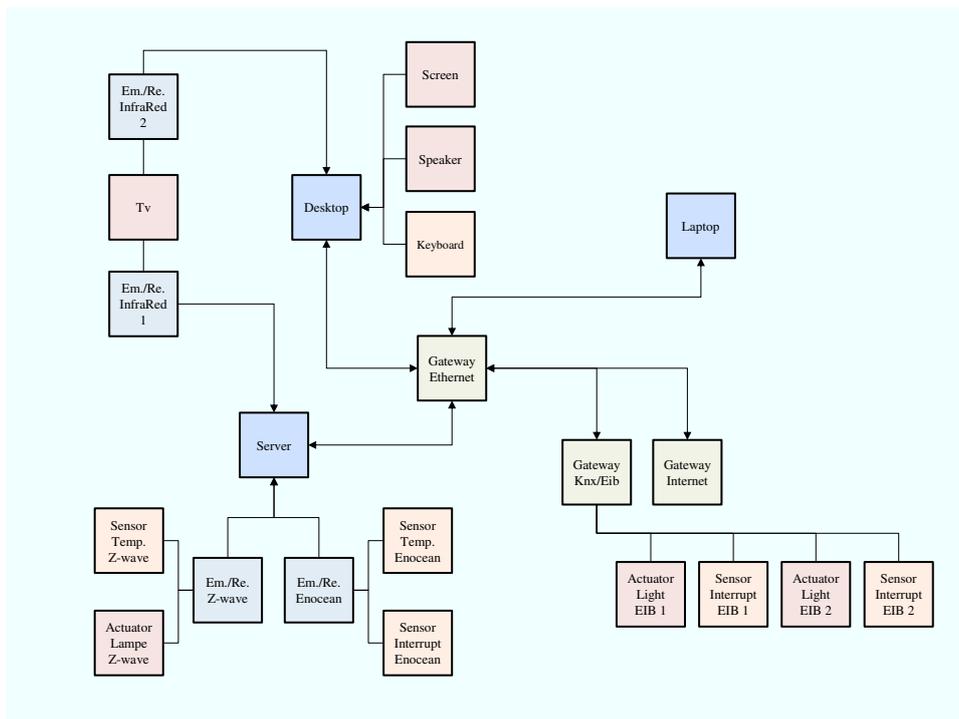


FIGURE 4.4 – Schéma topologique de la plateforme d'expérimentation actuelle

## 4.2 Mise en œuvre du système d'assistance domotique

Dans cette partie, nous souhaitons mettre en œuvre les étapes qui constituent notre système d'assistance domotique. Pour ce faire, nous simulons un patient et son environnement. Le patient et son profil sont inventés de toute pièce tandis que son environnement est calqué sur l'un des appartements tremplins de Kerpape, où nous avons mené nos expérimentations. Dans un premier temps, nous nous consacrerons à la description de la phase de conception puis nous simulerons son comportement vis-à-vis de l'utilisation du système d'assistance et du déploiement de service.

### 4.2.1 Cas d'étude d'un patient simulé dans un appartement tremplin

#### 4.2.1.1 Le profil du patient expérimental

Le patient est une personne âgée dont le profil indique des déficiences physiques et cognitives légères. Le patient présente toujours l'usage de ses membres supérieurs et inférieurs, mais ses déficiences physiques se traduisent par une fatigue plus prononcée et plus rapide. En d'autres termes, le patient ne possède pas de handicap physique, mais des difficultés à réaliser des tâches de la vie quotidienne. Cette condition pourrait cependant nécessiter une hospitalisation dans un EHPAD. Les troubles cognitifs légers du patient représentent un état cognitif entre la perte naturelle de facultés liée au vieillissement et les altérations fortes des maladies neurodégénératives. Les personnes présentant des troubles cognitifs légers ont davantage de risque de développer une démence. On parle parfois de maladie d'Alzheimer débutante pour qualifier les troubles cognitifs légers. En outre, le profil du patient indique qu'il possède un attrait pour les nouvelles technologies et présente une aisance dans leur utilisation. Enfin, même si sa condition liée à son âge demande

une attention particulière, une surveillance approfondie de ses constantes vitales n'est pas obligatoire.

#### 4.2.1.2 Scénario d'étude

Le scénario étudié est une période allant de 22 heures à 10 heures le lendemain. Le scénario regroupe donc la nuit du patient et une partie de sa matinée, où des habitudes fortes se mélangent à des événements plus ponctuels. Son emploi du temps peut se décrire comme suivant :

- 22h00 Se coucher
- 08h00 Se réveiller
- 08h30 Faire sa toilette
- 09h00 Prendre le petit déjeuner
- 09h10 Prendre ses médicaments
- 09h30 Arrivée de son auxiliaire de vie
- 10h00 Prendre le bus pour son rendez-vous

#### 4.2.1.3 Équipements possédés et proposés

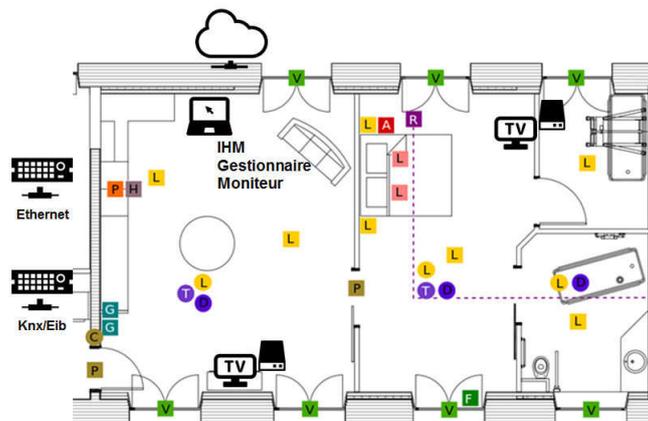


FIGURE 4.5 – Plan du domicile équipé du patient

Le profil du patient indique qu'il est familier avec la technologie. En étudiant son environnement, figure 4.5, on remarque que son domicile est équipé d'un bus domotique (EIB) sur lequel sont connectés les différentes lampes, les volets ou encore les portes des différentes pièces de son appartement. Le patient possède deux télévisions connectées à des serveurs domotiques faible encombrement et faible consommation (IgepV2) : une dans la chambre et l'autre dans le salon. Nous retrouvons un ordinateur portable dans la cuisine sur lequel sont exécutés les applications d'interface, du gestionnaire de tâches et du moniteur de ressources. Enfin, nous pouvons noter deux passerelles (Ethernet et Eib) qui permettent de faire le lien entre toutes les ressources présentes.

#### 4.2.1.4 L'offre personnalisée de services

Le profil du patient indique que sa mobilité lui permet de se déplacer dans tout son espace de vie et ses capacités physiques lui permettent d'interagir manuellement avec ses équipements. Par exemple, le patient peut allumer la lumière principale de son logement en appuyant sur l'interrupteur associé. Toutefois, en raison de son âge, il possède une sensibilité à la fatigue physique importante et la répétition de nombreuses actions variées impacte son état et dégrade sa condition physique et mentale rapidement. C'est pourquoi, il a été recommandé de faciliter le plus possible les interactions avec les équipements présents dans son environnement. Par exemple, le patient dispose d'une télévision dont le contrôle fait partie des services qui lui sont proposés. De même, le contrôle des différents équipements électriques requiert une prise en compte par le système : le patient possède plusieurs lampes, volets et portes dont le contrôle doit aussi être proposé dans son offre de service. Enfin, pour simplifier certaines actions complexes, toujours dans le but de diminuer la charge physique et cognitive du patient, des combinaisons de services sont offertes au patient sous la forme de scénarii, comme le scénario « sortir de chez soi » qui associe plusieurs services élémentaires dont l'ouverture de la porte d'entrée, l'extinction de toutes les lumières et après une temporisation la fermeture de la porte d'entrée. L'offre de service propose aussi des fonctions à tendance multimédia, qui diminuent aussi la charge de ce dernier. Ainsi, le patient peut consulter des pages internet, écouter une webradio ou encore demander la lecture d'un film en quelques actions à partir de son interface utilisateur graphique.

Outre, le contrôle de l'environnement le patient requiert une attention particulière puisque son état autant physique que cognitif est susceptible de se dégrader dans le temps. Aussi, il est important de s'intéresser à son comportement et notamment à ses activités quotidiennes. Analyser le comportement d'un sujet fragile permet d'anticiper ou au moins accélérer les actions à mettre en place. Par exemple, le patient peut se voir rappeler des événements comme la prise régulière de ses médicaments ou des rendez-vous programmés dans son agenda. On peut aussi détecter une anomalie dans ses habitudes de vie comme un malaise et ainsi déclencher une alerte rapidement. Aussi, lorsqu'on vieillit, la quantité d'eau disponible dans le corps pour compenser une déshydratation temporaire diminue et expose au risque de déshydratation grave. Ce risque est augmenté par d'autres facteurs liés à l'âge, dont la diminution de la sensation de soif. De fait, une surveillance accrue des conditions de température dans son environnement est indispensable et l'émission d'une recommandation sur le besoin de s'hydrater est requise.

Nous pouvons résumer une partie de son offre de service comme tel :

- Contrôle : Appareillage domotique, applications multimédias
- Analyse : Calendrier interactif [11], supervision non-intrusive [25], identification de services [75], messages d'alertes
- Application système : Interface Homme-Machine, gestionnaire de tâches, moniteur de ressources

#### 4.2.2 Conception : Modélisation de l'environnement

L'application de déploiement de services a été testé dans l'un des appartements du centre de Kerpape pendant une semaine. Pour concevoir le système d'assistance, et plus particulièrement construire l'ontologie qui sera l'image globale des éléments composant le

système, nous avons reproduit l’environnement sous forme d’un diagramme topologique, sur la figure 4.6. Cette représentation nous permet de clarifier les interactions entre les nombreux équipements présents.

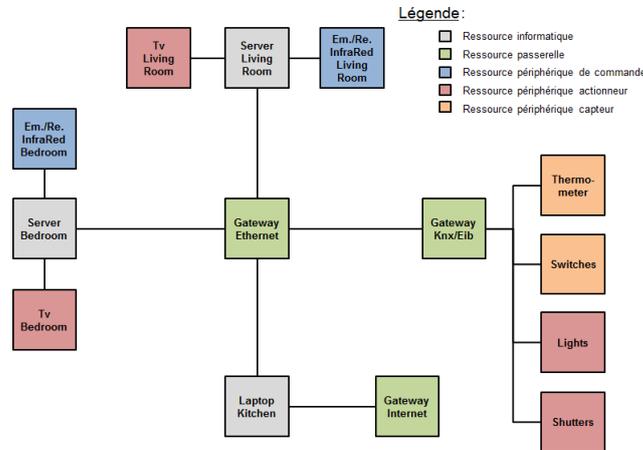


FIGURE 4.6 – Vue topologique partielle du domicile équipé du patient

L’application de déploiement de services a été testée dans l’un des appartements du centre de Kerpape pendant une semaine. Pour concevoir le système d’assistance, et plus particulièrement construire l’ontologie qui sera l’image globale des éléments composant le système, nous avons reproduit l’environnement sous forme d’un diagramme topologique, sur la figure 4.6. Cette représentation nous permet de clarifier les interactions entre les nombreux équipements présents. Après cette étape de simplification symbolique vient la modélisation ontologique de l’environnement. La représentation des différents éléments matériels et logiciels est modélisée sous la forme d’une ontologie pendant la phase de conception. La description d’une partie des éléments modélisés est proposée dans les sections suivantes.

Reprenons, donc, l’environnement décrit et schématisé sur les figures 4.5 et 4.6. Regardons comment sont décrites et modélisées certaines des ressources qui composent l’environnement domotique et les services proposés dans l’offre personnalisée.

#### 4.2.2.1 Les éléments matériels

**Remarque :** Sur les figures suivantes, décrivant la modélisation des ressources, les informations sémantiques sont écrites en italique et les informations sémantiques inférées sont écrites en gras et italique.

Parmi toutes les ressources de l’environnement domotique, nous en avons sélectionné quelques-unes pour approfondir leur description. La figure 4.7 représente deux des trois ressources informatiques présentes. Sur cette figure, la première modélisation que nous observons est celle de la ressource « *\_res\_Server\_LivingRoom* ». Ce serveur domotique, qui est une carte électronique Igep-v2, dont les capacités intrinsèques sont : le calcul, la mémoire et le stockage. Nous observons que cette ressource a hérité d’autres capacités liées aux connexions avec les autres équipements comme les capacités de commande (Eib, Internet, Infrarouge) ou encore des capacités d’actionneurs (audio et vidéo). D’un point de vue logiciel, le serveur domotique possède une machine virtuelle Java, le programme Ssh

et les applications relatives au fonctionnement de ses périphériques. Il possède aussi les sources de certaines applications. Pour terminer la modélisation, le serveur domotique exprime les différentes connexions qu'il possède avec les autres ressources de l'environnement.

La seconde ressource modélisée, sur la figure 4.7, est l'ordinateur portable disposé dans la cuisine. Nous pouvons observer la plupart de ses propriétés, mais toutes n'ont pas pu être représentées, notamment son catalogue de logiciels.

_res_Server_LivingRoom		
<b>Nom :</b>	_res_Server_LivingRoom	<b>User :</b> jdoe
<b>Neud :</b>	192.168.1.5	<b>Status :</b> aucun // string
<b>Capacités matérielles</b>	_capacity_Process _capacity_Memory _capacity_Storage _capacity_command_Internet	_capacity_command_Eib _capacity_command_Infrared _capacity_actuator_Video _capacity_actuator_Audio
<b>Bibliothèque de logiciel</b>	hasApplication(JRE) hasApplication(Ssh) hasApplication(EIbI) hasApplication(LIRC) hasApplication(OpenRemote)	hasSource(Gestionnaire) hasSource(BigBackBunny) hasSource(Moniteur) hasSource(AlerteMngt) hasSource(Manager) hasSource(AlerteHydr) hasSource(Interface)
<b>Liens individus</b>	isConnectedTo(_res_peripheral_command_Infrared) hasConnectionWith(_res_gateway_Ethernet)	hasConnectionWith(_res_gateway_Internet) hasConnectionWith(_res_gateway_Eib) hasConnectionWith(_res_Laptop_Kitchen) hasConnectionWith(_res_Television)
_res_Laptop_Kitchen		
<b>Nom :</b>	_res_Laptop_Kitchen	<b>User :</b> sequin
<b>Neud :</b>	192.168.1.7	<b>Status :</b> aucun // string
<b>Capacités matérielles</b>	_capacity_Process _capacity_Memory _capacity_Storage _capacity_sensor_Mouse _capacity_sensor_Keyboard	_capacity_actuator_Video _capacity_actuator_Audio _capacity_command_Internet _capacity_command_Eib
<b>Bibliothèque de logiciel</b>	hasApplication(JRE) hasApplication(Chrome) hasApplication(Vlc) hasApplication(EIbI) hasApplication(LIRC)	hasSource(Gestionnaire) hasSource(BigBackBunny) hasSource(Moniteur) hasSource(AlerteMedoc) hasSource(Manager) hasSource(AlerteMngt) hasSource(Interface) hasSource(AlerteHydr)
<b>Liens individus</b>	hasConnectionWith(_res_gateway_Ethernet) hasConnectionWith(_res_gateway_Internet) hasConnectionWith(_res_gateway_Eib) hasConnectionWith(_res_Server_LivingRoom) hasConnectionWith(_res_Server_BedRoom)	

FIGURE 4.7 – Description de ressources informatiques

Sur la figure 4.8, nous avons représenté deux passerelles. La première ressource, sur la partie supérieure, est la passerelle Eib qui fait le lien entre le réseau Ethernet et le bus Eib. La passerelle Internet est une box adsl qui permet de faire le lien entre le réseau Internet et le réseau Ethernet. Les passerelles sont des équipements bien plus limités que les ressources informatiques. Toutefois, elles possèdent des capacités intéressantes et leurs connexions avec l'ensemble des ressources du réseau leur permettent de partager facilement leurs capacités.

#### 4.2.2.2 Les éléments logiciels

Dans cette partie nous nous consacrons à l'étude de la modélisation de quelques tâches formant l'offre des services. Nous avons sélectionné un ensemble de tâches hétérogènes pour mettre en avant certains paramètres que nous allons expliquer ici. Notons que nous avons défini 2 niveaux de priorité pour faciliter le déploiement et la préemption des tâches. Le niveau le plus fort (2) est réservé aux alertes et aux tâches composant le flot de déploiement. La priorité (1), la plus faible, est attribuée à toutes les autres tâches. Ces priorités, comme d'autres paramètres, définies manuellement ici, sont issues des discussions avec le corps médical en connaissance du profil de production du handicap du patient.

*Les tâches de contrôle d'environnement* Sur la partie supérieure de la figure 4.9, nous pouvons noter la description d'une tâche d'aide au handicap allumer la lumière. Cette

_res_gateway_Eib	
Nom :	_res_gateway_Eib
Neud :	192.168.1.50
User :	aucun
Status :	aucun // string
Capacités matérielles	_capacity_command_Eib
Liens individus	<i>hasConnectionWith(_res_gateway_Ethernet)</i> <i>hasConnectionWith(_res_Server_BedRoom)</i> <i>hasConnectionWith(_res_gateway_Internet)</i> <i>hasConnectionWith(_res_Laptop_Kitchen)</i> <i>hasConnectionWith(_res_Server_LivingRoom)</i>

_res_gateway_Internet	
Nom :	_res_gateway_Internet
Neud :	192.168.1.2
User :	aucun
Status :	aucun // string
Capacités matérielles	_capacity_command_Internet
Liens individus	<i>hasConnectionWith(_res_gateway_Ethernet)</i> <i>hasConnectionWith(_res_Server_BedRoom)</i> <i>hasConnectionWith(_res_gateway_Eib)</i> <i>hasConnectionWith(_res_Laptop_Kitchen)</i> <i>hasConnectionWith(_res_Server_LivingRoom)</i>

FIGURE 4.8 – Description des ressources passerelles

dernière, nommée « `_task_LampeEib1_On` » correspond à l'éclairage de l'une des deux lampes connectées sur le bus eib. En effet, sa commande d'activation correspond à une requête sur le bus Eib qui demande le passage à 1 de la prise sur le port « 6/0/11 ». Cette tâche possède une version dégradée, qui est l'allumage de la seconde lampe sur le bus Eib, légèrement moins puissante et une version alternative qui est l'allumage de la télévision. Cette alternative peut sembler n'avoir aucun sens pourtant, dans certains cas, c'est la différence entre un minimum de luminosité et une luminosité nulle. Cette commande n'est pas préemptable puisqu'elle est instantanée (une fois exécutée elle libère toutes les capacités utilisées) et sa priorité est normale (de niveau 1). Nous pouvons aussi noter qu'elle ne possède aucun des paramètres suivants (PID, ActiveOn, Status) puisque ces paramètres lui sont donnés pendant son déploiement et durant l'intégralité de son cycle de vie. Le « PID » correspond à son identifiant donné par le système d'exploitation sur la machine, le paramètre « ActiveOn » correspond justement au nom de la machine qui l'exécute et son « Status » est son état actuel considéré par le système d'exploitation (Ready, Running, Zombie, etc.). Enfin, nous pouvons observer que pour son fonctionnement cette tâche requiert les capacités matérielles de calcul et de traitement, mais aussi de commande du bus Eib. La simple capacité matérielle de commande Eib ne suffit pas puisque la tâche nécessite aussi l'application pour contrôler cet élément (EIBd).

La partie inférieure de la figure 4.9 reprend la modélisation de la tâche d'aide au handicap « `_task_Informations_LeMonde` ». Ce service ouvre une fenêtre du navigateur web (Chrome) sur la page d'accueil du site du journal Le Monde, le tout projeté sur un actionneur visuel en plein écran. Cette tâche a donc besoin des capacités de traitement (calcul et mémoire), de la capacité d'accès à internet et celle de pouvoir projeter une image sur un écran. Du côté des applications, cette tâche ne requiert qu'un navigateur web pour être opérationnelle. Notons qu'il existe une alternative à cette tâche qui est le lancement d'une webradio « FranceInfo ».

La figure 4.10 nous présente la modélisation de tâches multimédia. Cette catégorie de

<b>_task_LampeEib1_On</b>		
<b>Nom :</b>	<i>_task_LampeEib_On</i>	<b>Préemption :</b> 0 // boolean
<b>Commande :</b>	\$ groupswrite ip:127.0.0.1 6/0/1 1	<b>Priorité :</b> 1 // integer
<b>Dégradée :</b>	<i>_task_LampeEib2_On</i>	<b>ProcessID :</b> aucun // string
<b>Alternative :</b>	<i>_task_Television_On</i>	<b>ActiveOn :</b> aucun // string
		<b>Status :</b> aucun // string
<b>Besoins matériels</b>	<i>_capacity_Process</i> <i>_capacity_Memory</i>	<i>_capacity_command_eib</i>
<b>Dépendances logicielles</b>	<i>needApplication(EiBa)</i>	

<b>_task_Informations_LeMonde</b>		
<b>Nom :</b>	<i>_task_Informations_LeMonde</i>	<b>Préemption :</b> 1 // boolean
<b>Commande :</b>	\$ chromium --incognito --start-fullscreen http://lemonde.fr/	<b>Priorité :</b> 1 // integer
<b>Dégradée :</b>	<i>aucun</i>	<b>ProcessID :</b> aucun // string
<b>Alternative :</b>	<i>_task_WebRadio_FranceInfo</i>	<b>ActiveOn :</b> aucun // string
		<b>Status :</b> aucun // string
<b>Besoins matériels</b>	<i>_capacity_Process</i> <i>_capacity_Memory</i>	<i>_capacity_actuator_Video</i> <i>_capacity_command_Internet</i>
<b>Dépendances logicielles</b>	<i>needApplication(Chrome)</i>	

FIGURE 4.9 – Description de tâches de contrôle d’environnement

<b>_task_Film_BigBuckBunny_HD</b>		
<b>Nom :</b>	<i>_task_Film_BigBuckBunny_HD</i>	<b>Préemption :</b> 1 // boolean
<b>Commande :</b>	\$ vlc -f --video-on-top /home/seguin/Videos/big_buck_bunny_720p_surround.avi	<b>Priorité :</b> 1 // integer
<b>Dégradée :</b>	<i>_task_Film_BigBuckBunny_SD</i>	<b>ProcessID :</b> aucun // string
<b>Alternative :</b>	<i>_task_Film_BigBuckBunny_YT</i>	<b>ActiveOn :</b> aucun // string
		<b>Status :</b> aucun // string
<b>Besoins matériels</b>	<i>_capacity_Process</i> <i>_capacity_Memory</i>	<i>_capacity_actuator_video</i> <i>_capacity_actuator_audio</i>
<b>Dépendances logicielles</b>	<i>needSource(big_buck_bunny_720p_surround.avi)</i> <i>needApplication(Vlc)</i>	

<b>_task_Film_BigBuckBunny_YT</b>		
<b>Nom :</b>	<i>_task_Film_BigBuckBunny_YT</i>	<b>Préemption :</b> 1 // boolean
<b>Commande :</b>	\$ chromium --incognito --start-fullscreen https://www.youtube.com/watch?v=YE7VqLp-4	<b>Priorité :</b> 1 // integer
<b>Dégradée :</b>	<i>aucune</i>	<b>ProcessID :</b> aucun // string
<b>Alternative :</b>	<i>aucune</i>	<b>ActiveOn :</b> aucun // string
		<b>Status :</b> aucun // string
<b>Besoins matériels</b>	<i>_capacity_Process</i> <i>_capacity_Memory</i>	<i>_capacity_actuator_video</i> <i>_capacity_actuator_audio</i> <i>_capacity_command_internet</i>
<b>Dépendances logicielles</b>	<i>needApplication(Chrome)</i>	

FIGURE 4.10 – Description d’une tâche multimédia

services est considérée comme un sous ensemble des tâches de contrôle d’environnement. En effet, ces services orientés utilisateurs offrent le contrôle d’appareils comme un ordinateur. C’est pourquoi nous ne faisons pas la différence entre ces deux types de tâche. Les applications modélisées, sur la figure 4.10, sont deux versions du service « Lecture du film Big Buck Bunny ». Ce service possède 3 formes :

- La forme principale est représentée sur la partie supérieure. La tâche correspond à la lecture d’un fichier vidéo en haute définition. Cette opération est réalisée en décodant un fichier vidéo source avec l’application VLC. Pour se

faire, elle requiert des capacités de traitements (calcul et mémoire) et aussi des dispositifs matériels pour projeter les images et le son. Pour satisfaire toutes les conditions, la ressource devra posséder dans sa bibliothèque logicielle non seulement l'application vlc, mais aussi le fichier vidéo.

- La forme dégradée de ce service n'est pas représentée. Expliquons, cependant, qu'il s'agit d'une tâche identique, mais dont le fichier source requis est une vidéo en basse définition. Cette forme dégradée ne change en rien les capacités nécessaires, mais diminue considérablement les besoins en traitement pour le décodage du fichier vidéo.

- La forme alternative de ce service est la lecture du même film, mais sur la plateforme de vidéo sur internet YouTube. En effet, si l'une des conditions ne peut être satisfaite, le déploiement de la version alternative peut être une solution envisageable. Notons que la tâche alternative ne requiert plus le fichier source, mais une capacité matérielle supplémentaire l'accès à internet pour déployer le service.

*Les tâches d'analyse comportementale* Sur la figure 4.11, nous pouvons observer le modèle de la tâche « Alerte hydratation ». Le comportement de ce service peut se décrire comme suit : lorsque le seuil de température défini est dépassé, un message d'alerte est émis pour prévenir le patient de penser à s'hydrater. L'alerte hydratation possède la particularité d'être un service composé de deux tâches : la surveillance de la température et l'affichage d'un message d'alerte. Notons que chaque tâche est aussi un service, mais qu'ils ne sont pas activables individuellement, c'est-à-dire en dehors du contexte de l'application « Alerte hydratation ». La première tâche est donc une application de surveillance de la température. Informatiquement, cela se traduit par un programme qui vient périodiquement récupérer la valeur d'un des capteurs de température. D'un point de vue de la modélisation, cette tâche requiert des capacités de traitement (process, memory), mais aussi de capteur de température (sensor temperature). Le message hydratation, la seconde tâche qui compose le service alerte hydratation, affiche un message visuel sur un écran. On note dans la modélisation que cette dernière tâche requiert en plus des capacités de traitement (process, memory), la capacité d'actionneur vidéo (actuator video). Techniquement, cette tâche est activée par la tâche surveillance de température. En effet, lorsque le seuil est dépassé c'est la surveillance de température qui fait appel au gestionnaire de tâche pour déployer la seconde partie du service d'alerte hydratation.

Sur la figure 4.12, nous présentons la modélisation schématisée d'un autre service d'analyse comportementale : « l'alerte prise de médicament ». Ce service existe en deux versions, sous la forme de bundles OSGi. À l'instar du service d'alerte-hydratation précédemment présenté, l'alerte-médicament est constituée de deux composants : le composant « gestion périodique » et le composant « émission de l'alerte ». Dans le premier composant, l'heure de la prise des médicaments est définie. L'horaire du système informatique qui exécute la tâche est alors scruté et un message d'alerte est émis au moment programmé. Le second composant possède deux versions : la première est visuelle et affiche un message sur un écran, la seconde est sonore et diffuse l'information au travers d'un haut-parleur. C'est à partir des mécanismes qu'offre OSGi que nous proposons ce service sous deux formes, une visuelle et une auditive.

Enfin, nous aurions pu modéliser la tâche d'aide au handicap qui se consacre à l'analyse des activités du patient pour émettre en cas de comportement anormal des alertes. Cette application de fouille de donnée scrute un fichier regroupant les différentes commandes

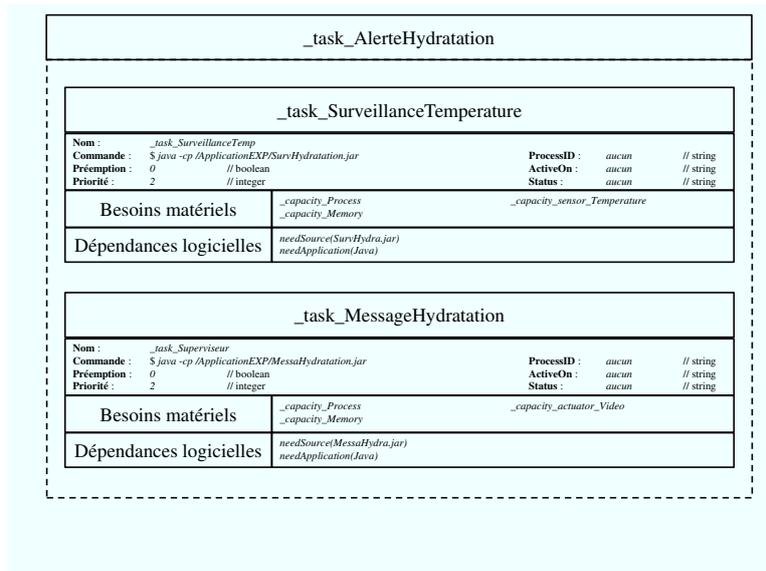


FIGURE 4.11 – Description de tâches d’analyse comportementale

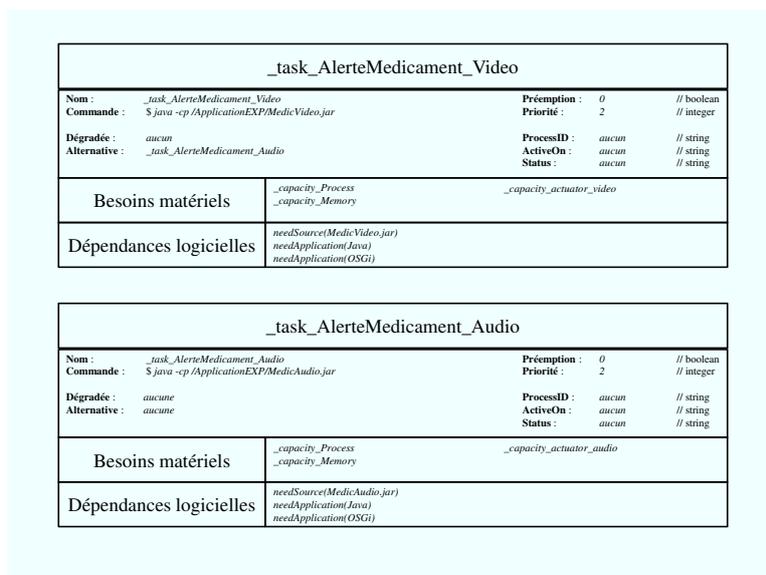


FIGURE 4.12 – Description d’une tâche OSGi

requis par le patient pour isoler des schémas et habitudes de vie. D’un point de vue matériel, elle requiert de fortes capacités de traitement, mais aussi de stockage. Une remarque sur ses propriétés de préemption et de priorité : cette tâche est préemptable et sa priorité est normale. De fait, elle peut être suspendue pour libérer les capacités utilisées et permettre à une alerte de plus haute importance de s’exécuter.

*Les tâches systèmes* Pour finir, nous proposons une modélisation de deux tâches systèmes : l’interface utilisateur graphique et le superviseur dont les modèles sont décrits sur la figure 4.13. L’application « interface graphique » est exécutée pendant l’initialisation, après que le flot de déploiement soit mis en place. La ressource qui accueillera l’interface

_task_Interface_utilisateur			
<b>Nom :</b>	<code>_task_Alerte_medicament_visuelle</code>	<b>Préemption :</b>	<code>0</code> // boolean
<b>Commande :</b>	<code>\$ java -cp /ApplicationEXP/Interface.jar labstic.interface.Interface</code>	<b>Priorité :</b>	<code>1</code> // integer
<b>Dégradée :</b>	<code>aucune</code>	<b>ProcessID :</b>	<code>aucun</code> // string
<b>Alternative :</b>	<code>aucune</code>	<b>ActiveOn :</b>	<code>aucun</code> // string
		<b>Status :</b>	<code>aucun</code> // string
<b>Besoins matériels</b>	<code>_capacity_Process</code> <code>_capacity_Memory</code>	<code>_capacity_actuator_video</code> <code>_capacity_sensor_mouse</code>	
<b>Dépendances logicielles</b>	<code>needSource(Interface.jar)</code> <code>needApplication(IRE)</code>		

_task_Superviseur			
<b>Nom :</b>	<code>_task_Superviseur</code>	<b>Préemption :</b>	<code>0</code> // boolean
<b>Commande :</b>	<code>\$ java -cp /ApplicationEXP/Superviseur.jar labstic.superviseur.Superviseur</code>	<b>Priorité :</b>	<code>2</code> // integer
<b>Dégradée :</b>	<code>aucun</code>	<b>ProcessID :</b>	<code>aucun</code> // string
<b>Alternative :</b>	<code>aucun</code>	<b>ActiveOn :</b>	<code>aucun</code> // string
		<b>Status :</b>	<code>aucun</code> // string
<b>Besoins matériels</b>	<code>_capacity_Process</code> <code>_capacity_Memory</code>		
<b>Dépendances logicielles</b>	<code>needSource(Superviseur.jar)</code> <code>needApplication(IRE)</code> <code>needApplication(Sa)</code>		

FIGURE 4.13 – Description de tâches systèmes

doit posséder des capacités de traitement, un dispositif visuel et un dispositif d'entrée « souris » pour que le patient interagisse avec l'application. La priorité de cette tâche est normale, mais elle ne peut pas être préemptée.

Tous ces éléments, définis individuellement, regroupés forment la base de connaissance de notre environnement. La base de connaissance est une ontologie écrite en langage OWL et est la base du système de déploiement puisque tous les composants de notre application y font référence, que se soit en l'interrogeant ou en la modifiant.

### 4.2.3 Exécution : Déploiement de tâches d'aide au handicap

Avant de débiter la procédure pour lancer notre système, faisons état des informations en notre possession. Nous possédons un fichier ontologie de conception qui est notre base de connaissances conçue pendant l'étape de modélisation et un programme d'initialisation de notre système. L'ontologie doit être placée dans un répertoire spécifique pour que le manager pendant l'étape d'initialisation crée l'ontologie de travail.

#### 4.2.3.1 Initialisation du système d'assistance

*Le manager d'ontologie* Au démarrage de notre système d'assistance, la première application du flot de déploiement instanciée est le manager d'ontologie. À partir de l'emplacement de l'ontologie de conception renseigné, le manager crée une copie de cette dernière. Ensuite, il se charge de vérifier la consistance des descriptions et s'assure que ces dernières n'ont pas été altérées. Le manager d'ontologie fait appel au moteur d'inférence, dans notre cas Pellet, et génère les inférences à partir des liens et des règles établies pendant la conception. La copie de l'ontologie enrichie des nouvelles informations sémantiques est sauvegardée sous le nom d'ontologie de travail qui sera le support global du flot de déploiement. Cette ontologie de travail est sauvegardée à un emplacement spécifique et dans une syntaxe RDF/XML différente d'OWL. Le changement de syntaxe nous permet de simplifier le parcours de la base de connaissances pendant la recherche d'information

des autres constituants du flot. Le code source de la méthode qui génère l'ontologie de travail est présenté sur la figure 4.14.

```
private owlManager(String aFileUrl) {
    fFileUrl = aFileUrl;
}

//private static final String BASE_URL = "/home/seguin/ontologies/cedricINF.owl";
public static void performInferences() throws OWLOntologyCreationException,
    OWLOntologyStorageException {
    /* Get hold of an ontology manager */
    OWLOntologyManager manager = OWLManager.createOWLOntologyManager();
    /* Create a file object that points to the local copy */
    File ontology = new File(fFileUrl); //Partitionner.fFileUrl //fFileTestUrl
    /* Load the local copy */
    OWLOntology localOnto =
        manager.loadOntologyFromOntologyDocument(ontology);
    /* Create Pellet reasoner */
    com.clarkparsia.pellet.owlapiv3.PelletReasoner reasoner =
        PelletReasonerFactory.getInstance().createReasoner(localOnto);
    /* Get hold of an inferred ontology manager */
    OWLOntology inferredOnto = manager.createOntology();
    /* Perform the inferences with Pellet reasoner */
    InferredOntologyGenerator generator =
        new InferredOntologyGenerator(reasoner);
    /* Fill the inferred Ontology with the obtained solution */
    generator.fillOntology(manager, inferredOnto);
    /* Save the inferred Ontology into the initial File */
    IRI ontoIRI = IRI.create(ontology);
    manager.saveOntology(inferredOnto, ontoIRI);
    //manager.saveOntology(inferredOnto, new OWLXMLOntologyFormat(), ontoIRI);
    /* Remove the ontology again so we can reload it later */
    manager.removeOntology(localOnto);
    manager.removeOntology(inferredOnto);
}
}
```

FIGURE 4.14 – Génération de l'ontologie de travail

*Le Superviseur de ressource* Une fois l'ontologie de travail générée, une instance du superviseur est créée pour mettre à jour l'ontologie de travail. La plupart des informations comme les taux d'utilisation des capacités ne sont pas renseignés. Le superviseur construit une liste des ressources présentes à partir des informations du contexte contenues dans l'ontologie. Pour chaque ressource informatique, il étudie les capacités qui lui sont attribuées. Le superviseur teste si la ressource est accessible et établit une connexion Ssh. Il interroge le système d'exploitation et récupère les taux d'utilisation de la mémoire et du processeur. Ces informations sont ensuite inscrites dans le fichier ontologie de travail.

Sur la figure 4.15 a), nous pouvons observer l'utilisation mesurée sur le système d'exploitation au moment de la supervision. La figure 4.15 b) représente la déclaration de l'individu « Desktop » dans l'ontologie de travail. Les taux d'utilisation des capacités ont bien été mis à jour. Remarquons que cette ressource est simplifiée puisque n'apparaissent que les informations liées aux capacités matérielles.

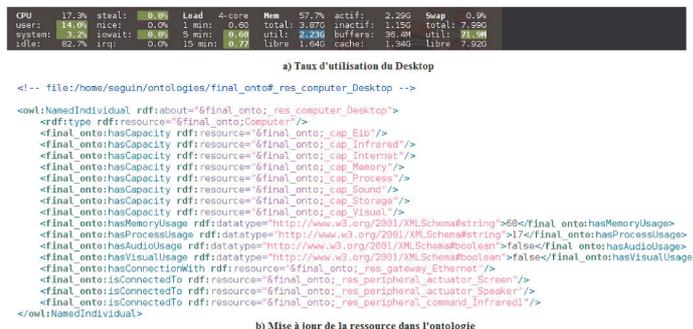


FIGURE 4.15 – Mise à jour de l'ontologie de travail

*Le gestionnaire de tâche* Le programme d'initialisation, crée une instance du gestionnaire de tâche pour déployer les premiers services. Après les services de départ déployés,

une instance du gestionnaire sera exécutée à chaque requête du patient. Nous pouvons donc avoir plusieurs applications gestionnaire exécutées en même temps. Cependant, cela ne gêne pas la gestion de ressources multiples puisque toutes ces instances se réfèrent à la même et unique base de connaissance. Cette base de connaissance est la seule image globale du système et ne peut être modifiée que par une instance à la fois.

*Les services de départ* Comme nous venons de le voir, il est possible de programmer certains services à déployer pendant l'initialisation. Cet ensemble de service est programmé dans l'application d'initialisation qui demande au gestionnaire de tâche leur déploiement.

La première mission demandée au gestionnaire, pendant l'initialisation du système, est de mettre l'interface utilisateur à disposition. Le déploiement de ce service suit le flot que nous avons défini dans le chapitre 3. Autrement dit, le gestionnaire se charge de trouver la ressource la plus adaptée à son exécution. L'interface utilisateur est déployée sur l'ordinateur portable. Le cheminement suivi par le gestionnaire pour aboutir à ce résultat est le suivant. Le gestionnaire étudie en premier lieu, si des ressources sont susceptibles de pouvoir accueillir la tâche. Prenant en considération toutes les capacités physiques ainsi que les besoins logiciels, le résultat est que seul le « Laptop » peut exécuter l'IHM. En effet, parmi les deux autres ressources informatiques « Serveur » ne possèdent pas de capacité d'interaction comme un clavier ou une souris. Le « Laptop » est donc la ressource élue et le gestionnaire exécute la commande de la tâche désirée. Sur l'écran du laptop apparaît l'interface que le patient utilisera pour demander des services. Le gestionnaire va modifier l'ontologie pour faire apparaître, notamment, l'utilisation de la capacité vidéo du « Laptop ». De plus, certaines informations sur la tâche déployée sont inscrites dans la liste des tâches actives, que nous avons décrite dans le chapitre 3.

Désormais, l'utilisateur dispose de son système d'assistance en parfait état de marche. L'ensemble des ressources de son environnement est sous la surveillance du superviseur et les informations sont stockées dans l'ontologie. L'ensemble de ses services est accessible à partir de son interface et leur déploiement est assuré par le gestionnaire de tâche. De plus, une partie des tâches qui assurent sa surveillance ont été lancées pendant l'initialisation.

#### **4.2.3.2 Fonctionnement courant du système**

Nous pouvons résumer le flot de déploiement à l'aide des digrammes de séquences représentés sur la figure 4.16. Le diagramme a) représente le comportement dans le temps du gestionnaire de tâche. Nous pouvons observer qu'à chaque demande de service le gestionnaire interprète la requête et récupère toutes les informations dont il a besoin pour faire son choix grâce à l'ontologie de travail. Lorsqu'une ressource est élue, la tâche est déployée.

Le diagramme b) représente l'exécution de la commande dans l'environnement après que la tâche soit déployée. Notons que le gestionnaire après avoir déployé la tâche récupère les informations fournies par le système d'exploitation comme le PID de la tâche, son statut ou encore éventuellement son SusPID pour les inscrire dans la liste des tâches actives. Cette table est ensuite partagée avec le superviseur qui garantit en plus du contrôle des ressources, l'intégrité des tâches déployées. Nous pouvons observer les cadres rouges qui mettent en évidence les vérifications périodiques des ressources par le superviseur. Remarquons que nous avons prévu une supervision spontanée à chaque nouvelle requête effectuée auprès du gestionnaire. Toutefois, elle n'apparaît pas sur les diagrammes de séquence présentés.

Pour bien comprendre les différentes approches et mécanismes de déploiement que nous

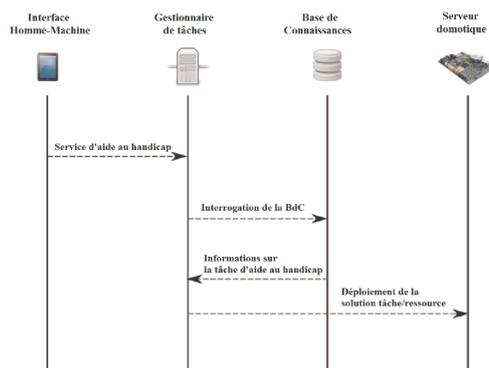


Figure a)

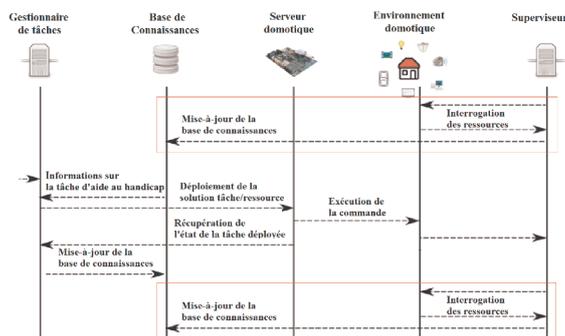


Figure b)

FIGURE 4.16 – Diagrammes de séquence du comportement du flot de déploiement

avons mis en œuvre, reprenons le scénario de notre patient simulé décrit dans la section 4.2.1.2. Si le scénario nominal repris ci-dessous paraît simple, il dans la pratique ponctué d'aléas. En effet, nous avons introduits des évènements qui nous ont été proposés par des professionnels du centre de Kerpape. L'apparition de ces aléas a pour but de mettre en évidence la fiabilité du système et les différentes approches du déploiement de service développées.

- 22h00 Se coucher
- 00h15 **Pris de douleur le patient se lève pour regarder un film**
- 08h00 Se réveiller
- 08h30 Faire sa toilette
- 09h00 Prendre le petit déjeuner
- 09h10 Prendre ses médicaments
- 09h30 Arrivée de son auxiliaire de vie
- 09h15 **Le serveur du salon est déconnecté par l'auxiliaire de vie**
- 10h00 Prendre le bus pour son rendez-vous

Nous pouvons donc noter qu'après s'être couché, le patient est pris de douleur au dos qui le réveille l'empêchent de dormir. Il décide de se lever vers 0h15 pour regarder un

film en attendant que les douleurs passent. Une demi-heure après, il retourne se coucher. Le second évènement perturbateur apparaît vers 9h30, après l'arrivée de son auxiliaire de vie. En effet, ayant besoin d'une prise pour un appareil électrique, cette dernière débranche une prise sur laquelle est connecté le serveur du salon. Le résultat est que le serveur du salon devient inaccessible et ne pourra plus être utilisé tant que la prise n'aura pas été rebranchée.

D'un point de vue du système, ces évènements vont influencer sur le déploiement des services, ce que nous allons étudier avec quatre exemples :

- 1 : Identification de services
- 2 : Lecture d'un film
- 3 : Alerte médicament
- 4 : Alerte rendez-vous

### **Identification de services [30]**

*Rappel de la situation* : Ce service est une tâche périodique planifiée tous les soirs à la même heure. En effet, la tâche identification de service est très consommatrice de ressource de calcul, c'est pourquoi il a été décidé, à la conception, de l'exécuter pendant la nuit, à partir de 23h, lorsque le système n'est plus utilisé par le patient.

*Déploiement du service* : La tâche identification de services est alors déployée suivant l'algorithme étudié dans le chapitre précédent et que nous avons simplifié et repris sur la figure 4.17. La première étape, dite recherche de solution, nous informe que deux ressources peuvent réaliser cette tâche fonction de leurs capacités. Dès lors, une recherche approfondie est effectuée pour savoir, parmi ces deux ressources équivalentes, laquelle est la plus apte à exécuter le service, en limitant l'impact global sur le système. L'algorithme de recherche approfondie est une moyenne de l'ensemble des capacités mesurables des ressources. Par exemple, dans ce cas précis, nous avons besoin d'une importante capacité de calcul, de mémoire, et de stockage. Nous moyennons ces trois variables pour ne garder qu'une valeur représentant la ressource la moins occupée. Ainsi, l'application ne garde qu'une seule ressource, la ressource optimale qu'elle va tenter de déployer dans l'environnement. La seconde étape est d'étudier, au niveau de la ressource sélectionnée, si l'exécution de la nouvelle tâche ne rentre pas en conflit avec une tâche précédemment déployée. Dans ce cas, aucun paramètre n'empêche le déploiement. La tâche est ainsi exécutée sur le serveur du salon et les informations relatives à cette exécution sont ajoutées à la base de connaissance.

### **Lecture d'un film**

*Rappel de la situation* : Vers minuit, alors que le patient est couchée, il est pris de douleur au dos qui l'empêchent de dormir. Il décide de se lever et de regarder un film pour passer son insomnie.

*Déploiement du service* : Par défaut, le serveur du salon est la solution prioritaire puisque la ressource est connectée à la télévision du salon, dont le confort est supérieur à tous les autres écrans. Toutefois, à la recherche de conflit au niveau de la ressource serveur le système de déploiement détecte une anomalie. En effet, pour s'exécuter dans des conditions idéales la tâche impose un minimum de 50% de charge CPU libre, comme nous pouvons le voir sur la modélisation de la tâche initiale figure 4.18. Or nous venons de déployer l'identification de services, une tâche très consommatrice. Le moniteur a d'ailleurs mis à jour le taux d'utilisation de cette ressource lors de sa dernière analyse. On peut voir

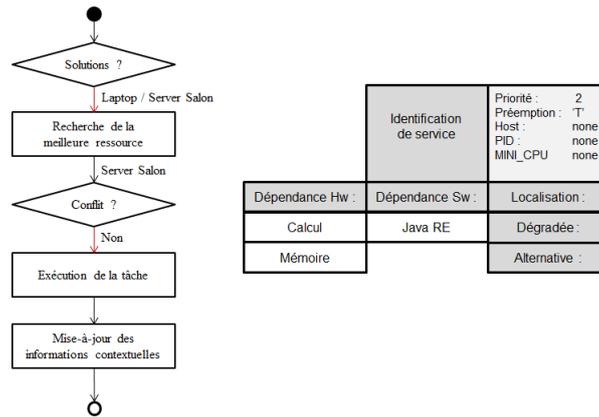


FIGURE 4.17 – Algorithme de déploiement de l'identification de services

que dans l'ontologie le taux d'utilisation du CPU du serveur du salon est utilisé à 70%. Le conflit mis en évidence, l'application de déploiement cherche si le service possède une version dégradée pour garantir à l'utilisateur le service demandé. Le film possède une version dégradée où la qualité est inférieure, mais le besoin de calcul pour décoder le fichier aussi. C'est donc cette tâche qui est choisi à la place du service initial. L'algorithme de déploiement reprend depuis le départ et nous ne notons pas de problème lors de son exécution. Les différentes étapes sont décrites sur la figure 4.18.

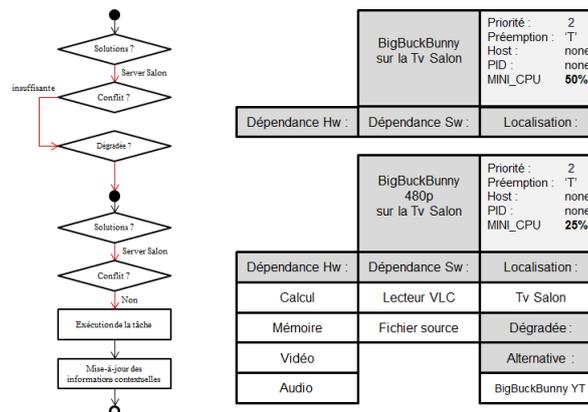


FIGURE 4.18 – Algorithme de déploiement d'un film pendant la nuit

### Alerte médicament

*Rappel de la situation* : Nous avançons dans le temps, vers 9h00. Le patient s'est réveillé comme prévu, il a fait sa toilette et a préparé son petit déjeuner. En prenant son petit déjeuner, il décide de regarder la fin du film commencé pendant la nuit. Le serveur du salon n'étant plus utilisé par le service d'identification, la version initiale est déployée sur la télévision par l'intermédiaire du serveur. Pris par son film le patient oublie de prendre ses médicaments. Une alerte est donc émise.

*Déploiement du service* : Lors du déploiement de la tâche représenté sur la figure 4.19, le serveur du salon est choisi comme la meilleure solution. Cependant, un conflit est repéré au niveau local. En effet, l'alerte utilise la capacité vidéo pour s'exécuter et cette

dernière est déjà occupée par la lecture du film que regarde le patient. L'application de déploiement va alors chercher à savoir quelle tâche est la plus prioritaire et si la tâche en cours peut être suspendue. Effectivement, nous notons que dans sa définition le film peut être préempté et sa priorité est inférieure à celle de l'alerte. L'application de déploiement par l'intermédiaire des mécanismes offert par le système d'exploitation suspend donc le film et exécute l'alerte. Les informations contextuelles sont alors inscrites dans l'ontologie. Cette instance du gestionnaire est alors mise en attente pendant l'exécution du nouveau service. Une fois ce dernier terminé où acquitté le gestionnaire reprend la lecture du film préalablement suspendu.

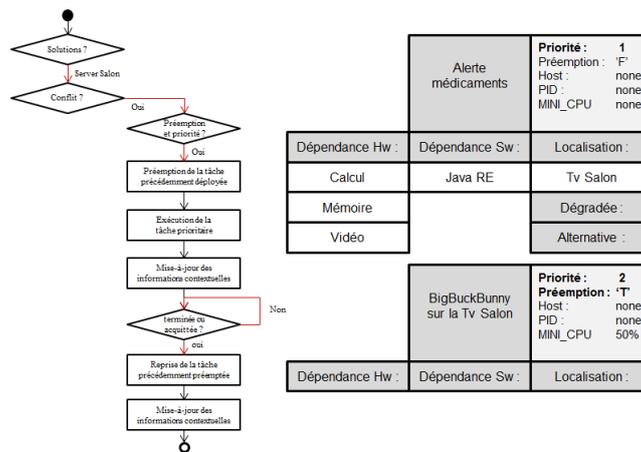


FIGURE 4.19 – Algorithme de déploiement de l'alerte médicament

## Alerte rendez-vous

*Rappel de la situation* : Pour ce dernier exemple, nous avançons encore dans la matinée puisque l'auxiliaire de vie est arrivée et a débranché le serveur du salon. Il est bientôt 10h et le patient ne doit pas oublier son rendez-vous. Le système émet une alerte visuelle pour rappeler au patient son programme de la journée.

*Déploiement du service* : Dans sa forme initiale le rappel de rendez-vous affiche une alerte visuelle. Le serveur du salon étant débranché et le serveur de la chambre ne proposant pas toutes les conditions nécessaires à l'exécution de la tâche, c'est l'ordinateur de la cuisine qui est choisi. Or, l'application de déploiement détecte un conflit puisque la capacité visuelle de cette dernière ressource est déjà utilisée par l'application interface homme-machine. L'application IHM n'est par définition par préemptable ce qui mène le gestionnaire à supprimer cette solution de la liste des ressources sélectionnables. Aucune solution n'est alors disponible et le gestionnaire cherche alors une tâche alternative pour satisfaire l'exécution du service. Le rappel de rendez-vous possède une forme alternative où l'information est transmise non plus par vidéo mais par audio. C'est donc cette solution alternative qui sera déployée par le gestionnaire de tâche comme le montre la figure 4.20.

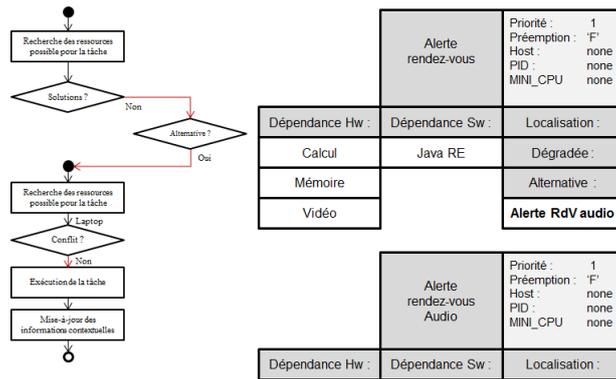


FIGURE 4.20 – Algorithme de déploiement du rappel du rendez-vous

## 4.3 Résultats et limites de la solution proposée

### 4.3.1 Intégration des recommandations de l'aide au handicap

Le monde de l'aide au handicap est soumis à certaines règles et recommandations. Les solutions technologiques doivent s'y soumettre autant, si ce n'est plus, qu'une assistance humaine. Par exemple, une aide-soignante, une infirmière, un ergothérapeute ou encore un médecin sont tous assujettis par le code de déontologie. Ils doivent, par exemple, respecter la vie privée du patient et ne pas divulguer d'informations confidentielles. Dans le cadre de la technologie, les recommandations de l'aide au handicap participent et ont un impact important sur l'acceptation ou le rejet d'une solution.

Nous avons vu que son coût était un critère prépondérant. En effet, nous sommes face à des populations qui paient plusieurs fois le prix de leur handicap, aussi plus nous limitons les coûts d'une solution plus nous augmentons les chances d'acceptation de cette solution. En réutilisant les ressources déjà présentes dans l'environnement qui possèdent des capacités intéressantes inexploitées, nous pouvons limiter ce coût.

Un second critère majeur est l'intrusivité de la solution. En effet, personne ne souhaite être épié constamment dans ses moindres faits et gestes. C'est pourquoi la solution doit se faire la plus discrète possible autant visuellement que de manière auditive. L'utilisation de ressources compactes et la réutilisation des propres ressources de l'utilisateur favorisent la discrétion de la solution.

L'adaptabilité d'une solution, autrement dit, sa capacité à prendre en compte rapidement les changements de l'infrastructure est un avantage considérable dans le déploiement des ressources. Par exemple, si un équipement tombe en panne et qu'il ne peut pas être réparé le modèle doit prendre en compte le changement de son infrastructure. De la même façon, la découverte d'un nouvel appareil doit être prise en compte si l'on veut offrir au patient ces nouvelles fonctionnalités qu'il apporte. Lorsque nous détectons qu'une ressource n'est plus accessible, notre modèle est assez flexible pour prendre en compte cette anomalie et supprimer la ressource.

L'individualité d'une solution, c'est-à-dire, la facilité avec laquelle la solution va pouvoir compenser les besoins spécifiques de la pathologie du patient, fait aussi partie des priorités. Chaque personne, chaque pathologie, chaque handicap est unique et la solution proposée ne peut pas se permettre d'être généraliste. Quel serait l'intérêt d'un système avec une interface tactile pour une personne souffrant de cécité

Outre ces spécifications issues du domaine de l'aide au handicap, le système domotique doit répondre à certains besoins spécifiques comme la fiabilité, la sûreté de fonctionnement ou encore la qualité d'un service. Par conséquent, le temps de réaction ou encore la qualité d'exécution des services sont autant de paramètres qui impactent la viabilité d'une solution et que nous allons étudier dans les sections suivantes.

### 4.3.2 Stratégies de déploiement

Nous venons de voir lors des quatre exemples présentés dans la section 4.2 la mise en pratique des différentes stratégies de déploiement. Ces stratégies se différencient par leur niveau d'action : global à l'échelle de l'environnement multiressources, et local à l'échelle d'une ressource hébergeant des tâches.

Au niveau global, toutes les ressources, répondant à l'ensemble des critères requis par une tâche, sont mises en concurrence. Si aucune ressource ne correspond aux critères requis alors l'application de déploiement va chercher à exécuter une version alternative de la tâche initiale. Si plusieurs ressources répondent aux besoins, le gestionnaire de tâche applique alors son algorithme de choix de la ressource optimale en cherchant à minimiser l'impact du déploiement de la tâche. Autrement dit, le système choisira toujours, pour deux ressources aux capacités équivalentes, celle dont le taux d'utilisation est le plus faible.

Au niveau local, c'est-à-dire une fois la ressource optimale sélectionnée, le gestionnaire de tâche va s'assurer que chacune des capacités requises est bien libre et suffisamment disponible. En effet, une ressource est amenée à exécuter plusieurs tâches simultanément, c'est pourquoi il faut s'assurer que l'exécution est possible. Le gestionnaire pourra alors déployer une version dégradée du service si la disponibilité des capacités n'est pas suffisante. Enfin, en cas de non-disponibilité d'une des capacités, le système va chercher à préempter la tâche occupant la ressource pour libérer la ressource.

### 4.3.3 Mesure des temps de réaction

Ces stratégies de déploiement ont un impact sur le temps d'exécution d'un service. Nous avons mené deux campagnes de mesure : la première sur la plateforme domotique développée par le Lab-STICC de Lorient, et la seconde pendant la semaine de mise en œuvre à Kerpape. Nous avons établi 'le temps réaction du système' comme la durée entre la demande de service effectuée par l'utilisateur et l'exécution résultant de l'effet dans son environnement.

Pour la plateforme domotique, les résultats donnent une réponse sans le gestionnaire inférieur à une demi-seconde. En effet, lorsqu'on demande l'ouverture d'une porte l'exécution est presque instantanée. Tandis qu'avec l'utilisation de l'application de déploiement le temps de réaction du système est de deux secondes. Pour l'appartement domotique de Kerpape, le temps de réaction est d'environ une seconde sans l'application de déploiement et de trois secondes en utilisant notre système.

Si le temps de réaction de notre système a augmenté lors des seconds tests, il faut bien comprendre que l'infrastructure de l'appartement est nettement plus importante que notre plateforme domotique. En effet, que ça soit les actionneurs ou les capteurs, nous avons multiplié le nombre d'éléments de manière significative. Or toutes ces ressources communiquent à travers le bus et peuvent affecter le temps d'exécution d'une tâche. Si l'infrastructure est un premier élément expliquant le temps de réaction du système, notons aussi que l'utilisation de Java applique un temps à prendre en compte. À chaque demande de service, une instance de notre gestionnaire est créée et par conséquent le temps de chargement de l'application ajoute un délai. Nous avons vu que l'ontologie n'était pas

modifiée avant chaque exécution de service, cela n'ajoute donc pas de délai de traitement de la base de connaissances. Toutefois, l'application nécessite de parcourir le fichier ce qui fonction de la taille de l'infrastructure peut jouer là aussi sur le temps de réaction. Enfin, la dernière source de latence introduite par l'application est la communication entre les ressources. En effet, le gestionnaire crée un canal de communication sécurisé avec la ressource qui doit exécuter la tâche. Des optimisations sur le code, notamment au niveau des traitements de la base de connaissances et des méthodes de communication, pourraient nous permettre de gagner un temps non négligeable.

#### 4.3.4 Limites du système de déploiement

Une fois le système initialisé, la base de connaissances peut être considérée comme figée. En effet, les ressources peuvent ainsi être supprimées en cas de défaut, mais les inférences ne sont pas remises à jour. Lorsque la ressource est de nouveau disponible, le système doit alors être réinitialisé pour prendre en compte les changements. Cela peut causer un problème si on cherche à ajouter une nouvelle ressource. Cette dernière ressource doit alors être déclarée dans l'ontologie, mais le système doit aussi être réinitialisé pour inférer les différentes relations avec les autres éléments de la base de connaissances.

Ce problème est encore plus vrai avec les ressources mobiles, telles qu'un téléphone portable, une tablette tactile, etc. Ces ressources qui peuvent sortir ou rentrer dans l'environnement, à tout moment, imposent une prise en compte particulière que nous n'avons pas intégré dans notre système. Cette problématique a donné lieu à une thèse en cours au Domus et au LabSTICC de Lorient.

Une autre limite de notre système est que nous n'intégrons pas le profil du patient. Pour résoudre la problématique du déploiement du service, le système peut être amené à faire des choix sans prendre en compte les besoins du patient. Nous considérons par contre que l'offre de service proposé à l'utilisateur ne dépend pas de nous, mais d'autres projets comme IntelHome. Le corps médical en lien avec le patient est plus apte de fournir une réponse adaptée à ses besoins. Cependant, cette offre de service est une entrée de notre système puisqu'à chaque service proposé, correspondent une ou plusieurs tâches.

Si le choix de la ressource tend à minimiser l'impact de la tâche sur l'ensemble du système, les critères de décision sont uniquement informatiques. Cela pose problème notamment puisqu'au aucun critère de choix ne s'intéresse au patient pourtant au centre de l'utilisation du système. Par exemple, Charles Gouin-Valérand favorisait les critères du patient au détriment des ressources. L'utilisation d'algorithmes multicritères pourrait alors renforcer l'optimisation du choix de la ressource idéale en prenant des critères plus variés en compte comme ceux du patient, mais aussi des ressources.

Le déploiement dynamique des tâches en cours d'exécution n'est pas non plus géré par notre système. Effectivement, une fois la tâche déployée sur une ressource cette dernière reste en place jusqu'à la fin de son exécution, et cela même si une meilleure solution se libère au cours du temps. Les mécanismes pour prendre en compte la migration en cours d'exécution des tâches rajoutent une complexité importante que nous n'avons pas souhaité ajouter, mais qu'il pourrait être intéressant à étudier.

Enfin, l'une des limites les plus importantes est la complexité de la création de la base de connaissance. Chaque ressource et chaque service doivent être ajoutés et décrits précisément manuellement par un expert en électronique. Toutefois, Willy Allègre a prouvé dans ces travaux de thèse que des méthodes de conception assistée peuvent faciliter de déploiement d'une telle solution.

# Conclusion

Dans ce document, nous avons tenté de répondre au verrou qui concerne l'exploitation conjointe des différents équipements composant un environnement domotique. L'optimisation au niveau d'une ressource informatique, par exemple, est assurée par son système d'exploitation, mais nous notons un manque d'outils appropriés pour optimiser les différentes ressources d'un même environnement simultanément.

La solution que nous proposons s'attaque au déploiement de services d'assistance à la personne, aussi variés soient-ils, sur une infrastructure matérielle hétérogène. Pour ce faire, nous avons axé nos travaux suivant deux directions : la conception et l'exécution.

Le déploiement d'un service dans un environnement multi-ressources ne peut être optimal si une entité n'a pas connaissance du contexte global. La phase de conception tend à résoudre cette problématique en uniformisant la description des appareils variés qui composent l'environnement. La modélisation approfondie des équipements domotiques mettant en avant leurs propriétés intrinsèques, favorise l'exploitation de leurs capacités matérielles et logicielles. De plus, nous permettons l'agrégation des équipements, comme des composants, pour former des ressources plus complexes, plus versatiles et plus performantes. En d'autres termes, la base de connaissances regroupe les informations individuelles de chaque élément matériel, composant l'environnement, qui une fois modélisé décrit un super système capable de déployer des services de contrôle d'environnement et d'analyse du comportement sur une infrastructure domotique diffuse. Au niveau des services, leur description se focalise sur les besoins matériels et logiciels requis pour être exécutés par l'infrastructure matérielle. La modélisation des services sous forme de tâche facilite leur décomposition en plusieurs éléments. Ainsi, ces éléments formant un petit scénario dépendant peuvent être exécutés indépendamment les uns des autres. Les ontologies nous ont offert la possibilité d'uniformiser et de réunir toutes les données de notre système d'assistance et de son contexte. Aussi, le déploiement de services d'assistance, sur une infrastructure multi-ressources s'appuie sur la base de connaissance, à l'exécution. L'amélioration des descriptions et modélisations matérielles a permis de prendre en compte l'ensemble des ressources présentes dans l'environnement du patient et d'en exploiter toutes les capacités disponibles. Ainsi, nous avons limité l'ajout de matériel dédié.

Le flot de déploiement que nous proposons est constitué de trois éléments : le manager d'ontologie, le superviseur de ressources et le gestionnaire de tâche. Si l'ontologie assure l'utilisation conjointe des différentes ressources, une version dite de travail est générée, par le manager d'ontologie, à partir du modèle de conception. Cette ontologie de travail est modifiable, altérable et interrogeable dynamiquement par les constituants du flot. Le superviseur est l'entité qui prend connaissance des différents équipements qui composent l'environnement et assure que l'ontologie de travail offre une représentation fidèle du contexte, en la mettant à jour régulièrement. Le gestionnaire de tâche est en charge de définir quelle ressource offre l'exécution la plus optimale du service. Conscient du contexte, grâce à son lien privilégié avec l'ontologie de travail, le gestionnaire élit l'équipement de

sorte que la charge de traitement soit répartie uniformément sur l'ensemble de l'infrastructure matérielle. Le gestionnaire assure le lien entre l'exploitation locale, au niveau d'une ressource, et l'optimisation globale, au niveau de l'infrastructure domotique. Autrement dit, il peut suspendre un service pour libérer les capacités, d'une ressource, réservées et il garantit le fonctionnement conjoint des différentes ressources.

#### DISCUSSIONS DES CHOIX TECHNOLOGIQUES

De nombreux choix technologiques ont été faits pendant nos travaux, nous en avons sélectionné quelques-uns pour tenter de les justifier et d'apporter un supplément d'information.

Le choix d'une infrastructure distribuée avec plusieurs ressources diffuses, au lieu d'une seule dédiée, offre une forte résilience face aux divers aléas possibles. En effet, si un nœud du système est endommagé nous pouvons migrer nos applications vers des nœuds sains et maintenir le fonctionnement global du système. De plus, la redondance de ressource nous offre de la flexibilité pour l'optimisation et permet le partage de capacité. Outre, l'utilisation des appareils domotiques déjà présents dans l'environnement du patient, nous avons donc décidé d'installer des équipements informatiques disposant d'une architecture ARM. Ces derniers sont souvent moins coûteux, moins consommateurs, moins bruyants et moins imposants que leurs homologues avec une architecture x86, sans souffrir d'une trop grande différence de performance. Ces appareils diffus dans l'environnement améliorent la discrétion de la solution et leur faible coût favorise l'accès à la technologie, deux recommandations de l'assistance au handicap qui participent à l'acceptation ou au rejet d'une solution domotique. Comme nous l'avons précisé, nous permettons l'intégration des ordinateurs personnels, souvent avec une architecture X86, car ils sont majoritairement répandus et que de nombreuses personnes en possèdent au moins un. Le portage a été facilité par l'utilisation du système d'exploitation Linux présent sur les deux architectures. Cependant, nous avons fait l'hypothèse que les ordinateurs personnels fonctionnaient tous sous Linux, ce qui dans les faits ne représente qu'une part minoritaire des utilisateurs. Linux s'est imposé à nous comme un outil facilitant l'intégration de plateformes matérielles différentes mais aussi comme un système d'exploitation facilement administrable, stable et performant. Cependant, ce système d'exploitation reste très peu répandu face à Windows que nous aurions pu aussi adresser même si cela aurait demandé d'intégrer des tâches d'aide au handicap avec deux versions (une Windows et une Linux). Un coût en stockage supplémentaire mais nous apportant une flexibilité non négligeable.

Parmi les choix logiciels, nous avons favorisé l'utilisation de Java. Sa machine virtuelle, multi-plateformes, nous offre la possibilité de déployer la même application indépendamment de l'architecture matérielle ou du système d'exploitation, au prix d'une légère perte de performances. Cependant, ce choix reste discutable puisque nous n'adressons qu'un seul système d'exploitation à l'heure actuelle. Nous pouvons cependant avancer que l'utilisation de Java nous a permis de bénéficier des mécanismes de déploiement OSGi. Pour formaliser notre description, nous avons décidé d'utiliser une base de connaissance, une ontologie, et plus particulièrement le langage OWL. Les ontologies nous offrent la possibilité de décrire des concepts hétérogènes et de les regrouper sous une même forme. Grâce à la logique de réflexion et au principe de monde ouvert, il nous est possible d'enrichir notre modèle de nouvelles informations. Le but principal était de faciliter la décision grâce aux inférences de l'ontologie, ce que nous n'avons pas réussi à faire complètement. Dès lors peut se poser la question de l'utilisation de cette technologie à la place d'une simple base de données dont l'impact sur le temps de réaction du système serait plus intéressant. Une première réponse est l'évolution que nous apportent les bases de connaissances. Les bases de don-

nées (monde fermé) sont déterminées et n'offrent pas les évolutions qu'apportent les bases de connaissances (monde ouvert).

Ces discussions nous amènent aux perspectives que nous envisageons.

## PERSPECTIVES

Dans un premier temps, nous n'avons validé qu'une preuve de concept en simulant l'environnement et le comportement d'un patient. Notre plateforme domotique bien que polyvalente (proposition de services de contrôle d'environnement et d'analyse comportementale) reste une maquette expérimentale assez limitée. Par conséquent, la prochaine étape serait de modéliser les besoins d'un patient en conditions réelles et son environnement domotique pour valider le fonctionnement continu de notre système. Les appartements tremplins du CMRRF de Kerpape où l'appartement domotisé en préparation à l'école d'ingénieur de Lorient, permettraient de bénéficier d'un environnement d'expérimentation adéquat. De plus, les tests en conditions réelles permettraient de mesurer plus précisément l'impact du temps de réaction et de la qualité de service de la solution.

Un autre axe d'amélioration de nos travaux serait la prise en compte d'algorithmes multi-critères pour la répartition de la charge de travail. Dans nos travaux, nous avons fait le choix de ne prendre en compte que la charge de traitement informatique (charge processeur et mémoire). Pour des services domotiques qui requièrent des traitements informatiques, notre algorithme de répartition uniforme répond assez correctement aux capacités de l'environnement. Néanmoins, nous pourrions améliorer considérablement le choix de la ressource en prenant d'autres critères plus complexes comme la taille d'un écran pour déployer une tâche visuelle. La proximité d'un équipement, par exemple, pourrait être un critère de décision intéressant. OSGi nous a permis de bénéficier des principes de la programmation orientée composant et des architectures orientées services pour faciliter le déploiement et la gestion des dépendances d'une application. Ces mécanismes sont mis en œuvre à l'échelle d'une ressource mais qu'en est-il du déploiement sur plusieurs ressources ? Les outils d'OSGi semblent apporter une réponse en proposant les outils d'OSGi à l'échelle d'un environnement multi-ressources distribuées.

D'un point de vue plus global, nous proposons de surveiller l'infrastructure support du système domotique au détriment d'une surveillance approfondie des besoins, des préférences et du profil des patients utilisateurs. Une approche conjointe des méthodes de déploiement orientées utilisateurs et des méthodes de déploiement basées sur l'infrastructure support permettrait de fournir un système de déploiement de services d'assistance adaptés au patient et à son environnement. Dès lors, nous pourrions même adresser des problématiques d'environnement multi-usagers multi-ressources plus complexes, comme dans un centre hospitalier, même si cela implique de tenir compte des problèmes de sécurité de l'information. Le passage à l'échelle, autrement dit l'adaptation aux environnements dynamiques (découverte et disparition d'équipements), serait là encore un véritable plus pour le déploiement. Adapter en continu les équipements domotiques donnerait la possibilité au système de s'accorder encore plus fidèlement à son contexte d'environnement. Nous avons fait le choix de ne pas intégrer de ressource mobile, dans notre environnement, comme un téléphone intelligent ou une tablette. Nous possédons bien un ordinateur portable mais nous faisons l'hypothèse qu'il n'est pas déplacé ou du moins toujours en mouvement avec le patient. Les ressources mobiles possèdent un comportement qui n'est pas prévisible de manière simple. Un téléphone intelligent pourra être présent, dans l'environnement domotique, à l'instant  $T$  et le quittera à l'instant  $T+1$ . De plus, le déploiement d'une tâche sur une ressource volatile impose des mécanismes de sauvegarde du contexte de l'application, de sécurité de transmission des données et de redéploiement dynamique, que nous n'avons

pas adressés.

Enfin, l'intégration de nos modèles aux outils de CAO, comme ceux développés au Lab-STICC, faciliterait sensiblement la description de l'environnement et de l'offre de services qui constituent notre base de connaissance.

# Bibliographie

- [1] Patrick Fougeyrollas and Réseau international sur le processus de production du handicap. *Classification québécoise—processus de production du handicap*. Lac St-Charles, Québec : RIPPH= SCCIDIH, 1998.
- [2] World Health Organization et al. World report on disability 2011. 2011.
- [3] Department for International Development (DFID). *Disability, Poverty and Development*. 2000.
- [4] Robert Metts. Disability and development. In *A Background paper prepared fro the Disability and Development Research Agenda Meeting Washington DC World Bank, November 16th, 2004*.
- [5] Willy Allègre, Thomas Burger, Pascal Berruet, Jean-Paul Departe, et al. Conception d’un système domotique pour l’assistance aux personnes dépendantes. *Sciences et Technologies pour le Handicap*, 2012.
- [6] Thomas Kleinberger, Martin Becker, Eric Ras, Andreas Holzinger, and Paul Müller. Ambient intelligence in assisted living : enable elderly people to handle future interfaces. In *Universal access in human-computer interaction. Ambient interaction*, pages 103–112. Springer, 2007.
- [7] Franck Poirier and Igor Schadle. état de l’art des méthodes de saisie de données sur dispositifs nomades : Typologie des approches. In *Proceedings of the 16th Conference on Association Francophone D’Interaction Homme-Machine, IHM 2004*, pages 133–140, New York, NY, USA, 2004. ACM.
- [8] Arie E Kaufman, Amit Bandyopadhyay, and Bernard D Shaviv. An eye tracking computer user interface. In *Virtual Reality, 1993. Proceedings., IEEE 1993 Symposium on Research Frontiers in*, pages 120–121. IEEE, 1993.
- [9] Manuel Jimenez, Francisca Rosique, Pedro Sanchez, Barbara Alvarez, and Andres Iborra. Habitation : a domain-specific language for home automation. *Software, IEEE*, 26(4) :30–38, 2009.
- [10] Pedro Sánchez, Manuel Jiménez, Francisca Rosique, Bárbara Álvarez, and Andrés Iborra. A framework for developing home automation systems : From requirements to code. *Journal of Systems and Software*, 84(6) :1008–1021, 2011.
- [11] Céline Descheneaux and Hélène Pigot. Interactive calendar to help maintain social interactions for elderly people and people with mild cognitive impairments. In *Ambient Assistive Health and Wellness Management in the Heart of the City*, pages 117–124. Springer, 2009.
- [12] Pierre-Yves Nivollet. Un habitat capable de reconnaître les activités planifiées dans un calendrier électronique. 2013.
- [13] Jérémy Bauchet. *Archipel : un framework objet pour une approche ubiquitaire de l’assistance cognitive*. Université de Sherbrooke, 2008.

- [14] Min Chen, Sergio Gonzalez, Athanasios Vasilakos, Huasong Cao, and Victor C Leung. Body area networks : A survey. *Mobile Networks and Applications*, 16(2) :171–193, 2011.
- [15] Hossein Mamaghanian, Nadia Khaled, David Atienza, and Pierre Vanderghyest. Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes. *Biomedical Engineering, IEEE Transactions on*, 58(9) :2456–2466, 2011.
- [16] Huaming Li and Jindong Tan. Heartbeat-driven medium-access control for body sensor networks. *Information Technology in Biomedicine, IEEE Transactions on*, 14(1) :44–51, 2010.
- [17] C Smarr, Cara Bailey Fausset, and Wendy A Rogers. Understanding the potential for robot assistance for older adults in the home environment. *Georgia Inst. of Technology*, 2011.
- [18] Steven W Brose, Douglas J Weber, Ben A Salatin, Garret G Grindle, Hongwu Wang, Juan J Vazquez, and Rory A Cooper. The role of assistive robotics in the lives of persons with disability. *American Journal of Physical Medicine & Rehabilitation*, 89(6) :509–521, 2010.
- [19] Panu Harmo, Tapio Taipalus, Jere Knuuttila, José Vallet, and Aarne Halme. Needs and solutions-home automation and service robots for the elderly and disabled. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3201–3206. IEEE, 2005.
- [20] Sébastien Saint-Aimé, Brigitte Le-Pevedic, Dominique Duhaut, and Takanori Shibata. Emotirob : companion robot project. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pages 919–924. IEEE, 2007.
- [21] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2) :74–82, 2011.
- [22] Eunju Kim, Sumi Helal, and Diane Cook. Human activity recognition and pattern discovery. *Pervasive Computing, IEEE*, 9(1) :48–53, 2010.
- [23] Belkacem Chikhaoui, Shengrui Wang, and H el ene Pigot. Activity recognition in smart environments : an information retrieval problem. In *Toward Useful Services for Elderly and People with Disabilities*, pages 33–40. Springer, 2011.
- [24] Patrice C Roy, Sylvain Giroux, Bruno Bouchard, Abdenour Bouzouane, Clifton Phua, Andrei Tolstikov, and Jit Biswas. A possibilistic approach for activity recognition in smart homes for cognitive assistance to alzheimer’s patients. In *Activity Recognition in Pervasive Intelligent Environments*, pages 33–58. Springer, 2011.
- [25] Willy Allegre, Thomas Burger, Jean-Yves Antoine, Pascal Berruet, and Jean-Paul Departe. A non-intrusive context-aware system for ambient assisted living in smart home. *health and technology*, 3(2) :129–138, 2013.
- [26] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection : A survey. *ACM Computing Surveys (CSUR)*, 41(3) :15, 2009.
- [27] C eline Franco, Jacques Demongeot, Christophe Villemazet, and Nicolas Vuillerme. Behavioral telemonitoring of the elderly at home : Detection of nycthemeral rhythms drifts from location data. In *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*, pages 759–766. IEEE, 2010.

- [28] Vikramaditya Jakkula and Diane J Cook. Anomaly detection using temporal data mining in a smart home environment. *Methods of information in medicine*, 47(1) :70–75, 2008.
- [29] Eric Campo, Marie Chan, Walid Bourennane, and Daniel Estève. Behaviour monitoring of the elderly by trajectories analysis. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 2230–2233. IEEE, 2010.
- [30] Thi Bich Thanh Truong, F Frizon de Lamotte, J-Ph Diguët, and Farida Saïd-Hocine. Alert management for home healthcare based on home automation analysis. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 2128–2131. IEEE, 2010.
- [31] Lionel M Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P Patil. Landmarc : indoor location sensing using active rfid. *Wireless networks*, 10(6) :701–710, 2004.
- [32] Paramvir Bahl and Venkata N Padmanabhan. Radar : An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. Ieee, 2000.
- [33] Aaron S Crandall and Diane J Cook. Coping with multiple residents in a smart environment. *Journal of Ambient Intelligence and Smart Environments*, 1(4) :323–334, 2009.
- [34] Marie Chan, Daniel Estève, Christophe Escriba, and Eric Campo. A review of smart homes—present state and future challenges. *Computer methods and programs in biomedicine*, 91(1) :55–81, 2008.
- [35] Norbert Noury, Gilles Virone, Pierre Barralon, J Ye, Vicent Rialle, and Jacques Demongeot. New trends in health smart homes. In *Enterprise Networking and Computing in Healthcare Industry, 2003. Healthcom 2003. Proceedings. 5th International Workshop on*, pages 118–127. IEEE, 2003.
- [36] Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3) :579–590, 2013.
- [37] Stephen S Intille. Designing a home of the future. *IEEE pervasive computing*, 1(2) :76–82, 2002.
- [38] Stephen S Intille. The goal : smart people, not smart homes. In *Proceedings of ICOST2006 : The International Conference on Smart Homes and Health Telematics. Amsterdam : IOS Press*, pages 3–6, 2006.
- [39] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. *Activity recognition in the home using simple and ubiquitous sensors*. Springer, 2004.
- [40] Sumi Helal, William Mann, Hicham El-Zabadani, Jeffrey King, Youssef Kaddoura, and Erwin Jansen. The gator tech smart house : A programmable pervasive space. *Computer*, 38(3) :50–60, 2005.
- [41] Eunju Kim and Sumi Helal. Fuzzy logic based activity life cycle tracking and recognition. In *Inclusive Society : Health and Wellbeing in the Community, and Care at Home*, pages 252–258. Springer, 2013.
- [42] Eunju Kim and Sumi Helal. Modeling human activity semantics for improved recognition performance. In *Ubiquitous Intelligence and Computing*, pages 514–528. Springer, 2011.

- [43] Sumi Helal, Nitin Desai, Varun Verma, and Choonhwa Lee. Konark-a service discovery and delivery protocol for ad-hoc networks. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 3, pages 2107–2113. IEEE, 2003.
- [44] Diane J Cook, Michael Youngblood, Edwin O Heierman III, Karthik Gopalratnam, Sira Rao, Andrey Litvin, and Farhan Khawaja. Mavhome : An agent-based smart home. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 521–521. IEEE Computer Society, 2003.
- [45] Sajal K Das, Diane J Cook, Amiya Battacharya, Edwin O Heierman III, and Tze-Yun Lin. The role of prediction algorithms in the mavhome smart home architecture. *Wireless Communications, IEEE*, 9(6) :77–84, 2002.
- [46] Gregory D Abowd and Elizabeth D Mynatt. Designing for the human experience in smart environments. *Smart environments : technologies, protocols, and applications*, pages 151–174, 2004.
- [47] Tiffany L Chen, Chih-Hung King, Andrea L Thomaz, and Charles C Kemp. Touched by a robot : An investigation of subjective responses to robot-initiated touch. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 457–464. ACM, 2011.
- [48] Sylvain Giroux, Jérémy Bauchet, H el ene Pigot, Dany Lussier-Desrochers, and Yves Lachappelle. Pervasive behavior tracking for cognitive assistance. In *Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments*, page 86. ACM, 2008.
- [49] Thomas Romain Camier, Sylvain Giroux, Bruno Bouchard, and Abdenour Bouzouane. Designing a nialm in smart homes for cognitive assistance. *Procedia Computer Science*, 19 :524–532, 2013.
- [50] H el ene Pigot, Dany Lussier-Desrochers, J er emy Bauchet, Sylvain Giroux, and Yves Lachappelle. A smart home to assist in recipe completion. *Technology and Aging, Assistive Technology Research Series*, 21 :35–42, 2008.
- [51] Toshiyo Tamura, Tatsuo Togawa, Mitsuhiro Ogawa, and Mikiko Yoda. Fully automated health monitoring system in the home. *Medical engineering & physics*, 20(8) :573–579, 1998.
- [52] Toshiyo Tamura, Atsushi Kawarada, Masayuki Nambu, Akira Tsukada, Kazuo Sasaki, and Ken-Ichi Yamakoshi. E-healthcare at an experimental welfare techno house in japan. *The open medical informatics journal*, 1 :1, 2007.
- [53] Tatsuya Yamazaki. The ubiquitous home. *International Journal of Smart Home*, 1(1) :17–22, 2007.
- [54] Tatsuya Yamazaki. Beyond the smart home. In *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, volume 2, pages 350–355. IEEE, 2006.
- [55] K Matsuoka. Aware home understanding life activities. In *Proceedings of 2nd International Conference On Smart Homes and Health Telematic, ICOST*, volume 4, pages 186–193, 2004.
- [56] Sunny Consolvo, Peter Roessler, and Brett E Shelton. The carenet display : lessons learned from an in home evaluation of an ambient display. In *UbiComp 2004 : Ubiquitous Computing*, pages 1–17. Springer, 2004.
- [57] Roger Orpwood, Sidsel Bj orneby, Inger Hagen, Outi M aki, Richard Faulkner, and P aivi Topo. User involvement in dementia product development. *Dementia*, 3(3) :263–279, 2004.

- [58] Timothy Adlam, Richard Faulkner, Roger Orpwood, Kerry Jones, Jurate Macijauskiene, and Ausra Budraitiene. The installation and support of internationally distributed equipment for people with dementia. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, 8(3) :253–257, 2004.
- [59] M Chan, C Hariton, P Ringiard, and E Campo. Smart house automation system for the elderly and the disabled. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, volume 2, pages 1586–1589. IEEE, 1995.
- [60] Alain Anfosso and Stéphane Rebaudo. Gérontechnologies et contrôle de l’environnement au service du maintien à domicile : le projet gerhome. *Gérontologie et société*, (1) :119–131, 2011.
- [61] E Campo, JP Scotto Di Rinaldi, D Estève, N Bailly, and F Benard. Développement d’une nouvelle génération de gestionnaire d’énergie auto-configurable pour l’habitat : le concept ergdom. In *Annales du Bâtiment et des Travaux Publics, ISSN*, pages 1270–9840, 2003.
- [62] Christophe Lohr. Motivations pour le protocole xaal. 2014.
- [63] Christophe Lohr, Philippe Tanguy, and Jérôme Kerdreux. xaal : A distributed infrastructure for heterogeneous ambient devices. *Journal of Intelligent Systems*.
- [64] Mehdi FARAJALLAH. Accès à internet et services numériques d’aides au maintien à domicile. 2012.
- [65] Mehdi FARAJALLAH and ENCG d’Agadir. Déterminants de l’adoption des services d’aide au maintien à domicile : une évidence empirique. l’exemple de sigaal. 2013.
- [66] Aissam Belabbas. *Méthodes de conception de systèmes contrôle/commande reconfigurables en présence d’aléas : application au handicap*. PhD thesis, PhD thesis, Laboratoire d’Électronique des systèmes Temps Réel, Université de Bretagne-Sud, 2007.
- [67] Said Lankri. *Services et Navigation pour Personnes Dépendantes en Environnement domotique*. PhD thesis, PhD thesis, Laboratoire en Sciences et Techniques de l’Information, de la Communication et de la Connaissance, Université de Bretagne-Sud, 2009.
- [68] Thi Bich Thanh Truong. *Home Automation Monitoring for Assisted Living Services and Healthcare*. PhD thesis, Lorient, 2010.
- [69] Willy Allègre. *Flot de conception dirigé par les modèles pour la commande et la supervision de systèmes domotiques d’assistance*. PhD thesis, Université de Bretagne Sud, 2012.
- [70] Aissam Belabbas, Pascal Berruet, André Rossi, Jean Luc Philippe, et al. A modeling approach to control a handicap technical assistance. *WSEAS Transactions on Information Science and Applications*, 3 :1460–1467, 2006.
- [71] Aissam Belabbas, Pascal Berruet, André Rossi, Jean-Luc Philippe, et al. Cooperative model generation for disabled people assistance. In *5th WSEAS International Conference. on TELECOMMUNICATIONS and INFORMATICS (TELE-INFO’06)*, 2006.
- [72] F De Lamotte, JP Departe, F Le Saout, JP Diguët, and JL Philippe. Quatra : Final report (technical report, in french). *Lab-STICC. See demo here : <http://www.youtube.com/watch>*, 6, 2008.

- [73] Saïd Lankri, Pascal Berruet, and Jean-Luc Philippe. Service reconfiguration in the danah assistive system. In *Ambient Assistive Health and Wellness Management in the Heart of the City*, pages 269–273. Springer, 2009.
- [74] Thi Bich Thanh Truong, F Frizon de Lamotte, and J-P Diguët. Proactive remote healthcare based on multimedia and home automation services. In *Automation Science and Engineering, 2009. CASE 2009. IEEE International Conference on*, pages 385–390. IEEE, 2009.
- [75] Thi Bich Thanh Truong, F Frizon de Lamotte, J-P Diguët, and Farida Saïd-Hocine. Assisted living service identification based on activity patterns. In *Pervasive Computing (JCPC), 2009 Joint Conferences on*, pages 417–422. IEEE, 2009.
- [76] Willy Allègre, Thomas Burger, Pascal Berruet, et al. Model-driven flow for assistive home automation system design. In *18th IFAC World Congress*, 2011.
- [77] Willy Allègre, Thomas Burger, Pascal Berruet, and Jean-Yves Antoine. A non-intrusive monitoring system for ambient assisted living service delivery. In *Impact Analysis of Solutions for Chronic Disease Prevention and Management*, pages 148–156. Springer, 2012.
- [78] Lab-STICC Vity technology. Projet asim : Assistance par serveur intelligent multi-protocoles, 2012.
- [79] Jean-Baptiste Waldner and Jean Baptiste Waldner. *Nano-informatique et intelligence ambiante : inventer l'ordinateur du XXIe siècle*. Hermès science publications, 2007.
- [80] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3) :94–104, 1991.
- [81] Jeffrey O Kephart and David M Chess. The vision of autonomic computing. *Computer*, 36(1) :41–50, 2003.
- [82] Anand Ranganathan, Chetan Shankar, and Roy Campbell. Application polymorphism for autonomic ubiquitous computing. *Multiagent and Grid Systems*, 1(2) :109–129, 2005.
- [83] Aly A Syed, Johan Lukkien, and Roxana Frunza. An architecture for self-organization in pervasive systems. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 1548–1553. European Design and Automation Association, 2010.
- [84] Charles Guin-Vallerand, Sylvain Giroux, and Bessam Abdulrazak. Tyche project : A context aware self-organization middleware for ubiquitous environment. In *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*, pages 924–929. IEEE, 2011.
- [85] Charles Guin-Vallerand, Bessam Abdulrazak, Sylvain Giroux, and Anind K Dey. A context-aware service provision system for smart environments based on the user interaction modalities. *Journal of Ambient Intelligence and Smart Environments*, 5(1) :47–64, 2013.
- [86] Mathieu Vallée, Fano Ramparany, Laurent Vercoüter, et al. Flexible composition of smart device services. *PSC*, 5 :165–171, 2005.
- [87] Mahmoud Ghorbel, Mounir Mokhtari, and Stéphane Renouard. A distributed approach for assistive service provision in pervasive environment. In *Proceedings of the 4th international workshop on Wireless mobile applications and services on WLAN hotspots*, pages 91–100. ACM, 2006.

- [88] Clément Escoffier, Richard S Hall, and Philippe Lalanda. ipojo : An extensible service-oriented component framework. In *Services Computing, 2007. SCC 2007. IEEE International Conference on*, pages 474–481. IEEE, 2007.
- [89] AK Bourke, JV O’Brien, and GM Lyons. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & posture*, 26(2) :194–199, 2007.
- [90] Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. Fall detection from human shape and motion history using video surveillance. In *Advanced Information Networking and Applications Workshops, 2007, AINAW’07. 21st International Conference on*, volume 2, pages 875–880. IEEE, 2007.
- [91] Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau. Robust video surveillance for fall detection based on human shape deformation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(5) :611–622, 2011.
- [92] 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.
- [93] Tom Broens, Stanislav Pokraev, Marten Van Sinderen, Johan Koolwaaij, and Patricia Dockhorn Costa. Context-aware, ontology-based service discovery. In *Ambient Intelligence*, pages 72–83. Springer, 2004.
- [94] Paul Pavlidis, Jie Qin, Victoria Arango, John J Mann, and Etienne Sibille. Using the gene ontology for microarray data mining : a comparison of methods and application to age effects in human prefrontal cortex. *Neurochemical research*, 29(6) :1213–1222, 2004.
- [95] Abraham Bernstein, Foster Provost, and Shawndra Hill. Toward intelligent assistance for a data mining process : An ontology-based approach for cost-sensitive classification. *Knowledge and Data Engineering, IEEE Transactions on*, 17(4) :503–518, 2005.
- [96] Davy Preuveneers, Jan Van den Bergh, Dennis Wagelaar, Andy Georges, Peter Rigole, Tim Clerckx, Yolande Berbers, Karin Coninx, Viviane Jonckers, and Koen De Bosschere. Towards an extensible context ontology for ambient intelligence. In *Ambient intelligence*, pages 148–159. Springer, 2004.
- [97] 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.
- [98] Lorenzo Sommaruga, Antonio Perri, and Francesco Furfari. Domoml-env : an ontology for human home interaction. In *SWAP*, volume 166. Citeseer, 2005.
- [99] Dario Bonino and Fulvio Corno. *Dogont-ontology modeling for intelligent domotic environments*. Springer, 2008.
- [100] 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, 2004.
- [101] Bessam Abdulrazak, Belkacem Chikhaoui, Charles Gouin-Vallerand, and Benoit Fraikin. A standard ontology for smart spaces. *International Journal of Web and Grid Services*, 6(3) :244–268, 2010.

- [102] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5) :907–928, 1995.
- [103] Dieter Fensel. *Ontologies*. Springer, 2001.
- [104] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology : tool for the unification of biology. *Nature genetics*, 25(1) :25–29, 2000.
- [105] Mike Dean, Guus Schreiber, Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L McGuinness, Peter F Patel-Schneider, and L Andrea Stein. Owl web ontology language reference. *W3C Recommendation February*, 10, 2004.
- [106] Kathrin Dentler, Ronald Cornet, Annette Ten Teije, and Nicolette De Keizer. Comparison of reasoners for large ontologies in the owl 2 el profile. *Semantic Web*, 2(2) :71–87, 2011.
- [107] Andrew S Tanenbaum and Herbert Bos. *Modern operating systems*. Prentice Hall Press, 2014.
- [108] Willy Allègre, Cédric Seguin, Thomas Burger, Florent De Lamotte, Pascal Berruet, Jean-Luc Philippe, and Jean-Philippe Diguët. Ambient assisted living with linux. In *Proceedings of the 1st Embed With Linux (EWiLi) Workshop*, ACM, 2011.
- [109] James Gosling. *The Java language specification*. Addison-Wesley Professional, 2000.
- [110] Andre LC Tavares and Marco Tulio Valente. A gentle introduction to osgi. *ACM SIGSOFT Software Engineering Notes*, 33(5) :8, 2008.
- [111] OSGi Alliance. *Osgi service platform, release 6*. IOS Press, Inc., 2012.
- [112] Universite de Sherbrooke Domus. Monitoring dataset, 2010.



## Résumé

À notre époque, la problématique de la prise en charge des personnes en situation de handicap est d'une importance majeure. Nombreuses sont les études qui prouvent qu'économiquement et socialement le handicap impacte fortement la société moderne. La dépendance, résultat des situations de handicap, est l'incapacité de réaliser seul les tâches de la vie courante. Une aide spécialisée, voire l'hospitalisation dans les cas les plus avancés, peut devenir nécessaire. Les politiques ne se sont pas trompés en faisant de cet enjeu une priorité, notamment en France où le gouvernement a débouqué d'importantes sommes d'argent. L'Organisation Mondiale de la Santé, dans sa dernière définition du handicap, explique que le handicap est situationnel et qu'un environnement adapté peut compenser l'incapacité et supprimer la dépendance. Nous pouvons nous demander dans quelle mesure la technologie peut apporter, dans la vie de tous les jours, un certain degré de confort et une autonomie aux personnes dépendantes. La technologie apporte, aujourd'hui par l'intermédiaire des habitats intelligents, des systèmes pervasifs et de la domotique, les moyens nécessaires aux personnes pour gagner en autonomie et vivre plus longtemps à domicile. Ces solutions s'appuient sur des services d'assistance spécialisés, adaptés à la singularité de la pathologie du patient. Ainsi, un patient en situation de handicap physique se verra proposer une solution de contrôle d'environnement tandis qu'un patient souffrant de trouble cognitif pourra compenser son handicap grâce aux solutions d'analyse du comportement. Nous proposons une solution d'assistance unifiant les deux champs d'action, pour compenser les situations de polyhandicaps.

Pour assurer autonomie, confort de vie, sécurité, surveillance et assistance à domicile des personnes dépendantes, les algorithmes d'assistance requièrent beaucoup de ressources de traitement et le manque de ces dernières peut ralentir voire bloquer le déploiement des différentes tâches et remettre en question considérablement l'expérience utilisateur et par conséquent l'acceptation de la solution. Négliger l'impact des ressources sur le déploiement peut mener à des situations de fonctionnement gênantes, voire dangereuses, dans le cas d'une urgence vitale. Peut-on vraiment proposer le meilleur service d'assistance possible si notre infrastructure ne permet pas sa réalisation ? Pour répondre à cette problématique, les concepteurs surdimensionnent la solution. Autrement dit, des équipements supplémentaires, spécialisés et coûteux, sont installés. Toutefois, cette attitude très répandue dans l'industrie ne respecte pas les recommandations de l'aide au handicap, qui cherchent à réduire les critères de coûts et d'intrusivité, notamment. Les habitudes de vie moderne impliquent que la maison d'aujourd'hui, sans modifications particulières, regorge déjà d'équipements domotiques et électroniques aux capacités inexploitées. Nous proposons une approche de modélisation des ressources pour mettre en évidence leurs capacités. Cependant, avoir conscience de ces capacités ne suffit pas, il faut aussi les connaître et savoir les gérer pour faciliter le déploiement des services d'assistance. En effet, si l'exploitation des capacités au niveau d'une ressource est correctement maîtrisée, l'optimisation conjointe des capacités de plusieurs ressources est quant à elle moins développée. Nous avons mis en place un flot de déploiement contextuel qui grâce à la connaissance de son environnement et des ressources qui le composent peut déployer un service sur n'importe quel équipement capable de le réaliser. Une répartition de la charge de traitement de manière uniforme permet de garantir une sûreté de fonctionnement en évitant un engorgement des ressources et une surcharge globale du système.

## Abstract

Nowadays, the problem of care for people with disabilities is a major concern. Many studies prove that handicaps strongly impact modern society, economically and socially. Dependence, resulting in disability situations, is the inability to perform alone the tasks of daily living. A specialized assistance or even hospitalization, in the most advanced cases, may become necessary. Politicians were not deceived by making this issue a priority, particularly in France where the government has provided large sums of money. The World Health Organization, in its latest definition of disability says that disability is situational and that a suitable environment can compensate for the inability and correct the dependency. We can wonder, to what degree, can technology bring in the life of every day, a certain degree of comfort and independence for disabled people? Today, technology brings through intelligent houses, pervasive systems and home automation, the necessary resources for people to gain autonomy and stay longer at home. These solutions are based on specialized support services adapted to the uniqueness of the patient's pathology. Thus, a patient with physical disabilities could have an environment control solution while a patient with cognitive impairment may compensate for his disability through behavior analysis solutions. We offer a healthcare solution combining the two fields of action, to compensate for situations of multiple handicaps.

To ensure independence, living comfort, security, health surveillance and home care for disabled people, algorithms require a lot of processing resources. The lack of processing devices can slow or even block the deployment of different assistive tasks and question considerably the user experience and therefore the acceptance of the solution. Neglecting the impact of resources for deploying can lead to inconvenient situations, even dangerous in the case of vital emergency. Can we really offer the best possible assistance service if our infrastructure does not allow its achievement? To address this problem, designers oversize the solution. In other words, extra dedicated and expensive equipments are installed in the person's environment. This attitude, although widely used in the industry, does not respect the recommendations of ambient assisted living, which aim to reduce the cost criteria and intrusiveness. Modern lifestyles imply that today's home, without any particular modifications, is already full of home-automation and electronic devices with unused capacities. We offer a resource modeling approach to highlight their abilities. However, be aware of these capabilities is not enough, the system also need to know how to manage them to facilitate the deployment of assistive services. Indeed, if the system can properly manage the utilization of the capabilities of one resource, the joint optimization of several resources, at the same, time is not enough controlled. We have set up a contextual deployment flow which, thanks to the knowledge of the environment and resources, can deploy a service on any devices capable of carrying out the operation. A distribution of the processing load, uniformly, helps to ensure dependability of our solution, by avoiding resource bottlenecks and overall system overload.