



**HAL**  
open science

# Une approche multi-agents pour le monitoring diffus de personnes en perte d'autonomie

Richard Fontaine

► **To cite this version:**

Richard Fontaine. Une approche multi-agents pour le monitoring diffus de personnes en perte d'autonomie. Intelligence artificielle [cs.AI]. Université de la Réunion, 2022. Français. NNT : 2022LARE0020 . tel-03847791

**HAL Id: tel-03847791**

**<https://theses.hal.science/tel-03847791>**

Submitted on 10 Nov 2022

**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.



# UNIVERSITÉ DE LA RÉUNION

## Thèse de Doctorat

Pour l'obtention du grade de  
**Docteur en science de l'université de La Réunion**

Spécialité : Informatique

---

---

Présentée et soutenue le 02/09/2022 par :

**Richard FONTAINE**

---

---

**Une approche multi-agents pour le monitoring diffus de personnes en perte  
d'autonomie**

### Composition du jury

<b>Dr. Thierry Antoine-Santoni</b>	Examineur
Università di Corsica Pasquale Paoli	
<b>Dr. Vincent Chevrier</b>	Rapporteur
LORIA Unité Mixte de Recherche	
<b>Dr Jean-Christophe Soulier</b>	Examineur
CIRAD	
<b>Présidente du jury</b>	
<b>Pr. Hélène Pigot</b>	Rapporteuse
Laboratoire DOMUS - Université de Sherbrooke	

### Direction de la thèse

<b>Rémy Courdier</b>	Directeur
Université de La Réunion	
<b>Denis Payet</b>	Co-Encadrant
Université de La Réunion	

Cette thèse a reçu le soutien financier de la Région Réunion et de l'Union Européenne  
(Fonds Européen de Développement Régional - FEDER) dans le cadre du  
Programme Opérationnel de Coopération Territoriale





# Remerciements

"Thèse" est un mot que l'on prononce avec appréhension quand on la commence et avec beaucoup d'émotions quand on la termine. Elle s'inscrit dans une période de vie un peu à part, où toute notre attention est focalisée sur un objectif, d'abord abstrait, et qui, au fil de l'eau, à force de réflexion, de travail et de discussions, se concrétise. Il est donc important pour moi de conclure cette aventure en remerciant les gens qui l'ont vécue avec moi, de près ou de loin. Je voudrais tout d'abord exprimer mes remerciements à mes encadrants Rémy et Denis pour m'avoir donné l'opportunité de préparer mon doctorat sous leur direction pendant ces dernières années. Ce fut pour moi l'occasion d'assouvir et de mettre à profit ma curiosité au sein d'un domaine qui m'était encore inconnu : Les Systèmes Multi Agents.

Je remercie les membres de mon comité de suivie de thèse, Jean-marc Pecontal, Fabien Michel et Noel Conruyt pour avoir consacré de leur temps au suivi de ma thèse et pour leurs perspicacités, commentaires et remarques.

Merci à mes rapporteurs de thèse, Hélène Pigot et Vincent Chevrier, pour leurs remarques qui ont donné un nouvel éclairage sur différents points de mes travaux. Merci aux examinateurs de ma thèse, Thierry Antoine-Santoni et Jean-Christophe Soulier, d'avoir accepté de faire partie du jury.

Je tiens aussi à remercier le laboratoire du LIM pour m'avoir accueilli et les directeurs successifs, d'abord Jean Diatta puis Christian Delhomme et Étienne Payet pour leurs engagements auprès des doctorants du laboratoire.

Merci aussi au laboratoire DOMUS et notamment à Sylvain Giroux pour son invitation au sein de l'université de Sherbrooke ainsi que Hubert Kenfack Ngankam pour son accueil.

De manière moins formelle, ma gratitude va à mes camarades de laboratoire Nathan Aky, Kévin Hoarau, Tahina Ralitera et Irène Velontrasina pour nos discussions joyeuses, mais aussi pour l'entraide qui règne dans ce groupe.

Enfin, je tiens à remercier mes proches, avec une pensée toute particulière à mes amis Julien, David, Julie, Charles, Cyril et à tant d'autres personnes ayant

vécu ma thèse de loin, mais dont la présence m'a permis de me changer les idées.

Ensuite, je veux exprimer toute ma gratitude à ma famille. Je remercie ainsi pour leur soutien : mes parents Christine et Jean-Jacques; mes frères et sœur Sofia, Kieran et Siobhan ainsi que Guy et Mylène et leurs enfants Thomas et Mathis.

Je remercie enfin de tout mon coeur Eileen, ma compagne, pour son amour et son soutien inconditionnel qui m'a permis d'avancer tout au long de cette aventure. Ta présence, tes encouragements ainsi que tes réflexions ont été, sont et me seront toujours essentiels. À tes côtés, rien n'est impossible!



*«Par notre manière de penser et nos attitudes, nous  
construisons notre bonheur ou notre malheur.»*

Paul Verlaine





# Résumé

Nous sommes à une étape de l'évolution technologique où l'environnement s'instrumentalise de plus en plus avec des équipements électroniques interconnectés. Cette révolution, à l'image de celle que nous avons connue avec l'ère d'Internet, se traduit par un changement rapide et radical de notre rapport avec le numérique. Nos ordinateurs, tels que nous les connaissons, sont ainsi voués à disparaître pour laisser place à une informatique omniprésente, ubiquitaire et disséminée dans chacun des objets de notre quotidien. Dans ce contexte, nous proposons au sein de cette thèse le modèle conceptuel d'affordance virtuel re-compositionnel (AVR), dont le but est de permettre une gestion écoconceptive de systèmes fortement connectés capable de tirer profit du maximum d'objets présents dans l'environnement courant. La conception de ce modèle est inspirée du concept d'affordance et peut être décrite sous la forme de deux approches complémentaires. La première exploite le fait que nos objets connectés sont composés de fonctionnalités primaires pour enrichir le milieu sans ajout d'objets supplémentaires ; la deuxième se sert de cette nouvelle expressivité de l'environnement pour permettre à des agents logiciel de recomposer dynamiquement des services. Ce modèle conceptuel générique est décliné sous la forme d'un modèle d'architecture ADW et implémenté dans un framework Java pour Android nommé AFFORD. En ce qui concerne le cadre d'application, nous nous intéressons dans ce manuscrit à l'informatique ambiante et plus précisément à son usage dans le domaine du suivi à domicile des personnes en perte d'autonomie. Nous proposons pour cela une application Android (Ubi-Health), basée sur notre framework AFFORD, doté de l'agent de gestion de chute "BeCareFall". Cette démarche permet de valider nos contributions par un scénario d'application en lien avec un contexte réel. En effet, du fait de l'allongement de la durée de vie, la prise en charge à domicile des personnes en perte d'autonomie représente un réel défi. Au travers de cette démarche, nous participons ainsi à la dynamique actuelle visant à permettre à ces personnes d'évoluer le plus longtemps possible à leurs domiciles et en toute sécurité grâce aux technologies modernes.

**Mots clés :** Affordance, Écoconception, Système multi-agents, Intelligence ambiante, Perte d'autonomie, Framework Android



# Abstract

We are at a stage of technological evolution where the environment is increasingly instrumentalized with interconnected electronic equipment. This revolution, like the one that we experienced with the Internet era, is marked by a quick and radical change in our relationship with the digital world. Our computers, as we know them now, are going to disappear and be replaced by an omnipresent, and ubiquitous computing, which will be disseminated in every object of our daily lives. In this context, we propose in this thesis the conceptual model of virtual recompositional affordance (VRA or AVR in French), that allows an ecoconceptive management of highly connected systems able to take advantage of the most existing objects in the current environment. The design of this model is inspired by the concept of affordance and can be described as two complementary approaches. The first one exploits the fact that our connected objects are composed of primary functionalities to enhance the environment without adding additional objects; the second one uses this new expressiveness of the environment to allow software agents to dynamically recompose services. This conceptual model is represented in the ADW architecture model and implemented in a Java framework for Android called AFFORD. Concerning the application scope, we are interested in ambient computing and more precisely in its use in the field of home monitoring for persons with a autonomy loss. For this purpose, we propose an Android application (Ubi-Health), based on our AFFORD framework, using the fall management agent "BeCareFall". This approach allows us to validate our contributions through an operational case linked to a real context. Indeed, due to the increase in life expectancy, home care for the elderly represents a real challenge. Through this approach, we are participating in the current dynamics aiming to allow these people to evolve as long as possible in their homes and in complete safety thanks to modern technologies.

**Keywords :** Affordance, Ecodesign, Multi-agent system, Ambient intelligence, Autonomy loss, Android Framework



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Contexte . . . . .	1
1.2	Problématique . . . . .	8
1.3	Approche et objectifs . . . . .	9
1.4	Contributions . . . . .	12
1.5	Organisation du mémoire . . . . .	14
<b>2</b>	<b>État de l'art</b>	<b>17</b>
2.1	La télémédecine . . . . .	17
2.2	Technologies . . . . .	25
2.3	Intelligence ambiante . . . . .	33
2.4	Les systèmes multi-agents . . . . .	41
2.5	L'affordance . . . . .	56
2.6	Conclusion générale . . . . .	62
<b>3</b>	<b>Description conceptuelle du modèle AVR</b>	<b>65</b>
3.1	Enrichissement des outils de l'environnement par décomposition	67
3.2	Interopérabilité de l'agent en contexte ubiquitaire . . . . .	71
3.3	Composition adaptative des services . . . . .	72
3.4	Exemple illustratif d'une mutualisation des composants : Dé- tection de chute . . . . .	75
3.5	Conclusion . . . . .	78
<b>4</b>	<b>Formalisation du modèle AVR</b>	<b>81</b>
4.1	L'artefact dispositionnel . . . . .	81
4.2	Workspace dispositionnel et environnement d'exécution . . . . .	85

4.3	L'agent préemptif . . . . .	86
4.4	Conclusion . . . . .	96
<b>5</b>	<b>Modèle d'architecture ADW</b>	<b>99</b>
5.1	Environnement . . . . .	101
5.2	Artefact . . . . .	103
5.3	Le workspace dispositionnel . . . . .	110
5.4	Architecture de l'agent . . . . .	113
5.5	Conclusion . . . . .	116
<b>6</b>	<b>Framework Java : AFFORD</b>	<b>119</b>
6.1	Android . . . . .	120
6.2	AFFORD Core . . . . .	122
6.3	AFFORD Services . . . . .	123
6.4	Entity Manager for Android (EMA) . . . . .	130
6.5	User Interface (UI) . . . . .	133
6.6	Conclusion . . . . .	133
<b>7</b>	<b>Une application Android : Ubi-Health</b>	<b>135</b>
7.1	BeCareFall : Un agent de détection de choc . . . . .	135
7.2	Description de l'application Ubi-Health . . . . .	137
7.3	Conclusion . . . . .	142
<b>8</b>	<b>Conclusion</b>	<b>145</b>
8.1	Rappel du contexte . . . . .	145
8.2	Contributions . . . . .	146
8.3	Couverture des exigences . . . . .	150
8.4	Discussions . . . . .	152
8.5	Perspectives de recherche . . . . .	153
	<b>Bibliographie</b>	<b>157</b>



# Abbreviations

**A&A** Agent & Artifact

**A<sub>2</sub>D** Android Decompositional Device

**ADW** Agent-Disposition-Workspace

**AFFORD** Android Framework For Omnipresent Real Device

**AGGIR** Autonomie G erontologique Groupes Iso-Ressources

**AVD** Assistant de Vie Diffus

**AVR** Affordance Virtuel Recompositionnel

**BAN** Body Area Network

**BDI** Belief-Desire-Intention

**CIC** Component Internal Communication

**DAF** Dispositional Artifact Framework

**EHPAD**  tablissements d'H bergement pour Personnes  g es D pendantes

**EMA** Entity Manager for Android

**PAN** Personal Area Network

**SMA** Syst me Multi-Agent

**UI** User Interface

**WPAN** Wireless Personal Area Network



# Table des figures

1.1	Présentation globale de la solution . . . . .	3
2.1	Architecture de OLDES [140] . . . . .	20
2.2	Exemple de textiles connectés [77] . . . . .	21
2.3	Architecture de CAALYX [7, 150] . . . . .	23
2.4	Exemple de BAN . . . . .	29
2.5	Exemple de partitionnement . . . . .	30
2.6	Problème d’empilement de middleware . . . . .	31
2.7	Processus d’évolution de l’informatique en cinq phases [178] . . .	35
2.8	Représentation schématique des principaux domaines de l’appren- tissage automatique [36]. . . . .	37
2.10	Le principe de TeleCare [29] . . . . .	50
2.11	Exemple d’architecture d’un système de gestion de composition [174]	55
2.12	Complexification de l’interaction à cause de la présence d’agents intermédiaires . . . . .	55
2.13	Exemple d’utilisation d’agent utilisé comme médiateur entre objets et agents [135] . . . . .	56
3.1	Nos objets connectés riches en fonctionnalités . . . . .	65
3.2	Vue globale de l’approche . . . . .	66
3.3	Une mise en valeur des dispositions . . . . .	68
3.4	Une disposition encapsulée . . . . .	70
3.5	Mutualisation des artefacts dispositionnels dans un espace d’inter- action . . . . .	72
3.6	Mise en avant des effectivités . . . . .	74
3.7	Cycle de détection des affordances . . . . .	75

3.8	Un environnement équipé . . . . .	76
3.9	Un environnement dispositionnel . . . . .	77
4.1	Représentation de la modélisation de l'artefact . . . . .	82
4.2	Une mise en valeur des dispositions sous forme d'artefacts . . . . .	84
4.3	Une mutualisation des composants au sein d'un workspace . . . . .	86
4.4	Modèle conceptuel de l'agent . . . . .	87
4.5	Fonction de juxtaposition $\Gamma$ . . . . .	88
4.6	Émergence de l'affordance . . . . .	90
4.7	Décomposition et résolution des buts de l'agent . . . . .	92
4.8	Émergence des possibilités d'action . . . . .	95
5.1	Interactions entre l'agent préemptif, l'artefact dispositionnel et le workspace dispositionnel . . . . .	99
5.2	Niveau d'architecture du modèle ADW . . . . .	101
5.3	Représentation abstraite d'un artefact . . . . .	104
5.4	Diagramme de classes du modèle de décomposition en artefact . . . . .	104
5.5	Représentation abstraite d'un artefact encapsulant une disposition . . . . .	105
5.6	Représentation abstraite d'un artefact encapsulant un capteur . . . . .	106
5.7	Représentation abstraite d'un artefact encapsulant un effecteur . . . . .	106
5.8	Représentation abstraite d'un artefact encapsulant un service . . . . .	107
5.9	Modèle conceptuel de décomposition sous forme d'artefact . . . . .	108
5.10	Exemple d'une décomposition d'un smartphone . . . . .	110
5.11	Architecture du workspace . . . . .	111
5.12	Modèle architectural du workspace . . . . .	112
5.13	Diagramme d'activité de l'agent préemptif . . . . .	114
5.14	Synthèse de l'architecture générale de la proposition . . . . .	118
6.1	Logo AFFORD . . . . .	119
6.2	Framework AFFORD . . . . .	121
6.3	Interfaces SensorArtifact/ActuatorsArtifact . . . . .	124
6.4	Exemple d'un artefact dispositionnel de type effecteur : Light . . . . .	125
6.5	Implémentation d'un SensorArtifact . . . . .	126

6.6	Implémentation de VoiceRecognition . . . . .	127
6.7	Implémentation de l'agent <i>AgentBlink</i> . . . . .	129
6.8	Méthode de création des SensorArtifact . . . . .	131
6.9	Méthode de personnalisation de l'interface . . . . .	133
7.1	Digramme illustrant le cas de la gestion de chute . . . . .	137
7.2	Interface utilisateur de l'application Ubi-Health . . . . .	138
7.3	Interface d'ajout d'un workspace . . . . .	139
7.4	Implémentation de l'artefact encapsulant la synthèse vocale . . . .	140
7.5	Implémentation de la détection du choc et de la demande d'infor- mation . . . . .	141



# Liste des tableaux

1.1	Différents capteurs présents au sein des smartphones . . . . .	11
2.1	Différents capteurs et leurs type de mesure . . . . .	26
2.2	La RFID se développe sous différents supports . . . . .	27
2.3	Liste d'éléments réutilisables présents dans l'habitat . . . . .	28
2.4	Exemple de détection de la propriété de rang supérieur $r$ grâce à la fonction de juxtaposition $j$ . . . . .	58
2.6	Résumé des solutions étudiées concernant le suivi des personnes en perte d'autonomie . . . . .	63
7.1	Quelques exemples de smartphones testés . . . . .	143
7.2	Descriptif détaillé de quelques appareils utilisés lors de la phase de test . . . . .	144
.2	Extrait de la norme de facto utilisé sur Android (source : <a href="http://dataqualitylabs.com/downloadDataFor/Sensor%20Info">http:// dataqualitylabs.com/downloadDataFor/Sensor%20Info</a> )	183

# Chapitre I

## Introduction

### I.1 Contexte

Depuis plusieurs décennies, l'évolution dans les milieux sanitaires, médicaux et sociaux[30] a aidé à soigner de nombreuses maladies autrefois mortelles et à minimiser les séquelles consécutives aux accidents graves. Ce progrès, qui a permis un accroissement général de la durée de vie, a cependant fait émerger une problématique qui ne se posait pas auparavant : celle du vieillissement de la population. D'après un rapport remis à la ministre française des Solidarités et de la Santé le 28 mars 2019 [112] :

- la part des 75 ans ou plus est passée de 6,6 % en 1990 à 9,1 % en 2015 ce qui représente une hausse de 2,5 points en 25 ans;
- en 2040, 14,6 % des Français auront 75 ans ou plus ce qui représente une hausse de 5,5 points en 25 ans;
- en 2015 environ 1 459 000 personnes de plus de 60 ans vivant à domicile souffraient d'une perte d'autonomie.

À cela s'ajoute le fait que même si l'espérance de vie des Français est l'une des plus grandes en Europe, ce n'est pas le cas de l'espérance de vie en bonne santé. Par exemple, une femme française peut espérer vivre encore près de 23,7 ans après ces 65 ans, dont 10,6 en bonne santé contre 16,6 en Suède, 12,4 en Allemagne et 11,9 au Danemark [112]. Ainsi, nous nous trouvons face à une population dont la santé se dégrade, mais ne nécessite pas forcément de prise en charge médicale dans un établissement spécialisé. Les défauts dans la gestion de cette partie de la

population ont d'ailleurs été mis en exergue lors des dernières périodes de crise traversées, comme la canicule d'août 2003 ou encore la crise COVID de 2020. Suite à ces événements, le taux de décès des personnes âgées avait fortement augmenté [119, 125], notamment à cause d'un manque de suivi de proximité approprié. Avec ces différents indicateurs, nous comprenons que la prise en charge des personnes en perte d'autonomie représente dorénavant un réel enjeu social [83, 125].

Pourtant, malgré des créations de postes, l'attractivité de l'emploi dans les Établissements d'Hébergement pour Personnes Âgées Dépendantes (EHPAD) est réduite, comparée à celle d'autres établissements (sanitaires, médico-sociaux) [136]. Ce désintérêt est multifactoriel avec par exemple des conditions de travail pénibles, une perte de reconnaissance du métier ou encore une inadéquation formation-emploi [136]. Ces éléments poussent les personnes à se diriger vers d'autres secteurs, participant ainsi, entre 2012 et 2017, à la diminution de 25 % du nombre de candidatures aux concours d'entrée des instituts de formation des aides-soignants (IFAS)[112].

Cet état de fait a motivé, en 2021, les pouvoirs publics à proposer un projet "visant à agir pour préserver l'autonomie et garantir les choix de vie de nos aînés" [125]. À travers ce dernier, l'Assemblée nationale souhaite privilégier un suivi à domicile pour ceux qui le souhaitent. Parmi ces volets, ce projet de loi vise notamment à donner une alternative aux institutions spécialisées grâce à un accompagnement à domicile. Pour cela, l'assemblée propose la création d'un centre national de preuves de la prévention de la perte d'autonomie et de ressources gérontologiques. Ce centre est chargé, entre autres, d'évaluer et de labelliser les équipements et les aides techniques individuelles favorisant le soutien à domicile. Cette volonté montre que la technologie représente une option envisagée, et probablement incontournable, afin de préserver l'autonomie de ces individus tout en leur proposant un niveau de suivis approprié. En effet, grâce aux avancées dans les domaines de l'informatique et de la communication, il est désormais possible d'instrumentaliser l'habitat avec des dispositifs susceptibles d'apporter une assistance adaptée aux besoins d'une personne [1, 2, 72].

L'usage de l'informatique est donc d'actualité avec une technologie de plus

en plus diffuse au sein de notre environnement. Au travers du terme "diffus", nous entendons le fait que l'accès à l'information devient accessible, rapidement et sans effort, pour tous, n'importe où et n'importe quand. Ces avancées représentent en réalité notre entrée au sein d'une nouvelle ère de l'informatique, dite ubiquitaire [186, 168]. Il s'agit ici de la troisième ère de l'histoire de l'informatique, qui succède à celles des ordinateurs personnels et des "super-ordinateurs". Ces dernières dizaines d'années, l'informatique est devenue une extension de notre environnement en répondant aux besoins des utilisateurs grâce à des technologies de plus en plus précises, nomades et efficaces. Ces innovations touchent de très nombreux domaines tels que la santé [91]; la gestion intelligente des villes[21], le bureau intelligent[142] ou encore le bâtiment intelligent [157]. Ce cadre est donc tout à fait propice à la mise en place de ce que nous définirons comme un assistant de vie diffus (AVD) (Figure 1.1).

AVD

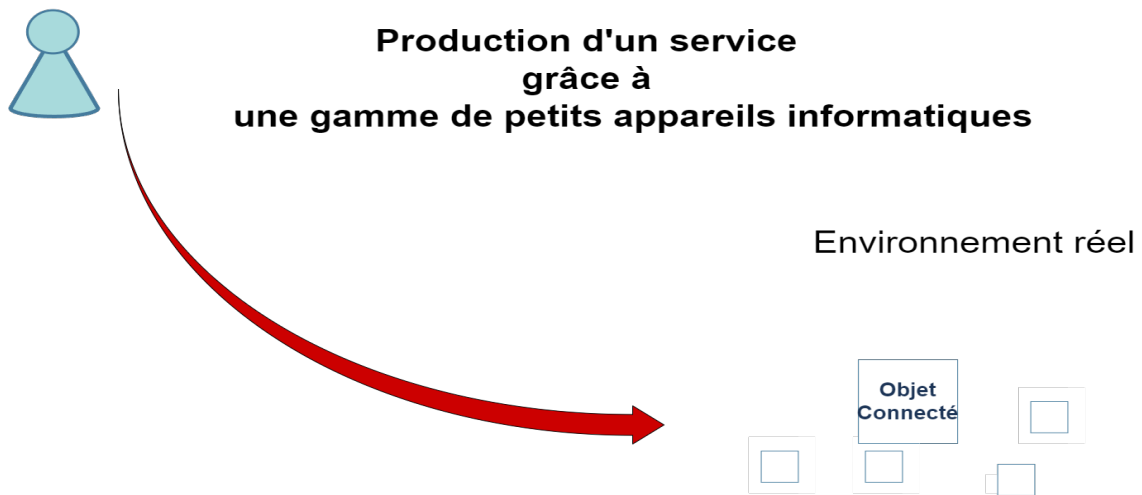


FIGURE 1.1 – Présentation globale de la solution

**Définition 1** (Assistant de Vie Diffus (AVD)). *L'Assistant de Vie Diffus est un système cyber-physique, disposant d'une gamme de petits appareils informatiques disséminés dans son environnement géographique d'action, qui aide à la prise en charge de différents types de public dont il maintient l'autonomie.*

Nous pouvons illustrer le concept d'assistant de vie diffus au travers d'une surveillance de chute. Dans ce cadre, cet AVD surveille en permanence un en-



semble de signaux de vie d'un sujet au travers d'un ensemble de capteur disséminés au sein de l'environnement. En cas de suspicion de chute, un AVD peut alors exploiter un ensemble d'entités physiques afin d'attirer l'attention du sujet via des stimuli ou encore décider de dialoguer directement avec lui. Selon les données captées, l'AVD peut alors décider d'agir en faisant intervenir d'autres acteurs, en contactant un aidant (famille ou professionnel) ou conclure à une fausse alerte.

Il est important de préciser que le but de ces AVD n'est ni de se substituer au corps médical ni de produire un système de surveillance d'individus en vue d'un traitement médical lourd. Il est ici question de proposer une solution souple exploitant dynamiquement le potentiel informationnel disponible dans un environnement de plus en plus informatisé (signes d'activités, signaux biologiques ...).

Cependant, même si les avancées dans le domaine de l'informatique ouvrent des portes dans la proposition d'une aide technologique aux personnes en perte d'autonomie, elle nous met aussi face à de nombreux défis que nous allons détailler au sein des sous-sections suivantes :

- Les contraintes de la perte d'autonomie (sous -section I.I.1)
- La diversité des acteurs en jeu (sous-section I.I.2)
- Les considérations éco-conceptive (sous-section I.I.3)
- Les problèmes d'hétérogénéité matérielle (sous-section I.I.4)

### **I.I.1 Adaptation aux contraintes de la perte d'autonomie**

La perte d'autonomie survient quand l'état de santé d'une personne engendre une diminution de ses capacités fonctionnelles. En France, l'instrument national d'évaluation du degré d'autonomie ou de perte d'autonomie est le modèle AGGIR (Autonomie Gérontologique Groupes Iso-Ressources). Ce modèle propose dix-sept points permettant d'évaluer la capacité d'une personne à réaliser des actes de la vie quotidienne. Ces points sont divisés en deux groupes : discriminants (cohérence, l'habillement, toilette, etc.) et illustratifs (cuisine, ménage, transport, etc.) Dans le cadre d'une détection d'une perte d'autonomie, la mise

en place d'une surveillance régulière [124] auprès de l'individu peut être nécessaire afin qu'il puisse continuer à vivre en sécurité. Les causes emmenant à ces pertes sont alors multiples et peuvent être brutales comme lors d'un accident, d'un choc psychologique, d'une maladie ou s'immiscer progressivement lors de l'avancée en âge. On distingue de plus des causes de nature physique, cognitive ou psychique [124] qui peuvent se manifester de différentes manières telles que :

- Causes physiques
  - des troubles de l'équilibre
  - des tremblements
- Causes cognitives
  - des troubles de la mémoire
  - un manque d'hygiène
- Causes psychiques
  - un repli sur soi
  - une dépression
  - des changements d'habitudes alimentaires

Il n'existe donc pas de définition unique et définitive de ce que peut être la perte d'autonomie ou de son expression. C'est pourquoi il est impossible de prévoir l'évolution des besoins de chacun. De cela résulte une difficulté dans la standardisation des solutions. Pour être envisageable, l'assistant de vie diffus devra donc être suffisamment flexible pour s'adapter aux besoins spécifiques, actuels et futurs, de chaque personne, tant sur les moyens de surveillance que sur leurs niveaux de services.

### **I.I.2 Des acteurs hétéroclites**

Le fait de dispenser un suivi à domicile fait intervenir de nombreux acteurs (proches ou professionnels) qui apportent un soutien aux personnes dépendantes dans l'accomplissement des actes essentiels de la vie courante. Afin de comprendre le rôle de chacun de ces intervenants, nous définirons, en nous basant sur le Code de la santé publique, trois types d'aidants : le professionnel de santé, l'aidant familial, l'aidant naturel.

**Définition 2** (Professionnel de santé). *Selon le Code de la santé publique (CSP), les professions de la santé se décomposent en trois catégories :*

- *les professions médicales (Art. L4111-1 à L4163-10);*
- *les professions de la pharmacie (Art. L4211-1 à L4244-2);*
- *les professions d'auxiliaires médicaux (infirmiers, aides-soignants, ambulanciers, etc.) (Art. L4311-1 à L4394-4).*

**Définition 3** (Aidant familial). *L'aidant familial est une personne qui entretient des liens étroits et stables avec l'aidé, mais qui ne possède pas de formation dans le domaine de la santé [108]. Selon le code de l'action sociale et des familles, l'aidant familial peut être [108] :*

- *Le conjoint*
- *L'ascendant (parents, grands-parents...), le descendant*
- *Le collatéral jusqu'au 4e degré (frères, sœurs, neveux...)*

**Définition 4** (Aidant naturel). *Selon le Code de la santé publique, le statut de l'aidant naturel représente une personne choisie par la personne aidée (généralement un ancien aidant familial) et formée par des professionnels de santé aux gestes liés aux soins prescrits [107].*

Il est à noter que chacun de ces aidants ne possède ni les mêmes exigences ni les mêmes droits d'accès aux informations personnelles de la personne aidée. Dans le cadre de la proposition d'un AVD, il est donc important de prendre en compte la diversité de ces acteurs afin de proposer un service cohérent et multimodal.

### 1.1.3 Considérations éco-conceptive

À la vue du nombre de personnes, d'objets et de cadres d'usage concernés par un tel système, il est de la responsabilité des concepteurs de considérer l'impact écologique. En effet, dans le cadre d'un système ambiant, une tendance naturelle consiste à ajouter de nouveaux équipements à chaque fois que l'on doit produire un nouveau service.

Cependant, une mise à l'échelle basée sur un tel postulat apporte de nombreuses problématiques environnementales. Notamment en ce qui concerne la prise en compte des réserves mondiales en matière première. Selon l'Union européenne [37], en 2020, il existait 30 matières premières critiques (contre 14 en 2011) et parmi elles, de nombreux éléments sont directement liés aux nouvelles technologies : le Cobalt (batteries lithium-ion); le Germanium (fibres optiques); le Hafnium (processeur); l'Indium (écran tactile); ou encore le Tantale (écrans à cristaux liquides, les puces de mémoire vive dynamique (DRAM)). De plus, l'année dernière la Bauxite (principal minerai permettant la production d'aluminium.), le Lithium (batterie) et le Titane (industrie) ont été ajoutés à cette liste pour la première fois.

Dans le cadre de leurs usages dans les nouvelles technologies, ces éléments sont majoritairement liés à la fabrication des ordinateurs, des imprimantes et d'autres objets numériques usuels. Mais depuis 2015, on remarque qu'un basculement s'est opéré [23] :

- Les télévisions représentaient 5 % à 15 % des impacts en 2010 contre 9 % à 26 % d'ici à 2025
- Les smartphones représentaient 2 % à 6 % des impacts en 2010 contre 4 % à 16 % d'ici à 2025
- Les objets connectés représentaient 1 % des impacts en 2020 contre 18 % à 23 % d'ici à 2025

À la vue de ces chiffres et du fort accroissement de la production des objets connectés, nous mesurons l'importance de l'impact de cette évolution technologique sur nos réserves mondiales de matières critiques. Ce constat alarmant nous révèle qu'il est nécessaire d'adopter une nouvelle vision éco-conceptive afin de diminuer les impacts liés aux technologies pouvant être créées, notamment en termes d'extraction des matières premières.

#### **I.I.4 Problèmes d'hétérogénéité matérielle**

Au sein d'un environnement cyber-physique composé d'une multitude de composants informatiques hétérogènes (capteurs, effecteurs), il est difficile de

déterminer toutes les possibilités fonctionnelles offertes par son environnement. Cette difficulté à gérer la complexité dans la détection des opportunités cyber-physiques nous rappelle le paradoxe de Moravec [122], connu en intelligence artificielle qui avance le fait suivant : *"il est relativement facile de faire en sorte que les ordinateurs présentent des performances de niveau adulte sur des tests d'intelligence ou de jeu de dames, et difficile, voire impossible de leur donner les compétences d'un enfant d'un an en matière de perception et de mobilité"*. Ce paradoxe met en avant le fait que les capacités sensori-motrices basiques des organismes vivants, dans notre cas l'interaction d'un humain avec son environnement, sont plus difficiles à reproduire artificiellement que les capacités cognitives de plus haut niveau tel que le raisonnement [122].

En ce qui concerne les capacités sensorielles, il est possible de faire un parallèle entre ce paradoxe et les difficultés rencontrées au sein des systèmes informatiques ubiquitaires fortement hétérogènes tant au niveau matériel (téléphone portable, montre, capteur) que logiciel (agent logiciel, base de données, traitement). L'enjeu est donc de réussir à abstraire cette complexité technologique afin de faciliter le développement de systèmes de gestion d'environnement connecté.

Sur le point de la motricité, à la vue de l'aspect ubiquitaire de notre proposition, nous ne visons pas la présence d'entité physique mobile. Ce qui réduit donc grandement l'ampleur du problème posé par le paradoxe.

## 1.2 Problématique

À partir des contraintes précédemment soulevées, un ensemble d'exigences émergent telles que :

- [C1] L'Ubiquité : Pouvoir interagir avec tout objet physique de l'environnement de l'utilisateur et permettre l'interaction entre les objets disséminés
- [C2] L'Adaptabilité : S'adapter à l'évolution et à la diversité des contextes et des besoins de chacun.
- [C3] L'Éco-conception : Privilégier une utilisation des appareils déjà présents et mettre en avant leurs mutualisations

- [C4] L'Interaction naturelle : Posséder une souplesse au niveau des médias d'interaction afin de s'adapter aux acteurs rencontrés
- [C5] Déploiement : Proposer une démarche qui permet un déploiement rapide sur un maximum d'appareils existants.
- [C6] Autonomie : Le système doit pouvoir fonctionner en toute autonomie sans la présence systématique d'un opérateur.
- [C7] Multi-pathologique : Le système doit répondre à des problématiques issues de diverses pathologies

Basés sur ces contraintes, nous pouvons résumer notre problématique de la manière suivante :

**Problématique :**

Comment concevoir un assistant de vie diffus capable de s'adapter à l'évolution des besoins d'une personne en perte d'autonomie tout en tirant profit d'un maximum d'objets connectés de son environnement courant et en s'adaptant à la dynamique, la distribution et l'imprévisibilité de cet environnement cyber-physique?

### 1.3 Approche et objectifs

Comme nous l'avons montré, notre environnement est de plus en plus informatisé. Nous avons ainsi pris l'habitude d'exploiter, dans des cadres précis, les fonctionnalités de ces objets connectés. D'ailleurs, cette tâche nous semble si naturelle que nous ne remarquons même plus leurs natures. Or, ces fonctionnalités sont en réalité des combinaisons en cascades de fonctionnalités primaires de plus bas niveau.

Pour répondre à l'ensemble des verrous présentés, nous proposons de combiner ces fonctionnalités primaires présentes au sein de nos appareils dans le but de produire des services de plus haut niveau. Pour cela, nous montrons dans un premier temps que notre environnement possède en réalité un potentiel sous-exploité.

Cette richesse se traduit par la présence au sein de nos objets de fonctionnalités de bas niveau tels que la captation d'information, l'interaction avec l'environnement ou le traitement de données. Ces fonctionnalités, que nous définissons pour l'instant comme primaires, peuvent alors être la source de services jusqu'alors passés inaperçus. Des exemples peuvent être donnés comme l'utilisation de capteurs de détection infrarouges, utilisés au sein des équipements anti-infractions [177], en tant que capteurs d'activité ou encore de l'utilisation des accéléromètres, présents au sein de tous nos smartphones en tant que détecteur de chute [4, 15, 109, 110].

Toutefois, la mise en place d'une telle approche pose la problématique de la révélation de ces opportunités cachées.

Cette démarche peut être synthétisée en deux étapes :

**Décomposition en disposition** Nous proposons de ne plus voir un objet de manière monolithique, mais plutôt en tant qu'entité composée d'un ensemble de dispositions atomiques. Analyser notre environnement sous ce nouveau prisme révèle alors qu'il est bien plus riche en opportunités que nous pourrions le penser. Une ressource issue d'un objet n'est ainsi plus associée à un seul usage, mais devient un outil virtuel mutualisé. Pour illustrer cette richesse, nous pouvons constater, au travers de l'ensemble des objets de notre quotidien (télévision, smartphone, ampoules connectés, ordinateur, etc.), que notre environnement fourmille de ressources locales ou distantes capables de transmettre et d'analyser de l'information ou d'agir sur un environnement. Pour aller plus loin, nous pouvons citer l'exemple des smartphones dont la liste de capteurs embarqués est impressionnante (Tableau I.1).

Type de capteurs	Détection
HRM	Rythme cardiaque
Microphone	Son/Volume sonore/souffle
Luxmètre	Mesure de luminosité
Iris/Empreinte	Authentification
Accéléromètre	Mouvement
Capteurs météorologiques	Humidité/température /pression atmosphérique

TABLEAU 1.1 – Différents capteurs présents au sein des smartphones

**Recomposition dynamique des services** Dans un second temps, nous proposons d'enrichir les fonctionnalités présentes au sein de l'environnement grâce à la présence d'entités intelligentes capables de recomposer dynamiquement des services. Nous nous appuyons alors directement sur notre nouvelle granularisation de l'environnement afin de composer des services jusqu'alors ignorés.

Grâce à ce découplage entre ressources et services, nous proposons ainsi un usage plus complet, souple et dynamique de notre environnement.

Afin d'illustrer cet objectif, nous proposons un scénario d'usage, explicitant le déroulé de la mise en place d'un tel système.

### 1.3.1 Scénario d'usage

Arthur est une personne âgée qui vit seul chez lui. Depuis le décès de son épouse, il ne reçoit de la visite que de la part de sa petite fille Lise, une fois toutes les deux semaines. Depuis quelques années, Arthur souffre de douleurs chroniques à la hanche. Parfois, il souffre de crises de douleurs aiguës qui l'empêchent de se mouvoir correctement et cela rend ses déplacements difficiles. Lors de sa dernière crise de douleur, Arthur est tombé chez lui. Alors dans l'incapacité d'appeler les secours, Arthur est resté longtemps sans assistance avant d'être pris en charge par sa petite fille. Cette dernière a contacté les secours qui les ont



transportés à l'hôpital. Suite à sa prise en charge, le médecin qui s'occupe de leur dossier, Jérôme, leur parle d'un AVD qui pourrait permettre à Arthur d'être assisté à domicile. Lise, qui a peur pour la santé de son grand-père, demande à avoir plus d'information et prend contact avec Fabrice qui est chargé de la mise en place du dispositif. Lors de leur entretien, Lise avoue à Fabrice qu'elle n'est pas en mesure d'acheter de nouveaux matériels ou outils pour assurer le suivi de son grand-père. À cela s'ajoute le fait qu'Arthur semble réfractaire à transformer son environnement habituel. Il semble cependant bien avoir intégré l'utilisation d'un smartphone, qu'il utilise uniquement pour appeler quelques proches, et d'une montre connectée, qu'il porte quotidiennement, mais dont il n'utilise que la fonction horloge. Lise craint donc qu'à la vue de leur situation, Fabrice ne puisse leur offrir d'alternatives. Fabrice leur propose alors un AVD : Ubi Health. Cette solution permettrait de proposer un système de détection de chute avec les dispositifs déjà présents chez Arthur. Fabrice leur précise qu'une fois le tout installé, aucun autre paramétrage ne sera à réaliser ni par Lise, ni par Arthur. Pour ce faire, il installe l'AVD sur le smartphone et la montre d'Arthur afin de détecter les chutes éventuelles. Fabrice leur explique rapidement que ces dispositifs contiennent des capteurs qui serviront à alerter Lise en cas de besoin. Suite à cette première installation, Fabrice installe l'AVD sur le téléviseur et le téléphone portable de Lise. Cette seconde installation permettra, si une chute est détectée chez Arthur, d'envoyer une alerte sur le téléviseur de Lise et de faire sonner son téléphone.

Ces installations correspondent tout à fait aux besoins et demandes d'Arthur et Lise et sont réalisées sur mesure. Fabrice leur assure cependant que la solution Ubi Health est souple et adaptative. Ainsi, si la situation de Lise et Arthur évolue, la solution pourra tout à fait faire de même. Fabrice, Lise et Arthur resteront donc en contact afin de suivre l'évolution des besoins d'Arthur.

## 1.4 Contributions

À travers ce changement dans la gestion de notre environnement informatisé, nous avons pour but d'exploiter pleinement nos ressources informatiques

et leurs fonctionnalités dans le cadre du suivi de la perte d'autonomie.

Ce travail vise ainsi un double objectif de recherche : d'une part, être acteur d'une révolution en marche par la proposition d'un modèle de gestion plus intelligente et respectueuse de l'environnement; et d'autre part, mettre à profit le potentiel positif de cette nouvelle ère technologique afin de contribuer à améliorer le cadre de vie des personnes en situation de perte d'autonomie (accident, maladie, personne âgée, ...) par la proposition d'outils informatiques adaptés à leur situation.

### Synthèse de la contribution :

Dans ce cadre, nous proposons de définir un modèle générique d'affordance virtuelle recompositionnelle, résilient et éco-conceptif, basé sur une décomposition et une mutualisation des dispositions fournies par les objets connectés.

Cette contribution est décrite au sein de ce manuscrit au travers de quatre propositions :

- Un modèle conceptuel générique : Affordance Virtuel Recompositionnel (AVR)
- Un motif d'architecture ainsi que son modèle associé : Agent-Disposition-Workspace (ADW)
- Un framework Java pour Android de gestion autonome d'environnements cyber-physiques : AFFORD
- Une application de gestion de chute : Ubi-Health

Ces propositions touchent, de plus, plusieurs disciplines de l'informatique :

**Éco-conception** Notre démarche s'appuie sur une volonté de limiter l'impact écologique lié à la surabondance des appareils [120]. Dans ce cadre, nous avons décidé de proposer une valorisation des possibilités d'un objet. Pour permettre cela, nous proposons de rendre l'utilisation d'une ressource informatique possible au-delà de l'objet qui le porte. Cela se traduit, au sein de notre approche, par un modèle conceptuel générique de décomposition et de mutualisation des possibilités réelles de nos ressources connectées. Cette proposition est ainsi une

option complémentaire à la vision standard de l'éco-conception, centrée autour de la consommation énergétique.

**Système collectif adaptatif** Grâce à leurs capacités d'adaptation et leurs autonomies, l'agent présente les caractéristiques nécessaires à la gestion d'environnements connectés [101, 116, 117, 135]. Dans ce cadre, nous proposons un modèle d'agent préemptif capable de composer dynamiquement les services en fonction des dispositions présentes au sein de l'environnement.

**Génie Logiciel** Afin de permettre la mise en place d'AVR en contexte cyber-physique, nous proposons un modèle d'architecture ADW. Cette architecture permet de fournir une solution aux défis d'hétérogénéité tant au niveau des services à assurer que de la gestion des ressources disponibles. Cette architecture est implémentée au sein du framework AFFORD et de l'application Ubi-Health.

## 1.5 Organisation du mémoire

Le reste de ce manuscrit est divisé comme suit.

Le chapitre 2 présente le contexte lié à ce travail de recherche et est divisé en deux grands axes. Nous décrivons tout d'abord de manière générale les caractéristiques des propositions actuelles en ce qui concerne le suivi à la personne et présentons les méthodes courantes de gestion et de captation de l'information dans un environnement connecté. Le second axe introduit les bases théoriques sur lesquels reposent nos propositions.

Basée sur cet état de l'art, la contribution de ce travail de thèse s'échelonne du chapitre 3 au chapitre 7.

Le chapitre 3 présente notre proposition aux problématiques posées par notre cadre de recherche. Nous y présentons dans un premier temps notre modèle AVR et nous l'illustrerons par un scénario d'usage. Par la suite, nous définissons et formalisons dans le chapitre 4 les différentes notions proposées dans le cadre de cette thèse

Le chapitre 5 est dédié à la présentation du modèle d'architecture ADW. Ce modèle décrit d'une manière symbolique et schématique les différents éléments mis en relations au sein du système.

Le chapitre 6 propose une implémentation sur Android du modèle d'architecture sous la forme d'un framework AFFORD. Cette section est surtout axée sur les solutions techniques liées à la mise en place de notre proposition.

Au chapitre 7, nous proposons une preuve de concept avec l'application Ubi-Health et son exemple d'un détecteur de chute.

Enfin, nous terminons ce manuscrit par un résumé des différentes contributions de cette thèse ainsi que par la mise en avant des perspectives qui lui sont associées à court et moyen terme.



# Chapitre 2

## État de l'art

Au sein de ce chapitre, nous allons proposer un état de l'art des différents concepts utiles à la compréhension des approches mis en avant dans ce manuscrit.

Pour cela, nous commencerons par une étude de la télémédecine en tant que pratique la plus répandue en ce qui concerne le suivi à domicile.

Puis, nous nous concentrerons sur les technologies permettant leur mise en place.

Nous exposerons ensuite la notion d'intelligence ambiante, en nous intéressant à l'enrichissement qu'elle peut fournir dans le cadre de la proposition d'un suivi, avant de nous axer sur l'utilisation des systèmes multi-agent.

Avant de conclure cet état de l'art, nous traiterons finalement de la notion d'affordance, en tant qu'alternative aux approches standards liée à l'interaction avec l'environnement.

### 2.1 La télémédecine

Lorsque l'on traite de l'apport de l'informatique appliqué au domaine de la santé, le premier concept qui émerge est souvent en rapport avec la télémédecine. Ce terme, composé du mot médecine accolé au préfixe télé- qui signifie en grec "au loin", désigne toutes les pratiques médicales qui peuvent être effectuées à distance.

En France, l'Organisation Mondiale de la Santé (OMS) définit dès 1997 la

télémédecine comme étant la partie de la médecine qui utilise la transmission par télécommunication d'informations médicales (images, comptes rendus, enregistrements, etc.), en vue d'obtenir à distance des prestations, des actes, ou encore d'effectuer une surveillance de l'état des patients. Cette pratique est définie plus précisément au sein de l'article L. 6316-1 du code de la santé publique comme étant *les actes médicaux, réalisés à distance, au moyen d'un dispositif utilisant les technologies de l'information et de la communication*.

Il est possible de classer ces propositions en quatre générations [11] :

- Génération (1) : aucun système de soins télé médicaux complet n'est développé et les mesures d'intérêt sont transférées à des prestataires de soins (infirmière ou médecin) de manière asynchrone.
- Génération (2) : Utilisation de transfert de données synchrone avec un centre disposant d'un personnel médical capable de reconnaître, au sein de plages d'heure définies, des changements importants dans les mesures essentielles.
- Génération (3) : Présence d'une aide constante à l'analyse et à la prise de décision même en dehors des heures de bureau.
- Génération (4) : Extension des systèmes de troisième génération, avec des données issues de dispositifs télé médicaux invasifs et non invasifs.

Au sein de ce manuscrit, nous ne nous intéresserons pas aux premières générations, dont les usages s'apparentent plus à du suivi téléphonique pouvant conduire à une intervention, qu'à la notion de télémédecine actuellement admise, automatisée et s'appuyant sur des capteurs connectés [10].

### 2.1.1 Exemples de projet

Parmi les actes de télémédecine, la pratique la plus proche de notre axe de recherche concerne la télé-vigilance qui a pour objet *de permettre à un professionnel médical d'interpréter à distance les données nécessaires au suivi médical d'un patient [...]*.

La télé-vigilance est actuellement au coeur des mesures visant à garder les personnes âgées à leur domicile et à mener à la création de nombreux outils de

surveillance à distance.

Ce sont donc ces propositions que nous allons expliciter ici.

**OLDES** Le projet OLDES (OLDer people E-Services at home) est un des projets financé par le CETIC (Centre d'Excellence en Technologies de l'Information et de la Communication) et propose de répondre à la problématique du suivi à domicile des personnes âgées en mettant à leur disposition des moyens de communication et de surveillance Low-Cost [27]. En effet, Oldes prend en charge une gamme évolutive à différents coûts. Ce projet a la particularité de proposer, en plus d'un volet sur la santé, toute une partie liée au bien-être des personnes âgées à domicile, par exemple en leur offrant des moyens de se divertir et de créer des réseaux sociaux. Le patient utilise pour cela un système domestique fourni par OLDES, chargé de faire le lien entre les capteurs mis en place et la clinique (Fig.2.1) [140]. La plateforme fait intervenir un opérateur, chargé de créer et d'enregistrer le profil du patient afin que ce dernier soit capable de lui venir en aide au quotidien. Ces données recueillies par le dispositif informatique permettent ainsi d'améliorer automatiquement le suivi par les médecins en affinant la pertinence des consultations.

**@Home** @Home est une plateforme de télévigilance dont le but [155] est de récupérer des données de suivi et de les analyser afin d'appeler des centres médicaux en cas de danger. @Home va surtout s'adresser aux patients souffrant de maladies chroniques dans le cadre d'une observation du traitement médicamenteux. L'architecture de @Home peut être divisé en trois sous-systèmes principaux :

- Le sous-système Homesub : responsable de l'acquisition, de la collecte, de l'analyse et de la transmission des signes vitaux du patient (SpO<sub>2</sub>, tension artérielle, ECG). Un ordinateur est mis à disposition chez le patient pour le monitorer et faire le lien entre lui et la clinique médicale.
- Le sous-système lié à l'hôpital : responsable de l'analyse, de l'évaluation, de l'archivage et de la représentation des données.



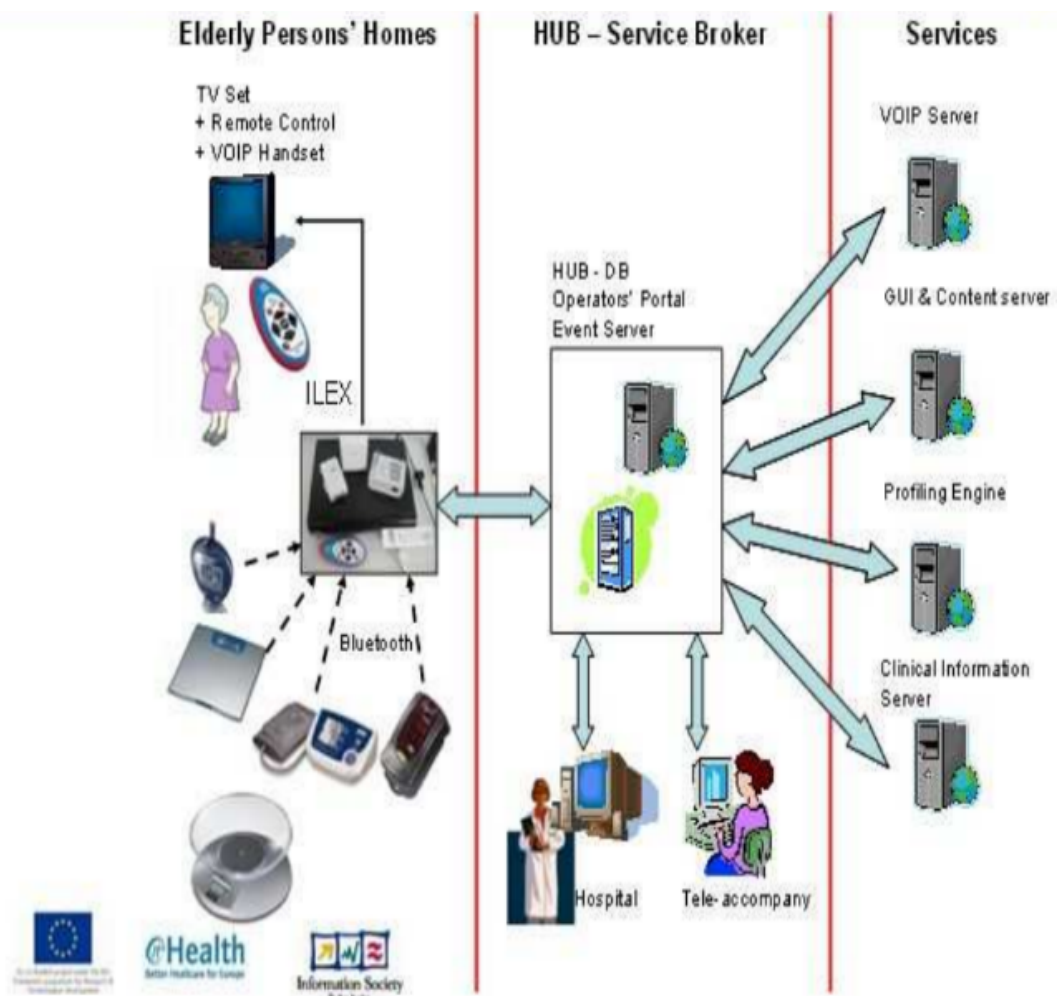


FIGURE 2.1 – Architecture de OLDES [140]

— Le sous-système télématique : c'est le système de communication qui transmet les données enregistrées par les capteurs au serveur de la clinique (GSM et Bluetooth).

Grâce à cette installation, en plus du déclenchement d'alertes, les médecins sont en mesure de vérifier à distance si l'état du patient s'améliore et s'il prend ses médicaments. Dans le cas contraire, ils engagent alors des entretiens avec les patients afin de modifier le protocole mis en place.

**MyHeart** MyHeart [77] est un projet européen pour la gestion des problèmes cardiovasculaires. Le consortium MyHeart implique 33 partenaires de 10 pays différents, composés d'industries (Philips, Nokia, Vodafone, Medtronic), d'instituts de recherche, d'universités et d'hôpitaux. L'approche du projet MyHeart

consiste à surveiller les signes vitaux du corps avec une technologie portable dans le but de donner des recommandations (thérapeutiques) au patient. Basé sur ces données, MyHeart propose soit une rétroaction locale directe à l'utilisateur, lui permettant d'ajuster son contexte, soit une aide distancielle, fournie par un professionnel (médecin ou infirmier).

L'élément clé de cette approche tient en une boucle fermée, qui non seulement surveille les patients, mais fournit également des solutions pour l'utilisateur. Pour cela, le système MyHeart comprend un élément télémédical (serveur distant), d'où un professionnel peut accéder aux données et contacter le patient.



FIGURE 2.2 – Exemple de textiles connectés [77]

Une des spécificités remarquables dans le cas de Myheart est l'intégration d'entreprises textiles italiennes et d'autres partenaires pour les modules électroniques dans le but de tirer profit des textiles connectés (Figure 2.2).

Ce projet est le socle de nombreux axes de recherches gravitant directement ou indirectement autour de la mise en place de l'outil de surveillance [170, 75, 169, 79].

**CAALYX** CAALYX [7, 150], est un système qui permet à une équipe médicale d'observer les signes vitaux d'un patient à distance et de détecter des schémas caractéristiques (Fig.2.3). La transmission des informations à l'équipe médicale est supervisée par un gardien qui est chargé de faire le tri dans les informations reçues afin de n'intervenir qu'en cas de besoin. Le système CAALYX met en avant deux fonctions :

- Le système peut recueillir des informations provenant de plusieurs capteurs, afin de les interpréter avant de déclencher une alarme.
- Le système fournit aux médecins et aux soignants un système de collecte d'informations issues à la fois de capteurs (ECG, un compteur SpO<sub>2</sub>, un capteur de température et un capteur de chute et de mobilité) et de questionnaires.

Un exemple d'urgence reconnu est celui de la détection de chute. Dans le cas où il y a potentiellement une chute, ou un problème pouvant être grave, l'ensemble des données et alertes produites sont reçues dans un centre de traitement, où le gardien évalue la situation. Il est alors à sa charge de déterminer la nature de la situation afin de faire intervenir, ou non, les secours. [26]

**E-Care** Le projet de télémédecine E-care a été développé pour proposer un suivi à domicile des patients en insuffisance cardiaque [8, 9]. Cette plateforme repose sur des capteurs médicaux permettant la récupération de la pression artérielle, la fréquence cardiaque, la saturation artérielle en oxygène (SaO<sub>2</sub>) et le poids du patient. La plateforme propose une tablette tactile destinée aux aidants dans le cadre de leurs tournées auprès des patients ou leurs visites. Le système E-care inclut aussi un serveur hébergeant les données des patients permettant aux divers professionnels de santé de consulter les données récupérées.

Ce projet a été déployé dès octobre 2013 dans un service de prise en charge de l'insuffisance cardiaque du CHRU (Centre Hospitalier Régional et Universitaire) de Strasbourg [8].

### 2.1.2 Conclusion

La télémédecine et plus particulièrement la télévigilance répond à un besoin central des personnes âgées qui est de rester le plus longtemps possible chez eux [20].

Dans ce cadre, différents projets mettent en avant les nouvelles technologies comme support de suivi à domicile. Ces propositions permettent ainsi aux per-



système distant est généralement composé d'une aide à la décision et d'une équipe de professionnels en charge de la prise de décision.

Ces deux sous-systèmes agissent ainsi de concert afin de proposer un suivi limité à un ensemble de pathologies. Ce choix présente l'avantage de faciliter la gestion et le traitement des informations. Chaque type de donnée récoltée est prévue et leurs significations peuvent être déterminées en amont par des comités d'expert du domaine.

Les tâches ainsi accomplies peuvent être décrites au sein d'un cycle dans lequel :

1. des données sont récoltées par le biais de dispositifs de monitoring (capteurs portés par la personne ou disséminés dans son environnement),
2. ces données sont ensuite envoyées sur une plateforme distante via des technologies de télécommunication où un traitement des données peut être réalisé afin de faciliter l'analyse des informations,
3. un opérateur gère les cas et effectue des mesures en conséquence, puis déclenche si nécessaire les actions appropriées.

Une telle plateforme permet ainsi à la fois : pour le patient d'être suivi, pour les aidants proches de se montrer attentifs sans forcément être présents au sein de l'habitat et pour les professionnels de mutualiser les suivis, à distance.

Cependant, dans le contexte de cette thèse, nous mettons en avant le fait que les personnes en perte d'autonomie sont multi-pathologiques [124] et qu'il est nécessaire de proposer un suivi évoluant avec leurs besoins. En l'état, la production d'un tel suivi peut ainsi rapidement se transformer en un patchwork de solutions disparates. Cette particularité limite donc l'évolutivité et la modularité du suivi, ce qui entre en contradiction avec nos exigences [C2] et [C7] qui est de s'adapter à l'évolution du contexte et aux besoins de chacun. À cela s'ajoute le fait que la présence d'un système fermé et la présence d'un opérateur distant ne correspondent pas non plus à nos exigences [C5] et [C6].

Au terme de cette section, il est à noter que nous nous sommes volontairement concentrés sur les architectures et les outils logiciels couramment utilisés dans le cadre d'un service de télé-vigilance. Cela afin d'en cerner leurs avantages,

mais aussi leurs limites par rapport à l'exigence de notre proposition concernant la nature évolutive des besoins des personnes suivies.

Dans ce contexte, les outils technologiques nécessaires à leur mise en place et leur caractéristique ont été volontairement éludés et seront développés dans la prochaine section.

## 2.2 Technologies

Interagir avec le monde réel est nécessaire au bon fonctionnement d'une plateforme informatisée [5]. Au sein de cette section, nous nous intéresserons donc à ces diverses technologies permettant la mise en place des outils d'action et de captation qui parsèment nos environnements informatisés (types de capteur, méthodes de mise en réseaux, middlewares).

### 2.2.1 Les capteurs et effecteurs

Dans le cadre d'un système autonome, la principale source d'information du système sur le monde réel est celle produite par les capteurs disséminés dans l'environnement. Un capteur est un élément capable de détecter un phénomène physique (niveau de température, de luminosité, etc.) et de le traduire en une information intelligible qui est à destination d'une partie commande (ordinateur, carte électronique programmable, etc.). L'acquisition de données est ainsi possible via le déploiement de différents types de capteurs dont l'utilisation dépend de ce que l'on souhaite capter (Tableau 2.1).

Dans le cadre d'un suivi, l'environnement est donc équipé de capteurs permettant, après analyse, de produire un flux d'information liée à des habitudes de vie.

Toutefois, pour des raisons de confidentialité, l'anonymisation peut être une caractéristique voulue au sein de certaines solutions [39].

Ainsi, il est courant de privilégier des capteurs dont les données brutes ne permettent pas directement d'identifier les personnes concernées telles que les capteurs de rythme cardiaque ou de pression. A contrario, les technologies di-

Type de capteurs	Détection
Microphone	Son/Volume sonore/souffle
Piézoélectrique	Luminosité/mouvement/passage
Potentiomètre	Distance/mouvement précis/pression
Capteurs physio.	Electro-encéphalogramme/Electro-cardiogramme
Accéléromètre	Mouvement
Capteurs météo.	Humidité/température/pression atmosphérique

TABLEAU 2.1 – Différents capteurs et leurs type de mesure

rectement "nominatives", telles que les caméras, ont tendance à provoquer des rejets. Pour identifier les personnes, l'usage tend à se diriger vers des technologies telles que les marqueurs de types RFID (Tableau 2.2) [179], aussi bien pour identifier des personnes que pour le suivi de biens personnels [167].

Dans un cadre comme dans l'autre, il reste cependant difficile de déterminer si une technologie est intrusive ou non. En effet, même issus de capteurs produisant des données personnelles anonymisées, il est quand même possible d'identifier les personnes. Pour l'illustrer, nous pouvons prendre l'exemple d'une ré-identification des dossiers médicaux des personnes qui a été rendu possible grâce à l'usage fait d'une version électronique d'une liste électorale (54 805 électeurs) [166]. Les retours montrent ainsi que 97 % étaient identifiables avec seulement le code postal complet et la date de naissance [166]. À la vue de ces résultats et la quantité de données générées par les systèmes de suivi, il est a priori difficile de déterminer si des données stockées et gérées par un serveur peuvent représenter une source de dérives potentielles. Cette problématique est d'ailleurs le coeur de toute une branche de la recherche dont le but est d'assurer l'anonymisation des données [87, 156, 192].

Une fois ces informations captées, il reste à déterminer la manière dont le système va produire une action au sein de l'environnement de l'individu.

Pour cela, le système aura besoin d'effecteurs. Un effecteur est un objet qui transforme l'énergie qui lui est fournie en un phénomène physique qui agit sur






Type de capteurs	Détection
	Identification des personnes, Paiement sans contact
	Identification des biens
	Identification des personnes
	Accès à des résidences, locaux et parking
	Identification

TABLEAU 2.2 – La RFID se développe sous différents supports

son environnement. On peut donner l'exemple d'une lampe qui transforme l'électricité reçue en lumière.

De la même manière que nous sommes entourés de capteurs, les effecteurs se sont démocratisés au point que leurs présences deviennent de plus en plus habituelles (Tableau 2.3).



Type d'effecteur	Objets caractéristiques	Utilisations
Lumière	Téléphone, télévision, ampoule connectée	Attire l'attention sur un emplacement
Chenillard	Électroménager	Transmettre un message simple
Écran	Électroménager, téléphone, télévision	Transmettre un message visuel
Écran tactile	Tablette, smartphone	Apporte un message complet, avec possibilité d'interaction avec l'utilisateur
Haut-parleurs	Télévision, smartphone, alarme	Attirer l'attention sur un problème détecté ou transmettre un message
Vibration	Smartphone, tablette, pda	Attirer l'attention vers l'écran de l'appareil

TABLEAU 2.3 – Liste d'éléments réutilisables présents dans l'habitat

Au travers de ces différents capteurs et effecteurs, notre environnement s'est instrumentalisé à tel point que nous pouvons de moins en moins nous soustraire à leurs présences.

Toutefois, prises indépendamment, ces technologies ne peuvent fournir les services que nous avons l'habitude de côtoyer.

En effet, la réelle force de ces nouvelles technologies naît de l'interconnexion de cette masse d'objets miniaturisés [25, 95]. C'est d'ailleurs de cette mise en réseau au travers d'Internet qu'est née ce que l'on appelle l'Internet des Objets (IdO).

### 2.2.2 Les technologies de communication

La mise en réseaux permet aux objets connectés de faire transiter leurs données soit sur de longues portées (avec Sigfox, LoRa, des technologies cellulaires) soit sur des courtes portées (avec le Wifi, le Z-Wave, le ZigBee, ou encore le Blue-

tooth). Pour se rapprocher des usages que l'on souhaite en faire, nous allons nous focaliser sur les réseaux ayant une portée limitée à l'habitat, que nous allons diviser au travers de deux normes de communication :

- Le Body Area Network (BAN) est défini par IEEE 802.15.6 [100, 129] comme étant une norme de communication optimisée pour les appareils à basse consommation qui fonctionnent sur, dans [32, 67] ou autour du corps humain [17]. Comme capteurs utilisés, on peut citer les accéléromètres, les capteurs de pression artérielle, les biocapteurs (détection d'une modification de paramètres biologiques), ou encore les thermomètres (Figure 2.4). Une fois mise en place, cette technologie est utilisée au sein de divers domaines médicaux : [171] (exemple : glycémie [96], cancer de la bouche [99] (analyse de la salive), maladie de Parkinson [172]).



FIGURE 2.4 – Exemple de BAN

- Le Personal Area Network (PAN) est un type de réseau informatique généralement mis en œuvre dans un espace restreint comme le domicile. À l'origine les transmissions se faisait via un câble ou une clé USB, mais avec la démocratisation des technologies telles que le Bluetooth et Zigbee on parle désormais de Wireless Personal Area Network (WPAN) (basée sur IEEE 802.15 [128]).

La force de ce genre de dispositifs vient des métadonnées créées à partir de ces informations récoltées. On peut donner comme exemple la pro-

position de [190] qui scinde l'habitat en zone (Figure 2.5) puis les dotent de détecteurs de présence pour déduire les déplacements possibles de l'individu. Cela permet entre autres de préparer les pièces qui vont être visitées prochainement. Ensuite, un automate est créé pour représenter l'ensemble des salles qui sont mitoyennes. À partir de cet automate, le système pourra prévoir les déplacements de la personne.



FIGURE 2.5 – Exemple de partitionnement

Comme nous l'avons remarqué avec le PAN, le moyen le plus simple conceptuellement pour relier nos objets est de les connecter, soit par liaison filaire, soit via une liaison sans fil (Wifi, Bluetooth, LoRa, 5G, etc.) à une centrale de commande (ou concentrateur). Il devient dès lors plus facile pour le concepteur de spécifier chaque fonctionnalité du système en amont, sans aucun problème de cohérence. Cette proposition, privilégiée par les solutions commerciales, permet un contrôle total des objets connectés et des services mis à disposition. Les exemples sur le marché sont nombreux comme les ampoules connectées, les thermostats ou les systèmes d'alarme, nécessitant un contrôleur domotique. Cette démarche commerciale, malgré son côté pratique pour le client, ne va pas dans le sens de l'exigence [C3] soulevées dans ce manuscrit quant à la mutualisation des objets connectés. Dans certains cas, elle va même à l'encontre en proposant des solutions fermées, faisant barrière à la genericité, qui empêche d'avoir un accès direct aux divers composants.

Ainsi, dans le but d'améliorer la modularité du système, il est devenu courant d'ajouter un intergiciel (ou middleware en anglais) au sein de l'architecture.

### 2.2.3 Intergiciel

Dans l'usage générique, le middleware (ou intergiciel) [126, 31] est le logiciel utilisé pour la communication entre deux applications dépourvues initialement de passerelles. L'intergiciel utilise ainsi des services de base du système d'exploitation afin de fournir des fonctions d'abstraction et de communication de haut niveau. Dans un cadre ambiant, le middleware est considéré comme une couche générique qui fournit des fonctions de base pour cacher les détails de bas niveau de la couche physique [82, 16]. Cette approche est couramment utilisée dans la gestion d'environnement intelligent [19, 69, 29, 104], mais aussi dans d'autres domaines tels que la robotique [121], l'Internet des objets [44] et les réseaux de capteurs [182].

Les intergiciels sont des outils adaptés à de nombreuses problématiques liés à la transmission d'information.

Cependant, dans notre cas, une des limites rencontrées est celle de la montée en charge. Cette limitation vient du fait que son usage fait intervenir de nom-

AVD

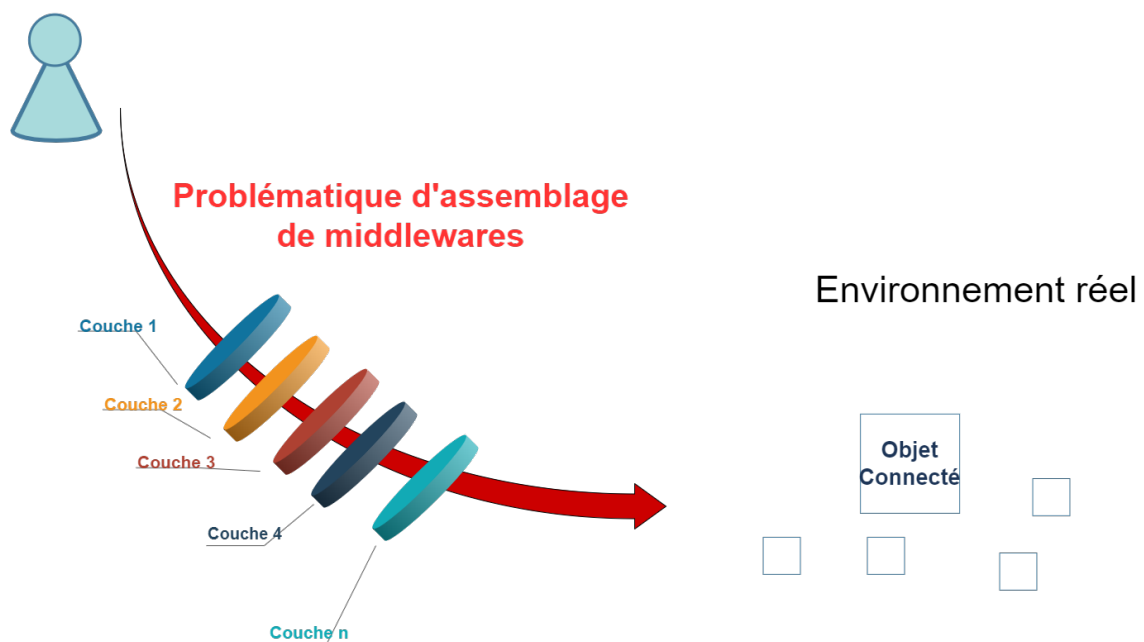


FIGURE 2.6 – Problème d'empilement de middleware

breux intermédiaires. Tout d'abord, de manière "verticale", un intergiciel se com-

plexifie à mesure où se creuse l'écart sémantique entre la nature de l'information des couches basse et celle voulue au sein des strates supérieures (Figure 2.6). Ensuite, de manière "horizontale", l'utilisation des middlewares peut être limitée par le fait qu'un système ne dispose généralement pas d'un seul middleware [97]. En effet, il est courant d'en utiliser autant qu'il y a de problèmes de communication [121]. Par conséquent, plus un système fait intervenir des entités de manière hétérogène, plus il risque de contenir un nombre important de middlewares afin d'en masquer les problèmes de communication. Dans notre cas, l'idée est de mutualiser toutes les dispositions présentes au sein de nos objets connectés, il serait difficile de proposer un middleware permettant la détection, la décomposition et la mise à disposition de l'ensemble des types d'objets existants et à venir. Des propositions comme celle de FraSCAti proposent la mise en place d'un "intergiciel d'intergiciels hétérogènes" afin de proposer une interface unique aux applications [158]. Cependant, même si ce type d'intergiciel propose un très grand nombre de fonctionnalités, il se complexifiera de plus en plus au fur et à mesure de l'apparition de nouvelles technologies. De plus, dans le cas d'une utilisation en contexte ubiquitaire, nous n'avons au préalable aucune connaissance sur les problématiques pouvant être rencontrées. Chaque appareil connecté devra alors supporter la charge totale du dispositif. Cette particularité limitera grandement son déploiement et amènera à écarter les appareils possédant de faibles capacités de mémoire et de calcul.

#### 2.2.4 Conclusion

Dans cette section, nous avons mis en avant les technologies qui servent de socles à la production de services informatisés. Que cela concerne les méthodes de captation, d'action ou de diffusion, l'informatique a fait énormément de progrès lors de ces dernières décennies. Cette évolution a permis d'intégrer et de démocratiser son usage dans nos activités du quotidien.

Dans ce cadre, l'un des problèmes principaux concerne les hétérogénéités des technologies à disposition. Cette caractéristique impose ainsi des contraintes au niveau de l'interaction avec et entre ces objets connectés.

Afin de lutter contre cette problématique, nous avons exposé le fait que l'intergiciel est l'outil, habituellement utilisé, qui fournit des fonctions génériques permettant d'abstraire la complexité de l'environnement. [16, 82]. Cependant, une limite à l'utilisation des intergiciels vient du fait que, généralement, un système a autant d'intergiciels qu'il y a de problèmes de communication. [97, 121]. Par conséquent, plus des entités hétérogènes sont impliquées dans un système, plus celui-ci peut contenir un grand nombre d'intergiciels afin de masquer les problèmes de communication. Une première piste peut être la mise en place d'un "middleware de middlewares" afin de proposer une interface unique pour l'ensemble des middlewares [158]. Cependant, une telle approche limitera tout de même l'adaptabilité du système lors de l'apparition de nouvelles technologies. Ce qui entre en contradiction à la fois avec notre exigence [C2] visant une adaptabilité aux contextes, mais aussi à notre exigence [C5] au travers de la difficulté de déploiement et de gestion de la plateforme.

Au terme de cette section, nous pouvons remarquer un élan technologique visant à produire un écosystème informatisé où l'informatique serait bien plus qu'un système d'information ou un ordinateur. Cette dynamique nous montre une tendance du domaine à s'éloigner d'une activité de traitement exclusivement centrée sur l'utilisateur vers une forme d'intelligence ambiante qui vise à intégrer naturellement les services dans les habitudes de vie de la personne.

## 2.3 Intelligence ambiante

Nous sommes à une étape de l'évolution technologique où les machines monolithiques telles que nous avons l'habitude de côtoyer tendent à disparaître, au profit d'équipements électroniques interconnectés, disséminés au sein de notre environnement. Ces changements dans nos habitudes sont attribués à l'ère de l'informatique ubiquitaire [186]. Cette ère, définie dès les années 1990 par l'informaticien Mark Weiser, est décrite comme la troisième ère de l'histoire de l'informatique, succédant aux ordinateurs personnels et aux architectures informatiques centralisées (Figure 2.7). Ce nouveau paradigme a ainsi pour but de faciliter l'accès à l'information n'importe où et n'importe quand. Dans ce type

d'environnement, l'humain est entouré par des réseaux de dispositifs dans le but de fournir des services [175, 53, 42] : sensibles au contexte, personnalisables, omniprésents, adaptatifs, réactifs et transparents.

Du point de vue technologique, ces nouveaux usages naissent d'une démocratisation d'un ensemble d'avancés dans de nombreux domaines tels que [14, 178] :

**Réseau** : L'amélioration de la couverture réseaux qui permet une meilleure interconnexion entre les outils mis en place. Ce point est de plus renforcé par la présence des réseaux sans fil de courtes (Zigbee, Bluetooth, WiFi, etc.) et longues portés (WWAN, WMAN, WLAN, etc.) [183, 193, 185]. Les services informatiques deviennent ainsi accessibles n'importe où.

**Interface** : L'interface entre l'homme et le virtuel tend à devenir "invisible" [141] en s'intégrant progressivement et naturellement dans nos habitudes de vie. Cette évolution se perçoit au travers de diverses technologies tels que les wearables computing, les interfaces tangibles [84, 85], les badges d'identification RFID, ou encore les écrans tactiles, etc.

**Hardware** : Les équipements matériels qui entrent dans la constitution de nos outils informatiques sont de plus en plus précis, nomades et efficaces. Á cela s'ajoute une diminution, jusqu'alors constante, du coût des composants, facilitant le déploiement des appareils. Nous le ressentons via l'adoption des smartphones par une grande partie de la population et l'arrivée de nouveaux appareils connectés tels que les montres, les lunettes, etc.

Dans ce cadre, et proposant une suite logique à l'informatique ubiquitaire [178], l'intelligence ambiante (ou AmI pour Ambient Intelligence) ajoute à ce contexte informatisé la notion d'intelligence (adaptatifs, sensibles au contexte, proactifs, etc.) dans le but de renforcer la capacité du système à analyser son contexte [178, 145]. Il est toutefois à noter que la notion de l'intelligence et de fait de l'intelligence artificielle n'est pas propre à la notion d'intelligence ambiante. En effet, l'intelligence est une notion transversale présente dans le cadre de la télémédecine par exemple, qui vient dans le cas de l'informatique ubiquitaire

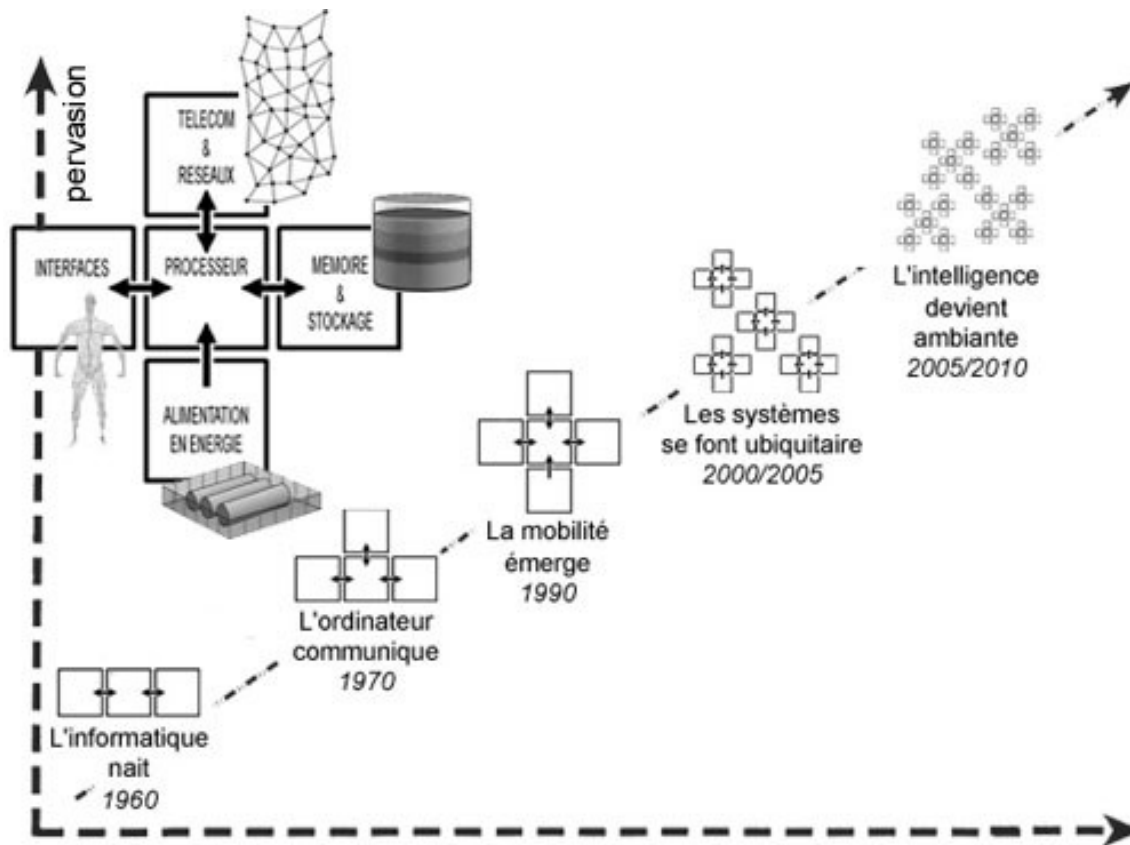


FIGURE 2.7 – Processus d'évolution de l'informatique en cinq phases [178]

apporter une dimension de "raisonnement" supplémentaire.

L'intelligence artificielle est, de plus, un domaine vaste qui fait en réalité allusion à de nombreux axes de recherche, provoquant des confusions fréquentes dans le débat public.

Il existe ainsi plusieurs voies, pouvant se mêler entre elles, tentant d'aboutir à l'intelligence artificielle. Nous allons ici traiter plus particulièrement celles concernant les systèmes experts et l'apprentissage automatique. Une troisième méthode liée à l'intelligence artificielle distribuée (IAD) fera l'objet d'une section à part entière.

Une première voie suivie dans le cadre de l'intelligence artificielle est donc celle de l'utilisation de systèmes experts en tant qu'outil capable de reproduire les mécanismes cognitifs d'un expert, dans un domaine particulier [111].

Plus précisément, un système expert est un logiciel capable de répondre à des questions, en effectuant un raisonnement à partir de faits et de règles connues.

Le principe des systèmes expert est simplement que l'expertise, acquise par



un spécialiste, est transférée d'un humain à un ordinateur sous la forme d'une base de connaissance.

La vision la plus courante du système expert se compose de trois parties [47] :

- une base de connaissance qui sert à rassembler l'expertise d'un domaine;
- une base de fait du problème à résoudre, où sont stockées les hypothèses de départ et les nouveaux faits produits par déduction;
- un moteur d'inférence qui est un ensemble de programmes capable de produire un raisonnement "humain".

Le but étant de permettre au moteur d'inférence de s'appuyer sur la base de règles afin de produire par déduction logique une conclusion précise. Ces conclusions sont alors semblables à celles provenant des consultants humains [13].

Au sein de notre cadre d'application, il est difficile de proposer un système expert se chargeant à la fois de la décomposition/recomposition des services tout en supportant la charge des services en eux-mêmes. En effet, l'expertise requise, qui doit être fournie par un spécialiste, est une ressource qui est difficile à récupérer à la vue de la transversalité des besoins rencontrés (médicales ou technologiques).

Sans cette dépendance à la collecte d'une expertise, la tendance dans le domaine de l'intelligence artificielle se dirige vers les voies liées à l'apprentissage automatique (machine learning) [144]. Le Machine Learning est une méthode d'apprentissage statistique dont le but est de reproduire une relation entre des éléments décrits dans une base de données  $X$  et des prédictions  $Y$  grâce à une fonction  $f$ . Il existe alors différents types d'algorithmes adaptés à la tâche à accomplir et aux types de données rencontrées (Régression linéaire, Régression logistique, Arbre de décision, Machines à vecteurs de support, K-plus proches voisins, etc.). Parmi les méthodes d'apprentissage automatique, une attention toute particulière a été donnée récemment à l'apprentissage profond (deep learning), plus adapté aux volumes de données non structurés actuellement générés (par exemple : images, son, texte) [181, 114]. L'engouement autour de ce concept est tel, qu'il existe à ce jour une confusion entre les termes d'intelligence artificielle et de deep learning, alors que ces notions ne sont pas équivalentes, mais

sont imbriquées (Figure 2.8) [36].

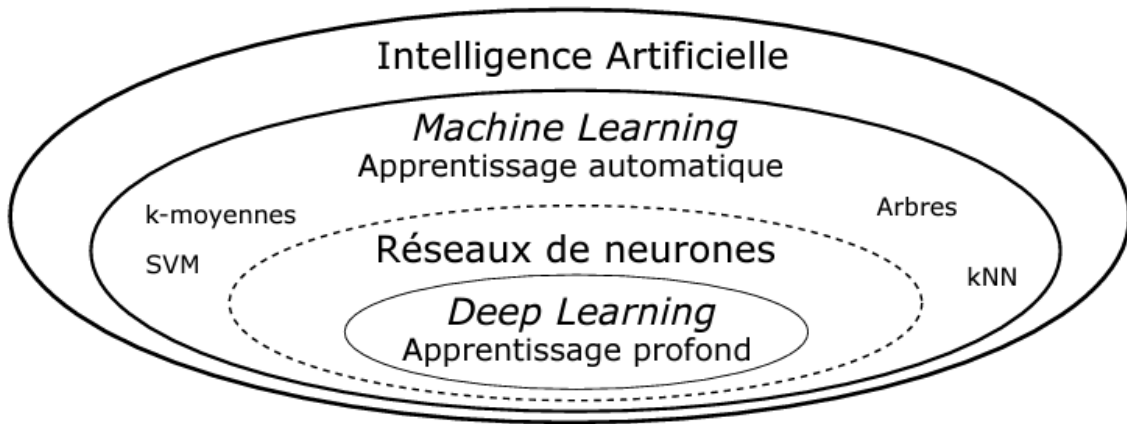


FIGURE 2.8 – Représentation schématique des principaux domaines de l'apprentissage automatique [36].

L'intelligence ambiante définit alors un environnement numérique capable de raisonner afin d'accompagner des personnes dans leur quotidien [41].

### 2.3.1 Des plateformes existantes

Du point de vue de son usage, l'intelligence ambiante a fait ses preuves, dès les années 1990, au sein de plusieurs domaines tels que :

- La redirection d'appel (premier usage datant de 1992) [184]
- La pièce intelligente [35]
- Les salles de classe augmentées [191]
- La gestion de musée [63]
- Le suivi environnemental [12]
- La mobilité des personnes aveugles dans les transports [18]
- l'Ambient Assisted Living (AAL) [28, 123].

Parmi les usages cités, nous allons nous intéresser plus particulièrement au domaine de l'Ambient Assisted Living (AAL) [28, 123].

Ce domaine se concentre sur la création d'une intelligence ambiante, dont le but est d'améliorer la qualité de vie des personnes, souvent âgées, et de leurs soignants [39, 123, 139].

Dans ce cadre, nous détaillerons ici que des plateformes en lien avec les AAL.

**COOK-Assistant Culinaire** L'orthèse cognitive pour la cuisine COOK (**C**ognitive **O**rthosis for **co**OKing) est une technologie d'assistance cognitive ambiante visant à fournir un soutien cognitif aux personnes souffrant de lésions cérébrales traumatiques graves (TBI) pour préparer des repas de manière indépendante [137]. Produite par le laboratoire DOMUS, COOK est une application web installée sur une tablette ou un ordinateur avec un écran tactile adossé à une cuisinière connectée équipée [73] :

- de capteurs de puissance afin de détecter quel brûleur est allumé
- de capteurs infrarouges qui détectent une chaleur anormale
- d'un capteur électromagnétique qui détecte l'ouverture de la porte du four
- de capteurs de température qui détectent les objets placés sur un brûleur

COOK est également connecté à un environnement intelligent composé de capteurs infrarouges (détection de la présence humaine), de capteurs électromagnétiques (ouverture des portes) et de débitmètres (détection de l'utilisation de l'eau).

**TRICARE** Le projet TRICARE [132] est un projet initié par le CETIC et piloté par la société AnB, spécialisée dans la conception et fabrication de systèmes d'alarme. Ce projet a pour vocation de superviser l'activité des personnes âgées, dans le respect de leur vie privée, via des systèmes non-intrusifs basés sur l'utilisation de capteurs basse fidélité (incertitudes de mesures d'un capteur). Toute la difficulté se trouve par conséquent dans le traitement de ce genre de signal. En contrepartie, il est possible de disposer de capteurs discrets, non-invasifs et peu coûteux.

Le but de ce projet se concentre nettement sur la proposition d'une solution peu coûteuse dont le coeur de recherche est liée à la production d'un composant générique utilisant les techniques de machine-learning pour identifier des activités à partir des événements notifiés par ces capteurs de basses fidélités et en fonction de l'historique.

**HeartFaid** HeartFaid [38] est un projet produit dans le cadre du suivi des problèmes cardiaques. Dans chacun des environnements, des dispositifs médicaux et des capteurs sont disséminés. Dans l'environnement clinique et à domicile, les données sont recueillies par des PC, tandis que dans l'environnement mobile, l'appareil d'acquisition de données est un appareil mobile porté par l'utilisateur. En local, un intergiciel permet la communication entre tous les appareils connectés médicaux.

Le système a l'avantage de proposer un système d'aide à la décision basé sur une ontologie. L'algorithme du système de décision produit des indications médicales en se basant sur des protocoles préétablis et accède à la base de connaissance à chaque alarme afin de suggérer des solutions automatiquement aux soignants selon la gravité des cas.

**MavHome** L'objectif du projet MavHome (**M**anaging **A**n Intelligent **V**ersatile **H**ome) est de créer une maison qui agit comme une entité intelligente.

Cette entité perçoit l'état de la maison grâce à des capteurs et agit sur l'environnement grâce à des contrôleurs de dispositifs [45]. L'objectif est alors de maximiser le confort et la productivité de ses habitants en minimisant le coût de fonctionnement.

Dans ce cadre, l'architecture de MavHome met à profit plusieurs algorithmes afin de prédire les activités possibles (SHIP, ALZ, TMM). MavHome, permet ainsi la détection de séquences régulières (Chauffage activé, alarme activée, alarme désactivée, éclairage de la chambre, cafetière activée, éclairage de la salle de bain, vidéo de la salle de bain activée, douche activée, chauffage désactivé, arrosage activé).

**KopAl** KopAl [66] est un système d'aide à domicile pour les patients atteints de démence dans le but de produire un suivi de ses caractéristiques telles que la désorientation.

En ce sens, la plateforme propose une aide à l'orientation, des rappels de rendez-vous et enfin des détections d'urgences (sortie du périmètre, oubliée, etc.) Le système KopAL comprend :

- Des assistants mobiles inter-connectés via des connexions ad hoc WiFi
- Des composants qui contiennent les fonctionnalités (protocole de routage, gestion de rendez-vous, interprétation de la parole ,etc.)
- Un serveur central qui gère les agendas
- Un ensemble de routeurs fixes (WLAN) qui agissent comme des stations relais et des repères

La proposition met ainsi en place d'une part un suivi discret utile en cas d'urgences et d'autre part propose un dispositif d'assistance tactile, permettant l'édition et la création de rendez-vous.

### 2.3.2 Conclusion

Le domaine de l'intelligence ambiante est un domaine scientifique qui a pour but de fournir des services, fondés sur les nouvelles technologies, capables de répondre de manière adaptée à différents secteurs d'activités. Les approches basées sur l'intelligence ambiante tendent ainsi à aller plus loin que les approches traditionnelles en cherchant notamment à automatiser la compréhension des risques tout en se fondant dans les habitudes de vie des personnes concernées.

L'environnement n'est ainsi plus parsemé d'interfaces, mais devient elle-même l'interface d'interaction avec le virtuel.

L'engouement autour de ce cadre de recherche permet ainsi l'apparition de travaux innovants, intelligents et de plus en plus diffus.

En ce qui concerne l'intelligence derrière ces propositions, nous avons traité deux approches liées à l'intelligence artificielle : les systèmes experts et l'apprentissage automatique.

Toutefois, au sein d'un environnement dans lequel la nature même des moyens de perception et d'action sur l'environnement est un défi, l'usage de ces méthodes ne semble pas approprié. De plus, la présence d'un environnement distribué complexifie la production d'un résolveur de problème unique et centralisé dans un seul lieu de traitement, qui demanderait une puissance de calcul difficilement disponible.

Dans ce cas, il serait judicieux de proposer une solution prenant en compte

la nature distribuée du système. Ainsi, à la différence de l'intelligence artificielle classique qui modélise le comportement d'une seule entité, il est possible de prospecter au niveau de l'intelligence artificielle distribuée (IAD) et plus particulièrement des Systèmes Multi-Agents (SMA) qui s'intéressent à la modélisation de comportement intelligent par la coopération entre un ensemble d'agents [102].

## 2.4 Les systèmes multi-agents

La notion d'agent, et par extension des systèmes multi-agents (SMA), émerge d'un abandon progressif des conceptions de la programmation axées sur les machines au profit de concepts qui s'inspirent de la manière dont nous comprenons le monde et le vivant [187]. Cette approche va ainsi permettre l'apparition d'un domaine axé sur une forme de décentralisation de l'outil informatique. Il est ainsi possible d'établir des liens de similarité entre les contextes et les buts ayant permis l'apparition de la notion d'agent [187] mais aussi de la notion de l'intelligence ambiante. On peut citer par exemple :

**Ubiquité** L'ubiquité fait surtout référence à l'introduction d'une puissance de traitement dans des dispositifs de plus en plus miniaturisés. Ce fait est dû à la réduction, jusqu'alors continue, du coût des composants couplée à une constante augmentation des capacités de calculs. Ce qui permet la proposition d'une informatique omniprésente dans notre environnement et non localisable par l'utilisateur, tout en imposant une contrainte en termes de décentralisation.

**Interconnexion** Alors que les premiers systèmes informatiques étaient des entités isolées qui communiquaient uniquement avec leurs opérateurs humains, les systèmes informatiques actuels sont généralement interconnectés. L'exemple le plus concret est celui de l'Internet Des Objets (IDO) qui est définie par la production de services basée sur des objets interconnectés grâce aux technologies de l'information et de la communication.

**Intelligence** La troisième tendance est celle de systèmes toujours plus in-

telligents. La complexité des tâches que nous sommes capables d'automatiser et de déléguer aux ordinateurs n'a cessé de croître au cours de ces dernières décennies. Ici aussi, la décentralisation a poussé à trouver des solutions adaptées au contexte.

Au coeur de ce contexte, l'agent vient ainsi naturellement, en suivant une tendance à une délégation toujours plus grande [187], répondre à de nombreuses problématiques rencontrées au sein du domaine de l'intelligence ambiante.

Dans cette section, nous allons dans un premier temps décrire le concept d'agent et les architectures couramment utilisées. À partir de ces définitions, nous traiterons de la capacité des agents à opérer en collectivité dans des systèmes multi-agent. Finalement, après avoir présenté des plateformes utilisant les agents, nous concluons sur sa pertinence dans le cadre de notre approche.

### 2.4.1 L'agent

L'agent est une "entité computationnelle" capable de capter de l'information et d'agir dans le but de déclencher une action. Il n'est d'ailleurs pas possible de définir davantage et de manière unanime la notion d'agent tellement il existe de définitions de ce concept.

Parmi les définitions les plus courantes, nous pouvons citer :

- " Un agent est une entité qui perçoit son environnement et agit sur celui-ci " [154]
- Un agent est un système informatique qui est situé dans un environnement et qui est capable d'agir de manière autonome dans cet environnement afin d'atteindre ses objectifs de conception [189].
- " Un agent est une entité qui fonctionne continuellement et de manière autonome dans un environnement où d'autres processus se déroulent et d'autres agents existent. " [160].
- Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances

et des interactions avec les autres agents [57, 59].

Bien qu'il ne se soit pas possible de proposer une définition générale, il est possible d'identifier des caractéristiques que l'on attribue fréquemment à l'agent telles que le fait d'être :

- Situé – Capacité d'agir sur son environnement à partir des entrées sensorielles reçues de ce même environnement;
- Autonome – Possibilité d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne;
- Proactif – Présence d'un comportement proactif et opportuniste
- Diligent Production d'une réponse dans le temps requis;
- Social – Interaction avec des autres agents (logiciels ou humains) permettant par exemple de planifier ses actions par coordination et négociation

Ces propriétés représentent ainsi un socle de propriétés, non absolu, partagées entre les agents.

De plus, cette multiplicité dans la "forme" de l'agent ne s'arrête pas là.

En effet, les agents vont aussi se différencier au niveau de leurs architectures, couramment divisées en deux catégories : réactive ou cognitive [56, 34].

**Les agents réactifs** Les agents réactifs sont de simples entités élémentaires [59], qualifiées de non-"intelligents" à cause de leur peu de capacité de raisonnement et de mémorisation, leur incapacité à produire une vision symbolique de leurs environnements ainsi que leur protocole de communication réduit.

Leurs comportements peuvent être résumés par un cycle alternant entre stimulus perçus de l'environnement et réponses de l'agent.

En s'appuyant sur la simplicité des calculs produits, le but d'un SMA constitué d'agents réactifs est généralement de proposer un grand nombre d'agents afin de faire émerger un comportement global. L'approche réactive apporte plusieurs avantages.

Tout d'abord, vu son usage de collection d'agent simple, elle apporte une solution robuste basée sur une forte redondance des entités du système et l'absence de contrôle central. De plus, leurs économies du point de vue cognitif



permettent aux approches réactives de proposer une meilleure réactivité du système en proposant un fort couplages entre les perceptions et les actions.

Pourtant, ces agents présentent plusieurs limitations, qui peuvent nuire à leurs performances. Tout d'abord, ces agents n'ont qu'une vision "instantanée" de la résolution d'un problème. L'agent ne peut ainsi décider s'il faut attendre afin d'espérer une meilleure opportunité d'action à exécuter. D'autre part, les agents réactifs ne maintiennent pas une représentation de l'environnement. Il est donc difficile pour ces agents de percevoir le meilleur moyen possible à leur disposition afin de produire une action. Ces interactions indésirables peuvent ainsi parasiter l'émergence de l'intelligence.

**Les agents cognitifs** Les agents cognitifs sont considérés comme plus complexes. Ces agents se construisent une représentation de l'environnement, possèdent une base de connaissances et sont aptes à mener un raisonnement. Ils peuvent ainsi tenir compte de différents paramètres tels que leurs passés, leurs environnements ou encore de l'état des autres agents afin de planifier leurs actions. Un paradigme classique pour formaliser le comportement des agents cognitifs est le paradigme Belief-Desire-Intention (BDI) [34, 143] décrit comme suit :

**Belief (Croyance)** Les croyances d'un agent sont les informations que l'agent possède sur l'environnement et sur d'autres agents au sein d'un même environnement. Les croyances peuvent être incorrectes, incomplètes ou incertaines et peuvent changer dans le temps.

**Desire (Désir)** Les désirs d'un agent représentent les changements d'états que l'agent aimerait voir réaliser, que cela soit d'un point de vue interne ou externe (environnement). Dans le cas de cooccurrence de désirs contradictoires l'agent doit choisir parmi ses désirs un sous-ensemble qui soit consistant.

**Intention (Intention)** Les intentions représentent l'état délibératif de l'agent et peuvent être décrits comme les actions que l'agent a décidé de faire pour accomplir ses désirs. Cette tâche est mise en place sous la forme séquences d'actions, pouvant inclure d'autres sous séquences, qu'un agent peut décider d'effectuer.

Comblant les "lacunes" de représentation que peut avoir l'agent réactif, l'agent cognitif est donc une entité plus polyvalente, capable d'une plus grande adaptation aux contextes.

Toutefois, cette "intelligence" est aussi la cause primaire de ses principales limites qui sont son empreinte mémoire et sa demande de puissance de calcul, bien supérieures à celles nécessaires aux agents réactifs.

### 2.4.2 Le système multi-agents

Un Système Multi-Agent (SMA) est un système composé d'un ensemble d'agents, autonomes, situés dans un certain environnement et interagissant entre eux. On compare souvent cette démarche à l'organisation des fourmis qui arrivent à effectuer des tâches "complexes" en se coordonnant [50].

Dans ce cadre, un système multi agent [59, 58, 80] est composé d' :

- un ensemble d'agents pouvant interagir avec d'autres agents ou avec des objets. Il existe plusieurs schémas d'interactions possibles :
  - collaboration : les tâches sont alors réparties entre les agents;
  - coopération : plusieurs agents partagent un même but;
  - coordination : gestion des dépendances entre les actions des agents;
  - compétition : chaque agent poursuit son but.
- un ensemble d'objets passif pouvant être modifié par les agents
- un environnement dont les agents pourront avoir leur représentation
- un ensemble de règles représentant les interactions possibles entre les divers éléments du système.

La force des SMA est de permettre la distribution de l'intelligence ou de l'expertise entre les sources de connaissances dans l'optique d'un but commun. Dès lors, le domaine des SMA propose des solutions à de nombreux champs d'application.

Parmi les modèles existants, nous allons exploiter dans cette thèse le méta-modèle Agent & Artifact (A&A) [130]. À la vue de nos exigences, cette approche est la plus propice à la mise en place de notre proposition. En effet, grâce à sa notion d'encapsulation des ressources au sein d'une composante virtuelle, elle

représente une opportunité en ce qui concerne la modélisation et la représentation des fonctionnalités primaires des objets. On peut donner un exemple de son utilisation au travers du framework Cartago [147] au sein de JaCaMo[22] pour la programmation de systèmes multi-agents opérant dans des environnements distribués [151]. Le méta-modèle A&A (Agent & Artifact) [130] se caractérise par trois principales abstractions :

**Les artefacts** Les artefacts A&A sont des composants passifs des systèmes, tels que les ressources et les médias, voués à être utilisés par des agents pour soutenir leurs comportements. Le concept d'artefact est conçu pour s'adapter à n'importe quelle situation, ainsi, les artefacts sont modélisés sans but précis. C'est en effet au moment de leur conception que chaque artefact est lié à ses fonctions par le concepteur. Enfin, de manière explicite ou implicite, un artefact expose son interface (à la manière d'une API), afin de mettre ses fonctions à la disposition des agents.

Les artefacts peuvent être équipés d'un manuel d'utilisation qui décrivent le comportement qu'un agent doit suivre pour interagir de manière significative avec l'artefact. Les instructions d'utilisation sont écrites dans un langage que l'agent peut comprendre au travers d'une sémantique qui définit précisément les séquences possibles d'actions de l'agent sur l'artefact [130]. Ce concept a conduit à la proposition de différents types d'artefacts, tels que l'artefact de coordination [131], l'artefact de réputation [81] ou encore l'artefact cognitif [138].

**Les agents** Les agents représentent les composants proactifs du système, encapsulant l'exécution des activités dans un environnement donné. À la différence des artefacts, les agents ne fournissent aucune interface pour être utilisé. Ainsi, seules les données (informations, connaissances) parviennent à l'agent par le biais d'une communication ou d'un environnement interposé. L'agent interagit avec les artefacts au moyen d'une action d'usage, où l'agent sélectionne un contrôle de l'interface d'utilisation, en spécifiant les paramètres éventuellement requis.

**Les workspaces** Les workspaces sont des conteneurs conceptuels d'agents et d'artefacts, utiles pour définir les topologies de l'environnement et les notions de localité. Cette structure est proposée dans le but de mettre en place une portée pour les interactions et l'observabilité des événements, ainsi que pour l'ensemble des activités connexes menées par un groupe d'agents. Un SMA complexe peut alors être organisé comme un ensemble de workspaces, répartis entre plusieurs nœuds du réseau. Les agents peuvent rejoindre plusieurs workspaces simultanément.

### 2.4.3 Les usages

Les SMA proposent une alternative à la résolution centralisée des problèmes simulés ou en temps réel. Soit parce que les problèmes sont eux-mêmes distribués, soit parce que la distribution de la résolution des problèmes entre différents agents se révèle être une manière plus efficace d'organiser la résolution des problèmes [59]. De nombreuses propositions mettent ainsi en valeur ces propriétés, aussi bien en simulation qu'en environnement réel.

#### Simulation

Les simulations faisant intervenir des systèmes multi-agents ont vocation à être des modèles basés sur les entités. Ces simulations peuvent être utilisées pour tester des hypothèses ou construire des théories [51]. Cette méthode produit ainsi des apports dans plusieurs domaines [78] :

- Prédiction (ventes des entreprises ex : [93])
- Faisabilité (ex : logistique urbaine [60])
- Découverte (ex : phénomènes sociétaux [54])
- Explication : Expliquer les processus qui produisent les comportements. (ex : comportemental [118]),
- Critique : proposer d'autres explications théoriques aux phénomènes proposés par les chercheurs,
- Prescription : suggérer de meilleurs modes de fonctionnement ou de meilleures méthodes d'organisation.

## Temps réel

En ce qui concerne le temps réel, il existe deux principaux cadres de recherche liées aux SMA [59] : la robotique [133, 146] et l'intelligence ambiante. Au sein de ce manuscrit, nous allons naturellement nous intéresser aux usages liés à l'amélioration du cadre de vie des personnes en perte d'autonomie (AAL).

**SAPHIRE** Le projet SAPHIRE met en place un système multi-agents pour la surveillance des maladies chroniques [105, 103]. Cette solution peut être mise en œuvre aussi bien dans un hôpital que dans les environnements domestiques. En se basant sur des informations de santé récoltées et enregistrées, SAPHIRE mettra en route un système d'aide à la décision responsable du déploiement et de l'exécution des recommandations cliniques. Cet outil est basé sur une archi-

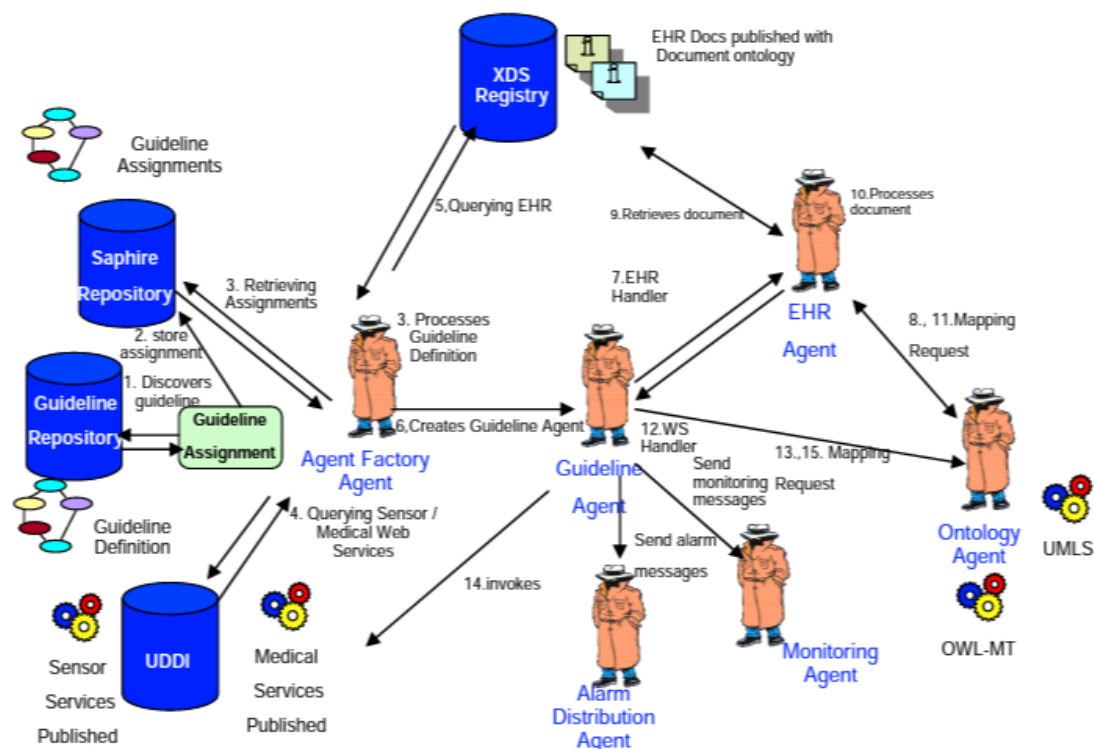


FIGURE 2.9 – Architecture multi-agents de SAPHIRE [105, 103]

tecture multi-agents divisée en (Figure 2.9) :

- Agent Factory Agent : Le Factory Agent est principalement responsable de la définition des recommandations à donner à un patient, et de la création de l'agent guideline qui exécutera ces recommandations cliniques.
- Agent EHR : L'agent EHR est une passerelle permettant d'accéder aux données cliniques des dossiers médicaux électroniques du patient et de les extraire.
- Agent ontologique : Au sein de l'architecture SAPHIRE, les recommandations et les références aux données du patient sont modélisées dans un modèle d'information de référence. L'agent ontologique se base ainsi sur cette modélisation afin de résoudre des problèmes d'interopérabilité sémantique.
- Agent guideline : L'agent de directives est l'entité principale qui exécute les recommandations cliniques.
- Agent de surveillance : L'agent de surveillance fournit une interface aux praticiens cliniques pour visualiser l'exécution des tâches.
- Agent de distribution d'alarmes : Les messages d'alarme et les rôles auxquels les messages sont attribués sont transmis à l'agent de distribution d'alarme, dans le but de distribuer ces messages aux destinataires nécessaires de la manière la plus efficace et la plus fiable.

Le dispositif a été testé en 2011 pour le suivi des personnes souffrant de maladies cardiovasculaires dans les urgences de l'hôpital de Bucarest ou encore à domicile en Allemagne dans le cadre d'une démonstration.

**TeleCARE** développés par le consortium TeleCARE Cette plateforme [29], basée sur le paradigme multi-agents a pour but d'améliorer la qualité de vie des personnes âgées et de leurs familles. Divers services sont assurés tels que : une surveillance du statut de vie, un système de rappels des activités quotidiennes, un mécanisme communautaire collaboratif, etc.

Pour cela, le Care Center produit des agents (basé sur les aglets de java [106]) spécialisés (agents mobiles) puis les envoie au niveau du système local pour donner des ordres de collecte et d'envoi d'informations aux différents agents stationnaires qui s'occupe de la collecte des données (Figure. 2.10). Ces agents mo-

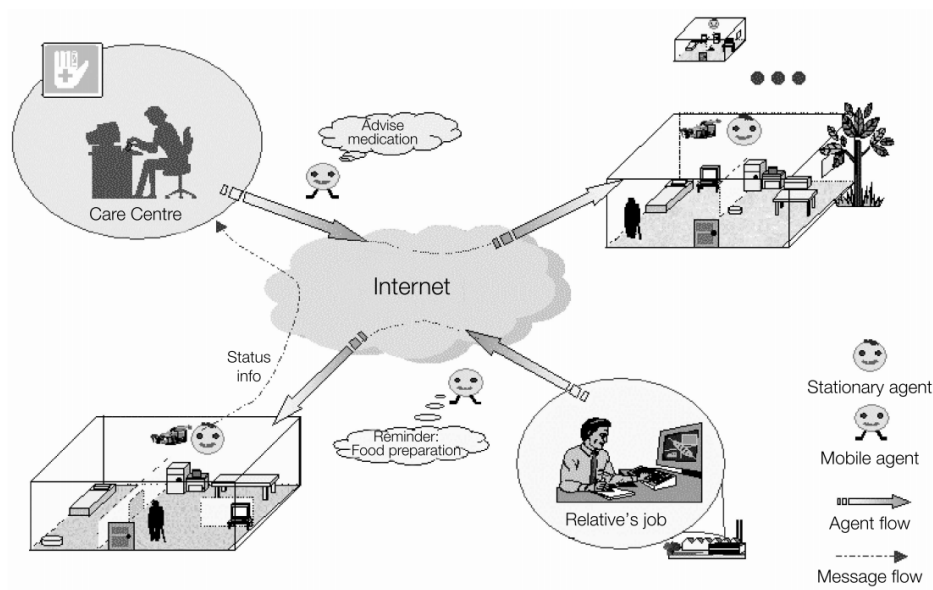


FIGURE 2.10 – Le principe de TeleCare [29]

biles sont par conséquent soumis à la demande d'un spécialiste de la santé. Par ailleurs, il est possible de rendre les agents plus autonomes en utilisant les interpréteurs de JESS et Prologue inclus dans les agents.

L'architecture de TeleCare se décompose en trois grandes parties : la première concerne les communications et les abstractions des appareils présents ; la deuxième est le cœur du service vu qu'elle est dédiée au fonctionnement du système multi-agents . Cela concerne autant la gestion des agents que les diverses ressources nécessaires à leurs fonctionnements ; enfin, la dernière partie est en grande partie dédiée à l'interfaçage avec le monde réel.

**HERA** Le projet HERA [161], cofinancé par le programme de recherche international " Assistance à l'autonomie à domicile", propose des services d'aide pour les personnes âgées souffrant de troubles cognitifs légers, Alzheimer légers/modérés, ou d'autres maladies (diabète, cardiovasculaire). HERA propose ainsi des jeux de renforcement cognitif, un système de communication passive (liée à la compréhension de l'utilisateur sur ces propres pathologies), des rappels (pilules, exercices, etc.), ainsi qu'une surveillance des constantes (tension artérielle et poids). Le module SMA est chargé d'apprendre les habitudes afin de personnaliser les services identifiés lors de la phase d'analyse des besoins. Le

système est composé de deux types d'agent :

- Les agents "Personal Assistant" ont un rôle central au sein de la plateforme. Chaque agent sert un utilisateur enregistré, stocke et gère son profil et ses données personnelles et apprend via l'historique des demandes afin d'adapter ses services à ses habitudes.
- L'agent Interface dont le but est de gérer les interactions avec les autres modules. Cet agent fait le pont entre les demandes de services et les agents en charge des services.

**OutCare** OutCare propose un système d'aide adapté aux personnes souffrant de problèmes cognitifs, plus précisément de la maladie d'Alzheimer. [180]. Cette proposition met en avant deux aspects principaux :

- Aide au cours de la vie quotidienne via la présentation de déclencheurs de mémoire simples afin de stimuler le patient au cours de tâches de routine simples ;
- Mise en place d'un outil de vigilance proposant la surveillance et l'apprentissage des routines quotidiennes des patients. En cas d'une détection d'écarts significatifs de comportements, une alerte est produite (soignants, parents, centres médicaux) par SMS, e-mail et voix.

L'architecture de Outcare peut être résumée en trois composants :

- Un composant lié au patient. Ce composant est installé sur un smartphone afin d'exploiter le capteur gps.
- Un composant lié au soignant permettant de localiser et renseigner l'état des patients. Le soignant peut aussi recevoir des alertes en cas de détection d'activités irrégulières du patient.
- Le serveur OutCare : ce serveur permet aux soignants d'accéder à des services web liés à l'assistance des utilisateurs dans leurs activités quotidiennes.

Dans ce cadre, Outcare fait alors intervenir un ensemble d'agents chargés des différentes facettes de l'application (Agent GUI, Agent de découverte de dispositifs, Agent GPS, Agent de zone, etc.).



**ALZ-MAS** ALZ-MAS (ALZheimer multi-agents System) [68] propose un service de surveillance de santé en résidences gériatriques des patients atteint d'Alzheimer. Le système vise à utiliser des capteurs présents dans des dispositifs de surveillance (sous forme de bracelet) pour collecter les informations des patients.

ALZ-MAS met en avant son usage d'une architecture multi-agents FUSION@ (Flexible User and ServIces Oriented multi-ageNt Architecture) [40]. FUSION@ est un framework composé d'agents intelligents basés sur le modèle de croyance, désir, intention (BDI). Ce framework propose un ensemble d'agents prédéfinis qui fournissent les fonctionnalités de base de l'architecture (communications, services, sécurité, interface, gestion des agents, etc.).

Basé sur ce framework, ALZ-MAS met ainsi en place son système de suivi au travers de l'ajout d'agent :

- Agent utilisateur : Gère les données personnelles et le comportement des utilisateurs (surveillance, localisation, tâches quotidiennes et anomalies).
- Agent administratif : S'exécute sur un poste de travail et joue à la fois le rôle de sécurité qui surveille l'emplacement des utilisateurs et l'état physique du bâtiment (température, lumières, alarmes, etc.) et le rôle de gestionnaire qui gère les bases de données et l'affectation des tâches.
- Agent superutilisateur : S'exécute sur des appareils mobiles (PDA) et fait le lien avec l'agent d'administratif.
- Agent ScheduleUser : Chargé des mécanismes de planification des activités quotidiennes des utilisateurs.
- Agent de périphériques : Contrôle tous les périphériques matériels et interagit directement avec les capteurs et les actionneurs pour recevoir des informations et contrôler les services physiques.

#### 2.4.4 Conclusion

Dans un contexte d'intelligence ambiante, l'utilisation de l'agent, grâce à leur capacité à être autonome et proactif [154, 160, 56, 86, 187] est une alternative

solide aux logiciels traditionnels. De plus, leur capacité à former des structures sociales au travers d'un système multi-agents permet de produire une solution décentralisée, adaptée à l'architecture des environnements ubiquitaires.

Toutefois, au sein de ce type d'architecture, l'agent fait face à une contrainte d'adaptation particulière qui est celle liée à l'inconstance de ses propres moyens de perception et d'action.

Face à ce problème, les solutions actuelles proposent d'utiliser un ou plusieurs middlewares afin de fournir à l'agent un niveau d'abstraction suffisamment élevé pour lui permettre de mener à bien ses tâches. Au sein des propositions liées aux SMA, ce middleware prend généralement la forme d'un ensemble d'agents dont les rôles sont de proposer un support à la production des services de suivi (Figure 2.13) [68, 180, 135]. Dans une architecture SMA, l'avantage de l'agent dans le cadre d'un middleware est de produire une solution plus adaptative qu'une solution standard tout en uniformisant les entités présentes dans le système. Cependant, comme nous l'avons remarqué précédemment, une telle architecture est adaptée à des approches où le cadre d'application reste tout de même défini à l'avance.

Notre proposition vise au contraire un cadre général écoconceptif basé sur une décomposition des objets de notre environnement afin de proposer une recomposition adaptative des services.

Tout naturellement, il est de plus possible de prospecter au niveau des architectures orientées services et plus particulièrement d'une variante proche de notre proposition : les microservices.

L'approche microservice est une méthode de développement logiciel qui privilégie la combinaison de sous système autonome [49]. Chaque application est définie comme un ensemble de services faiblement couplés dans le but d'accélérer son développement et permettre une gestion plus modulaire des composants applicatifs[49].

Des analogies entre notre approche et le domaine de découverte des microservices peuvent alors être effectuées avec notamment le fait que cette approche vise à recomposer des services à partir d'entités de complexité inférieure. Ce qui est fortement en lien avec notre idée de recomposer des services à partir de l'exis-

tant.

Dans ce cadre, pouvons citer comme exemple, dans un contexte différent des AAL (Ambient Assisted Living) ou de l'écoconception, l'intergiciel FCAP (Flexible Composite APplications) qui propose un système de composition d'applications attentives [174].

À la différence des AAL, dont le but est de promouvoir un système d'assistance, une application attentive est "un support contextualisé à des utilisateurs au cours d'une activité quotidienne dans laquelle ils ont à interagir avec un système d'information et de communication" [174]. On peut donner des exemples comme la messagerie instantanée, l'écoute de la musique ou encore la recherche de données sur le web.

Dans ce cadre, FCAP propose un système multi-agents de gestion de composition faisant intervenir (Figure 2.11) :

- des agents de résolution : chargés de mettre en place une configuration fournie par l'application (agent compositeur d'application) et de découvrir des fonctionnalités au sein de l'environnement (Agent superviseur de fonctionnalité)
- des agents d'interprétation : en charge d'interagir avec les agents de résolution, de collecter les informations nécessaires à l'interprétation et de réagir aux modifications de l'environnement (exemple : interpréteur de type de fonctionnalités, interpréteur de conditions de contexte, interpréteur de correspondance des services).

Au travers de cette architecture, le système devient alors capable de combiner des microservices, grâce à un ensemble d'agents, afin de produire des services.

Cette méthode, tout à fait adaptée à la composition d'applications attentives, est toutefois inadéquate à notre proposition d'un AVD adaptatif en contexte ubiquitaire. En effet, ajouter cette couche agents de recomposition à celles issues de la découverte des fonctionnalités primaires et de la production des services ne ferait que déplacer le problème de la montée en charge lié à la composition en cascade des middlewares, vers l'ajout systématique d'agents intermédiaires liés aux impératifs définis pour notre approche (Figure 2.12).

Enfin, ce type d'architecture demande, plus généralement, une connaissance

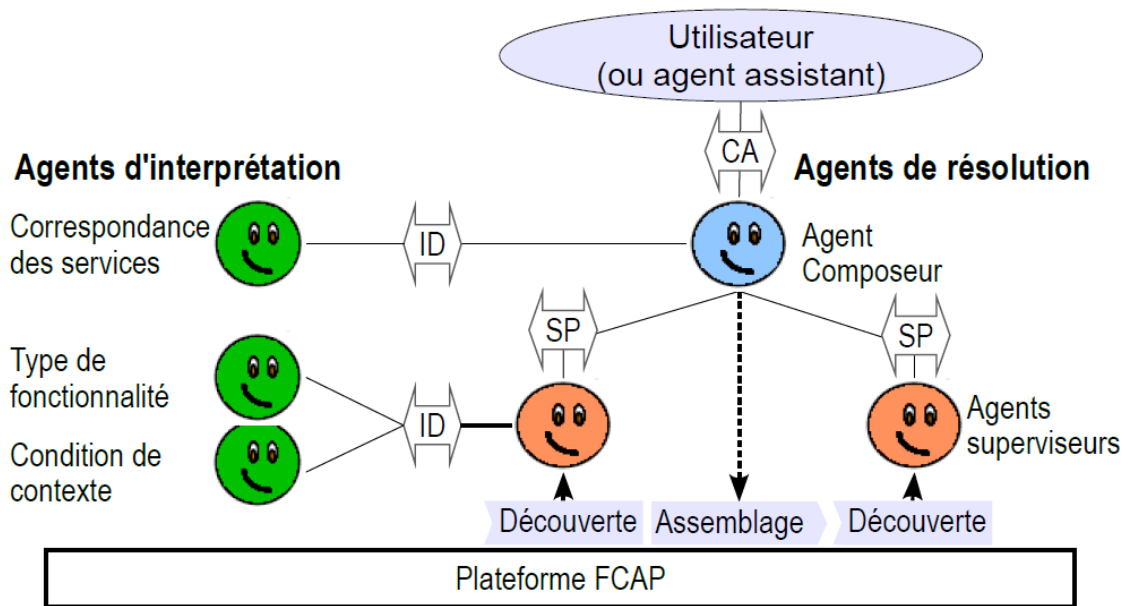


FIGURE 2.11 – Exemple d'architecture d'un système de gestion de composition [174]

a priori des services que l'on veut rendre et de celles qui seront potentiellement disponibles, or ce n'est pas le cas dans notre approche. Au contraire, nous proposons d'explorer les dispositions de l'environnement afin de déceler des services jusqu'alors inexploités.

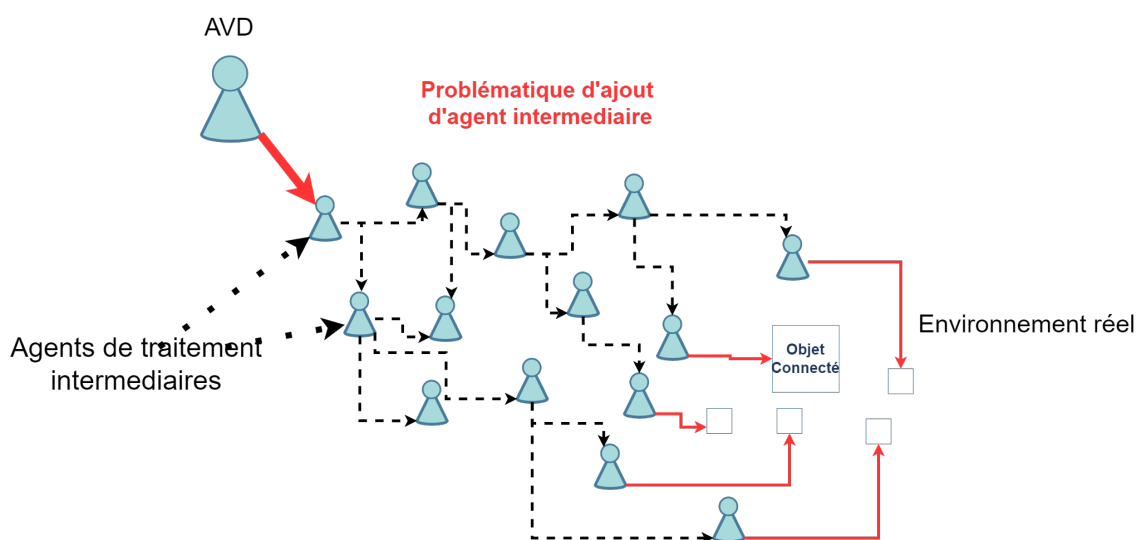


FIGURE 2.12 – Complexification de l'interaction à cause de la présence d'agents intermédiaires

Pour surmonter ces problèmes, il est ainsi nécessaire de proposer un nouveau moyen d'interaction plus direct entre l'agent et son environnement, permettant ainsi à l'agent d'appréhender les possibilités de son environnement de manière moins coûteuse.

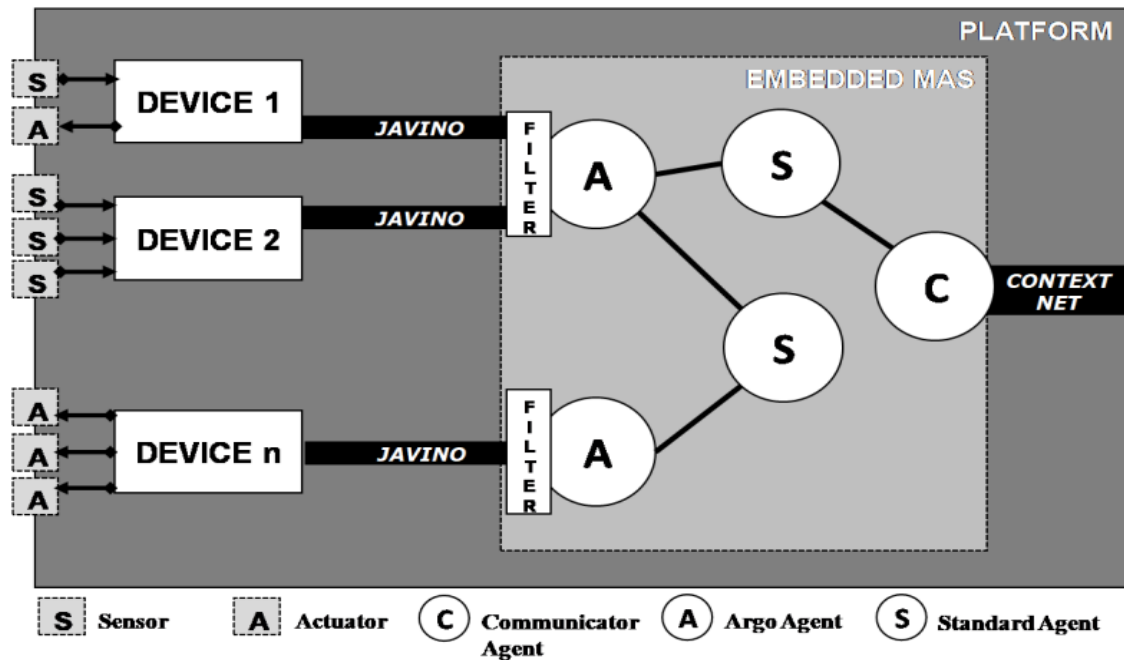


FIGURE 2.13 – Exemple d'utilisation d'agent utilisé comme médiateur entre objets et agents [135]

## 2.5 L'affordance

Issue de la psychologie cognitive, le concept d'affordance, proposé par *James J. Gibson* [71, 70], suggère que la conception des dispositifs présents dans l'environnement informe les utilisateurs sur la façon de les utiliser. Quelques exemples donnés sont les formes d'une poignée de porte qui correspond à la forme d'une main humaine, une surface rigide et horizontale qui encourage une personne à marcher dessus ou une chaise à s'asseoir dessus. En partant de cette hypothèse, Gibson propose la définition suivante de l'affordance : *The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill* [70]. Selon cette définition, la perception de l'environnement serait

donc un phénomène direct qui ne nécessite pas de passer par des mécanismes intermédiaires d’analyse [71].

Pour revenir à notre cadre d’usage, appliquer ce principe au sein d’une intelligence ambiante permettrait de diminuer directement la charge de calcul nécessaire à l’interaction entre un agent et son environnement.

Néanmoins, il existe des débats autour de la place de l’affordance [115] dans le système formé par la cooccurrence d’un acteur et d’un élément de son environnement (le système acteur-environnement). Il existe deux visions principalement défendues :

### **Affordance comme propriété émergente des propriétés de l’environnement.**

Ce premier point de vue suggère que les entités ("thing" dans la définition originale [173]) possèdent certaines propriétés qui sont potentielles, latentes ou possibles : les *dispositions* [173]. Par exemple, un cristal réfractera la lumière à condition qu’il soit déjà réfractif. Par conséquent, le cristal possède comme disposition le fait d’être réfractif qu’il soit exposé à la lumière ou non.

Dans ce cadre, une affordance est ici définie comme un type particulier de disposition, dont le complément est une disposition d’un organisme. En complémentarité à la notion d’affordance, l’auteur définit ainsi l’effectivité, comme la disposition d’un animal à effectuer ou à provoquer une action particulière.

De manière formelle, l’auteur définit ainsi qu’un acteur  $Z$  avec une disposition  $q$  est lié à un objet  $X$  avec une disposition  $p$  complémentaire pour former un système  $W_{pq}$  (ex : système "personne-montant-escalier") =  $j(X_p, Z_q)$  qui manifeste alors une nouvelle propriété  $r$ . La fonction de jonction  $j$  (Figure 2.4) est alors utilisé afin de juxtaposer des dispositions  $p$  et  $q$  que possèdent  $X$  et  $Z$  dans le but de révéler une propriété de rang supérieur  $r$  qui représente une

action de l'acteur réalisé sur un élément de son environnement telle que :

1.

$$W_{pq} = j(X_p, Z_q) \text{ possède } r$$

2.

$$W_{pq} = j(X_p, Z_q) \text{ ne possède ni } p \text{ ni } q$$

3.

$$\text{Ni } Z \text{ ou } X \text{ ne possède } r$$

$j$	$p_1$	$p_2$
$q_1$	$\emptyset$	$\emptyset$
$q_2$	$\emptyset$	$r$

TABLEAU 2.4 – Exemple de détection de la propriété de rang supérieur  $r$  grâce à la fonction de juxtaposition  $j$

Ce n'est que suite à l'apparition de la propriété  $r$  que la disposition de l'acteur et la disposition de l'objet sont identifiés respectivement par une *effectivité* et une *affordance*.

Selon cette définition, une affordance est donc une disposition d'un objet complété par une effectivité. L'affordance est ainsi un appel à l'action poussant inexorablement l'acteur (ou l'animal) à agir sans possibilité, a priori, de lutter contre son déroulement.

### **Affordance liée au système acteur environnement[163].**

Selon ce deuxième point de vue, les affordances doivent être définies en tant que propriétés du système acteur-environnement, et non de l'environnement seul [33, 163].

Ce nouveau point de vue se base sur l'hypothèse que l'affordance n'existe pas sans la présence d'un acteur. De fait, l'affordance ne peut donc être uniquement liée à une seule des parties de ce système.

De manière formelle, un acteur  $Z$  avec une disposition  $q$  est lié à un objet  $X$  avec une disposition  $p$  complémentaire pour former un système  $W_{pq} = (X_p, Z_q)$  qui manifeste une affordance  $h$  tel que :

1.

$$W_{pq} = (X_p, Z_q) \text{ possède } h$$

2.

$$Ni Z \text{ ou } X \text{ ne possède } h$$

Cette définition emmène donc à une différence fondamentale par rapport aux spécifications précédentes de l'affordance qui est que les affordances ne sont pas dans l'environnement. Selon [163], les affordances ne sont d'ailleurs même pas des propriétés mais des relations acteur-environnement. L'affordance n'est donc pas une chose "palpable" mais, comme toute relation entre deux éléments, elle reste tout à fait perceptible.

D'autre part, nous pouvons remarquer la disparition de la fonction de juxtaposition  $j$ . Ce fait est dû à une volonté de mettre en avant une forme d'inexpliquabilité de la nature de la relation acteur-environnement. L'interaction emmenant à l'apparition de l'affordance perd ainsi sa nature combinatoire et est ainsi noté simplement  $p/q$  (relation entre  $p$  et  $q$ ) [163]. En plus de ces modifications au niveau de l'affordance, *Stoffregen* définit une nouvelle capacité de l'acteur : *behavior is what happens at the conjunction of complementary affordances and intentions or goals*[163].

Selon cette hypothèse, il existe donc des affordances qui ne sont pas exploitées et dont l'usage est déterminé par les intentions [163].

De manière formelle, un acteur  $Z$  avec une intention  $i$  est liée à une affordance  $h$  pour former un système  $W_{pq}$  (ex : système "personne-montant-escalier") =  $m(i, h)$  produit un comportement  $b$  (par exemple, monter des escaliers) au travers de la fonction de choix psychologie de l'acteur  $m$  si et seulement si (et quand)  $h$  et  $i$  sont complémentaires.

En d'autres termes, contrairement à la définition précédente où la présence d'une affordance emmenait systématiquement au déroulement des actions as-



sociées, l'acteur peut choisir s'il réalise ou non une action permise par son environnement.

### 2.5.1 Usage

Le concept d'affordance a inspiré de nombreuses approches permettant de modéliser ou de formaliser le lien entre perception et action.

**Robotique** En robotique, la théorie de l'affordance est utilisée pour détecter une action possible sans avoir besoin de quantités importantes de données d'entraînement étiquetées. Par exemple, dans les travaux sur un robot humanoïde de *Peter Kaiser an al.* [89] [88][90] la solution qu'ils utilisent pour améliorer l'adaptation du robot est une détection optique de l'affordance basée sur des données perçues avec une caméra RGB-D. Cependant, la révélation des affordances n'est pas immédiate pour le robot [89].

Il est alors nécessaire d'utiliser des méthodes comme l'apprentissage [89] ou l'identification d'objets (exemples : code QR ou code à barres) pour permettre au robot de reconnaître les éléments disséminés dans l'environnement et de détecter ensuite l'affordance.

**Interaction Homme Machine (IHM)** En IHM, le concept d'affordance est utilisé pour améliorer la facilité d'utilisation, réduire le besoin d'instructions et améliorer la familiarité avec l'interface [164]. On peut, par exemple, citer un site internet qui utilise le soulignement pour suggérer la présence d'un lien hypertexte.

**Autres usages** En parallèle, de nombreux autres domaines mettent à profit ce concept comme dans les recherches liées à ergonomie des objets, l'apprentissage électronique [74] ou encore en simulations [98][3].

### 2.5.2 Conclusion

L’affordance est une notion qui, contrairement à une interprétation cognitive des perceptions, ne fait pas intervenir une représentation mentale intermédiaire.

Toutefois, ce néologisme anglais, dérivé originellement du verbe "to afford", porte plusieurs sens comme « être en mesure de faire quelque chose » ou encore « offrir ». Ces sens jouent un rôle dans la nature même du concept. D’un côté, nous avons le terme d’affordance qui est directement lié aux capacités d’un élément d’environnement *offrant des possibilités* [71, 173]. De l’autre, ce même terme est utilisé afin de caractériser la propriété émergente du couple agent-environnement [33, 163] *permettant de faire quelque chose*. Même si ces définitions diffèrent, les approches tendent cependant à s’accorder sur la présence de deux propriétés complémentaires, l’une appartenant à l’acteur et l’autre à l’environnement, dont émerge une troisième propriété liée à l’apparition d’un lien entre les entités concernées.

D’autre part, en ce qui concerne la littérature, nous notons que le domaine de l’intégration de l’affordance dans un tel contexte d’intelligence ambiante reste peu exploité, voire inexploré.

Ce constat peut être expliqué du fait qu’aucune des formalisations de l’affordance ne soient complètement adaptées au monde virtuel.

Nous pouvons citer par exemple : l’exploitation immédiate des affordances [173], qui ne prend pas en compte le libre arbitre de l’acteur dans son choix d’action; l’inexplicabilité de la relation entre un agent et son environnement, rendant son implémentation impossible [163]; l’impossibilité de l’agent à détecter les dispositions de son environnement simplement au travers du réseau.

Intégrer l’affordance dans une intelligence ambiante doit donc passer par une adaptation implémentable des concepts mis en jeu.

## 2.6 Conclusion générale

Dans ce chapitre, nous avons sélectionné et décrit des travaux en lien avec notre problématique. Dans un premier temps, nous avons pu mettre en avant le fait qu'aucune solution actuelle ne répondait complètement aux exigences définies en introduction (Tableau 2.6). Pour rappel, ces exigences concernent : l'ubiquité de l'approche ([C1]); l'adaptabilité aux contextes ([C2]); la prise en compte de l'éco-conception ([C3]); l'interaction naturelle ([C4]); le déploiement rapide sur un maximum d'appareils existants ([C5]); l'autonomie du système ([C6]); et la gestion multi-pathologique ([C7]).

Cet état de l'art nous a toutefois permis de soulever un ensemble de points à résoudre et d'opportunités à prendre en compte afin de proposer une solution de suivi écoconceptif répondant à toutes nos exigences. En ce qui concerne le type d'intelligence à mettre en place, les travaux étudiés ont mis en avant la pertinence de l'usage des agents logiciel, et plus précisément du méta modèle A&A [56, 86, 154, 187], dans le cadre de la mise en place d'un AVD au sein d'un environnement intelligent. En effet, l'agent, grâce à sa proactivité et son autonomie, représente l'entité la plus adéquate en ce qui concerne la conception d'un AVD. De plus, la nature distribuée de l'écosystème fournit par les systèmes multi agent (SMA) permet de proposer une solution adaptée au contexte de l'intelligence ambiante.

Dans ce contexte, la tendance dans ce domaine est actuellement à l'ajout de dispositifs dédiés à chaque fois que nous voulons ajouter un service dans un milieu. Cette volonté, mise en avant par les entreprises, permet une meilleure mise sur le marché tout en facilitant la manière de percevoir un service pour le consommateur. Toutefois, comme nous l'avons remarqué au sein de l'introduction, cette manière de produire incontrôlée emmène un ensemble de problèmes écologiques graves.

Cependant, dès lors que nous nous éloignons de ces solutions "ah hoc", il apparaît immédiatement une problématique liée à l'interaction avec les appareils disséminés. À cause de la complexité d'analyse des objets connectés, il est difficile pour un agent d'analyser de manière exhaustive son environnement.

Plateformes	Exigences						
	[C1]	[C2]	[C3]	[C4]	[C5]	[C6]	[C7]
	Télémédecine						
OLDES [27]	X	X	X	X	X	X	✓
@Home [155]	X	X	X	✓	✓	X	X
MyHeart [77]	X	X	X	✓	X	✓	X
Caalyx [7, 150]	X	X	X	✓	X	X	✓
E-Care [8, 9]	X	X	X	✓	✓	✓	X
	AAL (sans SMA)						
COOK-Assistant Culinaire [24]	✓	X	X	✓	✓	✓	X
TRICARE [132]	✓	✓	X	✓	✓	✓	X
HeartFaid [38]	✓	X	X	✓	X	✓	X
MavHome [45]	✓	✓	X	✓	✓	✓	X
KopAl [66]	✓	X	X	✓	X	✓	X
	AAL (avec SMA)						
SAPHIRE [105, 103]	✓	✓	X	✓	X	✓	✓
TeleCARE [29]	✓	✓	X	✓	X	✓	✓
HERA [161]	✓	X	X	✓	X	✓	X
OutCare [180]	✓	X	X	✓	X	✓	X
ALZ-MAS [68]	✓	✓	X	✓	X	✓	X

TABLEAU 2.6 – Résumé des solutions étudiées concernant le suivi des personnes en perte d'autonomie

En effet, dans un cadre ubiquitaire, l'agent se trouve dans une situation où les dispositifs, en plus d'être hétérogènes, ne sont observables que virtuellement. L'agent n'a ainsi que peu d'information sur ses propres médias d'interaction avec le monde réel, ce qui complexifie davantage la détection des fonctionnalités primaires telles que proposées au sein de notre approche. Cette constatation pousse les développeurs à produire un ensemble de couche intermédiaire de middleware afin d'abstraire la complexité des échanges d'information entre les entités présentes. Sous une forme purement ingénierique ou sous la forme d'un ensemble d'agents intermédiaires, ces middlewares, fortement utilisés et ayant fait leurs preuves pour des propositions ciblées, sont néanmoins difficilement réutilisables directement à la vue du cadre posé au sein de ce manuscrit.

Une piste pour résoudre cette problématique vient d'un parallèle que l'on peut faire avec la méthode humaine d'analyse de l'environnement : l'affordance. Cette approche met en avant le fait que l'environnement nous suggère les dispositions qui la composent. Ce fait emmènerait l'humain à analyser les possibilités de son environnement de manière préemptive, voire inconsciente.

Parvenir à transcrire cette faculté de manière computationnelle diminuerait alors grandement la complexité liée à l'interaction agent-environnement tout en enrichissant les possibilités disponibles.

Basé sur ce contexte, les chapitres suivants présentent les contributions de cette thèse liée à la proposition et à la mise en place d'un modèle conceptuel basé sur le concept d'affordance.

## Chapitre 3

# Description conceptuelle du modèle AVR

Pour rappel, cette thèse a pour objectif de concevoir un assistant de vie diffus écoconceptif capable de s'adapter à l'évolution des besoins d'une personne en perte d'autonomie tout en tirant profit des objets connectés déjà présents dans son environnement. Dans ce cadre, notre hypothèse principale réside dans la présence de fonctionnalités primaires présentes dans chacun des objets de notre quotidien mais dont l'utilisation est sous-exploitée, masquée par une vision monolithique (Figure 3.1).

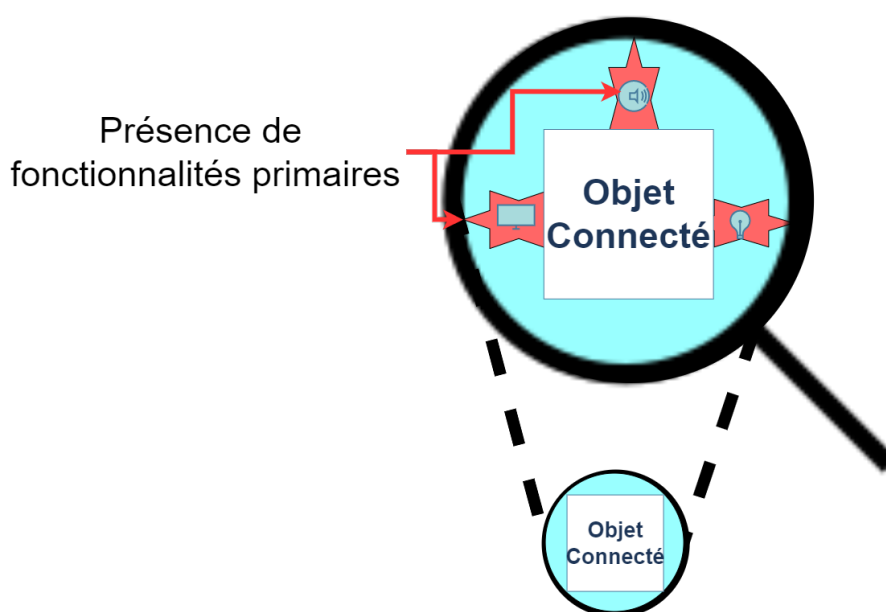


FIGURE 3.1 – Nos objets connectés riches en fonctionnalités

À la suite de notre état de l'art, nous avons conclu qu'aucune des approches

existantes ne proposait de solution répondant à l'ensemble de nos exigences. En effet, malgré le fait que les approches à base d'agent logiciel sont généralement adaptées à la production de service de vigilance, son usage reste limité à des contextes particuliers. Ce constat entre ainsi en contradiction avec notre solution proposée qui s'adapte à la fois au besoin des utilisateurs mais aussi à son milieu. Gérer cette hétérogénéité aux multiples critères ferait ainsi peser une charge trop forte à l'agent qui doit, en plus du service pour lequel il est implémenté, posséder une compréhension poussée des fonctionnalités de nos appareils. À ce problème de gestion, nous avons conclu que la notion d'affordance, basée sur une expressivité directe des possibilités de l'environnement, propose une alternative à moindre coût par rapport aux middlewares classiques.

Basés sur cette conclusion, nous proposons dans ce chapitre un modèle conceptuel générique d'affordance virtuelle recompositionnelle (AVR).

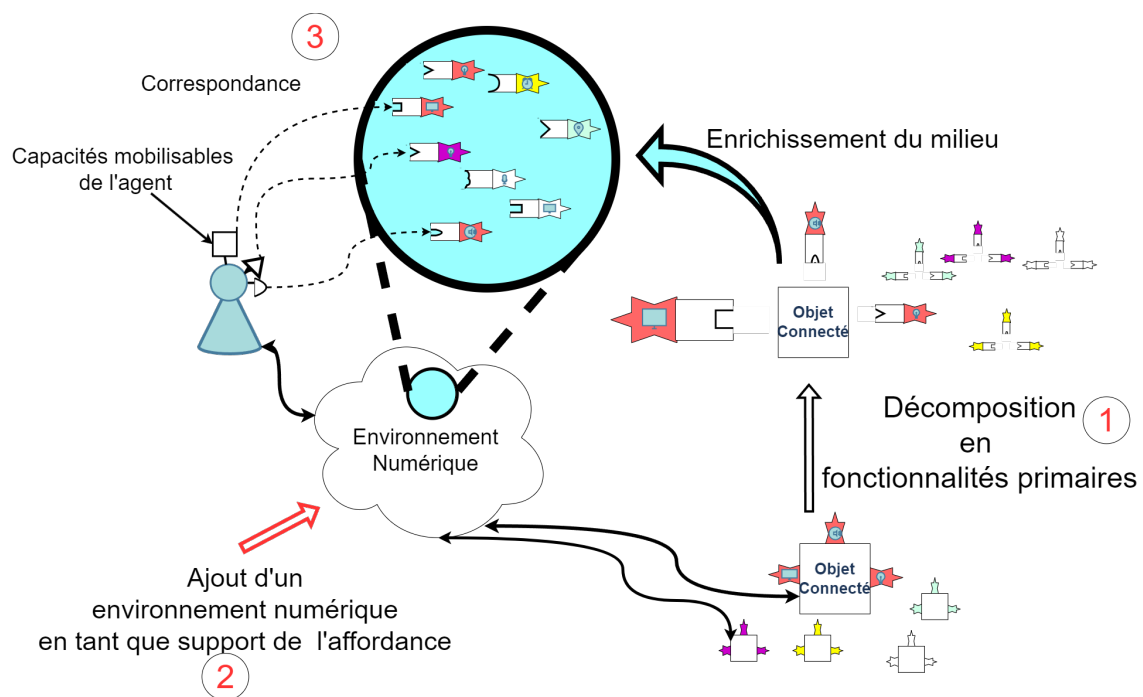


FIGURE 3.2 – Vue globale de l'approche

AVR est un modèle conceptuel qui s'articule autour de trois concepts [61, 62, 149] (Figure 3.2) :

1. **Enrichissement des outils de l'environnement par décomposition :**  
Pour cela, nous proposons une décomposition et une encapsulation vir-

tuelle des fonctionnalités primaires de nos objets connectés afin d'enrichir le milieu.

2. **Interopérabilité de l'agent en contexte ubiquitaire** Dans le but de proposer un support d'accès ubiquitaire aux dispositions, nous proposons la mise en place d'un environnement numérique dans lequel l'agent est en contact direct avec les fonctionnalités primaires présentes dans son milieu.
3. **Composition adaptative des services** Nous proposons la notion d'agent préemptif capable de recomposer dynamiquement des services grâce à la détection des affordances. Cette démarche permet ainsi de proposer un moyen de limiter la charge cognitive liée à la gestion de l'environnement tout en tirant profit d'un maximum d'objets connectés.

Dans ce chapitre, nous détaillerons, dans les trois premières sections, les concepts mis en jeu par notre proposition. Puis, nous conclurons après avoir illustré notre approche grâce à une mise en application du modèle au sein d'un contexte réel.

### 3.1 Enrichissement des outils de l'environnement par décomposition

Dans le cadre de l'affordance, la notion primaire est celle de la disposition. Pour rappel, les dispositions définissent les possibilités de l'environnement à effectuer des tâches génériques. Décrire informatiquement ce concept permet ainsi de proposer un moyen d'enrichir l'expressivité de l'environnement tout en définissant de manière plus précise la notion de fonctionnalité primaire. Toutefois, comme nous l'avons remarqué dans l'état de l'art, il est difficile pour un agent de les percevoir sans l'aide de couches intermédiaires.

Dans le but de révéler ces dispositions, nous proposons de décomposer les appareils connectés sous la forme d'un arbre dont les feuilles représentent les dispositions.



Sur la FIGURE 3.3, nous pouvons voir un exemple de décomposition d'un smartphone. Le smartphone y est décomposé jusqu'à atteindre un ensemble de dispositions, que nous définirons comme atomiques (exemple : capter une photo ou du son, produire du son ou une source lumineuse, disposer d'un calendrier, etc.). Dans ce manuscrit, nous ne nous intéresserons pas à l'ensemble

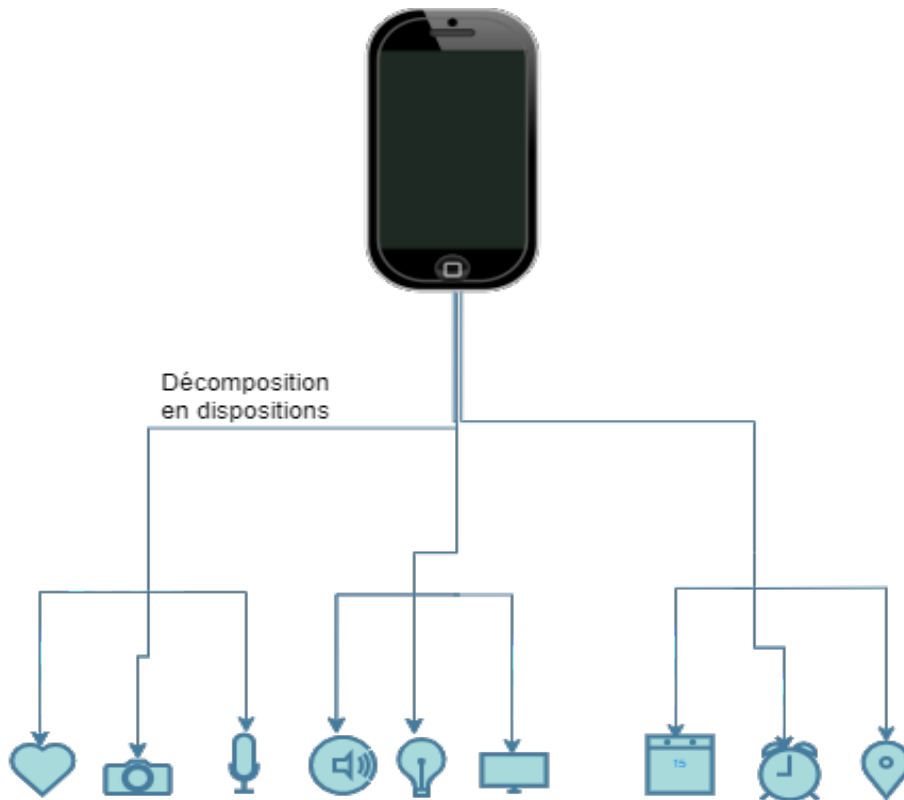


FIGURE 3.3 – Une mise en valeur des dispositions

des dispositions, mais seulement à celles pouvant apporter un support à des agents logiciel dont les actions sont dépendantes des outils logiciels interconnectés. Ainsi, nous en dénombrerons trois qui permettent :

- **Disposition de captation** De récupérer des données de son environnement (accélération, température, position, etc.).
- **Disposition de service** De traiter des processus courants (décompression de fichier, transformation voix en texte, calcul de plus court chemin, etc.)
- **Disposition d'action** D'agir sur son environnement (production de vibration, de lumière, etc.)

Une fois l'objet décomposé, il est ainsi décrit sous la forme d'un ensemble de dispositions virtuelles atomiques pouvant être utiles à l'usage.

**Définition 5** (Disposition Virtuelle). *Une Disposition Virtuelle (DV ou simplement disposition) est une qualité potentielle et latente d'un objet à effectuer des tâches génériques en réponse à une action qu'elle subit.*

Dans l'exemple de l'accéléromètre d'un smartphone ayant comme disposition, le fait de mesurer une accélération, cette disposition virtuelle n'apparaît pas à la suite de la demande de l'agent, mais est bien intrinsèque à l'existence même de ce capteur.

Une fois cette décomposition effectuée, nous proposons de mettre en place une interface normalisée afin de permettre à l'agent de comprendre les fonctions présentées par cette disposition et d'en faire usage. Une telle représentation permet de limiter la capacité cognitive nécessaire à leurs détections tout en évitant la production d'un assemblage de couche intermédiaire.

Dans ce but, nous proposons, en accord avec le méta-modèle Artefact & Agent (A&A), d'encapsuler ces dispositions au sein d'artefacts.

Toutefois, certaines propriétés couramment affiliées à la notion d'artefact ne sont plus adéquates. En effet, les caractéristiques intrinsèques à la notion de disposition nous poussent ainsi à nous positionner sur un certain nombre de points :

— **Encapsulation monofonctionnelle**

Selon la définition proposée par le concept Agent&Artefact (A&A), l'artefact possède une nature polyfonctionnelle. Chaque artefact peut ainsi encapsuler plusieurs fonctionnalités simultanément. Cependant, la notion d'affordance telle que nous la considérons se base sur un ensemble de dispositions atomiques décrivant des fonctionnalités primaires d'un objet. Cette contrainte implique la nécessité de proposer des entités de nature atomique, n'encapsulant donc qu'une seule disposition à la fois.

— **Suppression de la notion de manuel**

Les artefacts A&A sont équipés d'un manuel qui donne une description formelle des fonctionnalités et des modalités d'utilisation. Ce ma-

nuel, qui nécessite une démarche cognitive auprès de l'agent, entre en contradiction avec la notion de préemption mise en avant par le concept d'affordance. En effet, selon le concept, cette perception précède toute forme de réflexion et doit être immédiate. Cet aspect de l'artefact peut donc être supprimé afin de garder une cohérence avec le concept d'affordance.

— **Uniformisation**

À la vue de la nature atomique et prédéfinie des dispositions, l'interface d'usage peut être grandement uniformisée. Cette uniformisation assure une exploitation plus simple des dispositions présentes dans l'environnement tout en diminuant les capacités cognitives nécessaires à son usage.

À la vue des positionnements conceptuels notables avec la notion standard d'artefact, nous proposons donc une architecture adaptée au contexte de l'affordance : l'artefact dispositionnel.

**Définition 6** (Artefact dispositionnel). *Un artefact dispositionnel est une entité computationnelle, encapsulant une disposition d'un objet et la rendant accessible au travers d'une interface normalisée.*

Un artefact dispositionnel est une structure virtuelle qui encapsule une disposition et l'expose au travers d'une interface d'usage normalisée (FIGURE 3.4).

L'usage d'un connecteur normalisé permet de faciliter l'interaction avec l'agent en diminuant la charge cognitive nécessaire à la manipulation de disposition.

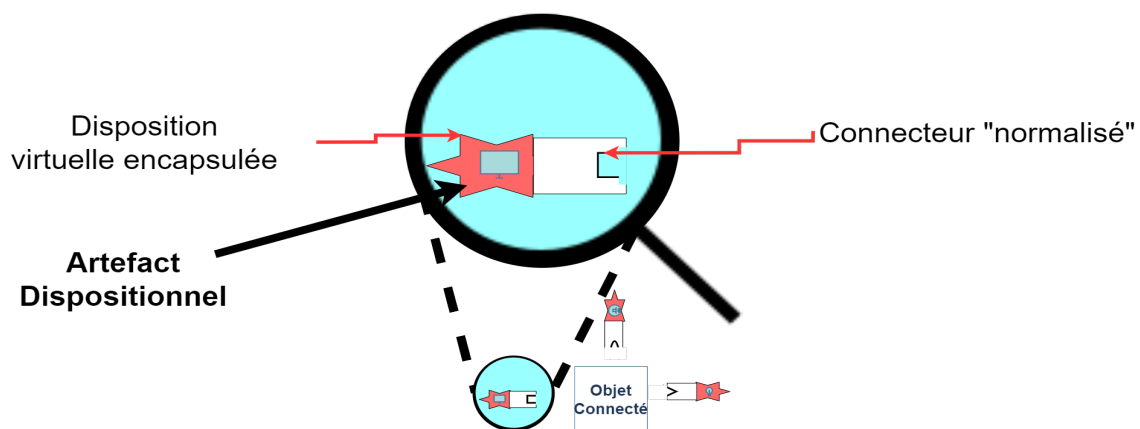


FIGURE 3.4 – Une disposition encapsulée

Une fois décomposé, chaque objet est ainsi représenté par un ensemble d'artefacts dispositionnels.

Grâce à cette démarche, nous enrichissons directement l'environnement sans ajouter de nouveaux appareils. Nous montrons de plus la possibilité de répondre à l'exigence [C3] en proposant une utilisation écoconceptive des appareils déjà présents dans l'environnement initial.

De plus, cette approche respecte notre exigence [C5], en proposant une solution ne nécessitant pas de modification de l'objet initial.

## 3.2 Interopérabilité de l'agent en contexte ubiquitaire

Une fois cette décomposition effectuée, l'environnement est alors enrichi d'outils jusqu'alors masqués au sein des objets de notre quotidien. Toutefois, à la vue du contexte ubiquitaire, il est important de proposer une connectivité offrant la possibilité d'utiliser n'importe quel artefact dispositionnel par n'importe quel agent en utilisant n'importe quel réseau.

Dans ce cadre, nous proposons de mutualiser les artefacts dispositionnels à l'intérieur de workspaces dispositionnels (FIGURE 3.5).

**Définition 7** (Workspace dispositionnel). *Un workspace dispositionnel est une structure topologique thématique contenant des artefacts dispositionnels.*

Un workspace dispositionnel est défini en tant que conteneur de dispositions et permet d'accéder aux artefacts dispositionnels mutualisés. Son but est de créer un espace d'interaction entre l'agent et les dispositions disponibles. Ce terme de "workspace dispositionnel", qui peut être simplifié dans la suite de ce manuscrit par l'utilisation du terme de "workspace", sert notamment à souligner la nature des artefacts qui le compose.

Chaque workspace est associé à une thématique par son créateur (lieu, type de service, cercle social, etc.) et permet une meilleure organisation en facilitant la recherche de dispositions. Si nous reprenons l'exemple de l'accéléromètre du

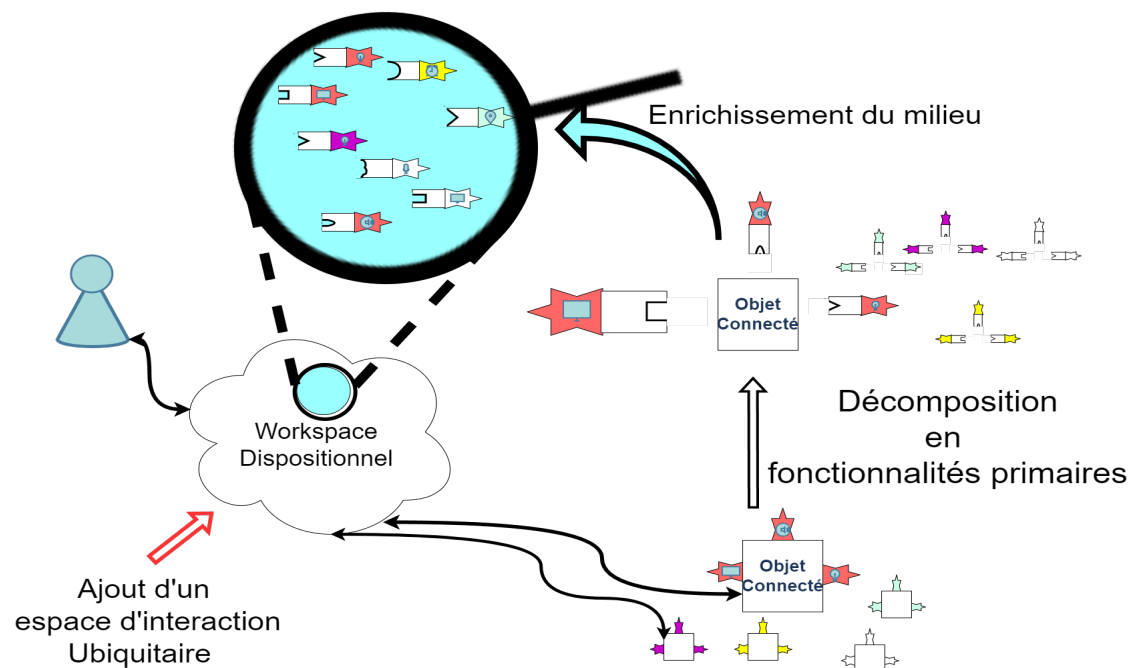


FIGURE 3.5 – Mutualisation des artefacts dispositionnels dans un espace d’interaction

smartphone, il est possible d’exposer l’artefact dispositionnel associé dans un workspace privé pour un usage personnel (bien-être, nombre de pas, suivi d’activité, etc.) ou au contraire de l’exposer dans un workspace partagé dans le cadre d’une alerte en lien avec des aidants (de chute par exemple). Du fait que sa topologie ne dépend pas de celle de l’environnement, cette structure représente une solution au verrou lié au caractère ubiquitaire [C1] de la proposition.

### 3.3 Composition adaptative des services

Une fois que les artefacts dispositionnels sont distribués au sein de workspace dispositionnel, nous proposons de recomposer les services grâce à des agents. En effet, l’agent est une entité complètement adaptée à la production de service autonome dans un contexte distribué. Ce choix permet ainsi de répondre à notre exigence [C2] visant à privilégier la présence d’entité intelligente.

Il est toutefois à noter que pour un agent ambiant immergé dans un environnement ubiquitaire, il est difficile d’appréhender ces possibilités. En effet,

sans liaison par défaut, il est à sa charge de déceler ces opportunités.

Il est donc important de mettre rapidement en adéquation les propres possibilités de l'agent aux différentes dispositions présentes.

Cependant, l'agent, tel qu'il est généralement modélisé, est en incapacité d'interagir directement avec un environnement dispositionnel. Pourtant, dans le cadre de l'affordance, cette interaction est considérée comme préemptive, simple et immédiate.

C'est pour cela que nous proposons une architecture d'agent capable d'interagir naturellement avec les dispositions de son environnement : l'agent préemptif.

**Définition 8** (Agent préemptif). *Un agent préemptif est un agent capable, avant toute action, de percevoir et d'exploiter les dispositions de son environnement afin d'en déduire les affordances disponibles.*

Un agent préemptif est défini en tant qu'entité autonome et proactive capable d'interagir avec les dispositions virtuelles présentes dans son environnement.

Son fonctionnement se différencie de l'architecture classique des SMA (Perception - Décision - Action) et peut être divisé en deux temps : une phase préemptive de détection des affordances et une phase active, plus classique, où l'agent exécute son comportement. Lors de sa première phase, l'agent préemptif est capable de repérer les dispositions présentes intéressantes pour lui.

Pour cela, l'agent traduit, en symétrie à la notion de Disposition Virtuelle, les capacités qu'il est capable de mobiliser au travers d'effectivités virtuelles (Figure 3.6).

**Définition 9** (Effectivité Virtuelle). *Une Effectivité Virtuelle (EV ou simplement effectivité) est une qualité propre d'un agent traduisant une action qu'il peut produire.*

Par exemple, un agent peut avoir comme Effectivité Virtuelle, la qualité propre d'analyser des données d'accélération afin de déterminer la nature d'une chute.

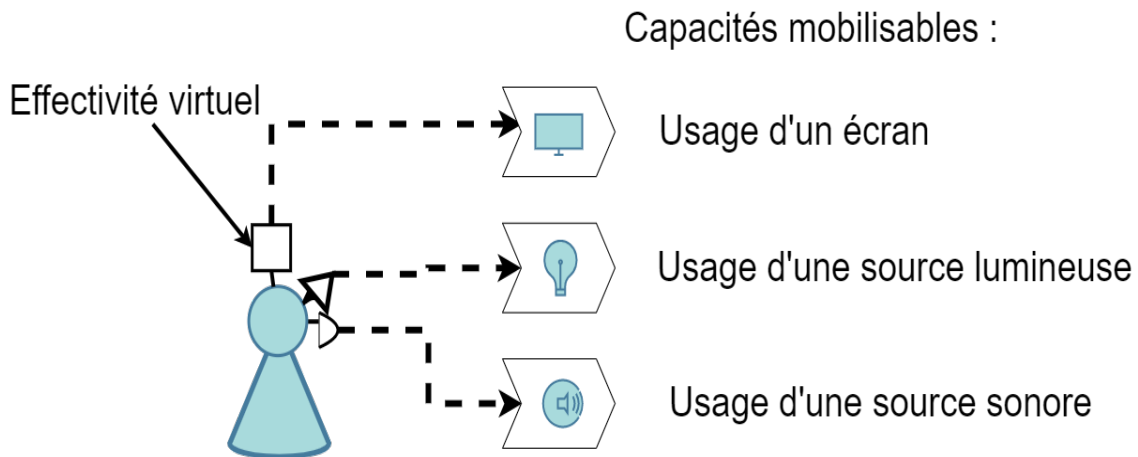


FIGURE 3.6 – Mise en avant des effectivités

Ainsi, en cas de présence d'un tel capteur, il lui sera possible, s'il le souhaite, d'analyser les données.

En présence d'un EV et d'un DV complémentaires, l'agent est capable de reconnaître ce que l'on définit, dans notre cadre d'usage, comme une affordance virtuelle.

**Définition 10** (Affordance virtuelle (AVR)). *Une affordance virtuelle est une propriété qui émerge d'un couple complémentaire Effectivité-Disposition (CM-CD), représentant une action possible de l'agent sur un objet.*

L'affordance virtuelle est une spécialisation de l'affordance adaptée aux contraintes liées à l'informatique dotée d'une émergence qui se fait au travers d'une correspondance entre les EV et les DV.

L'agent perçoit ainsi avec plus de précision et d'exhaustivité les opportunités d'interaction qui lui sont offertes par son environnement. Ces affordances permettent ainsi de renforcer l'adaptabilité de notre solution [C2] tout en proposant une souplesse dans l'utilisation des médias qui peuvent être adaptés aux personnes rencontrées [C4].

En parallèle à cette phase de détection, l'agent passe par un cycle actif où il expose un comportement, réactif ou cognitif, qui exploite de manière active les affordances détectées.

Dans ce manuscrit, nous détaillerons un comportement lié à une détection d'une chute, basé sur une boucle, perception-décision-action, couramment uti-

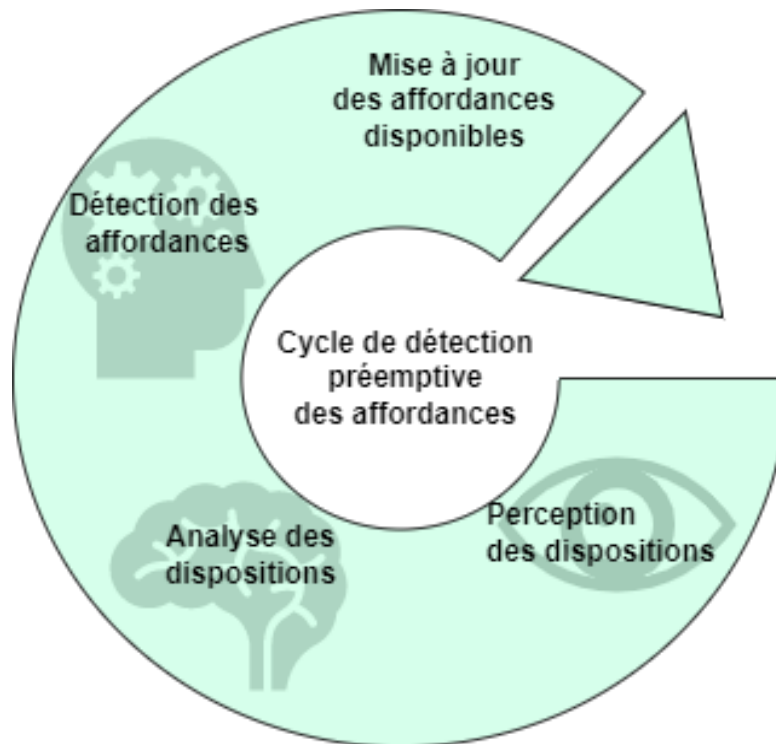


FIGURE 3.7 – Cycle de détection des affordances

lisé dans le domaine de l'agent.

### 3.4 Exemple illustratif d'une mutualisation des composants : Détection de chute

Pour illustrer nos propos, intéressons-nous à l'exemple d'un Assistant de Vie Diffus (AVD), chargé de surveiller Arthur afin d'alerter un aidant (famille ou professionnel) en cas de chute.

Dans le cas usuel, un AVD de gestion de chute serait géré par une seule entité cyber physique qui intégrerait l'ensemble des objectifs et des dispositifs et les limiterait à ceux-ci. Dans notre exemple, l'AVD est défini par un agent préemptif qui possède les effectivités virtuelles suivantes :

- Analyser les données d'un accéléromètre
- Utiliser une source sonore
- Faire clignoter une source lumineuse



Cet agent a accès à un ensemble de dispositifs classiques disséminés dans l'environnement d'Arthur (FIGURE 3.8) avec :

- Une télévision connectée dans le salon
- Un smartphone avec le patient
- Une radio dans la chambre
- Des ampoules connectées dans chaque pièce
- Un système d'alarme
- etc.

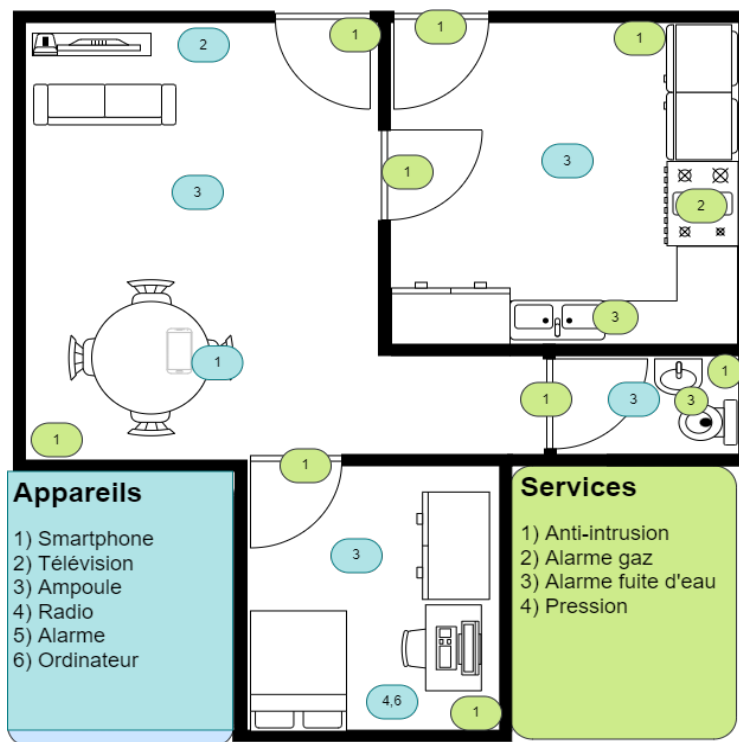


FIGURE 3.8 – Un environnement équipé

Dans cet exemple, il n'existe à première vue aucun appareil ni service capables de détecter une chute. Cependant, il est toutefois possible pour un AVD géré par un agent préemptif de voir plus loin que ces possibilités grâce à la décomposition de son environnement sous la forme d'un ensemble de dispositions. Ces dispositions peuvent, par exemple, permettre d'agir sur l'environnement via des effecteurs, ou de détecter de l'information via des capteurs (FIGURE 3.9).

De cette manière :

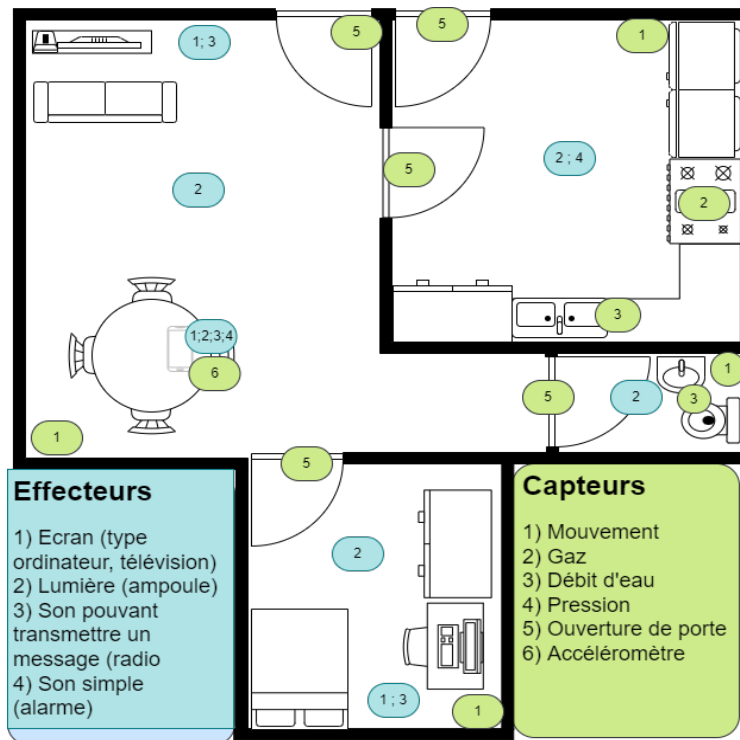


FIGURE 3.9 – Un environnement dispositionnel

- Un Smartphone peut être représenté sous la forme d'un ensemble de dispositions dont
  - une disposition de détection d'accélération
- Une télévision peut être représentée sous la forme d'un ensemble de dispositions dont
  - une disposition de production de son
  - une disposition d'affichage
- Une ampoule peut être représenté sous la forme
  - d'une disposition de production de lumière

Une fois exposé au sein d'un workspace, l'agent peut alors exploiter de manière préemptive les affordances qui lui sont offertes et accéder aux fonctionnalités associées aux dispositions encapsulées.

Ainsi, en ce qui concerne la détection de la chute, l'agent perçoit qu'il est possible pour lui de s'adapter à l'absence de détecteur de chute dédié en profitant de l'accéléromètre d'un smartphone. Il émerge ainsi pour l'agent une nouvelle affordance lui permettant d'utiliser l'accéléromètre en tant que détecteur de chute.

En ce qui concerne les méthodes de réactions de l'AVD, le workspace présente alors deux affordances lui permettant de déclencher des alertes (visuelles ou sonores).

Maintenant que ces affordances ont été détectées, l'agent préemptif postule qu'il est capable de repérer un cas de chute de la personne monitorée, et d'attirer l'attention d'un aidant proche en transmettant un message via la télévision ou d'alerter en faisant clignoter les ampoules.

L'agent se trouve ainsi en capacité d'exhiber un comportement de suivi de chute et ainsi proposer un service qui a priori n'était pas possible dans les conditions initiales.

Il est à noter que nous avons ici un sous-contexte particulier mettant en avant certains dispositifs de l'environnement. Ainsi, l'agent reste toujours en capacité de s'adapter en fonction d'autres contextes présentant les bonnes dispositions. Il est par exemple possible pour l'agent de s'adapter en utilisant l'écran du smartphone ou encore faire retentir une alerte au travers de la radio.

De même, il est possible d'utiliser des dispositifs distants pour alerter une tierce personne. Dans le cadre de l'exemple autour d'Arthur, il est tout à fait possible d'ajouter dans le même workspace les dispositions du smartphone de Lise (sa fille), qui sera ainsi alertée en cas de chute.

Enfin, cette même ampoule et cette même télévision peuvent servir de source d'alerte pour d'autres types de suivi (intrusion, température, hygrométrie, etc.).

### **3.5 Conclusion**

Au sein de ce chapitre, nous avons exposé le positionnement adopté par notre modèle conceptuel AVR.

Dans une première partie, nous avons décrit de manière générale les éléments constitutifs de notre modèle conceptuel AVR que nous avons scindé en trois phases. Une première phase de déconstruction qui se base sur une décomposition en artefacts dispositionnels de nos objets (Figure 3.2). Une deuxième phase de mutualisation des artefacts dispositionnels dans des workspaces dispositionnels afin de rendre les services interopérables en contexte ubiquitaire. En-

fin, une troisième phase de recomposition dynamique des services par des agents préemptifs capable d'exploiter les affordances détectées. Grâce à l'enrichissement de l'environnement par les dispositions et l'interopérabilité offerte par le workspace dispositionnel, l'agent préemptif peut dès lors recomposer des services afin de produire un système adaptatif capable de :

- Reconnaître les opportunités de son environnement en faisant simplement correspondre ses EV aux fonctionnalités primaires présentes
- Recomposer les fonctionnalités de son environnement venant d'appareils hétérogènes afin de fournir des services non prévus initialement
- Utiliser des appareils hétérogènes sans que cela influe sur son architecture ou ne nécessite la présence de programmes tiers
- S'adapter à des changements dans son milieu (ajout/suppression d'appareils)
- Faire évoluer ces services selon les possibilités présentes.

Au travers de ce prisme, il est ainsi possible d'établir plus précisément l'originalité des éléments constitutifs de notre contribution :

- Le modèle conceptuel générique AVR permettant :
  - La révélation des dispositions au travers d'un ensemble d'artefacts dispositionnels.
  - La mutualisation des artefacts dispositionnels à l'intérieur de workspaces dispositionnels.
  - La recombinaison de services par des agents préemptifs capable de manipuler des artefacts dispositionnels.
- Un modèle d'architecture générique **ADW** basé sur le modèle conceptuel AVR permettant la décomposition d'objet et la recomposition de services. Ce modèle décrit l'architecture générale liée aux agents préemptifs, aux artefacts dispositionnels et aux workspaces dispositionnels, ainsi que les algorithmes de décomposition d'objet et de recomposition de service.
- Une implémentation de notre modèle d'architecture ADW sous la forme d'un framework java : AFFORD.
- Une expérimentation en contexte réel de l'application Ubi-Health.



# Chapitre 4

## Formalisation du modèle AVR

Au sein de ce chapitre, nous présentons les bases théoriques de notre modèle AVR. La formalisation de notre contribution est inspirée des travaux liés aux domaines de l'affordance [163, 173] et des SMA [56]. La définition formelle de ce modèle est scindée en trois parties :

- Artefact dispositionnel et décomposition des objets
- Workspace dispositionnel et environnement d'exécution
- Agent préemptif et recomposition des services.

### 4.1 L'artefact dispositionnel

Un artefact dispositionnel est une entité passive, utilisable par les agents et conçue pour encapsuler une suite d'opérations liée à une disposition d'un objet d'un environnement *Env* défini tels que :

$$Env = \langle EnvState, O, \epsilon \rangle$$

Avec *EnvState*, l'état de l'environnement, un ensemble d'objets *O* et  $\epsilon$  la fonction de modification de l'état de l'environnement tel que :

$$\epsilon(EnvState) = EnvState'$$

En d'autres termes, la fonction  $\epsilon$  décrit les événements extérieurs pouvant impacter le contexte réel (perte de contact avec un dispositif, panne, changement d'appareils, etc.).

Chaque artefact dispositionnel  $Art$  représente une disposition  $disp$  particulière d'un objet  $Obj \in O$  (Figure 4.1).

**Définition II** (Artefact dispositionnel). *Une disposition d'un objet  $Obj$  peut être représentée par un artefact dispositionnel noté :*

$$Art_{disp}^{Obj} = \langle disp, \iota, \kappa \rangle$$

avec  $Art$  un artefact dispositionnel,  $disp$  de l'objet initial,  $\iota$  une fonction d'usage et  $\kappa$  la fonction d'action.

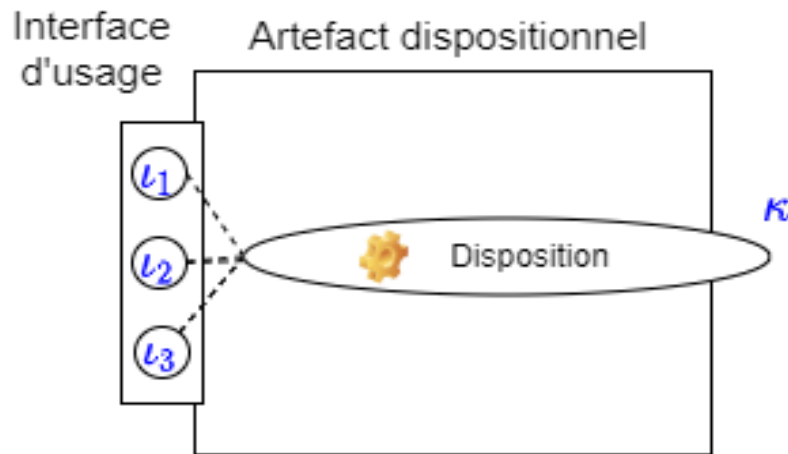


FIGURE 4.1 – Représentation de la modélisation de l'artefact

L'interaction avec un artefact dispositionnel se fait au travers de l'usage d'une fonction d'usage noté  $\iota$ . Ainsi sollicité, l'artefact active alors une fonction d'action  $\kappa$  et effectue la suite d'instruction définie par la disposition encapsulée. En nous basant sur cette définition, nous proposons une formalisation de la décomposition virtuelle de nos objets sous forme d'un ensemble d'artefacts dispositionnels[61]

**Définition 12** (Décomposition en artefact dispositionnel). *Chaque objet  $Obj$ , peut être représenté par un ensemble d'artefacts dispositionnels  $Art$  selon l'usage tel que :*

$$Decomp(Obj) \rightarrow \{Art_{disp}^{Obj}\}$$

avec  $Obj$  un objet connecté,  $Art_n$  un artefact dispositionnel, et  $Decomp$  une fonction de décomposition.

---

**Algorithme 1** : Parcours en profondeur des dispositions d'un objet

---

```

explorerDisposition(Arbre abr, Ensemble ArtifactsDisp, Disposition disp)
1 marquer le sommet disp comme visité
2 if  $\exists disp', disp' \in disp$  then
3   for  $disp'_n \in disp$  do
4     if  $disp'_n \notin abr$  then
5        $disp.addChildren(disp'_n)$ 
6       explorerDisposition(abr, ArtifactsDisp,  $disp'_n$ )
7   else
8     marquer le sommet disp comme disposition finale
9      $ArtifactsDisp = ArtifactsDisp \cup disp$ 
parcoursDispositionObjet(Objet Obj)
10 Arbre abr <- Ensemble ArtifactsDisp <-
11 for  $disp \in Obj$  do
12   if  $disp \notin abr$  then
13     explorerDisposition(abr, ArtifactsDisp, disp)

```

---

L'Algorithme 1 présente la fonction de décomposition d'un objet sous la forme d'un ensemble d'artefacts dispositionnels. Son objectif est d'analyser récursivement les dispositions d'un objet, représenté sous la forme d'un arbre, jusqu'à trouver une disposition ne pouvant être fragmentée. Une fois détectées, les dispositions finales sont encapsulées au sein d'artefacts dispositionnels. Dès lors, il est possible de les exposer au sein de workspaces.

#### 4.1.1 Exemple de décomposition sous forme de dispositions

Afin d'illustrer la décomposition sous forme de disposition, nous allons reprendre notre scénario d'usage.



Dans cet environnement, il est possible de décomposer les objets  $Obj^i$  de l'environnement selon leurs dispositions  $disp_n$  (Figure 4.2). Au sein de cet exemple,

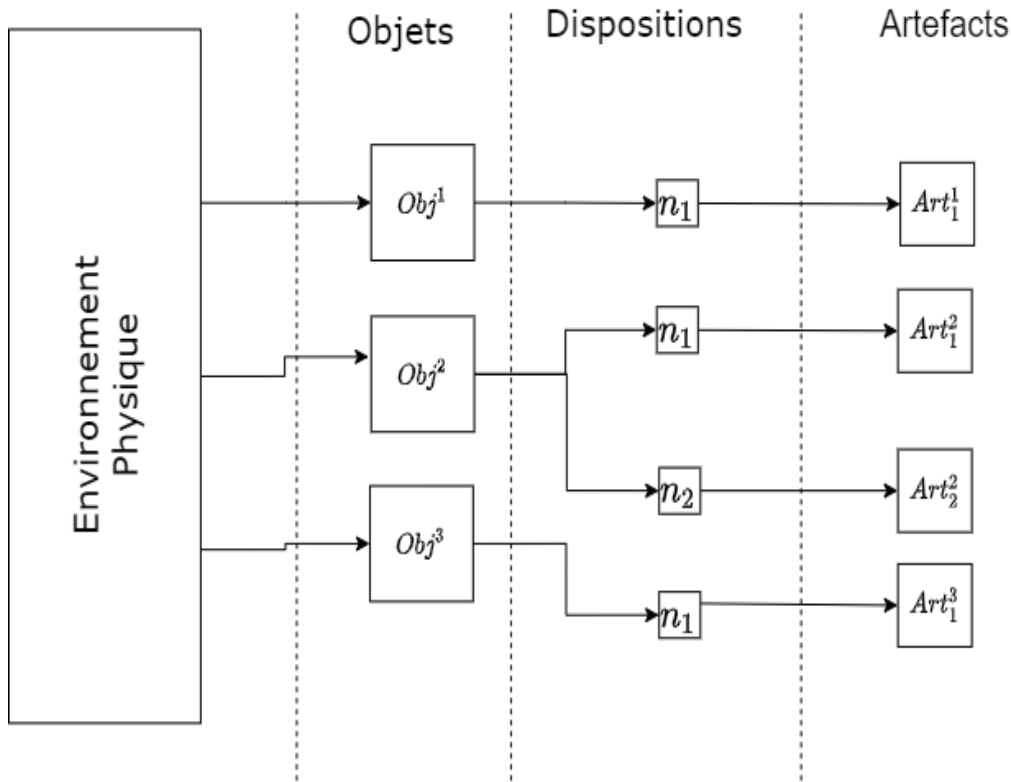


FIGURE 4.2 – Une mise en valeur des dispositions sous forme d'artefacts

nous allons nous intéresser à un sous ensemble de dispositions lié à un sous-ensemble d'objets de l'environnement nécessaire à l'évolution de notre démarche illustrative. Ainsi, il est possible de décrire, de manière non exhaustive, le smartphone, la télévision et l'ampoule connectés de la manière suivante :

- $Obj^1$  :  $Decomp(\text{Smartphone}) \rightarrow \{Art_1^1, \dots, Art_n^1\}$  dont
  - $Art_1^1$  : une disposition  $n_1$  de détections d'une accélération
- $Obj^2$  :  $Decomp(\text{Télévision}) \rightarrow \{Art_1^2, \dots, Art_n^2\}$  dont
  - $Art_1^2$  : une disposition  $n_1$  de production de son
  - $Art_2^2$  : une disposition  $n_2$  d'affichage
- $Obj^3$  :  $Decomp(\text{Ampoule connectée}) \rightarrow \{Art_1^3\}$  avec
  - $Art_1^3$  : une disposition  $n_1$  de production de lumière

## 4.2 Workspace dispositionnel et environnement d'exécution

Pour chaque objet  $Obj \in O$  où  $Decomp(Obj) \neq \emptyset$ , il est possible d'exposer les artefacts dispositionnels qui leur sont associés au sein de workspaces  $WS$ .

**Définition 13** (Workspace dispositionnel). *Un workspace dispositionnel  $WS$  est défini tel que :*

$$WS = \langle wis, \Sigma, E \rangle$$

avec  $wis$ , l'état interne du workspace,  $\Sigma$  l'ensemble d'artefacts  $\{Art_0, \dots, Art_n\}$  exposés au sein du workspace et  $E$  l'ensemble des influences produites modifiant le workspace.

Un workspace est une entité purement topologique qui remplit le rôle de conteneur d'artefact. Soumis à des influences d'ordre cyber-physique, le workspace peut évoluer au fil des expositions/suppressions d'artefacts ou encore de changement de règles internes d'accès. Ces influences peuvent être décrites comme une fonction  $E$  de modification de l'état du workspace tel que :

$$E(WS) = \langle wis', \Sigma' \rangle$$

Dans ce cadre dynamique,  $\Sigma$  représente l'ensemble des artefacts dispositionnels exposés au sein d'un workspace. Comme illustration, nous pouvons formaliser le workspace créé au sein de l'environnement présenté au sein de notre scénario d'usage au travers de l'ensemble d'artefacts qui le compose tel que  $\Sigma = \{Art_1^1, Art_1^2, Art_2^2, Art_1^3\}$  (FIGURE 4.3).

Ainsi mutualisés, les artefacts dispositionnels peuvent alors être manipulés par des agents préemptifs.

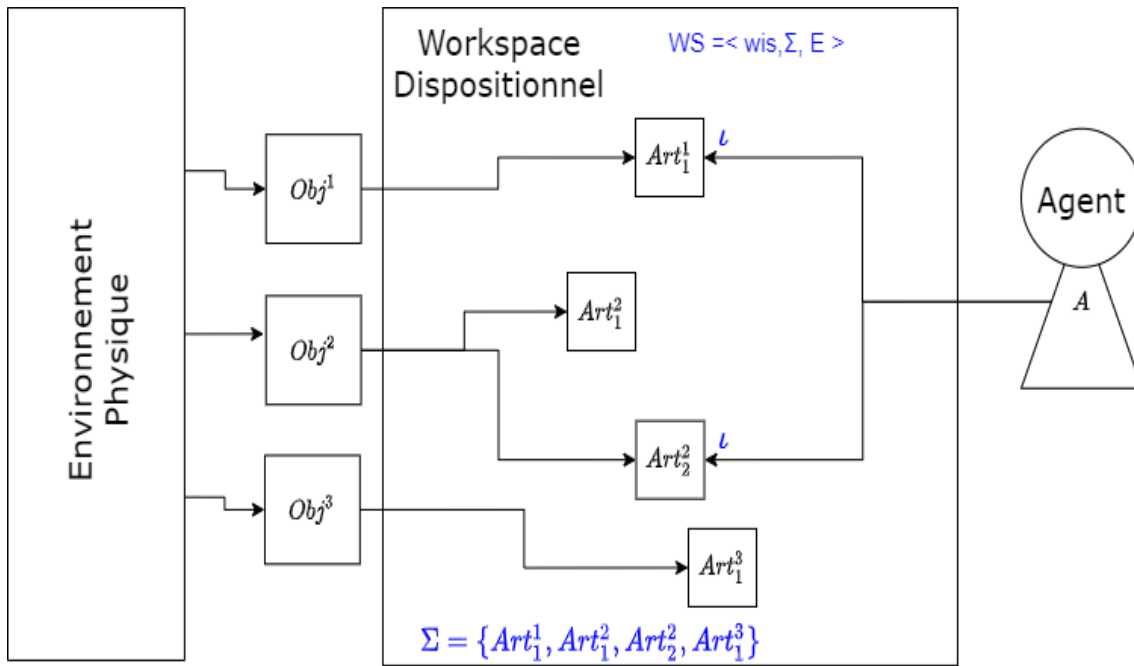


FIGURE 4.3 – Une mutualisation des composants au sein d'un workspace

### 4.3 L'agent préemptif

L'agent préemptif représente le composant proactif du système, capable d'animer un ensemble de services.

**Définition 14** (Agent préemptif). *Un agent préemptif  $A$  est défini tel que :*

$$A = \langle \alpha, effects, \Gamma, intState, \Omega, \Psi, \chi \rangle$$

avec  $\alpha$  la fonction de détection de disposition,  $effects$  l'ensemble des effectivités,  $\Gamma$  la fonction de détection des affordances,  $intState$  l'ensemble des états internes,  $\Omega$  l'ensemble des buts,  $\Psi$  la fonction de choix interne et  $\chi$  la fonction d'action.

Un agent préemptif  $A$  est défini selon un ensemble de propriétés (FIGURE 4.4), que nous proposons de scinder en deux groupes, avec :

- d'un côté les propriétés préemptives :  $effects, intState, \Gamma, \alpha$
- et de l'autre côté, les propriétés actives liées au cadre spatio-temporel :  $\Psi, \Omega, \chi$ .

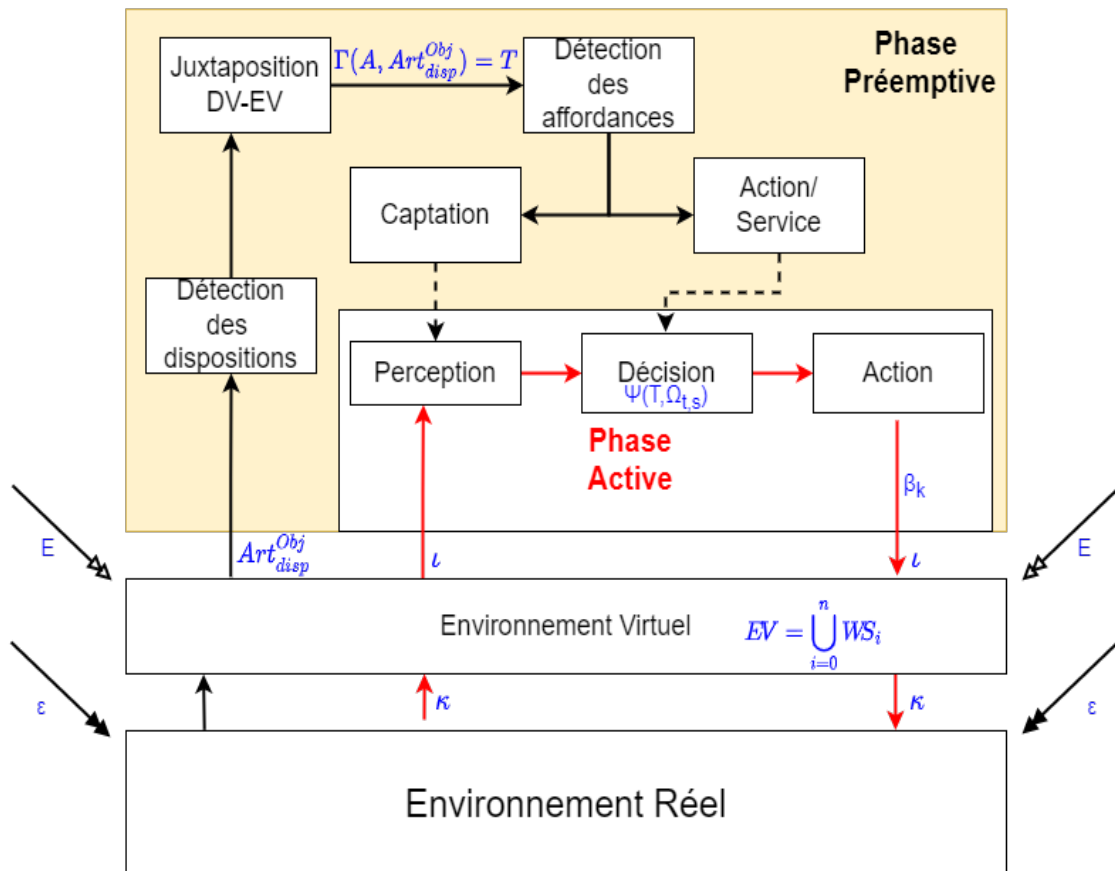


FIGURE 4.4 – Modèle conceptuel de l'agent

### 4.3.1 Propriétés préemptives

Chaque agent préemptif possède un ensemble de propriétés précédant toute forme d'action.

Parmi ces propriétés, l'agent possède notamment un état interne *intState*, un ensemble d'effectivités *effects* qui lui sont disponibles (e.g allumer la lumière, utiliser un vibreur, etc.) et une fonction  $\alpha$  de détection des dispositions présentes au sein de son environnement virtuel *EV* tel que :

$$EV = \bigcup_{i=0}^n WS_i$$

En d'autres termes, l'*EV* définit l'espace de visibilité auquel l'agent a accès. En se basant sur ces propriétés et fonctions primaires, l'agent est capable de trier, grâce

$$\begin{array}{ccc} \Gamma & m_1 & m_2 \\ n_1 & \emptyset & \emptyset \\ n_2 & \emptyset & \tau \end{array}$$
FIGURE 4.5 – Fonction de juxtaposition  $\Gamma$ 

à sa fonction  $\Gamma$  les dispositions utiles parmi l'ensemble d'artefacts dispositionnels disponibles. Cette fonction  $\Gamma$  peut être décrite comme une jonction, ou une juxtaposition, entre les  $m$  effectivités de l'agent et les  $n$  dispositions présentes dans son  $EV$  (Figure 4.5).

De cette fonction de juxtaposition peut émerger une affordance  $\tau$  issue d'une liaison complémentaire entre un agent  $A$  avec une effectivité  $effect$  (parmi ces  $m$  effectivités) et un objet  $Obj$  proposant une disposition  $disp$  (parmi ces  $n$  dispositions).

**Définition 15** (Affordance). *Une affordance  $\tau$  est définie telle que :*

$$\tau = \Gamma(disp, effect)$$

*avec  $\Gamma$  la fonction de juxtaposition,  $effect$  une effectivité de l'agent avec  $effect \in effects$  et  $disp$  une disposition.*

$\tau$  peut-être décrit de manière moins formelle comme étant le système "agent  $A$  capable d'utiliser, grâce à son effectivité  $effect$ , une disposition  $disp$  d'un objet".

Pour plus de confort et dû au fait que les dispositions sont encapsulées au sein d'artefacts dispositionnels ( $Art$ ) nous noterons finalement :

$$\tau = \Gamma(Art_{disp}, A)$$

Ensemble, toutes ces affordances individuelles sont représentées sous la forme d'une affordance globale  $T$  telle que :

$$T = \bigcup_{k=0}^x \tau_k$$

avec à chaque ajout de dispositions au sein de l'environnement virtuel (Algorithme 2) :

$$T(t+1) = T(t) \cup \{\Gamma_a(q, c) \text{ avec } c \in \text{effects}, \\ q \in \Sigma\}$$

et inversement (Algorithme 2) :

$$T(t+1) = T(t) \cap \{\Gamma_a(q, c) \text{ avec } c \in \text{effects}, \\ q \in \Sigma\}$$

---

**Algorithme 2** : Analyse d'un workspace
 

---

AnalyseWorkspace(*Arbre abr*, *Disposition disp*)

```

1 for  $\forall \text{artifact} \in \text{workspace}, \forall \text{effectivity} \in \text{Agent}$  do
2   if  $\exists p, q p \in \text{artifact}, q \in \text{Agent}$  then
3      $\tau \leftarrow \Gamma(p, q)$ 
4     if  $\tau \neq \emptyset$  then
5        $T \cup \{\tau\};$  // Ajout affordance
6     else
7       if  $\tau \in T$  then
8          $T \cap \{\tau\};$  // Suppression affordance
  
```

---

À partir de cette définition, il est alors possible de voir émerger de nouvelles affordances décompositionnelles  $\tau^*$  qui ne seraient pas possibles sans la révélation des dispositions.

**Définition 16** (affordance décompositionnelle). Soit  $W$  (un système agent-objet)  $= (Obj, A)$  et  $W_{pq}$  (un système agent-disposition)  $= (Art_p, A)$  composé d'un artefact  $Art$  possédant une disposition  $p$  et d'un agent possédant une effectivité  $q$ . Une affordance décompositionnelle  $\tau^*$  existe si et seulement si :

1.  $W = (Obj, A)$  ne possède pas  $\tau^*$
2.  $W_{pq} = \Gamma(Art_p, A)$  possède  $\tau^*$
3. Ni  $A$  ou  $Art$  ne possède pas  $\tau^*$

avec  $Art_p \in Obj$

### Exemple d'émergence des affordances décompositionnelles

Nous allons illustrer la phase d'émergence des affordances en nous basant sur l'exemple de l'agent  $A$  de gestion de chute. Cet agent possède, parmi toutes ses effectivités *effects*, les effectivités suivantes :

- $m_1$  : Analyser les données d'un accéléromètre
- $m_2$  : Utiliser une source sonore
- $m_3$  : Faire clignoter une source lumineuse

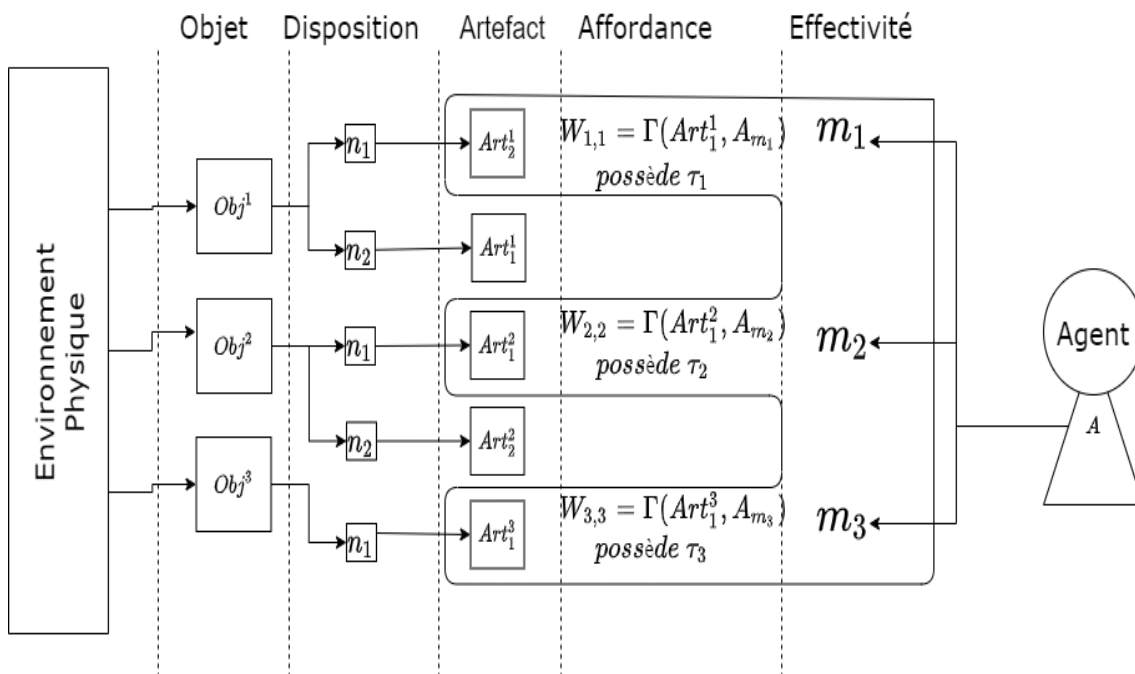


FIGURE 4.6 – Émergence de l'affordance

Dans cette configuration, le système présente alors un ensemble d'affordances  $\{\tau_1, \tau_2, \tau_3\}$  (Figure 4.6) que nous allons détailler.  $\tau_1$  représente l'affordance pour l'agent d'exploiter les données de l'accéléromètre venant du smartphone de la personne avec :

$$W_1 = \Gamma(\text{Art}_1^1, A) \text{ possède } \tau_1$$

$$Ni A \text{ ou } \text{Art}_1^1 \text{ ne possède } \tau_1$$

$\tau_2$  représente l'affordance pour l'agent d'utiliser la télévision comme source sonore avec :

$$W_2 = \Gamma(\text{Art}_1^2, A) \text{ possède } \tau_2$$

$$\text{Ni } A \text{ ou } \text{Art}_1^2 \text{ ne possède } \tau_2$$

$\tau_3$  représente l'affordance pour l'agent d'utiliser l'ampoule connectée comme source lumineuse avec :

$$W_3 = \Gamma(\text{Art}_1^3, A) \text{ possède } \tau_3$$

$$\text{Ni } A \text{ ou } \text{Art}_1^3 \text{ ne possède } \tau_3$$

Dans ce cadre,  $\tau_1$  et  $\tau_2$  sont définies comme des affordances décompositionnelles. En effet, ces affordances sont indisponibles à l'agent sans la décomposition sous forme de disposition. Ainsi :

$$W = (\text{Obj}^1, A) \text{ ne possède pas } \tau_1$$

$$W = (\text{Obj}^2, A) \text{ ne possède pas } \tau_2$$

Ces deux affordances doivent donc être notées  $\tau_1^*$  et  $\tau_2^*$

Ce qui n'est, par exemple, pas le cas de  $\tau_3$  dont la décomposition de l'objet initiale n'a révélé qu'une disposition, qui est de plus l'unique fonction de celui-ci.

### 4.3.2 Propriétés liées au contexte spatio-temporel

Pour un agent, la simple présence d'une affordance ne sera pas forcément suivie du comportement associé. En effet, ses actions peuvent changer au cours du temps afin de s'adapter aux tâches à effectuer pour atteindre son but global  $\Omega$ .

Ce but global  $\Omega$  peut-être diviser en buts  $\omega$  associés à sa réalisation telle que :

$$\Omega = \sum_{k=0}^x \omega_k$$



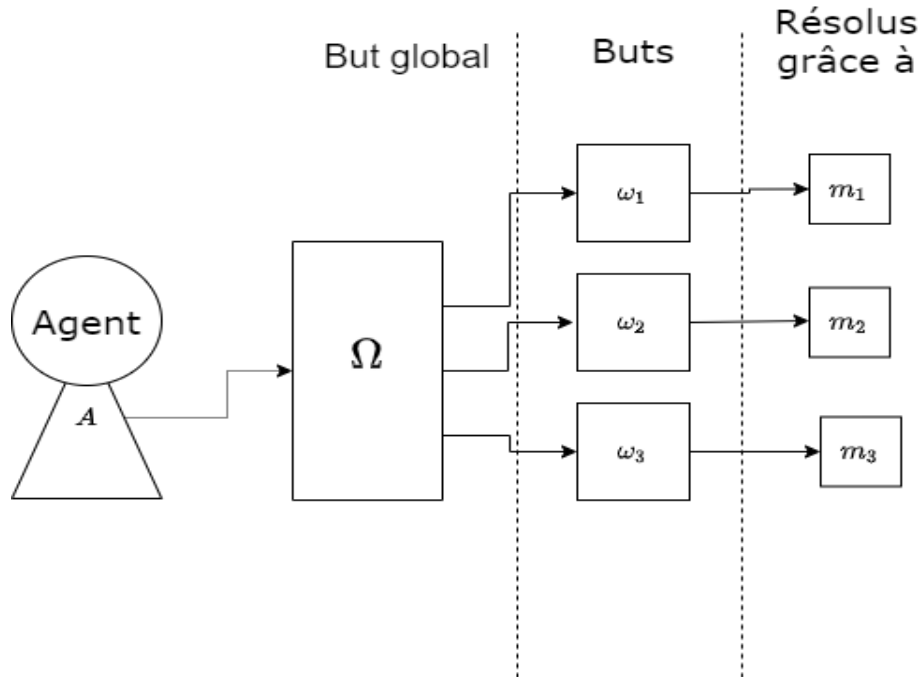


FIGURE 4.7 – Décomposition et résolution des buts de l'agent

Chacun de ces buts  $\omega_k$  peut être résolu au travers d'une suite d'actions basée sur les effectivités de l'agent (Figure 4.7).

Ces buts peuvent varier au cours du temps et l'agent exploite donc un sous-ensemble d'affordances en fonction de ses objectifs courants. Dans ce cadre, le comportement de l'agent est assujéti à la présence d'un but, lié au contexte spatio-temporel, définie comme une intention  $\omega^{t,s}$ .

**Définition 17** (Intention de l'agent). *Une intention  $\omega^{t,s}$  est définie comme un objectif  $\omega$  lié à contexte spatio-temporel  $t, s$ .*

On peut ainsi définir l'intention globale de l'agent  $\Omega^{t,s}$  tel que :

$$\Omega^{t,s} = \sum_{k=0}^x \omega_k^{t,s}$$

En d'autres termes,  $\Omega^{t,s}$  est à considérer comme un aperçu global des intentions de l'agent à un moment précis. Sur la base de cette intention, il est alors possible de définir la notion de possibilité d'action. Une possibilité d'action est ce qui émerge de la conjonction d'affordances et d'intentions complémentaires.

**Définition 18** (Possibilité d'action de l'agent). *Une possibilité d'action  $\phi_{\tau,\omega^{t,s}}$  est définie tel que :*

$$\phi_{\tau,\omega^{t,s}} = \Psi(\tau, \omega^{t,s})$$

*Avec  $\Psi$ , la fonction de choix interne de l'agent,  $\tau$  une affordance et  $\omega$  une intention de l'agent.*

La fonction  $\Psi$  peut être décrite comme une fonction de juxtaposition, entre les intentions de l'agent et les affordances proposées par son environnement. Il est possible de décrire le résultat de cette fonction de manière moins formelle comme étant le système "agent  $A$  a l'intention  $\omega^{t,s}$  d'exploiter l'affordance  $\tau$ ". En nous basant sur la définition de possibilité d'action, nous formalisons la notion de possibilité d'action globale  $\Phi$ .

**Définition 19** (Possibilité d'action globale de l'agent). *Une possibilité d'action globale  $\Phi_{T,\Omega^{t,s}}$  est définie tel que :*

$$\Phi_{T,\Omega^{t,s}} = \Psi(T, \Omega^{t,s})$$

*Avec  $\Psi$ , la fonction de choix interne de l'agent,  $T$ , l'ensemble des affordances et  $\Omega^{t,s}$  l'intention globale de l'agent. De plus :*

$$\Phi_{T,\Omega^{t,s}} = \sum_{k=0}^k \phi_k$$

*Avec  $\phi_k$ , une  $k$ -ième possibilité d'action de l'agent.*

Une fois que la possibilité d'action globale est définie, l'agent exécute alors chaque comportement  $\beta$  associé via une suite d'action réalisé par sa fonction  $\chi$  (Algorithme 3).

**Définition 20** (Comportement de l'agent). *Un comportement est défini tel que*

$$\beta = \chi(\phi_{\tau,\omega^{t,s}})$$

*Avec  $\beta$  le comportement exécuté,  $\chi$  la fonction d'action,  $\phi_{\tau,\omega^{t,s}}$  une possibilité d'action de l'agent.*

D'un point de vue externe, l'agent semble alors produire un comportement global  $B$  qui peut être définie comme le résultat de la mise en action simultanée d'un ensemble de possibilités d'actions ou de la somme des comportements mis en avant avec  $B = \sum_{k=0}^k \beta_k$ .

**Définition 21** (Comportement global de l'agent). *Un comportement global est défini tel que :*

$$B = \chi(\phi_{T,\Omega^{t,s}})$$

Avec  $B$  le comportement global exécuté,  $\chi$  la fonction d'action,  $\phi_{T,\Omega^{t,s}}$  un choix d'action globale de l'agent.

---

### Algorithme 3 : Exécution d'un comportement

---

```

DoABehavior(list < affordance > T)
1  for each intention  $\omega_k^{t,s}$  in  $\Omega_k^{t,s}$  do
2    | if  $\tau_k \notin T$  then
3    | | return False
4  for each intention  $\omega_k^{t,s}$  in  $\Omega_k^{t,s}$  do
5    | for each effectivité  $effect_j$  in effects do
6    | |  $\chi(effect_j)$ 
7  return True

```

---

### Exemple d'émergence du comportement

Comme illustration de l'émergence du comportement (FIGURE 4.8), nous allons reprendre l'exemple de l'agent  $A$  qui possède l'objectif global  $\Omega$  de prévenir un proche lorsqu'une personne est tombée. Cet objectif global  $\Omega$  peut être divisé en trois objectifs qui sont :

- $\omega_1$  : Détecter une chute de la personne
- $\omega_2$  : Alerter le proche par un son
- $\omega_3$  : Alerter le proche à l'aide d'une lumière clignotante

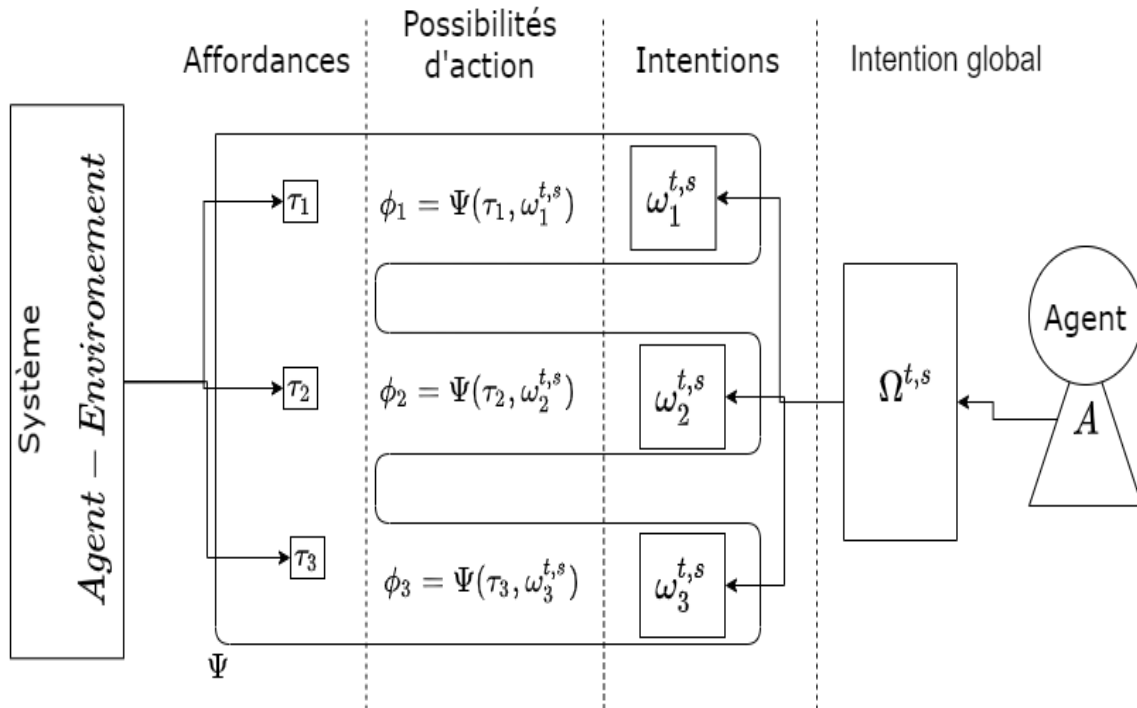


FIGURE 4.8 – Émergence des possibilités d'action

Tout d'abord, pour fonctionner, l'agent doit collecter des informations de chute. Dans ce cas, l'agent a en permanence l'objectif  $\omega_1$  actif sous la forme d'une intention  $\omega_1^{t,s}$ . À la vue de la présence d'un accéléromètre via  $Art_1^1$ , l'agent trouve ainsi une affordance  $\tau_1$  lui permettant de détecter une chute. Cette complémentarité entre une affordance et une intention complémentaire fait émerger une possibilité d'action :

$$\phi_1 = \Psi(\tau_1, \omega_1)$$

Ainsi, l'agent exhibe un comportement  $\beta_1$  de surveillance de données de l'accéléromètre avec :

$$\beta_1 = \chi(\phi_1)$$

Dans des conditions normales, même si les affordances  $\tau_2$  et  $\tau_3$  sont disponibles, les objectifs  $\omega_2$  et  $\omega_3$  ne sont pas sollicités.

Si une chute est détectée, l'agent doit cependant réagir et ses objectifs  $\omega_2$  et  $\omega_3$  deviennent alors actifs.

Dans cet environnement, l'agent à l'affordance  $\tau_2$  lui permettant d'utiliser la source sonore de la télévision  $Obj^2$ . Dans ce cas, où l'affordance  $\tau_2$  et l'intention  $\omega_2$  sont présentes simultanément, l'agent a la possibilité  $\phi_2$  de déclencher le comportement  $\beta_2$  associé aux objectifs d'alerte  $\omega_2$  :

$$\beta_2 = \chi(\phi_2)$$

Enfin, si l'agent ne peut pas résoudre le problème avec son comportement précédent, il essaie d'attirer l'attention de la personne via une alerte visuelle. Dans notre cas, où l'ampoule  $Obj^3$  est présent, l'agent a la possibilité d'action  $\phi_3$  de déclencher les comportements  $\beta_3$  associés aux objectifs d'alerte  $\omega_3$ , avec :

$$\beta_3 = \chi(\tau_1, \omega_1)$$

## 4.4 Conclusion

Au sein de ce chapitre, nous avons présenté le formalisme lié à notre modèle ADW, permettant l'interaction entre un agent et les dispositions présentes au sein de son environnement.

Cette formalisation s'articule autour de trois entités :

- l'artefact dispositionnel dont on définit la nature ainsi que les mécanismes mis en jeu lors de la décomposition d'un objet;
- le workspace dispositionnel, défini en tant que structure topologique contenant un ensemble d'artefacts dispositionnels et soumis aux aléas environnementaux et cyber-physiques;
- l'agent préemptif, défini en tant qu'entité proactive du système possédant à la fois des propriétés préemptives et actives.

En ce qui concerne la définition de l'agent, la section 4.3.2, permet d'aller plus loin que la description de la phase de détection préemptive des affordances en proposant une définition du comportement en tant que phénomène associé à une volonté d'agir de l'agent. Le fonctionnement de l'agent se réalise ainsi sur deux échelles différentes avec :

- d'une part, les affordances, émergeant de la concomitance d'une effectivité et d'une disposition complémentaire;
- d'autre part, le comportement, émergeant de la concomitance d'une affordance et d'une intention complémentaire.

Grâce à cette prise en compte de son contexte sur plusieurs échelles, l'agent peut ainsi produire un comportement adaptatif qui se calque sur les opportunités qui lui sont offertes.

Ainsi formalisés, ces différents concepts nous permettent de poser les bases de la modélisation de l'architecture utilisée au sein de notre implémentation.



# Chapitre 5

## Modèle d'architecture ADW

Au sein de notre approche, l'environnement est modélisé au travers d'un ensemble de workspaces dispositionnels thématiques (selon un lieu, une activité, un propriétaire, etc..) dans lesquels sont mutualisés des artefacts dispositionnels voués à être exploités par des agents préemptifs.

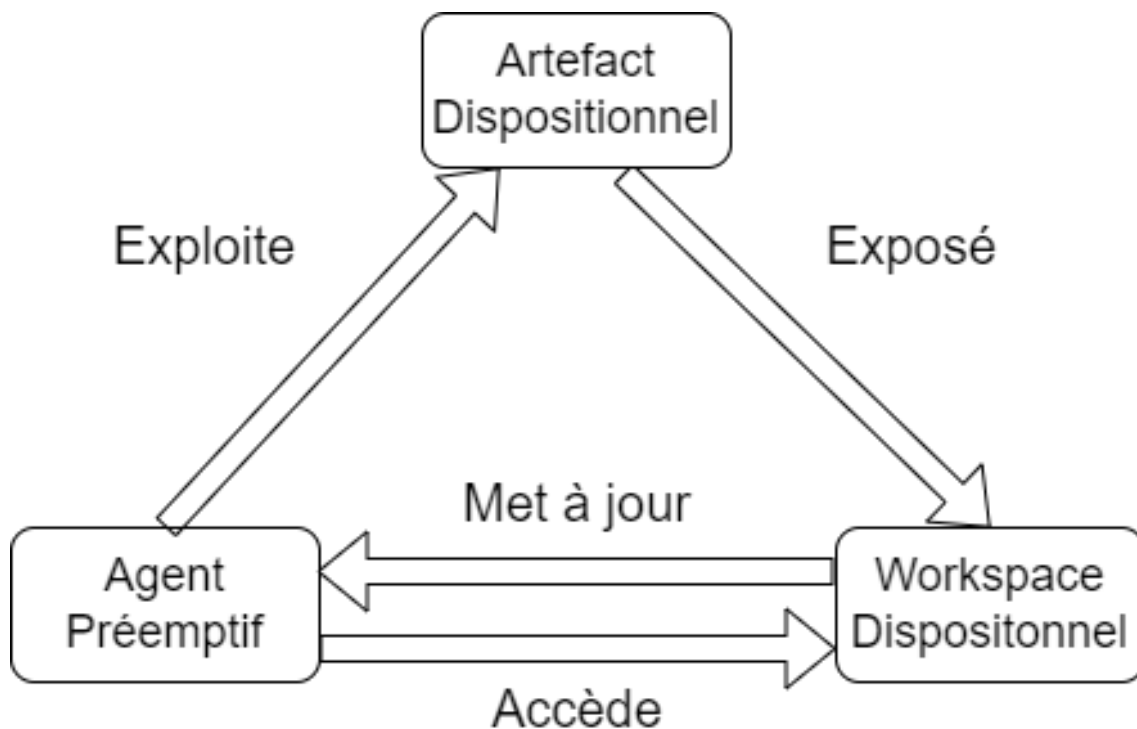


FIGURE 5.1 – Interactions entre l'agent préemptif, l'artefact dispositionnel et le workspace dispositionnel

Dans ce cadre, **ADW** (Agent-Disposition-Workspace) est un motif d'architecture logicielle, destiné aux usages ubiquitaires. Ce motif est composé de trois



types de modules (Figure 5.1) : l'artefact dispositionnel; l'agent préemptif; le workspace dispositionnel.

Chaque module est :

- **Indépendant** les uns des autres dans la mesure où :
  - un agent peut être affilié à plusieurs workspaces simultanément;
  - un artefact peut être exposé dans plusieurs workspaces;
  - et un workspace peut contenir des agents et des artefacts provenant d'environnements réels différents.
- **Interactif** entre eux au travers de trois liens :
  - les artefacts sont mutualisés dans des workspaces dispositionnels;
  - les workspaces reçoivent les demandes d'accès des agents et leur transmettent des modifications subites;
  - l'agent interagit avec les artefacts pour l'exploiter ou s'informer sur son état interne.

Basés sur ce motif, nous proposons un modèle d'architecture en quatre niveaux FIGURE (5.2) : Environnement, Disposition, Mutualisation, Décision.

- Environnement : Englobe tous les dispositifs disséminés dans le contexte de déploiement (matériel, logiciels et ressources externes) ainsi que l'ensemble des utilisateurs du système
- Disposition : Couche permettant la virtualisation des dispositions des objets connectés au travers d'artefacts dispositionnels
- Mutualisation : Couche de mutualisation des artefacts grâce aux workspaces dispositionnels
- Décision : Couche décisionnelle composée d'un ensemble d'agents préemptifs autonomes.

Les trois premiers niveaux de cette architecture représentent les degrés d'abstraction des dispositions aboutissant à leurs mutualisations effectives au sein de workspaces dispositionnels. Le dernier niveau concerne la mise en place des services grâce à une couche décisionnelle peuplée d'agents préemptifs. Dans ce qui suit, nous décrirons chaque niveau de ce modèle. Nous commencerons par le contexte de déploiement.

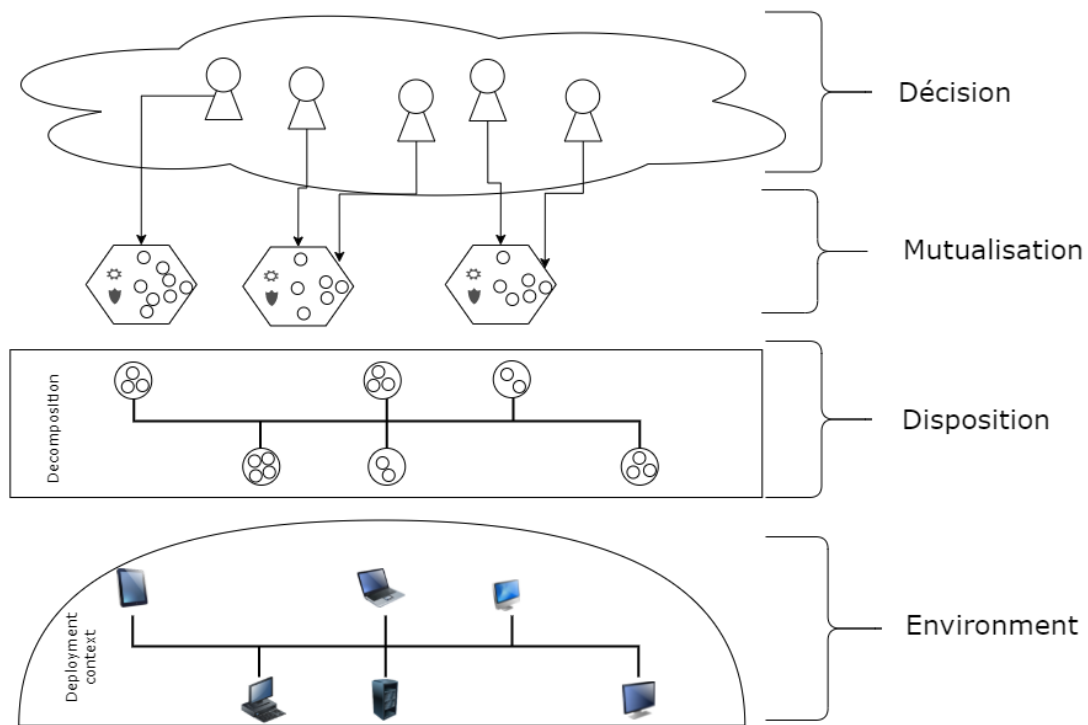


FIGURE 5.2 – Niveau d'architecture du modèle ADW

Puis, nous traiterons des étapes menant à la représentation des objets sous forme d'un ensemble d'artefacts dispositionnels et de leurs mutualisations.

Enfin, nous détaillerons l'architecture de l'agent préemptif.

## 5.1 Environnement

L'environnement réel représente le substrat technologique du système qui peut être considéré à la fois comme contexte d'exécution et comme source d'interactions.

Au sein de notre approche, le cadre de l'informatique ubiquitaire impose une architecture décentralisée et fortement connectée avec notamment : des appareils mobiles (tablettes, montre, etc.) ; des smartphones ; des petits appareils domestiques (décodeurs, télévisions interactives et consoles de jeu) ; des appareils embarqués dans les véhicules (ordinateur de bord) ; etc..

D'une part, ces supports, dont les capacités sont limitées, représentent les ressources requises pour le fonctionnement général du système (les mémoires

vives, les disques durs, les processeurs, les structures de communication). D'autre part, elles représentent une source d'interaction avec l'environnement, avec la présence de capteurs (cardiaque, accéléromètre, débitmètre, etc.) et de divers effecteurs (son, lumière, vibreur).

Une fois disséminés, ces appareils deviennent alors des médias d'interaction pour les utilisateurs afin d'influer sur le comportement du système.

Il est à noter qu'il existe plusieurs types d'utilisateurs qui ne possèdent ni les mêmes exigences ni les mêmes profils d'interaction avec le système :

**Le concepteur** Le concepteur est celui qui implémente les agents en charge des différents services et/ou les artefacts dispositionnels.

**Le technicien** Le technicien est celui qui dissémine les objets connectés dans l'environnement de vie de la personne suivie et s'assure de l'installation des outils logiciels. Il est par exemple à sa charge d'installer les AVD chez une personne en perte d'autonomie.

**L'administrateur** Le profil de l'administrateur est lié à la configuration du système ( autorisation, gestion des profils, etc.). Le fonctionnement et la structure du système dépendent de lui.

**Le bénéficiaire** Le bénéficiaire est la personne monitorée par le système. Ce profil peut être défini comme un utilisateur ayant appris à utiliser le système sans forcément comprendre son fonctionnement interne ainsi que la nature des matériels disséminés.

**Les acteurs auxiliaires** Les acteurs auxiliaires sont les profils utilisateurs en interaction avec le bénéficiaire.

Une fois l'environnement déployé et les profils déterminés, il est possible pour l'administrateur, en accord avec le bénéficiaire, de mutualiser les dispositions des objets disséminés.

### 5.1.1 Illustration : l'environnement d'Arthur

Comme illustration de cette couche, nous pouvons préciser le cadre établi lors de la proposition de l'AVD d'Arthur avec :

- Arthur : Le bénéficiaire,
- Fabrice : L'administrateur,
- Lise : l'actrice auxiliaire.

Nous pouvons de plus décrire les types de support présent au sein de cet environnement :

- le smartphone et la montre d'Arthur utilisés à la fois en tant que support d'exécution pour Ubi-Health et source d'information via les capteurs embarqués.
- le smartphone et la télévision de Lise utilisés à la fois en tant que support d'exécution pour Ubi-Health et source d'alerte.

## 5.2 Artefact

Dans notre architecture, l'artefact définit le module d'encapsulation des éléments passifs du système (Figure 5.3). Il est possible de consulter les informations liées à un artefact (nature de l'objet originel, type, disposition encapsulée, disponibilité, etc.) grâce à deux fonctions :

- *getState* La fonction *getState* qui permet de consulter l'état observable de l'artefact.
- *doViewHolder* La fonction d'affichage par défaut de l'artefact.

Ainsi représenté, l'artefact devient accessible et peut dès lors assurer sa fonction d'encapsulation.

Sur ce point, notre architecture propose deux types d'encapsulation : Structurale et Dispositionnelle (Figure 5.4).

Nous allons détailler ces différentes classes dans les sous-sections suivantes.

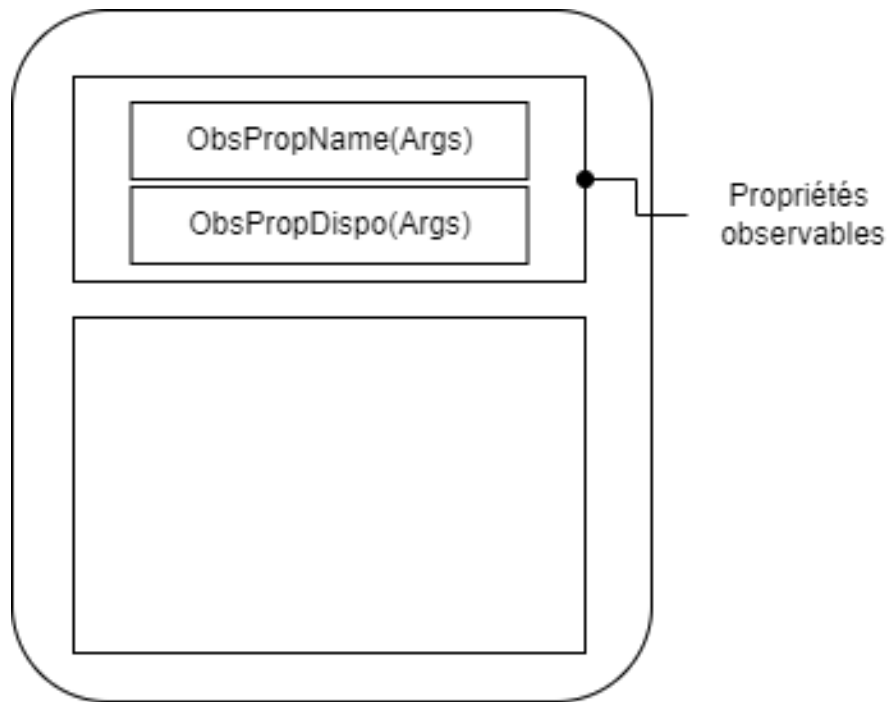


FIGURE 5.3 – Représentation abstraite d'un artefact

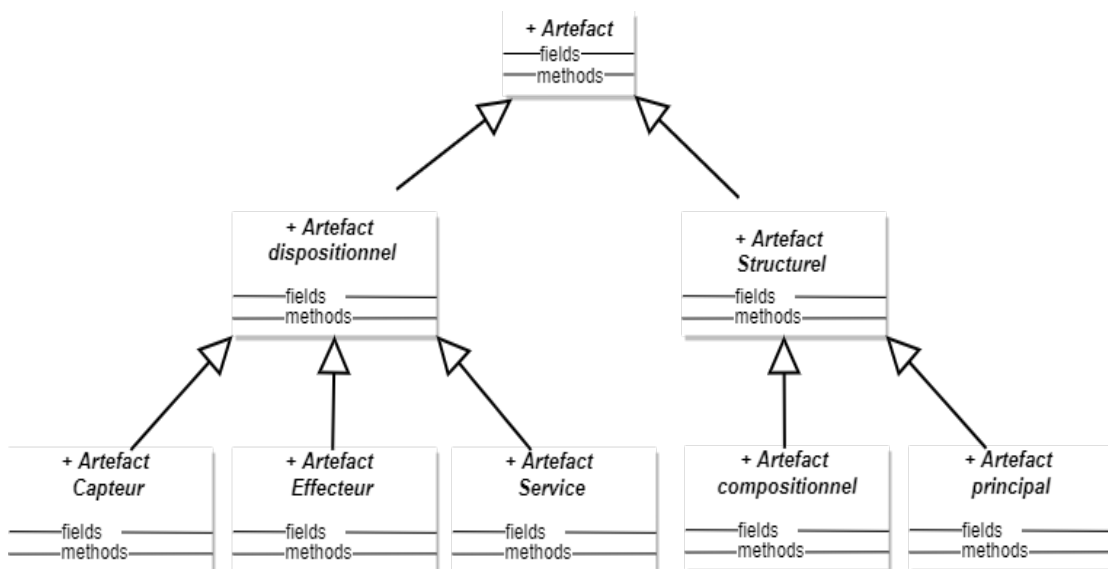


FIGURE 5.4 – Diagramme de classes du modèle de décomposition en artefact

### 5.2.1 Artefact dispositionnel

L'artefact dispositionnel représente la classe qui encapsule les dispositions (Figure 5.5).

Au niveau de son architecture, les artefacts dispositionnels, et les sous-classes associées (Capteur, Effecteur, Service), présentent une interface permettant l'exploitation de la disposition encapsulée. Il est de plus possible de les exposer au sein de workspaces dispositionnels grâce aux méthodes d'ajout *exposeArtifact* et de suppression *unexposeArtifact*.

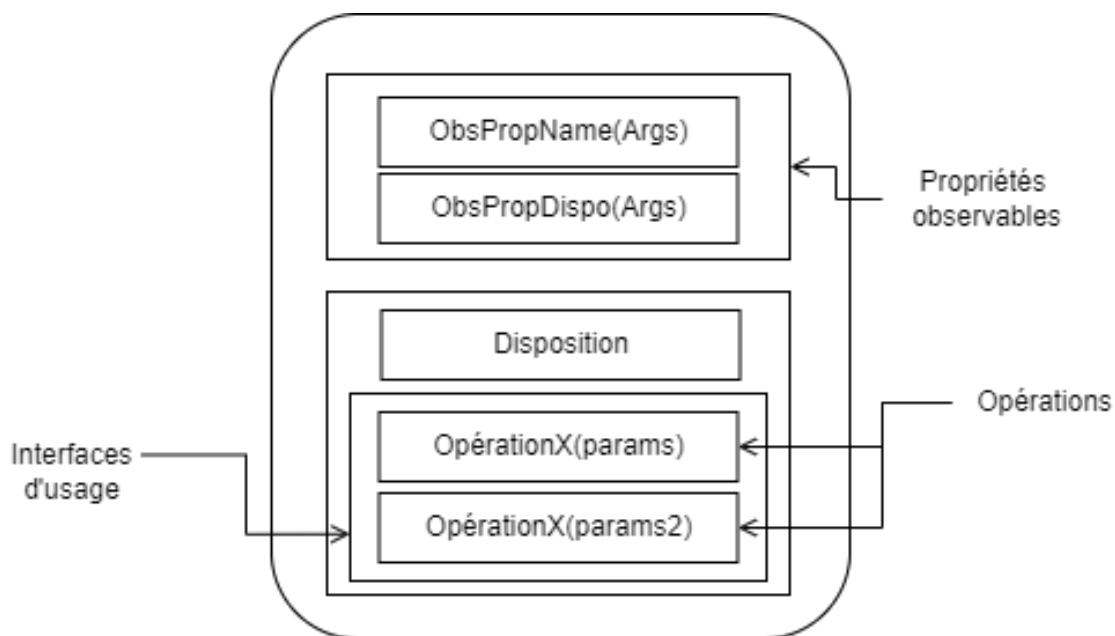


FIGURE 5.5 – Représentation abstraite d'un artefact encapsulant une disposition

En ce qui concerne la dénomination de ces dispositions, il est à noter qu'elle nécessite au préalable la mise en place d'une norme. Cette norme peut alors être : de "jure" c'est-à-dire approuvé par un institut de normalisation reconnu (ISO, CEN ou AFNOR) ou de facto, c'est-à-dire basée sur une entente consensuelle issue d'une pratique commune.

Il n'y a qu'une fois ce langage commun établi qu'il est possible de proposer un artefact dispositionnel exploitable pour les agents.

### Artefact capteur

L'artefact capteur, ou *SensorArtifact*, est une catégorie d'artefacts dispositionnels qui a comme fonction de transmettre les données issues d'un capteur (Figure 5.6).

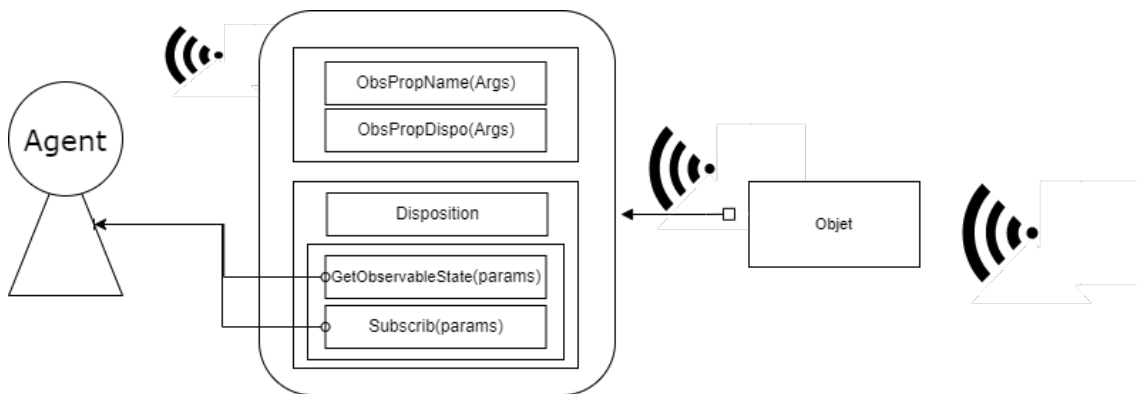


FIGURE 5.6 – Représentation abstraite d'un artefact encapsulant un capteur

En termes d'interaction, le *SensorArtifact* présente deux méthodes de diffusion. Soit, de manière directe via la méthode *getObservableState*, soit sous la forme d'un abonnement grâce à un système de Publish/Subscribe via la méthode *Subscrib*. Cette deuxième alternative permet aux artefacts de tenir informés les agents, sans leur demander d'opérations particulières.

### Artefact effecteur

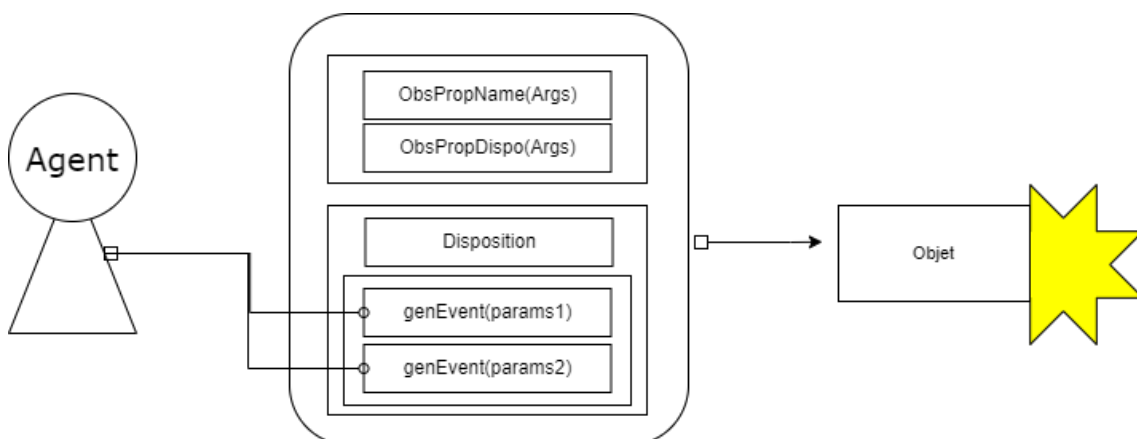


FIGURE 5.7 – Représentation abstraite d'un artefact encapsulant un effecteur

Les artefacts effecteurs, ou Actuators Artifacts, sont des types d'artefacts dispositionnels qui ont comme fonction de réagir aux stimuli reçus en exécutant la tâche demandée (Figure 5.7). Pour cela les agents les actionnent via les fonctions *genEvent*.

### Artefact de service

L'artefact de service, ou Service Artifact, est un type d'artefact dispositionnel qui représente une disposition qui renvoie une réponse suite à une demande (Figure 5.8). Cette catégorie ne peut donc ni être cataloguée en tant que Sensor ni en tant qu'Actuator. On peut donner des exemples tels que le codage de flux vidéo ou l'accès à des services externes tels que la reconnaissance vocale.

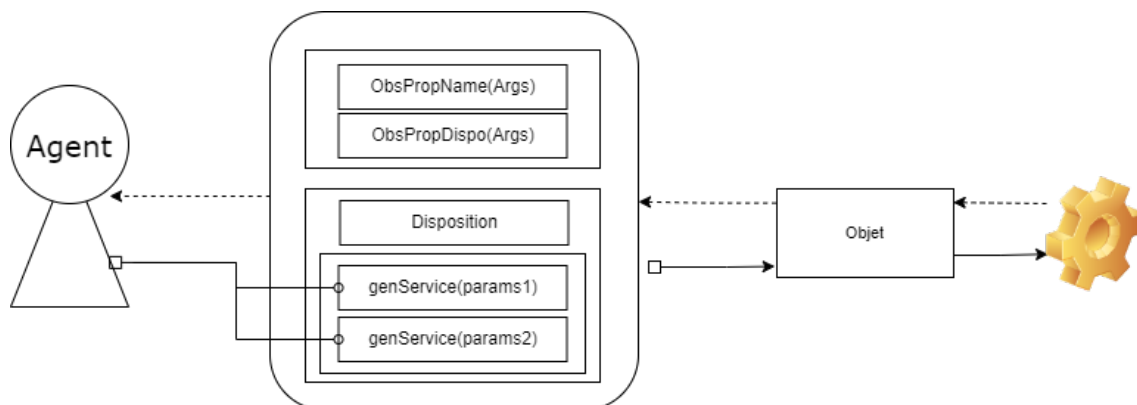


FIGURE 5.8 – Représentation abstraite d'un artefact encapsulant un service

Une demande de service se fait au travers des méthodes *genService* qui retournent à l'agent un objet répondant à sa demande. Cet objet est écrit dans un format léger d'échange de données (e.g. JSON).

#### 5.2.2 Artefact structurel

L'artefact structurel catégorise les éléments intermédiaires produits par la décomposition de l'objet et qui ne peuvent être représentés sous la forme d'artefact dispositionnel.



**Définition 22** (Artefact structurel). *Un artefact structurel est un type d'artefact encapsulant des informations en rapport avec la structure originelle de l'objet connecté*

Comme nous pouvons le voir sur la figure 5.9, ces classes permettent la description de la décomposition d'un objet.

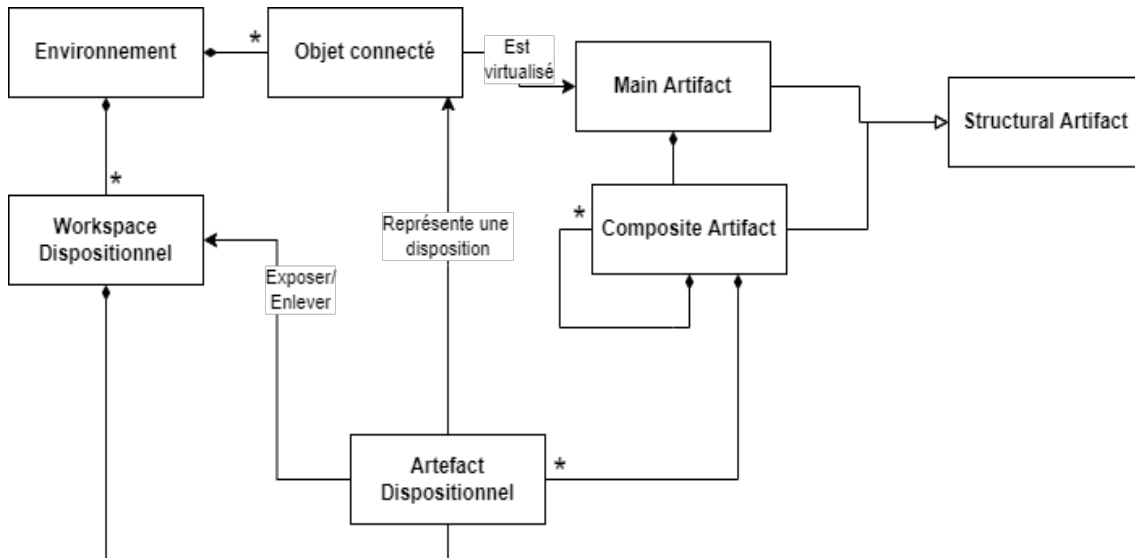


FIGURE 5.9 – Modèle conceptuel de décomposition sous forme d'artefact

Chaque objet est ainsi virtualisé sous la forme d'une structure arborescente dont la racine est encapsulée par un artefact principal (Main Artifact). À partir de cette racine, tous les éléments intermédiaires sont encapsulés par un ensemble d'artefacts composite jusqu'à la mise en avant des artefacts dispositionnels.

Un artefact structurel est défini comme une structure récursive qui permet de représenter, avec la même interface logicielle, les modules composites produits lors de la décomposition. Pour ce faire, chaque artefact dispositionnel est lié aux autres au travers de lien composant-composite qui se fait au travers de *add(child)/remove(child)*. Une fois le lien établi, il est possible d'accéder à un composant enfant grâce à *getChild()*. Conceptuellement, ce lien de dépendance est différent de l'interface de lien (LI) proposé par le modèle A&A qui permet le déclenchement d'opérations qui ne peuvent être déclenchées que par d'autres artefacts. Dans notre cas, cette fonction ne sert qu'à assurer la correspondance avec la structure de l'objet connecté que l'on souhaite garder.

Au sein de notre architecture, nous proposons deux classes d'artefact structurel :

**Artefact compositionnel** L'artefact compositionnel (ou Compositional Artifact) est principalement un conteneur d'Artifact. Son rôle est de préserver la structure que l'on souhaite garder lors de la décomposition de l'objet connecté originel. Ces fonctions sont liées à la gestion des artefacts qui le compose et il fait le lien entre les niveaux de décomposition. En plus de l'accès aux artefacts "enfant", un artefact composite peut accéder à l'artefact dont il est le composant via *getMaster*.

**Artefact principal** La classe des artefacts principaux (MainArtifact) représente virtuellement l'objet connecté dans sa globalité et est la racine de l'arbre de décomposition. Elle contient les informations liées à l'objet (nature, état, adresse Mac, propriétaire, etc.). Cette classe est celle en charge de la sécurité et de l'accès aux artefacts.

## Exemple de décomposition : Le Smartphone

La FIGURE 5.10 illustre une décomposition d'un smartphone en trois étapes. Comme illustration d'une décomposition d'un objet, nous allons prendre l'exemple du smartphone d'Arthur. En ce qui concerne sa décomposition, l'appareil est tout d'abord virtualisé au travers d'un MainArtifact. Dès lors, il est possible de proposer une décomposition au travers d'un ensemble d'artefacts composites. Dans notre cas, une étape de décomposition supplémentaire a été effectuée afin de différencier les types de dispositions (capteur, effecteur, service). Dans notre cas, cette décomposition peut aussi être vue comme un moyen de décentraliser la gestion des dispositions. Enfin, cette décomposition se termine au niveau des feuilles avec les Artefacts dispositionnels, qui représentent les dispositions. Dans le cas du smartphone d'Arthur, nous pouvons trouver par exemple :

- En tant qu'artefact capteur : l'accéléromètre
- En tant qu'artefact effecteur : le haut-parleur
- En tant qu'artefact service : l'analyseur vocal ( speech to text)

Le smartphone n'est alors plus représenté de manière monolithique, mais

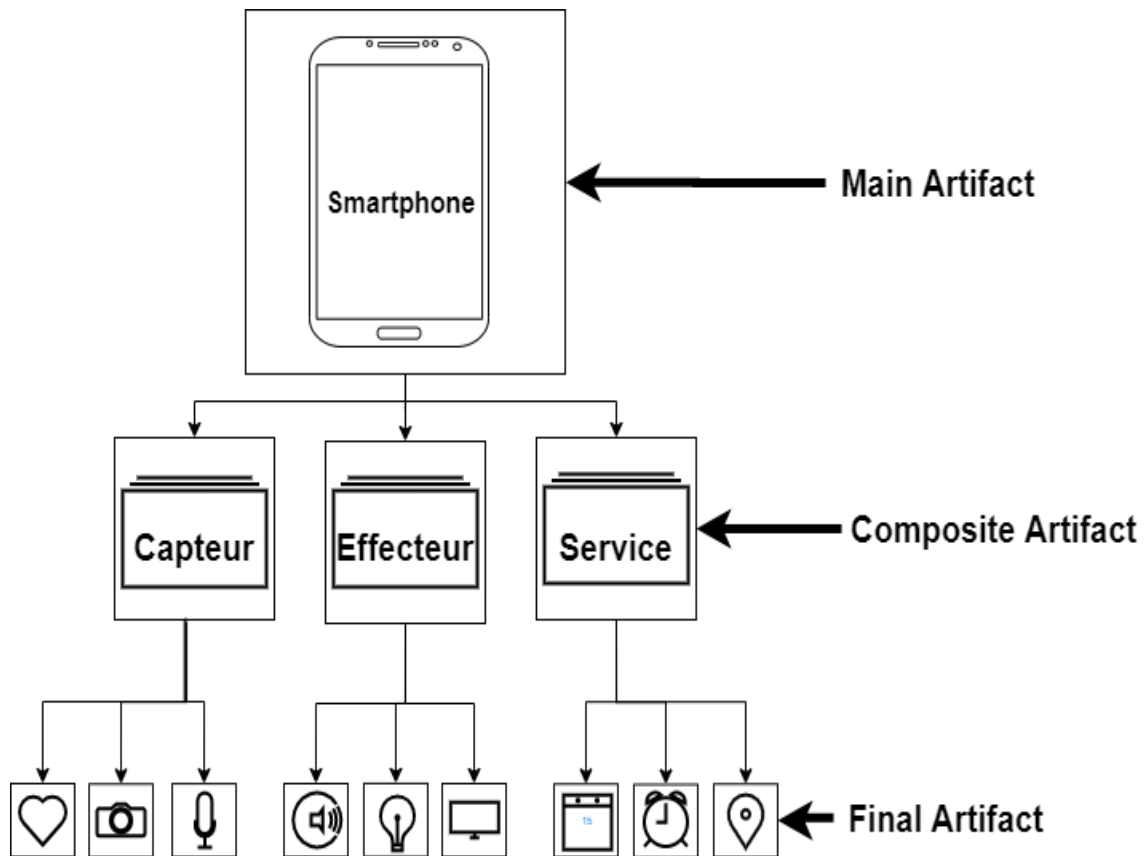


FIGURE 5.10 – Exemple d’une décomposition d’un smartphone

est décomposé en un ensemble d’artefacts dispositionnels qui seront mutualisés à l’intérieur de workspaces dispositionnels.

### 5.3 Le workspace dispositionnel

L’objectif principal du workspace dispositionnel est de fournir un espace topologique pour la mutualisation des artefacts dispositionnels.

Dans ce cadre, l’architecture du workspace dispositionnel se compose de trois couches : stockage, sécurité et agrégation, comme le montre la figure 5.11.

La couche stockage est définie sous la forme de différents nœuds où sont stockées “physiquement” les artefacts dispositionnels : localement ou à distance. Il est possible de déployer cette couche directement sur l’appareil porteur des dispositions ou l’externaliser au travers d’un appareil tiers (exemple : Raspberry, unité centrale, etc.). Dans la couche de sécurité, des mesures de cryptage/dé-

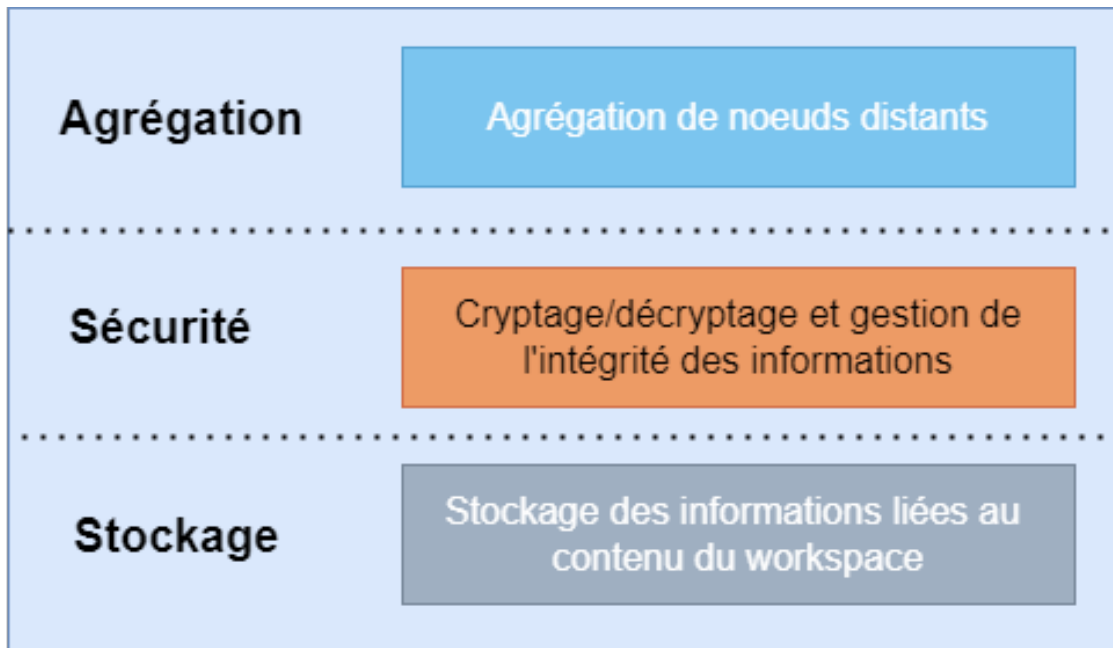


FIGURE 5.II – Architecture du workspace

cryptage des données d'intégrité peuvent être appliquées aux données afin de les protéger. Une identification peut alors être requise afin d'accéder aux informations. Enfin, dans la couche d'agrégation, le système crée un lien entre des noeuds d'un même workspace dispositionnel afin d'en abstraire la complexité liée au réseau.

Une fois le workspace dispositionnel créé, il est possible pour un administrateur ou un bénéficiaire d'y exposer les artefacts dispositionnels :

- **Artefact Entrée / Sortie**

- Exposer un artefact *exposeArtifact*;
- "De-exposer" un artefact *unexposeArtifact*;

Il est aussi possible d'autoriser l'accès des agents préemptifs dans le but d'assurer un ensemble de services (Figure 5.12) :

- **Agent préemptif Entrée / Sortie**

- Ajouter un agent *AgentAccess*;
- Supprimer un agent *removeAgent*;

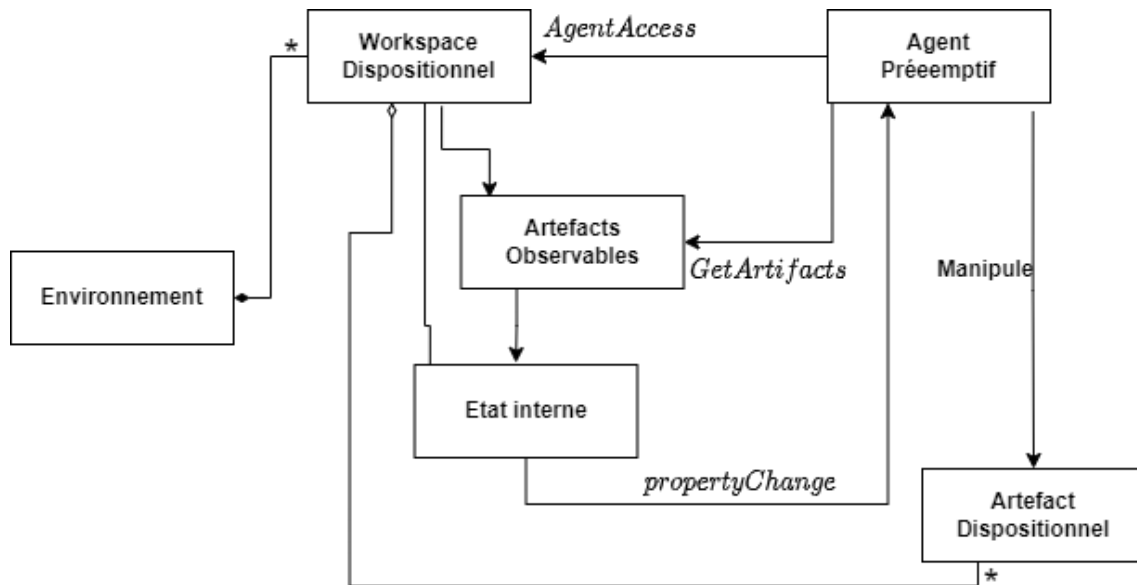


FIGURE 5.12 – Modèle architectural du workspace

Une fois que l'agent préemptif obtient l'accès au workspace, il lui est possible d'accéder à l'ensemble des artefacts dispositionnels mutualisés afin d'analyser ses affordances :

- **Perception**

- La récupération des artefacts dispositionnels *GetArtifacts*,
- L'envoi d'information suite à un événement *propertyChange*

### 5.3.1 Illustration : le workspace lié à la détection de chute

Dans le cadre du scénario d'usage de gestion de chute, la mise en place du workspace est à la charge de Fabrice. Il crée alors un workspace partagé entre les appareils à mutualiser (les smartphones, la montre, la télévision). Dans un premier temps, il configure sur chaque appareil les dispositions à mutualiser (*exposeArtifact*) :

- le smartphone d'Arthur :
  - l'accéléromètre,
- la montre d'Arthur :
  - l'accéléromètre,
- le smartphone de Lise :
  - le haut-parleur,

- l'écran,
- la télévision de Lise :
  - le haut-parleur,
  - l'écran

Une fois cette première étape effectuée, Fabrice active l'agent préemptif associé à la détection de chute sur le smartphone d'Arthur. Une fois l'accès donné au workspace (*Agent Access*), l'agent pourra explorer le workspace (*GetArtifact*) et sélectionner les artefacts dispositionnels qu'il souhaite manipuler.

## 5.4 Architecture de l'agent

L'agent préemptif représente l'entité décisionnelle qui exploite les dispositions, encapsulées par des artefacts dispositionnels, dans le but de produire des services.

Afin de définir les dispositions recherchées par l'agent, le concepteur doit formuler ses préférences en implémentant la fonction *dispositionInterest*. Cette fonction liste les types de disposition que l'agent est en capacité d'exploiter. Une fois ses effectivités et son comportement définis par son concepteur, l'agent peut dès lors être mis en fonction (Figure 5.13). Chaque agent possède comme fonctions primaires trois fonctions liées à son état :

- La fonction *start* : invoquée au démarrage de l'agent.
- la fonction *pause* : invoquée lorsque l'agent est mis en pause. Par exemple quand il n'est plus exposé sur des workspaces ou que tous ces workspaces sont injoignables.
- La fonction *stop* : qui sert à arrêter l'agent.

Ces fonctions assurent à l'administrateur un contrôle total sur l'agent.

À partir du moment où un agent est activé, il est possible pour l'administrateur de le mettre en relation avec un workspace dispositionnel au travers de la fonction *exposedOnWorkspace* (et supprimer d'un workspace dispositionnel avec *unexposedOnWorkspace*). Pour l'administrateur, il est possible de faire en sorte qu'un agent soit simultanément actif au sein de plusieurs workspaces.

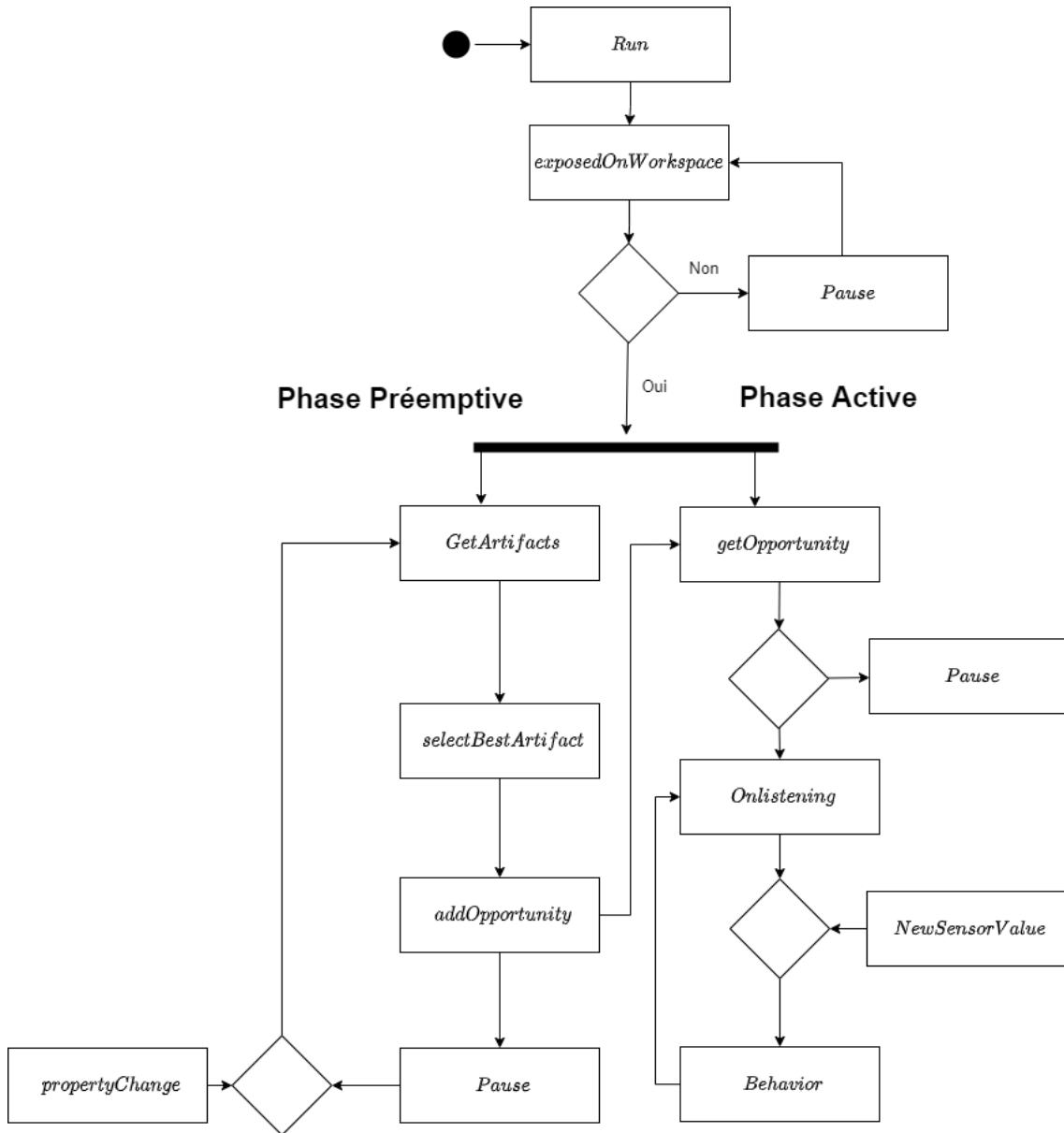


FIGURE 5.13 – Diagramme d'activité de l'agent préemptif

L'étendue de l'environnement accessible à l'agent est alors représentée par la somme des workspaces dispositionnels auxquels il a accès à un moment précis. Dès lors qu'un agent est exposé, il est en mesure d'explorer les workspaces et parcourir l'ensemble des artefacts mis à sa disposition.

L'agent passe tout d'abord dans une boucle préemptive où il va analyser les affordances disponibles.

Cette boucle commence par une analyse des dispositions disponibles. Cette première étape permet à l'agent de faire émerger les affordances proposées par le workspace dispositionnel via *addOpportunity* (et en mettre en veille avec *removeOpportunity*). S'il existe plusieurs affordances différentes pour une même tâche, la plus intéressante peut être discriminée via *selectBestArtifact*. On peut par exemple préciser des règles particulières : topologiques (e.g avoir  $n$  dispositions qui doivent être dans le même workspace); d'arité (p. ex. interagir avec toutes les dispositions de ce type simultanément); réseaux (p. ex. cette disposition doit être dans des sous réseau différent). Ces affordances sont alors sauvegardées et mises à jour à chaque modification opérée dans le workspace dispositionnel.

Une fois cette étape préemptive terminée, il est possible pour l'agent d'exhiber un comportement. Ce comportement suit alors une boucle perception-décision-action, qui est une approche classique dans le domaine des SMA [162, 43, 188] Dans ce cadre, l'agent perçoit des informations de son environnement au travers de *NewSensorValue*, les analyse et décide des réactions avec sa fonction *Behavior* avant de les mettre en action en exploitant les affordances disponibles.

#### 5.4.1 Illustration : l'agent de détection de chute

Dès l'activation de l'agent préemptif et son ajout dans le workspace dispositionnel d'Arthur, l'agent exécute sa phase préemptive et commence par parcourir l'ensemble des artefacts dispositionnels disponibles (*GetArtifacts*). L'agent perçoit alors un ensemble d'artefacts qu'il peut utiliser. Ces artefacts encapsulent cependant des dispositions dont les usages se chevauchent. On peut donner



l'exemple de l'accéléromètre à la fois présent dans le smartphone et la montre d'Arthur et le haut-parleur à la fois présent au sein de la télévision et du smartphone de Lise. Dans cette situation, l'agent doit faire un choix. Dans notre exemple, l'agent privilégiera l'accéléromètre de la montre connectée d'Arthur et le haut-parleur du smartphone de Lise (Algorithme 4). Dès lors, l'activité préemptive est mise en pause et ne sera réactivée que lors d'une mise à jour ou l'ajout d'un workspace dispositionnel.

---

**Algorithme 4** : Exemple de fonction de choix des dispositions

---

```
DispositionChoice(SelectBestArtifact artifact)
1 if artifact.getDisposition() == "Accelerometer"  $\wedge$  (
   getOpportunity("Accelerometer") ==  $\emptyset$   $\vee$ 
   artifact.getMaster().getType() == "Smartwatch") then
2    $\lfloor$  addOpportunity(artifact, "Accelerometer")
3 if artifact.getDisposition() == "Speaker"  $\wedge$  (
   getOpportunity("Speaker") ==  $\emptyset$   $\vee$ 
   artifact.getMaster().getType() == "Smartphone") then
4    $\lfloor$  addOpportunity(artifact, "Speaker")
5 return
```

---

Suite à cette phase préemptive, l'agent passe en phase d'écoute (*Onlistening*) et analyse les données issues de l'accéléromètre de la montre connectée. Enfin, en cas de suspicion de chute, l'agent réagit en utilisant le haut-parleur du smartphone de Lise afin de transmettre l'alerte.

## 5.5 Conclusion

Nous avons présenté dans ce chapitre notre modèle d'architecture ADW basé sur le motif d'architecture du même nom (Figure 5.14).

Cette architecture permet la décomposition et la mutualisation des dispositions des objets connectés sous la forme d'un ensemble d'artefacts normalisés (dispositionnel et structurel) dans le but de les rendre facilement utilisable par des agents.

Outre la diminution d'intermédiaire dans l'interaction entre les agents et les dispositions initié par notre modèle ADW, l'architecture apporte des originalités dans sa conception même.

D'une part, cette architecture propose un découplage entre chacune de ses entités constitutives, ce qui permet d'assurer une adaptation du système aux modifications d'environnements, de profils et de services. Par exemple, il serait possible de modifier l'environnement ainsi que la nature des services mis en place sans avoir à mettre à jour les autres niveaux du système.

D'autre part, ADW est suffisamment souple pour permettre la cohabitation avec d'autres architectures répandues. Cette propriété est notamment due à la diffusion des artefacts dispositionnels au sein d'un workspace et à l'usage d'agents préemptifs ubiquitaires. Nous pouvons l'illustrer par le fait que rien n'empêche l'agent au sein d'un même workspace de profiter soit de l'usage de services dans le cloud afin de profiter de l'accès à des services informatiques distants, soit d'outils propres à l'Edge computing en permettant d'exécuter des services en périphérie dans le but, par exemple, d'éviter la fuite de données personnelles. [159].

Suite à cette description architecturale de notre proposition, nous développerons dans les chapitres suivants une implémentation du modèle ADW au travers du framework AFFORD et de l'application Ubi-Health.

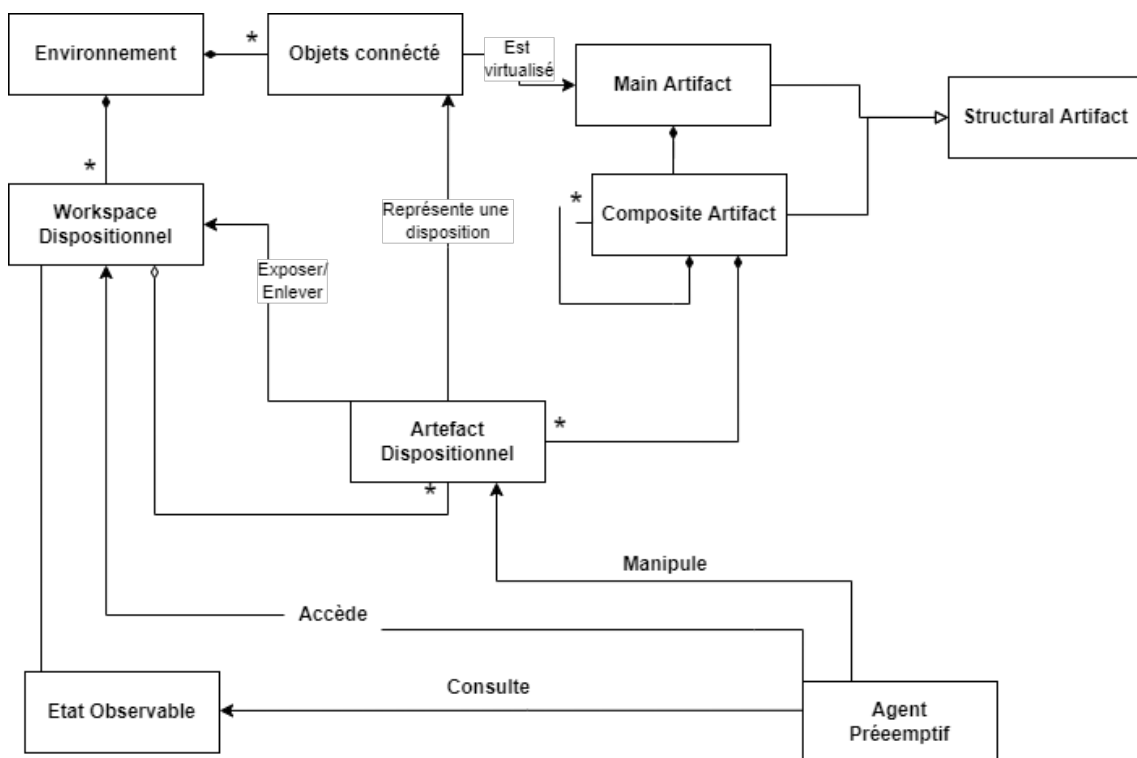


FIGURE 5.14 – Synthèse de l'architecture générale de la proposition

## Chapitre 6

### Framework Java : AFFORD

Basés sur notre proposition du modèle d'architecture ADW, nous montrons la pertinence des concepts proposés ainsi que leurs faisabilités grâce à un Framework Java pour Android : Agent Framework For Omnipresent Real Device (AFFORD)(Figure 6.1).

Le choix de la plateforme Android n'est pas anodin. Cette plateforme, appartenant à Google, est un système d'exploitation mobile, basée sur une version modifiée du noyau Linux. En 2019, Google a annoncé avoir atteint les 2,5 milliards de terminaux Android actifs, ce qui en fait le système d'exploitation le plus répandu. Conçu principalement pour les appareils mobiles tels que les smartphones et les tablettes, Android a su s'étendre vers d'autres types d'appareils tels que les ordinateurs, les télévisions (Android TV), les voitures (Android Auto), les Chromebook et autres montres connectées (Wear OS).

Ce panel de produits illustre parfaitement les types d'appareils les plus à même d'être porteur de capteurs et d'effecteurs dont les usages ne sont pas mu-



FIGURE 6.1 – Logo AFFORD

tualisés.

De plus, l'utilisation de cette plateforme nous permet l'utilisation de la norme de facto d'Android, basé sur une entente consensuelle entre les différents acteurs du milieu.

Dans ce cadre, notre framework AFFORD a pour but de proposer une structure logicielle à utiliser comme fondations pour les applications souhaitant exploiter le modèle ADW. Il ne s'agit donc pas d'un outil pouvant être utilisé directement par les utilisateurs, mais voué à être exploité par un développeur au sein de cadres d'implémentation divers.

Au sein de cette section, nous allons décrire les spécificités importantes pour la compréhension du framework AFFORD (Figure 6.2) :

- Android Runtime
- AFFORD Core
- AFFORD Services
- AFFORD Entity Manager for Android (EMA)
- AFFORD User Interface (UI)

## 6.1 Android

Afin de comprendre les spécificités d'implémentation de notre framework pour cette plateforme, il est important d'explicitier certains points techniques.

Du point de vue du développeur, une application Android est écrite en Java. Cependant, contrairement aux applications Java standard, une application Android possède plusieurs composants et chacun d'eux représente un point d'entrée dans le programme. Chacun de ces composants peut ainsi à tour de rôle devenir l'entité qui gèrera les cycles de vie des briques logiciels qui en dépendent. En effet, contrairement aux paradigmes de programmation dans lesquels les applications sont lancées avec une méthode *main()*, dans le système Android, l'interaction d'un utilisateur avec l'application ne commence pas toujours au même endroit.

Il est donc nécessaire pour un développeur de prendre en compte chacun des types de composants pouvant être la source de cette interaction :

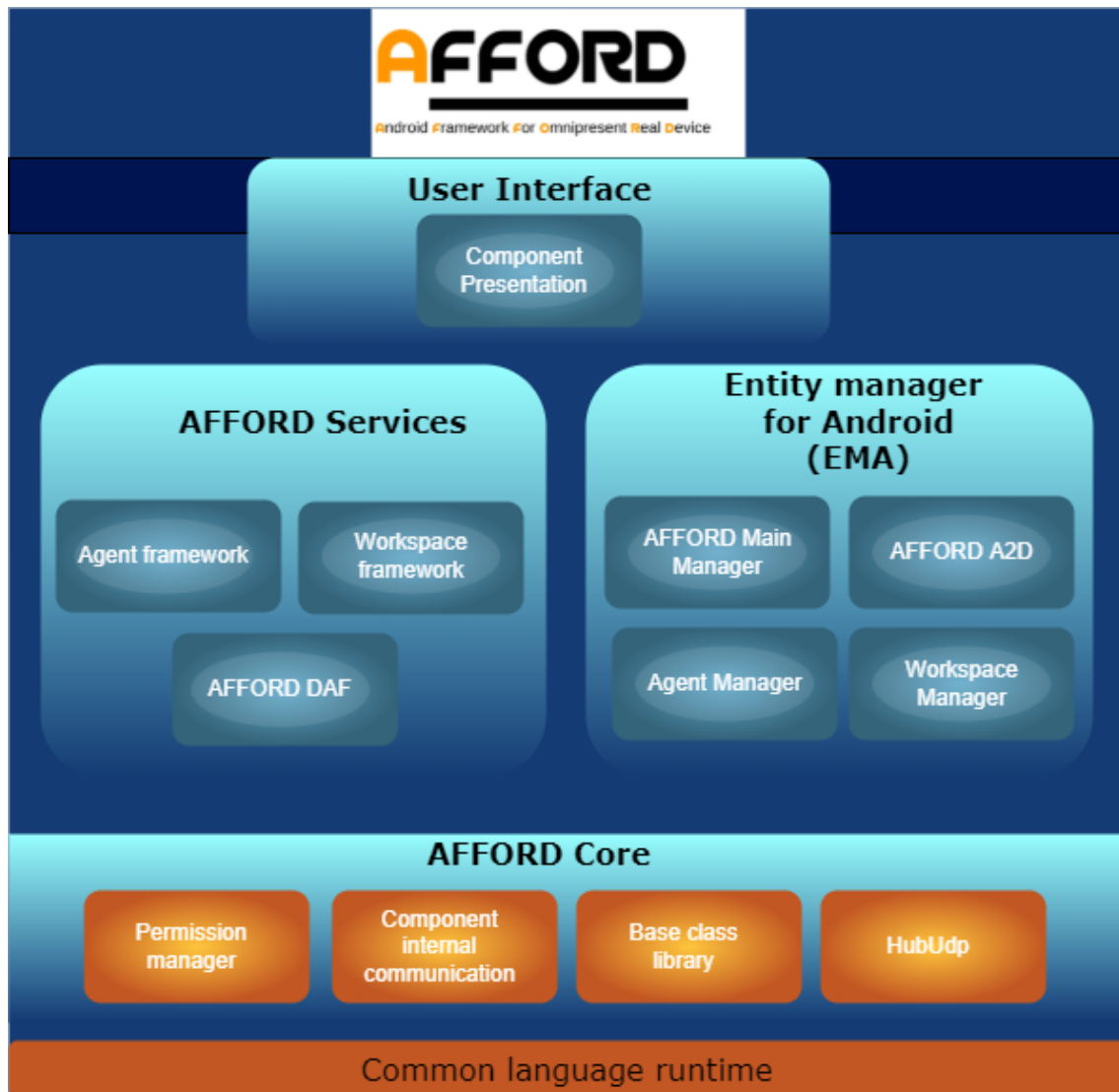


FIGURE 6.2 – Framework AFFORD

- Un BroadcastReceiver est un composant ne possédant aucune interface graphique, permettant de capter les messages émis sur le système (exemple : réception d'un nouveau SMS).
- Un composant Activity est une interface utilisateur s'exécutant au premier plan.
- Un Service est un composant effectuant des tâches en arrière-plan.
- Un ContentProvider est un composant servant d'interface entre l'application souhaitant accéder aux données et les données.

Une fois implémentés, ces composants se transmettent des paramètres grâce à des Intent. Les Intent (Intention) sont des messages asynchrones échangés

entre composants Android. Il est possible de créer ses propres Intent mais il existe aussi des intents spécifiés par Google (lancer lecteur PDF, ouvrir un navigateur Web, effectuer appel téléphonique, etc).

Une fois implémenté, l'application et les ressources qu'elle utilise sont proposées à l'utilisateur sous la forme d'un Android package (APK).

En plus des fichiers sources contenant le code et les fichiers multimédias, l'APK contient un `AndroidManifest.xml`. Ce fichier est créé par le développeur de l'application et contient les informations liées au fonctionnement de l'application telles que le nom du package de l'application, le composant Activity à lancer au lancement de l'application ou encore les permissions nécessaires. Ce dernier point concernant la sécurité est une notion importante modifiant la structure de l'implémentation. Android a ajouté au fil du temps un niveau de sécurité où chaque action externe pouvant être demandée par une application doit être déclarée dans un fichier Manifest (lire les contacts, prendre une photo, accéder à Internet, etc.). L'utilisateur peut à tout moment modifier les droits qu'il accorde à l'application.

## 6.2 AFFORD Core

Le module AFFORD Core regroupe tous les framework et bibliothèques de bas niveau du framework AFFORD. Nous présentons ici quatre modules :

- AFFORD Permission manager
- AFFORD Component Internal Communication (CIC)
- Java base class libraries
- HubUdp

**Java Base Class libraries** Au sein de cette appellation, nous regroupons toutes les bibliothèques externes utiles à notre proposition. Elle encapsule un grand nombre de fonctions basiques et essentielles pour les programmeurs, telles que la lecture et l'écriture, le rendu graphique, l'interaction avec le matériel et le système d'exploitation, la manipulation de documents, etc.

**AFFORD Permission manager** Comme nous l'avons précisé, les permissions représentent une notion importante d'Android. Au sein d'AFFORD, l'utilisateur final peut faire évoluer ces besoins au cours de la vie de l'application. En ce sens, le rôle de ce module est de gérer cette évolution en analysant et modifiant les permissions au gré des modifications souhaitées.

**AFFORD Component Internal Communication (CIC)** Au sein de notre framework, les entités sont indépendantes et sont en mesure de s'échanger des messages. Dans ce cadre, le module CIC sert à "router" les différents messages du système (BroadcastReceiver, ContentProvider, Intent).

**HubUdp** HubUDP est ici une brique de bas niveau qui permet la création de réseaux superposés. Cette partie de notre framework, est issue de la version JAVA du framework HubUdp issue de la plateforme "Software Kit for Ubiquitous Agent Development" SKUAD, développée au Laboratoire d'Informatique et de Mathématiques (LIM) de l'Université de La Réunion au sein du groupe de travail "Systèmes Collectifs Adaptatifs"[46]. Initialement conçues pour un cadre différent, des modifications ont été faites afin de l'adapter aux contraintes de la plateforme ANDROID.

## 6.3 **AFFORD Services**

Le module AFFORD Services, est le composant principal du framework AFFORD et est utile à l'implémentation des entités primaires de notre modèle ADW. En adéquation avec ce modèle, nous décrivons ce module au travers de trois sous-modules : DAF, Agent framework, Workspace framework.

### 6.3.1 **Dispositional Artifact Framework**

Avec le Dispositional Artifact Framework (DAF), le programmeur peut, en plus des primitives de base liées au comportement observable des artefacts, paramétrer l'ensemble des méthodes spécifiques liées aux contraintes d'Android



---

```

public interface SensorArtifact extends FinalArtifact {
    void setObservableState ( String state ); // Modification des valeurs observables
    void setObservableState ( byte [] state );
    byte [] getObservableState (); //Fonction accessible aux agents permettant de
        consulter la valeur courante
}
public interface ActuatorArtifact extends FinalArtifact {
    boolean genEvent (); //Fonction par défaut de demande d'action
    boolean genEvent( int ID, int new_value);
    boolean genEvent( int ID, float new_value);
    boolean genEvent( int ID, String new_value);
        [...]
}

```

---

FIGURE 6.3 – Interfaces SensorArtifact/ActuatorsArtifact

telles que la création d’IHM, les accès aux permissions et au stockage de l’appareil. Tous les artefacts dispositionnels implémentent les méthodes *exposeArtifact* et *unexposeArtifact* qui permettent respectivement de rendre disponible l’artefact sur un workspace dispositionnel. Cette demande est cependant assujettie à l’autorisation du Device Manager.

Via notre framework, le programmeur peut ainsi implémenter des artefacts dispositionnels grâce à différentes classes : *ActuatorArtifact*, *SensorArtifact*, *ServiceArtifact*.

### ActuatorArtifact

Cette classe implémente les fonctions liées aux effecteurs (FIGURE 6.3). Nous pouvons donner comme exemple la classe *Light* (FIGURE 6.4). Pour le concepteur, l’implémentation consiste à coder les différentes réactions de l’artefact aux stimuli reçus. Nous pouvons voir deux modalités d’interaction avec un artefact de type *LIGHT* (FIGURE 6.4) au travers de la fonction *genEvent*., La première avec une valeur indiquant si l’acteur souhaite allumer ou éteindre tandis que la deuxième modifie l’état de la lampe indépendamment de son état initial.

---

```

public boolean genEvent(int ID, int new_value){
    if (new_value == 0) {
        if ( flashLightStatus ) { flashLightOff (); return true ;}
    } else if (! flashLightStatus ) { flashLightOn (); return true ;}
    return false ;
}

public boolean genEvent(){
    if ( flashLightStatus ) { flashLightOff ();}
    else if (! flashLightStatus ) { flashLightOn ();}
    return true ;}
private void flashLightOn () {
    try {
        String cameraId ←cameraManager.getCameraIdList () [0];
        ...
        cameraManager.setTorchMode(cameraId, true );
        flashLightStatus ← true ;
    } catch ( CameraAccessException e) {
        ...
    }
}
private void flashLightOff () {
    try {
        String cameraId ←cameraManager.getCameraIdList () [0];
        ...
        cameraManager.setTorchMode(cameraId, false );
        flashLightStatus ← false ;
    } catch ( CameraAccessException e) {
        ...
    }
}
}

```

---

FIGURE 6.4 – Exemple d'un artefact dispositionnel de type effecteur : Light

## SensorArtifact

Cette classe implémente les fonctions liées aux capteurs (FIGURE 6.3). On peut voir un exemple d'artefacts encapsulant un capteur au travers du code de *ArtifactSensorImpl* (FIGURE 6.5). Pour cela, le concepteur n'a qu'à préciser le capteur à encapsuler, son type ainsi que le délai entre chaque prise d'information. À chaque mise à jour, l'état observable change et les données sont envoyées à tous les agents étant en lien avec l'artefact.

---

```

public class ArtifactSensorImpl extends FinalDeviceImpl implements
    SensorEventListener , SensorArtifact {
    int sensor_id ;
    byte [] value ;
    //Constructeur prenant les paramètres permettant la création de l' artefact
    public ArtifactSensorImpl ( String vname, String SENSOR_SERVICE, Integer
        SENSOR_TYPE, int SENSOR_DELAY, String Stype) {
        super(vname, SENSOR_TYPE, true);
        this .SENSOR_SERVICE←SENSOR_SERVICE;
        this .SENSOR_TYPE←SENSOR_TYPE;
        this .SENSOR_DELAY←SENSOR_DELAY;
        this .type←Stype;
    }
    // Fonction de mise a jour des informations observables de l' artefact
    @Override
    public void onSensorChanged(SensorEvent event) {
        if ( event . sensor . getType () = this .SENSOR_TYPE){
            setObservableState ( state );
        } } }
}

```

---

FIGURE 6.5 – Implémentation d'un SensorArtifact

## ServiceArtifact

Cette classe implémente les fonctions liées aux services. Elle encapsule une fonctionnalité nécessitant une réponse avec traitement. Dans le cas des services, l'artefact dispositionnel associé gère :

- l'asynchronicité liée à la dépendance d'une source tier
- La gestion des Intent

Cette gestion se déroule en 4 étapes :

1. Réception de la demande de l'acteur (agent ou humain).
2. Envoi de la demande par Intent au package correspondant
3. Attente de la réception.
4. Réception de la demande et envoi de la réponse à l'acteur.

Sur la FIGURE (6.6) nous mettons en avant un artefact dispositionnel de type service lié à la reconnaissance vocale(*VoiceRecognition*).

---

```

public MyVoiceRecognition( String vname, Activity activity ) extends ServiceArtifact {
    super(vname);
    ...
}
@Override
public synchronized Object genService ( String message){//1) Réception de la
    demande
    ...
    //2) Production de l'intent
    Intent intent ←new Intent( RecognizerIntent .ACTION_RECOGNIZE_SPEECH);
        intent .putExtra ( RecognizerIntent .EXTRA_LANGUAGE_MODEL,
            RecognizerIntent .LANGUAGE_MODEL_FREE_FORM);
    intent .putExtra ( RecognizerIntent .EXTRA_PROMPT, question);
    i ←System.identityHashCode( this );
    activity . startActivityForResult ( intent ,
        MyVoiceRecognition. this . IdService );
    this . interrupt ();
    ...
    data←new MyData();// création du token à fournir à l'agent
    ..
    return data;
}
@Override
public synchronized void SetData( int resultCode , Intent intent ){//Récéption de la
    réponse
    String message←"Vide";
    ...
    if ( resultCode = RESULT_OK){
        ArrayList <String> matches ← intent . getStringArrayListExtra (
            RecognizerIntent .EXTRA_RESULTS);//récupération du résultat
        message←matches.get(0) . toString ();
    }
    data . send( message );} //Envoie de la réponse
onUse←false;
}

```

---

FIGURE 6.6 – Implémentation de VoiceRecognition

### 6.3.2 Agent framework

Pour un programmeur, un agent possède trois fonctions principales permettant d'influencer son comportement :

- *Behavior* : Le comportement de l'agent est défini par le concepteur de l'agent. Lors de son exécution, l'agent suit ce comportement et s'appuie sur les dispositions captées afin d'interagir avec son environnement.
- *dispositionInterest* : Les dispositions recherchées par l'agent se font grâce à cette fonction. Il est possible par exemple pour l'agent de choisir entre un artefact distant ou présent en local.
- *SelectBestArtifact* : La fonction *SelectBestArtifact* permet la personnalisation des critères de sélection finale de l'agent parmi l'ensemble des dispositions détectées.

Nous pouvons voir un exemple avec l'implémentation de l'agent (FIGURE 6.7). Cet agent a pour but de faire clignoter une lampe à intervalle régulier, toutes les 500ms.

### 6.3.3 Workspace Framework

Le workspace framework se base en partie sur la brique HubUDP présente dans SKUAD, adaptée aux contraintes de l'implémentation de notre modèle sur Android. Son but est de proposer une entité réseau abstrayant les spécificités de l'environnement réel. Le framework propose un cadre logiciel afin de permettre la mise en réseau des artefacts dispositionnels. Afin d'accéder au workspace, les artefacts dispositionnels et les agents préemptifs voulant accéder aux workspaces sont soumis à une identification avec la demande d'un mot de passe. Le programmeur reste toutefois libre de modifier la politique de sécurité d'un workspace.

---

```

private void behavior () {
    if ( OpportunityList . get ( "Light" ) != null ) {
        light . genEvent () ;
        try {
            Thread . sleep ( 500 ) ;
        } catch ( InterruptedException e ) {
            e . printStackTrace () ;
        }
    }
}

public boolean dispositionInterest ( Artifact artifact ) {
    if ( artifact instanceof distantArtifact ) { // Si distant
        distantArtifact art ← ( distantArtifact ) artifact ;
        if ( art . type = "Light" ) { // Si cet artefact distant m'intéresse
            this . addNewOpportunity ( art , "Light" ) ;
            return true ; }
    }
    else if ( artifact instanceof DispositionArtifactImpl ) { // si local
        DispositionArtifactImpl art ← ( DispositionArtifactImpl ) artifact ;
        if ( art . getType () = "Light" ) {
            this . addNewOpportunity ( art , "Light" ) ;
            return true ; }
    }
    return false ; }

protected void selectArtifact () {
    if ( rolettoArtifacts . keySet () . contains ( "Light" ) &&
        opportunitytoArtifacts . get ( "Light" ) . size () > 0 ) { // Si il y a des candidats
        possibles
        boolean b ← false ;
        for ( int i ← 0 ; i < opportunitytoArtifacts . get ( "Light" ) . size () ; i ++ ) {
            ArtifactImpl light ←
                opportunitytoArtifacts . get ( "Light" ) . get ( o ) ; // Dans ce cas on
                choisit le premier trouvé dans le workspace
            if ( light . addAgentPropertyChangeListener ( this ) ) {
                OpportunityList . put ( "Light" , light ) ;
                break ; }
        }
    }
}

```

---

FIGURE 6.7 – Implémentation de l'agent *AgentBlink*

## 6.4 Entity Manager for Android (EMA)

EMA est la brique primaire de gestion des entités en interaction au sein de l'application. Ce module est divisé en quatre sous-modules :

- AFFORD Main Manager
- Android Decompositional Device (A2D)
- Agent Manager
- Workspace Manager

### 6.4.1 AFFORD Main Manager

Le AFFORD Main Manager met en place les éléments de gestion des entités proposées par ADW. Dans ce sous-module, il est possible de modifier des types d'autorisations accordés, le routage des informations, le type d'appareil géré, etc..

Dans le but de décentraliser la gestion du système, ce module délègue une partie de la charge à trois autres modules gérant respectivement les trois entités proposées par ADW : *AndroidDecompositionalDevice(A2D)*, *AgentManager*, *WorkspaceManager*.

### 6.4.2 Android Decompositional Device (A2D)

Le sous-module A2D est alloué à la décomposition de l'appareil sous la forme d'un ensemble d'artefacts dispositionnels. Cette décomposition se déroule en trois étapes :

1. Racine de l'arbre : Device Decompositional ROOT
2. Entités intermédiaires : SensorManager, EffectorManager, ServiceManager
3. Artefacts dispositionnels

#### Device Decompositional ROOT

Device Decompositional ROOT est défini en tant que racine de l'arbre de décomposition en disposition d'un appareil. À ce titre, c'est elle qui gère les créa-

tions des artefacts dispositionnels, les autorisations d'exposition sur les workspaces, la vérification des permissions accordées, les demandes d'autorisation des permissions restantes. Ce module permet l'introspection de l'appareil dans le cadre de la décomposition de l'objet sous forme de dispositions.

Cette décomposition est déléguée à trois entités : *ArtifactSensorManager*, *EffectorManager*, *ServiceManager*.

**SensorManager** SensorManager est responsable de la gestion de l'encapsulation des capteurs de l'appareil et a pour fonction principale d'analyser les possibilités de captations d'informations offertes par l'objet connecté dans le but de créer dynamiquement et automatiquement les artefacts dispositionnels associés.

---

```

public static ArtifactImplSensorsManager
CreateDefaultArtifactSensorArtifactsManager ( String name, Activity activity ) {
    ArtifactImplSensorsManager . activity ← activity ;
    SensorManager sensorManager ←(SensorManager)
        activity . getSystemService (SENSOR_SERVICE);
    ArtifactImplSensorsManager AASM ←new
        ArtifactImplSensorsManager(name,sensorManager);
    List<Sensor> list ←sensorManager. getSensorList ( Sensor .TYPE_ALL); //On liste
        l'ensemble des capteurs ( dispositions )
    if ( list != null) {
        Iterator iterator ← list . iterator ();
        while ( iterator .hasNext()) { //On parcourt la liste des dispositions de
            type capteur
            Sensor sensor ←(Sensor) iterator .next ();
            AASM.nbmax++;
            ArtifactSensorImpl devicesensor ←new
                ArtifactSensorImpl ( sensor .getName(), Context.SENSOR_SERVICE,
                sensor.getType(),
                SensorManager.SENSOR_DELAY_NORMAL,sensor.getStringType());
            //On encapsule la disposition dans un artefact
            devicesensor . setSensor ( sensor );
            devicesensor . addDeviceMaster (AASM);
        }
        return AASM;
    }
    return null ;
}

```

---

FIGURE 6.8 – Méthode de création des SensorArtifact



On peut voir un exemple de création de `sensorArtifacts` avec le manager associé dans l'exemple 6.8.

**EffectorManager** `EffectorManager` est responsable de la gestion de l'encapsulation des effecteurs de l'appareil. Le programmeur peut ajouter les types effecteurs qu'il souhaite si celui-ci n'est pas dans la liste proposée. Au lancement du programme, `EffectorManager` vérifie alors la présence des composants nécessaires à leurs réalisations avant de les rendre exposables.

**ServiceManager** La classe `ArtifactSensorManager` est responsable de la gestion de l'encapsulation des services de l'appareil. `ArtifactSensorManager` met à profit le *PackageManager* proposé par Android afin de récupérer des informations sur les applications installées sur l'appareil. Le développeur peut ainsi proposer un ensemble d'artefacts dispositionnels encapsulant des services. À partir de ces informations, si les services existent sur l'appareil (exemple : la synthèse vocale) et que les permissions accordées par l'utilisateur le permettent, les artefacts associés à leurs usages seront alors créés.

### 6.4.3 Agent Manager

Agent manager est responsable de la gestion des agents. Son rôle consiste essentiellement en un conteneur d'agent. Les agents ajoutés sont ainsi référencés au sein de l'IHM comme étant dépendants de ce manager. Il est alors à la charge de l'utilisateur de l'application finale de décider de l'activer.

### 6.4.4 Workspace Manager

Le workspace Manager est responsable de la gestion des workspaces en proposant la création de workspaces ou la connexion à des workspaces existants.

À l'usage, l'utilisateur peut rejoindre avec l'application un workspace grâce au nom du workspace en local auquel on ajoute l'IP, le port utilisé ainsi que le mot de passe dans le cadre d'un workspace distant.

## 6.5 User Interface (UI)

---

```
public interface HaveUI {  
    ViewHolder doViewHolder(ViewHolder viewHolder, View convertView, ListView  
        listView, ElementListAdapter elementListAdapter );  
    HaveUI getPrevious ();  
    void setPrevious (HaveUI prev);  
    boolean addNext(HaveUI prev);  
    boolean removeNext(HaveUI prev);  
    List getNext ();  
    void refreshIHM();  
}
```

---

FIGURE 6.9 – Méthode de personnalisation de l’interface

Il est possible pour un concepteur de personnaliser l’interface de son application. Pour cela, notre framework permet de modifier l’affichage et le comportement de chaque entité créée, quel que soit son type. Pour cela, le concepteur possède les méthodes suivantes (Figure 6.9) :

- `doViewHolder` : permettant de modifier le visuel de l’entité.
- `setPrevious` : pour gérer le comportement lors de l’appui sur la touche retour
- `AddNext/RemoveNext` : pour gérer le comportement lors du clic sur l’entité.

Par défaut, notre framework propose une présentation sous forme d’une liste dont les éléments associés à chacune des lignes sont modifiables.

## 6.6 Conclusion

Dans ce chapitre, nous avons présenté l’implémentation de notre modèle d’architecture au travers du framework AFFORD. Ce framework est divisé en quatre modules primaires :

- `AFFORDCORE` : Ensemble de modules permettant d’abstraire les couches les plus basses de l’architecture. Par exemple la gestion des autorisations,

le routage des messages entre les entités du système ou encore la mise en réseau.

- AFFORD Services : Socle d'implémentation facilitant la création des entités du système. Elle propose les différents patrons de conception nécessaires à la production des agents préemptifs, des workspaces dispositionnels et des artefacts dispositionnels.
- EMA : Ensemble de modules de gestion des entités liées au modèle d'architecture ADW.
- User Interface : Ensemble de solutions permettant la mise en place de l'interface.

En nous basant sur ce framework, nous proposons au sein du chapitre suivant l'implémentation d'une application de détection de chute : Ubi-Health.

## Chapitre 7

### Une application Android : Ubi-Health

Dans ce chapitre, nous allons décrire l'application basée sur le framework AFFORD : Ubi-Health. Cette application propose un exemple de système de détection de chute mettant à profit les différentes dispositions accessibles dans l'environnement. Nous allons, dans un premier temps, aborder les problématiques liées à la chute des personnes en perte d'autonomie. Puis, nous allons nous intéresser au fonctionnement de l'application Ubi-Health, avant de conclure par les tests réalisés avec les différents dispositifs.

#### 7.1 BeCareFall : Un agent de détection de choc

Selon un rapport de la Direction de la recherche, des études, de l'évaluation et des statistiques (DRESS), en France, une personne de plus de 65 ans sur cinq déclare avoir fait une chute dans l'année[64]. Ces chutes se font en général à domicile [94, 153] et représentent un danger pour des personnes déjà fragilisées et en situation d'isolement. Selon l'European Injury Data Base (données à l'échelle européenne), entre 2012 et 2014, les chutes ont causé 90480 décès en Europe [55]. Parmi ces chutes, 72 % concernaient des personnes de plus de 65 ans et depuis les années 2000, les années de vie perdues attribuées aux chutes ne font qu'augmenter [76, 152]. De plus, la chute à domicile est aussi la cause la plus fréquente d'admission à l'hôpital et de soins d'urgence pour des traumatismes domestiques, avec une plus grande fréquence après 65 ans et des conséquences plus graves après 85 ans [6, 176]. De ce constat émergent alors des propositions

de suivi à domicile basé sur diverses technologies telles que les cameras [52], la détection nerveuse [127], et d'autres objets connectés divers [113].

En ce qui concerne notre contexte d'application, nous allons nous concentrer sur l'utilisation de l'accéléromètre dans le cadre de la détection de chute [4, 15, 109, 110]. En termes de captation, l'accéléromètre permet de mesurer une accélération non gravitationnelle linéaire selon un axe. Dans l'usage, le terme d'accéléromètre fait donc allusion à la combinaison de trois accéléromètres qui calculent les accélérations linéaires selon trois axes orthogonaux. Dans le cadre de cette phase de preuve de concept, ce capteur est tout indiqué grâce à sa présence au sein de nombreux objets connectés où il sert en général à déterminer l'orientation de l'écran ou stabiliser la prise de photographie. En nous basant sur cette technologie, nous proposons au sein de notre application Ubi-Health l'implémentation d'un agent nommée "BeCareFall" capable d'utiliser les données d'un accéléromètre pour détecter un choc (FIGURE 7.1).

Dans le cas de "BeCareFall" nous transformons l'information tridimensionnelle issue de l'accéléromètre en une information unidimensionnelle grâce à la formule du SVM (Signal Vector Magnitude) [92] :

$$\sqrt{linear\_accX^2 + linear\_accY^2 + linear\_accZ^2}$$

Basés sur les résultats issus de cette méthode, nous avons produit de manière empirique un seuil de détection d'événement en adéquation avec notre usage illustratif. En ajustant les paramètres de sensibilité, il est en effet possible de discriminer différents types de chocs (chute, hauteurs, sens, présence d'un choc final). L'agent analyse ainsi les données de l'accéléromètre et réagit à un franchissement de seuil. Dans le cas d'une chute détectée et afin de minimiser le risque de faux positifs, "BeCareFall" propose un dialogue avec l'utilisateur. L'agent pose une suite de questions fermées donnant le choix à la personne interrogée de répondre à l'oral parmi un ensemble de réponses prédéfinies. Chaque réponse prononcée par l'utilisateur sera alors analysée par l'agent afin de faire un constat de la situation.

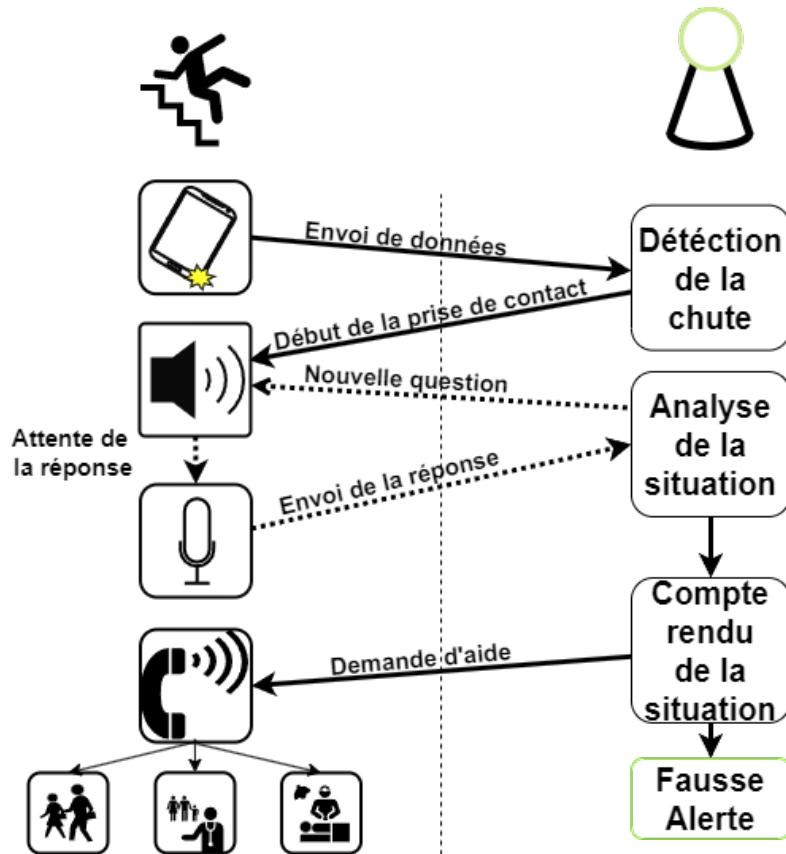


FIGURE 7.1 – Digramme illustrant le cas de la gestion de chute

## 7.2 Description de l'application Ubi-Health

Lors de sa mise en route, l'application Ubi-Health (Format apk, 9.13 Mo) détecte tous les capteurs embarqués au sein de l'appareil et propose, pour l'instant, sept effecteurs et deux services. L'application demande alors à l'utilisateur de valider les permissions qu'il souhaite autoriser. Dès lors, si le composant est disponible et que les autorisations le permettent, le système procède à la création de l'artefact dispositionnel associé.

Suite à cette mise en route de l'application, l'utilisateur peut directement naviguer, via l'interface graphique, afin de gérer indépendamment les dispositions qu'il souhaite utiliser, les agents qu'il souhaite activer ainsi que les workspaces qu'il souhaite rejoindre (Figure 7.2). Par défaut, un workspace propre à l'appareil est créé au sein de l'application. Ce workspace est principalement utilisé pour les services dont le cadre d'usage se focalise sur l'appareil porteur de l'application.

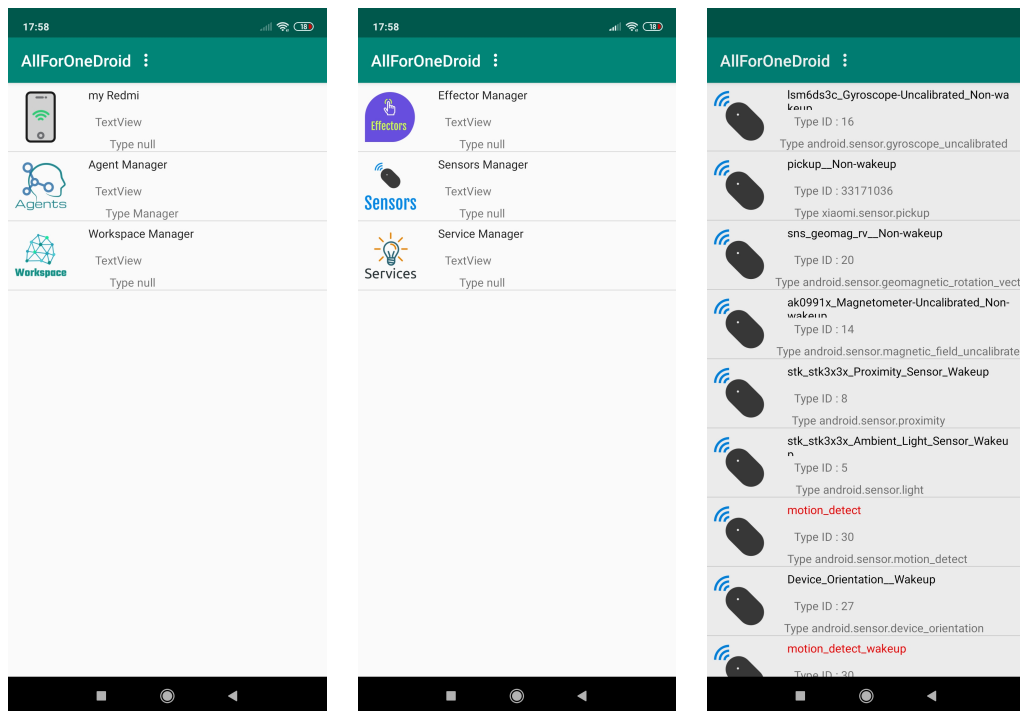


FIGURE 7.2 – Interface utilisateur de l'application Ubi-Health

Au-delà de cet usage, il est possible pour un utilisateur de créer ou de rejoindre un workspace mutualisé entre plusieurs appareils. Deux options s'offrent alors à l'utilisateur selon si le workspace qu'il souhaite rejoindre est (FIGURE 7.3) :

- Local : Il suffit de renseigner le nom du workspace à rejoindre ainsi que son mot de passe.
- Distant : En plus des éléments nécessaires vus précédemment, il faut renseigner l'adresse IP du workspace que l'on souhaite rejoindre.

Une fois les workspaces créés, l'utilisateur peut choisir quelles dispositions ou agents exposer. En ce qui concerne les dispositions, la navigation au sein de l'application se fait en suivant l'arbre de décomposition en débutant par la racine de celui-ci. Pour les agents, il suffit à l'utilisateur de permettre à l'agent de rentrer en contact avec un ou plusieurs workspaces.

Dès lors, les agents sont actifs, interagissent de manières autonomes avec les dispositions et sont à même de proposer leurs services.

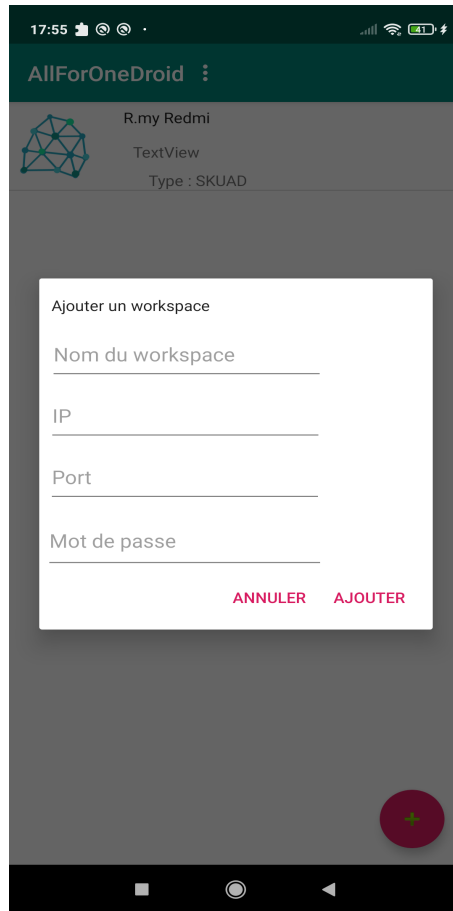


FIGURE 7.3 – Interface d'ajout d'un workspace

### 7.2.1 Exemple de fonctionnement de "BeCareFall"

Au sein de cet écosystème, il est alors possible de détailler en contexte réel le comportement global de l'agent "BeCareFall".

Dès son activation au sein d'un workspace, l'agent "BeCareFall" va rechercher de manière préemptive la présence de trois types d'artefacts

- Accéléromètre, présent au sein des smartphones et des montres connectées;
- Reconnaissance vocale, présent au sein des smartphones et des montres connectées;
- Synthèse vocale (FIGURE 7.4), présent au sein des smartphones et des montres connectées.

Comme nous pouvons le remarquer au sein du code décrit en (1) au niveau de la FIGURE 7.5, cette tâche se traduit par l'ajout au sein de la fonction *dispositionInterest*



---

```

public Speaker( String vname){
    super(vname);
    tts ←new TextToSpeech(getContext(), this);
    tts . setPitch (0.8 f); //hauteur de la voix
    tts . setSpeechRate (0.9 f); // Vitesse de lecture
    this . type←"Speaker";
}
public boolean genEvent( String text ){
    if (isReady) {
        while( tts . isSpeaking () ) {...;} //Attend que la demande précédente soit finie
        tts . speak( text , TextToSpeech.QUEUE_ADD, null, null); //Lis le texte proposé en
            paramètre
    }
}

```

---

FIGURE 7.4 – Implémentation de l’artefact encapsulant la synthèse vocale

des types d’artefacts recherchés par l’agent. Il est à noter que, l’agent peut certes utiliser les artefacts de manière indifférenciée, mais peut aussi s’adapter à la nature de l’appareil possédant l’accéléromètre (smartphone ou montre par exemple).

Une fois que les dispositions ont été trouvées, l’agent passe en phase de vigilance et commence ses analyses via *isInMotion* (en (2) sur la FIGURE 7.5) afin de déduire la présence d’une chute grâce à la formule du SVM (Signal Vector Magnitude).

Si une chute potentielle est détectée, l’agent va entamer la phase de dialogue. Pour cela, il utilise la reconnaissance vocale pour traduire des paroles en texte et le speaker pour transcrire la réponse en voix ((3) sur la FIGURE 7.5). Ces deux fonctions sont, en général, assurées par le smartphone.

Ces éléments vont permettre à l’agent d’entamer une boucle conversationnelle séparée en trois étapes :

1. L’agent pose une question via la synthèse vocale
2. L’utilisateur répond à la question via la reconnaissance vocale
3. Selon la réponse, une modification de l’état interne de l’agent est opérée.

Suite à une boucle conversationnelle, l’agent peut être amené à modifier son état et décider de :

---

```

public boolean dispositionInterest ( Artifact artifact ){// (1)
    if( artifact instanceof DistantArtifact ){//Si sur un autre appareils
        SkuadDevicetoArtifactAdapterN art ←
            (SkuadDevicetoArtifactAdapterN) artifact ;
        if( art.getType() = "android.sensor.accelerometer" ){
            this.addNewOpportunity(art,"ACCELEROMETER");
            return true;
        }
    }
    else if( artifact instanceof LocalArtifact ){// si présent sur le smartphone
        DispositionArtifactImpl art ←( DispositionArtifactImpl ) artifact ;
        if( art.getType() = "android.sensor.accelerometer" ){// Accelerometre
            this.addNewOpportunity(art,"ACCELEROMETER");
            return true;
        }
        if( art.getType() = "VoiceRecognition" ){// Reconnaissance vocale
            this.addNewOpportunity(art,"VoiceRecognition");
            return true;
        }
        if( art.getType() = "Speaker" ){//Haut parleur
            this.addNewOpportunity(art,"Speaker");
            return true;
        }
    }
    return false ; }

public boolean isInMotion(double x,double y, double z){//(2)
    double sqrt ←Math.sqrt(x * x + y * y + z * z);
    if( sqrt>seuil ) { //Un choc est détecté
        return true ;}
    return false ; }

public void PoseQuestion( String question ){// (3)
    Speaker speaker ←(Speaker) OpportunityList.get("Speaker");
    Thread thread ←new Thread() {
        public void run() {
            isSpeaking ←true;
            speaker.genEvent( question );
            while( isSpeaking ) {}
            reponse←null;
            (( ServiceImpl ) OpportunityList.get("VoiceRecognition")). genService (o,
                "Repondez par oui ou par non".getBytes ());
            while( reponse= null) {...} //Attend la réponse
            String message ←reponse;
            reponse←null;
            nextState (message);//on passe à l'état suivant
            this.interrupt () ;}};
    thread.start () ;}

```

---

FIGURE 7.5 – Implémentation de la détection du choc et de la demande d'information

- Demander des informations complémentaires en jouant une boucle conversationnelle
- Confirmer la chute et agir en contactant un aidant
- Conclure à une fausse alerte

Une fois la tâche effectuée, l'agent passe à nouveau en état d'écoute des données de l'accéléromètre afin de détecter une nouvelle chute.

### 7.3 Conclusion

Au sein de ce chapitre, nous avons proposé l'application Ubi-Health permettant :

- la création de workspace dispositionnel,
- la création automatique et l'exposition des dispositions de l'appareil,
- l'activation d'agents préemptifs.

En rapport avec notre cadre d'usage, nous proposons l'agent "BeCareFall" qui est capable de s'adapter aux dispositions proposées par l'utilisateur afin de fournir un service de gestion de chute. En ce qui concerne la captation de la chute, l'agent utilise l'accéléromètre présent dans la majorité des appareils Android. En cas de chute potentiel, "BeCareFall" interagit avec la personne concernée au travers d'une boucle conversationnelle afin de s'assurer de la nature de la situation. Si la chute est avérée, l'agent prévient alors les aidants et provoque une alerte en s'adaptant aux dispositions présentes.

Cette première phase a permis de confirmer que de nombreuses possibilités étaient insoupçonnées au sein de nos appareils Android (Smartphone et tablette, montre).

Comme exemple, nous pouvons mettre en avant le fait qu'au sein du panel d'appareils utilisés, chacun était en mesure de proposer une vingtaine d'artefacts de type Sensor.(Fig.7.1)

En plus de proposer une preuve de concept, cette mise en application, nous a ainsi permis de montrer qu'un simple smartphone ou une tablette suffit à la production de services non prévus initialement.

Marques	Samsung	Motorola	Samsung	Samsung	Xiaomi
Modèle	S8	one zoom	SM-T870	SM-N960F	Note 8T
RAM (Go)	4	4	8	6	4
Sensor	26	44	20	27	22

TABLEAU 7.1 – Quelques exemples de smartphones testés

En effet, en plus des dispositions, nous pouvons constater que l'application déploie virtuellement et schématiquement un système cyber-physique ayant plusieurs processeurs, batteries, RAM et disque dur (FIGURE 7.1). Nous pouvons l'illustrer en détaillant un sous ensemble réduit d'appareils utilisés lors de cette phase de test (FIGURE 7.2).

Cette puissance est bien supérieure à celle requise dans un usage courant et peut représenter un socle solide en termes de capacité de traitement des informations récupérées par les entités de captation.

Cela nous montre qu'il existe bien un cadre propice à la diminution du nombre de batteries, d'écrans et de composants liés au calcul au sein de ces appareils disséminés (RAM, processeur, stockage).

Modèle	Samsung galaxy s8	Galaxy Tab A (2016)	Acer TravelMate P257
Cadence Processeur	2,3 GHz	1.6GHz	2,4 GHz
Système	Android 9.0	Android 7.0	Ubuntu 16.04
RAM	4 Go	2 Go	6Go
Stockage	64 Go	16 Go	500Go
Connectivité (indicatif)	4G, NFC, Bluetooth 5.0, GPS, Wi-Fi, reconnaissance d'iris/faciale/digitale, USB Type C	USB 2.0, prise jack 3.5mm Stereo, Bluetooth v4.2, Wi-Fi 802.11, HDMI 2.0	Wi-Fi AC, Bluetooth 4.0, Webcam, Jack 3.5mm, RJ45, USB 2.0, USB 3.0, HDMI Femelle, VGA
Dimensions	148.9 x 68.1 x 8 mm pour 152 g	254.2 x 155.3 x 8.2 pour 525 g	256 x 381.6 mm x 29.2 mm pour 2.4 kg
Modèle	Diesel 2.5 Gen.4	Xiaomi-mi-box-s	Acer TravelMate P257
Cadence Processeur	1.2 GHz	1.5GHz	2,4 GHz
Système	Android wear	Android 8.1	Ubuntu 16.04
RAM	1 Go	2 Go	6Go
Stockage	4 Go	8 Go	500Go
Connectivité (indicatif)	USB 2.0, Bluetooth 4.1, Wi-Fi, GPS, Moniteur de fréquence cardiaque, Podomètre	USB 2.0, GPS, prise jack 3.5mm Stereo, Bluetooth v4.2, Wi-Fi 802.11	Wi-Fi AC, Bluetooth 4.0, Webcam, Jack 3.5mm, RJ45, USB 2.0, USB 3.0, HDMI Femelle, VGA
Dimensions	1 47 x 57 x 11 mm pour 230 g	95.25 x 95.25 x 16.7 pour 147 g	256 x 381.6 mm x 29.2 mm pour 2.4 kg

TABLEAU 7.2 – Descriptif détaillé de quelques appareils utilisés lors de la phase de test

# Chapitre 8

## Conclusion

### 8.1 Rappel du contexte

Les avancées de la médecine ont permis d'améliorer l'espérance de vie générale de la population. De cela découle une augmentation du nombre de personnes en perte d'autonomie vouée à vivre plus longtemps à domicile. Parmi cette catégorie de population fragile, nous retrouvons un panel varié de personnes souffrant de divers troubles physiques et cognitifs dont l'état de santé nécessite un suivi régulier.

Dans ce cadre, l'apport de l'informatique, de plus en plus présent au sein des objets de notre quotidien, représente un atout pour la transformation de l'habitat en un milieu d'assistance. Nous pouvons alors y voir un moyen accessible à tous afin de monitorer une personne en perte d'autonomie sans avoir besoin de personnels présents de manière permanente.

Cependant, il est difficile de proposer une solution universelle à cause des verrous liés à l'hétérogénéité du contexte, tant du point de vue des dispositifs, des profils d'utilisateur que des pathologies possibles. À cela s'ajoute le fait que la redondance des objets, qui est à ce jour la solution classique afin de rendre un service critique plus résilient, en plus d'avoir un coût, participerait à l'épuisement de nos réserves mondiales en matières premières. Il devient donc important de proposer une approche parcimonieuse avec un impact environnemental limité.

Dans ce cadre, l'originalité de notre démarche repose sur une granularisation plus fine des possibilités de notre environnement dans le but de proposer

une recomposition dynamique de services, afin d'offrir des fonctionnalités plus diversifiées que ceux qui existaient initialement.

## 8.2 Contributions

Dans cette thèse, nous nous sommes initialement intéressés aux plateformes existantes dans le cadre du suivi de la personne en perte d'autonomie. Comme nous l'avons vu au sein de l'état de l'art, les propositions actuelles ne permettent pas de répondre à l'ensemble des exigences que nous avons identifiées pour notre périmètre d'application (c.f Tableau 2.6 de la section 2.6). L'analyse des différentes approches connexes à la gestion d'environnements informatisés nous montre que la révélation des fonctionnalités sous-exploitées de nos objets connectés doit passer par une modification du cadre conceptuel.

À cet égard, nous proposons de redéfinir l'interaction entre les entités porteuses de services et les objets de l'environnement en nous inspirant du concept d'affordance afin de mettre en avant ces suggestions d'actions réalisables, définies au sein de ce manuscrit grâce à la notion de disposition.

Dans ce cadre, nous proposons de décomposer les objets connectés de notre quotidien en dispositions (ou propriété primaire de l'objet) afin de les recomposer dynamiquement dans le but de produire des services. Plus précisément, nous représentons virtuellement ces objets au travers d'un ensemble de dispositions encapsulées au sein d'artefacts dispositionnels. Une fois encapsulées, ces dispositions sont mutualisées au sein de workspaces dispositionnels et recomposées par des agents préemptifs dans le but de produire des services.

Notre proposition vise ainsi à répondre à diverses situations en proposant un meilleur réemploi des dispositions présentes au sein de l'environnement tout en assurant l'amélioration de la résilience du système

Au sein de ce manuscrit, nous avons décrit les différentes contributions liées à cette approche au travers de deux volets. Le premier concerne les aspects théoriques en lien avec la modélisation et la formalisation du concept et de l'architecture du système. Le second concerne la preuve de concept via l'implémentation de notre modèle.

Nous allons synthétiser au sein des sous-sections suivantes les quatre propositions décrites dans cette thèse : un modèle conceptuel **AVR** (Affordance Virtuelle Recompositionnelle), un modèle d'architecture générique **ADW** (Agent Disposition Workspace), un framework Java **AFFORD** (Agent Framework For Omnipresent Real Device) et une application Android (Ubi-Health).

### 8.2.1 Modèle générique

La contribution principale de ce manuscrit est la proposition et la formalisation d'un modèle conceptuel basé sur le concept d'affordance. Ce modèle propose une séparation entre le service à proposer, les objets dont sont issues les dispositions et l'espace de travail dans lequel évolue l'agent.

Ce découplage permet ainsi la recombinaison dynamique de services qui, renforcée par l'aspect ubiquitaire de la solution, octroie une plus grande adaptation aux situations rencontrées. Par rapport à la problématique posée, cette solution répond de plus au besoin de robustesse nécessaire à la proposition d'une solution utilisée au sein d'un cadre critique comme celui du suivi des personnes en perte d'autonomie.

Ce modèle, construit en accord avec le méta-modèle A&A [130], fait intervenir trois notions :

- L'agent préemptif : composant proactif du système, encapsulant l'exécution d'activités dans un environnement donné et capable de mettre à profit les dispositions d'un objet;
- L'artefact dispositionnel : composant passif du système, encapsulant une disposition et destiné à être utilisé par les agents pour soutenir leurs comportements;
- Le workspace dispositionnel : conteneur conceptuel d'artefacts dispositionnels, utile pour définir la topologie de l'environnement ainsi que sa thématization (lieu, type de service, cercle social, etc..). Ce terme est généralement simplifié par l'utilisation du mot "workspace".

Basés sur ces concepts, nous proposons une décomposition des objets sous la forme d'un ensemble d'artefacts dispositionnels mutualisés au sein de works-



paces dispositionnels. Il est alors possible pour les agents préemptifs d'entrer en contact avec ces dispositions afin de recomposer un ensemble de services. Pour cela, l'agent préemptif est capable, en plus des propriétés courantes de l'agent dans le cadre des SMA (autonomie, fonctionnement temps réel, collaboration, proactivité), de détecter, avant toute action, les affordances. Ce qui permet à l'agent d'adapter son comportement selon la situation.

### 8.2.2 Modèle d'architecture

La deuxième contribution est la proposition du modèle d'architecture Agent-Disposition-Workspace (ADW) qui définit, en plus des entités primitives, la manière dont leurs relations sont modélisées au sein du système. Dans ce cadre, nous proposons une architecture modélisée en deux parties :

**Modèle d'architecture de décomposition des dispositions** Afin de représenter un objet connecté sous la forme d'un ensemble de dispositions, nous proposons de le décomposer en suivant une structure d'arbre. Dans ce cadre, nous définissons deux classes primaires d'artefacts :

**Structural Artifact** Les classes structurelles ont pour but de rendre homogène la structure de l'arbre. Cet outil architectural permet notamment de définir la nature des éléments intermédiaires produits lors de la décomposition et qui ne peuvent donc être considérés comme des dispositions.

**Dispositional Artifact** Les artefacts dispositionnels, que l'on peut qualifier de feuilles de l'arbre de décomposition, représentent les artefacts encapsulant une disposition considérée comme atomique. Ces classes dispositionnelles sont les seules classes d'artefacts exposées à l'intérieur de workspaces dispositionnels.

La proposition de cette dichotomie entre disposition et structure vient directement en réponse à la problématique architecturale liée à la nature des étapes intermédiaires produites lors de la décomposition d'un objet. En effet, ces étapes intermédiaires ne sont pas des dispositions atomiques et ne peuvent donc être

définies au travers d'un artefact dispositionnel. Grâce à cet ajout architectural, nous proposons ainsi une cohérence dans la représentation des concepts encapsulés.

**Modèle d'architecture de recomposition préemptif des dispositions** Dans le but de modifier la nature de l'interaction entre les agents et leurs environnements, nous définissons une nouvelle architecture d'agent préemptif. La principale particularité se trouve ici dans la présence au sein de l'agent d'une brique d'analyse préemptive des affordances. Cette étape précède le déroulé des comportements et tient compte des modifications opérées au niveau de l'environnement virtuel. Ainsi, l'agent s'adapte et fait évoluer ses services tout en limitant la charge cognitive mise en oeuvre lors de l'exécution de ses comportements.

### 8.2.3 Une implémentation en Java d'un framework

#### **Android : AFFORD**

Une première phase de preuve de concept est proposée sous la forme d'une implémentation en Java d'un framework Android : Agent Framework For Omnipresent Real Device (AFFORD).

Ce framework fournit un ensemble d'outils et de primitives nécessaires à la création et la gestion d'artefacts dispositionnels, d'agents pré-emptif et de workspaces dispositionnels.

Le but est de proposer aux concepteurs et aux programmeurs un outil logiciel permettant la mise en place de notre modèle en environnement réel.

Au sein de nos travaux, nous faisons une démonstration d'usage de ce framework dans le cadre de notre application Android : Ubi-Health.

### 8.2.4 Application Android : Ubi-Health

Afin de mettre à profit le potentiel positif engendré par notre proposition, une mise en contexte de notre framework AFFORD a été effectuée au travers d'une application Android de détection de chute : Ubi-Health.

Ubi-Health est une application à installer qui permet de gérer et contrôler les dispositions d'un appareil Android dans une seule et même interface. L'outil permet de faire communiquer et interagir de façon autonome et intelligente les dispositions révélées et les agents porteurs de services.

Dans ce manuscrit, nous présentons comme exemple, l'agent "BeCareFall" capable de détecter une chute grâce à la présence d'un accéléromètre, présent au sein de nos smartphones et de nos montres connectées. En cas de suspicion de chute, l'agent mobilise le haut-parleur et le microphone d'un smartphone afin de proposer une boucle conversationnelle permettant de juger de l'urgence de la situation. Enfin, une fois la chute détectée un signal est alors transmis selon les dispositions présentes à l'intérieur de ses workspaces.

Grâce à cette démonstration, nous montrons ainsi qu'il est possible, sans ajout systématique de nouveaux dispositifs et avec des objets de tous les jours, de fournir un ensemble de services de surveillance multimodale non prévus initialement.

### 8.3 Couverture des exigences

Dans le cadre de nos travaux, nous avons mis en avant différentes contributions permettant la mise en place d'un suivi de vie diffus, autonomes et capables de tirer profit d'un maximum d'objets connectés présents dans l'environnement. Afin d'évaluer les avancées liées à nos travaux, nous allons faire un parallèle avec les exigences définies au début de cette thèse et les solutions apportées par notre approche à chacune d'entre-elles :

- [C1] **Le workspace dispositionnel en tant que support à l'ubiquité** L'ubiquité est une caractéristique nécessaire pour la mise en place d'un suivi discret. Dans le but d'assurer l'accès permanent aux dispositifs, nous proposons la notion de workspace dispositionnel. Au sein de notre proposition, le workspace dispositionnel agit ainsi en tant que structure virtuelle permettant l'interaction avec toute sorte de disposition, en local ou à distance.

- [C2] **Adaptation aux contextes via le concept d’affordance** En nous basant sur le concept d’affordance, nous proposons un système qui est à même de s’adapter de manière préemptive à la présence ou à la disparition d’opportunités dans son milieu. Ce choix dans la conceptualisation de l’interaction entre un agent et son environnement permet au système de s’affranchir de la nécessité d’un ensemble de dispositifs minimal pour fonctionner.
- [C3] **Une décomposition éco-conceptive des objets** Concernant notre engagement à la production d’une solution éco-conceptive, nous proposons dans cette thèse une approche basée sur une décomposition de nos objets sous la forme de fonctionnalités primaires (dispositions). Cette démarche favorise ainsi la mutualisation des dispositifs présents dans notre environnement et propose une meilleure utilisation des outils déjà disponibles dans le but de diminuer la redondance des composants.
- [C4] **Une interaction naturelle basée sur la recomposition de service** Basés sur la décomposition vue précédemment, nous proposons une recomposition dynamique des services en utilisant comme substrat les médias voulus et acceptés par les acteurs rencontrés. L’interaction avec le système peut ainsi être adaptée aux habitudes de vie des personnes au travers d’interactions naturelles.
- [C5] **Un déploiement facilité** Le fait de proposer un déploiement rapide sur un maximum d’appareils existants est une notion présente tout au long du processus de réflexion de cette thèse. Cette attention a été menée depuis la définition du modèle conceptuel, qui n’impose aucune modification des objets connectés, en passant par notre volonté de proposer un framework réutilisable, jusqu’au choix d’Android en tant que plateforme la plus répandue.
- [C6] **Une autonomie grâce à l’agent** L’utilisation d’agent permet au système de fonctionner en toute autonomie sans la présence systématique d’un opérateur. Ce choix permet de plus la proposition d’une solution en adéquation avec la nature distribuée des systèmes d’intelli-

gence ambiante. Nous rappelons toutefois que le rôle de cet agent n'est pas de se substituer aux aidants, mais bien d'être un support supplémentaire au suivi.

- [C7] **Une plateforme non liée à un ensemble de pathologie** Le découplage proposé par notre système permet la mise en place de services, sans présupposer de la nature des pathologies à prendre en charge. Il est ainsi tout à fait possible de faire évoluer l'environnement de la personne selon l'évolution des pathologies des patients. De plus, nous montrons au travers de notre proposition que cette évolution n'est pas forcément liée à un ajout de dispositifs, mais peut être effectuée de manière uniquement logicielle, par l'ajout d'agents.

En conclusion, nous pouvons affirmer que les exigences définies sont satisfaites, dans les limites et selon les hypothèses exposées dans le cadre de cette thèse.

## 8.4 Discussions

Le modèle générique **ADW** permet de révéler des dispositions existantes, de les mutualiser et de les exploiter afin de proposer des services. Bien qu'il réponde aux contraintes que nous nous sommes fixés, il existe des points que nous n'avons pas abordés dans ce manuscrit et qu'il nous semble important d'exposer ici.

### 8.4.1 Normalisation des dispositions encapsulées

Au sein de notre approche actuelle, rien n'assure que pour une même disposition, la dénomination proposée par l'artefact soit en adéquation avec celle de l'agent. En effet, les artefacts dispositionnels se basent sur une hypothèse forte concernant leur normalisation. Cela peut poser un problème si le workspace contient des artefacts ou des agents issus de concepteurs différents. De ce fait, un agent peut se retrouver au contact de workspace où une même disposition peut être définie de manière différente. Sur ce point, nous pensons que les fonctions offertes par les artefacts seraient enrichies par l'ajout d'une norme commune.

### 8.4.2 Utilisation d'une ontologie

De nombreuses recherches ont démontré l'intérêt des ontologies dans le cadre de l'exploitation des objets par un logiciel [65, 134, 148]. La proposition d'une ontologie serait donc tout indiquée dans le cas d'une mutualisation pour permettre une description plus poussée des possibilités offertes par les artefacts.

### 8.4.3 Caractérisation des effectivités de l'agent

Au sein de notre proposition, les agents ne possèdent pas de modélisation de la décomposition de leurs effectivités. Cette caractéristique pousse le concepteur à prévoir, sans notion de priorité ou de dépendance, les modalités d'interaction possibles entre l'agent et les dispositions. Une décomposition des effectivités pourrait permettre de les structurer afin de faciliter les choix d'affordances de l'agent. Sur ce point, une piste serait de s'inspirer des travaux récents sur la décomposition des comportements d'un agent [165].

## 8.5 Perspectives de recherche

Les résultats obtenus dans cette thèse ouvrent des voies de recherche concernant la gestion de notre habitat connecté. Nous terminerons donc avec un récapitulatif des perspectives possibles, à courts et moyens termes, concernant nos travaux.

**Extension à d'autres usages** ADW est un modèle générique et son usage est possible au travers de divers :

- **Cadres de recherches**

ADW peut enrichir d'autres approches visant à mettre à disposition les fonctionnalités des objets connectés, comme c'est le cas pour le WoT (Web of Thing).

On peut, par exemple, y voir un moyen d'améliorer la résilience d'un service en proposant des alternatives locales ou encore désengorger le

réseau en permettant d'exécuter des services en périphérie (Edge computing [159]).

— **Cadres applicatifs**

Dans cette thèse, nous nous sommes focalisés sur la surveillance à domicile des personnes en perte d'autonomie. Or il est tout à fait possible de proposer d'autres types de services et étendre le framework AFFORD à des usages différents.

— **Types d'appareils**

Pour notre phase d'implémentation, nous nous sommes focalisés sur un usage sur Android. Cependant, en s'appuyant sur la démocratisation de Java, il serait intéressant de proposer un élargissement des types d'appareils utilisés. On pourrait, par exemple, exploiter d'autres types de supports, tels que les micro-ordinateurs, les microcontrôleurs, ou d'autres types de capteurs (raspberry-pi, arduino, module d'assistant vocal, etc).

**Étude quantitative** Une étude des critères d'évaluation en termes de gains effectifs de composants lors des mutualisations permettrait de quantifier l'apport environnemental de l'approche. Ainsi, nous pourrions mettre en avant l'écart entre les solutions actuelles et notre proposition en ce qui concerne le nombre d'appareils utilisés. Cette quantification pourrait aussi donner des informations sur les dispositifs les plus intéressants à mutualiser. Ces résultats pourraient, à terme, répondre à un ensemble de questions lié au choix des dispositifs à privilégier au sein d'un habitat.

**Apprentissage** La proposition actuelle ne propose pas de méthode d'apprentissage des dispositions utilisables par l'agent. Une perspective serait ainsi d'étudier les différentes options possibles afin de permettre la mise à jour des dispositions adéquates à la réalisation d'une effectivité.

**Usage en contexte réel** Afin de produire des services effectifs et d'affiner les informations liées au cadre d'usage, il serait possible d'étendre la phase de test en exploitant un cadre réel riche tel que celui d'un living lab. Une proposition de ce

type, qui n'a pas encore abouti à cause de l'épidémie de COVID, avait été mise en place dans cette thèse au travers d'un projet au sein du Laboratoire Domus (Université de Sherbrooke), financé et sélectionné par le Fonds de recherche du Québec [48].

À la vue de la stabilisation du contexte sanitaire, nous poursuivrons cette volonté et espérons avoir l'opportunité d'intervenir dans des contextes applicatifs divers afin d'enrichir les usages de notre modèle.





# Bibliographie

- [1] Emile Aarts and Reiner Wichert. Ambient intelligence. In *Technology Guide*, pages 244–249. Springer, 2009.
- [2] Giovanni Acampora, Diane J Cook, Parisa Rashidi, and Athanasios V Vasilakos. A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*, 101(12) :2470–2494, 2013.
- [3] Zoubida Afoutni. *Un modèle multi-agents pour la représentation de l'action située basé sur l'affordance et la stigmergie*. PhD thesis, Université de la Réunion, 2015.
- [4] Mark V Albert, Konrad Kording, Megan Herrmann, and Arun Jayaraman. Fall classification by machine learning using mobile phones. *PLoS one*, 7(5) :e36556, 2012.
- [5] Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare : A survey. *Computer Networks*, 54(15) :2688–2710, 2010.
- [6] Anne Felicia Ambrose, Geet Paul, and Jeffrey M Hausdorff. Risk factors for falls among older adults : a review of the literature. *Maturitas*, 75(1) :51–61, 2013.
- [7] Athanasios Anastasiou, Paul Quarrie, and M. Boulos. Personal location aware healthcare in europe-the challenges from prototype to product : the caalyx experience. *eHealth International Journal*. 2009, 12 :18–29, 2009.

- [8] Emmanuel Andrès, Samy Talha, A Benyahia, Olivier Keller, Mohamed Hajjam, A Moukadem, A Dieterlen, J Hajjam, S Ervé, and A Hajjam. Expérimentation d'une plateforme de détection automatisée des situations à risque de décompensation cardiaque (plateforme e-care) dans une unité de médecine interne. *La Revue de Médecine Interne*, 37(9) :587–593, 2016.
- [9] Emmanuel Andrès, Samy Talha, A Ahmed Benyahia, Olivier Keller, M Hajjam, A Moukadem, A Dieterlen, Jawad Hajjam, S Ervé, and A Hajjam. E-health : A promising solution for optimizing management of chronic diseases. example of the national e-health project e-care based on an e-platform in the context of chronic heart failure. *European Research in Telemedicine/La Recherche Européenne en Télémédecine*, 4(3) :87–94, 2015.
- [10] E. Andrès, M. Hajjam, S. Talha, L. Meyer, N. Jeandidier, J. Hajjam, S. Ervé, A.-A. Zulfiqar, and A. Hajjam. Télémédecine dans le domaine de l'insuffisance cardiaque. : État des lieux et focus sur le projet de télémédecine 2.0 e-care. perspectives dans le domaine de la diabétologie. *Médecine des Maladies Métaboliques*, 12(2) :224–231, 2018.
- [11] Stefan D Anker, Friedrich Koehler, and William T Abraham. Telemedicine and remote management of patients with heart failure. *The Lancet*, 378(9792) :731–739, 2011.
- [12] Thierry Antoine-Santoni. *Des réseaux de capteurs sans fil à l'intelligence ambiante dans le suivi environnemental. Synthèse de travaux*. PhD thesis, Université de Corse Pasquale Paoli, 2019.
- [13] Jay E Aronson, Ting-Peng Liang, and Richard V MacCarthy. *Decision support systems and intelligent systems*, volume 4. Pearson Prentice-Hall Upper Saddle River, NJ, USA :, 2005.
- [14] Juan Augusto Wrede. Ambient intelligence : The confluence of ubiquitous/pervasive computing and artificial intelligence. *Intelligent Computing Everywhere*, 01 2007.

- [15] Fabio Bagala, Clemens Becker, Angelo Cappello, Lorenzo Chiari, Kiamar Aminian, Jeffrey M Hausdorff, Wiebren Zijlstra, and Jochen Klenk. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PloS one*, 7(5) :e37062, 2012.
- [16] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4) :263–277, 2007.
- [17] Deena M Barakah and Muhammad Ammad-uddin. A survey of challenges and applications of wireless body area network (wban) and role of a virtual doctor server in existing architecture. In *Intelligent Systems, Modelling and Simulation (ISMS), 2012 Third International Conference on*, pages 214–219. IEEE, 2012.
- [18] Genevieve Baudoin, Olivier Venard, Gérard Uzan, A Paumier, and J Cesbron. Le projet rampe : système interactif d’information auditive pour la mobilité des personnes aveugles dans les transports publics. In *Proceedings of the 2nd French-speaking conference on Mobility and ubiquity computing*, pages 169–176, 2005.
- [19] Fabio Bellifemine, Giovanni Caire, Agostino Poggi, and Giovanni Rimassa. Jade : A software framework for developing multi-agent applications. lessons learned. *Information and Software Technology*, 50(1-2) :10–21, 2008.
- [20] Mariette Beyeler. *Métamorphouse : transformer sa maison au fil de la vie*. PPUR Presses polytechniques, 2014.
- [21] Marc Böhlen and Hans Frei. Ambient intelligence in the city overview and new perspectives. In *Handbook of ambient intelligence and smart environments*, pages 911–938. Springer, Boston, MA, 2010.
- [22] Olivier Boissier, Rafael H Bordini, Jomi F Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6) :747–761, 2013.

- [23] Frédéric Bordage. Study - The environmental footprint of the digital world, 2019.
- [24] Amé Lie Boudreault, Julie Bouchard, Bruno Bouchard, and Shawn Whittom-Ross. Validation of a smart stove for traumatic brain injury patients in a cooking task. In *2016 12th International Conference on Intelligent Environments (IE)*, pages 230–233. IEEE, 2016.
- [25] Nasreddine Bouhaï and Imad Saleh. *Internet of things : evolutions and innovations*. John wiley & sons, 2017.
- [26] Maged Boulos, Artur Rocha, Angelo Martins, Manuel Vicente, Armin Bolz, Robert Feld, Igor Tchoudovski, Martin Braecklein, John Nelson, Gearóid Ò Laighin, Claudio Sdogati, Francesca Cesaroni, Marco Antomarini, Angela Jobes, and Mark Kinirons. CAALYX : a new generation of location-based services in healthcare. *International Journal of Health Geographics*, 6(1) :9, 2007.
- [27] Massimo Busuoli and Emanuela Pipitone. Oldes. <http://www.oldes.eu/>. [ Visité le : 2018-09-30].
- [28] Davide Calvaresi, Daniel Cesarini, Paolo Sernani, Mauro Marinoni, Aldo Franco Dragoni, and Arnon Sturm. Exploring the ambient assisted living domain : a systematic review. *Journal of Ambient Intelligence and Humanized Computing*, 8(2) :239–257, 2017.
- [29] LM Camarinha-Matos, Hamideh Afsarmanesh, et al. Telecare : Collaborative virtual elderly care support communities. *The Journal on Information Technology in Healthcare*, 2(2) :73–86, 2004.
- [30] Emmanuelle Cambois and Jean-Marie Robine. L’allongement de l’espérance de vie en europe. quelles conséquences pour l’état de santé. *Revue européenne des sciences sociales. European Journal of Social Sciences*, 55(55-1) :41–67, 2017.

- [31] Anthony Chandor, John Graham, Robin Williamson, et al. *Dictionary of computers*. Eng.] Penguin Books, 1970.
- [32] Raúl Chávez-Santiago, Ali Khaleghi, Ilangko Balasingham, and Tor A Ramstad. Architecture of an ultra wideband wireless body area network for medical applications. In *2009 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pages 1–6. IEEE, 2009.
- [33] Anthony Chemero. An outline of a theory of affordances. *Ecological psychology*, 15(2) :181–195, 2003.
- [34] Kim On Chin, Kim Soon Gan, Rayner Alfred, Patricia Anthony, and Dickson Lukose. Agent architecture : An overviews. *Transactions on science and technology*, 1(1) :18–35, 2014.
- [35] Michael H Coen et al. Design principles for intelligent environments. *the proceedings of National Conference on Artificial Intelligence AAAI/IAAI*, 547 :554, 1998.
- [36] Jean-Marc Commenge. Big data et intelligence artificielle pour le génie des procédés, 2020.
- [37] European Commission. Eu commission, critical raw materials, internal market, industry, entrepreneurship and smes. [https://ec.europa.eu/growth/sectors/raw-materials/specific-interest/critical\\_en](https://ec.europa.eu/growth/sectors/raw-materials/specific-interest/critical_en), may 2020. [ Visité le : 2021-01-13].
- [38] Domenico Conforti, Domenico Costanzo, Christos Biniaris, and Andrew Marsh. Biomedical data acquisition and transmission infrastructure in the heartfaid platform. In *Medical Measurements and Applications, 2009. MeMeA 2009. IEEE International Workshop on*, pages 127–131. IEEE, 2009.

- [39] Diane J Cook, Michael Youngblood, Edwin O Heierman, Karthik Gopalratnam, Sira Rao, Andrey Litvin, and Farhan Khawaja. Mavhome : An agent-based smart home. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003).*, pages 521–524. IEEE, 2003.
- [40] Juan M Corchado, Dante I Tapia, and Javier Bajo. A multi-agent architecture for distributed services and applications. *International Journal of Innovative Computing, Information and Control*, 8(4) :2453–2476, 2012.
- [41] Joëlle Coutaz and Laurence Nigay. Multimodalité et plasticité des interfaces homme-machine en informatique ambiante : concepts et espaces de conception, 2012.
- [42] Cecile KM Crutzen. Invisibility and the meaning of ambient intelligence. *The International Review of Information Ethics*, 6 :52–62, 2006.
- [43] Vassilis Cutsuridis, Amir Hussain, and John G Taylor. *Perception-action cycle : Models, architectures, and hardware*. Springer Science & Business Media, 2011.
- [44] Mauro AA da Cruz, Joel José PC Rodrigues, Jalal Al-Muhtadi, Valery V Korotaev, and Victor Hugo C de Albuquerque. A reference model for internet of things middleware. *IEEE Internet of Things Journal*, 5(2) :871–883, 2018.
- [45] Sajal K Das, Diane J Cook, Amiya Battacharya, Edwin O Heierman, and Tze-Yun Lin. The role of prediction algorithms in the mavhome smart home architecture. *IEEE Wireless Communications*, 9(6) :77–84, 2002.
- [46] Payet Denis. Skuad, software kit for ubiquitous agent development. <http://skuad.onover.top/>, 2018. [ Visité le : 2019-09-30].
- [47] Jasbir S Dhaliwal and Izak Benbasat. The use and effects of knowledge-based system explanations : theoretical foundations and a framework for empirical evaluation. *Information systems research*, 7(3) :342–362, 1996.

- [48] Laboratoire domus [visité le 17/06/2022]. <https://domus.recherche.usherbrooke.ca/>.
- [49] Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina. Microservices : yesterday, today, and tomorrow. *Present and ulterior software engineering*, pages 195–216, 2017.
- [50] Alexis Drogoul. *De la simulation multi-agents à la résolution collective de problèmes*. PhD thesis, Thesis at University of Paris IV, 1993.
- [51] Alexis Drogoul. Systèmes multi-agents situés. *Mémoire d’habilitation à diriger des recherches (habilitation thesis)*, 17, 2000.
- [52] Amandine Dubois and François Charpillet. Automatic fall detection system with a rgb-d camera using a hidden markov model. In *International Conference on Smart Homes and Health Telematics*, pages 259–266. Springer, 2013.
- [53] Ken Ducatel, Union européenne. Technologies de la société de l’information, Union européenne. Institut d’études de perspectives technologiques, and Union européenne. Société de l’information conviviale. Scenarios for ambient intelligence in 2010 (final report), 2001.
- [54] Bruce Edmonds and Ruth Meyer. Simulating social complexity. In *A Handbook*. Springer, 2013.
- [55] Eurosafe. Injuries in the european union. summary of injury statistics for the years 2012-2014. rapport technique european association for injury prevention and safety promotion. [https://www.eurosafe.eu.com/uploads/inline-files/IDB\\_Report\\_2014\\_final%202010-2012.pdf](https://www.eurosafe.eu.com/uploads/inline-files/IDB_Report_2014_final%202010-2012.pdf), 2016. [ Visité le : 2018-03-27].
- [56] J. Ferber. *Les systèmes multi-agents : vers une intelligence collective*. I.I.A. Informatique intelligence artificielle. InterEditions, France, 1995.



- [57] Jacques Ferber. Les systèmes multi-agents. *Paris : Inter ditions*, 1995.
- [58] Jacques Ferber. Les systèmes multi-agents : un aperçu général. *Technique et Science Informatiques*, vol.16 n°8, 1997.
- [59] Jacques Ferber. *Multi-agent systems : an introduction to distributed artificial intelligence*, volume 1. Addison-Wesley Reading, 1999.
- [60] N Firdausiyah, E Taniguchi, and AG Qureshi. Modeling city logistics using adaptive dynamic programming based multi-agent simulation. *Transportation Research Part E : Logistics and Transportation Review*, 125 :74–96, 2019.
- [61] R Fontaine, N Aky, R Courdier, and D Payet. Vers une utilisation éco responsable des objets connectés par la mutualisation de leurs composants physiques : Une approche basée sur le concept d’artefact. In *Actes de la conférence nationale en intelligence artificielle CNIA - PFLA 2020, Angers, 2020.*, volume 23, page 38, 2020.
- [62] Richard Fontaine, Rémy Courdier, and Denis Payet. Vers une utilisation éco responsable des objets connectés par la mutualisation de leurs composants physiques : Une approche basée sur le concept d’artefact. *Revue Ouverte d’Intelligence Artificielle*, 3(3-4) :393–413, 2022.
- [63] Fabrice Forest, Nathalie Candito, and Élisabeth Shimells. L’introduction des rfid dans les musées. expérimentation de l’intelligence ambiante dans les dispositifs de médiation. *Publications du musée des Confluences*, 2(1) :85–104, 2008.
- [64] Nathalie Fourcade et al. L’état de santé de la population en france. *Suivi des objectifs annexés à la loi de santé publique*. Paris : DRESS, 2017.
- [65] Artur Freitas, Alison R Panisson, Lucas Hilgert, Felipe Meneguzzi, Renata Vieira, and Rafael H Bordini. Integrating ontologies with multi-agent systems through cartago artifacts. In *2015 IEEE/WIC/ACM Inter-*

- national Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 2, pages 143–150. IEEE, 2015.
- [66] Sebastian Fudickar and Bettina Schnor. Kopal—a mobile orientation system for dementia patients. In *International Conference on Intelligent Interactive Assistance and Mobile Multimedia Computing*, pages 109–118. Springer, 2009.
- [67] Sebastian JF Fudickar, Sebastian Faerber, and Bettina Schnor. Kopal appointment user-interface : an evaluation with elderly. In *Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments*, pages 1–6, 2011.
- [68] Óscar García, Dante I Tapia, Alberto Saavedra, Ricardo S Alonso, and Israel García. Alz-mas 2.0; a distributed approach for alzheimer health care. In *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, pages 76–85. Springer, 2009.
- [69] Benjamin Gateau, Yannick Naudet, and Jarogniew Rykowski. Ontology-based smart iot engine for personal comfort management. In *2016 11th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, pages 35–40, Thessaloniki, Greece, 2016. IEEE, IEEE.
- [70] James J Gibson. The theory of affordances. *Hilldale, USA*, 1(2) :56–60, 1977.
- [71] James J Gibson. *The ecological approach to visual perception : classic edition*. Psychology Press, 1986.
- [72] Sylvain Giroux, Jérémy Bauchet, Hélène 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.

- [73] Sylvain Giroux and H el ene Pigot. *From Smart Homes to Smart Care : ICOST 2005, 3rd International Conference on Smart Homes and Health Telematics*, volume 15. IOS Press, 2005.
- [74] Bego na Gros and Francisco J Garc a-Pe alvo. Future trends in the design strategies and technological affordances of e-learning. *Learning, Design, and Technology : An International Compendium of Theory, Research, Practice, and Policy*, pages 1–23, 2016.
- [75] Illapha Cuba Gyllensten, Alberto G Bonomi, Kevin M Goode, Harald Reiter, Joerg Habetha, Oliver Amft, John GF Cleland, et al. Early indication of decompensated heart failure in patients on home-telemonitoring: a comparison of prediction algorithms based on daily weight and noninvasive transthoracic bio-impedance. *JMIR medical informatics*, 4(1) :e4842, 2016.
- [76] Juanita A Haagsma, Nicholas Graetz, Ian Bolliger, Mohsen Naghavi, Hideo Higashi, Erin C Mullany, Semaw Ferede Abera, Jerry Puthenpurakal Abraham, Koranteng Adofo, Ubai Alsharif, et al. The global burden of injury : incidence, mortality, disability-adjusted life years and time trends from the global burden of disease study 2013. *Injury prevention*, 22(1) :3–18, 2016.
- [77] Joerg Habetha. The myheart project–fighting cardiovascular diseases by prevention and early diagnosis. In *Conf Proc IEEE Eng Med Biol Soc*, pages 6746–6749, 2006.
- [78] J Richard Harrison, Zhiang Lin, Glenn R Carroll, and Kathleen M Carley. Simulation modeling in organizational and management research. *Academy of management review*, 32(4) :1229–1245, 2007.
- [79] J. Henriques, P. Carvalho, S. Paredes, T. Rocha, J. Habetha, M. Antunes, and J. Morais. Prediction of heart failure decompensation events by trend analysis of telemonitoring data. *IEEE Journal of Biomedical and Health Informatics*, 19(5) :1757–1769, 2015.

- [80] Bryan Horling and Victor Lesser. A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19(4) :281–316, 2004.
- [81] Jomi Fred Hübner, Laurent Vercouter, and Olivier Boissier. Instrumenting multi-agent organisations with artifacts to support reputation processes. In *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*, pages 96–110. Springer, 2008.
- [82] IEEE. *Middleware and application adaptation requirements and their support in pervasive computing*, Los Alamitos, CA, USA, 2003. IEEE Computer Society.
- [83] INSEE. Population par âge : Tableaux de l'économie française insee. <https://www.insee.fr/fr/statistiques/4277619?sommaire=4318291&q=Population+par+%C3%A2ge>, 2020. [ Visité le : 2021-07-02.
- [84] Hiroshi Ishii. The tangible user interface and its evolution. *Communications of the ACM*, 51(6) :32–36, 2008.
- [85] Hiroshi Ishii and Brygg Ullmer. Tangible bits : towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 234–241. ACM, 1997.
- [86] Nicholas R Jennings. On agent-based software engineering. *Artificial intelligence*, 117(2) :277–296, 2000.
- [87] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan : Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.
- [88] Peter Kaiser, Eren E Aksoy, Markus Grotz, and Tamim Asfour. Towards a hierarchy of loco-manipulation affordances. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2839–2846. IEEE, 2016.

- [89] Peter Kaiser and Tamim Asfour. Autonomous detection and experimental validation of affordances. *IEEE Robotics and Automation Letters*, 3(3) :1949–1956, 2018.
- [90] Peter Kaiser, Nikolaus Vahrenkamp, Fabian Schültje, Júlia Borràs, and Tamim Asfour. Extraction of whole-body affordances for loco-manipulation tasks. *International Journal of Humanoid Robotics*, 12(03) :1550031, 2015.
- [91] Achilles Kameas and Ioannis Calemis. Pervasive systems in health care. In Hideyuki Nakashima, Hamid Aghajan, and Juan Carlos Augusto, editors, *Handbook of Ambient Intelligence and Smart Environments*, pages 315–346. Springer US, Boston, MA, 2010.
- [92] Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine*, 10(1) :156–167, 2006.
- [93] Nataliia B Kashchena, Dmytro Solokha, Nataliia Trushkina, Leonid Potemkin, and Rahima Mirkurbanova. Use of multi-agent simulation modeling for predicting the sales of wholesale trade companies. *Journal of Management Information and Decision Sciences*, 22(4) :483–488, 2019.
- [94] Michael D Keall, David Ormandy, and Michael G Baker. Injuries associated with housing conditions in europe : a burden of disease study based on 2004 injury data. *Environmental Health*, 10(1) :1–10, 2011.
- [95] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. Future internet : the internet of things architecture, possible applications and key challenges. In *2012 10th international conference on frontiers of information technology*, pages 257–260. IEEE, 2012.

- [96] Spyros Kitsiou, Guy Paré, Mirou Jaana, and Ben Gerber. Effectiveness of mhealth interventions for patients with diabetes : an overview of systematic reviews. *PloS one*, 12(3) :e0173160, 2017.
- [97] Kristian Ellebæk Kjær. A survey of context-aware middleware. In *Proceedings of the 25th Conference on IASTED International Multi-Conference : Software Engineering, SE'07*, page 148–155, USA, 2007. ACTA Press.
- [98] Franziska Klügl. Using the affordance concept for model design in agent-based simulation. *Annals of Mathematics and Artificial Intelligence*, 78(1) :21–44, 2016.
- [99] Suveen Kumar, Jai Gopal Sharma, Sagar Maji, and Bansi Dhar Malhotra. Nanostructured zirconia decorated reduced graphene oxide based efficient biosensing platform for non-invasive oral cancer detection. *Biosensors and Bioelectronics*, 78 :497–504, 2016.
- [100] Kyung Sup Kwak, Sana Ullah, and Niamat Ullah. An overview of ieee 802.15. 6 standard. In *2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010)*, pages 1–6. IEEE, 2010.
- [101] Jöel Kwan, Yassine Gangat, Denis Payet, and Rémy Courdier. A agentified use of the internet of things. In *Full Paper in 9th IEEE International Conference on Internet of Things (iThings 2016)*. IEEE CS, 2016.
- [102] Sofiane Labidi and Wided Lejouad. *De l'intelligence artificielle distribuée aux systèmes multi-agents*. PhD thesis, INRIA, 1993.
- [103] Gokce B Laleci, Asuman Dogac, Mehmet Olduz, Ibrahim Tasyurt, Mustafa Yuksel, and Alper Okcan. Sapphire : a multi-agent system for remote healthcare monitoring through computerized clinical guidelines. In *Agent technology and e-health*, pages 25–44. Springer, 2007.
- [104] Gokce B. Laleci, Asuman Dogac, Mehmet Olduz, Ibrahim Tasyurt, Mustafa Yuksel, and Alper Okcan. Sapphire : A multi-agent system for remote

- healthcare monitoring through computerized clinical guidelines. In Roberta Annicchiarico, Ulises Cortés, and Cristina Urdiales, editors, *Agent Technology and e-Health*, pages 25–44, Basel, 2008. Birkhäuser Basel.
- [105] Gokce Banu Laleci Erturkmen and TMustafa Yüksel. SAPHIRE | SRDC. <http://www.srdc.com.tr/rndprojects/saphire/>. [ Visité le : 2019-11-15].
- [106] Danny B Lange and Mitsuru Oshima. Mobile agents with java : the aglet api. *World Wide Web*, 1(3) :111–121, 1998.
- [107] legifrance. Article 1111-6-1 du code de la santé publique. [https://www.legifrance.gouv.fr/codes/article\\_lc/LEGIARTI000006685775/](https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000006685775/). [ Visité le : 2021-12-16].
- [108] legifrance. Article 1245-7 du code de l'action sociale et des familles. [https://www.legifrance.gouv.fr/codes/article\\_lc/LEGIARTI000018782122/](https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000018782122/). [ Visité le : 2021-12-16].
- [109] Yanjun Li, Gan Chen, Yueyun Shen, Yihua Zhu, and Zhen Cheng. Accelerometer-based fall detection sensor system for the elderly. In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, volume 3, pages 1216–1220. IEEE, 2012.
- [110] Hanghan Liang and Wipawee Usaha. Fall detection using lifting wavelet transform and support vector machine. In *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 877–883. IEEE, 2017.
- [111] Shu-Hsien Liao. Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert systems with applications*, 28(1) :93–103, 2005.
- [112] Dominique Libault. Grand âge, le temps d’agir. *La documentation française*, 2019. [ Visité le : 2021-12-16].

- [113] Khawla Ltifi, Réjane Dalce, Imen Megdiche, and Thierry Val. Une canne connectée lorawan, wifi et ble pour le suivi des personnes âgées. In *JET-SAN 2021*, 2021.
- [114] Yuqian Lu and Xun Xu. Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services. *Robotics and Computer-Integrated Manufacturing*, 57 :92–102, 2019.
- [115] Marion Luyat and Tony Regia-Corte. Les affordances : de james jerome gibson aux formalisations récentes du concept. *L'Année psychologique*, 109(2) :297–332, 2009.
- [116] Zakaria Maamar, Noura Faci, Khoulood Boukadi, Emir Ugljanin, Mohamed Sellami, Thar Baker, and Rafael Angarita. How to agentify the internet-of-things? In *2018 12th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–6. IEEE, 2018.
- [117] Zakaria Maamar, Noura Faci, Slim Kallel, Mohamed Sellami, and Emir Ugljanin. Software agents meet internet of things. *Internet Technology Letters*, 1(3) :e17, 2018.
- [118] Noah P Mark. Cultural transmission, disproportionate prior exposure, and the evolution of cooperation. *American Sociological Review*, pages 323–344, 2002.
- [119] France Meslé. Progrès récents de l'espérance de vie en france. *Population*, 61(4) :437–462, 2006.
- [120] Helen Micheaux. La gestion des déchets et la responsabilité élargie du producteur, un modèle français qui fait référence. *Le journal de l'école de Paris du management*, 6(6) :23–30, 2020.
- [121] Nader Mohamed, Jameela Al-Jaroodi, and Imad Jawhar. Middleware for robotics : A survey. In *RAM*, pages 736–742, Chengdu, China, 2008. IEEE.



- [122] Hans Moravec. *Mind children : The future of robot and human intelligence*. Harvard University Press, 1988.
- [123] Taketoshi Mori, Akinori Fujii, Masamichi Shimosaka, Hiroshi Noguchi, and Tomomasa Sato. Typical behavior patterns extraction and anomaly detection algorithm based on accumulated home sensor data. In *Future generation communication and networking (FGCN 2007)*, volume 2, pages 12–18. IEEE, 2007.
- [124] Saad Z Nagi. An epidemiology of disability among adults in the united states. *The Milbank Memorial Fund Quarterly. Health and Society*, pages 439–467, 1976.
- [125] Assemblée Nationale. Proposition de loi visant à agir pour préserver l'autonomie et garantir les choix de vie de nos aînés. [https://www.assemblee-nationale.fr/dyn/15/textes/115b4257\\_proposition-loi](https://www.assemblee-nationale.fr/dyn/15/textes/115b4257_proposition-loi), 2021. [ Visité le : 2021-12-16].
- [126] Peter Naur and Randell B. Software engineering-report on a conference sponsored by the nato science committee garimisch, germany. In *NATO Software Engineering Conference*, volume 8, 1968.
- [127] Norbert Noury, Ronald Nocua, Claudine Gehin, Bertrand Massot, and Eric McAdams. Le «fall perceptron»-détection de l'activation du système nerveux autonome lors de la chute. In *Instrumentation et Interdisciplinarité*, pages 189–194. EDP Sciences, 2021.
- [128] Institute of Electrical and Electronics Engineers. Ieee 802.15 working group for wireless specialty networks (wsn). <http://www.ieee802.org/15/>. [ Visité le : 2021-08-19].
- [129] Institute of Electrical and Electronics Engineers. Ieee 802.15 wpan™, task group 6 (tg6), body area networks. <http://www.ieee802.org/15/pub/TG6.html>. [ Visité le : 2021-08-19].

- [130] Andrea Omicini, Alessandro Ricci, and Mirko Viroli. Artifacts in the a&a meta-model for multi-agent systems. *Autonomous agents and multi-agent systems*, 17(3) :432–456, 2008.
- [131] Andrea Omicini, Alessandro Ricci, Mirko Viroli, Cristiano Castelfranchi, and Luca Tummolini. Coordination artifacts : Environment-based coordination for intelligent agents. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 286–293, 2004.
- [132] Gustavo Ospina and Lotfi Guedria. TRICARE - CETIC - your connection to ICT research. <https://www.cetic.be/TRICARE-2809>. [Visité le : 2019-11-15].
- [133] Jun Ota. Multi-agent robot systems as distributed autonomous systems. *Advanced engineering informatics*, 20(1) :59–70, 2006.
- [134] Alison R Panisson, Artur Freitas, Daniela Schmidt, Lucas Hilgert, Felipe Meneguzzi, Renata Vieira, and Rafael H Bordini. Arguing about task reallocation using ontological information in multi-agent systems. In *12th International Workshop on Argumentation in Multiagent Systems*, volume 108, 2015.
- [135] Carlos Eduardo Pantoja, Heder Dorneles Soares, José Viterbo, and Amal El Fallah-Seghrouchni. An architecture for the development of ambient intelligence systems managed by embedded agents. In *SEKE*, pages 215–214, 2018.
- [136] Renaud Petit and Véronique Zardet. Attractivité, fidélisation et implication du personnel des ehpad : une problématique sectorielle et de management. @ *GRH*, 1(1) :31–54, 2017.
- [137] Stéphanie Pinard, Carolina Bottari, Catherine Laliberté, Helene Pigot, Marisnel Olivares, Mélanie Couture, Sylvain Giroux, and Nathalie Bier. Design and usability evaluation of cook, an assistive technology for meal

- preparation for persons with severe tbi. *Disability and Rehabilitation : Assistive Technology*, 16(7) :687–701, 2021.
- [138] Michele Piunti and Alessandro Ricci. Cognitive use of artifacts : Exploiting relevant information residing in mas environments. In *International Workshop on Knowledge Representation for Agents and Multi-Agent Systems*, pages 114–129. Springer, 2008.
- [139] Martha E Pollack. Intelligent technology for an aging population : The use of ai to assist elders with cognitive impairment. *AI magazine*, 26(2) :9–9, 2005.
- [140] Christophe Ponsard, Mike Martin, Sarah Walsh, Susan Baines, Sébastien Rousseaux, Giovanni Rinaldi, and Fulvio Tamburriello. OLDES : Designing a low-cost, easy-to-use e-care system together with the stakeholders. In *International Conference on Computers for Handicapped Persons*, pages 1285–1292. Springer, 2008.
- [141] David Pucheu. Effacer l’interface : Une trajectoire du design de l’interaction hommemachine. *Interfaces numériques*, 5(2) :276–276, 2018.
- [142] Carlos Ramos, Goreti Marreiros, Ricardo Santos, and Carlos Filipe Freitas. Smart offices and intelligent decision rooms. In Hideyuki Nakashima, Hamid Aghajan, and Juan Carlos Augusto, editors, *Handbook of Ambient Intelligence and Smart Environments*, pages 851–880. Springer US, Boston, MA, 2010.
- [143] Anand S Rao and Michael P Georgeff. Modeling rational agents within a bdi-architecture. *KR*, 91 :473–484, 1991.
- [144] Parisa Rashidi and Diane J Cook. Keeping the resident in the loop : Adapting the smart home to the user. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 39(5) :949–959, 2009.

- [145] 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, 2012.
- [146] Wei Ren, Randal W Beard, and Ella M Atkins. A survey of consensus problems in multi-agent coordination. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 1859–1864. IEEE, 2005.
- [147] Alessandro Ricci, Michele Piunti, Mirko Viroli, and Andrea Omicini. Environment programming in cartago. In *Multi-agent programming*, pages 259–288. Springer, 2009.
- [148] Alessandro Ricci, Mirko Viroli, and Andrea Omicini. Carta go : A framework for prototyping artifact-based environments in mas. In Danny Weyns, H. Van Dyke Parunak, and FabienEditors Michel, editors, *Environments for Multi-Agent Systems III*, volume 4389, page 67–86. Springer Berlin Heidelberg, 2007.
- [149] Fontaine Richard, Courdier Rémy, and Payet Denis. A step towards more eco-responsible computing. In *International Conference on Mobile Web and Intelligent Information Systems*, pages 175–184. Springer, 2021.
- [150] Artur Rocha, Angelo Martins, José Celso Freire, Maged N. Kamel Boulos, Manuel Escriche Vicente, Robert Feld, Pepijn van de Ven, John Nelson, Alan Bourke, and Gearóid ÓLaighin. Innovations in health care services : The CAALYX system. *International journal of medical informatics*, 82(11) :e307–e320, 2013.
- [151] Mário Lucio Roloff, Marcelo Ricardo Stemmer, Jomi Fred Hübner, Robert Schmitt, Tilo Pfeifer, and Guido Hüttemann. A multi-agent system for the production control of printed circuit boards using jacamo and prometheus aeolus. In *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, pages 236–241. IEEE, 2014.

- [152] M Roser and H Ritchie. Burden of disease. [https://ourworldindata.org/burden-of-disease.](https://ourworldindata.org/burden-of-disease), 2020. [ Visité le : 2021-04-22].
- [153] Carol W Runyan, David Perkis, Stephen W Marshall, Renee M Johnson, Tamera Coyne-Beasley, Anna E Waller, Carla Black, and Lorena Bacca-  
glini. Unintentional injuries in the home in the united states : Part ii :  
Morbidity. *American journal of preventive medicine*, 28(1) :80–87, 2005.
- [154] Stuart J Russell. Rationality and intelligence. *Artificial Intelligence*,  
1(94) :57–77, 1997.
- [155] Jlias Sachpazidis. @ home : A modular telemedicine system. In *Mo-  
bile Computing in Medicine, Second Conference on Mobile Computing in  
Medicine, Workshop of the Project Group MoCoMed, GMDS-Fachbereich  
Medizinische Informatik & GI-Fachausschuss 4.7*, pages 87–95. GI, 2002.
- [156] Pierangela Samarati. Protecting respondents identities in microdata  
release. *IEEE transactions on Knowledge and Data Engineering*,  
13(6) :1010–1027, 2001.
- [157] Gérard Sauce, Jean-Laurent Burlet, and Olivier Graziani. Caractérisation  
de l’intelligence des bâtiments. *Academic Journal of Civil Engineering*,  
36(1) :72–75, 2018.
- [158] Lionel Seinturier, Philippe Merle, Romain Rouvoy, Daniel Romero, Va-  
lerio Schiavoni, and Jean-Bernard Stefani. A Component-Based Middle-  
ware Platform for Reconfigurable Service-Oriented Architectures. *Soft-  
ware : Practice and Experience*, 42(5) :559–583, May 2012.
- [159] Weisong Shi and Schahram Dustdar. The promise of edge computing.  
*Computer*, 49(5) :78–81, 2016.
- [160] Yoav Shoham. Agent-oriented programming. *Artificial intelligence*,  
60(1) :51–92, 1993.

- [161] Nikolaos I Spanoudakis and Pavlos Moraitis. Using agent technology for ambient assisted living. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 518–525. Springer, 2013.
- [162] Christian Stary. Digital twin generation : Re-conceptualizing agent systems for behavior-centered cyber-physical system development. *Sensors*, 21(4) :1096, 2021.
- [163] Thomas A Stoffregen. Affordances as properties of the animal-environment system. *Ecological psychology*, 15(2) :115–134, 2003.
- [164] Thomas A Stoffregen, Benoît G Bardy, and Bruno Mantel. Affordances in the design of enactive systems. *Virtual Reality*, 10(1) :4–10, 2006.
- [165] François Suro, Jacques Ferber, Tiberiu Stratulat, and Fabien Michel. A hierarchical representation of behaviour supporting open ended development and progressive learning for artificial agents. *Autonomous Robots*, 45(2) :245–264, 2021.
- [166] Latanya Sweeney. Weaving technology and policy together to maintain confidentiality. *The Journal of Law, Medicine & Ethics*, 25(2-3) :98–110, 1997.
- [167] Judith A Symonds and David Parry. Using rfid to track and trace high value products : the case of city healthcare. *Journal of Cases on Information Technology (JCIT)*, 10(1) :1–13, 2008.
- [168] Pierrick Thebault. *La conception à l'ère de l'Internet des Objets : modèles et principes pour le design de produits aux fonctions augmentées par des applications*. PhD thesis, Ecole nationale supérieure d'arts et métiers-ENSAM, 2013.
- [169] Alessandro Tognetti, Federico Lorussi, Raphael Bartalesi, Silvana Quaglioni, Mario Tesconi, Giuseppe Zupone, and Danilo De Rossi. Wearable kinesthetic system for capturing and classifying upper limb gesture in

- post-stroke rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 2(1) :1–16, 2005.
- [170] Alessandro Tognetti, Federico Lorussi, M Tesconi, and D De Rossi. Strain sensing fabric characterization. In *SENSORS, 2004 IEEE*, pages 527–530. IEEE, 2004.
- [171] Antonio Tricoli, Noushin Nasiri, and Sayan De. Wearable and miniaturized sensor technologies for personalized and preventive medicine. *Advanced Functional Materials*, 27(15) :1605271, 2017.
- [172] Makoto Tsunoda, Masaaki Hirayama, Takao Tsuda, and Kinji Ohno. Noninvasive monitoring of plasma l-dopa concentrations using sweat samples in parkinson’s disease. *Clinica chimica acta*, 442 :52–55, 2015.
- [173] Michael T Turvey. Affordances and prospective control : An outline of the ontology. *Ecological psychology*, 4(3) :173–187, 1992.
- [174] Mathieu Vallee. *Un intergiciel multi-agent pour la composition flexible d’applications en intelligence ambiante*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2009.
- [175] Athanasios Vasilakos and Witold Pedrycz. *Ambient intelligence, wireless networking, and ubiquitous computing*. Artech House, Inc., 2006.
- [176] Santosh K Verma, Joanna L Willetts, Helen L Corns, Helen R Marucci-Wellman, David A Lombardi, and Theodore K Courtney. Falls and fall-related injuries among community-dwelling adults in the united states. *PLoS one*, 11(3) :e0150939, 2016.
- [177] Gilles Virone and Norbert Noury. Télé-surveillance automatique de l’activité dans un habitat intelligent pour la santé. *JFIM2002, Congrès Francophone d’Informatique Médicale*, 2002.
- [178] Jean-Baptiste Waldner. *Nanocomputers and swarm intelligence*. John Wiley & Sons, 2013.

- [179] Samuel Fosso Wamba, Abhijith Anand, and Lemuria Carter. A literature review of rfid-enabled healthcare applications and issues. *International Journal of Information Management*, 33(5) :875–891, 2013.
- [180] Jie Wan, Caroline Byrne, Gregory MP O’Hare, and Michael J O’Grady. Outcare : Supporting dementia patients in outdoor scenarios. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 365–374. Springer, 2010.
- [181] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X Gao, and Dazhong Wu. Deep learning for smart manufacturing : Methods and applications. *Journal of manufacturing systems*, 48 :144–156, 2018.
- [182] Miao-Miao Wang, Jian-Nong Cao, Jing Li, and Sajal K Dasi. Middleware for wireless sensor networks : A survey. *Journal of computer science and technology*, 23(3) :305–326, 2008.
- [183] Zhonghai Wang and Xiaoya Xu. Smart home m2m networks architecture. In *2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*, pages 294–299. IEEE, 2013.
- [184] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1) :91–102, 1992.
- [185] Hung-yu Wei and Richard D Gitlin. Wwan/wlan two-hop-relay architecture for capacity enhancement. In *2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No. 04TH8733)*, volume 1, pages 225–230. IEEE, 2004.
- [186] Mark Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7) :75–84, 1993.
- [187] Michael Wooldridge. Agent-based software engineering. *IEE Proceedings-software*, 144(1) :26–37, 1997.



- [188] Michael Wooldridge. Intelligent agents : The key concepts. In *ECCAI Advanced Course on Artificial Intelligence*, pages 3–43. Springer, 2001.
- [189] Michael Wooldridge and Nicholas R Jennings. Intelligent agents : Theory and practice. *The knowledge engineering review*, 10(2) :115–152, 1995.
- [190] Jiawei Wu, Min Xia, and Shaolong Shu. Location tracking of a single inhabitant in smart home : A discrete event system approach. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 615–620. IEEE, 2016.
- [191] Weikai Xie, Yuanchun Shi, Guanyou Xu, and Dong Xie. Smart classroom-an intelligent environment for tele-education. In *Pacific-Rim Conference on Multimedia*, pages 662–668. Springer, 2001.
- [192] Jinsung Yoon, Lydia N Drumright, and Mihaela Van Der Schaar. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE journal of biomedical and health informatics*, 24(8) :2378–2388, 2020.
- [193] Yao-lin Zhu, Rong Li, Xue-bin Liu, and Xu Jian. Wireless communication technology in family health monitoring system. In *2011 International Conference on Business Management and Electronic Information*, volume 3, pages 64–67. IEEE, 2011.

# ANNEXE

id	name	type	vendor
1	LSM6DSL Accelerometer	1	STM
2	AK09918 Magnetometer	2	Asahi Kasei Microd.
3	LSM6DSL Gyroscope	4	STM
4	TMD3725 Light	5	AMS, Inc.
5	LPS25H Barometer	6	STM
6	TMD3725 Proximity	8	AMS, Inc.
7	AK09918 Magnetometer UnCalibrated	14	Asahi Kasei Microd.
8	LSM6DSL Gyroscope Uncalibrated	16	STM
9	Samsung Significant Motion	17	Samsung Inc.
10	Samsung Step Detector	18	Samsung Inc.
11	Samsung Step Counter	19	Samsung Inc.
12	Samsung Tilt Detector	22	Samsung Inc.
13	Samsung Pick Up Gesture	25	Samsung Inc.
14	Interrupt Gyroscope	65579	Samsung Inc.
15	SContext	65586	Samsung

id	name	type	vendor
16	TMD3725 Light CCT	65587	AMS, Inc.
24	LSM6DSM Accelerometer	1	STMicroelectronics
25	AK09915 Magnetometer	2	AKM
43	Orientation	3	Google
44	Tilt	22	Google
45	Device Pickup	25	Google
46	Device Orientation	27	Google
47	Binned Brightness	33171034	Google
48	Double Twist	33171033	Google
58	LSM6DSL Gyroscope Uncalibrated	16	STMicroelectronics
59	TMD4906 PROX	8	AMS TAOS
60	TMD4906 ALS	5	AMS TAOS
61	LPS22HB Barometer	6	STMicroelectronics
62	Gravity	9	QTI
63	Linear Acceleration	10	QTI
64	Rotation Vector	11	QTI
65	Step Detector	18	Samsung Electronics
66	Step Counter	19	Samsung Electronics
67	Significant Motion Detector	17	Samsung Electronics
68	Game Rotation Vector	15	QTI

id	name	type	vendor
69	Orientation	3	QTI
70	Tilt Detector	22	Samsung Electronics
71	Pick Up Gesture	25	Samsung Electronics
72	Gyro Interrupt	65579	Samsung
73	Iris Prox	65582	Samsung
74	TMD4906 Wide ALS	65578	AMS TAOS
75	SensorHub type	65586	Samsung
76	TMD4906 ALS-CCT	65587	AMS TAOS

TABLEAU .2 – Extrait de la norme de facto utilisé sur Android (source : <http://dataqualitylabs.com/downloadDataFor/Sensor%20Info>)