



HAL
open science

Fusion de données crédibilistes dans le cadre de l'intelligence ambiante

Vincent Ricquebourg

► **To cite this version:**

Vincent Ricquebourg. Fusion de données crédibilistes dans le cadre de l'intelligence ambiante. Automatique. Université de Valenciennes et du Hainaut-Cambrésis, 2008. Français. NNT: 2008VALE0027. tel-03011784

HAL Id: tel-03011784

<https://uphf.hal.science/tel-03011784>

Submitted on 18 Nov 2020

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.

UNIVERSITE DE VALENCIENNES ET DU HAINAUT CAMBRESIS

THESE

Présentée par

VINCENT RICQUEBOURG

Pour l'obtention du grade de

**DOCTEUR DE L'UNIVERSITE DE VALENCIENNES
ET DU HAINAUT CAMBRESIS**

Discipline : Automatique

Spécialité : Robotique

**FUSION DE DONNEES CREDIBILISTE DANS LE
CADRE DE L'INTELLIGENCE AMBIANTE**

Soutenue le 19 Septembre 2008 devant le jury composé de :

- Pr. Yacine AMIRAT, IUT de Créteil-Vitry- SCTIC/LiSSi, Rapporteur
- Pr. Philippe BONNIFAIT, Université de Technologie de Compiègne - Heudiasyc, Rapporteur
- Pr. Alain DERYCKE, Université des Sciences et Technologies de Lille - TRIGONE/LIFL
- Pr. Gérard POISSON, IUT de Bourges, PRISME
- Dr. HDR François DELMOTTE, Université de Valenciennes et du Hainaut Cambresis - MCS/LAMIH
- Mr. David MENGA, EDF Recherche & Développement Clamart
- Mr. Gérard BRISSON, EDF Recherche & Développement Clamart
- Pr. Anne Marie JOLLY-DESODT, Ecole Polytechnique d'Orléans, Directrice de thèse
- Dr. Laurent DELAHOUCHE, IUT d'Amiens - LTI, Directeur de thèse
- Dr. Bruno MARHIC, IUT d'Amiens - LTI, Directeur de thèse



Cette thèse a fait l'objet d'un contrat CIFRE entre le laboratoire d'innovation d'EDF R&D de Clamart (92) et le laboratoire GEMTEX de Valenciennes (59) représenté par l'IUT d'Amiens (80).

Remerciements

J'adresse mes sincères remerciements aux rapporteurs de cette thèse, Philippe Bonnifait et Yacine Amirat, pour avoir accepté de rapporter sur ce travail. Je remercie également Gérard Poisson et Alain Derick d'avoir accepté d'être dans mon jury en tant qu'examinateur.

Je tiens à remercier l'équipe innovation d'EDF Recherche et Développement de Clamart pour m'avoir accueilli et en particulier David Menga et Gérard Brisson sans qui cette thèse n'aurait pas eu lieu. Merci pour leurs encadrements et leurs conseils. Je remercie l'équipe dirigeante, Méhana Chami, Philippe Chabault et Michelle Jacquemelle pour m'avoir fait confiance durant ces trois années. Merci à toutes les personnes qui m'ont accompagné pendant cette période, Brigitte, Stéphane, Jean Michel, André, Olaf, Denis, Philippe, Yves, Damien, Si-Hoang, Antoine et Sylvain.

J'exprime toute ma gratitude au Laboratoire GEMTEX et en particulier à Anne Marie Jolly Desodt pour son encadrement et sa précieuse aide dans les démarches administratives. Je tiens à remercier François Delmotte pour ses conseils avisés.

Je remercie également le groupe RTeAM et actuellement le laboratoire LTI, et en particulier Bruno Marhic et Laurent Delahoche pour m'avoir accueilli au sein de leur équipe et encadré durant cette thèse. Merci pour la confiance et la liberté qu'ils m'ont accordées.

Un grand merci à Eric Brassart, ancien Chef du Département Informatique de l'IUT, pour m'avoir permis d'enseigner au sein du département. Merci aux collègues, actuels ou non, Amaury, Cyril, Dr Kovacs, Brigitte, Raynald, Sandrine, Claude, Sabine, Claudine, Keith, Marie-Christine, Sonia, Fabrice, Cyril pour leur bonne humeur.

Merci également à DD, Mélanie, Kristof, Arnaud. Vos remarques et soutiens m'ont aidé à avancer et à progresser.

Muchas Gracias à Dominique pour la période passée dans son bureau et pour ses précieux cours d'espagnol...

Merci à Vanessa pour sa compréhension, sa patience et son soutien durant cette période de thèse, peu favorable à la vie de famille. Je ne peux oublier mon petit Quentin sans qui je n'aurais pu apprécier autant de magnifiques levés de soleil...

Enfin, je remercie mes parents pour leur aide et soutien et pour m'avoir permis d'arriver jusqu'ici.

Table des matières

Introduction générale	1
Cadre et objectifs	1
Organisation de ce mémoire	3
Chapitre 1 : L'habitat communicant au service de l'habitant	7
1 L'habitat communicant : Une vision futuriste de l'habitat ?	8
1.1 Introduction	8
1.2 La notion d'habitat communicant	8
1.2.1 Répondre au problème du maintien à domicile avec l'habitat communicant ?	9
1.2.2 Un habitat communicant pour la surveillance et/ou le confort	13
1.3 Comment rendre un habitat communicant ?	13
1.4 Une approche : L'intelligence ambiante	16
2 Le contexte - savoir ce qui se passe pour savoir quoi faire	18
2.1 Présentation	18
2.2 La perception du contexte : les difficultés rencontrées	21
2.3 La représentation du contexte	23
2.3.1 L'approche paires/triplets	23
2.3.2 L'approche orientée modèles	24
2.3.3 L'approche basée sur la logique	24
2.3.4 L'approche orientée ontologie	25
2.3.5 Comparatifs des approches	26
2.4 La prise de décision : un problème d'inférence contextuelle	27
2.4.1 Présentation	27
3 La fusion de données, l'incertain et le contexte	28
3.1 Présentation	28
3.2 La conversion symbolique : Comment la réaliser à partir de données numériques ?	29
3.3 La fusion de données	30
3.3.1 Présentation	30
3.3.2 Les méthodes probabilistes	32
3.3.3 La logique floue et la théorie des possibilités	33
3.3.4 Les méthodes non paramétriques	34
3.3.5 Les réseaux neuronaux	34
3.3.6 La théorie de Dempster-Shafer (ou théorie de l'évidence ou théorie des fonctions de croyance)	34
4 La défaillance capteur	36
4.1 Introduction	36
4.2 La tolérance aux pannes	36
4.3 Détection de défaillance capteurs : Méthodes utilisées	37
5 Conclusion	38

Chapitre 2 : La prise en compte du contexte dans l'habitat communicant	39
1 Introduction	40
2 Le modèle PCIA (Perception-Contexte-Inférence-Action)	40
3 Architecture logicielle de gestion du contexte dans l'habitat communicant	41
3.1 Introduction	41
3.2 Approche par les composants	42
3.3 Approche à service	42
3.4 OSGi par rapport à JADE	44
3.5 OSGi et le modèle PCIA	44
3.5.1 La couche Perception	45
3.5.2 La couche Contexte	48
3.5.3 La couche Inférence	50
3.5.4 La couche Action	54
3.5.5 Intégration des quatre couches au sein d'une architecture OSGi	54
3.6 Interfaçage de l'ontologie avec l'architecture à service	56
3.6.1 Description du service de gestion d'ontologie	56
3.6.2 Intégration de la volonté de l'utilisateur	60
4 Validation de l'architecture de gestion de contexte dans l'habitat communicant	61
4.1 Mise en œuvre de l'architecture de gestion de contexte	61
4.2 Limites de l'approche	65
4.2.1 Le temps de réponse	65
4.2.2 Inconsistances et ajout de règles	67
5 Conclusion	67
Chapitre 3 : Gestion de l'incertain et fusion de données dans l'habitat communicant : Application à la détection de défaillance	69
1 La fusion de données et l'incertain	70
1.1 Introduction	70
1.2 Approche crédibiliste	71
1.2.1 Fonction de crédibilité	73
1.2.2 Fonction de plausibilité	73
1.2.3 Lien avec les probabilités	74
1.2.4 Assignation des fonctions de masse de croyance	75
1.3 Fusion de données crédibiliste	75
1.3.1 Combinaison de deux sources d'information	76
1.3.2 Propriétés de la loi de combinaison de Dempster	77
1.3.3 Affaiblissement	78
1.3.4 Décision	78
1.4 Le Modèle des Croyances Transférables	80
1.5 Limite de la fusion de données	82
1.5.1 Le conflit	82
1.5.2 Résoudre le problème du conflit	83
2 La fusion de données crédibiliste et l'habitat communicant	86
2.1 Intégration de la fusion de données dans notre architecture de gestion de contexte	86
2.2 Les fonctions de Croyance et l'informatique embarquée	87

3	Détection et identification d'un capteur défaillant dans un réseau de capteurs	88
3.1	Introduction	88
3.2	Détection d'un capteur défaillant	89
3.2.1	Présentation	89
3.2.2	Méthode	90
3.3	Identification du capteur défaillant	94
3.4	Estimation de la fiabilité d'un capteur	95
3.5	Conclusion	96
4	Détection des dérives de comportement capteur	97
4.1	Introduction	97
4.2	Modélisation d'un capteur par un processus stochastique	97
4.3	Principe de la détection de dérives	99
4.3.1	Méthode 1 : La prédiction Markovienne	99
4.3.2	Méthode 2 : La prédiction Markovienne au sens crédibiliste	102
4.3.3	Détection d'une dérive par analyse du conflit entre observation et prédiction	104
5	Conclusion	107

Chapitre 4 : Application de la fusion de données crédibiliste à la surveillance d'activités humaines dans l'habitat communicant **109**

1	Introduction	110
2	La fusion de données : Application à la surveillance d'activités humaines – Etape E1	111
2.1	Introduction	111
2.2	Caractérisation d'une posture	112
2.2.1	Premier capteur retenu : Le capteur de pression mécanique	113
2.2.2	Second capteur retenu : le capteur à vision omnidirectionnelle	114
2.2.3	Troisième capteur retenu : le capteur accélérométrique	118
2.2.4	Autres capteurs	120
2.3	Résultats expérimentaux	122
3	Détection et identification d'un capteur défaillant – Etape E2	130
3.1	Combinaison deux à deux des capteurs	131
3.2	Analyse des conflits	132
3.3	Identification du capteur défaillant	135
3.4	Estimation de la fiabilité des capteurs	136
4	Détection de dérives comportementales – Etape E3	143
4.1	Modélisation du comportement humain par Chaîne de Markov et prédiction markovienne	144
4.2	Application de la prédiction Markovienne à la détection de posture	145
4.3	Analyse du conflit entre observation et prédiction et identification d'une dérive comportementale	150
4.4	Amélioration de la prédiction : une approche crédibiliste	152
4.5	Impact d'une défaillance capteur sur la détection de dérives comportementales	155
5	Conclusion	156

Conclusion générale **159**

Rappel de la problématique **159**

Améliorations et perspectives **162**

Bibliographie	167
Bibliographie de l'auteur	175
Workshops	175
Congrès internationaux avec comité de lecture	175
Congrès nationaux avec comité de lecture	175

Table des figures

Figure 1 : Population des 60 ans et plus (tous sexes confondus). Évolution depuis 1980 et projections pour 2050 en France.	10
Figure 2 : Gator Tech Smart Home.	11
Figure 3 : L'identificateur RFID et la caméra d'avertissement de présence avec visionnage du visiteur sur l'écran du salon.	12
Figure 4 : Une dalle du smart floor équipé de son capteur de pression (1ère version filaire et seconde version sans fil) avec visionnage du visiteur sur l'écran du salon et une prise équipée du « Smart Plug ».	12
Figure 5 : La smart Board d'Intel, le miroir communicant de Phillips HomeLab, le réfrigérateur communicant de LG, l'aspirateur robotisé Roomba, le siège robotisé de Jelte van Geest (http://www.jeltevangeest.nl/), autant d'objets communs qui deviennent communicants.	15
Figure 6 : Principe de la maison communicante.	15
Figure 7 : L'évolution de l'informatique selon Mark Weiser.	16
Figure 8 : Fonctionnement d'un environnement pervasive.	17
Figure 9 : Exemple d'un système implémenté dans un environnement pervasive.	18
Figure 10 : Les différents niveaux d'abstraction nécessaires à l'élaboration d'un système sensible au contexte [COU05].	19
Figure 11 : Architecture de perception de contexte.	20
Figure 12 : Le iBracelet d'INTEL et une brosse à dent ainsi que le tube de dentifrice portant un tag RFID.	29
Figure 13 : Les différents niveaux de fusion [AUD04].	31
Figure 14 : Les couches du modèle PCIA et leurs interactions.	41
Figure 15 : Exemple de déploiement de Bundles OSGI (source Didier Donsez).	43
Figure 16 : Le modèle Publish/Subscribe pour des capteurs.	46
Figure 17 : Ensemble de couches de connexion au bus événementiel.	47
Figure 18 : Exemple de relations de composition et d'héritage.	49
Figure 19 : Exemple de relations sémantiques entre les classes.	49
Figure 20 : Règles à chaînage arrière décrivant ce qu'est une maison occupée et une maison inoccupée.	51
Figure 21 : Exemple de règles à chaînage avant.	53
Figure 22 : Le modèle Publish/Subscribe pour des actionneurs.	54
Figure 23 : Architecture de gestion de contexte orientée service.	55
Figure 24 : Traitement des données par le service ontologie.	57
Figure 25 : Interface du service ontologie pour la configuration.	58
Figure 26 : Interface du service ontologie en cours de fonctionnement.	58
Figure 27 : Interfaçage entre le bus événementiel et le service ontologie.	59
Figure 28 : Architecture logicielle détaillée.	62
Figure 29 : Mise en relation des capteurs/actionneurs dans l'architecture de gestion de contexte.	63
Figure 30 : Remontée d'un événement capteur.	63
Figure 31 : Exemple de relations dans l'ontologie.	64
Figure 32 : Chaîne complète de la gestion du contexte.	70
Figure 33 : Le cerveau humain et son système de perception comme un exemple de processus de fusion.	71
Figure 34 : La théorie des probabilités vs la théorie des fonctions de Croyance [ALM05].	72
Figure 35 : Relations entre crédibilité, plausibilité et incertitude.	74
Figure 36 : Problème de chevauchement d'intervalles complexifiant la décision.	79
Figure 37 : Les deux niveaux du MCT.	81

Table des figures

Figure 38 : Principe de la transformation pignistique.	81
Figure 39 : La fusion statique de N sources.	86
Figure 40 : Architecture de détection de défaillance capteur.	89
Figure 41 : Chaîne complète de la gestion du contexte avec prise en charge de la défaillance capteur.	89
Figure 42 : Exemples de signaux de conflits et résultat du filtrage passe-bas.	91
Figure 43 : Fonctions de masses m_{E1ij} et m_{E2ij} des deux experts en charge de l'analyse du signal de conflit.	92
Figure 44 : Architecture de détection et d'identification de capteur défaillant dans un réseau de trois capteurs.	94
Figure 45 : Chaîne de Markov pour un capteur à trois états.	98
Figure 46 : Principe global de la détection de dérives comportementales.	99
Figure 46 : La Chaîne de Markov et sa matrice de transition associée.	100
Figure 47 : Processus de conversion de masses dans le cadre d'une prédiction Markovienne.	100
Figure 48 : Chaîne complète d'analyse comportementale d'un individu avec prise en charge de la défaillance capteur.	111
Figure 49 : La chaise équipée du capteur de pression mécanique et un le signal issu du capteur lorsqu'un individu s'assoit.	113
Figure 50 : Fonction de masse du capteur de pression.	114
Figure 51 : Exemple d'image d'énergie et d'historique du mouvement	116
Figure 52 : Le capteur à vision omnidirectionnelle, l'image originale d'une personne debout, son équivalent traitée par MHI et le traitement d'une personne allongée.	116
Figure 53 : Estimation de la distance entre la direction moyenne d'un individu et du centre de l'image omnidirectionnelle.	117
Figure 54 : Fonction de masses du capteur à vision omnidirectionnelle.	118
Figure 55 : Données accélérométriques d'un individu marchant.	118
Figure 56 : Données accélérométriques d'un individu qui s'assoit.	119
Figure 57 : Données accélérométriques d'un individu qui s'allonge.	119
Figure 58 : Fonction de masse de l'accéléromètre placé à la ceinture de l'individu.	120
Figure 59 : Détermination de la posture à partir de la fusion de 3 capteurs.	120
Figure 60 : Principe de fonctionnement d'une dalle du paillason communicant.	121
Figure 61 : Le paillason communicant et sa carte de traitement à base de microcontrôleur.	121
Figure 62 : Principe de détection d'entrée ou sortie dans la pièce à partir du paillason communicant.	121
Figure 63 : Le capteur à base de thermopiles et son microcontrôleur chargé du traitement des données.	122
Figure 64 : Principe de détection d'entrée ou sortie dans la pièce à partir du capteur à thermopiles.	122
Figure 65 : Architecture matérielle d'acquisition des 3 capteurs.	122
Figure 66 : Carte prototype d'expérimentation pour l'acquisition des données accélérométriques et de pression.	123
Figure 67 : Architecture logicielle de fusion des 3 capteurs.	123
Figure 68 : Signaux expérimentaux obtenus à partir des 3 capteurs pour le premier scénario.	124
Figure 69 : Conversion numérique vers symbolique des mesures de pression.	125
Figure 70 : Prise de décision pignistique sur le jeu de masse m_1 .	125
Figure 71 : Conversion numérique vers symbolique des mesures issues de la vision omnidirectionnelle.	125
Figure 72 : Conversion numérique vers symbolique des mesures accélérométriques.	126
Figure 73 : Fusion des trois jeux de masses m_1 , m_2 et m_3 afin de quantifier l'incertitude sur la posture estimée (première portion du graphique) et progression du conflit lors des fusions (en rouge, seconde portion du graphique).	127
Figure 74 : Décision par calcul du maximum de probabilité pignistique et comparaison entre les postures estimées par fusion et le scénario réellement mis en œuvre.	127
Figure 75 : Signaux expérimentaux avec ajout d'un dysfonctionnement sur le troisième capteur.	128

Table des figures

Figure 76 : Fusion des trois jeux de masses m_1 , m_2 et m_3 , la transformation pignistique du jeu de masse fusionné et la prise de décision correspondante. _____	129
Figure 77 : Décision par calcul du maximum de probabilité pignistique. _____	129
Figure 78 : Détection et identification d'un capteur défaillant parmi les trois capteurs. _____	130
Figure 79 : Ensemble de conflits résultants de la fusion deux à deux entre les capteurs. _____	131
Figure 80 : Ensemble de conflits filtrés par un filtre passe-bas. _____	132
Figure 81 : Application des deux experts sur le conflit k_{12} et leur fusion. _____	133
Figure 82 : Application des deux experts sur le conflit k_{13} et leur fusion. _____	134
Figure 83 : Application des deux experts sur le conflit k_{23} et leur fusion. _____	134
Figure 84 : Estimation de l'état de fonctionnement des 3 capteurs. _____	135
Figure 85 : Estimation du coefficient d'affaiblissement en fonction de l'état de fonctionnement des 3 capteurs. _____	137
Figure 86 : Fusion des trois jeux de masses m_1 , m_2 et m_3 prenant en compte la fiabilité des capteurs ainsi que la prise de décision pignistique correspondante. _____	138
Figure 87 : Signaux expérimentaux obtenus à partir des 3 capteurs pour le second scénario. _____	139
Figure 88 : Fusion des trois jeux de masses m_1 , m_2 et m_3 ainsi que la prise de décision pignistique correspondante pour le second scénario. _____	140
Figure 89 : Fusion des trois jeux de masses m_1 , m_2 et m_3 prenant en compte la fiabilité des capteurs ainsi que la prise de décision pignistique correspondante pour le second scénario. _____	141
Figure 90 : Estimation du coefficient d'affaiblissement en fonction de l'état de fonctionnement des 3 capteurs pour le second scénario. _____	142
Figure 91 : Fusion des trois jeux de masses m_1 , m_2 et m_3 prenant en compte la fiabilité des capteurs avec une analyse plus fine du conflit entre capteurs ainsi que la prise de décision pignistique correspondante pour le second scénario. _____	142
Figure 92 : Etat de fonctionnement des trois capteurs pour le second scénario. _____	143
Figure 93 : Chaîne de Markov modélisant un comportement humain. _____	144
Figure 94 : Application de la prédiction Markovienne aux mesures expérimentales de la Figure 68. _____	146
Figure 95 : Calcul de la distance entre observation et prédiction. _____	147
Figure 96 : Signaux expérimentaux obtenus à partir des 3 capteurs possédant une anomalie de comportement. _____	148
Figure 97 : Application de la prédiction Markovienne aux mesures expérimentales de la Figure 96 sur lesquelles apparaissent une anomalie de comportement (à $t=85$). _____	149
Figure 98 : Zoom sur la période d'anomalie de comportement à $t=85$. _____	149
Figure 99 : Conflit $m_{fusion}()$ entre observation et prédiction. _____	150
Figure 100 : Application du premier expert sur le conflit $m_{fusion}()$ entre observation et prédiction. _____	150
Figure 101 : Application du second expert sur le conflit $m_{fusion}()$ entre observation et prédiction. _____	151
Figure 102 : Application du troisième expert sur le conflit $m_{fusion}()$ entre observation et prédiction. _____	151
Figure 103 : Fusion des trois experts et prise de décision sur un problème de comportement. _____	152
Figure 104 : Application de la prédiction Markovienne basée sur la matrice de transition crédibiliste aux mesures expérimentales de la Figure 96. _____	154
Figure 105 : Comparaison des distances de similarité entre observation et prédiction pour les deux matrices de transition. _____	154
Figure 106 : Comparaison des conflits entre observation et prédiction pour les deux matrices de transition. _____	155
Figure 107 : Mise en cascade des approches présentées précédemment. _____	156

Introduction générale

Cadre et objectifs

Le développement rapide de l'informatique a permis de passer de l'utilisation d'imposantes stations de travail à des ordinateurs de poches quasi-invisibles. Aujourd'hui, l'informatique et l'électronique sont entrées dans notre vie de tous les jours. Le moindre objet de la vie courante est désormais pourvu d'intelligence : nos montres, nos clés, nos vêtements, ... Il nous est même parfois difficile de croire que ce que nous utilisons dans la vie courante est pourvu d'électronique.

L'informatique s'est progressivement fait une place importante dans tous types de domaines : l'aviation, l'automobile, l'industrie, les loisirs, ... Son rôle est prépondérant puisqu'elle est capable d'avertir, de conseiller, c'est-à-dire qu'elle est capable de rendre un service à l'utilisateur. Le paradigme de l'informatique évolue rapidement selon la loi de Moore. Auparavant, elle était visible, imposante et inesthétique. Aujourd'hui, elle est dissimulée et rendue invisible grâce à son intégration de plus en plus poussée.

Il s'avère qu'un contexte reste épargné par cette intrusion. Il s'agit de l'endroit où nous passons une grande partie de notre existence : notre habitat. En effet, il existe bien des moyens pour équiper un habitat avec de l'informatique et de l'électronique mais beaucoup moins déployées que dans l'automobile. Actuellement, l'habitat dit « évolué » se résume à l'utilisation exceptionnelle de matériels d'automatisme permettant principalement de commander des objets de l'habitat : fermer les volets, éteindre la lumière, ... Le plus souvent ceci est réalisé à partir d'un ensemble de télécommandes formant un agrégat peu standardisé et encore moins fonctionnel.

Il est nécessaire de modifier l'approche que nous avons pour rendre l'habitat plus évolué voire intelligent. Il ne faut plus rénover un habitat mais concevoir intelligemment les maisons. Cela passe par une réflexion plus profonde sur la structure de l'habitat qui ne doit plus être figée. Une telle approche permet d'obtenir un habitat aux fonctionnalités multiples adaptées à la vie et aux activités de ses occupants. Elle doit induire des économies d'énergie et financières, apporter un confort de vie et de sécurité.

Afin de fournir l'ensemble de ces services, l'informatique doit être omniprésente au sein de l'habitat et doit être capable de deviner les souhaits de ses occupants pour répondre de façon pertinente à leurs besoins. L'informatique omniprésente ou informatique ambiante est actuellement un domaine de recherche en plein essor. Cette thématique de recherche est née au début de ce millénaire grâce à la miniaturisation de l'informatique et à la possibilité de l'intégrer au sein même des objets dit intelligents. L'Europe mise fortement sur l'informatique ambiante¹ dans le cadre du 7^{ème} PCRD (Programme-cadre de recherche et développement européen).

Le cadre général de cette thèse est l'habitat intelligent. Cette étude est née d'une simple question : comment scruter en permanence les intentions des habitants et essayer d'anticiper leurs besoins dans leurs

¹ Ambient Computing

activités présentes ? On comprend que la solution ne peut passer que par l'utilisation de moyens de perception permettant d'observer ce que nous appellerons l'activité humaine. La notion de perception renvoie immédiatement à la notion de capteurs. L'interprétation des données sensorielles est bien sûr ce qui permettra de prendre une décision cohérente. De nombreuses méthodes et solutions ont été proposées pour l'accroissement de l'autonomie en robotique mobile autonome. A partir de ce constat, nous avons réfléchi à l'interdépendance entre l'habitat intelligent et la robotique. Nous pouvons en fait parler de dualité robotique/habitat intelligent. Cette dualité implique une connexion forte avec les méthodes d'intégration de données sensorielles propres à la robotique. Cette notion est le point de départ de nos travaux.

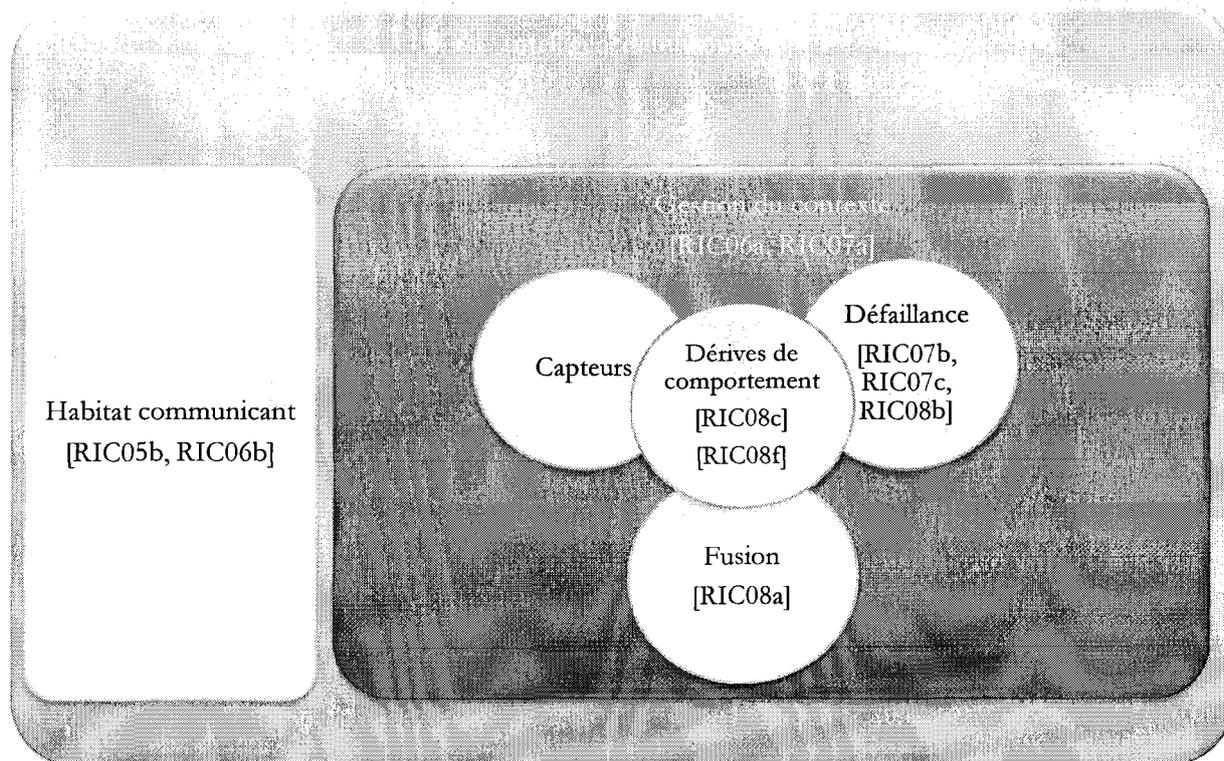
Les objectifs de cette thèse sont multiples. Il s'agit de proposer une architecture de gestion de contexte au sein d'un habitat intelligent permettant une prise en compte aisée des données sensorielles. Le fil conducteur est la production de services les plus adaptés, à la bonne personne, au bon endroit et au moment adéquat. Cette thématique de recherche s'appelle la sensibilité au contexte. Nous traiterons donc la problématique de construction d'un contexte afin de réagir de façon pertinente. Cette architecture est composée de différents niveaux allant de la perception du contexte au service délivré en passant par une phase d'inférence.

Dans ces travaux, nous nous sommes fortement focalisés sur la couche de perception de contexte. Un problème majeur lors de l'intégration des informations sensorielles est la prise en compte de la crédibilité (ou croyance) de celles-ci. Les données capteurs sont imprécises et incertaines et les architectures à contexte existantes ne tiennent pas compte de ces imperfections. Il sera nécessaire de procéder à la combinaison des données sensorielles perçues puisqu'elle permet d'obtenir des informations de meilleures qualités et la construction de contextes complexes. Dans le cadre de la combinaison de données sensorielles, nous abordons le problème de la fiabilité capteur. Nous proposons ainsi deux approches pour la détection de défaillance capteur. La première s'appuie sur la notion de redondance de données sensorielles alors que la seconde s'appuie sur un modèle de comportement normal que le capteur est sensé suivre. Le formalisme pour la propagation des incertitudes au sein de notre architecture de gestion de contexte est la théorie des fonctions de Croyances. Ce formalisme, dont les variantes sont la théorie de Dempster-Shafer et le Modèle des Croyances Transférables (MCT), est utilisé depuis les capteurs jusqu'aux couches les plus hautes liées aux services.

Nos contributions concernent :

- * La description d'une architecture logicielle de gestion de contexte dans l'habitat communicant allant de la perception jusqu'à la prise de décision et au rendu de services contextuellement adaptés [RIC06a] [RIC07a].
- * La fusion de données capteurs crédibiliste MCT pour l'habitat communicant appliquée à la détection de postures humaines [RIC08a].
- * La prise en compte de la fiabilité capteur au sein d'une architecture de fusion de données crédibiliste MCT [RIC07b] [RIC07c].
- * La modélisation et la détection de dérives comportementale d'un capteur dans une architecture de fusion de données crédibiliste MCT [RIC08b] [RIC08f].
- * L'application de la prise en compte de la fiabilité capteur dans le cadre de la détection de dérives de comportements humains [RIC08c].

Le synopsis suivant présente l'imbrication et les relations entre les différentes contributions de ces travaux.



Si le thème général est la sensibilité au contexte au sein d'un habitat intelligent, notre contribution scientifique porte essentiellement sur la fusion de données, en particulier la fusion de données crédibiliste au sens du *MCT*. Cette approche nous permettra d'introduire au sein des processus d'inférence et de prise de décision la notion d'incertitude. Cette dernière est prépondérante pour une cohérence optimale des décisions produites par les couches supérieures de notre architecture. La quantification et la propagation de ces incertitudes au sein d'une architecture à contexte peut être vue comme la contribution majeure des ces travaux.

Organisation de ce mémoire

Le **chapitre 1** permet de présenter le cadre général de cette thèse à partir de l'existant sur l'habitat intelligent. Un « état de l'art » sur les architectures logicielles et les moyens permettant une gestion efficace du contexte dans le cadre global de l'informatique ambiante sera dans un premier temps amené. Les difficultés rencontrées pour l'élaboration de contexte à partir de données capteurs rattachées à la problématique de l'habitat communicant seront analysés. A partir des problèmes évoqués sur l'élaboration de contextes complexes, nous détaillerons les différentes méthodes permettant de combiner des informations au sein d'une architecture de fusion de données. Les problèmes résultants de la combinaison de données hétérogènes et non fiables, nous pousserons à détailler les méthodes de résolution prenant en compte le conflit entre sources fusionnées. Finalement, nous quantifierons l'impact d'un capteur défaillant par rapport à N sources. Des algorithmes de diagnostic seront testés et validés.

Le **chapitre 2** traite de la conception d'une architecture de perception de contexte avec comme base la plateforme logicielle de services OSGi (Open Service Gateway Initiative). L'objectif est de faire interagir au mieux les algorithmes de perception, de fusion, de construction de contexte et de raisonnement. L'approche logicielle orientée service a été retenue lors de travaux antérieurs [REN07]. Dans cette approche, l'apparition ou la disparition des services pendant l'exécution peut être prise en compte. Cette disponibilité dynamique permet de construire des applications pouvant s'adapter à diverses situations telles que la sensibilité au contexte. Nous avons finalement défini une architecture orientée services (SOA :

Service-Oriented Architecture) qui intègre les quatre couches nécessaires à la caractérisation de ce contexte. Elles s'appuient sur le modèle PCIA (Perception-Contexte-Inférence-Action).

La seconde partie de la thèse, la plus importante, porte sur la validité des données issues des capteurs. Deux chapitres sous-tendent cette approche. Le **chapitre 3** se propose d'analyser le conflit résiduel inter sources pour établir une identification du capteur défaillant. Nous décrivons une méthode permettant de détecter et d'isoler un capteur défaillant dans un réseau de N capteurs. Notre travail se base sur l'approche de Dubois et Prade, visant à reporter la masse conflictuelle sur la disjonction d'hypothèses correspondant aux deux capteurs en contradiction. Notre approche a l'originalité de prendre en compte l'évolution temporelle du conflit entre capteurs ainsi que d'identifier le capteur à l'origine du conflit. Cette approche se propose d'analyser le conflit résultant de l'ensemble des fusions deux à deux des N capteurs du réseau. La fusion est réalisée grâce à l'aide de l'opérateur de Smets, qui permet d'isoler le conflit résultant de la fusion. L'analyse temporelle du conflit se base sur la fusion des opinions de deux experts permettant de déterminer si oui ou non deux capteurs sont en contradiction. A partir de l'état de fonctionnement estimé, nous calculons un coefficient d'affaiblissement pour chacun des capteurs. Cet affaiblissement permet de pondérer l'impact d'un capteur défaillant sur le processus de fusion final.

Dans ce même chapitre, nous nous intéressons à la détection de changement d'état inhérente à la surveillance d'une activité humaine, toujours en nous appuyant sur l'analyse du conflit. Nous présumons qu'une des causes possibles de l'accroissement du conflit dues à la défaillance capteur, a été traitée en amont. Cette élimination nous permet de poser un diagnostic fiable sur un comportement anormal, toujours par analyse du conflit, mais cette fois ci en aval. La méthode mise en œuvre détecte une anomalie de comportement dans le cadre de la théorie de Dempster-Shafer. L'approche consiste à fusionner un état observé avec un état prédit. La prédiction repose sur un apprentissage modélisé dans un premier temps par une Chaîne de Markov et dans un second par un paradigme MCT. Les deux approches sont comparées et analysées.

Dans le **chapitre 4**, nous présentons les résultats expérimentaux obtenus relativement aux points exposés dans le chapitre précédent. Dans un premier temps, nous montrons l'impact d'un capteur défaillant dans un processus de fusion statique de N capteurs. Nous observons qu'un capteur défaillant peut fausser la prise de décision finale (obtenue par calcul de la probabilité pignistique) voire même rendre impossible la prise de décision. L'affaiblissement du jeu de masse du capteur défaillant permet d'accroître la croyance envers la décision prise permettant ainsi une décision. Dans un second temps, nous présentons des détections de dérives comportementales à partir du mécanisme de prédiction. Nous montrons la faisabilité de la méthode à partir de données expérimentales et nous comparons les deux approches de prédiction. Le couplage d'une prédiction probabiliste et d'un diagnostic crédibiliste montre que le traitement temporel du signal de conflit reste complexe à analyser. Une troisième évaluation expérimentale porte sur le couplage des deux approches vues précédemment et présente l'impact d'un capteur défaillant sur le diagnostic de dérives comportementales dans l'analyse de l'activité humaine. Nous observons qu'un capteur défaillant peut engendrer une détection et une caractérisation d'anomalie erronée. L'objectif de ce chapitre est de montrer qu'il est possible d'effectuer de la séparation de sources sur le signal du conflit. De plus, nous montrons qu'en atténuant une source d'accroissement du conflit, il devient possible de détecter finement et de manière plus fiable des changements d'état dans l'activité humaine.

Enfin, une **Conclusion Générale** résume les contributions des approches proposées tant sur le versant crédibiliste que sur celui de la prise en compte de l'incertain dans une architecture à contexte. Par ailleurs, nous discutons des perspectives sur les améliorations possibles de notre architecture de gestion de contexte, mais aussi vis à vis de l'aspect apprentissage des fonctions de croyance et des modèles de comportement.

*'In the past, shoes could stink.
In the present, shoes can blink.
In the future, shoes will think.'
Groupe Things That Think du
MIT.*

Chapitre 1

L'habitat communicant au service de l'habitant

1 L'habitat communicant : Une vision futuriste de l'habitat ?

1.1 Introduction

Les possibilités de communication sans fil, Internet et la miniaturisation de l'électronique ont permis aux environnements informatisés ou non, d'entrevoir un bouleversement dans leur mode de fonctionnement. En effet, de nouveaux champs de recherches et usages apparaissent principalement dans les domaines des technologies de l'information et de la communication (ITC) et notamment dans l'habitat qui est maintenant à même de se connecter au monde extérieur. Cette connectivité omniprésente permet d'envisager aujourd'hui l'ouverture de l'habitat vers des ressources extérieures mais surtout aux objets de l'habitat d'avoir une possibilité de communication vers l'extérieur. Sur ces bases, nous pouvons alors évoquer le terme futuriste de 'maison intelligente' où les objets courants, la maison et l'habitant coexistent au sein d'un même écosystème.

1.2 La notion d'habitat communicant

L'habitat intelligent ou 'Smart Home' représente une intégration complexe de technologies, de matériels électriques ou thermiques, de télécommunication, d'informatisation. A l'heure actuelle, le terme 'intelligent' peut paraître surestimé pour ce genre d'habitat. L'intelligence est définie par :

« Qui est pourvu de la faculté de comprendre, qui est capable de concevoir et de raisonner »²

En s'appuyant sur cette définition, il est plus judicieux de parler pour l'instant d'habitat communicant plutôt que d'habitat intelligent. Néanmoins, l'habitat communicant tend à devenir intelligent. Ceci constituera le cadre de nos travaux.

La notion d'habitat communicant est un concept qui émerge en Europe. Le projet 'mam@isonnet'³ à Etiolles (Sud de Paris) est le premier prototype français de maison communicante, né d'un partenariat entre Cisco, le promoteur immobilier Kaufman & Broad et France Télécom. Cet habitat propose principalement des solutions de télécommunications grâce à un Intranet domestique permettant la mise en réseau des équipements⁴, la gestion du confort dans la maison (chauffage, éclairage, volets, alarme), le partage des ressources télécoms et informatiques (téléphonie, internet, imprimantes, périphériques..), un accès à Internet.

La société Legrand propose la maison domotique qui est un lieu de présentation de leurs nouveaux matériels. L'objectif est de renouveler complètement l'appareillage électrique, ses fondements et usages, la distribution des données. Cette maison est basée sur le Serveur Internet Habitat de la société LegrandTM qui a pour fonction de faire la liaison entre les dispositifs électriques et le monde extérieur en s'appuyant sur le protocole TCP/IP.

² Dictionnaire de l'Académie française, 8e édition, 1932

³ www.cisco.com/global/FR/mamaisonnet/pages/1maison/benefice.html

⁴ Basé sur la technologie WRAP (Web Ready Appliances Protocol) permettant un échange d'informations empruntant le réseau électrique ordinaire de la maison.

En Allemagne, se trouve la maison intelligente de Duisbourg nommée InHaus⁵. Cette maison communicante est à l'initiative de Sony, Miele, Deutsch Telecom,... et a pour objectif de proposer des services liés à la gestion du confort et de la sécurité grâce à un réseau domestique permettant l'interconnexion des différents organes de la maison : électroménager, multimédia,... C'est le cas de E2Home⁶ qui est un projet entre Ericsson et Electrolux visant à permettre l'interconnexion entre les produits blancs Electrolux grâce aux moyens de communication fournis par Ericsson. Ce projet s'appuie également sur un réseau domestique commun relié à un réseau extérieur.

De même, on peut citer Philips qui met en œuvre des solutions multimédia centrées principalement autour du confort de l'habitant au sein de son prototype nommé Homelab⁷. A Eindhoven, aux Pays Bas, Philips a créé en 2002 le Homelab, laboratoire d'études comportementales, qui a pour objectif de tester ses prototypes et ses nouvelles technologies en situation réelle.

Aux États-Unis, un nombre relativement important de projets a vu le jour. Parmi les projets plus importants, nous pouvons citer :

1. Adaptive House, University of Colorado: <http://www.cs.colorado.edu/~mozer/nnh/>
2. Carnegie Mellon's Intelligent Workspace : <http://www.arc.cmu.edu/cbpd/iw/index.html>
3. Duke University Smart House : <http://www.smarthouse.duke.edu>
4. Georgia Tech Aware Home : <http://www.cc.gatech.edu/fce/ahri/>
5. Humboldt State CCAT : <http://www.humboldt.edu/~ccat/>
6. MavHome at University of Texas Arlington : <http://mavhome.uta.edu/>
7. Medical Automation Research Center @ UVA : <http://marc.med.virginia.edu/>
8. MIT House_n : http://architecture.mit.edu/house_n/
9. MIT Media Laboratory : <http://www.media.mit.edu/>
10. NC State Solar Center : <http://www.ncsc.ncsu.edu/>
11. Oberlin College : <http://www.oberlin.edu/ajlc/ajlcHome.html>
12. Smart Medical Home : http://www.futurehealth.rochester.edu/smart_home/
13. UNC Office of the Future : <http://www.cs.unc.edu/~raskar/Office/>
14. The Gator-Tech Smart House de l'Université de Floride : <http://www.rerc.ufl.edu/>
15. The MARC Smarthouse de l'Université de Virginie : <http://marc.med.virginia.edu/>

1.2.1 Répondre au problème du maintien à domicile avec l'habitat communicant ?

Nous remarquons que la majorité de projets concernant l'habitat communicant s'attaque au problème du vieillissement de la population. Ceci s'explique par l'accroissement du nombre de personnes âgées et surtout la hausse de l'âge moyen des individus, créant, entre autres, un besoin en terme de solutions alternatives à l'hospitalisation.

⁵ www.inhaus-duisburg.de/en/projektbeschreibung/intelligentes_wohnen.htm

⁶ www.e2-home.com

⁷ www.research.philips.com/technologies/misc/homelab

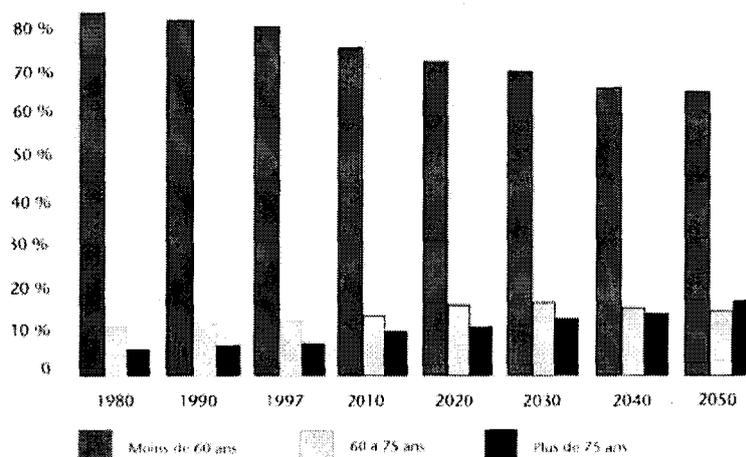


Figure 1 : Population des 60 ans et plus (tous sexes confondus). Évolution depuis 1980 et projections pour 2050 en France.

Sur la Figure 1, nous pouvons observer une projection de la répartition des personnes âgées de plus de 60 ans à l'horizon 2050. On peut ainsi constater que cette population est en augmentation, tandis que les personnes âgées de moins de 60 ans diminuent.

L'engorgement des hôpitaux, le manque de personnel conduit les professionnels de la santé à penser à de nouveaux moyens d'hospitalisation. Le maintien à domicile et en particulier l'hospitalisation à domicile (HAD) est une solution face à ce problème. La HAD « recouvre l'ensemble des soins médicaux et paramédicaux délivrés à domicile à des malades dont l'état ne justifie pas le maintien au sein d'une structure hospitalière. Ces soins doivent être d'une nature et d'une intensité comparables à ceux qui étaient susceptibles de leur être prodigués dans le cadre d'une hospitalisation traditionnelle » [LAT97].

Il y a donc un réel besoin dans le domaine de la HAD de technologies d'aide et de surveillance au sein d'une structure hors hospitalière. L'habitat est donc une solution pour l'hospitalisation à domicile mais aussi pour des personnes victimes d'autres pathologies :

- Les personnes handicapées [MOK07],
- Les personnes ayant une déficience visuelle,
- Les malentendants,
- Personnes souffrant d'une déficience cognitive,

Mais aussi le reste de la population qui peut éprouver le besoin d'être suivi ou aidé dans les tâches quotidiennes.

Le Centre de Santé du futur de l'Université de Rochester a été fondée en 1998 afin de pouvoir fournir aux patients, aux soignants et aux médecins des informations sur leurs états de santé. L'objectif est de proposer un système proactif qui prend en charge les personnes nécessitant des soins. Un système proactif décide seul de l'action à entreprendre en fonction de ce qu'il a observé. Un exemple concret de service fourni de manière proactive est la détection de chute dans le cadre d'une application dédiée aux personnes à mobilité réduite. L'habitat observe la position de l'habitant, sa démarche, sa posture, ses informations physiologiques. En cas de chute, l'habitat détermine ce problème et réagit de manière autonome : il contacte les secours les plus proches. Le système de santé proactif de Centre de Santé du futur permet de mesurer les signes vitaux traditionnels : pression sanguine, pulsation cardiaque et la respiration. Il permet aussi de mesurer des signaux plus complexes tels que la démarche, le comportement, le sommeil. Pour réaliser ces mesures, le système s'appuie sur des capteurs hétérogènes : capteurs infrarouges, capteurs

biomédicaux et des caméras vidéos situés dans 5 pièces de l'habitat. Les données provenant de ces capteurs sont ensuite analysées et traitées afin de détecter des paramètres anormaux et prévenir la famille et le médecin en cas d'anomalies.

La maison intelligente Gator Tech (cf. Figure 2) est une initiative de l'université de Floride visant à accueillir des personnes âgées. Cette maison est un lieu permettant de tester les nouvelles idées développées par les membres de ce projet. Cette maison comporte différentes fonctionnalités visant particulièrement au confort de la personne et à sa sécurité. Afin de remplir ces différentes fonctions, la maison est équipée de capteurs hétérogènes permettant de recueillir des données.

Les Figure 3 et Figure 4 présentent un ensemble de capteurs que nous pouvons trouver dans cette maison intelligente : Un lecteur RFID (Identification par Radio-Fréquence) afin d'identifier la personne désirant entrer dans la maison. Ce système permet aux personnes de s'identifier grâce à un badge RFID. Une fois le badge reconnu, la maison ouvre la porte et autorise l'entrée de la personne. Une caméra permet de prévenir la présence de visiteurs, évitant à l'habitant de se déplacer.

Un troisième procédé, très intéressant, est dédié à la localisation des personnes et éventuellement à détecter une chute. C'est un ensemble de capteurs formant un plancher intelligent (smart floor). Ce plancher est équipé de dalles sur lesquelles sont disposés des capteurs de pression. Lorsqu'une personne marche sur une dalle, son capteur est activé et fournit une information indiquant qu'il y a pression sur cette dalle. Ainsi, si plusieurs dalles sont activées simultanément, on peut penser à une personne allongée sur le sol ce qui indique une chute.

Cette maison est pourvue de prises intelligentes (smart plugs) [ZAB05], capteurs permettant de commander à distance un appareil électrique. Ce capteur permet aussi de déterminer quel est l'objet connecté à sa prise grâce à un lecteur RFID permettant de lire un tag RFID disposé sur le cordon d'alimentation. Ainsi quelque soit l'endroit où l'on connecte un objet électrique, la maison pourra le détecter, connaître les caractéristiques de l'objet connecté et l'activer ou le désactiver en modifiant l'état de son alimentation.



Figure 2 : Gator Tech Smart Home.



Figure 3 : L'identificateur RFID et la caméra d'avertissement de présence avec visionnage du visiteur sur l'écran du salon.

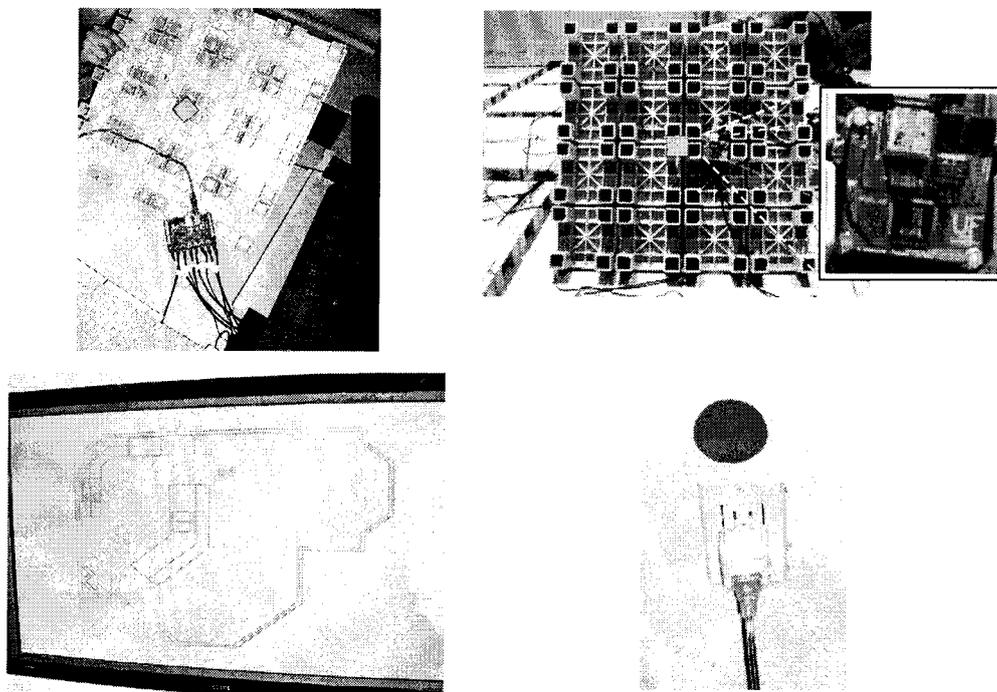


Figure 4 : Une dalle du smart floor équipée de son capteur de pression (1ère version filaire et seconde version sans fil) avec visionnage du visiteur sur l'écran du salon et une prise équipée du « Smart Plug ».

Cette maison est certainement la maison la plus avancée en terme d'équipement de perception. On peut y noter aussi l'utilisation d'une caméra [ZAB06] et d'un robot aspirateur *Roomba*⁸ [ZAB07] afin de détecter les objets présents dans la maison (canapé, table,...) et de les répertorier.

Un constat simple émerge de cette présentation non-exhaustive : l'habitat communicant doit être capable d'observer en permanence ses habitants dans le but de détecter d'éventuels problèmes. Nous constatons que la plupart des projets d'habitat communicant basent leur travail sur la notion de perception afin de détecter les problèmes.

1.2.2 Un habitat communicant pour la surveillance et/ou le confort

Grâce aux exemples présentés précédemment, deux principales orientations sont exprimées. Certains voient la maison communicante comme un moyen de surveiller en permanence des personnes âgées ou handicapées tel que le Centre de Santé du futur de l'Université de Rochester tandis que d'autres portent principalement sur le multimédia et le confort tel que le HomeLab de Phillips.

Pour le premier cas, ces travaux s'attachent à la sécurité des personnes ou il s'agit de connaître en permanence leur état de santé. Pour cela, il faut acquérir des informations physiologiques de la personne (ECG, tension,...) mais aussi des données relatives à son comportement. Les informations physiologiques de la personne permettent de connaître l'évolution de la santé de la personne afin de détecter des signes avant-coureurs (crise cardiaque,...). La capacité d'analyser le comportement d'une personne fournit des indications précieuses sur sa santé. En effet, si une personne vient à chuter, cette chute est sûrement due à un malaise ou alors un acte maladroit qui va probablement engendrer un problème de santé par la suite (rupture du col du fémur,...). La maison intelligente de Georgia Tech est un exemple de maison communicante destinée à recevoir des personnes âgées et à leur proposer des services visant à les maintenir à leur domicile plutôt que dans des instituts spécialisés.

Pour le second cas, portant principalement sur le confort et l'accès au multimédia, les habitats se destinent à fournir du confort thermique (mise en route du chauffage automatiquement selon les préférences des habitants comme dans l'Adaptive House du Colorado,..). Ces habitats sont destinés à répondre aux attentes de ses propriétaires en terme de confort : optimisation du chauffage dans un but d'économie d'énergie, applications multimédia,...

Néanmoins, ces deux notions sont relativement proches. En effet, seul le rôle de la maison diffère dans ces deux cas alors que le substrat technologique reste similaire. La maison se contente d'observer la scène afin de fournir le service le plus adapté à la personne. Il faut alors que l'habitat acquière un nombre important d'informations relatives à l'environnement des habitants.

1.3 Comment rendre un habitat communicant ?

Quand certains voient dans le terme 'maison intelligente', une maison équipée de volets électriques, d'un système d'alarme à activer manuellement en cas de départ, d'autres se préoccupent plutôt d'installer une intelligence supplémentaire permettant de rendre un service particulier en fonction d'une situation. Le terme maison intelligente prend tout son sens. Une maison capable de détecter la chute d'une personne et

⁸ <http://www.irobot.com/>

d'alerter les secours ne repose pas sur la simple utilisation d'un médaillon sur lequel il faut appuyer en cas de chute. Alors, la maison observe la scène et détermine s'il y a chute ou non. En cas de chute, elle choisit les ressources les plus adaptées pour contacter les secours.

Il est important que l'habitat intelligent fournisse le service le plus adapté à la situation courante. Nous appellerons ce genre de service, services à l'habitant ou services contextualisés. La notion de contexte est présentée dans la section suivante.

Comme nous l'avons vu dans le paragraphe précédent, quelque soit l'objectif de l'habitat communicant, il est important d'acquérir des informations sur les habitants mais aussi sur son environnement. L'environnement d'un habitant revient à considérer tout objet de l'habitat mais aussi l'habitat lui-même. Aussi, nous pouvons nous demander comment il est possible de prendre en compte cette masse d'informations de natures fortement hétérogènes.

Il faut alors un système intelligent au sein de la maison afin d'agrèger et d'interpréter toutes ces données, d'informer les ressources extérieures (pompiers en cas d'incendie, SAMU en cas de chute,...). La maison devient intelligente par l'intermédiaire d'entités informatiques discrètes dont le rôle est d'observer et d'agir en conséquence en fournissant le service le plus adapté. Afin de communiquer avec les ressources extérieures, il faut un accès vers l'extérieur de l'habitat communicant (connexion *ADSL* par exemple) connecté en permanence afin de rendre le service le plus rapidement possible.

Schématiquement, un habitat communicant peut se décrire comme un habitat ouvert sur le monde. L'habitat communicant est équipé d'objets communicants permettant soit d'interagir avec l'individu, soit de l'observer. Les objets communicants (*Smart Objects*) émergent actuellement dans notre société sans que nous nous en rendions compte. On peut noter l'apparition de *Smart Door* [SIE04], *Smart Carpet* [SAV07], *Smart Bed* [MAR05], ... Ainsi, tout objet peut actuellement se décliner sous une forme intelligente et communicante. Hubert Guillaudva [GUI07] va même jusqu'à dire que les objets en devenant intelligents et communicants deviennent une extension de nous-mêmes.

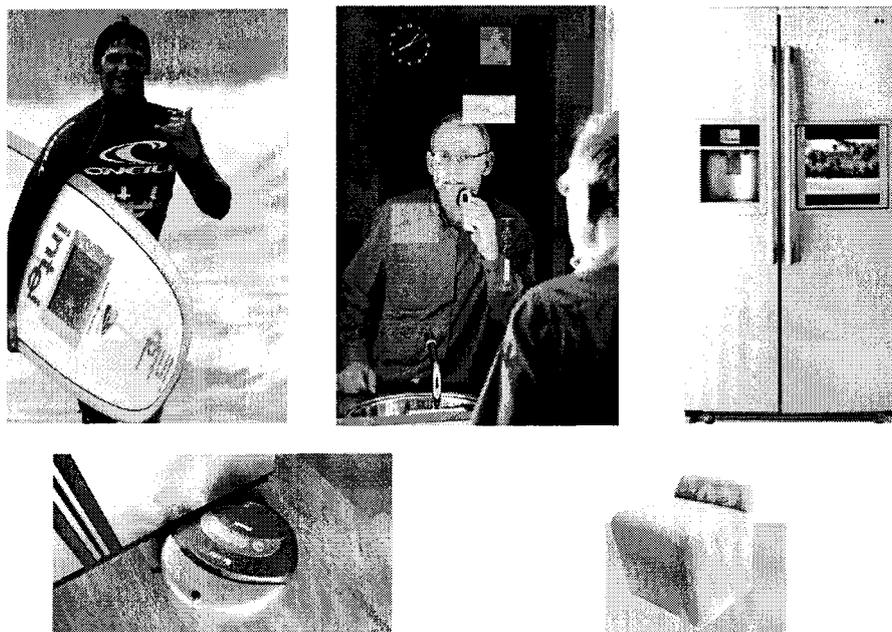


Figure 5 : La smart Board d'Intel, le miroir communicant de Phillips HomeLab, le réfrigérateur communicant de LG, l'aspirateur robotisé Roomba, le siège robotisé de Jelte van Geest (<http://www.jeltevanageest.nl/>), autant d'objets communs qui deviennent communicants.

Ces objets communicants de natures fortement hétérogènes se trouvent disséminés sur un réseau interne à l'habitat. Ensuite, nous trouvons une passerelle résidentielle. Le rôle de cette passerelle est d'offrir un accès à l'extérieur de la maison par le biais d'un réseau Ethernet et/ou Internet. Ceci offre la possibilité à des habitats de communiquer ensemble ou d'accéder à des services extérieurs. Un fournisseur de service est alors présent sur le réseau. Ce fournisseur de service propose des services adaptés à l'habitat. A la demande, le fournisseur télécharge le(s) service(s) désiré(s) vers la passerelle résidentielle. Le rôle de la passerelle est alors d'activer ce(s) service(s) afin de le(s) proposer aux habitants (cf. Figure 6).

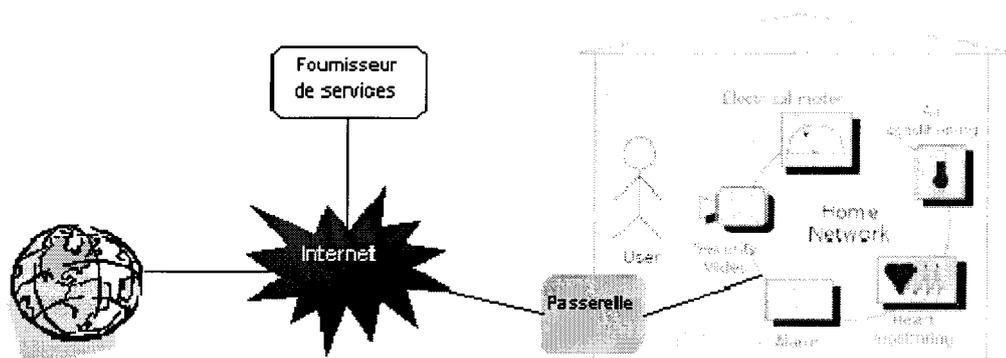


Figure 6 : Principe de la maison communicante.

Comme nous pouvons le voir sur la Figure 6, l'habitat est équipé d'objets communicants que nous pouvons ranger en deux principales familles. La première famille d'objets communicants correspond aux objets permettant une interaction avec l'habitant. La seconde famille regroupe les objets observant ce qui se passe à l'intérieur de la maison ainsi que les objets permettant d'offrir des services à l'habitant.

La première famille d'objets permet une interaction avec un individu. L'interaction permet à un individu de définir ses souhaits auprès de l'habitat par l'intermédiaire d'un objet. L'individu informe l'habitat de ses

désirs de toutes les manières possibles (parole, geste,...). Dans cette première famille nous retrouvons les travaux liés à l'IHM, Interface Homme Machine ou plus récemment l'IHE, Interaction Homme Environnement. L'IHM est le fruit de recherches portant sur les interactions possibles entre les individus et les machines informatisées. L'IHE est l'évolution de l'IHM et concerne l'interaction naturelle entre un individu et son environnement.

Les objets deviennent intelligents et communicants et permettent d'interagir directement ou indirectement avec les individus. C'est ce que l'on nomme aujourd'hui l'IHE, c'est-à-dire l'informatisation dissimulée des objets de la vie courante. Il n'y a plus une interaction directe entre l'individu et la machine. Néanmoins l'habitant doit émettre un souhait qui est analysé par les objets rendus intelligents. Ainsi, l'IHM devient l'IHE par le biais d'objets intelligents et communicants.

La seconde famille d'objets permet à l'habitat d'observer un individu et de lui fournir un service. Il n'y a pas d'interaction directe entre l'individu et l'habitat. Dans ce cas, l'interaction est tacite, c'est à dire que l'habitat détermine seul le service le plus adapté à rendre en s'appuyant sur des informations ou indices perçus dans l'habitat.

1.4 Une approche : L'intelligence ambiante

La notion d'intelligence ambiante peut être perçue comme une généralisation de la notion d'habitat communicant. La notion d'intelligence ambiante (AMBIANT Intelligence ou AMI) a été abordée par Philips Consumer Electronics dans un discours sur l'habitat intelligent en 2000. L'informatique ambiante est une notion qui découle de l'intelligence ambiante et qui vise à intégrer des microprocesseurs dans les objets de la vie quotidienne afin de les rendre invisibles à nos yeux et d'utilisation naturelle.

J.C.R. Licklider [LIC60] fut une des premières personnes à introduire la notion de symbiose entre l'humain et l'ordinateur. Il imaginait un monde dans lequel les hommes et les ordinateurs étaient capables de communiquer afin de décider et de contrôler des situations complexes. C'est en 1991 que le chef du département informatique du Xerox Parc, Mark Weiser [WEI91], introduisit le concept plus général d'informatique omniprésente (*ubiquitous computing*). Il découpe l'évolution de l'informatique en trois parties (voir Figure 7) [WEI96]. La première partie évoluant de 1940 à 1980, présente un ordinateur pour plusieurs individus. La seconde évolution, de 1980 à 2005, présente un ordinateur pour un individu. Cette évolution est encore d'actualité puisque c'est celle que nous connaissons actuellement avec nos PC personnels. Néanmoins, nous sommes dans une période charnière et nous entamons la troisième évolution de l'informatique, qui prévoit plusieurs ordinateurs que se partagerons chacun d'entre nous.

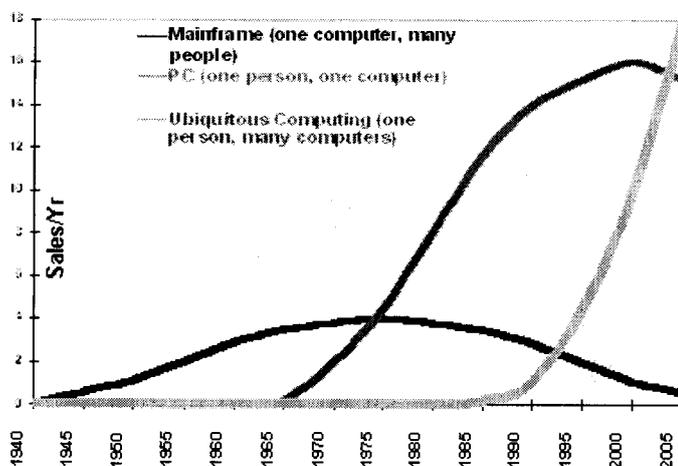


Figure 7 : L'évolution de l'informatique selon Mark Weiser.

La troisième évolution décrite par Mark Weiser se vérifie de nos jours. Les solutions de connexion sans fil aux réseaux ou à Internet, permettent à l'homme d'être en mouvement en permanence et d'avoir à disposition toutes les informations souhaitées. On peut ainsi définir l'homme « connecté » [BER03]. Ce principe est actuellement appelé informatique diffuse (pervasive computing) [NIS00] qui est simplement le terme utilisé de nos jours pour désigner l'ubiquitous computing.

En s'inspirant du concept introduit par Mark Weiser, on peut s'interroger quant à l'évolution des communications liées à cette évolution informatique. Il semblerait que les entités communicantes deviennent différentes. Parmi les modes de communication, la communication de personne à personne fut la première et la plus naturelle. Ensuite, on a vu apparaître la communication de la personne à l'objet en permettant une interaction entre l'utilisateur et l'objet. Une troisième source de communication apparaît maintenant avec la communication d'objet à objet [COR02] où les objets sont maintenant équipés d'intelligence et de moyens de communication. C'est ce que l'on nomme le machine-to-machine. (M2M [LAW04]).

Grâce à l'informatique diffuse, nous pouvons voir des systèmes informatiques émerger dans nos environnements. Chez soi comme au travail, nous sommes entourés d'objets intelligents et communicants [DRO01]. Nous pouvons alors imaginer que dans un futur proche, nous allons être entourés d'objets intelligents et communicants. Ils seront dissimulés dans notre environnement, non intrusifs, interconnectés entre eux, dynamiques en étant autos-configurables. Ils seront ainsi mobiles, autonomes, capables de s'adapter et d'être présents dans notre environnement de tous les jours.

Pour Lee [LEE02], l'environnement pervasif peut être vu comme la réunion de trois couches étroitement liées. La première couche suppose que des systèmes sont installés pour collecter des informations à propos du monde réel et pour réagir en fonction des actions des utilisateurs ou des autres systèmes. Ensuite la seconde couche reçoit en continu les informations, les organise et les synthétise pour permettre au système de réagir selon le monde réel. Enfin la dernière couche, permet de prendre des décisions et de choisir les actions à entreprendre qui auront un impact physique sur le monde réel. Ainsi la Figure 8 résume globalement ce qu'est un environnement pervasif.

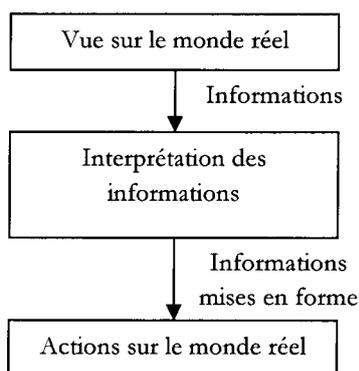


Figure 8 : Fonctionnement d'un environnement *pervasif*.

Pour illustrer la Figure 8, nous pouvons prendre l'exemple de l'allumage automatique de l'éclairage dans une pièce de la maison. On voit une première partie permettant d'avoir une information de l'état de luminosité de la pièce grâce à un capteur de luminosité. Ce capteur fournit une information sur le taux d'illumination de la pièce. Le signal mis en forme indique que le taux d'illumination est alors faible. L'action entreprise est d'immédiatement activer l'éclairage de la pièce.

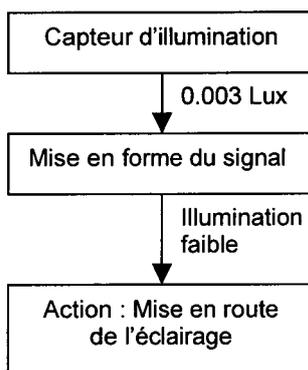


Figure 9 : Exemple d'un système implémenté dans un environnement *pervasif*.

Dans le cadre d'application *pervasif*, l'action ne sera pas obligatoirement une action de type matérielle. L'action pourra bien entendu être logicielle en indiquant à l'utilisateur qu'un événement a lieu (par exemple, envoi d'un *sms*).

Nous venons de voir que l'environnement de l'utilisateur n'est plus statique comme auparavant mais est devenu dynamique. L'homme est au centre d'un environnement qui s'adapte automatiquement à ses besoins afin d'être le plus dynamique possible. Les applications et services fournis aux utilisateurs doivent ainsi devenir dynamiques. Dans le cadre de l'intelligence ambiante, Aarts [AAR03] définit 5 points clefs qui vont permettre de développer l'intelligence ambiante :

- Anticipation : les systèmes AMI sont capables d'anticiper les besoins des utilisateurs
- Adaptation : la réponse d'un système AMI est capable de s'adapter en fonction de l'utilisateur
- Personnalisation : les systèmes AMI peuvent se redéfinir pour répondre aux besoins de l'utilisateur
- Sensibilité au contexte : les systèmes AMI sont capables de reconnaître les utilisateurs et les contextes d'utilisation.
- Intégrés : dans un système AMI, les réseaux sont intégrés aux objets pour une communication optimale.

On peut ainsi s'apercevoir que les préoccupations de l'intelligence ambiante sont identiques à celles de l'habitat intelligent : fournir le service qui semble le plus approprié face à une situation donnée. Ainsi, il faut que l'habitat soit capable d'observer l'individu et d'agir en conséquence. Pour cela, l'habitat est équipé d'un système permettant d'observer l'habitant et de recueillir des informations pertinentes sur son environnement proche afin de lui fournir le service le plus adapté à la situation en cours. C'est ce que nous avons définis comme seconde famille dans l'énumération des objets communicants : des objets permettant d'observer l'habitant.

Ces informations sont appelées informations de contexte et les systèmes gérant ce contexte sont appelés des systèmes sensibles au contexte. Le contexte est ainsi toujours défini autour de l'habitant afin de lui rendre le service le plus adapté.

2 Le contexte - savoir ce qui se passe pour savoir quoi faire

2.1 Présentation

La sensibilité au contexte est une notion relativement récente et la communauté travaillant sur cette notion est principalement celle de l'IHE. L'environnement *pervasif* doit s'adapter aux changements de situation.

Ainsi les systèmes équipant cet environnement pervasive, doivent être réactifs aux changements qu'ils pourront détecter. Un tel système doit alors être sensible au contexte afin de s'adapter aux changements de situations [WAN04].

Typiquement, selon Dey [DEY01a] et dans le sens du pervasive computing, le contexte se définit par « toutes informations pouvant être utilisées pour caractériser la situation d'une entité ». On peut alors dire que le contexte est une information caractérisant la situation. Cette information est dans ce cas, la somme de données recueillies auxquelles on ajoute des méta-informations [ZIM04]

Nous trouvons un ensemble de définitions du contexte dans la littérature. La définition proposée par Gaëtan Rey [REY04] est celle de l'ensemble de circonstances dans lesquelles se situe une action, ou qui entourent un fait et permettent de le comprendre. Ainsi, il définit un système interactif comme étant sensible au contexte lorsqu'il est capable d'identifier les circonstances qui entourent l'action (et notamment celle de l'utilisateur) ceci pour offrir le service le plus adapté.

La notion de contexte peut alors être utilisée dans deux cas distincts. Premièrement, le système décide seul de l'action à entreprendre. C'est ce que nous désignerons par interaction proactive. Dans le second cas, le système est à l'écoute des demandes de l'utilisateur. C'est ce que nous désignerons par interaction tacite.

Si nous appliquons cette approche du contexte au sein de l'habitat communicant, nous pouvons schématiser l'habitat communicant comme un moyen de comprendre le contexte dans lequel évolue l'habitant afin d'offrir le service le plus adapté à la situation en cours. En effet, le rôle principal de l'habitat, dans sa version intelligente, est d'offrir le bon service à la bonne personne, au bon moment et sans limitation spatiale. Ceci ne peut être réalisé qu'en permettant la prise en compte du contexte.

Afin de prendre en compte le contexte au sein de l'habitat, celui-ci doit être équipé d'entités permettant d'appréhender ce contexte, de l'analyser, de prendre une décision et d'appliquer cette décision.

Pour définir une architecture de perception de contexte, nous nous appuyons sur les quatre couches définies par Joëlle Coutaz [COU05]. Nous voyons sur la Figure 10 les différents niveaux d'abstraction nécessaires à l'élaboration d'un système sensible au contexte. Pour alléger notre architecture, nous fusionnons les deux niveaux « perception » et « sensing » que nous modéliserons par le simple terme de 'perception'. Ainsi, nous fusionnons ces deux niveaux car ils sont étroitement liés. En effet, la couche 'sensing' seule n'a aucun intérêt dans le sens où elle fournit des informations sans sémantique. Nous considérons que la partie intéressante se situe au niveau de la couche 'perception' qui fournit des données avec une sémantique plus importante, comme sont capables de nous fournir les capteurs actuels.

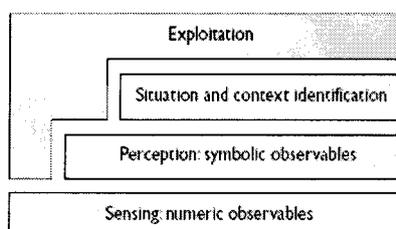


Figure 10 : Les différents niveaux d'abstraction nécessaires à l'élaboration d'un système sensible au contexte [COU05].

Pour déterminer un contexte, il faut que le système sensible au contexte soit alimenté en données contextuelles ayant un niveau sémantique plus ou moins élevé. Ces informations contextuelles sont représentatives de l'environnement et perçues dans celui-ci. Lorsque nous faisons référence à la perception, nous sommes renvoyés à l'utilisation de capteurs qui permettent de quantifier un phénomène physique. Ainsi, notre architecture de perception de contexte se décompose de la manière suivante:



Figure 11 : Architecture de perception de contexte.

Sur la Figure 11, nous observons trois parties. La partie la plus basse de l'architecture porte sur les capteurs. C'est à ce niveau que s'effectue l'acquisition des données contextuelles. Ensuite, une seconde partie porte sur le contexte. Cette partie est chargée d'exploiter les données contextuelles afin de fournir des informations de sémantique plus haute, plus précises. La troisième partie porte davantage sur l'utilisation/gestion de ces données contextuelles.

Deux grandes communautés de recherche travaillent activement sur la problématique du contexte. D'un côté, la communauté informatique et de l'autre, la communauté militaire. Dans le domaine informatique, on parle d'attention au contexte (context awareness) et dans le domaine militaire, de 'situation awareness' [CHA03b]. Ces deux termes expriment la même idée : l'envie de «savoir ce qui se passe pour savoir quoi faire ». Les deux communautés sont d'accord sur les moyens à mettre en œuvre pour réaliser de tels systèmes : il faut d'abord acquérir des informations sur le contexte actuel afin de comprendre ce contexte. Le système est alors capable de décider et d'agir en conséquence.

Les principales caractéristiques de l'information de contexte selon quatre principaux axes [DEY01b] :

→ Les caractéristiques temporelles :

- * Les informations de contexte peuvent être soit dynamiques soit statiques,
- * L'historique de contexte peut former une partie de la description du contexte.

→ Les informations de contexte sont imparfaites :

- * Inconsistance,
- * Incomplète.

→ Diversité des représentations du contexte :

- * Traitements utiles pour passer d'une information source (capteur) à une information exploitable.

→ Relations entre informations de contexte :

- * Relation directe,
- * Relation par dérivation.

Cette description relativement complète présente les principaux points freinant le développement des applications sensibles au contexte.

La temporalité dans le contexte

Le premier point abordé propose de considérer la temporalité dans le déroulement du contexte. La prise en compte de la temporalité est un problème majeur dans la caractérisation des activités d'individus [STI07] par exemple. Une activité ne peut se déterminer à partir d'une vue statique à un instant t mais à partir d'un ensemble d'actions caractérisant une activité générale. Par exemple, la préparation du café est une activité complexe à représenter nécessitant la prise en compte de la temporalité dans l'organisation des actions (prendre du café en poudre, prendre un filtre à café, prendre de l'eau, ... [STI08]).

Imperfection des données

L'imperfection des données est abordée. Ces imperfections regroupent les problèmes dus aux méthodes de mesures, de bruits de mesures, d'approximation, ... tout élément pouvant venir entacher d'erreurs une information acquise. Cela comprend l'incomplétude des données, l'inconsistance des données.

La modélisation du contexte

Cette description aborde aussi le problème de la modélisation du contexte qui est la problématique dominante dans les recherches sur le contexte. En effet, comment représenter le contexte de manière informatique à partir de données de bas niveau provenant de capteurs par exemple ? Il n'est pas simple de répondre à cette question car le problème se situe à plusieurs niveaux. Le premier se situe sur la couche capteurs : que signifient les données capteurs perçues ? Le second se situe dans la gestion des données capteurs : les données sont-elles redondantes, complètes ? Le troisième se situe dans l'utilisation des données capteurs : que dois-je faire face à cette donnée capteur ?

Relations entre contextes

Finalement, les relations entre données contextuelles sont abordées. Ce point pose la question de l'existence de relations entre données contextuelles permettant de définir des éléments de plus haut niveau.

Ces quatre points sont autant de problématiques ouvertes. Plusieurs approches tentent de répondre à ces questions. Pour commencer, nous allons ainsi voir les difficultés rencontrées au plus bas de l'architecture, c'est-à-dire au niveau de la perception du contexte.

2.2 La perception du contexte : les difficultés rencontrées

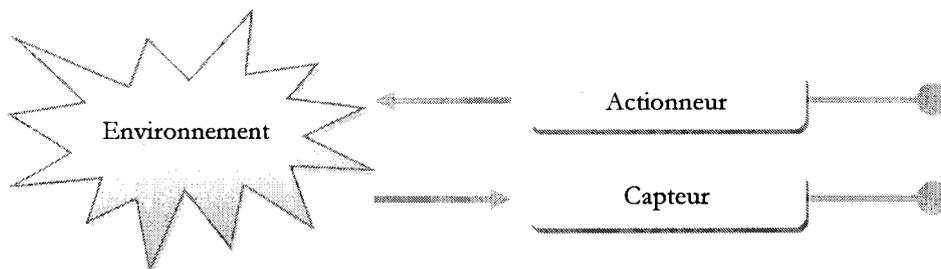
Les capteurs sont le point de départ de tout système sensible au contexte. Nous trouvons différentes sortes de capteurs que l'on peut qualifier de passif ou d'actif. Les capteurs actifs fournissent une information en continu ou de manière périodique mais c'est le capteur seul qui décide de l'envoi des données. Le système tiers désirant connaître ces données nécessite simplement de se mettre à l'écoute du capteur. Parmi les capteurs actifs, on observe deux types de données que peut potentiellement produire un capteur. Des données numériques qui correspondent à une donnée brute (plus ou moins traitée au préalable) ou des données événementielles correspondant à une information de sémantique supérieure à une donnée brute. Les données événementielles sont souvent liées à des événements critiques (alarme incendie, intrusion, ...)

Contrairement, un capteur passif répond à une sollicitation d'un tiers. Le tiers demande une donnée auprès du capteur, ce dernier lui répondant en fonction de sa demande. Ici la notion d'événement n'est pas

introduite vu le caractère du capteur. Celui-ci doit être interrogé afin de fournir une information. Un événement se doit d'être traité de manière synchrone et non asynchrone comme l'impose un capteur actif. Une seconde caractéristique de certains des capteurs passifs est la possibilité de fournir leur état de fonctionnement et la possibilité de modifier leur état de fonctionnement. Un tiers peut alors mettre en arrêt un capteur en agissant sur son état.

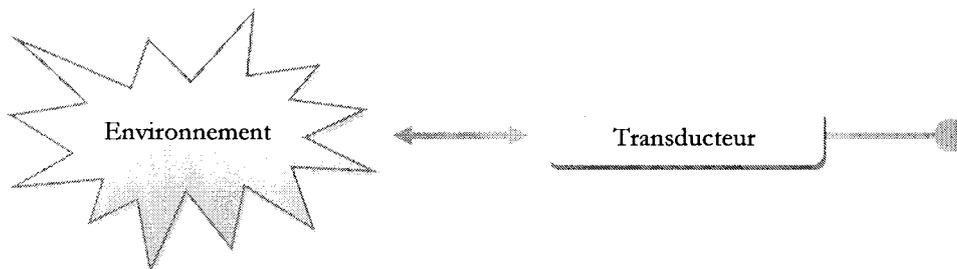
L'ensemble des actionneurs est le second point. On peut voir un actionneur comme un capteur inversé. Le capteur perçoit une information de l'environnement et traduit cette information en une grandeur électrique alors qu'un actionneur reçoit une grandeur électrique afin d'agir sur une information de l'environnement (un moteur électrique par exemple).

Schématiquement :



Comme les capteurs, les actionneurs peuvent être passifs ou actifs. Dans le cas des actionneurs passifs, un tiers émet une consigne vers l'actionneur et ce dernier essaie de modifier l'environnement. Les actionneurs actifs peuvent fournir l'état d'une action. Ceci permet à un tiers de vérifier si l'actionneur a correctement effectué la consigne envoyée précédemment. Donc, tout comme le capteur, un actionneur peut recevoir des informations et il peut fournir des informations.

D'une manière plus générique, nous pouvons regrouper les capteurs et les actionneurs sous le terme transducteurs. Les transducteurs sont des dispositifs traduisant proportionnellement un phénomène physique en un autre phénomène physique.



Actuellement, les capteurs que nous avons à disposition fournissent des données diverses selon un protocole propre au concepteur du capteur. Il est alors problématique de trouver un modèle générique adapté à chacun des capteurs. Il faut considérer les différents protocoles envisageables et les adapter à nos besoins. Une solution consiste à utiliser un adaptateur qui permettra de « traduire » les données provenant des capteurs en données standardisées compréhensibles par notre architecture.

Néanmoins, on dénote un désir de standardisation dans le domaine du capteur. La norme IEEE 1451 [LEE00] a pour objectif de définir une interface standardisée pour les réseaux de capteurs. Celle-ci offre la possibilité de configurer automatiquement les capteurs en y intégrant une interface spécifique et une fonction d'auto-reconnaissance. Cette norme spécifie le format d'une fiche technique embarquée dans la

mémoire du capteur. Cette fiche possède, entre autre, l'identité du capteur (son fabricant, numéro de série,...), ses caractéristiques (sensibilité,...) ainsi que la possibilité à l'utilisateur d'ajouter des données personnelles (localisation du capteur,...)

Cette norme n'étant pas totalement achevée, la réalisation d'applications basées sur des capteurs se tourne vers des méthodes plus empiriques. Les concepteurs tentent de donner une description la plus complète en décrivant les capteurs, leurs données,...

On peut citer les travaux de Chapron [CHA05] qui propose une taxonomie des capteurs contextuels. Il s'intéresse en particulier à la conception de systèmes sensibles au contexte. Il propose une méthode d'aide dans le choix de capteurs contextuels face à un problème posé mais ne s'occupe pas de l'interfaçage de ces capteurs avec un système existant. Jurdak [JUR04] propose un modèle d'ontologie pour caractériser un réseau de capteurs. Ces travaux ne s'attardent pas à décrire proprement les capteurs mais le réseau sur lequel les capteurs sont connectés.

Ces travaux ne prennent pas en compte la nature fortement hétérogène des données fournies par les capteurs. Les phénomènes observés par les capteurs étant de natures différentes, les données qui seront fournies par les capteurs seront elles-mêmes de natures hétérogènes. Il est donc important de passer par une phase de description des capteurs et de leurs données afin de les regrouper ou de les agréger. C'est le rôle du modèle de contexte qui permet de représenter le contexte obtenu à partir de ces capteurs. La section suivante présente les approches existantes permettant de représenter le contexte.

2.3 La représentation du contexte

Le modèle de contexte permet de représenter le contexte en s'appuyant sur les données issues des capteurs. Behloui [BEH06] classifie la modélisation du contexte selon quatre types d'approches :

- * l'approche paires/triplets,
- * l'approche orientée modèles,
- * l'approche basée sur la logique,
- * l'approche orientée ontologie.

2.3.1 L'approche paires/triplets

Cette approche permet de modéliser les valeurs contextuelles observées en s'appuyant sur des structures de données très simples pour les décrire. Schilit [SCH93] propose de sauvegarder chaque observation contextuelle dans une variable d'environnement constituée d'une paire clés/valeurs. Schmidt [SCH99] utilise un triplet pour modéliser le contexte. Il est constitué de l'observable, de la valeur contextuelle observée et d'un degré de certitude sur la cohérence des données observées. Nos travaux s'appuient sur cette constitution afin de prendre en compte la certitude envers une donnée contextuelle. Le Context Toolkit de Dey [DEY01a] fournit une représentation du contexte sous forme de triplets attribut-valeur.

Bien que la modélisation paires/triplets utilise des structures de données simples à gérer, cela ne permet pas de décrire complètement le contexte. De même, elle ne permet pas de décrire les relations qui peuvent exister entre les informations contextuelles ainsi que la notion de temporalité. Le dernier point freinant l'utilisation d'un modèle paires/triplets pour la représentation du contexte est l'impossibilité de raisonner sur le modèle précédemment établi.

2.3.2 L'approche orientée modèles

Cette approche utilise des modèles formels pour représenter les informations contextuelles. L'objectif d'une telle approche est de permettre l'encapsulation du contexte ainsi que sa réutilisation.

UML (Unified Modeling Language) est utilisé dans la modélisation du contexte. Sheng [SHE05] propose ContextUML, un méta-modèle basé sur une extension d'UML qui permet de modéliser le contexte dans le cadre de la fourniture de services Web. Henricksen propose le langage CML (*Context Modeling Language*) [HEN04], langage orienté objet dérivé de l'ORM (Object Role Modeling) [HEN05] permettant la modélisation des caractéristiques des informations contextuelles et de leurs propriétés d'une manière formelle.

Le langage de balise tel XML, a été utilisé par Pascoe [PAS97] dans le cadre du protocole ConteXtML [RYA05] pour décrire le contexte observé associé à un utilisateur et pour décrire le profil de l'application. Ce profil représente le contexte nécessaire à cette application pour qu'elle puisse s'exécuter. Néanmoins, ce langage décrit un format d'échange de données mais ne permet pas de décrire le contexte. Capra [CAP01] utilise le langage XML afin de modéliser le profil de l'utilisateur ainsi que le profil d'une application. Cette modélisation est utilisée par CARISMA [CAP03] un intergiciel réflexif, permettant la reconfiguration ou l'adaptation dynamique d'une application sensible aux changements de contexte.

Ainsi l'approche orientée modèles permet une description plus riche du contexte que les approches paires/triplets. Cette approche offre la possibilité de réutilisation de modèles existants. Les modélisations basées sur un langage de balises permettent de décrire des profils ou des contextes simples, mais ne permettent pas de décrire des relations de dérivation et de dépendance entre les informations contextuelles. Les modélisations basées sur UML permettent quant à elles de décrire ces relations entre informations contextuelles mais ne prennent pas en compte la description des dépendances entre ces informations, ni la qualité ou la temporalité des données. CML apporte des solutions à ces problèmes en proposant un modèle avec visualisation graphique qui permet de typer le contexte, et de décrire un ensemble de relations entre plusieurs contextes observables. Finalement, cette approche ne permet pas le raisonnement sur le contexte de manière native.

2.3.3 L'approche basée sur la logique

Les modèles basés sur la logique utilisent l'algèbre booléenne et la logique du premier ordre pour modéliser le contexte. La logique permet de définir des conditions qui nécessitent de déduire des faits ou des expressions à partir d'un autre ensemble d'expressions ou de faits. Par conséquent, dans les modèles basés sur la logique, le contexte est défini comme des faits, des expressions ou des règles. Les travaux pionniers de McCarthy [MCC93] [MCC94] définissent le contexte comme une entité mathématique abstraite.

La logique est particulièrement bien adaptée au monde de l'intelligence artificielle dont l'objectif est le raisonnement sur les données collectées. Prolog est l'un des plus célèbres langages de programmation logique [BRA00]. Prolog repose sur le calcul des prédicats du premier ordre. Ce langage est fondé sur des règles permettant de raisonner sur les clauses de Horn. Dans le même domaine, LISP [BRO05].

Néanmoins, cette modélisation ne permet ni de décrire la temporalité des informations contextuelles, ni les relations qui peuvent exister entre ces informations contextuelles. Cependant, cette modélisation s'avère être très efficace pour raisonner sur le contexte et en déduire des actions à réaliser face à un contexte pertinent. Ainsi, cette approche peut être utilisée dans le cadre d'applications sensibles au

contexte afin d'intégrer et d'inférer sur les données contextuelles acquises. Malheureusement, il est difficile de décrire des contextes avec cette approche, ce qui la rend inapte à la modélisation du contexte.

2.3.4 L'approche orientée ontologie

Le terme « Ontologie » désigne à l'origine une branche de la Philosophie portant sur l'étude de l'Être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe. En informatique, les ontologies traitent de la représentation des connaissances d'un domaine particulier à travers l'utilisation de concepts et de rôles. L'intérêt de l'utilisation d'ontologies par des programmes et/ou des agents logiciels est qu'elles permettent de partager la compréhension d'un domaine afin de pouvoir obtenir une interprétation sans équivoque ou ambiguïté de la signification des ressources présentes sur le web.

Le W3C a créé en novembre 2001 un groupe de travail (le Web-Ontology Working Group) qui a abouti à la recommandation OWL (*Ontology Web Language*) en février 2004 [W3C04b]. OWL définit une syntaxe pour décrire et construire des vocabulaires afin de créer des ontologies. OWL est un langage basé sur RDF (Resource Description Framework) [W3C04a], il est défini en trois sous-langages de plus en plus expressifs, chacun étant une extension du précédent :

- * OWL-Lite est le sous-langage le plus simple. Il définit une hiérarchie de classification et fournit la possibilité d'ajouter des contraintes simples.
- * OWL-DL est fondé sur la logique descriptive permettant une expressivité maximum sans pour autant perdre la complétude du calcul (toutes les inférences sont assurées d'être prises en compte) et la décidabilité (tous les calculs seront terminés dans un intervalle de temps fini) des systèmes de raisonnement.
- * OWL-Full est la version la plus complexe de OWL, mais également celle qui permet le plus haut niveau d'expressivité. Toutefois, les problèmes liés à la complexité de calcul en un temps acceptable le rendent pour le moment inutilisable.

Les ontologies sont caractérisées par la possibilité de partager des connaissances entre plusieurs systèmes. De plus, elles sont ouvertes et extensibles, ce qui permet à chaque système ou utilisateur de les enrichir et d'exploiter les notions qui y sont déjà définies. En effet, il est possible d'étendre des ontologies existantes et d'employer diverses ontologies existantes pour compléter ou créer une nouvelle ontologie. À chaque ontologie, on peut associer un moteur d'inférence qui permet de raisonner sur les informations contextuelles en exécutant des règles d'inférence prédéfinies.

Un nombre conséquent de travaux s'appuie aujourd'hui sur des ontologies pour modéliser le contexte. Wang [WAN04] présente une représentation du contexte et une méthode de raisonnement sur les données contextuelles en se basant sur une ontologie. Une approche à base d'ontologie pour l'informatique diffuse est présentée par Euzenat [EUZ06] dans le cadre de la fourniture de Services Web. Parmi les travaux les plus aboutis, on peut citer CoBrA-ONT (*Context Broker Architecture ONTology*) [CHE04a]. CoBrA-ONT est une ontologie écrite en OWL-DL permettant la création d'applications multi-agents sensibles au contexte. Cette ontologie a pour objectif de décrire le contexte de pièces intelligentes situées dans un habitat communicant [CHE04b]. CONON (*CONtext ONtology*) est une ontologie de contexte écrite en OWL-DL dont l'originalité est l'extensibilité. Elle permet de décrire le contexte auquel des services peuvent être sensibles dans un environnement *pervasif* [WAN04].

Ces deux approches utilisent les ontologies comme un moyen de représenter le contexte et de classifier les informations contextuelles acquises par le biais de capteurs ou d'agents logiciels. Néanmoins, on peut remarquer que les ontologies décrites ici sont étroitement liées au domaine d'application. En effet, il n'est pas simple de trouver un modèle générique de représentation de tous les domaines. Comme le précise

Kokar [KOK06], il est préférable de construire sa propre ontologie, spécifique à son domaine d'application, et ensuite de trouver un moyen de fusionner ces ontologies (appelé aussi alignement d'ontologies) [EUZ05].

Cette approche exploite les avantages des ontologies pour modéliser le contexte de manière rigoureuse et complexe. En effet, il est possible d'établir des méta-modèles complexes de description du contexte permettant de décrire les relations entre données contextuelles. De plus, les ontologies permettent la gestion native de moteurs d'inférence afin de déduire des contextes de plus haut niveau à partir des données collectées.

2.3.5 Comparatifs des approches

Nous venons de présenter quatre approches permettant de représenter le contexte. Le Tableau 1 établit un comparatif entre ces approches. Ce comparatif est basé sur les caractéristiques indispensables que doit avoir un système sensible au contexte : prise en compte de la temporalité, prise en compte de l'imperfection des données, prise en compte de l'hétérogénéité des données, description des relations entre informations contextuelles. De même, un point important est la possibilité de raisonner sur le modèle de contexte.

Approche de modélisation de contexte	Gestion de la temporalité	Gestion de l'imperfection des données	Gestion de l'hétérogénéité des données	Description des relations	Gestion du raisonnement sur le contexte
Approche paires/triplets	Non	Non	Non	Non	Non
Approche orientée modèles	Non	Non	Non	Oui	Oui
Approche basée sur la logique	Non	Non	Non	Non	Oui
Approche orientée ontologie	Non	Non	Oui	Oui	Oui

Tableau 1 : Tableau comparatif des différentes approches de modélisation de contexte.

Ainsi, l'approche répondant aux cinq caractéristiques indispensables que doit avoir un système sensible au contexte semble être l'approche par ontologie. Les ontologies sont de plus en plus utilisées dans le cadre de l'Ingénierie des Connaissances, en particulier le Web Sémantique, car elles ont pour objectif de supporter la gestion des connaissances et le raisonnement sur ces connaissances. Nous choisissons une ontologie pour décrire le contexte de l'habitat communicant. En effet, cette ontologie fournit la représentation physique de l'habitat (porte, mur, fenêtre,...) ainsi que les objets composants l'habitat (meuble, électroménager,...). De même, cette ontologie permet de représenter l'habitant en modélisant ses caractéristiques ainsi que ses préférences (localisation, identification, activités,...). Cette ontologie est complétée par les informations contextuelles provenant des capteurs ce qui permet d'obtenir une représentation de l'individu et de son environnement à un instant t . On dit alors que l'ontologie est peuplée.

En choisissant une approche à base d'ontologie, nous obtenons une représentation formelle du contexte ainsi que des méthodes performantes de raisonnement sur les données contextuelles. La section suivante présente les outils permettant de réaliser un raisonnement à partir d'une ontologie.

2.4 La prise de décision : un problème d'inférence contextuelle

2.4.1 Présentation

Le contexte peut devenir rapidement très vaste. En effet, chacune des données contextuelles peut avoir un rôle important sur la prise de décision finale et le service rendu à l'habitant. Afin de prendre une décision face à un contexte, il est primordial d'observer une phase d'inférence sur les données acquises. Cette phase est basée sur une base de règles appliquées à une base de connaissance. Grâce aux règles, il est possible de conclure à propos du service le plus adapté à fournir face à une situation.

En modélisation de la connaissance, on différencie deux types de règles : les règles à chaînage avant et les règles à chaînage arrière. Le langage OWL n'utilise que les règles à chaînage arrière et ne permet pas de traduire celles à chaînage avant. La différence entre ces deux types est que pour les règles à chaînage arrière on se place du point de vue des classes de l'ontologie et non des individus, ce qui fait que l'on raisonne sur des ensembles plutôt que sur des éléments. Dans le cas des règles à chaînage avant, on se place du point de vue des individus. Les règles sont de la forme « *Corps* => *Tête* », où le corps de la règle est un ensemble de conditions à remplir (« $A \wedge B \wedge C \wedge \dots$ ») pour que la règle soit appliquée tandis que la tête représente l'information qui doit être déduite par le raisonneur si la règle est appliquée. Ce type de règle est donc beaucoup plus intuitif car il correspond à des règles facilement formulable en langage naturel : « SI telles ET telles conditions sont remplies, ALORS on en déduit telle information ».

Des solutions ont été proposées pour ajouter des règles à chaînage avant dans une ontologie modélisée en OWL. Certains raisonneurs ou certaines plateformes de manipulation d'ontologies offrent la possibilité de charger en mémoire une ontologie modélisée en OWL et d'y ajouter ensuite les règles à chaînage avant modélisées dans un langage propriétaire spécifique à l'application. C'est le cas par exemple des plateformes Jena 2 [JENA] ou Flora 2 [FLORA]. Néanmoins, cette approche est problématique puisque le modèle complet est alors spécifique à une application et n'est donc plus portable, la transcription des règles à chaînage avant devant être modifiée si l'on souhaite changer de raisonneur.

Un travail du W3C est en passe de devenir une recommandation. Celui-ci définit un langage de règles entièrement basé sur OWL : le Semantic Web Rule Language (SWRL) [HOR04]. S'inspirant du langage non standardisé RuleML, SWRL s'utilise très simplement lorsque l'on modélise en OWL puisqu'il suffit d'importer l'ontologie OWL correspondant à SWRL dans l'ontologie que l'on est en train de créer. Pour l'instant, le langage SWRL ne fournit que des règles obéissant à la logique du premier ordre (utilisation de connecteurs logiques ET (\wedge), IMPLIQUE (\rightarrow), NEGATION (\neg)). Le langage SWRL propose des extensions permettant d'utiliser des opérateurs plus évolués tel que des opérateurs de comparaison, des opérateurs mathématiques (addition, soustraction,...), la gestion des booléens, des chaînes de caractères, ... La liste complète est présentée dans [HOR04].

Très peu de travaux couplant SWRL et contexte existent. Nous pouvons citer les travaux de Viana [VIA07] qui propose une ontologie du contexte offrant une description détaillée des aspects informatiques disponibles. Cette description est associée à un ensemble de règles de raisonnement décrites en SWRL qui permettent de décider quels sont les services qui seront activés en fonction du contexte acquis. Le frein dans l'utilisation de SWRL est dû au fait que ce langage est récent et que peu d'outils permettent de le manipuler facilement. En effet, le langage SWRL est encore relativement jeune (proposé en mai 2004) et

les raisonneurs du marché nécessitent des modifications profondes pour prendre en charge le langage SWRL. Nous noterons cependant que de réels efforts sont faits dans ce sens à l'heure actuelle. Nous avons répertorié trois solutions :

- * Bossam Rule/OWL Reasoner [BOSSAM]
- * Pellet OWL Reasoner [PELLET]
- * KAON2 [KAON2]

Notre choix s'est porté sur KAON2 qui est un raisonneur gratuit. Il traite les ontologies en OWL et SWRL avec de très bonnes performances, mais a pour principal inconvénient de ne supporter ces deux langages que partiellement.

Malgré le peu d'outils existants (dont le nombre est croissant), l'approche basée sur SWRL permet de facilement définir un modèle auquel on ajoute un ensemble de règles autorisant le raisonnement sur l'ontologie peuplée. Une étude menée dans le cadre du projet A-MUSE [PLA06] présente les restrictions du langage SWRL dans le cadre d'applications sensibles au contexte. Le premier problème évoqué est l'impossibilité de prendre en compte la temporalité dans les informations contextuelles. Le second problème évoqué est l'impossibilité de prendre en compte naturellement la notion d'incertitude sur les données contextuelles. Le raisonnement effectué à partir des règles SWRL peut uniquement s'effectuer sur des données parfaites, non entachées d'incertitudes, ce qui peut fausser les décisions prises. Un travail d'ajout d'incertitude est en cours de réalisation [PAN05]. Cette initiative vise à intégrer la logique floue dans le développement des règles d'inférence. Ce travail en est à ses débuts et il n'apparaît à l'heure actuelle, aucun raisonneur capable d'intégrer cette nouvelle spécificité.

3 La fusion de données, l'incertain et le contexte

3.1 Présentation

Il existe différents niveaux d'inférence possibles [KOR00]. A partir de contextes simples tels que les données de température, les conditions d'illumination,... la logique du premier ordre peut déterminer le service le plus adapté à la situation perçue. Nous pouvons alors constater qu'il existe un fossé entre les données capteurs et l'ontologie. En effet, les données stockées dans l'ontologie ont une représentation symbolique contrairement aux données issues des capteurs. Le fossé entre les deux niveaux est généralement comblé par l'utilisation de méthodes de conversion symbolique.

Néanmoins, des contextes plus complexes peuvent se présenter [KOR03]. Généralement, ces contextes s'établissent dans le temps, sont définis par un ensemble d'indices et peuvent être issus de plusieurs sources de données. L'activité d'une personne est un contexte complexe qui se détermine par un ensemble d'actions, organisées temporellement.

Le projet d'INTEL [CHO06] portant sur le concept de l'homme invisible a pour objectif de déterminer les activités de la vie quotidienne (*ADL = Activities of Daily Living*) de personnes atteintes de démence. Ce projet s'appuie sur les habitudes des personnes en se basant sur l'utilisation qu'ils ont des objets de la maison dans laquelle ils vivent [CHO08]. Des Tags *RFID (Radio Frequency IDentification)* équipent certains objets de la maison. Les personnes sont équipées d'un bracelet (le *iBracelet*) contenant un lecteur de tag *RFID* qui permet de lire les Tags apposés sur les objets. En s'appuyant sur les habitudes des personnes [WU07] et sur des modèles temporels d'activités (représentés par des réseaux bayésiens), ils parviennent à déterminer les activités réalisées par les personnes ainsi que les activités inhabituelles.

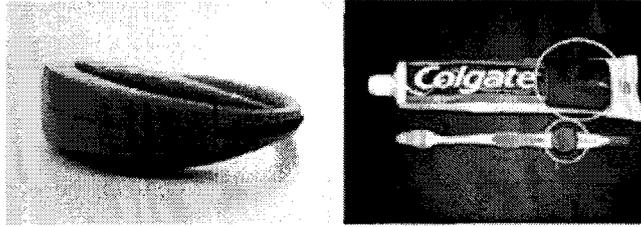


Figure 12 : Le *iBracelet* d'INTEL et une brosse à dent ainsi que le tube de dentifrice portant un tag *RFID*.

La logique du premier ordre ne permet pas de réaliser un tel niveau de raisonnement, c'est-à-dire prendre en compte l'évolution temporelle des actions effectuées par les individus. De même, cette approche est basée sur des données homogènes d'un niveau symbolique leur permettant d'être facilement manipulables et directement intégrables dans un moteur de raisonnement.

Dans le cadre d'applications « pervasive » et plus précisément dans le cadre de l'habitat communicant, il est aisé d'intégrer un nombre important de capteurs. En effet, les conditions d'installation et d'utilisation sont quasi-parfaites dans un environnement intérieur. Généralement, la construction d'une donnée contextuelle complexe est réalisée à partir d'un ensemble de données hétérogènes et d'un niveau sémantique non exploitable par l'ontologie [COU02].

Cependant, les capteurs peuvent utiliser différents principes physiques, peuvent couvrir différents espaces d'information, peuvent générer des données dans des formats différents, à différents taux d'actualisation. De même, les informations captées peuvent avoir des propriétés différentes en termes de résolution, de précisions et de fiabilité. Il est nécessaire de procéder à une conversion symbolique capable d'agréger cet ensemble d'informations hétérogènes au sein d'un même système. C'est ce que nous présentons dans la section suivante.

3.2 La conversion symbolique : Comment la réaliser à partir de données numériques ?

A l'origine, toutes les données perçues dans l'environnement sont de type analogique (capteur de température, capteur CCD, ...). Grâce aux convertisseurs Analogique/Numérique, il est possible de convertir ces données analogiques en données numériques, directement interprétables par un ordinateur. Néanmoins, il peut être nécessaire de donner davantage de sémantique à une donnée capteur. Par exemple, le relevé de la température ambiante d'une pièce donne 13°C. Cela correspond à la donnée symbolique 'Froid'.

Eric Benoit [BEN93] a introduit la notion de « capteur symbolique » permettant d'effectuer la conversion des données numériques en données symboliques directement exploitables par un moteur d'inférence symbolique. Ce principe est particulièrement bien adapté à notre problème. En effet, les ontologies utilisent des concepts symboliques. Il faut donc un mécanisme permettant de convertir une donnée numérique en une donnée symbolique le cas échéant.

Deux cas de conversion sont à préciser. Le premier correspond à l'utilisation d'une seule source de données dans le but d'obtenir une donnée contextuelle. Le second correspond à l'utilisation de plusieurs sources de données dans le but d'obtenir une donnée contextuelle complexe. La conversion symbolique devient alors un problème de fusion de données permettant d'agréger plusieurs sources de données afin d'obtenir une information symbolique affinée. Les méthodes de fusion de données permettent généralement de combiner un ensemble de sources de données et sont caractérisées par les modalités de

fusion. Ainsi il est possible de fusionner des données incertaines et/ou imprécises car celles-ci ne sont jamais parfaites.

L'exemple suivant tiré de l'article de Philippe Smets [SME99] définit et précise la différence entre ces deux termes que l'on confond fréquemment :

1. John a au moins deux enfants et j'en suis sûr.
2. John a trois enfants mais je n'en suis pas sûr.

Dans l'exemple 1, le nombre d'enfants est imprécis mais sûr. Dans l'exemple 2, le nombre d'enfants est précis mais incertain.

Le plus souvent, ce qui est le plus imprécis est le plus certain et ce qui est le plus précis est le plus incertain. La majorité des travaux traite de l'imprécision sans se soucier de la croyance envers cette donnée, c'est-à-dire suis-je certain que cette proposition est vraie? La fusion des données permet d'accroître la croyance que l'on peut avoir envers une proposition mais elle peut aussi accroître la précision d'une proposition.

3.3 *La fusion de données*

3.3.1 Présentation

Trop d'informations empêchent la compréhension du message que l'on désire faire passer. Il est alors indispensable de combiner les informations afin d'obtenir une information plus claire et plus précise. C'est le rôle de la fusion de données. Deux types de combinaison peuvent être réalisés dans les problèmes de fusion de données :

- * Une combinaison d'informations redondantes. L'intégration et la fusion de l'information redondante aide alors à accroître la précision et le degré de confiance d'une estimation obtenue avec plusieurs observations distinctes.
- * Une combinaison d'informations complémentaires. L'intégration et la fusion de données complémentaires permettent généralement de fournir une information ne pouvant être obtenue séparément avec chacun des capteurs.

La fusion de données est une thématique de recherche issue du domaine militaire dans le cadre de l'agrégation de données provenant de capteurs hétérogènes. Un modèle de fusion de données décrit les fonctions d'une application de fusion de données. La fusion de données collecte et synthétise les informations acquises à partir d'un ensemble de capteurs grâce à trois fonctions principales que sont :

- * L'alignement qui consiste à normaliser les informations dans un même référentiel. Dans le cas de données capteurs, il s'agit de transformer ces données pour exprimer les mesures dans un repère identique.
- * L'association qui consiste à affecter une observation capteur à un objet. Ces objets sont appelés « piste ». Ces pistes correspondent la plupart du temps à des ennemis (avions, chars,...) [MUS05] [BAR88]. Cette fonction est réalisée à partir de trois sous-fonctions :
 - La génération d'hypothèses permettant de définir des hypothèses possibles d'associations.
 - L'évaluation d'hypothèses détermine la confiance, d'ordre probabiliste, des associations réalisées.

- La sélection d'hypothèses basée sur la confiance maximale, qui permet de déterminer les associations à garder et à rejeter.
- L'estimation d'état réalise une combinaison (appelée aussi filtrage) des informations associées, permettant d'estimer de manière plus précise et plus sûre, la position d'une piste ainsi que son identité.

Pour réaliser une fusion de données, trois méthodes permettent d'agencer les fonctions précédentes [HAL01]:

- la fusion au niveau du signal qui combine directement les mesures de chaque capteur,
- la fusion au niveau des attributs qui combine des caractéristiques extraites par chaque capteur,
- la fusion au niveau des décisions qui combine l'information après que chaque capteur ait déterminé une localisation, une identité partielle des objets présents.

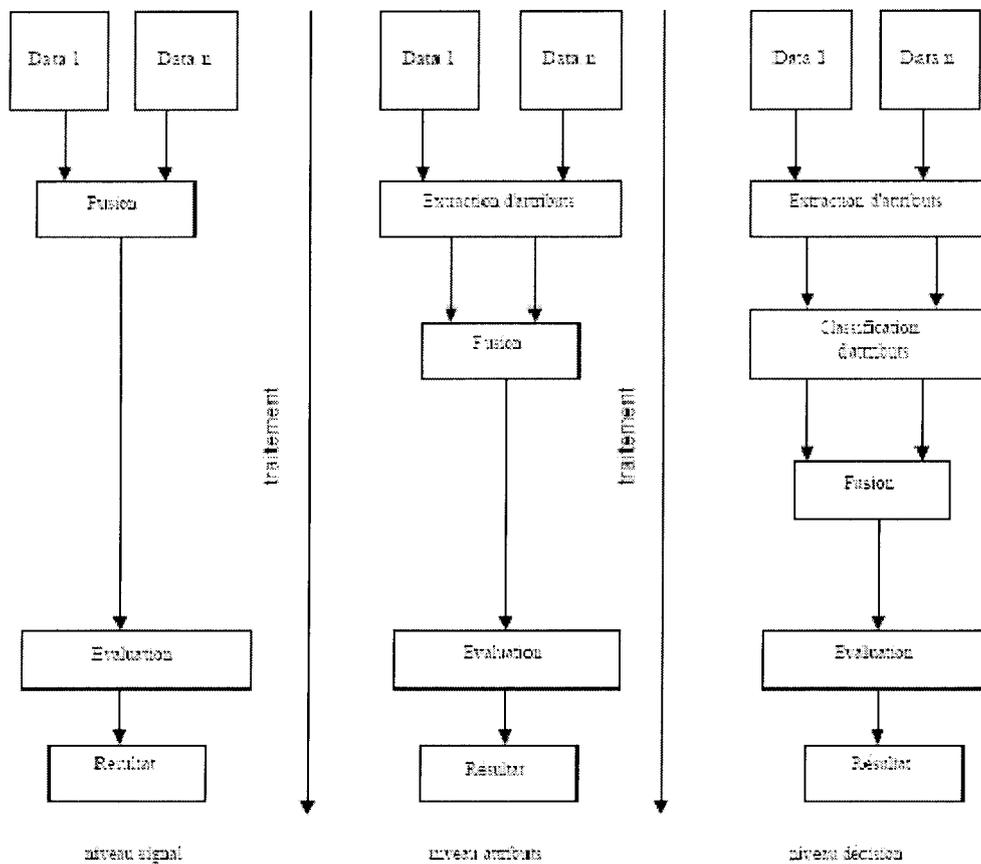


Figure 13 : Les différents niveaux de fusion [AUD04].

La Figure 13 présente les trois types de fusion possibles. La fusion la plus simple à réaliser est la fusion au niveau du signal car elle concerne les données physiques mesurées. Au niveau des attributs, ce sont des éléments caractéristiques de l'environnement qui sont extraits des données acquises en utilisant des prétraitements qui sont ensuite fusionnés. Au niveau de la décision, les entrées sont traitées individuellement pour extraire des informations de haut niveau sémantique puis sont fusionnées. Les décisions sont recombinaées en appliquant des règles de décision pour renforcer l'interprétation obtenue. Cette fusion est utilisée en particulier pour l'agrégation d'opinions d'expert.

Dans le cadre des applications sensibles au contexte, la définition que nous avons choisie est celle proposée par Wald [WAL99] :

« La fusion de données constitue un cadre formel dans lequel s'expriment les moyens et techniques permettant l'alliance des données provenant de sources diverses. Elle vise à l'obtention d'information de plus grande qualité. La définition exacte de 'plus grande qualité' dépendra de l'application ». Lucien Wald.

Cette définition, beaucoup plus large que celle spécifiée par le JDL (Joint Directors of Laboratories, définition proposée par le Department of Defense of the United States of America) [USD91], correspond mieux aux applications sensibles au contexte. En effet, les caractéristiques à déterminer sont plus nombreuses et très hétérogènes.

Janez [JAN96] présente cinq principales approches permettant de réaliser la fusion de données :

- Les méthodes probabilistes,
- La logique floue et la théorie des possibilités,
- La théorie des fonctions de croyance,
- Les méthodes non paramétriques,
- Les réseaux de neurones.

3.3.2 Les méthodes probabilistes

Basée sur des fondements mathématiques solides, la théorie des probabilités est la théorie la plus ancienne et reste la théorie la plus exploitée en fusion de données. Cette théorie a été introduite par Thomas Bayes en 1763 [BAY1763] et s'appuie sur le théorème des probabilités conditionnelles pour calculer la probabilité de réalisation d'une classe sachant qu'une observation est réalisée. Si chaque observation appartient à une classe et à une seule (la suite des classes est complète), alors on peut bâtir un classifieur de Bayes en s'appuyant sur une simple règle de décision sur un maximum de probabilité d'appartenance.

Dans le cadre de la fusion de données, la combinaison de données est réalisée à l'aide de la règle de Bayes, qui nécessite la connaissance de la probabilité a priori, définies sur un espace commun. La fusion peut être réalisée à l'aide de la règle de Bayes de deux façons :

1. au niveau de la modélisation à l'aide de la règle de Bayes,
2. par la règle de Bayes directement, où l'information issue d'une source vient mettre à jour l'information estimée d'après les sources précédentes.

Une seconde approche permettant d'effectuer une fusion au sens probabiliste revient à considérer l'existence d'une probabilité par source. La fusion consiste alors à combiner ces probabilités pour trouver la probabilité globale. Cette approche revient à considérer la fusion comme un problème d'estimation, et permet d'utiliser des opérateurs de combinaison différents du produit (méthodes de moyenne ou moyenne pondérée, de médiane, de consensus,...)

Une fois la fusion réalisée, la dernière étape concerne la décision. Cette décision binaire peut être assortie d'une mesure de la qualité de cette décision, pouvant éventuellement conduire à la rejeter. La règle la plus utilisée pour la décision probabiliste et bayésienne est le maximum a posteriori. Il existe de nombreuses méthodes de décision dépendantes du contexte d'utilisation : maximum de vraisemblance, maximum d'entropie, marginale maximale, espérance maximale, risque minimal,...

Il est à noter que la théorie des probabilités est utilisée dans le cadre de l'estimation. Dans cette catégorie, nous pouvons noter l'existence de divers filtrages ayant pour objectif le suivi. Le filtre de Kalman [NGU01] permet le traitement des données imparfaites, dans le cas où l'imprécision peut être modélisée sous forme d'un bruit aléatoire de distribution connue. Plus récemment des techniques de filtrage non-linéaire basées sur des techniques de Monte Carlo ont été développées. Le filtrage particulaire [NAK07] est un filtre non-linéaire fréquemment utilisé.

Dans le cadre des applications sensibles au contexte, Sekkas [SEK07] propose une approche de fusion de données basée sur des réseaux bayésiens dynamiques. Les réseaux bayésiens dynamiques sont la fusion de la théorie des graphes et de la théorie des probabilités. Ils permettent de modéliser l'évolution temporelle des contextes complexes. Cette approche permet de fusionner différentes sources de données dans le but de localiser des individus en environnement intérieur. Les auteurs présentent un exemple où sont utilisées deux sources de données : une localisation par Wifi et une localisation par IR.

Afin de s'adapter à l'environnement des utilisateurs et de leur proposer les services les plus adaptés, Nurmi [NUR06] propose une architecture de reconnaissance et d'apprentissage de contexte dans un environnement mobile. Pour cela, leur implémentation est basée sur des règles du premier ordre ainsi que sur des classificateurs de Bayes naïf. Les classificateurs de Bayes naïf reposent sur la formule générale de fusion Bayésienne.

Dans le cadre de la localisation, Hightower [HIG02] propose une architecture générique permettant la fusion de données de localisation. Cette architecture est divisée en sept couches dont une couche dédiée à la fusion de données de localisation. La fusion est réalisée sur des données imprécises grâce à la règle de fusion Bayésienne.

Il existe une grande quantité d'outils permettant de modéliser les distributions de probabilités et des outils permettant l'apprentissage de ces modèles. Cette théorie est la plus usitée dans le domaine de la fusion de données car elle permet une modélisation aisée de l'imprécision sur les données et offre une combinaison à l'aide de la règle de Bayes. Néanmoins, cette théorie souffre de lacunes qui ne lui permettent pas d'être utilisée dans tous les problèmes de fusion de données. En effet, le calcul des probabilités a priori est problématique dans le cas où la connaissance sur les sources est faible. De même, dans le cadre d'une utilisation multi-sources, la théorie des probabilités ne prend pas en compte naturellement le conflit entre les sources et ne modélise pas la méconnaissance envers les données, ce qui peut donner des résultats aberrants lors de leur fusion. Elle ne permet donc pas de représenter l'ignorance totale qui est remplacée par l'équiprobabilité.

3.3.3 La logique floue et la théorie des possibilités

La théorie des possibilités [DUB80], basée sur la théorie des ensembles flous, a été introduite par Dubois et Prade en 1980. Cette théorie permet de représenter l'imprécision à l'aide de distributions de possibilités et des fonctions de possibilité et de nécessité qui caractérisent une information. L'approche possibiliste peut être vue comme un enrichissement de la représentation logique des connaissances : plutôt que de considérer un état de croyance comme un ensemble de situations exclusives, on considère certaines situations comme plus ou moins possibles que d'autres. Cette théorie ne modélise pas un degré de vérité mais plutôt la préférence que l'on a pour une hypothèse. Cette théorie autorise une représentation plus souple que l'approche probabiliste. L'information qui peut être représentée à l'aide de la théorie des possibilités est cependant couverte aussi par la théorie des fonctions de croyances qui peuvent représenter un degré de vérité plus ou moins incertain.

Dans le domaine des applications sensibles au contexte, il existe très peu de travaux utilisant la théorie des possibilités. Anagnostopoulos [AGN07] propose une architecture de fusion de données contextuelles basée sur la théorie des possibilités. Cette architecture permet d'intégrer et de fusionner diverses sources d'informations : un PDA équipé de capteurs d'illumination, de bruits, de localisation, d'un détecteur de saisie et d'un accéléromètre trois axes. De même, le profil de l'utilisateur ainsi que les caractéristiques du bâtiment sont intégrées.

La théorie des possibilités permet une modélisation aisée de la connaissance à l'aide de fonctions de distributions. Elle permet une modélisation efficace de la méconnaissance envers des hypothèses. Cette théorie est fortement utilisée dans le domaine de la fusion de données car il est aisé de mettre en œuvre des distributions de possibilités trapézoïdales ou triangulaires. Néanmoins, la théorie des possibilités ne gère pas nativement le conflit entre sources mais une gestion peut être réalisée par des règles adaptatives. De même, elle ne permet pas une modélisation de l'ignorance que l'on peut avoir envers des hypothèses.

3.3.4 Les méthodes non paramétriques

Les méthodes non paramétriques sont des méthodes d'estimation classiques comme les k plus proches voisins [MAU04] ou les C-moyennes supervisées [BEZ81]. Elles sont surtout utilisées dans les problèmes de reconnaissance et de classification. Janez précise que ces méthodes sont simples sur des problèmes complexes mais sont coûteuses en temps de calculs, ce qui les rend difficilement utilisables. De plus, ces théories ne permettent pas une manipulation aisée de l'incertitude sur les données, c'est pourquoi nous ne les développerons pas davantage dans cette section.

3.3.5 Les réseaux neuronaux

Les réseaux de neurones [MCC43] permettent d'approcher directement les probabilités a posteriori d'appartenance aux classes. Les sorties de réseau représentent l'étiquetage en classe des exemples de la base. L'inconvénient de ces méthodes est le besoin d'une importante banque d'exemples pour l'apprentissage, ce qui les rend difficiles à mettre en œuvre lorsque peu de connaissance a priori existe.

3.3.6 La théorie de Dempster-Shafer (ou théorie de l'évidence ou théorie des fonctions de croyance)

La théorie de Dempster-Shafer a été introduite par Shafer en 1976 [SHA76] en se basant sur les travaux de Dempster sur les intervalles de probabilité [DEM67]. Une variante de cette théorie a été proposée par Smets en 1990 [SME90] sous le nom de Modèle des Croyances Transférables (MCT ou TBM : Transferable Belief Model) en fournissant une base axiomatique à cette théorie. La théorie de Dempster-Shafer permet la modélisation de l'incertitude associée au fonctionnement d'une source de données et de sa capacité de discernement entre plusieurs hypothèses. De même, cette théorie permet naturellement d'exprimer le conflit entre différentes sources fusionnées. La théorie de Dempster-Shafer offre un cadre fédérateur des théories de l'incertain puisque qu'elle est la généralisation de la théorie des probabilités et de la théorie des possibilités. Cette théorie s'exprime sous la forme d'intervalles de probabilité et permet de quantifier la vraisemblance des différentes hypothèses ainsi que d'exprimer un degré de confiance par l'écart entre la borne supérieure et la borne inférieure des mesures de probabilité. Contrairement à la théorie des probabilités, la théorie de Dempster-Shafer permet de modéliser des événements non exclusifs et offre la possibilité de modéliser la méconnaissance envers des hypothèses.

La théorie de Dempster-Shafer a été appliquée dans de nombreux domaines. Nedelec [NED03] l'utilise dans le cadre de l'étude de la déforestation au Brésil. Trois paramètres sont ainsi fusionnés : des cartes d'espaces protégés, des cartes d'espaces défrichés définis à partir d'une classification d'images satellites et

un modèle de défrichement réalisé à partir d'observations d'images satellites. Ainsi, ces trois entrées permettent un suivi de l'évolution de la déforestation et sont utilisées dans des modèles à des fins prédictives qui permettent l'élaboration de scénarii à court terme.

Dans le domaine de la robotique mobile, Clémentin [CLE02] propose une approche multi-critères de construction de carte et de localisation simultanée (SLAM : Simultaneous Localisation And Mapping). Il s'appuie sur la théorie de Dempster-Shafer pour fusionner les informations provenant de deux capteurs extéroceptifs. Cette approche multi-niveaux permet de modéliser l'incertitude envers les attributs reconnus dans l'environnement, la carte globale créée par fusion des attributs ainsi que sur la position du robot mobile.

Néanmoins, dans le domaine des applications sensibles au contexte, peu de travaux existent à l'heure actuelle. On peut noter Huadong Wu [WU02] [WU03], pionnier en la matière, qui propose une architecture générale pour les applications sensibles au contexte intégrant la notion de fusion et d'incertitude sur les données contextuelles. Il s'appuie sur la théorie de Dempster-Shafer afin de modéliser et fusionner les données contextuelles. Cela constitue une des rares approches basées sur la théorie de Dempster-Shafer dans le domaine des applications sensibles au contexte.

En localisation intérieure, Saaman [SAM05] utilise la théorie de Dempster-Shafer afin de prédire les localisations futures à partir du comportement d'un individu. Dans le cadre de la mobilité des voyageurs, l'objectif de ces travaux est la prédiction avec précision de la trajectoire et de la destination de voyageurs en utilisant des connaissances a priori : les préférences de l'utilisateur, ses buts ainsi que l'historique de ses déplacements.

ALARM-NET [WOO06] propose un réseau de capteurs sans-fil destiné à l'habitat dans le cadre du maintien à domicile. A partir de capteurs environnementaux et physiologiques, une association de données est réalisée grâce à la théorie de Dempster-Shafer. Cette théorie permet d'associer les données recueillies avec la bonne personne, dans le cas d'une utilisation multi-personnes. Ainsi, la théorie de Dempster-Shafer permet d'effectuer la fusion de données de localisation, de données physiologiques ainsi que les activités détectées. Ainsi, il est possible de réaliser un rapport concis et précis au médecin chargé de la surveillance.

Dans le cadre de la fusion de données, la théorie de Dempster-Shafer propose une modélisation naturelle du conflit entre les sources fusionnées. A partir de l'analyse du conflit, il peut être possible par exemple d'identifier une source non fiable. Néanmoins, cette théorie souffre d'un problème d'explosion combinatoire dû à la prise en compte des événements non-exclusifs ce qui peut freiner l'utilisation de cette théorie dans le cadre d'application embarquée. Michèle Rombaut [ROM06] précise que la théorie de Dempster-Shafer permet de modéliser la connaissance dont on dispose sur l'objet observé alors que la théorie des probabilités modélise les chances que l'on a si l'on choisit l'une ou l'autre des hypothèses. Ainsi, la théorie de Dempster-Shafer est adaptée à la perception des phénomènes alors que la théorie des probabilités est plutôt adaptée pour le processus décisionnel. C'est ce que prouve Smets dans [SME05] où il permet le passage des fonctions de croyance en probabilités dans le cadre de la transformation pignistique.

Le constat que nous avons fait sur les modèles de contexte basés sur une ontologie est l'impossibilité de prendre en compte l'incertitude sur les données contextuelles. De même, il est impossible aux ontologies de raisonner sur des données incertaines, les données présentes dans l'ontologie étant considérées comme parfaites. Nous nous proposons ici de déporter la prise en compte de l'incertitude en amont de l'ontologie

afin de faciliter le travail de modélisation. Ainsi, la représentation de l'incertitude et la fusion de ces données incertaines sont effectuées avant le peuplement de l'ontologie.

Malgré une avancée théorique des probabilités sur la prise en compte de l'incertitude [MAS07], la théorie de Dempster-Shafer a été retenue car elle permet une prise en compte naturelle de l'incertitude sur les données contextuelles. De même, la modélisation du conflit est un atout essentiel dans la modélisation du contexte. En effet, à partir de l'étude du conflit il est possible de déterminer la fiabilité envers les capteurs retenus au sein de l'habitat. Cette notion de fiabilité permet, entre autre, de réaliser un diagnostic fin de l'état des capteurs au sein de l'habitat en vue d'une maintenance éventuelle et le remplacement de capteurs défectueux.

4 La défaillance capteur

4.1 Introduction

Dans les systèmes basés capteurs, le dysfonctionnement est un problème primordial. Des mesures erronées peuvent conduire à de faux jugements et donc à de fausses commandes. Afin d'éviter au système de mal fonctionner, il est judicieux de détecter l'apparition d'un symptôme, c'est-à-dire détecter un changement par rapport à un comportement normal, afin de prévenir les fausses mesures. Si le système est capable de détecter un comportement anormal d'un capteur, il est alors capable d'informer quel capteur est défectueux afin de procéder à sa réparation ou à son changement.

Différentes méthodes existent permettant de détecter des défauts majeurs tels que la panne complète d'un capteur (traduite par un saut du signal ou un changement du niveau de bruit fourni par le capteur). Par contre, les légères dérives sont difficilement détectables dans le cadre d'une analyse mono-capteur. Les approches multi-capteurs permettent de faire apparaître le défaut en détectant des changements dans les relations entre les mesures.

4.2 La tolérance aux pannes

La capacité d'un système à fonctionner en mode dégradé même en cas de dysfonctionnement s'appelle la tolérance aux pannes. La tolérance aux pannes (fault tolerance) est l'aptitude d'un système à demeurer fonctionnel malgré certaines pannes de ses composants. La gestion des fautes a pour but de permettre à l'application de fournir un service conforme aux désirs de l'utilisateur malgré la présence de fautes. La maintenance permet de maintenir ou de rétablir une installation capable d'assurer un service déterminé. Nous trouvons deux méthodes de maintenance : maintenances actives (maintenance préventive) ou passives (maintenance curative).

La différence entre ces deux méthodes de maintenance se trouve dans la nature de l'intervention. La maintenance préventive consiste à surveiller le système ou à procéder à des remplacements systématiques d'un ou plusieurs composants quel que soit leurs états. La maintenance curative s'appuie quant à elle sur le dépannage. Elle consiste à détecter le problème mais sans se soucier de la faute qui est à l'origine de cette panne.

Pour obtenir une maintenance de bonne qualité, il est primordial de superviser le système et principalement chacun des composants de ce système afin de détecter précisément quel est le composant fautif. La supervision consiste à observer un système en cours d'exécution et à utiliser ces observations pour pouvoir vérifier si son comportement satisfait ou ne satisfait pas un ensemble de propriétés. Les notions de monitoring et de supervision sont étroitement liées. En effet, la supervision relève de

l'information alors que le monitoring analyse ces informations afin de vérifier leurs validités puis agir en cas de faute.

Les différentes méthodes de maintenance sont critiquables dans le sens où certaines deviennent rapidement onéreuses. La maintenance préventive visant à changer périodiquement des composants ainsi que la maintenance curative entraînent un coût important et donnent un caractère bloquant à l'application lors du changement de composants. En effet, arrêter une application s'exécutant 24 heures sur 24 peut coûter très cher à une entreprise. Ici on bloque la production (dans le cadre d'une industrie), ce qui empêche le déroulement normal de l'application. De la même manière, au sein d'un habitat, il faudrait qu'un technicien vienne régulièrement remplacer des composants, ce qui est inimaginable et peu pratique pour les habitants.

La solution consiste alors à détecter de manière précoce les dysfonctionnements et à tenter de les réparer de manière totalement invisible pour l'utilisateur sans influencer sur le fonctionnement normal de l'application. C'est ce que l'on appelle la reconfiguration dynamique d'applications, c'est-à-dire que la reconfiguration de l'application s'effectue en cours d'exécution.

4.3 Détection de défaillance capteurs : Méthodes utilisées

Comme nous l'avons présenté précédemment, nous nous plaçons dans le contexte de la modélisation d'informations contextuelles. À partir d'une ontologie, nous réalisons une conversion numérique vers symbolique des données issues des capteurs. De nombreuses méthodes de diagnostic existent et sont souvent basées sur les données numériques provenant des capteurs. Par contre peu de travaux s'appliquent au diagnostic de données symboliques.

Dans la littérature, trois principales méthodes sont mises en œuvre pour la détection du dysfonctionnement d'une source de données :

1. Soit la source de données est parfaitement connue et les valeurs fournies sont bornées. Si les valeurs sortent des bornes, il est possible d'en déduire un dysfonctionnement [LI06]. Ces bornes sont issues d'une expertise humaine ou par apprentissage.
2. À partir d'un modèle défini par expertise humaine ou automatique par apprentissage, il est possible de réaliser une phase de prédiction-vérification. Cette phase permet de détecter un comportement anormal d'une seule source de données. Grâce à un filtre de prédiction, il est possible de prédire les données à l'instant suivant. Par vérification entre l'observation et la prédiction, il est alors possible de déterminer un comportement anormal. Un filtre crédibiliste a été utilisé dans [RAM06] pour la détection de changement d'état et pourrait être appliqué à la détection de dysfonctionnements. Le filtre de Kalman [JEA96] ou le filtrage particulaire [CAR06] sont de même utilisés pour déterminer des comportements anormaux.
3. Ici, l'expertise humaine n'est pas nécessaire. Le système s'appuie sur plusieurs sources de données qui produisent des données redondantes sur une même variable. Par comparaison des sources, il est possible de déterminer une anomalie entre deux sources voire déterminer précisément la source de données en faute lorsqu'il y a au minimum trois sources identiques.

Dans le cadre de l'utilisation d'informations contextuelles symboliques, l'utilisation de bornes (première proposition) est à proscrire. En effet, la détection de dépassement de bornes est à effectuer lors de la conversion numérique vers symbolique. Les deux autres approches peuvent être appliquées à notre cas.

Les deux approches sont à utiliser dans deux cas distincts. Les méthodes à base de prédiction-vérification sont à utiliser lorsque nous sommes en présence d'une seule source de données alors que les méthodes basées sur la redondance s'utilisent par définition lorsque nous sommes en présence de plusieurs sources de données.

5 Conclusion

La thématique de recherche portant sur la sensibilité au contexte est large. Elle intègre un nombre important de domaines de recherche, passant de l'Interaction Homme Machine au Web sémantique, en passant par l'Intelligence Artificielle. Dans le cadre de l'habitat communicant, nous avons pu voir que trois grands axes émergent.

Le premier axe porte sur la modélisation du contexte. La majeure partie des travaux actuels s'oriente vers une modélisation par ontologie. Les ontologies permettent une modélisation plus riche et plus complexe des informations contextuelles. Elles intègrent les notions d'héritage et d'associations qui sont pratiques pour décrire des contextes complexes. Dans le chapitre 2 de ce mémoire, nous présentons une proposition d'architecture de gestion contexte se basant sur un modèle d'ontologie pour l'habitat communicant.

Le second axe porte sur l'intégration de contextes incertains et imprécis. En effet, la majorité des travaux portant sur la sensibilité au contexte n'intègre pas ou peu ces notions fondamentales. Nous avons pu voir que les données issues des capteurs ne sont pas parfaites. Elles sont imprécises et pas toujours certaines. Il est nécessaire d'intégrer ces notions afin de ne pas rendre un mauvais service à l'habitant. Les modèles à base d'ontologies ne permettant pas de prendre en compte nativement l'incertain et l'imprécision, il faut déporter ces traitements en amont de l'ontologie grâce à des conversions numériques vers symboliques grâce à des formalismes adaptés. Nous avons vu que le formalisme de Dempster-Shafer se prêtait bien à la prise en compte de données incertaines et imprécises.

Le troisième axe est la prise en compte de la défaillance potentielle des capteurs. Les capteurs peuvent subir des défaillances au cours de leur fonctionnement. Ainsi, il est possible d'effectuer une maintenance préventive en observant le comportement des capteurs soit par redondance d'informations, soit par vérification de la prédiction du comportement estimé à partir d'un modèle. Ainsi il est possible de procéder au changement du capteur défaillant ou de passer dans un mode dégradé permettant de fournir un service lui-même dégradé. Ces deux derniers grands axes sont développés dans les chapitres 3 et 4 de ce mémoire.

Nous nous proposons d'exploiter ces trois axes dans nos travaux en les intégrant au sein d'une même architecture de sensibilité au contexte dans le cadre de l'habitat communicant.

*'The Future Is Not What It Used
To Be'
Quentin Hardy*

Chapitre 2

La prise en compte du contexte dans l'habitat communicant

1 Introduction

Dans l'état de l'art, nous avons dans un premier temps présenté l'importance de la prise en compte du contexte au sein de l'habitat communicant. Dans le cadre de cette problématique, l'observation du contexte permet de rendre le service le plus adapté à l'habitant. A partir des différentes couches intervenant dans la gestion du contexte que nous avons présentées, nous proposons dans ce second chapitre une architecture logicielle de gestion du contexte pour l'habitat communicant. L'objectif est de fournir une architecture complète basée sur des standards logiciels proposant des outils pour la construction d'applications sensibles au contexte dans l'habitat.

Nous nous sommes fixés deux principales contraintes lors de l'étape de réflexion sur l'élaboration d'une telle architecture. La première concerne la dynamique des services proposés. En effet, matériels et logiciels peuvent évoluer dynamiquement, par choix personnel de l'habitant, par choix technique ou du fait du fournisseur de service. Notre seconde contrainte est la prise en compte de l'hétérogénéité du matériel utilisé. Les matériels ayant des caractéristiques différentes, l'architecture doit être capable de s'adapter. L'architecture doit permettre de fournir un service en fonction du contexte perçu dans l'habitat : elle doit pouvoir observer ce qui se déroule au sein de l'habitat, et fournir le service le plus adapté à l'habitant, selon le contexte observé à un instant t . Une telle architecture doit enfin permettre une mise à jour aisée à distance. En effet, le fournisseur de service doit pouvoir faire évoluer l'architecture ou une partie de celle-ci sans avoir à se déplacer chez l'habitant.

Dans un premier temps, nous présentons le modèle multicouche permettant de rendre un service à l'habitant en fonction d'un contexte. Afin de rendre cette architecture adaptative, nous présentons les différents aspects logiciels autorisant une gestion dynamique de l'application. Nous détaillons ensuite les différentes briques logicielles de base permettant de mettre en œuvre notre modèle multicouche [RIC06a]. Finalement, nous présentons un cas d'application de notre architecture et concluons sur les performances de notre approche et sur la pertinence des choix retenus [RIC07a].

2 Le modèle PCIA (Perception-Contexte-Inférence-Action)

Les objectifs de notre architecture sont multiples. Nous avons identifié quatre principales parties afin de fournir le service le plus adapté à l'habitant. Dans un premier temps, il s'agit d'acquérir un ensemble de données hétérogènes disséminées dans l'habitat. Dans un deuxième temps, nous devons interpréter et modéliser ces informations, extraites de l'habitat. Cette modélisation nous permet de représenter un contexte à un instant t . Ensuite, nous devons sélectionner le service le plus adapté à l'habitant, grâce à une prise de décision faite à partir de la modélisation d'un contexte particulier. Finalement, nous devons mettre en œuvre les moyens pour offrir le service à l'habitant.

Nous avons défini le modèle *PCIA* pour l'habitat communicant, synthétisant ces quatre parties :

P : *Perception* – La couche perception permet d'obtenir une vue numérique des informations collectées dans l'habitat. Elle donne une représentation des phénomènes physiques ou des connaissances au sein de l'habitat. Cette couche possède des informations courantes qui peuvent varier au cours du temps.

C : *Contexte* – La couche Contexte a pour rôle de définir le nouvel état actuel de l'habitat à travers les informations perçues et interprétées. Nous les nommons « données contextuelles ». Cette couche évaluant l'état courant, elle autorise l'anticipation du *futur immédiat* à partir d'un modèle prédictif.

I : *Inférence* – A partir de l'état courant défini par la couche Contexte, la couche Inférence répertorie un ensemble de schémas d'actions à délivrer. Cette couche permet de raisonner à partir des données

contextuelles afin de fournir le service le plus adapté à l'habitant. C'est dans cette couche que siège le module décisionnel central de l'architecture.

A : Action – La couche Action a pour rôle de fournir le service à l'habitant. Pour chacune des décisions prises, cette couche délivre une action élémentaire à exécuter, dans le but de modifier un phénomène physique ou logiciel de l'habitat.

Nous pouvons observer que les couches *perception* et *action* proposent des interfaces logicielles avec des objets de perception (*capteurs physiques*) et d'action (*commandes et actionneurs*). Le modèle PCIA que nous proposons ici est schématisé sur la Figure 14. Elle montre les interactions entre les différentes couches du modèle, développées dans la suite de ce chapitre.

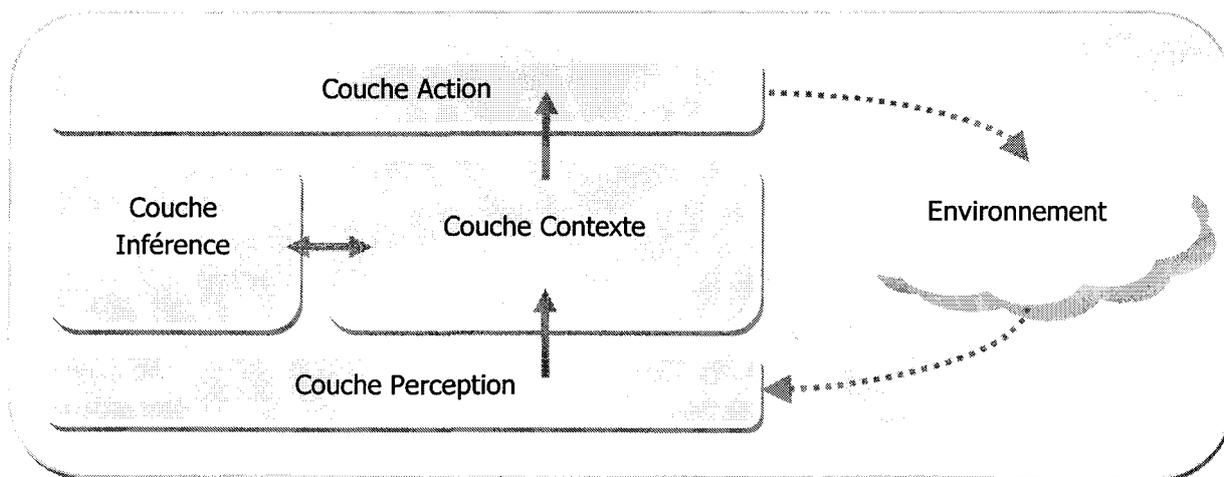


Figure 14 : Les couches du modèle PCIA et leurs interactions.

3 Architecture logicielle de gestion du contexte dans l'habitat communicant

3.1 Introduction

Une architecture de gestion de contexte dans l'habitat doit permettre au fournisseur de service d'offrir automatiquement des services selon le contexte ambiant dans lequel évoluent les habitants. Pour cela, une plateforme de déploiement de services doit être installée chez l'habitant, constituant ainsi une passerelle domestique vers l'extérieur. Son but est d'activer ou de désactiver des éléments logiciels (des services), indépendamment les uns des autres. Par exemple, il doit être possible d'installer un service supportant un nouveau type de capteur, sans que nous ayons à réinitialiser toute l'architecture. De même, lors de la mise à jour d'un service, les autres services doivent rester disponibles. La plateforme est donc en charge de la gestion du « cycle de vie » de tous les services qui y sont déployés.

Ce déploiement dynamique de services s'articule autour de deux approches logicielles. La première approche est la programmation orientée composant (POC) ; la seconde étant la programmation orientée service (POS). Toutes deux sont issues du domaine de la programmation orientée objet (POO), et offrent des possibilités intéressantes en matière d'architecture et de conception logicielle, que nous développons dans cette partie.

3.2 Approche par les composants

L'utilisation d'une approche par les composants favorise le découplage de ses différentes fonctionnalités. Cette méthodologie qui soutient en effet l'idée que la construction d'applications peut être réalisée à partir de l'assemblage d'entités logicielles réutilisables appelées *composants* [SZY98]. Un composant peut être assemblé avec d'autres composants, permettant ainsi la réutilisation de code même s'ils ne partagent pas le même contexte d'utilisation. Les fonctionnalités offertes par les composants sont définies grâce à des interfaces, permettant au concepteur d'applications leur intégration au sein d'une architecture, leur réutilisation, et d'en maîtriser les interactions.

Lorsque le modèle utilisé est dynamique, il est possible de reconfigurer l'application en cours d'exécution, en modifiant son architecture afin de l'adapter aux évolutions de son contexte d'exécution. Parmi les plateformes à composants les plus connues, nous pouvons citer EJB, CCM, COM/.NET et Fractal. Elles sont généralement destinées à des domaines d'applications spécifiques :

- Les EJB (*Enterprise Java Beans*) de SUN sont principalement utilisés dans le domaine du commerce électronique sur Internet et permettent d'effectuer des transactions sécurisées de manière très simple pour le programmeur.
- Le modèle de composant COM datant de 1994, devenu .NET depuis 2002, fournit une compatibilité entre diverses applications de natures hétérogènes conçues dans des langages de programmation différents. Ce modèle est principalement orienté vers des applications business ou industrielles telles que la gestion d'entreprise, bancaire ou de production.
- Le CCM (*Corba Component Model*) est une surcouche de l'architecture distribuée CORBA, issue de l'OMG⁹. Elle repose sur la définition de composants logiciels communiquant à travers le bus CORBA, et profite ainsi de ses fonctionnalités de base (transparence à la localisation, indépendance de la plateforme d'exécution). Selon l'implantation utilisée et les outils fournis, ce modèle devient malgré tout rapidement complexe à mettre en application.
- Fractal [DAV05] est un modèle de composants développé par France Télécom Recherche & Développement et distribué dans le cadre du consortium ObjectWeb¹⁰. C'est un modèle de composants dynamique et réflexif, permettant de construire des applications à partir d'un modèle à composants supportant une structure hiérarchique. Fractal est le seul modèle qui est suffisamment général pour être appliqué à tout domaine, mais reste encore cantonné au milieu académique.

3.3 Approche à service

Un service est un comportement défini de façon contractuelle et dont le contrat est rempli entre un client et un fournisseur de services. Dans l'approche orientée service, la disparition ou l'arrivée des services pendant l'exécution est prise en compte. Cette disponibilité dynamique permet de construire des applications pouvant s'adapter à diverses situations telles que la sensibilité au contexte.

Contrairement à l'approche par les composants, l'approche orientée service se focalise sur la description et l'organisation de services afin de supporter leur découverte dynamique au moment de l'exécution [BUR00].

⁹ L'OMG a pour objectif de standardiser et de promouvoir le modèle objet - <http://www.omg.org/>

¹⁰ <http://fractal.objectweb.org>

Notre choix s'est porté sur la plateforme de déploiement de services OSGi (*Open Service Gateway initiative*¹¹) car elle répond aux contraintes de dynamique. Cette approche a déjà été retenue lors de travaux antérieurs dans notre équipe [REN07]. L'expérience montre que la plateforme OSGi permet un téléversement et une mise à jour aisée des services déployés sur une passerelle résidentielle. Ce point répond à une contrainte que nous nous étions fixés, à savoir la possibilité de mettre à jour des services chez l'habitant sans nécessairement se déplacer chez lui. OSGi s'appuie sur Java, ce qui lui permet d'être mise en place sur tout système. OSGi est principalement destiné à être mis en place sur des passerelles embarquées telles que les passerelles domestiques vues précédemment.

OSGi offre un environnement standard pour les applications (*appelées Bundles*) et fournit différents services tels que la gestion du cycle de vie de l'application (installation, désinstallation, démarrage, arrêt et actualisation) de façon dynamique sans avoir besoin de procéder à un redémarrage complet de la passerelle. Un Bundle est constitué d'une archive JAR, contenant un fichier « *manifest* » et une combinaison de classes Java, de bibliothèques en code natif, de fichiers JAR embarqués et de ressources. Sous OSGI, un service est composé d'un ou d'un ensemble de Bundles. Un ensemble de services Java¹² sont définis, portant sur le système lui-même, sur la plateforme, sur les protocoles et divers services.

L'objectif principal de la plateforme de services OSGi est de profiter des capacités d'indépendance de plateforme et de chargement de code dynamique du langage Java pour faciliter le développement et le déploiement dynamique d'applications pour des systèmes embarqués.

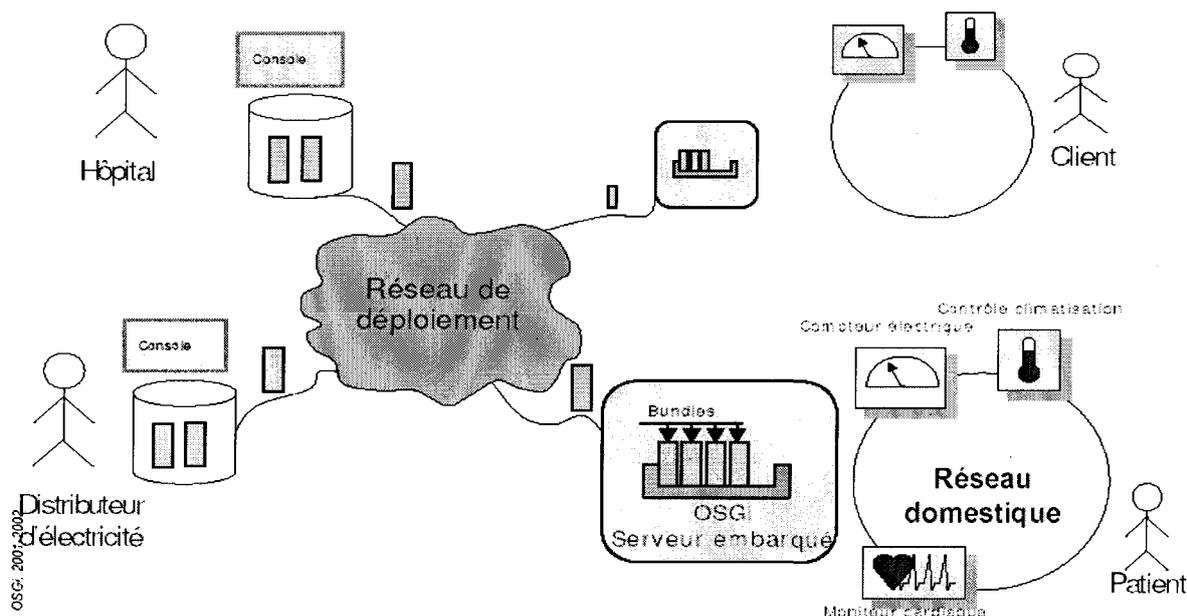


Figure 15 : Exemple de déploiement de Bundles OSGI (source Didier Donsez).

Sur la Figure 15, un exemple de déploiement de Bundles est présenté. Le médecin de l'hôpital désire connaître l'état de santé de son patient. Pour cela, un Bundle est déployé, permettant le relevé des données

¹¹ <http://www.osgi.org/>

¹² http://www.osgi.org/osgi_technology/

physiologiques de son patient (ici des données cardiaques). Il transmet ensuite, via le réseau, les données qui seront affichées sur la console du médecin. De même, le distributeur d'électricité a la possibilité de relever à distance la consommation électrique de ce même patient. Pour cela, il déploie un Bundle de lecture du compteur électrique et d'émission des données vers l'opérateur électrique. Ainsi, plusieurs Bundles peuvent coexister sans interférer mutuellement sur leur comportement. Dans le cadre de l'habitat communicant, le serveur embarqué sera vraisemblablement une passerelle domestique.

Ainsi, même si l'approche par les composants offre une souplesse de mise en œuvre, notre choix s'est porté sur l'approche à service et en particulier la plateforme de déploiement de services OSGi. L'approche à service se distingue de l'approche par composant par sa gestion du dynamisme et une gestion poussée du cycle de vie.

3.4 OSGi par rapport à JADE

Il existe quelques plateformes de développement logiciel. Parmi celles-ci, nous avons identifié une plateforme intéressante pour le développement d'applications nécessitant une couche d'inférence. Comme nous l'avons vu précédemment, cette couche est primordiale pour notre architecture. Il s'agit de la plateforme JADE¹³ (Java Agent DEvelopment Framework), utilisée dans certains travaux, comme substrat logiciel de développement d'applications sensibles au contexte [SUB07]. JADE est une plateforme multi-agents assez similaire à la plateforme OSGi mais davantage orientée intelligence artificielle. L'unité de déploiement n'est pas un Bundle mais un *agent communicant*. Cet agent peut rendre un ou plusieurs services qui ont été indexés par un agent « annuaire » déployé automatiquement au lancement de la plateforme. La particularité des agents communicants est qu'ils peuvent échanger des messages entre eux selon plusieurs protocoles de communication, et ce de manière relativement flexible.

Le principal atout de cette plateforme est sa capacité à communiquer avec d'autres plateformes JADE de manière totalement transparente pour les agents déployés. Ainsi, les agents peuvent communiquer entre eux sans avoir à se préoccuper de savoir s'ils s'exécutent sur la même plateforme ou sur des machines/plateformes différentes. De même, cette plateforme intègre une compatibilité forte avec certains moteurs d'inférence, ce qui permet d'ajouter facilement une composante de raisonnement aux agents communicants, satisfaisant la couche I de notre modèle. Ces avantages ont néanmoins un coût notable en termes de capacité de calcul et de mémoire requise, en comparaison avec OSGi.

Comme nous l'avons spécifié précédemment, l'expertise existante que nous avons d'OSGi nous pousse à conserver ce choix. En effet, l'intégration de JADE, qui paraît séduisante, impliquerait un nombre assez important de développements logiciels pour pallier le fossé entre la plateforme JADE et l'existant sous OSGi. Il est enfin utile de se poser la question de la nécessité de déployer un grand nombre d'agents dans l'habitat, alors que nous effectuons une prise de décision centralisée (couche I du modèle PCIA)

3.5 OSGi et le modèle PCIA

Notre objectif est de mettre en œuvre le modèle PCIA au sein d'une architecture de déploiement de services, et plus particulièrement OSGi. Nous présentons ici chacune des couches du modèle PCIA et leur intégration logicielle au sein d'une architecture à services.

¹³ <http://jade.tilab.com/>

3.5.1 La couche Perception

Nous avons défini précédemment que les objets de perception sont, dans le cas de l'habitat communicant, des capteurs. Il s'agit donc d'interfacer ces capteurs dans une approche à service. Actuellement, les capteurs que nous avons à disposition fournissent des données diverses, selon un protocole et un formalisme propres aux concepteurs du capteur. Il devient alors problématique de trouver un modèle générique adapté à l'ensemble des capteurs dont nous pouvons avoir besoin. Il nous faut considérer les différents protocoles envisageables et les adapter à nos besoins. Une des solutions envisagées consiste à utiliser un adaptateur permettant de « traduire » les données provenant des capteurs en données standardisées compréhensibles par notre architecture.

Dans ce sens, nous proposons une méta-description des capteurs et de leurs données afin de faciliter l'intégration de capteurs hétérogènes dans notre architecture.

Dans le monde des capteurs, il existe différents modes de fonctionnement permettant aux capteurs de fournir les données dont une entité externe a besoin :

1. *En continu* : Le capteur fournit en permanence des données à une fréquence préétablie. Cette fréquence peut être fixe ou modifiable dynamiquement. Nous pouvons citer comme exemple une caméra, qui délivre en permanence des images à une fréquence prédéfinie et fixe.
2. *Par événement* : Le capteur fournit des données lors de l'apparition d'un événement d'intérêt. L'événement transporte une information à sémantique plus haute que dans le cas du capteur à transmission continue. Par exemple, un détecteur de mouvement, qui fournit une information ponctuelle renseignant sur le passage -ou l'absence de passage- d'un individu. Le capteur est à l'initiative du déclenchement de l'événement.
3. *Sur demande* : Le capteur fournit des données en réponse à une requête d'une entité externe. Celle-ci interroge le capteur et se met en attente d'une réponse. Le capteur est quant à lui toujours en attente d'une interrogation. En aucun cas, il ne prendra l'initiative d'envoyer une mesure.
4. *Par méthode hybride* : Combine les trois modes précédents.

Dans le cas où nous déployons un réseau de capteurs, nous devons nous demander comment les données capteurs peuvent être véhiculées sur ce réseau. Sur celui-ci, les capteurs deviennent des producteurs d'informations alors que les entités ayant besoin de ces données deviennent des consommateurs d'informations.

Ce mode d'échange d'informations s'appuie sur la notion de bus logiciel permettant de gérer un ensemble d'entités dynamiques, pouvant apparaître ou disparaître du bus. Cette notion de dynamique correspond aux contraintes d'un système à base de capteurs car il est possible d'ajouter ou de retirer un capteur (lors d'une défaillance, d'une maintenance préventive, d'une mise à jour, etc.).

Dans le cas qui nous concerne, il est préférable d'utiliser le bus en mode asynchrone [LAU00]. En effet, les données capteurs sont fournies sur le bus de manière aléatoire. Afin de ne pas scruter en permanence le bus, il est plus judicieux d'être averti lorsqu'une donnée intéressante a été déposée sur le bus. Le couplage entre producteur et consommateur étant d'autant plus faible avec une telle stratégie. Ce découplage permet de supprimer les dépendances entre producteur et consommateur et facilite le transit des données au sein de l'architecture. Il est inutile de connaître précisément le producteur de l'information, seule la nature de l'information est nécessaire pour l'obtenir.

Le modèle *publish/subscribe* est un modèle de communication qui convient à notre architecture auquel il faut ajouter la notion d'événements du modèle événementiel.

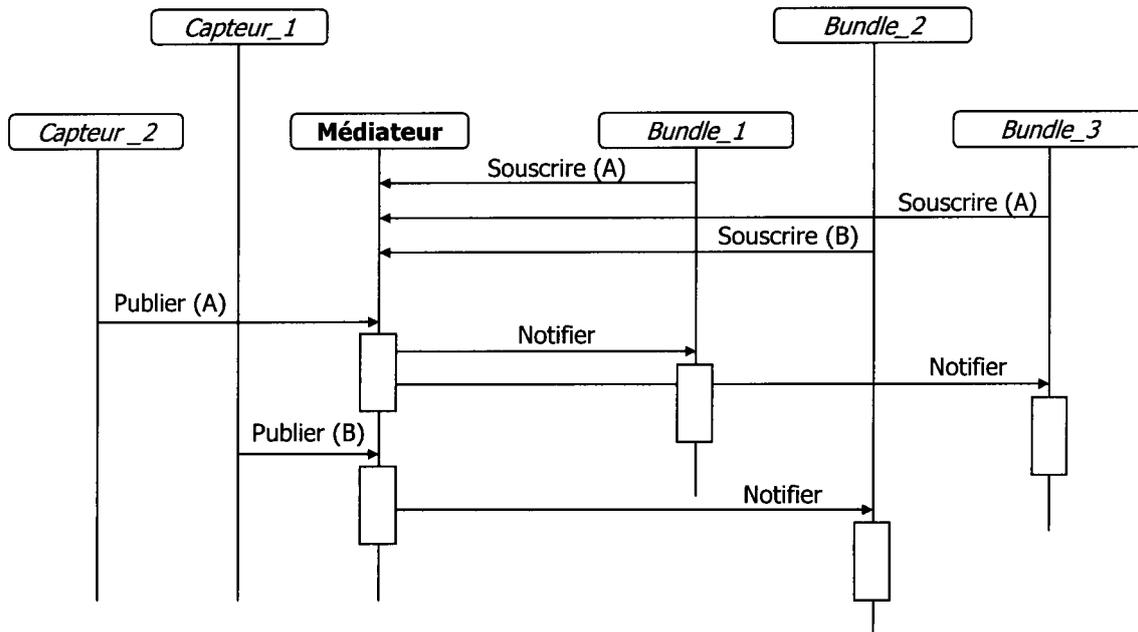


Figure 16 : Le modèle Publish/Subscribe pour des capteurs.

La Figure 16 illustre le concept du modèle *Publish/Subscribe*. Nous y observons deux producteurs (capteur_1 et capteur_2) fournissant des données (A et B) au médiateur du bus. Ce médiateur a pour rôle de gérer les souscriptions à un type d'événements, leur réception, leur filtrage, et leur routage vers le consommateur intéressé.

Toujours dans une optique de dynamique, il faut permettre aux capteurs de se connecter à tout moment au bus événementiel. Pour cela, le bus doit pouvoir supporter les quatre modes de transmission vus précédemment. Afin de correspondre au mieux à la notion de bus événementiel, le troisième mode de transmission d'information est à modifier. En effet, pour garder un faible couplage, il faut que la notion de requête soit transparente pour le consommateur de données. Si le consommateur doit interroger le producteur, il faut que le consommateur ait une connaissance a priori du producteur.

Pour éviter ce problème, nous avons opté pour l'insertion d'un proxy entre le capteur et le bus événementiel. Ce proxy va simuler une requête auprès du capteur et publier les données reçues sur le bus [SCH00]. Nous pouvons ainsi définir la fréquence à laquelle les données seront publiées sur le bus. Les premier et second modes de transmission peuvent s'interfacier sur le bus en respectant la fréquence de publication prédéfinie.

Concernant les données capteur, un prétraitement en vue de leur exploitation est souvent nécessaire. Par exemple, dans le cadre d'une application de vision par ordinateur, les prétraitements consistent à formater l'image reçue par la caméra (binarisation, application de filtre,...). Cette étape est cruciale lorsque nous sommes en présence de capteurs fournissant une masse importante de données. Une couche intermédiaire de prétraitement de données est donc requise.

La couche Adaptateur permet de formater les données issues des capteurs afin de les émettre sur le bus événementiel. Nous pouvons imaginer un capteur directement interfaçable avec notre bus.

Les couches Adaptateur, Prétraitement et Proxy ne sont pas obligatoires. En effet, le proxy intervient lors de l'utilisation d'un capteur fonctionnant par requêtes.

Finalement, le fait d'émettre des données en permanence sur le bus présente tout de même un inconvénient : il arrive que nous soyons confrontés à un engorgement du bus au fur et à mesure de la mise à disposition de données. Pour éviter ce problème, il est judicieux de mettre en place une notion de validité temporelle. Lorsqu'une donnée déposée sur le bus reste à disposition pendant une certaine durée alors qu'aucun consommateur n'est intéressé par cette donnée, celle-ci doit être retirée du bus par le médiateur de bus. Pour pouvoir évaluer la validité d'une donnée, nous avons choisi d'horodater les données (TimeStamp) afin de connaître sa validité temporelle.

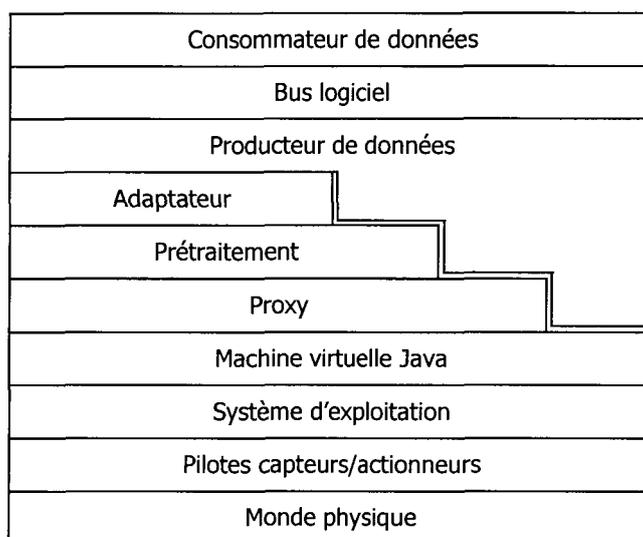


Figure 17 : Ensemble de couches de connexion au bus événementiel.

Au final, nous obtenons une architecture complète permettant l'acquisition de données capteurs et leur mise à disposition. Les événements transitant sur ce bus événementiel ont le format suivant :

- **Topic** : Définit le type de l'événement émis.
- **TimeStamp** : L'horodatage permet de connaître l'instant auquel un événement a été émis.
- **SensorValue** : La valeur fournie par le capteur.
- **SensorUnit** : Représente la classe d'information fournie par le capteur (Température, présence,...).
- **Uncertainty** : Cette donnée permet d'obtenir une valeur reflétant de l'incertitude que l'on peut avoir envers une mesure effectuée par un capteur. Cette incertitude représente une probabilité, en fonction des traitements que l'on pourra rencontrer dans la chaîne d'acquisition de la mesure.
- **SensorID** : Identifiant unique du capteur émettant la donnée.
- **MsgID** : Identifiant unique du message émis. Cet identifiant est incrémenté après chaque envoi de message. Il permet à un consommateur d'ignorer d'anciens messages qui se seraient perdus sur le bus

Le format des événements émis respecte la norme IEEE 1451 et reprend les champs d'une mesure imposés par cette norme. Ainsi, une compatibilité est possible de notre format avec le futur format standard de données capteurs.

Dans le cas de l'habitat communicant, nous passons outre l'étape d'extraction terminologique automatisée et nous définissons une ontologie à partir des connaissances expertes que nous avons du domaine.

L'intérêt des ontologies dans la problématique de l'habitat communicant est de permettre une modélisation précise et complète, à la fois de l'habitat (topographie des lieux), des différents appareils qui s'y trouvent, des services qu'ils peuvent rendre et des occupants en tant qu'acteurs au sein de la maison. Grâce à cette modélisation, on peut ensuite, et toujours dans le cadre de l'ontologie, ajouter un ensemble de règles de fonctionnement et d'interactions de tous ces éléments.

Une fois ce modèle établi, il suffit de transcrire sous forme de règles les connaissances que l'on souhaite voir appliquées dans les différents domaines relatifs à l'habitat communicant (économie d'énergie, aide aux personnes âgées, ...). A partir du moment où ces règles prennent en compte les données relatives à l'état de la maison présentes dans l'ontologie, on voit clairement apparaître la notion de services contextualisés recherchée précédemment.

3.5.3 La couche Inférence

La couche Inférence fournit les outils permettant de raisonner à partir des données provenant de la couche Contexte. A partir de l'ontologie de l'habitat communicant, la couche Inférence doit permettre un raisonnement basé sur le contenu du modèle. En modélisation de la connaissance, le raisonnement est une opération qui consiste à tirer une conclusion à partir d'un ensemble de propositions tenues pour vraies. Généralement, les conclusions sont effectuées à partir de règles de base. C'est ce que l'on appelle la déduction, un des principes de la logique.

En modélisation de la connaissance, le raisonnement se base sur un ensemble de règles, différenciées selon deux types : les règles à chaînage avant et les règles à chaînage arrière. Le langage OWL n'utilise que les règles à chaînage arrière et ne permet pas de traduire celles à chaînage avant.

Les règles à chaînage arrière : Pour les règles à chaînage arrière, on se place du point de vue des classes de l'ontologie et non des individus, ce qui fait que l'on raisonne sur des ensembles plutôt que sur des éléments. Ces règles se définissent comme des conditions d'appartenance à une classe, ces conditions pouvant être nécessaires ou nécessaires et suffisantes. Ainsi un individu ne peut être rattaché à une classe que s'il remplit toutes les conditions nécessaires et suffisantes d'appartenance à cette classe. A l'inverse, un individu n'appartenant pas à la classe mais remplissant toutes les conditions sera automatiquement classé si on applique un raisonneur (ou « moteur d'inférence ») à l'ontologie. Les raisonneurs sont présentés par la suite dans ce paragraphe.

```

<owl:Class rdf:about="#Maison_occupée">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Maison"/>
        <owl:Restriction>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#contient"/>
          </owl:onProperty>
          <owl:someValuesFrom rdf:resource="#Personne"/>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
  <owl:disjointWith rdf:resource="#Maison_inoccupée"/>
</owl:Class>

```

1

```

<owl:Class rdf:ID="Maison_inoccupée">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class>
          <owl:complementOf>
            <owl:Class rdf:ID="Maison_occupée"/>
          </owl:complementOf>
        </owl:Class>
        <owl:Class rdf:ID="Maison"/>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
  <owl:disjointWith>
    <owl:Class rdf:about="#Maison_occupée"/>
  </owl:disjointWith>
</owl:Class>

```

2

Figure 20 : Règles à chaînage arrière décrivant ce qu'est une maison occupée et une maison inoccupée.

La Figure 20 présente un ensemble de règles à chaînage arrière qui définit ce qu'est une maison occupée et une maison inoccupée. La première partie montre les conditions nécessaires et suffisantes pour définir ce qu'est une maison occupée : il faut qu'elle contienne au moins une personne.

De la même manière, la seconde partie définit ce qu'est une maison inoccupée. Pour cela, l'ontologie indique qu'une maison inoccupée est le complément d'une maison occupée.

Les règles à chaînage avant : Dans le cas des règles à chaînage avant, on se place du point de vue des individus. Les règles sont de la forme « Corps => Tête », où le corps de la règle est un ensemble de conditions à remplir (« $A \wedge B \wedge C \wedge \dots$ ») pour que la règle soit appliquée alors que la tête représente l'information qui doit être déduite par le raisonneur si la règle est appliquée. Ce type de règle est donc beaucoup plus intuitif car il correspond à des règles facilement formulables en langage naturel : « *SI* telles et telles conditions sont vraies, *ALORS* on en déduit telle information ».

Dans le cadre de l'habitat communicant, certains manques d'expressivité du langage OWL font apparaître des difficultés de modélisation, notamment la limitation aux règles à chaînage arrière. S'il est en effet possible en OWL de définir des conditions pour que des individus appartiennent à une classe, il n'est pas possible de poser des conditions pour qu'un individu possède ou non une propriété.

D'autre part, si OWL permet d'appliquer des restrictions sur le nombre de propriétés d'un individu (opérations de comparaison ou d'égalité), il ne permet pas de faire des opérations sur les valeurs mêmes de ses propriétés. Il n'est par exemple pas possible de faire des additions ou des soustractions sur des données capteurs, ou de manière plus gênante, de faire des comparaisons entre les données présentes dans l'ontologie – par exemple entre une température captée et un seuil de confort défini par l'utilisateur.

Finalement, l'ontologie développée dans le cadre de l'habitat communicant doit satisfaire à des besoins de flexibilité. En effet, l'utilisateur doit pouvoir à tout moment débrancher un appareil ou ajouter un nouveau capteur. Pour cela, seul le peuplement de l'ontologie (les individus) doit être concerné par ces changements car le modèle en lui-même (classes, propriétés et règles) ne peut en aucun cas être modifié dynamiquement. Or une certaine partie de l'expressivité du langage OWL n'est accessible qu'en définissant des classes spécifiques à un individu donné. Cette contrainte nous oblige à connaître a priori le peuplement de l'ontologie, et est donc contraire à toute notion de dynamicité.

Pour combler ces manques d'expressivité, il est nécessaire de trouver une solution pour ajouter des règles à chaînage avant dans une ontologie modélisée en OWL. En la matière, certains raisonneurs ou certaines plateformes de manipulation d'ontologies offrent la possibilité de charger en mémoire une ontologie modélisée en OWL et d'y ajouter ensuite les règles à chaînage avant, modélisées dans un langage propriétaire spécifique à l'application. C'est le cas de la plateforme Jena [JENA] par exemple. Néanmoins, cette approche est problématique puisque le modèle complet est alors spécifique à une application et n'est donc plus portable, la transcription des règles à chaînage avant devant être modifiée si on souhaite changer de raisonneur.

Notre choix s'est donc porté sur les travaux du W3C, en passe de devenir une recommandation, qui définit un langage de règles entièrement basé sur OWL : le Semantic Web Rule Language (SWRL) [4]. S'inspirant du langage non standardisé RuleML, SWRL permet d'importer l'ontologie OWL correspondant à SWRL dans l'ontologie en cours de création. Cette importation crée de nouveaux concepts relatifs aux règles (tête, corps, variable,...). Celles-ci qui permettent d'ajouter des individus qui vont représenter les différentes composantes des règles à chaînage avant que l'on souhaite mettre en œuvre. De cette manière, le travail est déporté au niveau du raisonneur, qui interprète cette partie de l'ontologie de manière spécifique.

Le deuxième point fort du langage SWRL est l'intégration d'opérateurs mathématiques prédéfinis (appelés « built-ins »). Dans le cadre de notre problématique, ils permettent donc de faire toutes les opérations et comparaisons nécessaires au traitement des données capteurs. En pratique, ces built-ins sont des individus particuliers ajoutés lorsque l'on importe l'ontologie spécifique à SWRL dans l'ontologie que l'on crée.

$Lampe(?l) \wedge Maison_inoccupée(?m) \wedge estLocalisée(?l, ?m)$ $\rightarrow Lampe_éteinte(?l)$
$Radiateur(?r) \wedge Maison_inoccupée(?m) \wedge estLocalisée(?r, ?m)$ $\rightarrow Radiateur_éteint(?r)$
$Capteur_d_identification(?c) \wedge Etre_vivant(?e) \wedge Localisation(?loc) \wedge$ $estLocalisée(?c, ?loc) \wedge aldentifié(?c, ?e)$ $\rightarrow estLocalisée(?e, ?loc)$
$Radiateur(?r) \wedge Capteur_de_température(?c) \wedge Pièce(?p) \wedge Maison_occupée(?m) \wedge$ $estLocalisée(?r, ?p) \wedge estLocalisée(?c, ?p) \wedge estLocalisée(?r, ?m) \wedge$ $consigneBasse(?r, ?trad) \wedge valeurCaptée(?c, ?tcapt) \wedge swrlb:lessThan(?tcapt, ?trad)$ $\rightarrow Radiateur_allumé(?r)$

Figure 21 : Exemple de règles à chaînage avant.

Sur la Figure 21, nous présentons un ensemble de règles à chaînage avant au format SWRL définissant :

1. Si une pièce est inoccupée alors on éteint les lampes qui y sont situées.
2. Si la maison est inoccupée alors on éteint les radiateurs qui y sont situés.
3. Si un capteur d'identification a identifié un individu et que ce capteur est situé dans une pièce alors on cet individu est situé dans cette même pièce.
4. Si la maison est occupée et que la température d'une pièce est inférieure à la consigne basse du radiateur qui y est situé alors on allume le radiateur de cette pièce.

Pour l'instant, le langage SWRL ne fournit que des règles obéissant à la logique du premier ordre (utilisation de connecteurs logiques ET (\wedge), IMPLIQUE (\rightarrow), NEGATION (\neg)). Le langage SWRL propose des extensions permettant d'utiliser des opérateurs plus évolués tels que des opérateurs de comparaison ('*swrlb:lessThan*' sur la Figure 21), des opérateurs mathématiques (addition, soustraction, ...), la gestion des booléens, des chaînes de caractères, ... La liste complète est présentée dans [HOR04].

Le raisonneur : En traitement de la connaissance, le terme raisonneur ou moteur d'inférence désigne le programme chargé de classer une ontologie peuplée. Il a pour objectif de déduire toutes les données possibles à partir d'un modèle et de son peuplement et, en théorie, quelque soit l'ordre des règles. Pour cela, il applique l'ensemble des règles de l'ontologie à la population de départ (base de faits initiale) et, à chaque fois qu'un nouveau fait est déduit, il réapplique les règles sur lesquelles ce fait a un impact. Les implémentations existantes se basent généralement sur des algorithmes connus tels que l'algorithme de RETE [FOR82] ou les algorithmes de raisonnement par tableaux, sur lesquels sont rajoutées diverses optimisations spécifiques au raisonneur.

Dans le cas de l'ontologie de l'habitat communicant, le raisonneur doit satisfaire à certaines conditions introduites par l'utilisation des règles à chaînage avant et les objectifs de service à l'habitant : le raisonneur doit faire les déductions dans un temps raisonnable pour l'habitant, et doit être compatible non seulement avec les langages OWL et SWRL. En effet, comme nous l'avons expliqué précédemment, le raisonneur doit prendre en compte la portion de l'ontologie correspondant au SWRL de manière à l'interpréter comme des règles à chaînage avant, et non comme du code OWL « simple ».

Comme le langage SWRL est encore relativement jeune (proposé en mai 2004) et qu'il nécessite des modifications profondes au niveau du raisonneur pour être pris en charge, il existe encore peu de raisonneurs qui le supportent en totalité.

KAON2 n'est pas un raisonneur open-source mais est gratuit dans le cadre d'un usage académique ou de recherche. Il est développé en Java et est fourni avec une API qui permet de l'intégrer dans toute application développée en Java. De plus, il intègre une plateforme de manipulation d'ontologies qui lui est propre. Il traite les ontologies en OWL et SWRL avec de très bonnes performances, mais a comme principal inconvénient un support partiel de ces deux langages. En effet, certaines spécifications telles que les restrictions sur les valeurs des propriétés en OWL, ou certains built-ins en SWRL ne sont tout simplement pas acceptées par le raisonneur si elles sont rencontrées lors du chargement de l'ontologie.

3.5.4 La couche Action

La couche Action permet de fournir physiquement le service à l'habitant par le biais d'actionneurs. Cette couche est similaire à la couche Perception vue d'une manière inverse. En effet, les capteurs fournissent des informations à disposition d'une entité externe ; alors que dans le cas des actionneurs, c'est l'entité externe qui met à disposition des informations aux actionneurs. Les actionneurs ont alors pour rôle de mettre en œuvre la consigne émise par l'entité externe.

Afin de garder un couplage faible entre l'entité externe et l'actionneur, il est préférable que le producteur d'informations (l'entité externe) et le consommateur d'informations (l'actionneur) n'aient pas une interaction directe entre eux. Le producteur émet une consigne, si un consommateur est capable d'appliquer la consigne reçue, elle sera mise en œuvre, sinon la consigne disparaît. Nous constatons ainsi que le modèle *publish/subscribe* est adapté à la gestion d'actionneurs. Ce modèle permet un découplage total entre producteurs et consommateurs et autorise l'ajout, la suppression et la mise à jour d'actionneurs, sans mettre en péril l'architecture complète.

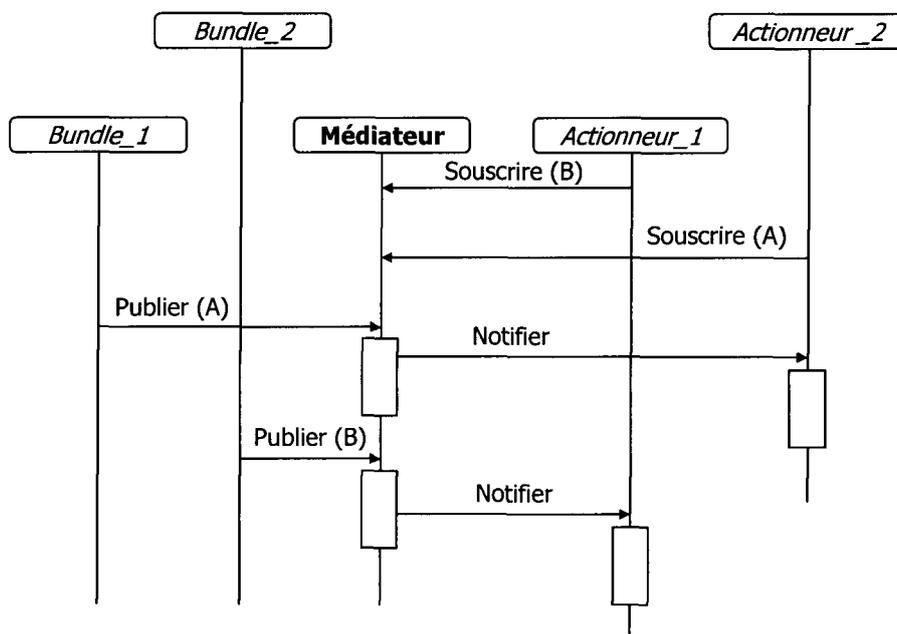


Figure 22 : Le modèle Publish/Subscribe pour des actionneurs.

3.5.5 Intégration des quatre couches au sein d'une architecture OSGi

Dans cette section, nous décrivons l'intégration dans une architecture de déploiement OSGi des quatre couches présentées. La Figure 23 présente l'architecture globale que nous avons mise en place afin de gérer le contexte pour l'habitat communicant.

Ce middleware s'inspire des travaux de l'équipe de Sumi Helal [YAN06]. Ce travail présente une architecture de programmation pour les espaces pervasifs. Notre architecture se démarque de ces travaux au niveau de la couche événementielle.

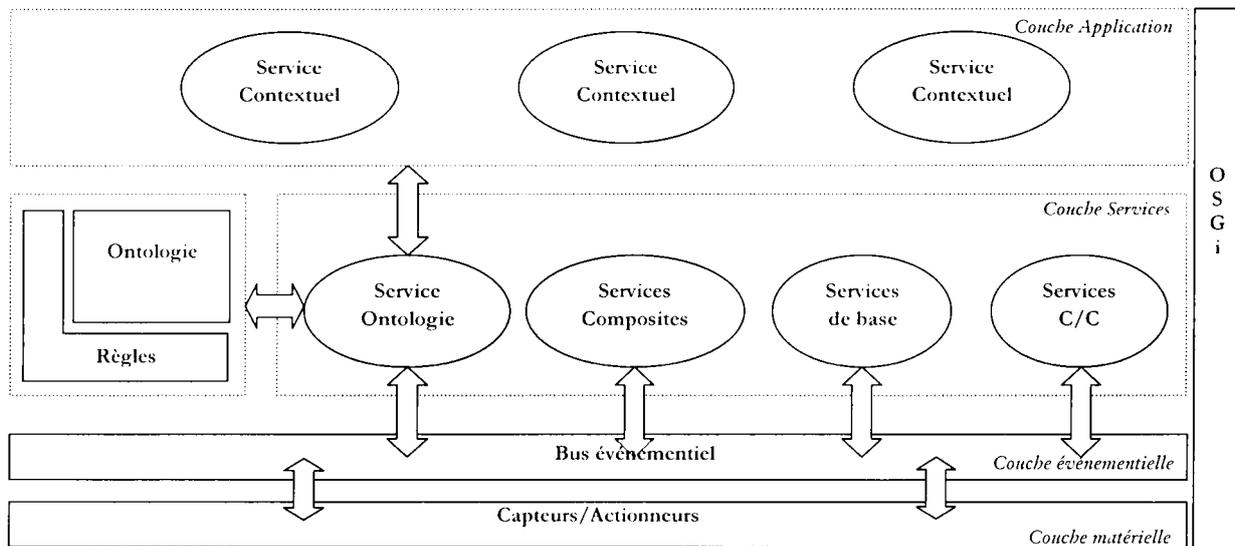


Figure 23 : Architecture de gestion de contexte orientée service.

La couche matérielle : Elle permet au système informatique d'interagir physiquement avec l'environnement de la maison. Cette couche est constituée de deux parties permettant d'obtenir des informations de l'environnement (capteurs) et de les modifier (actionneurs). Vis-à-vis du modèle PCIA, cette couche matérielle réalise les couches P et A.

La couche service : Elle regroupe quatre types de services.

1. Les services de *base* qui ont comme tâche de fournir des données contextuelles sur une seule classe de capteurs ou actionneurs.
2. Les services *composites* qui ont pour rôles d'agréger un ensemble de données contextuelles afin d'en tirer des données d'un degré sémantique plus haut.
3. Les services *Contrôle/Commande (C/C)* qui permettent de rendre des services simples comme *j'appuie sur un bouton, la lampe s'allume*. Ces services sont de l'ordre de l'automatisme que nous connaissons actuellement.
4. Les services *contextuels* sont des services fournis à l'habitant et basés sur les informations observées dans l'environnement. Ils peuvent utiliser les données issues des services de bases, ainsi que les données issues des services composites ; ou encore les données issues des services C/C.

Les services contextuels sont à considérer au niveau de la couche application. Ils sont liés à l'état du modèle de contexte. C'est à partir de ce modèle et de son inférence que les services contextuels seront fournis. Contrairement aux services C/C, ce type de service est rendu de manière transparente pour l'habitant. Par exemple, « *j'entre dans ma maison, la maison allume l'éclairage, diffuse ma musique préférée et me diffuse mes nouveaux mails sur mon écran de télévision* ». Ainsi, le service contextuel prend en compte l'entrée d'une personne, détermine son identité et fournit les services adaptés.

Les services peuvent, en outre, avoir des actions localisées. En effet, un service peut être local à une pièce et assurer la gestion simple du chauffage. Un service peut également être global : c'est le cas d'un service anti-intrusion, qui ne s'adresse pas à une seule pièce de l'habitat. Nous en déduisons la nécessité de nommer les données captées : ce nommage permet d'agir de manière localisée pour les services localisés et d'une manière plus globale pour les services globaux.

Le service *ontologie* est un service particulier, en charge de la modélisation de l'habitat. Il est à l'écoute des données circulant sur le bus événementiel assure le remplissage du modèle de contexte à partir de ces données. De la même manière, lors d'un changement de contexte, il est en charge d'émettre sur le bus événementiel les consignes envers les actionneurs ou services intéressés.

La couche événementielle : Cette couche permet de véhiculer les informations provenant des capteurs ou allant vers les actionneurs. Elle s'appuie notamment sur un mécanisme de publication/souscription événementielle, où des entités publient des données et d'autres souscrivent à certaines données.

La plateforme de déploiement de services OSGi fournit nativement, depuis la version 4, un bus événementiel. C'est le service EventAdmin. La pile événementielle implémentée par ce service propose un système de messagerie événementielle synchrone ou asynchrone.

La couche Ontologie+Règles : Vis-à-vis du modèle PCIA, cette couche matérielle réalise les couches C et I. Cette couche regroupe la partie modélisation du contexte au sein de l'habitat et le phénomène d'inférence basé sur des règles. C'est à ce niveau que la modélisation OWL est effectuée avec un raisonnement SWRL basé sur des règles prédéfinies. Le service ontologie est un pont important entre l'ontologie et le reste de l'architecture. Nous développerons davantage ce principe dans les parties suivantes.

3.6 Interfaçage de l'ontologie avec l'architecture à service

3.6.1 Description du service de gestion d'ontologie

A partir des choix évoqués précédemment, nous nous focalisons sur le service ontologie en charge de modéliser le contexte et de rendre les services contextualisés à l'habitant.

Le fonctionnement général de ce service particulier et notamment la manière dont il traite les données est représenté sur la Figure 24. Le principe de fonctionnement consiste à charger en mémoire l'ontologie de l'habitat au démarrage de la passerelle domestique (modèle générique non peuplé sous la forme de fichiers texte) puis à peupler cette ontologie au travers d'une configuration initiale. Ensuite, connaissant l'état de la maison à un instant t , un raisonnement périodique est effectué afin de définir le nouvel état des actionneurs.

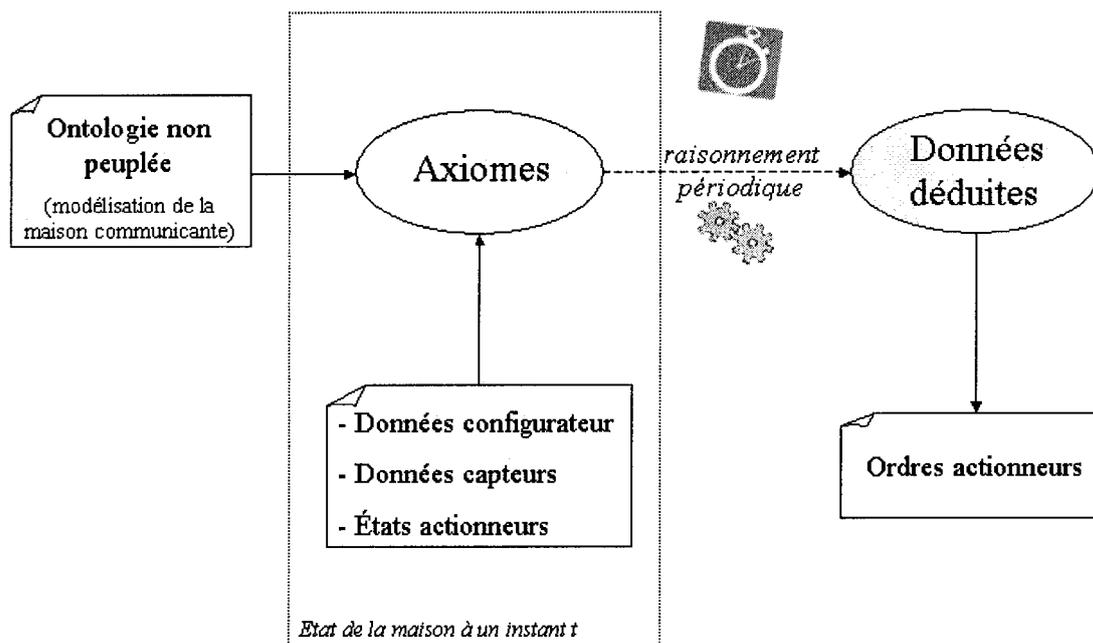


Figure 24 : Traitement des données par le service ontologie.

Dans le cadre de la plateforme OSGi déployée sur la passerelle domestique, l'ontologie et le raisonneur qui lui est associé sont mis en œuvre sous forme d'un Bundle OSGi. Tout en intégrant ces deux éléments dans la plateforme, ce Bundle doit permettre l'interaction entre les différents services déployés sur la plateforme (sous forme de Bundles eux aussi) et l'ontologie. Quatre fonctions principales ont ainsi émergé permettant l'interaction entre l'ontologie et les couches de l'architecture :

- **Ajouter** un nouvel individu dans l'ontologie de manière à pouvoir configurer l'ontologie lors de l'installation de la passerelle domestique dans la maison (définition des différentes pièces, des différentes personnes qui y vivent, etc.) ou à faire évoluer celle-ci au cours de son utilisation (nouveau thermomètre, nouveau radiateur, ...),
- **Supprimer** un individu existant pour les mêmes besoins que la fonction précédente,
- **Modifier** les propriétés ou la classe d'un individu de manière à prendre en compte dynamiquement les changements de l'environnement (changement de température dans une pièce, mise en fonctionnement d'un appareil par l'utilisateur, ...),
- **Consulter** l'état d'un individu (propriétés, classe, etc.) car l'ontologie détermine l'état *théorique* le plus adéquat à un instant donné pour l'ensemble des appareils de la maison. Il revient ensuite aux appareils de se conformer à ces choix, en modifiant éventuellement leur état.

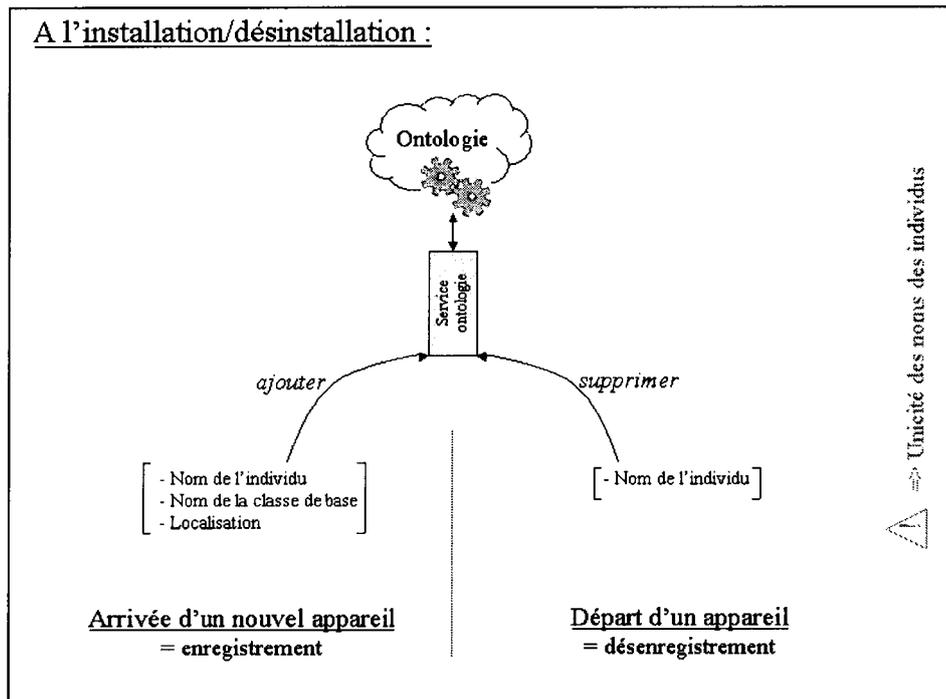


Figure 25 : Interface du service ontologie pour la configuration.

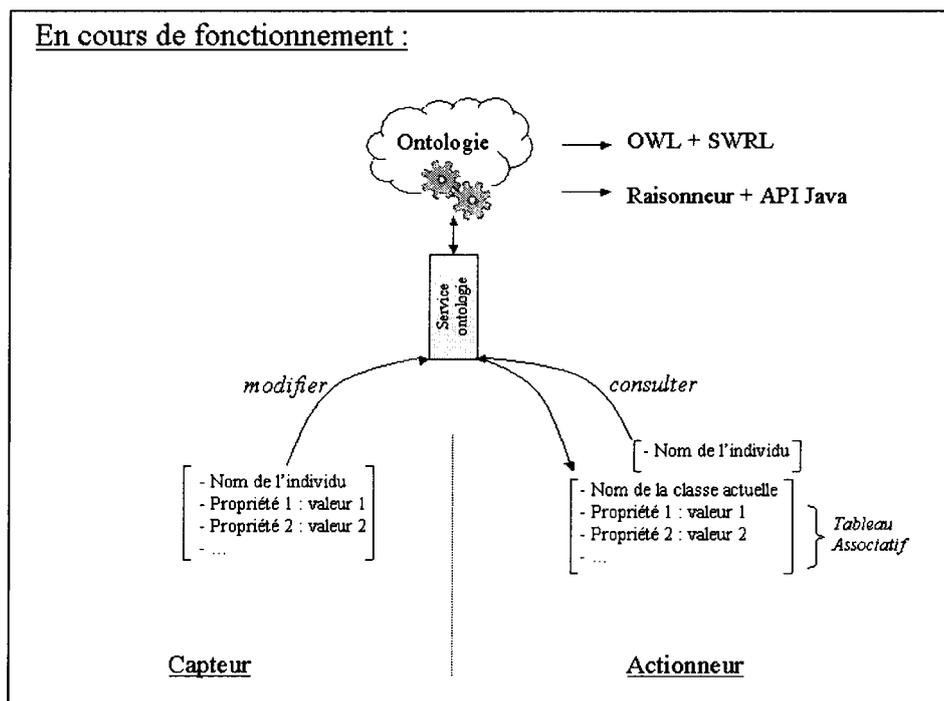


Figure 26 : Interface du service ontologie en cours de fonctionnement.

Cet ensemble est ainsi implémenté sous forme d'un Bundle s'interfaçant avec le reste de l'architecture, et notamment avec le bus événementiel. Une couche logicielle supplémentaire est ajoutée et est chargée de transmettre les données en provenance et à destination du bus événementiel et donc de gérer le protocole sous-jacent de souscription et publication d'évènements de types appropriés. Cette couche est réalisée à l'aide de deux Bundles, l'un s'occupant des données en provenance des capteurs, l'autre des données en provenance et à destination des actionneurs (voir Figure 27).

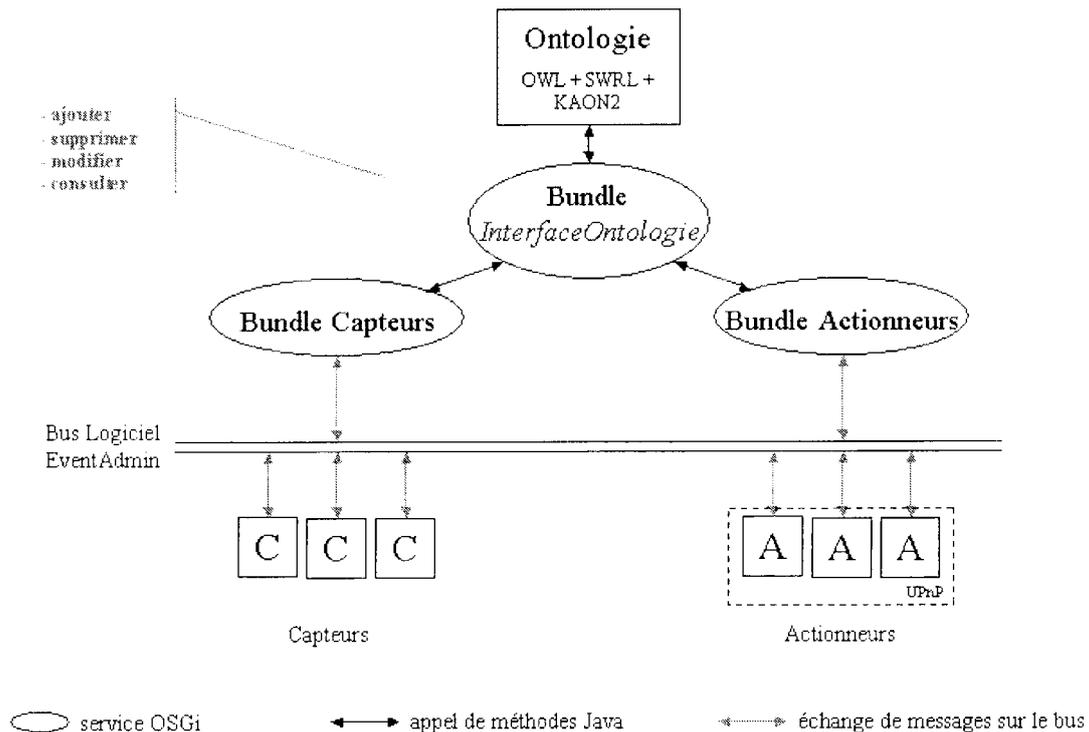


Figure 27 : Interfaçage entre le bus événementiel et le service ontologie.

Il ressort de ce choix d'architecture un élément important quant à la gestion des actionneurs. En effet, du côté des capteurs le sens de circulation de l'information est unique : toujours des capteurs vers l'ontologie. Grâce à cette particularité, il n'est pas nécessaire de savoir au niveau de l'ontologie quelle est la technologie de chaque capteur ni quel est son moyen de communication. Il suffit de savoir quelle grandeur physique il transmet (température, luminosité, etc.) et de souscrire sur le bus à tous les événements relatifs à ces données.

Du côté des actionneurs, la problématique est différente : plusieurs technologies de commande existent (X10¹⁵, Konnex¹⁶,...) et elles s'appliquent parfois à plusieurs moyens de communication (CPL¹⁷, hertzien,...). Si l'objectif était de commander les actionneurs directement au niveau de l'ontologie, il serait donc nécessaire de gérer à ce niveau toutes les technologies et moyens de communications possibles. Une telle approche est peu pertinente puisqu'on mélange dans ce cas le service rendu par l'appareil et la technologie sur laquelle il s'appuie pour rendre ce service.

¹⁵ X10 est un standard de communication utilisé principalement pour la domotique. La transmission des données s'effectue à travers le réseau électrique par un système à courants porteurs. Le système X10 date de 1975, créé par le Turbang Yannick à Arlon en Belgique dans le but de pouvoir contrôler différents appareils électriques au sein d'une habitation. Contrairement à ses concurrents, le protocole est ouvert et donc libre de droits, ce qui offre de nombreux avantages.

¹⁶ Le standard Konnex (KNN) est né de la convergence des Bus Bâtibus, EIB et EHS. Il devient ainsi le standard européen de la domotique. Il regroupe plus d'une centaine de fabricants européens dans le domaine de l'électrotechnique ce qui garanti en théorie l'interopérabilité des produits et de la pérennité du système.

¹⁷ Les courants porteurs en ligne (CPL) est une technologie permettant le transfert d'informations numériques passant par les lignes électriques d'une habitation.

Afin d'éviter ce problème, le service ontologie s'appuie sur la technologie UPnP et ses extensions à la domotique. L'utilisation d'UPnP permet de créer une couche d'abstraction au dessus des actionneurs de manière à ne transmettre au niveau de l'ontologie que l'état à adopter par l'actionneur (allumé ou éteint par exemple) et non pas comment atteindre cet état. Nous faisons abstraction complètement de la technologie utilisée par l'actionneur pour se concentrer sur sa fonction.

UPnP : UPnP (Universal Plug and Play) a pour objectif de permettre la réalisation de réseaux se configurant de manière totalement autonome sans intervention de l'utilisateur. Pour cela, UPnP permet l'auto-configuration, la découverte de périphériques et de services, le contrôle et la transmission d'informations entre périphérique de façon relativement simple. Pour offrir ceci, UPnP utilise plusieurs standards tels que TCP/IP, HTTP, SOAP, GENA et XML. La norme UPnP ne s'appuie plus sur des pilotes (drivers), le pilote est remplacé par des protocoles universels. On peut donc interfacer les périphériques UPnP avec tous les langages de programmation ou système d'exploitation. UPnP forum¹⁸ est l'investigateur de cette norme et a défini une pile de protocoles permettant cette communication entre périphériques de manière dynamique et visant à simplifier l'implémentation des réseaux.

3.6.2 Intégration de la volonté de l'utilisateur

Afin d'être pleinement accepté, le système mis en œuvre doit laisser la main à l'utilisateur final. La priorité dans un tel système décisionnel est la volonté de l'habitant : il est impératif que l'utilisateur n'ait pas le sentiment de perdre le contrôle de son lieu de vie. Il est alors primordial de prendre en compte les souhaits de l'habitant dans le modèle. Par exemple, il paraît peu envisageable que le système décide seul d'éteindre la lampe que l'habitant vient d'allumer sous prétexte d'économie d'énergie.

Une première solution à ce problème consiste à considérer que l'utilisateur doit toujours avoir le dernier mot et que le système doit se conformer à sa volonté. Certains paradoxes apparaissent néanmoins lorsque l'on part de ce principe. Par exemple, si l'utilisateur allume une lampe dans sa maison et que peu de temps après, il part en urgence en oubliant d'éteindre cette lampe : il serait intéressant en termes d'économie d'énergie que la maison éteigne d'elle-même la lampe à la place de l'utilisateur. Pourtant, en laissant le dernier mot à l'utilisateur, son choix d'allumer la lampe ne peut être remis en question et le système est dans l'incapacité d'éteindre la lampe. Cette solution n'est donc pas satisfaisante.

La seconde solution consiste à solliciter l'utilisateur à chaque fois que son comportement rentre en contradiction avec une décision prise par le système de manière à ce qu'il choisisse l'état à adopter par le système et garde ainsi le dernier mot. Plusieurs inconvénients découlent cependant de cette démarche. Il faut tout d'abord que le système ait la capacité d'entrer en contact avec l'utilisateur, y compris lorsque celui-ci n'est pas chez lui. D'autre part, le risque est de voir l'utilisateur submergé par le nombre de sollicitations ou d'être sollicité à un moment inopportun. Vraisemblablement, l'utilisateur ne répondra plus aux demandes du système tel que nous l'observons avec le Contrôle des Comptes d'Utilisateurs de Windows Vista (UAC : User Account Control) qui se révèle être plus gênant qu'utile.

Pour rendre au mieux les services à l'habitant, il faut donc trouver un compromis entre les deux approches précédentes. Pour cela, lors de la modélisation, nous avons choisi de classer les règles (et donc les services associés) en trois groupes :

¹⁸ www.upnp.org/

1. Les règles sur lesquelles la volonté de l'utilisateur est toujours prioritaire (par exemple des règles touchant à l'éclairage d'une pièce lorsque l'utilisateur s'y trouve) ;
2. Les règles dont on sait qu'elles n'entrent pas directement en contradiction avec ce que souhaite l'utilisateur et qui sont donc toujours prioritaires par rapport à sa volonté (par exemple éteindre les lampes quand la maison est inoccupée) ;
3. Les règles qui peuvent présenter une incertitude en fonction de ce que l'utilisateur a en tête et qui nécessitent de lui demander explicitement quelle est la meilleure attitude à adopter pour le système.

Si l'utilisateur a en plus la possibilité d'activer ou non les services associés aux règles prioritaires, cette solution lui permet de toujours garder un sentiment de contrôle de sa maison tout en délivrant au mieux les services proposés. Ainsi, s'il souhaite partir de chez lui en laissant les lumières allumées pour, par exemple, faire croire que la maison est occupée, libre à lui de désactiver le service d'économie d'énergie.

Pour que cette approche reste pertinente, il est cependant nécessaire de limiter au maximum le nombre de règles entrant dans le troisième groupe de manière à ne pas retomber sur un problème de sur-sollicitation. Cette contrainte nécessite dans la pratique un effort supplémentaire notable au niveau de la modélisation car il faut essayer de deviner le comportement et les souhaits de l'utilisateur de manière à lui rendre un maximum de services à la fois sans le solliciter et sans le gêner.

4 Validation de l'architecture de gestion de contexte dans l'habitat communicant

4.1 Mise en œuvre de l'architecture de gestion de contexte

L'ensemble des choix architecturaux expliqués précédemment et des solutions d'implémentation adoptées a permis d'aboutir à l'architecture logicielle détaillée Figure 28. Cette architecture a été déployée sur un prototype de passerelle domestique ce qui nous a permis de constater en pratique ses forces et ses faiblesses (voir section 4.2 de ce chapitre).

La liste des Bundles associés à des pilotes particuliers (X10, KNX, etc.) est non exhaustive. En effet, le point fort de la plateforme que nous avons développée est de pouvoir s'adapter à n'importe quelle technologie domotique existante.

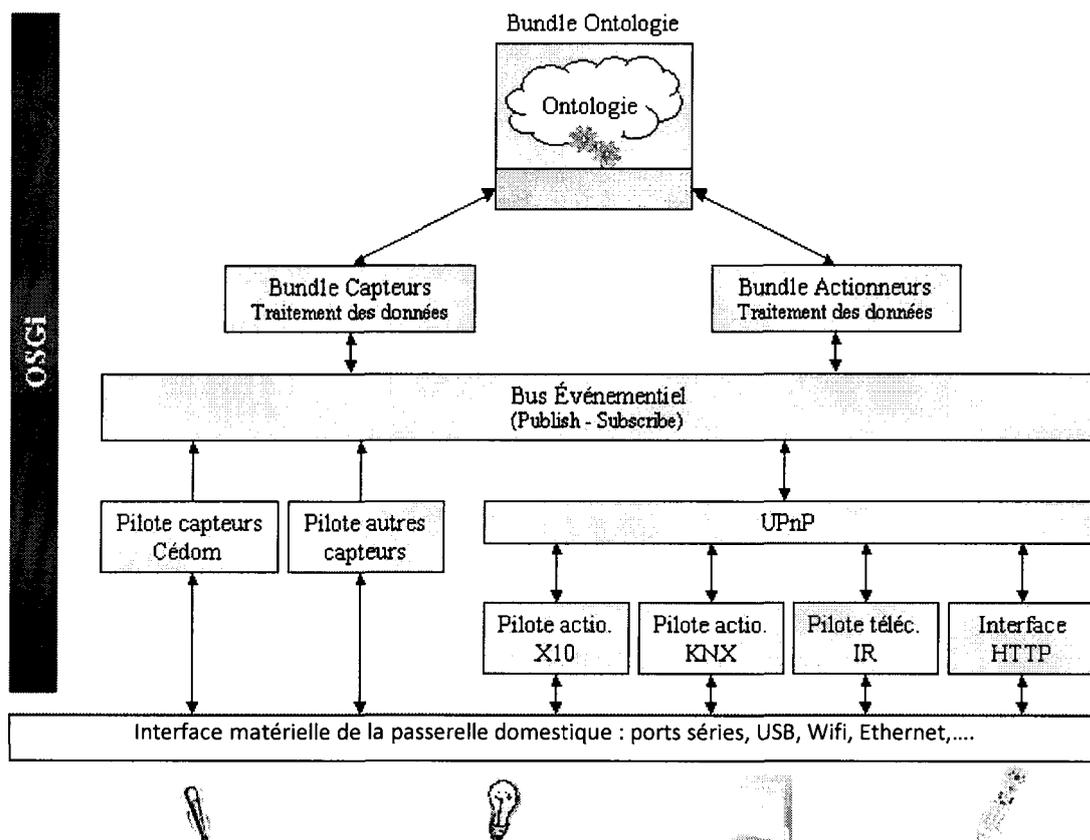


Figure 28 : Architecture logicielle détaillée.

Afin de valider notre architecture, nous avons procédé à la mise en œuvre d'une expérimentation en conditions réelles. Ce cas nous a permis de tester notre système en validant plusieurs points : la robustesse du bus événementiel intrinsèque à OSGi™ (EventAdmin dans la Release 4 d'OSGi™), la validité partielle de l'ontologie, le mécanisme d'inférence et les actions correspondantes.

Pour cela, nous avons élaboré le scénario suivant : « je me déplace dans une pièce et le système allume automatiquement la lampe la plus proche de moi ». Pour cela, nous avons placé des capteurs de détection de présence dont on peut définir la zone de détection. Ainsi, nous pouvons découper notre pièce en plusieurs zones et détecter grossièrement la localisation de l'individu. Connaissant a priori la localisation des individus, il est possible de mettre en œuvre une règle permettant de lier la localisation d'un individu et la localisation d'une lampe. Nous pouvons rapprocher ce scénario au Follow-me [SAT05] dans lequel le système suit un individu et propose une continuité de services durant son déplacement.

Les capteurs de détection de présence que nous utilisons ont une fréquence de 2 Hz, c'est à dire que nous pouvons obtenir une information de présence toutes les 500 millisecondes. Cette fréquence d'émission permet de valider la robustesse du bus événementiel.

Sur la Figure 29 est représentée la manipulation que nous avons mise en place pour valider notre architecture. Nous y trouvons un réseau de capteurs de détection de mouvement sans fil fournissant respectivement leurs données à un ordinateur par le biais d'une passerelle agrégeant les données de chacun des capteurs.

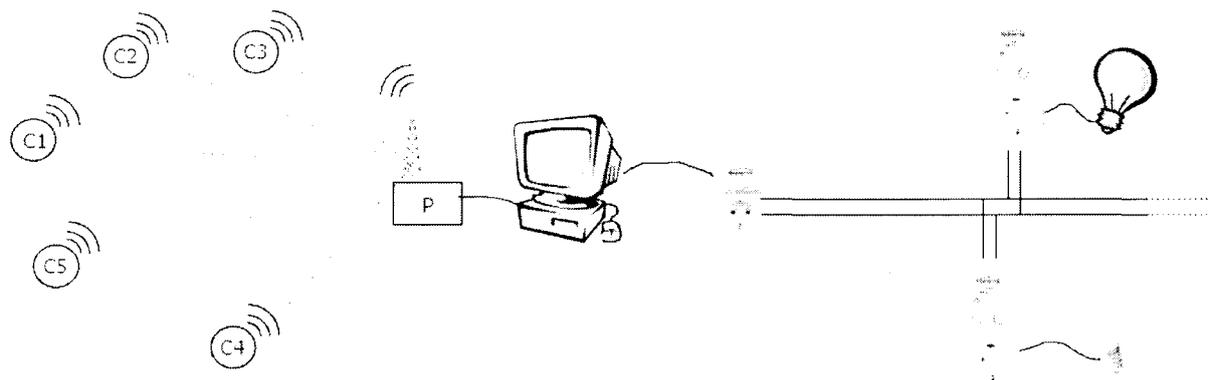


Figure 29 : Mise en relation des capteurs/actionneurs dans l'architecture de gestion de contexte.

De l'autre côté de l'ordinateur, nous avons le réseau d'actionneurs qui permet d'allumer ou d'éteindre les lampes. Ces lampes sont commandées grâce à leurs prises électriques et plus particulièrement par l'intermédiaire de matériel X10. Cette technologie s'appuie sur les courants porteurs en ligne (CPL) pour véhiculer l'information.

L'ordinateur utilisé pour nos tests fonctionne sous Windows XP sur lequel nous avons déployé Knopflerfish¹⁹, une des implémentations les plus abouties de la version 4 d'OSGi™. Elle a également l'avantage d'intégrer le bus événementiel EventAdmin.

Au niveau logiciel, lorsque l'individu occupe une zone, cette information est émise sur le bus publish/subscribe afin d'être prise en compte dans l'ontologie.

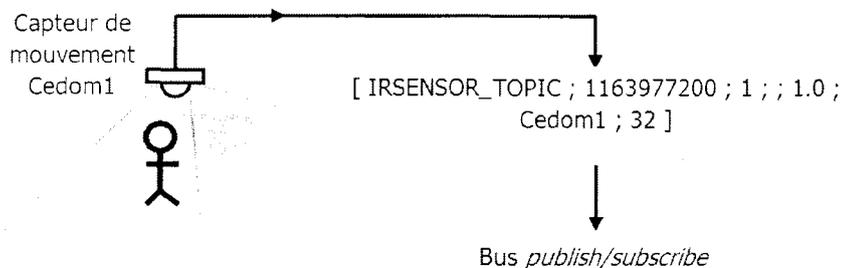


Figure 30 : Remontée d'un événement capteur.

¹⁹ <http://www.knopflerfish.org/>

L'acheminement d'une donnée capteur suit les étapes suivantes :

Etape 1 :

La première étape consiste à la détection d'un individu dans une des zones. Quand un capteur de mouvement perçoit un individu dans son champ de détection, le pilote du capteur émet un événement sur le bus publish/subscribe (voir Figure 30). L'événement émis transporte les informations concernant son sujet (IRSENSOR_TOPIC), l'horodatage (1163977200), la valeur captée (1), l'unité (ici pas d'unité pour ce type de capteur), la confiance accordée à cette valeur (1.0), l'identifiant du capteur (Cedom1) et l'identifiant du message (32).

Etape 2 :

Une fois l'évènement déposé sur le bus, les services ayant souscrit au sujet IRSENSOR_TOPIC reçoivent l'évènement. Le service ontologie fait également partie de ces services. Il souscrit à tout type de sujet portant sur des données capteurs afin de remplir le modèle ontologique. Dans notre exemple, il parcourt l'ontologie dans l'objectif de trouver l'individu correspondant, puis met à jour la valeur captée correspondante. Grâce aux relations de composition, d'héritage et de sémantique entre les classes, il devient possible d'extraire un nombre significatif d'informations relatives à l'individu. La Figure 31 montre une partie de l'ontologie dans laquelle certaines relations de composition et d'héritage sont présentées lorsque le capteur Cedom1 est activé. Nous pouvons déduire ainsi des informations complémentaires : le capteur Cedom1 est localisé dans la cuisine, la cuisine est une pièce localisée dans l'habitat, la cuisine est occupée par un individu,...

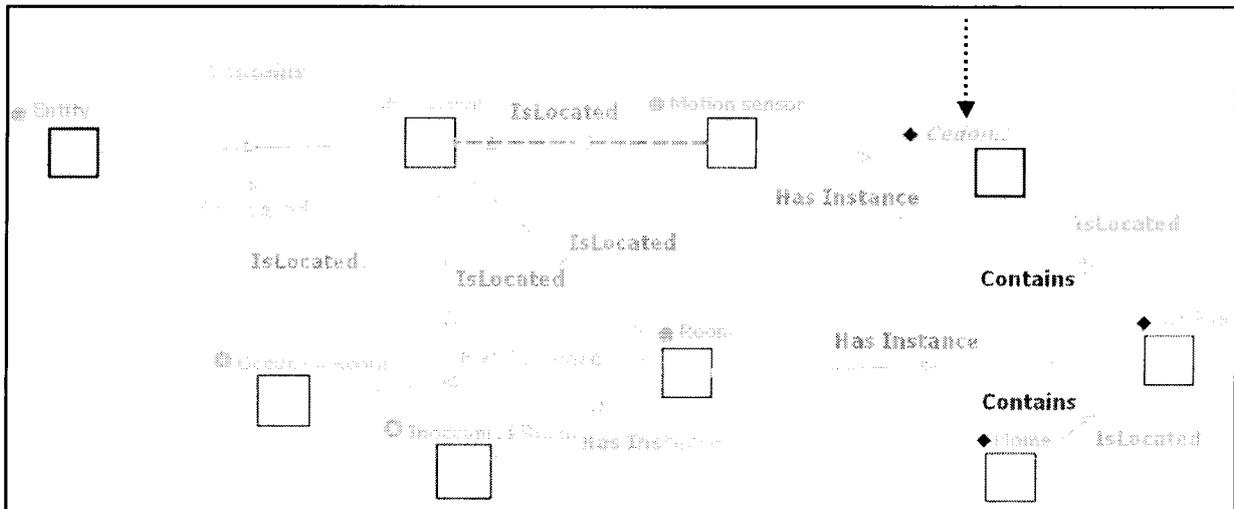


Figure 31 : Exemple de relations dans l'ontologie.

Etape 3 :

Une fois que l'ontologie est remplie grâce aux données capteurs, il est possible de déduire l'action à entreprendre à partir des règles d'inférence définies au préalable. Dans notre exemple, la règle appliquée est la suivante :

$$\text{Lampe} (?L) \wedge \text{Pièce_Occupée} (?P) \wedge \text{estLocalisé} (?L, ?P) \rightarrow \text{Lampe_Allumée} (?L)$$

Cette règle permet d'allumer la lampe située dans la zone de détection activée.

Etape 4 :

Le service ontologie observe si un changement d'état intervient dans l'ontologie et plus particulièrement au niveau des actionneurs. Lorsqu'un changement d'état est effectif, le service ontologie émet un événement vers l'actionneur concerné dans le but de mettre à jour son état physique. Dans l'exemple, il s'agit de l'état de la lampe située dans la cuisine. Si plusieurs lampes sont présentes dans la cuisine, leurs états seront modifiés.

Etape 5 :

Enfin, quand le Bundle de l'actionneur concerné est notifié de l'événement qui lui est destiné, il met à jour son état grâce à la consigne émise par le service ontologie.

4.2 Limites de l'approche

Au final, les résultats obtenus sont ceux escomptés : selon la zone dans laquelle un individu se trouve, la lumière située dans cette zone s'allume.

4.2.1 Le temps de réponse

Un inconvénient majeur réside dans le temps de réponse. Un décalage temporel important entre le moment où la personne est détectée dans une zone et la mise en route de la lampe est perceptible. Il est de l'ordre de 8 secondes avec le matériel utilisé. Le temps de réponse est un élément important à prendre en compte dans le système intégrant l'ontologie. En effet, certains services nécessitent un déclenchement rapide pour qu'ils soient efficaces : c'est le cas de la détection de contextes accidentogènes comme la chute d'une personne, le feu dans l'habitat, ...).

Néanmoins les performances actuelles de notre architecture ne permettent pas d'être aussi réactif. Nous avons constaté que la mise en œuvre de certains services peut raisonnablement être en retard de plusieurs secondes par rapport à l'événement déclencheur.

L'utilisation d'un raisonneur associé à une ontologie relativement riche et complexe est très coûteuse en ressources processeur et calcul. Les temps de calcul nécessaires à l'inférence sont non négligeables. A titre d'exemple, sur un ordinateur personnel cette durée de raisonnement peut varier de quelques secondes à quelques dizaines de secondes en fonction de la complexité de l'ontologie et des ressources disponibles (processeur, mémoire, ...).

Il est donc important de prendre en compte cet élément à la fois lors de la modélisation mais aussi lors de la définition des services que l'ontologie doit rendre à l'habitant. Ainsi, pour certains services, il n'est pas nécessaire d'utiliser le processus d'inférence. Par exemple, pour des services de contrôle/commande simples, l'utilisation du processus d'inférence ajoute une durée non négligeable (de l'ordre de plusieurs secondes) entre le moment où l'habitant appuie sur un interrupteur et le moment où la lampe correspondante s'allume. Ainsi, ne doivent être intégrés dans l'ontologie que des services qui ne seront pas affectés par un pas de temps d'exécution de l'ordre de quelques secondes.

Améliorations possibles

Afin de minimiser le temps de réponse, nous avons mis en œuvre deux améliorations. Celles-ci permettent de réduire d'environ dix pourcents les temps de calcul.

La première consiste à fractionner la méthode de consultation de l'état des individus dans l'ontologie en sous-méthodes spécialisées, dont une se concentrant uniquement sur l'état des appareils Tout-Ou-Rien (c'est-à-dire l'état allumé ou éteint). C'est en effet un des types de requête les plus présents en cours de fonctionnement.

La seconde amélioration consiste à ne fournir, lors de la consultation, que des informations sur les valeurs des propriétés, et aucune sur la classe de l'individu. En effet, l'API Java choisie ne permet de renvoyer directement la ou les classes d'un individu donné. Lorsque l'on souhaite obtenir cette information, il est nécessaire d'interroger successivement toutes les classes jusqu'à trouver l'individu concerné. Cela se traduit par un surplus important au niveau des calculs. Dans certains cas, il peut être intéressant de dupliquer ou de transformer certaines informations relatives à la classe de l'individu sous forme de propriétés. Par exemple, l'information véhiculée par les sous-classes « Lampe_allumée » et « Lampe_éteinte » peut avantageusement être transformée en une propriété « état » pouvant prendre les valeurs booléennes Vrai ou Faux.

Malgré ces améliorations, le temps de réponse reste un élément bloquant pour certains services nécessitant une réactivité inférieure à la seconde. Nous avons pu constater en pratique que ce problème se traduit en outre par une relative monopolisation des capacités de calcul de la passerelle domestique, ce qui handicape dans certains cas les autres services déployés sur la plateforme. Pour limiter ces répercussions, il est toujours possible de baisser la priorité d'accès aux ressources du service ontologie par rapport aux autres services, mais la conséquence directe est un allongement supplémentaire du temps de calcul.

Avec la disponibilité croissante et bon marché de processeurs puissants et peu gourmands en consommation, ce temps de latence ne constitue pas selon nous un obstacle majeur à l'utilisation des ontologies dans ce domaine.

Pistes pour l'amélioration

Une piste consiste à optimiser les raisonneurs existant ou en créer de plus performants en se basant notamment sur les avancées faites en parcours d'arbres dans le domaine du traitement instantané de la parole. Dans les travaux de Le Pham [PHA06], l'optimisation du raisonnement sur des ontologies est réalisée en décomposant une ontologie en plusieurs petites ontologies. Le raisonnement est ainsi effectué en parallèle sur chacune des sous-ontologies. Il semblerait que cette méthode améliore la vitesse des raisonneurs. Néanmoins, la performance du raisonnement dépend de l'algorithme de décomposition d'ontologies utilisé et par le genre de l'ontologie.

Une autre solution envisageable à plus court terme consisterait à implémenter un système à reconnaissance de formes basé sur un réseau neuronal. Ce type d'approche requiert une phase d'apprentissage pouvant être soit intégrée au fonctionnement normal de l'habitat communicant, soit préalablement à sa mise en fonctionnement chez l'habitant. En couplant ce système à reconnaissance de formes avec le système déjà implémenté (basé sur le traitement formel d'une ontologie), le premier système peut faire son apprentissage à partir des décisions du deuxième. Une fois la phase d'apprentissage effectuée, le temps de réponse dû à la reconnaissance d'une forme est suffisamment court pour être perçu comme instantané par l'utilisateur. Dans ce cadre de fonctionnement, le deuxième système peut alors être désactivé automatiquement tout en gardant la possibilité de le réactiver lorsqu'une forme n'est pas reconnue. Ainsi, le système n'est jamais bloqué par une forme qu'il ne reconnaît pas.

4.2.2 Inconsistances et ajout de règles

Dans une ontologie, le terme inconsistance désigne une incohérence dans le modèle aboutissant à une contradiction entre deux règles (à chaînage avant ou à chaînage arrière). Par conséquent lorsqu'une inconsistance apparaît, le raisonneur traitant l'ontologie se trouve dans l'incapacité de prendre une décision et se bloque. Dans le cadre de l'habitat communicant, c'est la chaîne complète de gestion des services contextuels qui se bloque, rendant impossible le rendu de service. Le point critique lors de l'élaboration de cette architecture est ainsi l'écriture des règles. Il faut veiller à ce que le modèle ne comporte pas d'inconsistance.

Il est d'autant plus difficile de détecter une inconsistance puisqu'elle n'apparaît que lorsque les deux règles qui entrent en conflit sont appliquées en même temps. Par conséquent, si une nouvelle règle conflictuelle est ajoutée dans l'ontologie, mais qu'elle ne s'applique que très rarement, le système va sembler fonctionner correctement pendant plusieurs jours ou même plusieurs semaines avant de se bloquer lorsque lesdites règles conflictuelles seront utilisées. La difficulté est alors de retrouver les règles à l'origine du problème.

Des outils permettent de vérifier en amont de tout raisonnement la consistance d'une ontologie. Le problème est qu'à l'heure actuelle ces outils ne fonctionnent qu'avec des ontologies écrites uniquement en OWL. Le langage SWRL, trop récent, n'est pas encore supporté. La conséquence directe de cette absence d'outil de vérification est que seules les personnes maîtrisant les langages OWL et SWRL et possédant une bonne connaissance de l'ontologie de l'habitat communicant, peuvent être autorisées à ajouter de nouvelles règles d'inférence dans l'ontologie. Par défaut, cela exclut l'utilisateur alors que celui-ci doit garder un sentiment de maîtrise sur son habitat.

En attendant l'apparition d'un outil prenant en compte SWRL, il est possible de contourner la difficulté en regroupant les règles par services rendus à l'habitant et en essayant de fournir dans l'ontologie une palette de services la plus large possible. L'habitant ne devrait alors pas éprouver le besoin d'ajouter de nouveaux services.

5 Conclusion

Dans ce chapitre, nous avons décrits les différentes étapes d'une architecture de gestion de contexte pour l'habitat communicant. Nous avons présenté dans un premier temps le modèle PCIA partant de la couche de perception jusqu'à la couche de mise en œuvre d'une décision. Nous avons démontré l'importance d'utiliser une approche multicouche afin d'obtenir une supervision efficace du contexte.

Nous avons également présenté les briques logicielles pour la mise en œuvre de l'architecture de gestion de contexte pour l'habitat communicant. Nous avons montré l'importance d'une approche à services pour le déploiement distant des services logiciels contextualisés. En partant de la notion de service, nous avons présenté les différents moyens logiciels de mise en œuvre du modèle PCIA au sein de cette architecture. A l'aide d'une couche événementielle, nous sommes parvenus à remonter les informations contextuelles vers un modèle d'ontologie adapté à l'habitat communicant. Une fois ce modèle peuplé, un processus d'inférence basé sur des règles du premier ordre nous a permis de déterminer l'action à entreprendre, et donc de sélectionner le service à fournir à l'habitant.

Enfin, l'architecture logicielle de gestion de contexte pour l'habitat communicant a été validée par une expérimentation complète mettant en œuvre un réseau de capteurs et d'actionneurs. Les informations véhiculées au sein de l'architecture logicielle nous ont permis d'obtenir une prise de décision correspondant au contexte observé.

Néanmoins, nous avons montré que l'architecture logicielle mise en œuvre reste relativement lourde d'un point de vue ressources, tant au niveau de la consommation mémoire que processeur. Le temps de réponse demeure un inconvénient majeur. Nous avons présenté des solutions et des pistes envisageables pour améliorer ce temps de réponse, en espérant que le matériel évoluera en ce sens. Nous avons pointé un inconvénient majeur dans l'utilisation d'une ontologie, et en particulier dans l'utilisation de règles pour le phénomène d'inférence, résidant dans l'écriture des règles. Il est nécessaire de veiller à ce que les règles n'entrent pas en contradiction. L'inconsistance entre règles peut entraîner un blocage complet du système de raisonnement et donc du rendu de service. Il faut veiller à ce que les règles soient écrites par le fournisseur de services et non par l'habitant afin d'éviter tout problème d'incohérence.

Finalement, nous avons montré que les enjeux de l'habitat communicant sont multiples : acquérir des contextes dans l'habitat afin d'en extraire une décision pertinente et contextualisée ; prendre en compte la volonté de l'habitant dans une architecture décisionnelle automatique et décrire des scénarios correspondant à un modèle de fonctionnement de l'habitat.

L'aspect négatif des règles de la logique du premier ordre est l'obligation de transformer les valeurs numériques issues des capteurs en attributs symboliques binaires [PRA04]. Cependant, les règles utilisées ici sont insuffisantes dans le cadre de l'élaboration de contexte complexe. Il faudrait pouvoir gérer les valeurs numériques et les imperfections de ces valeurs (en termes d'imprécision et d'incertitude) avec des attributs non-binaires, ce que n'autorise pas la modélisation par ontologie. Les limites de la logique du premier ordre sont rapidement atteintes. L'ontologie ne permet pas de modéliser le comportement graduel des données capteurs et leurs incertitudes. Il est donc nécessaire de déporter cette prise en charge de l'incertitude sur les données. De même, les règles du premier ordre ne permettant pas de prendre en compte l'incertitude sur leurs attributs, il est nécessaire de déporter l'inférence sur les données incertaines.

Dans le chapitre suivant, nous détaillons un formalisme permettant de prendre en compte les imperfections des données capteurs et offrant une solution au problème d'inférence à partir de processus de combinaison de données. Cette approche est développée en amont de l'ontologie afin de construire des contextes complexes.

"The world is noisy and messy"
Daphne Koller, chercheur à
Stanford.

Chapitre 3

Gestion de l'incertain et fusion de données dans l'habitat communicant : Application à la détection de défaillance

1 La fusion de données et l'incertain

1.1 Introduction

Le chapitre précédent a permis de présenter l'architecture globale pour la gestion du contexte au sein de l'habitat communicant. Nous avons vu les problèmes inhérents à la couche de perception ainsi que les difficultés qu'éprouve le langage SWRL, pour déterminer des contextes complexes. Ce troisième chapitre se propose de répondre à ces problèmes en fournissant une architecture permettant l'élaboration de contextes complexes à partir de processus de fusion de données. L'objectif recherché est de proposer une architecture permettant la prise en compte de la nature incertaine des données provenant des capteurs, difficilement réalisable à partir d'une description ontologique. Le déport des traitements complexes permet un raisonnement plus crédible, permettant de prendre en compte les imperfections des données capteurs.

La Figure 32 présente la chaîne complète que nous avons mise en œuvre. Nous détaillerons dans les paragraphes suivants les différentes étapes qui la composent. Ainsi, nous décrirons l'étape permettant d'obtenir des données symboliques à partir de données numériques en veillant à prendre en compte l'incertitude liée aux données. Puis nous présenterons l'étape permettant de fusionner ces données symboliques en prenant en compte leurs incertitudes respectives.

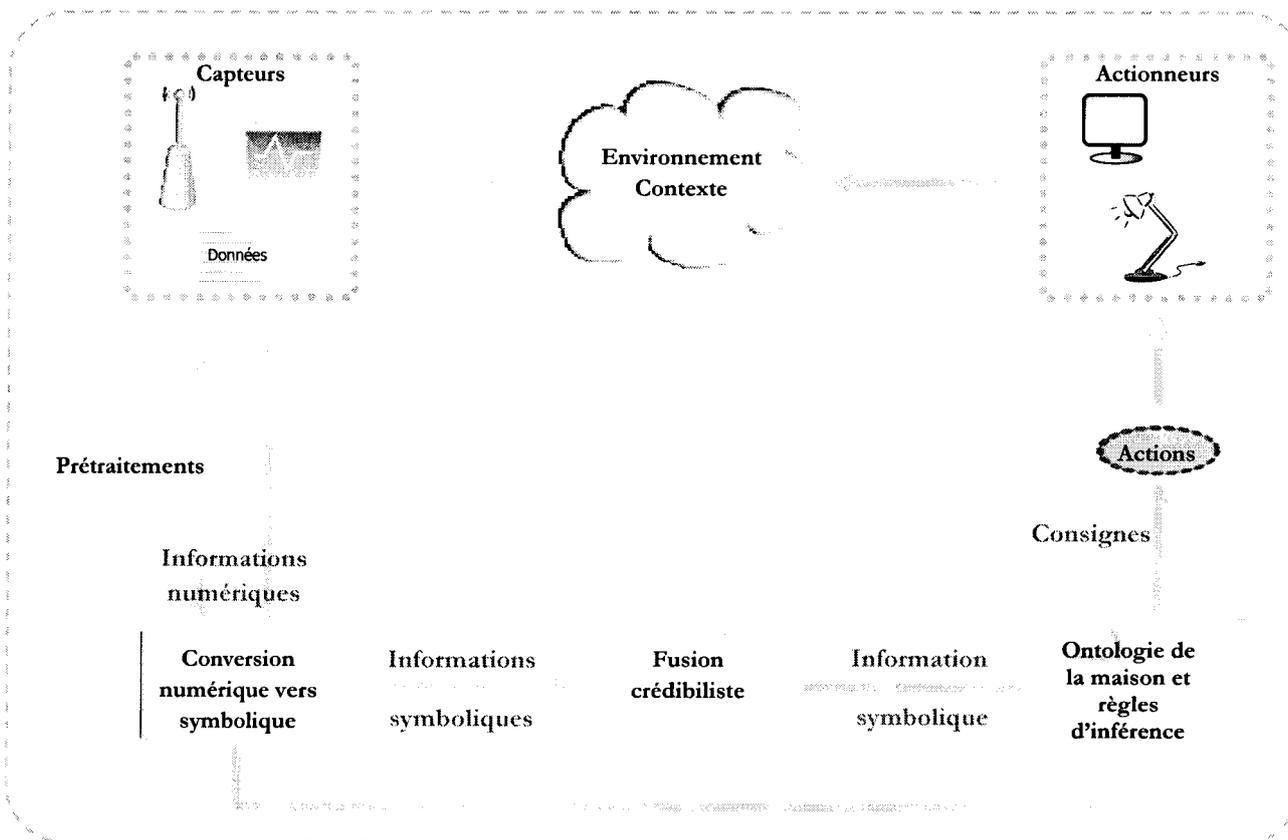


Figure 32 : Chaîne complète de la gestion du contexte.

L'obtention de données symboliques peut s'effectuer de deux façons différentes : soit à partir de capteurs physiques (un capteur de température par exemple), soit à partir de capteurs logiques (données issues de prétraitements). Dans les deux cas, l'objectif reste le même : donner à une information une signification symbolique et non plus une signification simplement numérique. Cette étape permet d'estimer l'incertitude sur les informations symboliques. L'étape suivante consiste à fusionner les informations symboliques en prenant en compte l'incertitude de chacune des informations fusionnées. Cette fusion permet d'agréger un

ensemble d'informations et fournit une information contextuelle complexe qui est ensuite injectée dans l'ontologie. La fusion d'informations redondantes et/ou complémentaires permet d'accroître la crédibilité que l'on peut avoir envers l'information résultante. La Figure 33 présente un processus de fusion complexe et évolué, le cerveau humain. Le cerveau est capable d'interpréter des informations hétérogènes et de les fusionner afin de prendre une décision face à un contexte donné.

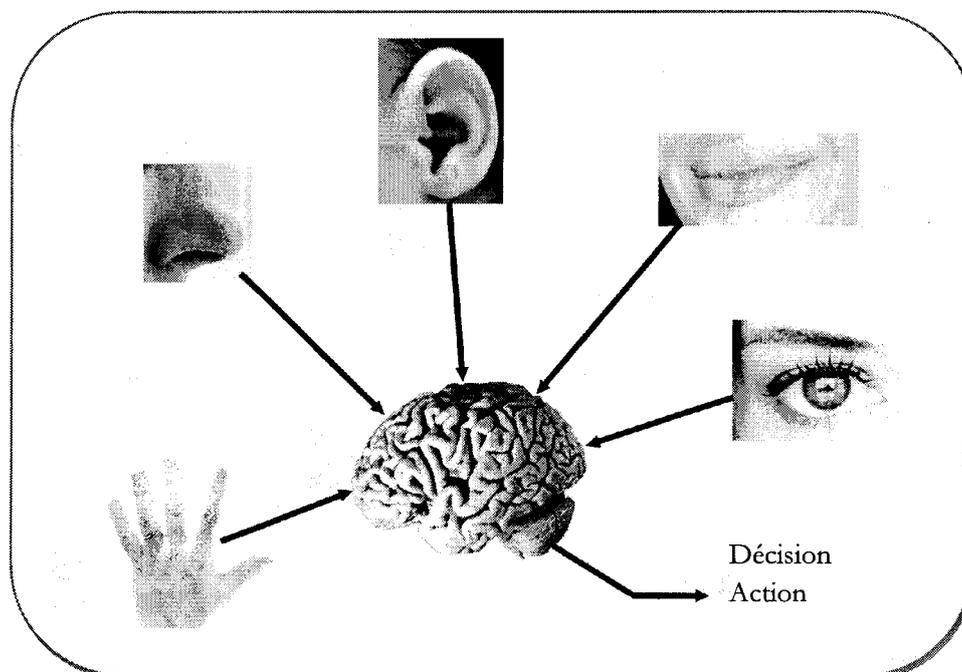


Figure 33 : Le cerveau humain et son système de perception comme un exemple de processus de fusion.

1.2 Approche crédibiliste

Notre approche de fusion de données est basée sur le Modèle de Croyance Transférables (MCT ou TBM : Transferable Belief Model) proposé par Smets [SME90] qui offre un formalisme pour la prise en compte de l'incertain et de l'ignorance et est adapté dans le cadre de l'utilisation de sources de données non fiables ou fournissant des informations partielles.

Le MCT est une variante de la théorie des fonctions de croyances, aussi connue sous le nom de théorie de l'évidence ou théorie de Dempster-Shafer. Cette théorie a été initialement introduite par Dempster puis formalisée mathématiquement par Shafer [SHA76].

Cette théorie est la généralisation de la théorie Bayésienne car elle permet de prendre en compte l'imprécision (propre à la théorie Bayésienne), l'incertitude ainsi que l'ignorance totale (non modélisées par la théorie Bayésienne).

Cette théorie permet d'évaluer la croyance que l'on a sur des propositions aussi bien simples (ex : $P1 = \langle x \text{ est } H_i \rangle$) que composites (ex : $P2 = \langle x \text{ est } H_i \text{ ou } H_j \rangle$). Cette capacité permet de représenter l'incertitude sur un événement en attribuant une masse de croyance sur le doute que l'on modélise par ' $H_i \cup H_j$ ', c'est-à-dire qu'il est impossible de se prononcer sur l'une ou l'autre hypothèse. La Figure 34 présente la différence entre ces deux formalismes. Du côté probabiliste, on peut définir une probabilité sur les hypothèses singletons 'Oui' et 'Non' tandis que le formalisme des fonctions de Croyance permet d'attribuer une masse

de croyance à tous les ensembles d'hypothèses sans se limiter aux hypothèses singletons 'Oui' et 'Non' (ici la disjonction d'hypothèse $Oui \cup Non$ est modélisée par 'Peut être').

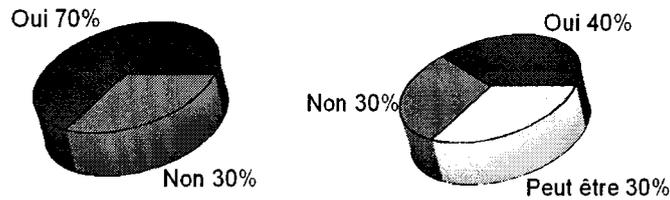


Figure 34 : La théorie des probabilités vs la théorie des fonctions de Croyance [ALM05].

La théorie des fonctions de Croyance suppose au départ la définition d'un ensemble Θ appelé cadre de discernement composé de n hypothèses dont l'une d'elles est la solution au problème posé.

$$\Theta = \{H_1, H_2, \dots, H_n\}$$

Ces n hypothèses doivent être, sous l'hypothèse d'un monde fermé :

- Exhaustives : La solution au problème est obligatoirement l'une des hypothèses du cadre de discernement.
- Exclusives : $\forall i \neq j H_i \cap H_j = \emptyset$, ce qui signifie que la solution est unique.

A partir du cadre de discernement Θ , on déduit le référentiel de définition noté 2^Θ , comprenant l'ensemble des 2^n sous-ensembles A de Θ :

$$2^\Theta = \{A / A \subseteq \Theta\} = \{\emptyset, H_1, H_2, \dots, H_n, H_1 \cup H_2, \dots, \Theta\}$$

Il comprend les hypothèses singletons de Θ , mais également l'ensemble de toutes les disjonctions possibles de ces hypothèses. Cet ensemble sert de référentiel de définition pour l'ensemble des grandeurs utilisées par la théorie des fonctions de Croyance pour évaluer la croyance d'une proposition.

La croyance sur une proposition est définie par la masse de probabilité élémentaire. Celle-ci présente une grande analogie avec la notion de distribution de probabilité à la différence que l'on peut répartir une masse parmi les éléments de 2^Θ , c'est-à-dire non seulement parmi l'ensemble des hypothèses singletons, mais aussi parmi les disjonctions d'hypothèses.

La fonction de masse élémentaire $m(\cdot)$ associée à une source est donc définie par :

$$m: 2^\Theta \rightarrow [0,1]$$

et vérifie les propriétés suivantes :

- $m(\emptyset) = 0$
- $\sum_{A \subseteq \Theta} m(A) = 1$

La masse $m(A)$ représente la croyance placée exactement sur la proposition A . Si cette proposition A est différente d'une hypothèse singleton, cette masse représente la croyance qui n'a pas pu, compte tenu de l'état des connaissances, être affectée à un sous-ensemble de A .

On remarque bien ici l'apport de la théorie des fonctions de Croyance par rapport à la théorie Bayésienne. Le fait de ne pas être obligé de répartir la masse totale sur des singletons permet d'avoir une attitude moins arbitraire. Ainsi, lorsque l'on affecte une masse à une proposition A différente d'une hypothèse singleton, on exprime le fait que l'ensemble des hypothèses de A nous paraît crédible mais sans pour autant prendre parti pour l'une d'entre elles. Cette souplesse de la méthode permet de représenter facilement l'ignorance totale par le jeu de masses suivant :

$$m(\Theta) = 1$$

et

$$m(A) = 0 \quad \forall A \neq \Theta$$

La modélisation issue de cette fonction est appelée un jeu de masses.

Les éléments focaux sont les éléments A de 2^Θ dont la masse $m(A)$ est non nulle. L'ensemble des ces éléments constitue le noyau N :

$$N = \{A \in 2^\Theta / m(A) > 0\}$$

Une structure de croyance est compatible avec un ensemble de distributions de probabilité. Les bornes minimale et maximale de la probabilité de chaque élément de 2^Θ sont données par la définition des fonctions de crédibilité et de plausibilité.

1.2.1 Fonction de crédibilité

La masse $m(A)$ représente la partie du degré de croyance placée exactement sur la proposition A . Pour prendre en compte toute la croyance placée sur cette même proposition, il faut rassembler les masses de toutes les propositions B qui impliquent A . En effet, de tels sous-ensembles B contribuent également à la croyance de A . Par exemple, la proposition « x est un rond » implique la proposition « x est un rond ou un carré ». Si la première proposition est vraie, alors la seconde l'est aussi. Ainsi, pour prendre en compte toute la croyance sur A , on calcule sa crédibilité. La fonction de Crédibilité Cr renvoie, pour toute proposition A de 2^Θ , la somme de tous les degrés de croyance placés sur des propositions impliquant A :

$$Cr(A) = \sum_{B \subseteq A} m(B)$$

$Cr(A)$, aussi notée $bel(A)$ pour *belief*, mesure à quel point les informations données par une source soutiennent la proposition A .

1.2.2 Fonction de plausibilité

La fonction de plausibilité Pl est la fonction duale de la crédibilité. Elle renvoie, pour toute proposition A de 2^Θ , la somme de tous les degrés de croyance placés sur des propositions compatibles avec A :

$$Pl: 2^\Theta \rightarrow [0,1]$$

avec

$$Pl(A) = 1 - Cr(\bar{A})$$

soit

$$Pl(A) = \sum_{A \cap B \neq \emptyset} m(B)$$

Elle mesure l'intensité avec laquelle on ne doute pas de la proposition \mathcal{A} . Autrement dit, elle mesure à quel point les informations données par une source ne contredisent pas \mathcal{A} .

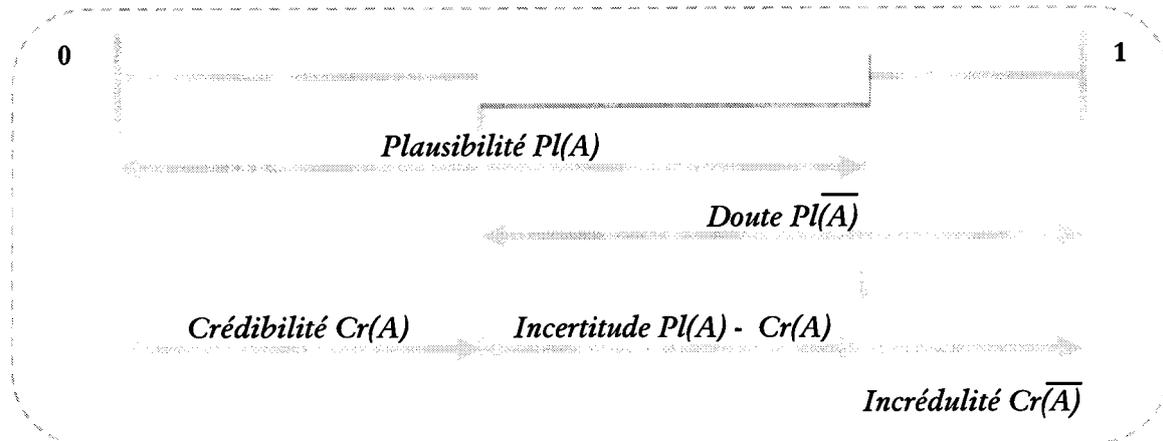


Figure 35 : Relations entre crédibilité, plausibilité et incertitude.

La Figure 35 présente les relations entre crédibilité, plausibilité et incertitude. Nous pouvons noter l'existence de la fonction symétrique à la crédibilité, c'est-à-dire le doute et l'incrédulité, fonction symétrique de la plausibilité.

Par exemple, prenons un capteur S_1 se prononce sur le cadre de discernement Θ composé de trois éléments H_1 , H_2 et H_3 . Le référentiel de définition correspondant $2^\Theta = \{\emptyset, H_1, H_2, H_3, H_1 \cup H_2, H_1 \cup H_3, H_2 \cup H_3, \Theta\}$. Le jeu de masse m_{s1} fournit par ce capteur est le suivant :

$$\begin{aligned} m_{s1}(H_1) &= 0.6 \\ m_{s1}(H_2) &= 0 \\ m_{s1}(H_3) &= 0 \\ m_{s1}(H_1 \cup H_2) &= 0.3 \\ m_{s1}(H_1 \cup H_3) &= 0 \\ m_{s1}(H_2 \cup H_3) &= 0 \\ m_{s1}(\Theta) &= 0.1 \end{aligned}$$

La crédibilité sur les hypothèses singletons de ce jeu de masse est la suivante :

$$\begin{aligned} Cr(H_1) &= m_{s1}(H_1) = 0.6 \\ Cr(H_2) &= m_{s1}(H_2) = 0 \\ Cr(H_3) &= m_{s1}(H_3) = 0 \end{aligned}$$

La plausibilité sur les hypothèses singletons de ce jeu de masse est la suivante:

$$\begin{aligned} Pl(H_1) &= m_{s1}(H_1) + m_{s1}(H_1 \cup H_2) + m_{s1}(H_1 \cup H_3) + m_{s1}(\Theta) = 0.6 + 0.3 + 0.1 = 1 \\ Pl(H_2) &= m_{s1}(H_2) + m_{s1}(H_1 \cup H_2) + m_{s1}(H_2 \cup H_3) + m_{s1}(\Theta) = 0.3 + 0.1 = 0.4 \\ Pl(H_3) &= m_{s1}(H_3) + m_{s1}(H_1 \cup H_3) + m_{s1}(H_2 \cup H_3) + m_{s1}(\Theta) = 0.1 \end{aligned}$$

1.2.3 Lien avec les probabilités

Lorsque les éléments focaux sont uniquement des singletons, nous définissons la distribution de masses comme Bayésienne. Ainsi, lorsque toute la masse de croyance est équi-répartie sur les hypothèses, nous

définissons cette distribution de masses comme masse Bayésienne uniforme. Alors, les fonctions de crédibilité et de plausibilité sont confondues avec une mesure de probabilité Pr , tel que $Cr(A)=Pl(A)=Pr(A)$

1.2.4 Assignment des fonctions de masse de croyance

Le problème de l'affectation des masses de croyances aux différentes hypothèses est un problème majeur. Néanmoins, la première étape consiste à définir le cadre de discernement qui est fortement dépendant de l'application. Le cadre de discernement peut correspondre au nombre de régions à discerner dans le cadre de la segmentation d'images, au nombre de classes pour les problèmes de discrimination,... Cette première étape est généralement établie de manière supervisée par un expert bien que certains travaux utilisent des algorithmes de classification non supervisée.

Une fois cette étape réalisée, la modélisation des fonctions de masse est un problème crucial où il n'y a pas de méthode générique. Il s'agit de choisir correctement et précisément les éléments focaux pour obtenir une modélisation riche et flexible capable d'appréhender une large gamme de situations que l'application risque de rencontrer. Selon le type d'application rencontrée, il existe différentes méthodes permettant d'élaborer les masses de croyances. La plus simple et la plus utilisée reste encore l'expertise humaine dans laquelle les coefficients sont établis manuellement. D'autres méthodes telles que l'utilisation des probabilités comme base de conversion [APP91], la notion de distance [DEN95], de vraisemblances [APP98], de taux d'erreurs ou encore des sorties de réseaux de neurones [ROG94] sont employées.

Néanmoins, il n'existe pas à l'heure actuelle de méthode garantissant une affectation optimale des masses de croyance. Les méthodes que nous venons de présenter sont acceptables à partir du moment où elles sont appliquées à un domaine spécifique et bien connu.

1.3 Fusion de données crédibiliste

L'intérêt majeur de la théorie des fonctions de Croyance en fusion de données repose sur la possibilité de construire une fonction de masse unique par combinaison de plusieurs fonctions de masses issues de plusieurs sources d'information distinctes. La loi de Dempster permet de combiner M sources d'informations définies sur le même cadre de discernement.

Néanmoins, l'utilisation de cette loi de combinaison nécessite que les sources d'information soient indépendantes. Cette notion d'indépendance est une notion qui reste ambiguë et floue. Janez [JAN96] propose une définition intéressante pour définir ce que sont des sources indépendantes. Il définit les conditions nécessaires pour pouvoir utiliser la loi de combinaison de Dempster. Dans le cas d'utilisation de sources de données, celles-ci sont décrites comme étant indépendantes si elles sont distinctes. C'est le cas par exemple de capteurs délocalisés. Lors de l'utilisation d'experts, ceux-ci seront dits distincts si chacun donne son avis sur le problème traité sans avoir connaissance de l'avis des autres.

Supposons que nous soyons en présence de M sources d'information distinctes. Chaque source S_j ($j \in [1, M]$) possède son propre jeu de masses $m_{S_j}(\cdot)$ défini sur le même cadre de discernement Θ . La loi de combinaison de Dempster consiste alors à déduire le jeu de masses unique $m(\cdot)$ résultant de la fusion de ces M jeux de masses élémentaires $m_{S_j}(\cdot)$:

$$m = m_{S_1} \oplus m_{S_2} \oplus \dots \oplus m_{S_M} \quad \text{où } \oplus \text{ désigne l'opérateur de combinaison.}$$

1.3.1 Combinaison de deux sources d'information

Soient deux sources d'information S_1 et S_2 . Soient $m_{S_1}(\cdot)$ et $m_{S_2}(\cdot)$ leurs deux jeux de masses respectifs.

Alors la masse totale affectée à une proposition A est égale à :

$$m(A) = \sum_{A_i \cap B_j = A} m_{S_1}(A_i) \cdot m_{S_2}(B_j)$$

Le problème lié à cette combinaison est qu'il peut exister des conjonctions vides d'éléments focaux ($A_i \cap B_j = \emptyset$). Or, par définition, dans l'hypothèse d'un monde fermé tel que l'oblige Dempster, la masse affectée à \emptyset doit être nulle et la somme des masses des éléments focaux doit être égale à un. Pour respecter ces deux conditions, il est nécessaire de réaliser une renormalisation sur toute la masse totale non affectée à l'ensemble vide. Par convention, on note k cette masse totale.

On a donc :

$$k = \sum_{A_i \cap B_j = \emptyset} m_{S_1}(A_i) \cdot m_{S_2}(B_j)$$

Ce coefficient traduit l'incohérence de la combinaison. Il représente le conflit existant entre les deux sources d'information.

En particulier, on obtient les valeurs suivantes :

- * $k=1$ lorsque les deux sources d'information sont totalement en conflit. Dans ce cas, la combinaison est impossible. Il est alors nécessaire de revoir le problème.
- * $k=0$ lorsque les sources sont totalement concordantes.

Le facteur de normalisation est alors :

$$K = \frac{1}{1 - k}$$

Et l'expression normalisée de la combinaison est :

$$m(A) = K \cdot \sum_{A_i \cap B_j = A} m_{S_1}(A_i) \cdot m_{S_2}(B_j)$$

Cette combinaison renforce la croyance envers les hypothèses pour lesquelles les deux sources d'information sont concordantes et affaiblit le degré de croyance envers les hypothèses pour lesquelles elles sont en conflit. En redistribuant les masses d'évidence, elle procède au recoupement des informations et conduit à une nouvelle connaissance plus précise.

La combinaison de Dempster est associative et commutative. On peut donc combiner plus de deux sources d'information en procédant à une suite de combinaisons en cascade :

$$m = m_{S_1} \oplus m_{S_2} \oplus m_{S_3} \oplus \dots \oplus m_{S_M} = (((m_{S_1} \oplus m_{S_2}) \oplus m_{S_3}) \oplus \dots) \oplus m_{S_M}$$

Une structure exprimant l'ignorance totale (c'est à dire. $m(\emptyset) = 1$) est élément neutre de la combinaison, ce qui se traduit par le fait qu'une ignorance totale ne modifie pas la compréhension du système. Une

structure de croyance ayant un seul singleton comme élément focal (c'est à dire $m(\mathcal{A}) = 1$) est l'élément absorbant de la combinaison, ce qui signifie qu'une certitude est définitivement acquise.

La combinaison de Dempster présente néanmoins plusieurs inconvénients. Tout d'abord, elle n'est pas idempotente, ce qui signifie que la combinaison d'un jeu de masses avec lui-même produit généralement un jeu de masse différent. Cette propriété présente l'inconvénient de renforcer la connaissance exprimée par une source alors qu'aucune information supplémentaire ne la soutenait. Il est donc primordial de respecter la condition d'indépendance des sources combinées que nous avons vues précédemment.

De même, la normalisation de la combinaison de Dempster masque l'aspect conflictuel des jeux de masse fusionnés. Nous présenterons par la suite des opérateurs de combinaison permettant de prendre en compte ce conflit et d'adapter la fusion finale.

1.3.2 Propriétés de la loi de combinaison de Dempster

Cette loi de combinaison possède un certain nombre de propriétés mathématiques :

◁ Commutativité

La somme orthogonale est commutative.

$$m_1 \oplus m_2 = m_2 \oplus m_1$$

⊙ Associativité

La somme orthogonale est associative.

$$m_1 \oplus (m_2 \oplus m_3) = (m_2 \oplus m_1) \oplus m_3$$

⊛ Élément neutre

Le jeu de masses traduisant l'ignorance totale est l'élément neutre de la loi de Dempster. Un jeu de masses quelconque combiné avec celui-ci reste inchangé. Ceci est fondamental pour garantir qu'une information totalement incertaine ne vienne pas "polluer" la connaissance acquise par ailleurs.

⊜ Élément absorbant

Le jeu de masses qui met toute la masse sur une hypothèse singleton H_i ($m(\{H_i\})=1$) constitue l'élément absorbant de la loi de combinaison. Sa combinaison avec un jeu de masses quelconque le laisse inchangé (sauf en cas de conflit). Ceci assure qu'une certitude est définitivement acquise et ne peut pas être remise en cause par une autre information.

⊛ Idempotence

La loi de combinaison de Dempster n'est pas idempotente.

$$m \oplus m \neq m$$

1.3.3 Affaiblissement

Lors de la fusion, il peut être intéressant de pouvoir prendre en compte la fiabilité relative des informations délivrées par les différentes sources. En effet, il est possible d'avoir des connaissances a priori à propos d'une source permettant de déterminer un coefficient de fiabilité envers cette source.

Dans le formalisme de la théorie des fonctions de Croyance, la fiabilité d'une source S_j est prise en compte au niveau du jeu de masses $m(\cdot)$ qui lui est associé.

Soit α_{S_j} ($\alpha_{S_j} \in [0,1]$) l'indice de fiabilité associé à la source d'information S_j , avec en particulier

- $\alpha_{S_j} = 1$: La source S_j est totalement fiable
- $\alpha_{S_j} = 0$: La source S_j n'est pas du tout fiable

On peut alors réaliser une opération d'affaiblissement permettant d'obtenir un nouveau jeu de masses défini par :

$$m_{s_j}(A) = \alpha_{S_j} m_{S_j}(A) + (1 - \alpha_{S_j}) m_{S_j}(\Theta)$$

L'application d'un coefficient d'affaiblissement permet d'accroître l'incertitude envers le jeu de masses considéré. Ceci permet ainsi de pondérer l'importance de la source au sein de la fusion, sans pour autant l'éliminer complètement.

Par exemple, un capteur S_1 se prononce sur le cadre de discernement Θ composé de deux éléments H_1 et H_2 . Le référentiel de définition correspondant $2^\Theta = \{\emptyset, H_1, H_2, \Theta\}$. Le jeu de masse m_{S_1} fournit par ce capteur est le suivant :

$$\begin{aligned} m_{S_1}(H_1) &= 0.9 \\ m_{S_1}(H_2) &= 0 \\ m_{S_1}(\Theta) &= 0.1 \end{aligned}$$

Hors une connaissance préalable indique que ce capteur n'est fiable qu'à 80%. Le coefficient d'affaiblissement correspond $\alpha_{S_1} = 0.8$ permet d'accroître l'incertitude envers le jeu de masse m_{S_1} :

$$\begin{aligned} m_{s_1}(H_1) &= 0.8 * 0.9 = 0.72 \\ m_{s_1}(H_2) &= 0.8 * 0 = 0 \\ m_{s_1}(\Theta) &= (1-0.8) + 0.8 * 0.1 = 0.28 \end{aligned}$$

1.3.4 Décision

La combinaison de jeux de masses permet d'aboutir à un jeu de masses représentant l'ensemble des informations incertaines disponibles. Il peut être nécessaire de prendre une décision concernant l'hypothèse la plus vraisemblable lors de problèmes de classification ou d'aide à la décision.

La prise de décision est une tâche délicate puisque la vraisemblance d'une hypothèse est basée sur un intervalle composé d'une borne minimale (la crédibilité) et une borne maximale (la plausibilité).

Idéalement, il faudrait que la crédibilité d'une hypothèse soit supérieure à la plausibilité de toute autre hypothèse. Cependant, il peut y avoir chevauchement ou inclusion de ces intervalles, rendant le choix de

l'hypothèse plus complexe à réaliser (cf. Figure 36). Les règles de décision ont donc été élaborées en se basant sur une seule des deux valeurs crédibilité ou plausibilité. Ceci entraîne une perte d'information mais permet une décision nette.

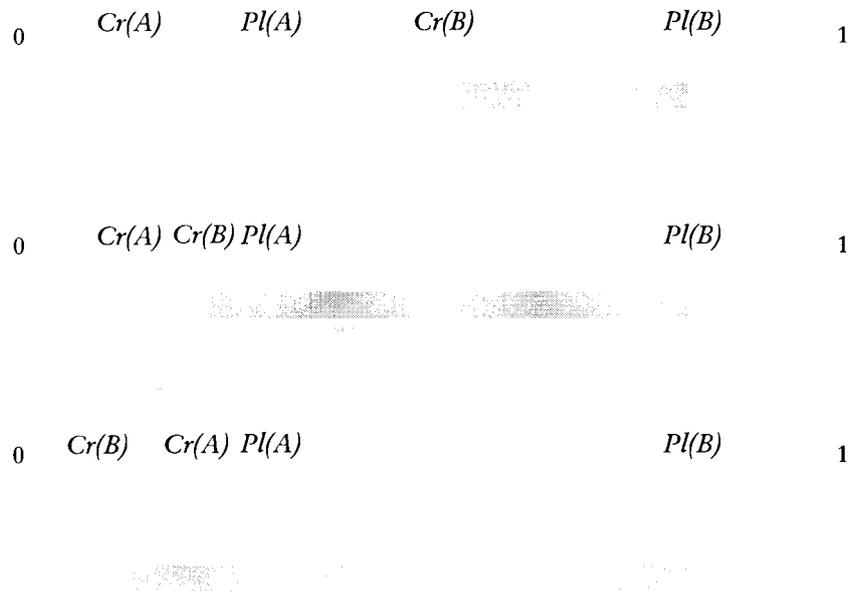


Figure 36 : Problème de chevauchement d'intervalles complexifiant la décision.

Plusieurs règles de décision ont été définies :

le maximum de plausibilité

$$\max_i [Pl(H_i)]$$

La plausibilité d'une hypothèse permet de déterminer toute la confiance que l'on aurait sur cette hypothèse si toutes les masses des propositions l'incluant étaient placées entièrement sur elle. Ce critère permet de déterminer la meilleure hypothèse en donnant le maximum de chance à chacune d'elles. C'est un critère de décision dit optimiste.

le maximum de crédibilité

$$\max_i [Cr(H_i)]$$

La crédibilité d'une hypothèse permet de déterminer toute la confiance qu'on aurait, si toutes les masses placées sur les disjonctions d'hypothèses l'incluant n'étaient pas placées sur elle mais sur les autres. Ce critère permet de déterminer la meilleure hypothèse en donnant le minimum de chance à chacune d'elles. C'est un critère de décision dit pessimiste.

le maximum de compromis : $\max_i [Cr(H_i) + Pl(H_i)]$

Cette règle réalise un compromis entre les deux règles précédentes.

Si nous reprenons le capteur S_1 précédent, se prononçant sur le cadre de discernement Θ composé de trois éléments H_1 , H_2 et H_3 avec le jeu de masse m_{s1} suivant :

$$\begin{aligned} m_{s1}(H_1) &= 0.6 \\ m_{s1}(H_2) &= 0 \\ m_{s1}(H_3) &= 0 \\ m_{s1}(H_1 \cup H_2) &= 0.3 \\ m_{s1}(H_1 \cup H_3) &= 0 \\ m_{s1}(H_2 \cup H_3) &= 0 \\ m_{s1}(\Theta) &= 0.1 \end{aligned}$$

avec

$$\begin{aligned} Cr(H_1) &= m_{s1}(H_1) = \mathbf{0.6} \\ Cr(H_2) &= m_{s1}(H_2) = 0 \\ Cr(H_3) &= m_{s1}(H_3) = 0 \end{aligned}$$

et

$$\begin{aligned} Pl(H_1) &= m_{s1}(H_1) + m_{s1}(H_1 \cup H_2) + m_{s1}(H_1 \cup H_3) + m_{s1}(\Theta) = 0.6 + 0.3 + 0.1 = \mathbf{1} \\ Pl(H_2) &= m_{s1}(H_2) + m_{s1}(H_1 \cup H_2) + m_{s1}(H_2 \cup H_3) + m_{s1}(\Theta) = 0.3 + 0.1 = 0.4 \\ Pl(H_3) &= m_{s1}(H_3) + m_{s1}(H_1 \cup H_3) + m_{s1}(H_2 \cup H_3) + m_{s1}(\Theta) = 0.1 \end{aligned}$$

La décision par maximum de plausibilité propose l'hypothèse H_1 puisque $Pl(H_1) > Pl(H_2)$ et $Pl(H_1) > Pl(H_3)$.

La décision par maximum de crédibilité propose aussi l'hypothèse H_1 puisque $Cr(H_1) > Cr(H_2)$ et $Cr(H_1) > Cr(H_3)$.

La décision par maximum de compromis propose aussi l'hypothèse H_1 puisque $Cr(H_1) + Pl(H_1) > Cr(H_2) + Pl(H_2)$ et $Cr(H_1) + Pl(H_1) > Cr(H_3) + Pl(H_3)$

1.4 Le Modèle des Croyances Transférables

Le Modèle des Croyances Transférables (MCT) a été proposé par Smets [SME90]. Le MCT est une extension des fonctions de croyance qui se distingue des approches vues précédemment de par son caractère non-probabiliste. Ce modèle offre une justification axiomatique des principaux concepts de la théorie des fonctions de Croyance. L'explication non-probabiliste de ce modèle se trouve dans sa décomposition en deux niveaux distincts : le niveau *crédal* et le niveau *pignistique* (voir Figure 14). Ce modèle est basé sur les fonctions de croyance et permet de combiner des masses de croyance fournies par des sources d'information pouvant être complémentaires, redondantes et éventuellement non-indépendantes [RAM07b]. Contrairement à une approche probabiliste, ce modèle est décomposé en deux niveaux distincts. Dans la théorie des probabilités, représentation des informations, les mécanismes de combinaison et de décision ne sont pas distingués.

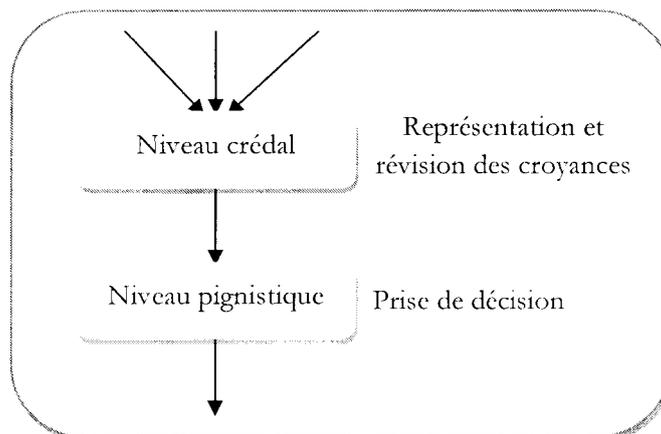


Figure 37 : Les deux niveaux du MCT.

Le premier niveau, crédal, contient la modélisation et la combinaison des fonctions de croyances. Le second niveau, pignistique, est utilisé lors de la prise de décision. Ce niveau impose la transformation des fonctions de croyance en distributions de probabilité (voir Figure 38), appelée *probabilité pignistique*.

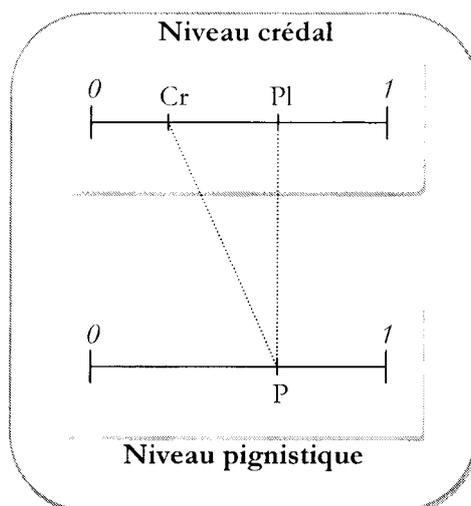


Figure 38 : Principe de la transformation pignistique.

Ainsi, le couple (crédibilité, plausibilité) est approximé par cette mesure de probabilité pignistique en équi-répartissant la masse placée sur chaque proposition, différente d'une hypothèse singleton, sur les hypothèses qui la composent [SMF05] :

$$\forall Hi \in \Theta \quad P(Hi) = \sum_{\substack{A \in 2^\Theta \\ Hi \subset A}} \frac{1}{|A|} m(A)$$

Où $|A|$ représente le cardinal de l'ensemble $A \subseteq \Theta$.

La règle décisionnelle consiste à choisir l'hypothèse singleton la plus probable :

$$\max_i [P(Hi)]$$

C'est une règle de décision "prudente".

Nous pouvons observer que cette règle de décision associe une mesure de probabilité à une fonction de Croyance. Cette transformation engendre une perte d'information qui fait perdre les avantages liés aux fonctions de Croyance.

Exemple : Le capteur S_1 précédent se prononce sur le cadre de discernement Θ composé de trois éléments H_1 , H_2 et H_3 avec le jeu de masse m_{s1} suivant :

$$\begin{aligned} m_{s1}(H_1) &= 0.6 \\ m_{s1}(H_2) &= 0 \\ m_{s1}(H_3) &= 0 \\ m_{s1}(H_1 \cup H_2) &= 0.3 \\ m_{s1}(H_1 \cup H_3) &= 0 \\ m_{s1}(H_2 \cup H_3) &= 0 \\ m_{s1}(\Theta) &= 0.1 \end{aligned}$$

La transformation pignistique propose :

$$\begin{aligned} P(H_1) &= m_{s1}(H_1) + (m_{s1}(H_1 \cup H_2) + m_{s1}(H_1 \cup H_3))/2 + m_{s1}(\Theta)/3 = 0.3 + 0.3/2 + 0.1/3 = \mathbf{0.78} \\ P(H_2) &= m_{s1}(H_2) + (m_{s1}(H_1 \cup H_2) + m_{s1}(H_2 \cup H_3))/2 + m_{s1}(\Theta)/3 = 0.3/2 + 0.1/3 = 0.18 \\ P(H_3) &= m_{s1}(H_3) + (m_{s1}(H_1 \cup H_3) + m_{s1}(H_2 \cup H_3))/2 + m_{s1}(\Theta)/3 = 0.1/3 = 0.03 \end{aligned}$$

Finalement, la décision pignistique propose l'hypothèse H_1 . Nous constatons alors que la mesure de la probabilité pignistique offre un compromis intéressant entre la crédibilité et la plausibilité.

1.5 Limite de la fusion de données

1.5.1 Le conflit

La gestion d'un conflit élevé après combinaison des sources biaise le processus de prise de décision. Les causes du conflit sont relativement diverses, on peut en distinguer trois [LEF01]:

1. La défaillance d'une source qui amène des mesures aberrantes dans la fusion. En effet, une mesure, située hors du domaine de fonctionnement nominal du capteur peut engendrer des conflits lors de la combinaison. Cette mesure aberrante peut être due :
 - * soit à un mauvais fonctionnement du capteur lors de l'acquisition de la mesure,
 - * soit à une mauvaise évaluation de la plage de fonctionnement du capteur déterminant le fonctionnement nominal du capteur. Dans le cas où le capteur fonctionne correctement, le conflit pourrait correspondre à la non prise en compte d'une hypothèse dans le cadre de discernement (classe inconnue).
2. Une modélisation non cohérente et donc non pertinente des fonctions de croyance. Les principaux modèles de détermination des jeux de masses reposent soit sur l'étude d'un voisinage soit sur l'apprentissage supervisé ou non de probabilités. Un mauvais choix de distance dans l'approche voisinage ou une mauvaise estimation des vraisemblances dans l'approche apprentissage de probabilités peut provoquer des problèmes de modélisation des fonctions de croyance qui peuvent engendrer l'apparition de conflit lors du processus de fusion. Une mauvaise modélisation peut ainsi engendrer une non exhaustivité du cadre de discernement Θ . Suivant cette hypothèse, Smets propose alors d'interpréter le conflit comme une croyance envers une nouvelle

hypothèse, non prise en compte lorsque le problème fut posé. C'est que l'on appelle travailler en « monde ouvert ».

3. Un nombre trop important de sources impliquées dans le processus de fusion. Plus le nombre de sources sera important, plus les risques de contradiction entre ces sources seront importants, et donc plus le conflit aura de chance d'apparaître.

Ces trois causes montrent l'importance de la gestion du conflit lors du processus de fusion. Comme il est impossible de résoudre simultanément l'ensemble de ces différents problèmes, il est primordial d'isoler chacune des causes et de les traiter séparément. Pour le premier cas, la défaillance d'une source, il est possible de détecter cette défaillance et de pondérer son influence lors du processus de fusion ou de simplement la retirer du processus de fusion. Pour le second cas, une modélisation non cohérente, il est possible de revoir la méthode de modélisation à partir d'une étude plus poussée du problème. Pour le troisième cas, un nombre trop important de sources, il serait peut être judicieux de revoir le problème en réduisant le nombre de sources impliquées dans le processus de fusion selon des critères de fiabilité par exemple.

1.5.2 Résoudre le problème du conflit

Plusieurs pistes ont été explorées pour résoudre le problème du conflit. La littérature en propose deux principales. La première famille regroupe des règles de combinaison partant du principe que les sources à fusionner sont fiables. Cette famille s'appuie sur l'utilisation d'opérateurs de combinaison conjonctifs. La seconde famille considère la non fiabilité partielle des sources. Cette famille s'appuie sur les opérateurs de combinaison conjonctive et disjonctive.

Les règles de combinaison

Lorsque du conflit apparaît au niveau du processus de fusion en présence de sources fiables, celui-ci ne peut être provoqué que par un problème mal posé. La non prise en compte d'une ou de plusieurs hypothèses dans le cadre de discernement est alors l'unique cause de conflit. Les opérateurs de Dempster [DEM67] et Smets [SME90] partent de ce postulat. Smets propose de ne pas redistribuer la masse conflictuelle K sur l'ensemble des propositions mais uniquement sur l'ensemble vide \emptyset . Yager [YAG83] propose une approche quasi-similaire dans laquelle il introduit une nouvelle hypothèse dans le cadre de discernement. Le conflit issu de la fusion est alors reporté sur cette nouvelle hypothèse sans entacher les hypothèses initiales du cadre de discernement.

Le conflit peut être généré par un défaut de fiabilité d'une partie des sources d'information. Cet argument a été repris dans le cadre des opérateurs présentés par Yager [YAG87] et par Dubois et Prade [DUB88]. Rombaut [ROM01] proposent d'attribuer la masse conflictuelle K à l'ensemble Θ . La répartition de la masse conflictuelle proposée par Dubois et Prade est plus précise que l'approche de Yager. A la différence de l'opérateur de Yager, le conflit n'est pas géré de manière globale mais partiellement (notion de masse conflictuelle partielle). Les masses de croyance engendrant le conflit permettent de calculer la redistribution de la masse conflictuelle. Dans le cadre de la fusion de deux capteurs S_1 et S_2 , S_1 soutient la proposition B avec une masse croyance $m_1(B)$ et S_2 soutient la proposition C avec une masse $m_2(C)$. Lorsque les deux capteurs sont en contradictions et ne sachant pas quel capteur est fiable, l'opérateur de Dubois et Prade redistribue la masse associée à cette contradiction, soit $m_1(B).m_2(C)$, sur l'union des propositions $B \cup C$.

Une approche différente permettant de gérer la combinaison de sources non fiables s'appuie sur la définition de coefficients d'affaiblissement (présentés §1.3.3). Lorsque la source S_i est totalement fiable,

l'information apportée par cette source ne devrait pas engendrer de conflit lors de la combinaison. Le coefficient α_i correspondant est alors égal à 1 et la fonction de Croyance n'est pas modifiée. Au contraire, si la source S_i est non fiable, lors de la combinaison avec d'autres sources celle-ci peut alors engendrer une information conflictuelle. En introduisant un coefficient $\alpha_i = 0$, la fonction de Croyance m_i associée à la source S_i devient alors une fonction de Croyance d'ignorance totale $m_i(\Theta) = 1$ et donc élément neutre pour la combinaison. Ainsi l'intérêt de l'affaiblissement est de pondérer l'influence des sources selon leur fiabilité respectives avant le processus de combinaison. Il est fréquemment nécessaire d'ajouter une information supplémentaire afin de déterminer précisément les coefficients d'affaiblissement. Dans certains cas, l'affaiblissement est calculé en fonction de la confiance que l'on a sur l'apprentissage réalisé pour déterminer les hypothèses. Dans [MER08] les coefficients d'affaiblissement sont calculés en se basant sur des informations contextuelles. La définition des coefficients reste cependant complexe et intuitive.

Redistribution du conflit

Plus récemment, des propositions de redistribution du conflit plus fine et plus adaptée ont été proposées. *Lefevre* propose un formalisme unificateur des règles de fusion. Il s'agit de redistribuer le conflit sur les propositions de façon moins arbitraire [LEF01]. A partir de poids w associés à chaque sous-ensemble $A \subseteq \Theta$, il propose de redistribuer le conflit partiel proportionnellement à ces poids parmi les sous-ensembles qui l'ont engendré. Le calcul des poids est basé sur les masses des sous-ensembles impliqués dans l'apparition du conflit partiel. Plusieurs méthodes d'obtention des poids ont été proposées. Nous pouvons citer l'apprentissage des poids par minimisation d'une fonction d'erreur sur des données d'apprentissage. Cette fonction d'erreur est calculée à l'aide de l'erreur quadratique moyenne entre la probabilité pignistique et l'indicateur d'appartenance à chaque hypothèse.

L'opérateur *minC* (minimal contradiction/conflict combination) a été proposé par *Daniel et Josang* [DAN00] [JØS03]. L'objectif de cet opérateur est d'identifier les différents types de conflits partiels pour ensuite les transférer selon une redistribution proportionnelle. L'opérateur est historiquement le premier opérateur mettant en œuvre une redistribution proportionnelle du conflit.

Ces différentes approches montrent qu'il n'y a pas une méthode de redistribution du conflit optimale et générale mais au contraire des méthodes adaptées à un contexte de fusion particulier. La littérature fait ressortir l'analyse suivante :

- * L'opérateur de *Dempster* est préconisé lorsque les sources fusionnées ne sont pas ou peu conflictuelles.
- * L'opérateur de *Dempster non normalisé* ou opérateur de *Smets* est préconisé lorsque l'hypothèse du monde ouvert est utilisée. Ainsi, le cadre de discernement est non-exhaustif et il existe potentiellement des hypothèses non définies.
- * L'opérateur de *Dubois et Prade* est préconisé lorsque les sources fusionnées sont conflictuelles. Il sera préféré à l'opérateur de *Yager* car plus précis dans la redistribution du conflit. Par contre, un inconvénient majeur avec cet opérateur réside dans le fait qu'il gère mal l'ignorance totale $m(\Theta) = 1$.
- * L'opérateur de *Lefevre* a le défaut d'être non associatif. Il est utilisé lorsque les sources fusionnées sont fortement conflictuelles.
- * L'opérateur *minC* permet une gestion efficace et intuitive du conflit entre les sources fusionnées. Il souffre d'un problème d'explosion combinatoire lié à la taille du cadre de discernement. Il est donc préconisé sur des problèmes de fusion dont le cadre de discernement est restreint.

Ainsi, à chaque problème, un opérateur est adapté. Le plus souvent, le choix d'un opérateur de fusion se fait par expertise préalable.

Les règles de redistribution proportionnelle du conflit semblent prometteuses. Dans la continuité de l'opérateur minC, les règles PCR (*Proportional Conflict Redistribution*) permettent cette redistribution rigoureuse du conflit. Ces règles ont été proposées par *Dezert et Smarandache* [SMA05] et sont au nombre de 5 (PCR1 -> PCR5), évoluant progressivement en terme de complexité et de sophistication. Dans [SMA06] sont présentées de manière complète ces règles PCR.

Ici, nous présentons uniquement la règle la plus évoluée et la plus efficace, c'est-à-dire la règle numéro 5 nommée PCR5. L'objectif de cette règle est de redistribuer le conflit partiel créé lors de la combinaison de deux fonctions de masse seulement aux éléments focaux qui sont à la base de ce conflit partiel.

$$m_{PCR5} = m_{12}(A) + \sum_{A \cap B = \emptyset} \left[\frac{m_1(A)^2 m_2(B)}{m_1(A) + m_2(B)} + \frac{m_2(A)^2 m_1(B)}{m_2(A) + m_1(B)} \right]$$

$$m_{PCR5}(\emptyset) = 0$$

Une version plus intuitive de la règle PCR5, nommée PCR6, a été récemment proposée par *Martin et Oswald* [MAR06]. PCR6 généralise PCR5 à N sources. Le comportement de PCR6 est identique à PCR5 en présence de deux sources mais diffère en présence de plus de deux sources.

Dans la lignée de leurs travaux sur les règles PCR, *Dezert et Smarandache* ont proposé leur propre théorie. Cette récente contribution théorique est la *théorie de Dezert-Smarandache* [DEZ02] (connue aussi sous le nom de *DSmT*). Cette théorie du *raisonnement plausible et paradoxale* étend la théorie des fonctions de Croyance en permettant la combinaison formelle de n'importe quel type d'information (certain, incertain, paradoxal) avec un nouvel opérateur de fusion. La *DSmT* est appliquée à la fusion d'informations, en particulier là où la règle de fusion de *Dempster* devient inappropriée surtout lors de l'apparition de forts conflits entre les sources fusionnées (appelés *paradoxes* dans le cadre de *DSmT*).

Contrairement à la théorie des fonctions de Croyance, la *DSmT* intègre dès la définition du cadre de discernement, la nature potentiellement conflictuelle des hypothèses. Le référentiel de définition 2^Θ est étendu à l'expression d'un «*hyper-power set*» noté D^Θ de Θ . Cet ensemble regroupe tous les éléments construits à partir des éléments de Θ avec les opérateurs ensemblistes classiques de conjonction \cap et de disjonction \cup . Ainsi, pour un cadre de discernement composé de deux hypothèses A et B, l'hyper power set est défini comme suit :

$$\Theta = \{A, B\}$$

$$D^\Theta = \{\emptyset, A, B, A \cup B, A \cap B\} \text{ et } |D^\Theta| = 5$$

Cette théorie fournit le substrat théorique essentiel pour modéliser et fusionner des jeux de masses dits généralisés. Ainsi, nous retrouvons les fonctions de croyance et de plausibilité généralisées définies à partir d'un jeu de masse généralisé définie sur D^Θ , la transformation pignistique généralisée permettant de construire une mesure de probabilité subjective à partir du jeu de masse généralisé. La combinaison de jeux de masses généralisés m_1 et m_2 s'obtient par la *règle de fusion de Dezert-Smarandache* suivante :

$$\forall C \in D^\Theta, m(C) = [m_1 \oplus m_2](C) = \sum_{A, B \in D^\Theta, A \cap B = C} m_1(A) m_2(B)$$

Comme pour la théorie des fonctions de Croyance, cette règle de fusion conserve les propriétés d'associativité et de commutativité. Elle est adaptée à la fusion de sources de natures différentes et en particulier pour la fusion de sources hautement conflictuelles.

Peu de travaux exploitent à l'heure actuelle cette théorie. Nous pouvons noter les travaux de S. Corgne [NED03] qui utilise la DSmT dans le cadre de la résolution de la prédiction de la couverture agricole en période hivernale à partir de données satellitaires. La question du problème d'explosion combinatoire peut se poser. En effet, l'augmentation de la taille du cadre de discernement augmente proportionnellement la charge de calculs ainsi que l'espace mémoire nécessaire.

Un aspect peu pris en compte est la variation dynamique du conflit entre sources. Récemment, Delmotte [DEL07] a proposé, dans le cadre de la théorie des possibilités, de mettre à jour les coefficients d'affaiblissement en fonction du conflit produits par les sources. Le postulat est que certaines sources peuvent devenir déficientes après un certain temps, même si au départ elles ont été considérées fiables. Ainsi, une estimation dynamique des coefficients d'affaiblissement des sources est réalisée. Dans le cadre de l'habitat communicant, les capteurs sont initialement choisis selon le critère de fiabilité et de bon fonctionnement. Néanmoins, l'état de fonctionnement des capteurs peut évoluer dans le temps (usure du capteur, de l'électronique interne, des liaisons de communications). Nous pensons qu'il est primordial d'effectuer régulièrement un diagnostic des sources utilisées. La section suivante traite de ce problème dans le cadre de la théorie des fonctions de Croyance.

2 La fusion de données crédibiliste et l'habitat communicant

2.1 Intégration de la fusion de données dans notre architecture de gestion de contexte

La fusion de données au sens crédibiliste dans le cadre des applications sensibles au contexte est peu utilisée. En effet, les travaux pionniers de Huadong Wu [WU02][WU03] sont vraisemblablement les seuls dans ce domaine. Notre approche se distingue de cette dernière par la prise en charge du conflit au niveau des processus de fusion grâce à l'utilisation du Modèle des Croyances Transférables [RIC08a]. L'approche que nous retenons est décrite sur la Figure 39.

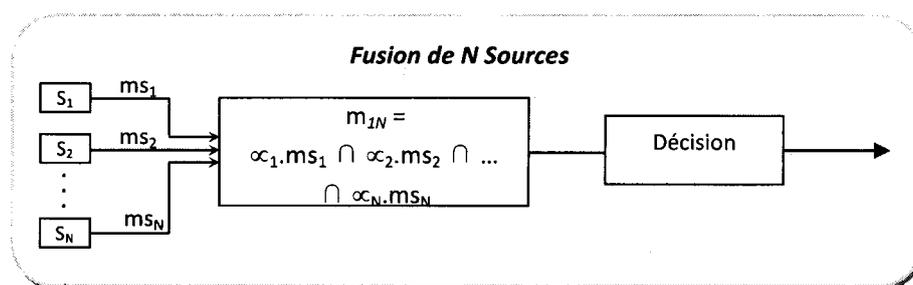


Figure 39 : La fusion statique de N sources.

On distingue deux types de fusion applicable dans le cadre d'une architecture de perception de contexte : la fusion statique et la fusion dynamique.

La *fusion statique* correspond à la fusion de données disponibles simultanément telles que peuvent l'offrir un ensemble d'experts. Le résultat de la combinaison est obtenu indépendamment des états antérieurs (les données à l'instant courant t sont uniquement employées). La fusion multicapteurs dans l'habitat

communicant est un processus de fusion statique permettant d'agréger un ensemble de données disponibles à un instant t .

La *fusion dynamique* correspond à une combinaison tenant compte des états antérieurs. Les processus de fusion itératifs sont alors des processus de fusion dynamique puisque les états antérieurs sont utilisés lors de la fusion à un instant t . Le filtrage de Kalman est un processus de fusion dynamique.

Le contexte étant par définition hautement dynamique, la détection et l'identification d'activités humaines s'établissent progressivement dans le temps [RAM07a]. Ainsi, le résultat d'un processus de fusion dynamique devient une des entrées d'un système de fusion statique. En effet, un processus de fusion dynamique a pour objectif de renforcer la connaissance envers une donnée contextuelle, c'est le cas pour la localisation d'un individu à partir d'un filtre de Kalman par exemple.

2.2 Les fonctions de Croyance et l'informatique embarquée

Une des préoccupations principales de l'informatique ambiante, applicable à la problématique de l'habitat communicant, est l'utilisation de matériels informatique embarqués et dissimulés, c'est-à-dire de faibles empreintes. Cela implique que les ressources disponibles sont restreintes. Il est alors important que les calculs effectués soient légers et nécessitent peu de ressources matérielles. Paradoxalement, la théorie des fonctions de Croyance est souvent critiquée par rapport à son aspect fortement combinatoire due à la combinaison de Dempster. Le temps de calcul d'une combinaison entre deux fonctions de masses est exponentiellement proportionnel à la taille du cadre de discernement et en particulier aux nombres d'éléments focaux que les fonctions de masse possèdent. Des travaux portant sur la diminution du temps de calcul ont été proposés en approximant les fonctions de masses par d'autres fonctions en allégeant le cadre de discernement utilisé. Il apparaît que la transformée rapide de Möbius est une solution d'approximation proportionnelle à la taille du cadre de discernement [DEN02] [TIO89][KEN90] ce qui peut alléger et donc accélérer les calculs sur des matériels embarqués.

Une approche inverse revient à considérer l'évolution rapide de l'électronique et principalement des circuits programmables. La miniaturisation et les progrès sur les matériels électroniques laissent penser à un futur proche où les microprocesseurs, microcontrôleurs seront pourvus de capacités importantes en terme de mémoire et de traitement. Les réseaux de neurones de type multi-couches ont déjà été implantés sur des circuits logiques programmables de type *FPGA* (field-programmable gate array, réseau de portes programmables in-situ) ou *ASIC* (Application-Specific Integrated Circuit, circuit intégré spécialisé), ce qui paraissait impensable il y a une vingtaine d'années. Ainsi, on peut imaginer l'arrivée sur le marché de circuits programmables de hautes performances capables d'effectuer des combinaisons complexes de fonctions de masse au sens de la théorie des fonctions de Croyance.

Néanmoins, malgré les progrès considérables en termes de matériels et de performances, les matériels utilisés sont potentiellement défaillants. Il existe différents niveaux potentiellement défaillants au sein de notre architecture :

- ◊ Défaillance de la passerelle domestique,
- * Défaillance du matériel réseau,
- * Défaillance des objets communicants,
- * Défaillance des capteurs.

Dans le cas qui nous incombe, nous avons vu que les données capteurs sont primordiales pour l'obtention de données contextuelles. Nous allons nous pencher sur la détection de défaillance capteur.

Une défaillance capteur se définit par une déviation du comportement de celui-ci par rapport à son comportement nominal. Toute déviation par rapport à cette plage nominale pourra être interprétée comme une défaillance du capteur.

Nous pouvons ainsi discerner deux cas distincts. Le premier cas porte sur des mesures aberrantes fournies par le capteur. Ces mesures ne reflètent pas du phénomène effectivement observé. Par exemple, un capteur de température biaisé par un phénomène externe fournit une température de 10°C alors que la température réelle est de 25°C. Dans ce cas précis, il est nécessaire d'observer la mesure à un instant précis afin de détecter cette défaillance.

Le second cas porte sur l'interprétation des mesures du capteur. Dans ce cas, les mesures sont correctes, mais le comportement du capteur est erratique. Par exemple, un capteur de température passe de la température 10°C à 35°C en une acquisition. Les mesures sont situées dans la plage nominale de fonctionnement du capteur, néanmoins le comportement du phénomène observé est erratique si on le compare au modèle de comportement nominal de ce capteur. Dans ce cas précis, il est nécessaire d'observer temporellement les mesures capteurs afin d'observer une dérive du comportement du capteur.

Dans la suite de ce chapitre, la section 3 présente une première méthode pour détecter et identifier un capteur défaillant au sein d'un réseau de capteurs. Dans la section 4 de ce chapitre, nous présentons une méthode permettant de détecter ces dérives de comportement capteur à partir d'un modèle de comportement nominal.

3 Détection et identification d'un capteur défaillant dans un réseau de capteurs

3.1 Introduction

Dans cette section nous présentons une approche crédibiliste de détection de défaillance capteur au sein d'un réseau de capteurs. L'objectif de ce réseau de capteurs est de fournir des informations contextuelles simples potentiellement fusionnables au sein d'une architecture de fusion crédibiliste.

La variation dynamique du conflit entre sources est peu prise en compte lors des processus de fusion. Comme le propose Delmotte [DEL07], notre proposition sera basée sur une évaluation en continu de l'état de fonctionnement des sources à fusionner. Divers phénomènes peuvent entacher ce bon fonctionnement :

- Tout type d'usure : Usure du capteur, de l'électronique interne, des liaisons de communications,...
- Un phénomène imprévu perturbant le phénomène observé,
- Un arrêt complet du capteur.

Si une défaillance capteur intervient, le processus de fusion ne doit pas s'écrouler. C'est dans cet objectif que nous présentons ici une méthode d'analyse temporelle de l'évolution de l'état de fonctionnement des capteurs [RIC07b] [RIC07c]. Cette méthode se place dans le cadre de la fusion de données multisources et plus particulièrement dans le cadre de la théorie des fonctions de Croyance. Ainsi, il nous sera possible d'analyser l'accord entre les capteurs et de pondérer l'influence du capteur défaillant.

Nous traiterons trois étapes :

- La détection d'un problème capteur,
- L'identification du capteur fautif,
- La pondération du capteur fautif.

3.2 Détection d'un capteur défaillant

3.2.1 Présentation

La Figure 40 présente notre architecture de détection de défaillance capteur au sein d'un processus de fusion multicapteurs. Elle est composée de quatre parties distinctes :

- Les N capteurs utilisés (S_1 à S_N) fournissant une fonction de masse respective ms_1 à ms_N ,
- La combinaison des masses grâce à l'opérateur de Smets,
- La prise de décision,
- La supervision.

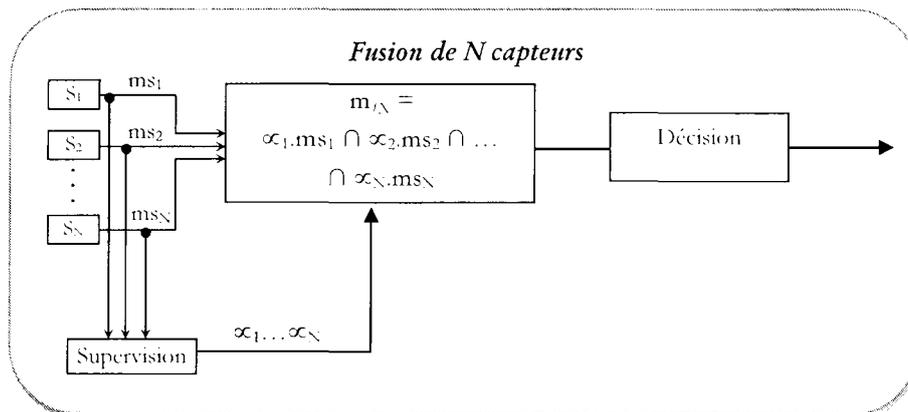


Figure 40 : Architecture de détection de défaillance capteur.

La partie supervision est en charge de détecter un capteur défaillant et de l'identifier. La Figure 41 présente l'intégration de notre algorithme de diagnostic de source défectueuse dans l'architecture générale.

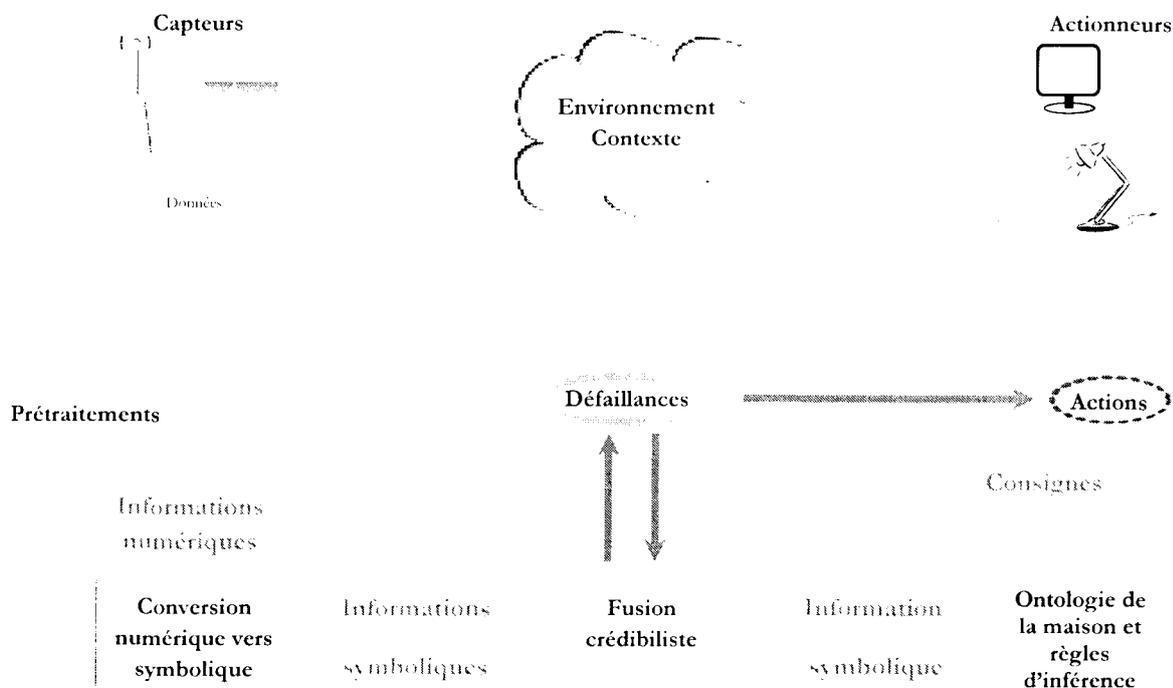


Figure 41 : Chaîne complète de la gestion du contexte avec prise en charge de la défaillance capteur.

Afin de détecter les éventuelles défaillances (panne franche d'un capteur, panne légère d'un capteur engendrant des données fausses, résultats issus de traitements faux), l'évolution du conflit (au sens du MCT) est prise en compte entre les différents capteurs qui sont fusionnés. Pour cela, nous fusionnons deux à deux les capteurs et observons les conflits issus de ces fusions. En recoupant les conflits obtenus, il est alors possible de détecter et de déterminer précisément le capteur en faute.

Après reconnaissance du capteur défaillant, il est judicieux de pondérer son influence sur la décision finale. Pour cela, nous procédons à un affaiblissement de la croyance attribuée à ce capteur en associant un indice de fiabilité $\alpha_j \in [0,1]$ à chacun des capteurs, avec en particulier :

- Si $\alpha_j = 1$ alors le capteur S_j est totalement fiable,
- Si $\alpha_j = 0$ alors le capteur S_j est défaillant.

Les coefficients a_1, a_2, a_3 ajoutés au processus de combinaison permettent d'affaiblir le jeu de masses des sources S_1, S_2 et S_3 et d'accroître leur incertitude sans toutefois les éliminer complètement [JAN96].

3.2.2 Méthode

Notre méthode repose sur l'analyse temporelle du conflit induit par la fusion de N sources. Elle consiste à pointer les combinaisons partielles conflictuelles mettant en jeu la source défaillante. Ainsi, notre méthode permet d'affiner la gestion du conflit au sein d'un processus de fusion en déterminant le capteur à l'origine de ce conflit. La méthode utilisée pour détecter la défaillance d'un capteur parmi N ($N > 2$) repose sur la combinaison 2 à 2 des capteurs entre eux. En effet, cette combinaison permet de mettre en avant le conflit que peut engendrer un capteur lorsqu'il est fusionné avec les autres capteurs.

La première étape consiste fusionner 2 à 2 les jeux de masses des N capteurs. L'opérateur de Smets est utilisé et le conflit résultant de la fusion est isolé afin d'être analysé et de déterminer s'il est significatif ou non. Nous obtenons alors I_n^o signaux de conflit.

Par exemple,

- pour N=3 capteurs, nous obtenons I_3^o 3 signaux de conflit,
- pour N=4 capteurs, nous obtenons I_4^o 2 signaux de conflit,
- ...

Considérons m_{S_i} la fonction de masse fournie par le capteur S_i et m_{S_j} la fonction de masse fournie par le capteur S_j avec $i, j = 1 \dots N$ et $i \neq j$. La fusion 2 à 2 des jeux de masses des N capteurs est assurée par :

$$k_{ij} = \sum_{\substack{A \cap B = \emptyset \\ j=1..N, i \neq j}} m_{S_i}(A) \cdot m_{S_j}(B) \quad \text{with } i=1..N$$

$K_{ij} \in [0,1]$ représente le mesure de conflit entre les capteurs S_i et S_j . Plus cette valeur est faible, plus les capteurs sont en accords, plus cette valeur est forte, plus les sources sont en désaccord. Cette formule permet de fusionner 2 à 2 les jeux de masses des N capteurs et offre une valeur de conflit K_{ij} représentative de l'accord entre chacun des capteurs.

La seconde étape concerne l'analyse des signaux de conflit résultant de la fusion des N sources. L'analyse du conflit est basée sur l'utilisation de deux experts par conflit. Le premier expert se prononce sur la valeur du conflit. Le second expert se prononce sur la durée du conflit. En effet, si un capteur connaît une

défaillance, il est logique que celle-ci provoque une montée du conflit d'une durée proportionnelle à la durée de la défaillance de la source.

Or un capteur défaillant pendant une courte durée peut être négligé si son fonctionnement normal reprend après un court instant. Cette défaillance ponctuelle peut avoir comme origine un bruit extérieur, un problème inhérent aux prétraitements, une mauvaise modélisation d'une fonction de masses, un changement d'état brusque ou encore une désynchronisation entre les capteurs. Aucun de ces cas ne correspond à une défaillance qui est à prendre en compte. Ainsi, nous faisons l'hypothèse que ces conflits peuvent ainsi être négligés car ils ne sont pas assez significatifs en termes de durée.

Pour les négliger automatiquement, ces signaux sont filtrés dans le but de faire disparaître les conflits de courte durée à l'aide d'un filtre passe-bas. Ce filtre permet d'obtenir un pic atténué dont la hauteur est proportionnelle à la durée du pic de conflit.

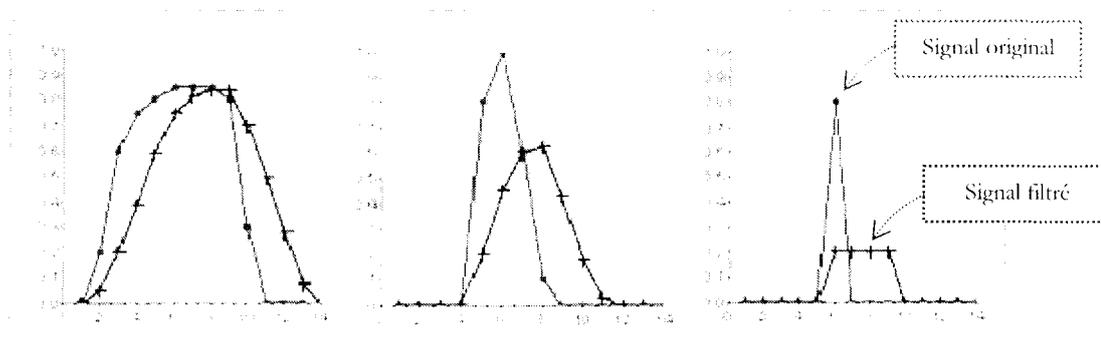


Figure 42 : Exemples de signaux de conflits et résultat du filtrage passe-bas.

La Figure 42 présente trois signaux auxquels est appliqué un filtre passe-bas. Lorsque l'on applique ce filtre sur un signal dont la durée à un état haut est longue (fréquence basse, premier cas, signal rouge), la convolution entre ce signal avec le filtre passe-bas fournit un signal dont l'amplitude est importante (premier cas, signal bleu). Si l'on applique ce même filtre à un signal dont la durée à un état haut est faible (fréquence haute, troisième cas, signal rouge), on observe que le signal résultat est de faible amplitude (troisième cas, signal bleu).

Ainsi, le second expert s'applique sur le conflit auquel est appliqué un filtre passe-bas.

Les experts se prononcent sur le caractère significatif du conflit. Le cadre de discernement utilisé $\Theta_{\text{fonctionnement}}$ est composé de deux éléments *OUI* et *NON*, correspondants respectivement aux assertions « *OUI, le fonctionnement est correct* » et « *NON, le fonctionnement n'est pas correct* »

Donc $\Theta_{\text{fonctionnement}} = \{OUI, NON\}$

Pour chacun de ces deux experts, nous avons déterminé expérimentalement les fonctions de masses $m_{1:1i}$ et $m_{1:2i}$ suivantes :

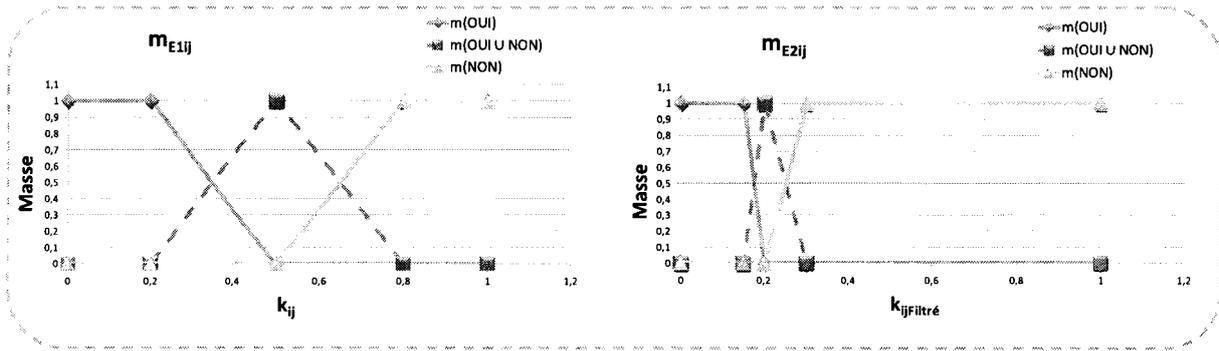


Figure 43 : Fonctions de masses m_{E1ij} et m_{E2ij} des deux experts en charge de l'analyse du signal de conflit.

Avec

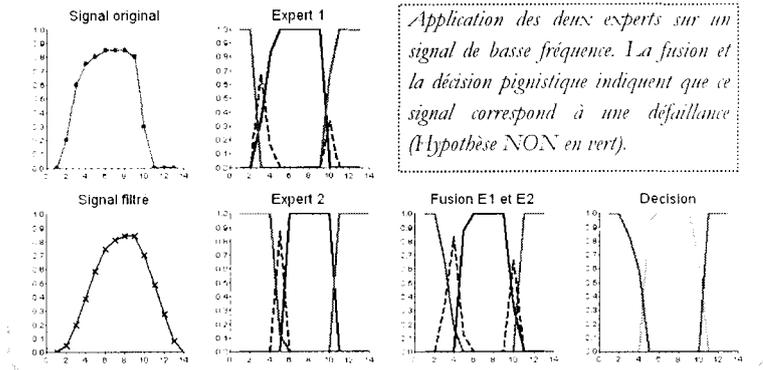
$$m_{E1ij} = f_1(k_{ij})$$

$$m_{E2ij} = f_2(k_{ijFiltré}) \text{ avec } i,j=1 \dots N \text{ et } i \neq j.$$

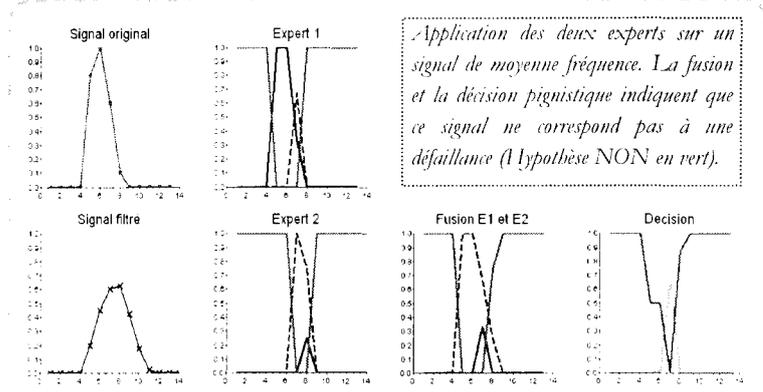
La combinaison de Smets est utilisée pour fusionner ces deux experts. Ainsi, en sortie de fusion, nous obtenons un jeu de masses définissant s'il y a un problème entre deux capteurs. Cela permet de définir si le conflit entre deux capteurs est significatif ou pas.

Si nous appliquons ces fonctions de masses aux signaux vus Figure 42, nous obtenons les résultats présentés sur les figures suivantes.

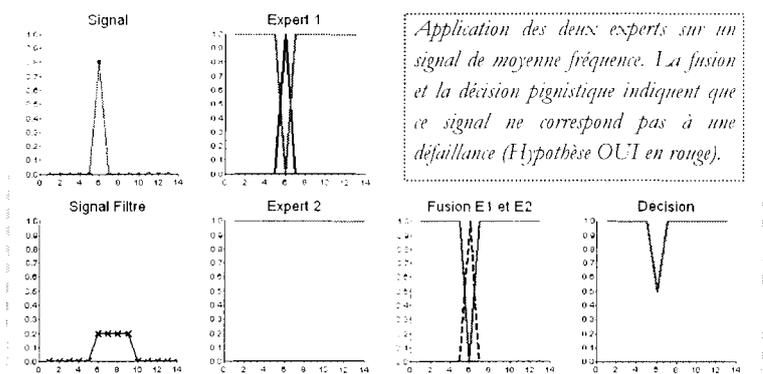
Ce premier cas présente un signal de basse fréquence. L'application des deux experts vus précédemment et leur fusion indiquent que ce signal représente une défaillance car son intensité et sa durée sont importantes. Ainsi, la prise de décision pignistique indique que la quasi-globalité du signal correspond à un signal de défaillance.



Ce second cas présente un signal de moyenne fréquence. L'application des deux experts vus précédemment et leur fusion indiquent que ce signal ne représente pas une faible défaillance car malgré une intensité importante, sa durée est faible. Ainsi, la prise de décision pignistique indique que ce signal ne correspond pas à un signal de défaillance.



Ce troisième cas présente un signal de haute fréquence. L'application des deux experts vus précédemment et leur fusion indiquent que ce signal ne représente pas une défaillance car sa durée est très faible. Ainsi, la prise de décision pignistique indique que la globalité du signal ne correspond pas à une défaillance.



L'ensemble des figures précédentes présente le comportement des deux experts en présence de trois signaux de types différents. Ainsi, lorsque l'on applique ces experts sur un signal dont la durée à un état haut est longue (fréquence basse), la fusion des deux experts fournit une décision sur l'hypothèse 'NON, le fonctionnement n'est pas correct' alors que si l'on applique ces experts à un signal dont la durée à un état haut est faible (fréquence haute), la décision sur l'hypothèse 'OUI, le fonctionnement est correct' est majoritaire.

D'une manière générale, la fusion des deux experts est réalisée grâce à l'opérateur de Smets selon $m_{1:i_j} = m_{1:i_j} \cap m_{1:2j}$ avec $i,j=1 \dots N$ et $i \neq j$. Le conflit issu de cette fusion est reporté sur l'ignorance.

Ainsi, ces deux experts permettent une gestion efficace des pics de conflits en permettant de conserver ou non des conflits dont la durée est significative ou pas.

A la sortie de l'étape 3, nous obtenons alors un jeu de masse $m_{1:i_j}$ portant sur une défaillance potentielle entre deux capteurs S_i et S_j , sans spécifier le capteur en faute. L'étape suivante permet de définir précisément si une source est défaillante ou non.

3.3 Identification du capteur défaillant

Dans cette section, nous présentons l'étape permettant d'identifier précisément le capteur défaillant. Pour réaliser ceci, nous nous appuyons sur l'étape précédente, visant à déterminer si une défaillance potentielle existe entre deux capteurs S_i et S_j . Ainsi, l'étape actuelle consiste à recouper ces informations de défaillance entre capteurs et à fusionner tous les experts se prononçant sur une défaillance potentielle dans laquelle est impliqué un capteur particulier.

Pour estimer l'état de fonctionnement d'un capteur S_i , nous utilisons le processus de fusion suivant basé sur l'opérateur de Dempster :

$$m_{\text{Fonct}S_i} = m_{E_{i1}} \oplus m_{E_{i2}} \oplus m_{E_{i3}} \oplus \dots \oplus m_{E_{iN}} \text{ avec } i=1\dots N$$

Ainsi, nous obtenons N fonctions de masses $m_{\text{Fonct}S_i}$ ($i=1\dots N$) représentant l'état de fonctionnement des sources. Le cadre de discernement utilisé $\Theta_{\text{Fonctionnement}}$ est le même que précédemment et est composée de deux éléments *OUI* et *NON*, correspondants respectivement aux assertions «*OUI, le fonctionnement est correct*» et «*NON, le fonctionnement n'est pas correct*»

Donc $\Theta_{\text{Fonctionnement}} = \{\text{OUI}, \text{NON}\}$

L'exemple suivant applique notre méthode de détection et d'identification de capteur défaillant. Nous utilisons un réseau composé de 3 capteurs. Ainsi, nous obtenons l'architecture suivante :

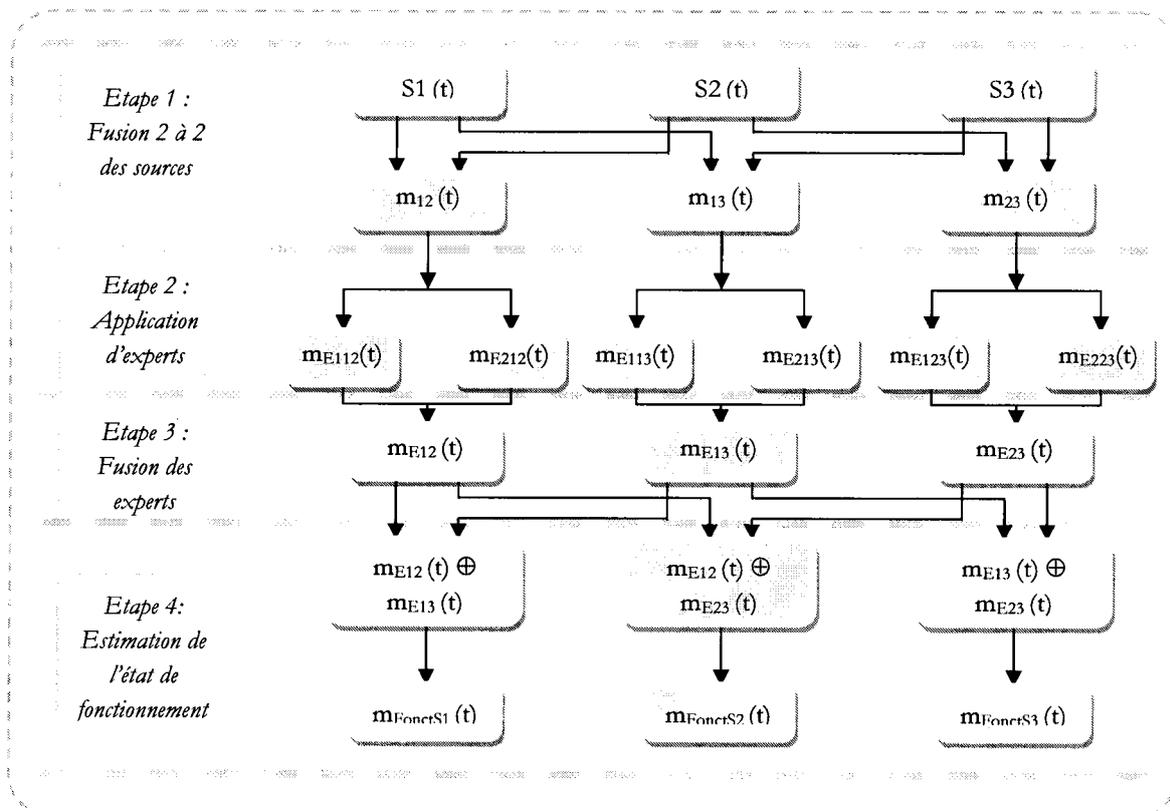


Figure 44 : Architecture de détection et d'identification de capteur défaillant dans un réseau de trois capteurs.

Prenons à un instant t , les jeux de masse ms_1 , ms_2 et ms_3 fournis respectivement par les 3 capteurs S_1 , S_2 , S_3 sur le cadre de discernement composé des assertions *OUI* et *NON* :

$$\Theta_{\text{Fonctionnement}} = \{OUI, NON\}, 2^{\Theta_{\text{Fonctionnement}}} = \{ \emptyset, OUI, NON, \Theta_{\text{Fonctionnement}} \}$$

S_1 :

$$ms_1(OUI) = 0.8$$

$$ms_1(NON) = 0$$

$$ms_1(OUI \cup NON) = 0.2$$

S_2 :

$$ms_2(OUI) = 0$$

$$ms_2(NON) = 0.75$$

$$ms_2(OUI \cup NON) = 0.25$$

S_3 :

$$ms_3(OUI) = 0.6$$

$$ms_3(NON) = 0$$

$$ms_3(OUI \cup NON) = 0.4$$

Ces trois jeux de masses indiquent que S_1 et S_3 sont en accord car ils privilégient l'hypothèse *OUI*. Le capteur S_2 privilégie quant à lui l'hypothèse *NON*. Ainsi, ce comportement anormal peut être détecté et isolé grâce à notre méthode.

La fusion 2 à 2 des trois capteurs permet d'obtenir les trois jeux de masses suivants :

- | | | |
|---------------------------------|---------------------------------|--------------------------------|
| • $m_{12}(OUI) = 0.2$ | • $m_{13}(OUI) = 0.92$ | • $m_{23}(OUI) = 0.15$ |
| • $m_{12}(NON) = 0.15$ | • $m_{13}(NON) = 0$ | • $m_{23}(NON) = 0.3$ |
| • $m_{12}(OUI \cup NON) = 0.05$ | • $m_{13}(OUI \cup NON) = 0.08$ | • $m_{23}(OUI \cup NON) = 0.1$ |
| • $k_{12} = 0.6$ | • $k_{13} = 0$ | • $k_{23} = 0.45$ |

Avec $m_{12} = ms_1 \cap ms_2$, $m_{13} = ms_1 \cap ms_3$, $m_{23} = ms_2 \cap ms_3$

Nous pouvons observer que le conflit entre capteurs est révélateur d'un problème au niveau du capteur S_2 . En effet, les conflits k_{12} et k_{23} sont importants. Ainsi, en recoupant ces deux conflits, nous pouvons observer que le capteur S_2 est le capteur commun dans lequel du conflit important est observé. Nous pouvons alors déduire que ce capteur est à l'origine de ces deux conflits et qu'il est vraisemblable qu'il soit défaillant. Par contre, comme nous l'avons dit précédemment, les conflits de courte durée peuvent être négligés. L'observation d'un conflit important à un instant t n'est pas significative, il faut observer des signaux de conflit durable afin de déterminer si oui ou non un capteur est réellement défaillant.

3.4 Estimation de la fiabilité d'un capteur

Lorsqu'un capteur a été identifié comme défaillant, il est intéressant d'affaiblir le jeu de masse lui correspondant afin de modérer son influence dans le processus de fusion final. Comme nous l'avons vu précédemment, la technique de l'affaiblissement associe un indice de fiabilité $a_i \in [0,1]$ à un capteur (voir Figure 40).

L'établissement du coefficient d'affaiblissement s'effectue en fonction de l'état du capteur correspondant. Ainsi, lorsqu'un capteur est complètement défaillant, $a = 0$. En revanche, lorsque le capteur n'est pas du tout défaillant, $a = 1$.

Le calcul des coefficients d'affaiblissement est basé sur l'estimation de fonctionnement des capteurs utilisés. L'estimation de fonctionnement $FonctS_i$ du capteur S_i correspond au calcul de la probabilité pignistique de sa fonction de masse m_{FonctS_i} respective :

$$\forall H_j \in \Theta \quad P_{FonctS_i}(H_j) = \sum_{\substack{A \in 2^\Theta \\ H_j \subset A}} \frac{1}{|A|} m_{FonctS_i}(A) \quad \text{avec } i=1..N$$

Et

$$FonctS_i = \max_j [P_{FonctS_i}(H_j)]$$

Au vue de la fréquence d'acquisition des données capteurs, il est pertinent de graduer ce coefficient en fonction d'un temps maximal $TMax$, estimé statistiquement et représentant la durée maximale pendant laquelle un capteur peut être défaillant avant d'être considéré comme étant en panne.

Ainsi, lorsque la défaillance est maximale, c'est à dire lorsque $FonctS_i(NON) = 1$, le coefficient d'affaiblissement sera nul, $a_i = 0$. D'une manière contraire, lorsque le capteur fonctionne correctement, $FonctS_i(NON) = 0$, $a_i = 1$.

En se basant sur ces deux postulats, la durée du conflit et l'intensité de celui-ci, les coefficients d'affaiblissement à un instant t s'obtiennent de la manière suivante :

$$\alpha'_i = \frac{TMax_i - D'_i}{TMax_i} \quad \text{avec} \quad D'_i = \sum_{k=t}^{TMax_i} FonctS_i(NON)(k)$$

Ces coefficients d'affaiblissement sont alors injectés dans l'équation de combinaison basée sur l'opérateur de Smets (voir Figure 40). Ainsi, $m_{i/N} = \alpha_1^t \cdot m_{S1}(t) \cap \alpha_2^t \cdot m_{S2}(t) \cap \alpha_3^t \cdot m_{S3}(t) \cap \dots \cap \alpha_N^t \cdot m_{SN}(t)$

3.5 Conclusion

Dans cette section, nous avons décrit une méthode permettant de détecter et d'isoler un capteur défaillant dans un réseau de capteurs. Cette méthode sur son principe, rejoint l'approche de Dubois et Prade visant à reporter la masse conflictuelle sur la disjonction d'hypothèses correspondant aux deux capteurs en contradiction. Nous utilisons les conflits sur les combinaisons partielles pour mettre en évidence, et ce par redondance, un capteur défaillant. La notion de temporalité est prise en charge par notre proposition.

Nous avons présenté une analyse du conflit résultant de l'ensemble des fusions deux à deux des N capteurs du réseau de capteurs. Cette analyse est basée sur la fusion de deux experts permettant de déterminer si oui ou non deux capteurs sont en contradiction. Afin de déterminer l'état de fonctionnement d'un capteur, une fusion est effectuée entre tous les experts se prononçant sur une défaillance potentielle dans lequel est impliquée le capteur évalué.

A partir de l'état de fonctionnement, nous calculons un coefficient d'affaiblissement pour chacun des capteurs. Cet affaiblissement permet de pondérer l'impact d'un capteur défaillant sur le processus de fusion final dans le cadre de la théorie de l'évidence.

Néanmoins, même si cette méthode est généralisée à N sources, elle ne mettra en exergue une source défaillante que si une et une unique source perturbe le système et de plus, de façon franche. C'est pour cette raison que nous nous intéressons dans le paragraphe suivant à une approche basée sur la modélisation du comportement d'un capteur par un apprentissage.

4 Détection des dérives de comportement capteur

4.1 Introduction

Nous pouvons nous demander comment détecter la dérive du comportement d'un capteur? La connaissance a priori sur le comportement normal de la source de données est primordial afin de pouvoir détecter un éventuel comportement anormal. Dans le cadre de nos travaux, un capteur possède différents états symboliques parmi lesquels il est susceptible d'être en cours de réalisation. Avec une approche symbolique, le fonctionnement du capteur est ainsi prédictif. En effet, à partir d'un état on peut estimer l'état ou l'ensemble des états suivants dans lequel le capteur est susceptible de se trouver à l'instant suivant.

Prenons l'exemple d'un capteur de température. Arbitrairement, nous pouvons définir trois états possibles (Froid, Moyen, Chaud) dans lesquels est susceptible de se trouver ce capteur. A partir d'une classification basée sur des sous ensembles flous, nous pouvons définir un enchaînement naturel du comportement du capteur. Par exemple, lorsque le capteur se trouve dans l'état Froid, il peut passer à l'état Moyen ou rester dans l'état Froid s'il y a une hausse minimale de la température. Ce changement d'état est fortement lié à la classification au préalable des états. De la même manière, lorsque le capteur se trouve dans l'état Chaud, il peut passer à l'état Moyen ou rester dans l'état Chaud s'il y a une baisse minimale de la température. L'état Moyen est un état intermédiaire qui permet de passer soit dans l'état Froid soit dans l'état Chaud. Par contre, le capteur ne peut passer de l'état Froid à l'état Chaud directement en cas d'une hausse minimale de la température.

Cette description définit un processus stochastique [INF06] et plus particulièrement une Chaîne de Markov [DOD99]. Notre choix est guidé par les caractéristiques propres à une Chaîne de Markov, à savoir la prédiction du futur à partir du présent ne nécessite pas la connaissance du passé.

4.2 Modélisation d'un capteur par un processus stochastique

Un processus stochastique $\{X_t : t \in T\}$ est une suite de variables aléatoires indexées par T à valeurs dans un ensemble χ .

T est l'ensemble des indices. Souvent t représente le temps mais t peut être de dimension multiple. Si T est à valeur discrète, on parle de processus à temps discret. Si l'ensemble des valeurs de T est continu, on parle de processus à temps continu.

χ est l'ensemble des états du processus. L'ensemble des états peut être continu ou discret. Dans le cadre de cette étude, nous nous limitons au cas où ces états sont en nombre fini.

La notion de processus élargit la notion de variable aléatoire. Une réalisation d'un processus est appelée trajectoire. C'est donc la suite des réalisations des variables aléatoires X_t .

Les réalisations d'une même variable aléatoire pouvant être différentes, les réalisations d'un même processus peuvent donner des trajectoires différentes.

En théorie des probabilités, un processus stochastique vérifie la propriété markovienne si et seulement si la distribution conditionnelle de probabilité des états futurs, étant donné l'instant présent, ne dépend que

de ce même état présent et pas des états passés. Un processus qui possède cette propriété est appelé processus de Markov.

Une Chaîne de Markov [FAU89] est un processus stochastique possédant la propriété markovienne. Dans un tel processus, la prédiction du futur à partir du présent ne nécessite pas la connaissance du passé. Une Chaîne de Markov en temps discret est une séquence X_1, X_2, X_3, \dots de variables aléatoires. L'ensemble de leurs valeurs possibles est appelé l'espace d'états, la valeur X_n étant l'état du processus au moment n . Si la distribution de probabilité conditionnelle de X_{n+1} sur les états passés est une fonction de X_n seul, alors :

$$P(X_{n+1} = x | X_0, X_1, X_2, \dots, X_n) = P(X_{n+1} = x | X_n).$$

Où x est un état quelconque du processus.

Une Chaîne de Markov est caractérisée par la distribution conditionnelle :

$$P(X_{n+1} | X_n)$$

qui est aussi appelée *probabilité de transition d'un pas* du processus.

Cette distribution de probabilité peut ainsi être représentée par une matrice de transition T . Cette matrice de probabilités de transition représente l'évolution d'un processus au cours du temps. Ainsi lorsqu'une transition d'un état à un autre est possible, la probabilité de transition est strictement positive, sinon lorsqu'une transition est impossible, la probabilité de transition est nulle.

Prenons l'exemple d'un capteur à trois états connus X_0, X_1 et X_2 . Il existe une probabilité de transition $P_{ij} \in [0,1]$ permettant le passage de l'état X_i à X_j . La Figure 45 présente la Chaîne de Markov pour un capteur à trois états.

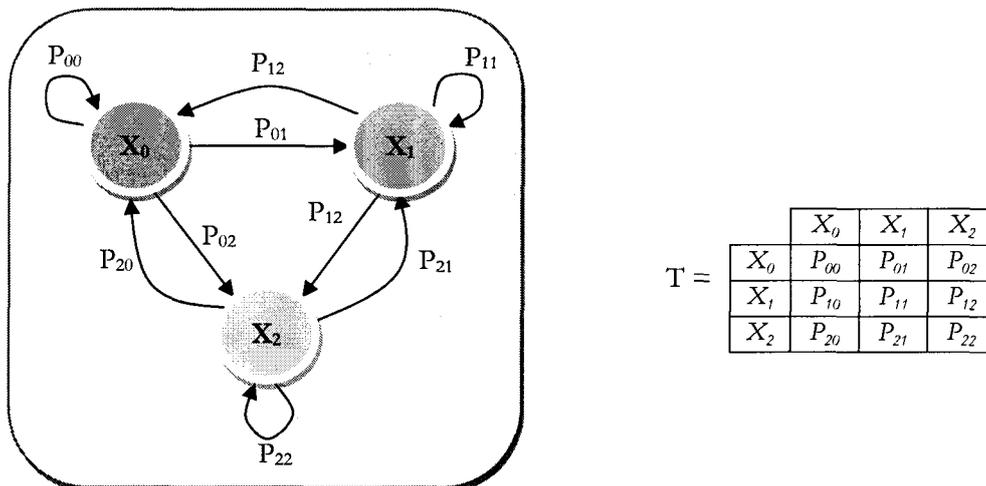


Figure 45 : Chaîne de Markov pour un capteur à trois états.

Le comportement normal d'un capteur sera lié à l'état dans lequel il se trouve et à son état précédent. En effet, si le capteur suit correctement l'évolution de la Chaîne de Markov, c'est-à-dire que pour le passage d'un état à un autre, une probabilité de transition existe, son comportement sera alors jugé comme normal. Par contre, lorsque le capteur passe d'un état à un autre alors que la probabilité de transition est nulle, alors son comportement est anormal.

Nous proposons donc pour détecter la défaillance d'un capteur :

1. de construire une prédiction stochastique de l'état que doit fournir une source à l'instant $t+1$,
2. de vérifier cette prédiction par rapport à ce qui est observé.

4.3 Principe de la détection de dérives

Tout comme dans les problèmes de filtrage probabiliste, tel que le filtrage de Kalman, nous allons élaborer une prédiction à partir d'une ou plusieurs observations. Le filtrage de Kalman est une méthode intéressante pour la prédiction et la vérification par observation. En nous appuyant sur cette philosophie, nous avons mis en œuvre une méthode pour la détection de dérives capteur reposant sur quatre étapes principales [RIC08b] présentées ci-après et dont l'enchaînement est présenté sur la Figure 46 :

- L'observation, qui est le jeu de masses initial à un instant t fourni par le capteur à diagnostiquer,
- La prédiction, qui est l'opération permettant d'estimer l'état du capteur à l'instant $t+1$ (Étape D1),
- La fusion, qui est l'opération consistant à combiner observation et prédiction (Étape D2),
- La vérification, qui consiste à vérifier le résultat de la fusion et à détecter le désaccord entre l'observation et la prédiction (Étape D3).

Le schéma général est le suivant :

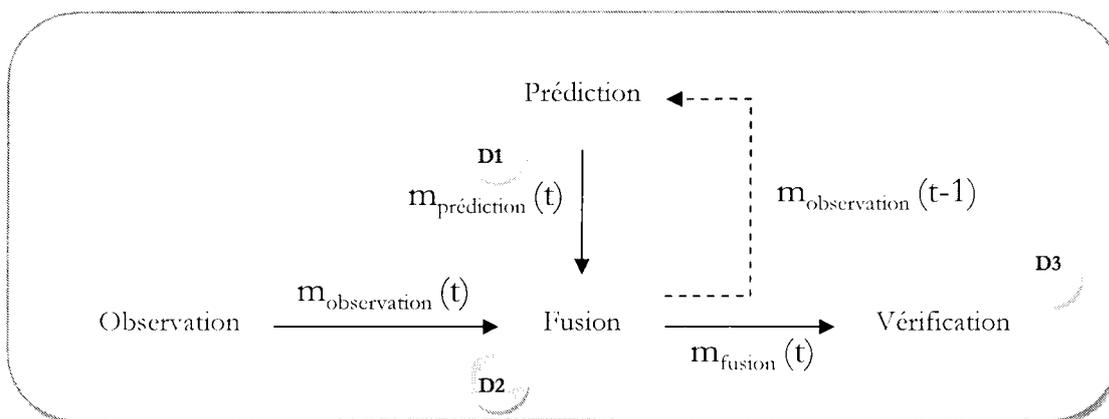


Figure 46 : Principe global de la détection de dérives comportementales.

L'observation (le capteur) fournit un jeu de masse $m_{\text{observation}}$ sur l'ensemble des hypothèses et disjonctions d'hypothèses du cadre de discernement utilisé. À partir de cette observation il est possible de construire un jeu de masses prédit $m_{\text{prédiction}}$ calculé à partir d'un modèle d'évolution. Dans notre cas, ce modèle d'évolution est une Chaîne de Markov représentant l'évolution des états possibles du capteur. À partir de cette Chaîne de Markov, il est possible d'effectuer une prédiction Markovienne. La prédiction constitue l'étape D1 de notre méthode.

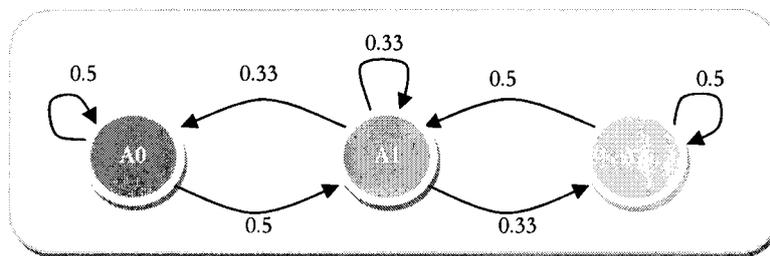
4.3.1 Méthode 1 : La prédiction Markovienne

Dans le cadre de l'utilisation de la prédiction markovienne, l'entrée de la prédiction est un vecteur d'observation probabiliste $V_{\text{Observation}}$ correspondant à l'état du capteur à l'instant t . Le calcul de la prédiction Markovienne à partir de la matrice de transition T , qui est un vecteur $V_{\text{Prédiction}}$ de même taille n que le vecteur $V_{\text{Observation}}$, s'obtient par :

$$V_{\text{Prédiction}} = V_{\text{Observation}} * T \quad (2)$$

La prédiction Markovienne est à la base de nombreux travaux. Dans [ZHU02] par exemple les Chaînes de Markov y sont utilisées dans le cadre de la navigation sur Internet. Elles permettent de prédire les hyperliens potentiellement atteignables à partir d'une page web. Ainsi, cette méthode assiste les nouveaux utilisateurs lors de leur navigation en les aidant à trouver de l'information plus précisément. De même, les travaux de [KOH00] s'appuient sur des Chaînes de Markov dans le cadre de la transmission de la voix sur l'Internet. Les Chaînes de Markov sont utilisées pour compenser la perte de paquet de données et fournir au récepteur un flux audio même en cas de perte, de suppression ou de retard de paquets.

Pour illustrer le principe de la prédiction Markovienne, prenons par exemple la matrice de transition T suivante modélisant trois états. A partir de l'état A0 à un instant t, le processus peut rester en A0 ou se trouver en A1 à l'instant suivant. De la même manière, à partir de l'état A3 à un instant t, le processus peut rester en A2 ou se trouver en A1 à l'instant suivant. Lorsque le processus se trouve à l'état A1 à l'instant t, le processus peut rester en A1, ou se trouver en A0 ou en A2 à l'instant suivant. Ainsi, la description précédente est traduite en une matrice de transition T et un graphe orienté qui sont les suivants :



$$T = \begin{array}{c|ccc} & A0 & A1 & A2 \\ \hline A0 & 0.5 & 0.5 & 0 \\ A1 & 0.33 & 0.33 & 0.33 \\ A2 & 0 & 0.5 & 0.5 \end{array}$$

Figure 47 : La Chaîne de Markov et sa matrice de transition associée.

Si nous prenons un vecteur d'observation $V_{Observation} = (1 \ 0 \ 0)$, c'est-à-dire que le processus se trouve dans l'état A0 à l'instant t, nous pouvons calculer le vecteur de prédiction $V_{Prédiction} = V_{Observation} * T = (0.5 \ 0.5 \ 0)$. Cette prédiction informe que lorsque le processus est dans l'état A0 avec une probabilité de 1, il se trouvera dans l'état A0 ou A1 à l'instant suivant.

Comme nous pouvons le voir avec la prédiction Markovienne, celle-ci s'effectue sur un jeu de probabilité en entrée et fournit un jeu de probabilité en sortie, c'est-à-dire un jeu de probabilités prédit. Or en entrée de notre système, nous avons un capteur fournissant un jeu de masse d'observation. Dans un souci de compatibilité, il faut procéder à une phase de conversion afin de permettre le calcul de la prédiction Markovienne à partir d'un jeu de masse d'observation tel que présenté Figure 48.

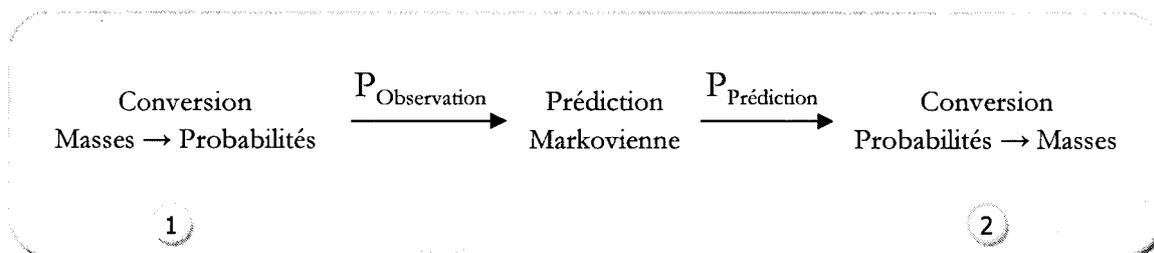


Figure 48 : Processus de conversion de masses dans le cadre d'une prédiction Markovienne.

1 - Processus de conversion pignistique

Afin de permettre le calcul de la prédiction à partir d'un jeu de masses, il faut procéder à une transformation du jeu de masses d'observation en un jeu de probabilités d'observation. Le calcul de la probabilité pignistique est une solution à ce problème. La formule suivante présente la méthode de calcul de la probabilité pignistique.

$$\forall Hi \in \Theta \quad P(Hi) = \sum_{\substack{A \in 2^\Theta \\ Hi \subset A}} \frac{1}{|A|} m(A)$$

Il est alors possible de calculer la prédiction Markovienne du jeu de probabilités ainsi obtenu. Ce jeu de probabilités prédit doit être à nouveau converti en un jeu de masse de prédiction afin de pouvoir effectuer la fusion entre prédiction et observation.

2 - Processus de conversion crédibiliste

Notre jeu de probabilités étant complet (Chapitre), il est possible de directement convertir ce jeu de probabilités en un jeu de masses. Les probabilités sur les hypothèses singletons sont alors directement placées sur les jeux de masses concernant les hypothèses singletons correspondantes. On obtient alors :

$$m(Hi) = P(Hi)$$

Ainsi, les masses sur les hypothèses non singletons sont nulles.

Prenons un exemple avec la Chaîne de Markov vue précédemment avec le jeu de masses $m_{\text{Observation}}$ sur le cadre de discernement Θ suivant : $\Theta = \{ A0, A1, A2 \}$. Le Tableau 2 présente l'évolution d'un jeu de masses initial à travers les différentes étapes de calcul de prédiction Markovienne.

<i>Le jeu de masses d'observation à t=0</i>	<i>Le jeu de probabilités calculé par transformation pignistique à t=0</i>	<i>Le jeu de probabilités résultant de la prédiction Markovienne pour t=1</i>	<i>Le jeu de masses résultant de la prédiction pour t=1</i>
$m_{\text{Observation}}(A0) = 0.7$	$P_{\text{Observation}}(A0) = 0.85$	$P_{\text{Prédiction}}(A0) = 0.4745$	$m_{\text{Prédiction}}(A0) = 0.4745$
$m_{\text{Observation}}(A1) = 0$	$P_{\text{Observation}}(A1) = 0.15$	$P_{\text{Prédiction}}(A1) = 0.4745$	$m_{\text{Prédiction}}(A1) = 0.4745$
$m_{\text{Observation}}(A2) = 0$	$P_{\text{Observation}}(A2) = 0$	$P_{\text{Prédiction}}(A2) = 0.0495$	$m_{\text{Prédiction}}(A2) = 0.0495$
$m_{\text{Observation}}(A0 \cup A1) = 0.3$			$m_{\text{Prédiction}}(A0 \cup A1) = 0$
$m_{\text{Observation}}(A0 \cup A2) = 0$			$m_{\text{Prédiction}}(A0 \cup A2) = 0$
$m_{\text{Observation}}(A1 \cup A2) = 0$			$m_{\text{Prédiction}}(A1 \cup A2) = 0$
$m_{\text{Observation}}(\Theta) = 0$			$m_{\text{Prédiction}}(\Theta) = 0$
$m_{\text{Observation}}(\emptyset) = 0$			$m_{\text{Prédiction}}(\emptyset) = 0$

Tableau 2 : Evolution du jeu de masses dans le cadre de la prédiction Markovienne.

Nous pouvons observer que le jeu de masse d'observation à $t=0$ indique une forte tendance vers l'état A_0 . Le jeu de masse prédit indique qu'à l'instant suivant, l'état sera vraisemblablement A_0 ou A_1 .

4.3.2 Méthode 2 : La prédiction Markovienne au sens crédibiliste

Dans le cadre de la prédiction Markovienne, nous venons de voir que nous ne travaillons pas directement sur des jeux de masses mais sur des jeux de probabilités, imposant un processus de conversion pignistique. Ce processus de conversion de masses vers probabilités, engendre une perte d'information au niveau de la précision de la représentation de l'information. La notion de doute est absorbée par la transformation pignistique et fait perdre les avantages liés aux fonctions de croyance. Nous avons donc décidé de passer par une matrice de transition permettant de directement travailler avec des jeux de masses.

La différence fondamentale entre la théorie des probabilités et la théorie des croyances réside dans le fait que pour la théorie des probabilités, la distribution se fait sur les hypothèses singletons alors que pour la théorie des croyances, la distribution se fait sur les hypothèses singletons et les disjonctions d'hypothèses. Ainsi, si nous nous tenons à cette différence, nous pouvons convertir une Chaîne de Markov probabiliste en une « Chaîne de Markov crédibiliste » [RIC08e] :

$$T_{\text{Cred}} =$$

	X_0	X_1	X_2	$X_0 \cup X_1$	$X_0 \cup X_2$	$X_1 \cup X_2$	Θ
X_0	P_{00}	P_{01}	P_{02}	$P_{0 \cup 1}$	$P_{0 \cup 2}$	$P_{0 \cup 2}$	$P_{0 \Theta}$
X_1	P_{10}	P_{11}	P_{12}	$P_{1 \cup 1}$	$P_{1 \cup 2}$	$P_{1 \cup 2}$	$P_{1 \Theta}$
X_2	P_{20}	P_{21}	P_{22}	$P_{2 \cup 1}$	$P_{2 \cup 2}$	$P_{2 \cup 2}$	$P_{2 \Theta}$
$X_0 \cup X_1$	$P_{0 \cup 1 \ 0}$	$P_{0 \cup 1 \ 1}$	$P_{0 \cup 1 \ 2}$	$P_{0 \cup 1 \ 0 \cup 1}$	$P_{0 \cup 1 \ 0 \cup 2}$	$P_{0 \cup 1 \ 1 \cup 2}$	$P_{0 \cup 1 \ \Theta}$
$X_0 \cup X_2$	$P_{0 \cup 2 \ 0}$	$P_{0 \cup 2 \ 1}$	$P_{0 \cup 2 \ 2}$	$P_{0 \cup 2 \ 0 \cup 1}$	$P_{0 \cup 2 \ 0 \cup 2}$	$P_{0 \cup 2 \ 1 \cup 2}$	$P_{0 \cup 2 \ \Theta}$
$X_1 \cup X_2$	$P_{1 \cup 2 \ 0}$	$P_{1 \cup 2 \ 1}$	$P_{1 \cup 2 \ 2}$	$P_{1 \cup 2 \ 0 \cup 1}$	$P_{1 \cup 2 \ 0 \cup 2}$	$P_{1 \cup 2 \ 1 \cup 2}$	$P_{1 \cup 2 \ \Theta}$
Θ	$P_{\Theta \ 0}$	$P_{\Theta \ 1}$	$P_{\Theta \ 2}$	$P_{\Theta \ 0 \cup 1}$	$P_{\Theta \ 0 \cup 2}$	$P_{\Theta \ 1 \cup 2}$	$P_{\Theta \ \Theta}$

Cette Chaîne de Markov crédibiliste est donnée pour un capteur à trois états X_0 , X_1 et X_2 . Nous observons alors que les probabilités de transitions sont étendues aux disjonctions d'hypothèses. Ainsi pour chaque état ou disjonction d'état, nous obtenons une probabilité $P_{ij} \in [0,1]$ permettant le passage de l'état X_i à X_j avec $i, j \in 2^\Theta$.

La version crédibiliste de la Chaîne de Markov facilite le travail de prédiction. En effet, les deux étapes de conversion de masses vers probabilités et inversement (cf Figure 48) ne sont plus nécessaires. Maintenant, il suffit d'utiliser un vecteur d'observation crédibiliste $V_{\text{Observation}}$ correspondant à l'état du capteur à l'instant t . Le calcul de la prédiction Markovienne crédibiliste à partir de la matrice de transition crédibiliste T_{Cred} s'obtient de la même manière :

$$V_{\text{Prediction}} = V_{\text{Observation}} * T_{\text{Cred}}$$

A partir de la matrice de transition de la Figure 47, nous pouvons transposer la matrice de transition T en une matrice de transition T_{Cred} présentée ci-après:

$$T_{\text{Cred}} =$$

	A_0	A_1	A_2	$A_0 \cup A_1$	$A_0 \cup A_2$	$A_1 \cup A_2$	Θ
A_0	1/3	1/3	0	1/3	0	0	0
A_1	0	0	0	0	0	0	1
A_2	0	1/3	1/3	0	0	1/3	0
$A_0 \cup A_1$	1/3	1/3	0	1/3	0	0	0
$A_0 \cup A_2$	0	0	0	0	0	0	0
$A_1 \cup A_2$	0	1/3	1/3	0	0	1/3	0
Θ	1/7	1/7	1/7	1/7	1/7	1/7	1/7

Cette matrice crédibiliste redistribue les probabilités de transition sur les hypothèses et sur les disjonctions d'hypothèses. Ainsi, sur la Figure 47 nous voyons que lorsque le capteur se trouve dans l'état A_0 , il pouvait soit se trouver dans les états A_0 ou A_1 à l'instant suivant. La version crédibiliste est plus souple, plus fine et donc plus précise et permet d'attribuer une probabilité de transition vers les disjonctions d'états $A_0 \cup A_1$. Elle est néanmoins beaucoup plus combinatoire.

Prenons cette Chaîne de Markov crédibiliste avec le jeu de masses $m_{\text{Observation}}$ en entrée sur le cadre de discernement Θ suivant : $\Theta = \{ A_0, A_1, A_2 \}$.

Le Tableau 3 présente le jeu de masses final suite au calcul de la prédiction Markovienne crédibiliste à partir d'un jeu de masses en entrée.

<i>Le jeu de masses d'observation</i>	<i>Le jeu de masses résultant de la prédiction</i>	<i>Le jeu de masses résultant de la prédiction à partir du processus de conversion pignistique (pour comparaison)</i>
$m_{\text{Observation}}(A_0) = 0.7$	$m_{\text{Prédiction}}(A_0) = 0.33$	$m_{\text{Prédiction}}(A_0) = 0.4745$
$m_{\text{Observation}}(A_1) = 0$	$m_{\text{Prédiction}}(A_1) = 0.33$	$m_{\text{Prédiction}}(A_1) = 0.4745$
$m_{\text{Observation}}(A_2) = 0$	$m_{\text{Prédiction}}(A_2) = 0$	$m_{\text{Prédiction}}(A_2) = 0.0495$
$m_{\text{Observation}}(A_0 \cup A_1) = 0.3$	$m_{\text{Prédiction}}(A_0 \cup A_1) = 0.33$	$m_{\text{Prédiction}}(A_0 \cup A_1) = 0$
$m_{\text{Observation}}(A_0 \cup A_2) = 0$	$m_{\text{Prédiction}}(A_0 \cup A_2) = 0$	$m_{\text{Prédiction}}(A_0 \cup A_2) = 0$
$m_{\text{Observation}}(A_1 \cup A_2) = 0$	$m_{\text{Prédiction}}(A_1 \cup A_2) = 0$	$m_{\text{Prédiction}}(A_1 \cup A_2) = 0$
$m_{\text{Observation}}(\Theta) = 0$	$m_{\text{Prédiction}}(\Theta) = 0$	$m_{\text{Prédiction}}(\Theta) = 0$
$m_{\text{Observation}}(\emptyset) = 0$	$m_{\text{Prédiction}}(\emptyset) = 0$	$m_{\text{Prédiction}}(\emptyset) = 0$

Tableau 3 : Evolution du jeu de masses dans le cadre de la prédiction Markovienne crédibiliste.

On peut constater que par rapport à la première version, la masse sur le doute $m(A_0 \cup A_1)$ est non-nulle. Cette propriété est importante car elle permet de nuancer le jeu de masse prédit et de mettre en avant le doute existant entre 2 hypothèses prédites.

D'une manière empirique, nous pouvons rendre plus précise la matrice de transition T_{Cred} . En effet, nous avons précédemment défini comme équiprobable le passage d'un état à un autre, qu'il s'agisse d'un état

singleton ou non. Or lorsque, le processus se trouve dans un état singleton, son fonctionnement implique un passage par un état intermédiaire puis à un état singleton. Par exemple, prenons un capteur de température ayant deux états singletons « Froid » et « Chaud ». Lorsque la température se situe en état Froid, il se trouvera certainement dans un état intermédiaire composé de Froid C Chaud à l'instant suivant plutôt que directement en Chaud. Cette notion d'états intermédiaires est ici répercutée au niveau des coefficients de la matrice de transition T_{Cred} en favorisant le passage par un état intermédiaire à partir d'un état singleton:

$$T_{Cred} =$$

	A_0	A_1	A_2	$A_0 \cup A_1$	$A_0 \cup A_2$	$A_1 \cup A_2$	Θ
A_0	0.4	0.2	0	0.4	0	0	0
A_1	0	0	0	0	0	0	1
A_2	0	0.2	0.4	0	0	0.4	0
$A_0 \cup A_1$	1/3	1/3	0	1/3	0	0	0
$A_0 \cup A_2$	0	0	0	0	0	0	0
$A_1 \cup A_2$	0	1/3	1/3	0	0	1/3	0
Θ	1/7	1/7	1/7	1/7	1/7	1/7	1/7

<i>Le jeu de masses d'observation</i>	<i>Le jeu de masses résultant de la prédiction</i>
--	---

$m_{\text{Observation}}(A_0) = 0.7$	$m_{\text{Prédiction}}(A_0) = 0.379$
$m_{\text{Observation}}(A_1) = 0$	$m_{\text{Prédiction}}(A_1) = 0.239$
$m_{\text{Observation}}(A_2) = 0$	$m_{\text{Prédiction}}(A_2) = 0$
$m_{\text{Observation}}(A_0 \cup A_1) = 0.3$	$m_{\text{Prédiction}}(A_0 \cup A_1) = 0.379$
$m_{\text{Observation}}(A_0 \cup A_2) = 0$	$m_{\text{Prédiction}}(A_0 \cup A_2) = 0$
$m_{\text{Observation}}(A_1 \cup A_2) = 0$	$m_{\text{Prédiction}}(A_1 \cup A_2) = 0$
$m_{\text{Observation}}(\Theta) = 0$	$m_{\text{Prédiction}}(\Theta) = 0$
$m_{\text{Observation}}(\emptyset) = 0$	$m_{\text{Prédiction}}(\emptyset) = 0$

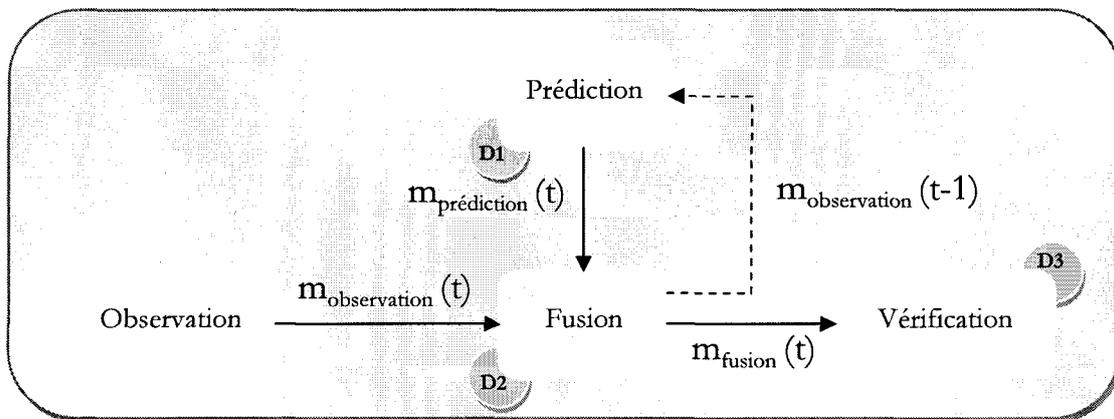
Tableau 4 : Evolution du jeu de masses dans le cadre de la prédiction Markovienne crédibiliste.

Avec cette nouvelle matrice de transition, nous pouvons observer que le comportement de la prédiction représente plus précisément le comportement attendu (voir Tableau 4). Le jeu de masses d'observation informe que le processus se trouve en état A_0 en nous indiquant qu'il a un léger doute entre A_0 et A_1 . Ce jeu de masses peut alors correspondre à un début de phase de changement d'état. Le jeu de masses prédit informe alors qu'à l'instant suivant, le processus sera soit dans l'état A_0 ou A_1 (avec tout de même une préférence pour rester en A_0) mais sans oublier le doute entre A_0 et A_1 , ce qui conforte bien le fait que le processus est dans une phase de changement d'état.

4.3.3 Détection d'une dérive par analyse du conflit entre observation et prédiction

Pour effectuer la fusion entre le jeu de masse de prédiction $m_{\text{Prédiction}}$ et le jeu de masse d'observation $m_{\text{Observation}}$ nous utilisons l'opérateur de Smets \cap permettant d'isoler le conflit résultant de la fusion entre $m_{\text{Prédiction}}$ et $m_{\text{Observation}}$: $m_{\text{fusion}} = m_{\text{Prédiction}} \cap m_{\text{Observation}}$

Ce niveau de fusion représente l'étape D2 de notre méthode. Cette étape est cruciale pour la suite de notre travail car elle permet de caractériser le conflit entre observation et prédiction.



Pour détecter une dérive de comportement du capteur, nous observons l'évolution temporelle du conflit entre observation et prédiction (Etape D3). Nos essais ont montré qu'une dérive de comportement se caractérise par un saut soudain du conflit. En effet, lorsque la prédiction et l'observation seront différents, la fusion des jeux de masses $m_{\text{Observation}}$ et $m_{\text{Prédiction}}$ provoque la hausse du conflit $m_{\text{fusion}}(-)$. Les essais montrent également que le conflit subit une hausse soudaine en cas de dérive puis retrouve une valeur stable une fois que le prédicteur Markovien se stabilise.

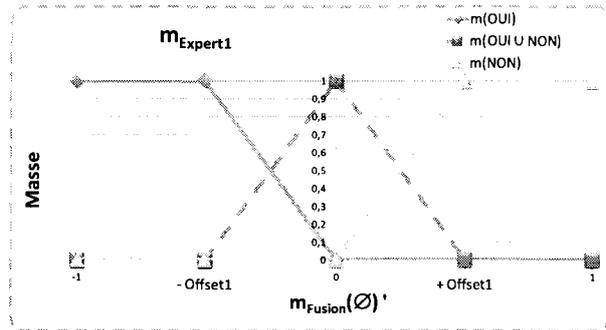
Ainsi, une dérive est caractérisée par une hausse du conflit pendant un temps court, c'est-à-dire un pic de conflit. Pour automatiquement détecter ce pic de conflit, nous utilisons trois experts. Ils fournissent une opinion sur la forme du signal de conflit. Le premier expert porte sur la dérivée du signal de conflit afin de détecter des pics montants et non descendants. Le second expert porte sur l'amplitude des pics afin de détecter les sauts soudains de conflits. Le troisième porte sur la durée des pics afin de détecter les forts pics de courte durée. A partir de la fusion de ces trois experts, nous isolons les pics de conflit correspondant à des sauts d'états impossibles.

Les experts se prononcent sur la pertinence du conflit. Le cadre de discernement utilisé $\Theta_{\text{Comportement}}$ est composé de deux éléments *OUI* et *NON*, correspondant respectivement aux assertions « *OUI, le comportement est normal* » et « *NON, le comportement n'est pas normal* »

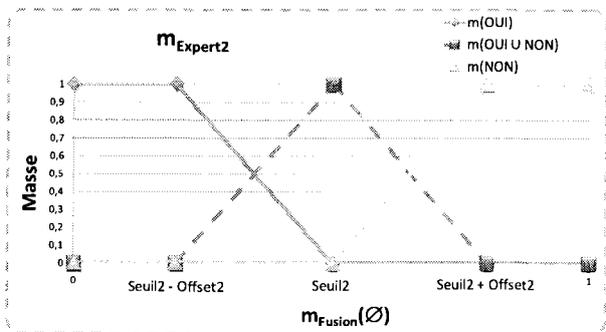
Donc $\Theta_{\text{Comportement}} = \{OUI, NON\}$

Pour chacun de ces trois experts, nous avons déterminé expérimentalement les fonctions de masses m_{Expert1} , m_{Expert2} , m_{Expert3} .

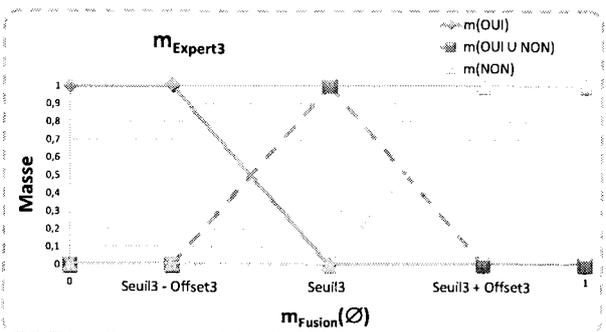
Cette première fonction de masse fournit un jeu de masse $m_{Expert1}$ portant sur la direction entre deux points consécutifs. Cette fonction de masse est appliquée sur la dérivée du signal de conflit $m_{Fusion}(e)$. Elle se base sur la différence entre deux points consécutifs afin de déterminer la direction. Ainsi, cet expert permet de se prononcer sur la montée ou la descente du conflit. Quand le conflit augmente, sa dérivée est négative, quand le conflit baisse, sa dérivée est positive. L'écart 'offset1' permet de précisément déterminer la zone de doute lorsque la valeur est proche de 0, c'est-à-dire lorsque le conflit est constant.



Cette seconde fonction de masse fournit un jeu de masse $m_{Expert2}$ portant sur l'amplitude du conflit à un instant t. Cette fonction de masse est appliquée sur le signal de conflit $m_{Fusion}(e)$. Cet expert permet de se prononcer sur un haut pic ou un faible pic de conflit. Quand l'amplitude dépasse la valeur seuil 'seuil2', cela signifie que le pic est important, sinon le pic est faible. L'écart 'offset2' permet de précisément déterminer la zone de doute sur l'amplitude du conflit, c'est-à-dire lorsque le pic de conflit est moyen.



Cette troisième fonction de masse fournit un jeu de masse $m_{Expert3}$ portant sur la variation de valeur entre N points consécutifs. Cette fonction de masse est appliquée sur le signal de conflit $m_{Fusion}(e)$. Elle se base sur la différence entre N points consécutifs. Cet expert permet de se prononcer sur un conflit de courte durée. Quand la variation entre N points consécutifs dépasse la valeur de seuil 'seuil3', cela signifie que c'est un pic de conflit, sinon cela signifie que c'est simplement une hausse de conflit non significative. L'écart 'offset3' permet de précisément déterminer la zone de doute sur la variation entre N points consécutifs du conflit, c'est-à-dire lorsque le pic de conflit est de durée moyenne.



Avec

$$m_{Expert1} = f_1(m_{Fusion})$$

$$m_{Expert2} = f_2(m_{Fusion})$$

$$m_{Expert3} = f_3(m_{Fusion})$$

Les résultats des ces trois experts sont fusionnés grâce à l'opérateur de Dempster afin d'obtenir l'estimation de fonctionnement du capteur :

$$m_{\text{fonct}} = m_{\text{expert1}} \oplus m_{\text{expert2}} \oplus m_{\text{expert3}}$$

m_{fonct} fournit un jeu de masse pour chacun des points du signal de conflit et indique si ce point correspond à une dérive ou non. La décision pignistique sur ce jeu de masse permet de détecter une dérive de comportement.

Cette section a permis de décrire une méthode de détection de dérives de comportement capteur dans le cadre de la théorie des fonctions de Croissance. La méthode présentée s'appuie sur la comparaison entre un état observé et un état prédit. L'état prédit se base sur le modèle de comportement du capteur défini au préalable. Deux jeux de masses sont obtenus et fusionnés. Le conflit entre les jeux de masses observés et prédits est analysé grâce à trois experts.

Deux approches sont présentées. La première se base sur l'utilisation d'un modèle de comportement probabiliste à l'aide d'une Chaîne de Markov. Cette première approche a l'inconvénient de faire perdre les avantages liés aux fonctions de croissance en passant par un processus de conversion de masses vers probabilités. La seconde approche résout ce problème et propose une approche à base de fonction de Croissance. Nous avons défini une Chaîne de Markov crédibiliste, plus adaptée dans le cadre de la théorie des fonctions de Croissance.

Ainsi la seconde approche permet d'obtenir une prédiction plus précise du comportement attendu. Celle-ci facilite alors la constitution du jeu de masse prédit et la recherche de dérive de comportement capteur.

5 Conclusion

Dans ce chapitre, nous avons présenté une approche crédibiliste dédiée à la fusion de données dans le cadre de la construction de contexte. Nous avons mis en exergue le problème de l'accroissement de conflit, caractéristique d'une architecture où le nombre de capteurs est important et où l'interaction de nombreux expert peut introduire des dissonances. Après avoir présenté le cadre général de la théorie de Dempster-Shafer, connue sous le nom de théorie de l'évidence ou théorie des fonctions de croissance, nous avons synthétisé les différentes approches permettant de traiter le problème du conflit. Nous avons mis en évidence différentes causes de conflit, ainsi que l'adéquation de certaines méthodes de redistribution du conflit avec certains contextes de fusion. Fort de ce constat nous avons proposé une approche générique d'affaiblissement de conflit basée sur la détection et l'identification d'un capteur défaillant dans un réseau de N capteurs. L'objectif de cette proposition est d'une part d'isoler la source d'accroissement du conflit lié à la défaillance capteur et d'autre part de solutionner le problème de défaillance par un affaiblissement dynamique. L'identification du capteur défaillant permet de pondérer son influence lors du processus de fusion final et de permettre de prendre une décision plus rigoureuse. La détection d'un capteur défaillant est effectuée en analysant le conflit résultant de l'ensemble des fusions deux à deux des N capteurs du réseau de capteurs. L'état de fonctionnement de chacun des capteurs est le résultat d'une fusion entre tous les experts se prononçant sur une défaillance potentielle dans lequel est impliqué le capteur évalué. A partir de l'état de fonctionnement du capteur, nous pouvons estimer un coefficient d'affaiblissement permettant de pondérer l'influence du capteur défaillant sur le processus de fusion final.

Après avoir fixé les limites de cette première approche, nous avons voulu poser les bases d'une solution intégrant un mécanisme de modélisation du comportement du capteur. Cette modélisation repose sur la construction d'une Chaîne de Markov qui permet de prédire l'état fourni par un capteur à un instant $t+1$.

La méthode présentée s'appuie sur la comparaison entre cet état prédit et celui observé. L'état prédit s'appuie sur une expertise ou sur un apprentissage qui pourra être supervisé ou non. Le conflit entre l'observation et la prédiction est analysé grâce à trois experts qui statuent après combinaison sur l'identification de la défaillance. La résolution du problème est abordée comme dans la première approche.

L'utilisation d'un modèle de comportement probabiliste à l'aide d'une Chaîne de Markov étant peu adaptée dans un cadre global crédibiliste, nous proposons une modélisation du comportement capteur purement crédibiliste. Cette dernière permet d'obtenir une prédiction plus précise du comportement attendu. Elle est globalement plus homogène avec l'architecture et permet de solutionner la perte d'information lors du calcul des probabilités pignistiques.

Dans la suite, après avoir étayé notre propos avec des résultats expérimentaux, nous montrons que les trois stratégies développées peuvent être utilisées dans un cadre beaucoup plus général de détection d'états anormaux. C'est pour cette raison que nous proposons dans le chapitre suivant une mise en cascade de ces approches pour analyser la quantité résiduelle de conflit inhérent cette fois ci à une dérive comportementale dans le cadre de la surveillance de l'activité de la personne. Nous montrons qu'après avoir isolé la source de conflit liée à la défaillance capteur, il nous est possible d'analyser finement des changements d'états anormaux propres cette fois ci à ce qu'observent les capteurs et au comportement de l'individu observé.

*Where was I going? What was I
doing? I'd soon find out.
Jack Kerouac, On The Road.*

Chapitre 4

Application de la fusion de données crédibiliste à la surveillance d'activités humaines dans l'habitat communicant

1 Introduction

Dans ce chapitre, nous mettons en œuvre une architecture de fusion de données dans le cadre de la reconnaissance d'activités humaines. Ce chapitre focalise sur une méthode de fusion de données capteurs permettant de caractériser la posture d'un individu à un instant t à partir d'un ensemble de données hétérogènes. La prise en compte de la fiabilité capteur est effectuée en veillant à détecter toute anomalie de données capteurs.

La fiabilité des capteurs est estimée grâce au mécanisme de détection et d'identification de capteur défaillant présenté à la section 3 du chapitre précédent. La reconnaissance de dérive de comportement d'un individu est réalisée en se basant sur un modèle de comportement supposé normal de l'individu. Ici, nous surveillons l'enchaînement temporel des postures d'un individu afin de détecter une éventuelle anomalie. Cette surveillance est réalisée à partir de la méthode présentée à la section 4 du chapitre précédent. A partir d'une modélisation du comportement normal d'un individu, une prédiction est réalisée visant à estimer la posture la plus vraisemblable à l'instant suivant. Par comparaison avec la posture observée, il est possible de détecter une anomalie.

Nous permettons ainsi une séparation des sources à l'origine du conflit, soit une défaillance capteur, soit une anomalie de comportement. En fonction de la nature du conflit, nous adaptons la fiabilité du capteur estimé défaillant, soit nous émettons une alerte de dérive comportementale.

Ce chapitre est organisé de la manière suivante. La première section présente une architecture de fusion de données dédiée à la reconnaissance de posture d'un individu. Nous présentons dans un premier temps les capteurs retenus ainsi que la chaîne complète permettant d'atteindre un niveau symbolique afin de permettre une fusion finale.

Ensuite, nous présentons une première expérimentation dans laquelle les capteurs utilisés fonctionnent correctement. Dans une seconde expérimentation, un capteur devient défaillant. L'objectif est d'observer l'impact de capteur sur le processus de fusion finale et d'effectuer une comparaison avec la mise en œuvre de la détection de capteur défaillant. Nous montrons alors l'intérêt de la prise en compte des capteurs défaillants et de leurs pondérations lors du processus de fusion finale.

La dernière section présente une expérimentation d'analyse de l'activité d'un individu. Cette analyse repose sur la posture d'un individu et vise à détecter des enchaînements de postures humaines considérées comme anormales. Nous mettons en œuvre le mécanisme de fusion observation/prédiction vu au chapitre 3. Celui-ci permet alors de détecter des comportements anormaux qui pourraient s'avérer dangereux pour l'individu.

Finalement, nous présentons l'impact d'un capteur défaillant sur l'analyse du comportement de l'individu. Nous montrons que le découplage dans la recherche de l'origine du conflit est important et qu'il permet de prendre des décisions adaptées même en cas de fort conflit.

La Figure 49 présente l'enchaînement des différentes parties que nous allons voir dans ce chapitre. Ainsi, l'architecture classique de fusion de données est représentée ici en bleue, étape notée E1. Celle-ci combine des données issues de N capteurs afin d'extraire une donnée contextuelle : la posture de l'individu observé. Une fois cette donnée extraite, celle-ci est stockée dans notre ontologie. La partie verte, étape E2, représente le mécanisme d'estimation de la fiabilité des capteurs utilisés. Elle a pour charge de superviser le fonctionnement des capteurs et de déterminer des défaillances capteur. Ainsi, cette partie agit directement sur le processus de fusion puisqu'elle détermine les coefficients d'affaiblissement

dynamiquement pour chacun des capteurs fusionnés. Finalement la partie orangée, étape E3, permet de déterminer une anomalie de comportement d'un individu. Cette partie élabore une prédiction à partir de la fusion réalisée entre les différents capteurs et effectue une vérification entre observation et prédiction. Cette partie est capable de déterminer une donnée contextuelle relative au comportement de l'individu et plus particulièrement lorsqu'il y a une anomalie de comportement.

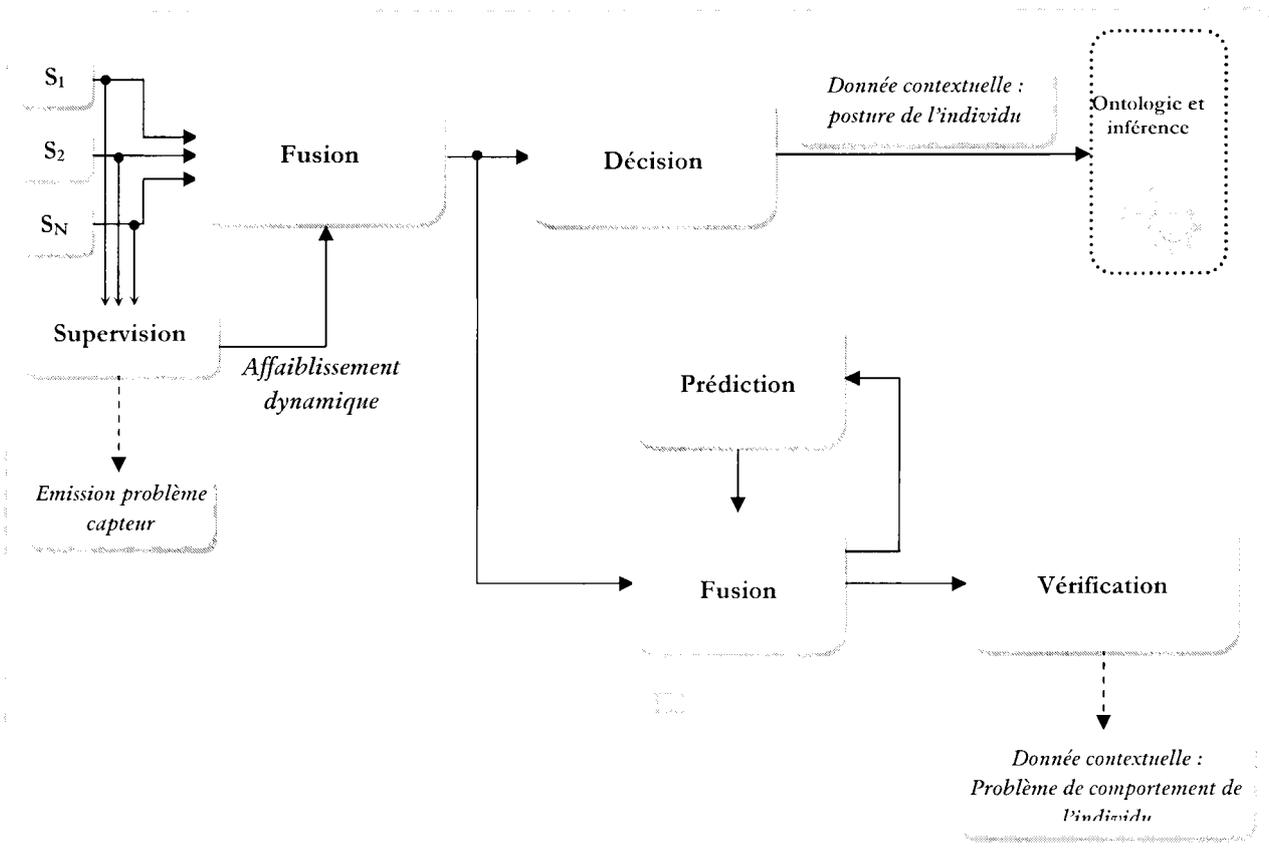


Figure 49 : Chaîne complète d'analyse comportementale d'un individu avec prise en charge de la défaillance capteur.

2 La fusion de données : Application à la surveillance d'activités humaines – Etape E1

2.1 Introduction

Dans le cadre du maintien à domicile, le suivi des activités de la vie quotidienne est un point important dans la détection des problèmes de santé. En effet, certains médecins [MAN03] affirment que l'identification de changements dans les activités quotidiennes d'une personne âgée (préparation des repas, rangement,...) peut s'avérer plus informative que le suivi biométrique de cette personne. Ce suivi régulier n'est alors possible que par l'intermédiaire de relevés de données physiologiques visant à détecter des problèmes de santé. Typiquement, une activité est vue comme une suite d'actions que le système connaît au préalable ou qu'il va apprendre. Une activité se détermine par un ensemble d'indices, qui selon une certaine organisation, permettent de caractériser une activité. Ainsi, l'activité 'je fais du café' se décompose en actions primaires : je prends de l'eau, je prends du café en poudre, je prends un filtre à café, j'allume la cafetière,...

Le changement d'habitude. Un point important, dont on se rend compte particulièrement chez les personnes âgées, est la notion d'habitude. Nous connaissons tous une grand-mère ou un grand-père répétant maintes fois les mêmes rituels de manière périodique. La notion d'habitude est alors importante dans la vie des personnes âgées. Cette aptitude à réaliser de manière périodique et organisée un acte, est un point important qui permet de déterminer si tout va bien ou si quelque chose se passe mal. Par exemple, si Quentin est habitué à préparer son café à 7h00 et que ce lundi, à 7h30, Quentin n'a toujours pas préparé son café, que peut-on penser ?

Deux cas sont envisageables :

1. La cafetière ne fonctionne plus, il n'y a plus de café en poudre,... c'est ce que nous appelons ici un *problème de perception*.
2. Quentin est tombé et il est bloqué, il a eu un malaise, il n'est pas chez lui,... c'est ce que nous appelons ici un *problème de comportement*.

Ce changement d'habitude utilisé seul, ne peut indiquer précisément la nature du problème. En effet, un problème de comportement peut découler d'un problème de perception : Quentin ne prépare pas son café à 7h00 car il n'y a plus de café en poudre. Nous observons alors l'impact d'un problème de perception : quel cas doit-on envisager ? Problème de comportement ou problème de perception ?

Un problème de perception peut ainsi masquer des comportements qu'ils soient normaux ou anormaux. Afin d'éviter ce problème, il faut dissocier ces deux problèmes et opter pour un découpage en deux phases: la phase de perception et la phase d'analyse comportementale. La phase de perception vise à obtenir des informations exploitables et à gérer les problèmes qui peuvent survenir. C'est ce que nous appellerons des problèmes de formes de perception. La phase d'analyse comportementale vise à observer l'enchaînement des actions et à détecter d'éventuelles anomalies de comportement. C'est ce que nous appellerons des problèmes de fond de perception.

Nous allons exploiter les méthodes vues au chapitre 3 afin de détecter des problèmes de formes sur la perception : c'est-à-dire la fusion de données avec détection et identification d'un capteur défaillant, ceci formera la phase de perception. Ensuite, nous exploitons la seconde méthode vue au chapitre précédent (*chapitre 3 - §4*) afin de détecter des problèmes de fond sur la perception : c'est-à-dire la phase de détection d'anomalies comportementales à partir d'un processus de fusion entre observation et prédiction.

L'objectif est de détecter des anomalies dans l'enchaînement des postures d'une personne âgée. Ainsi, si une personne passe d'une posture debout à une posture allongée, la probabilité qu'il y ait une chute est forte. Nous nous intéressons à ce problème car la chute est la première cause de décès chez les personnes de plus de 65 ans. Même si la mortalité ne s'en suit pas, une chute peut laisser de graves séquelles telles qu'une fracture. Statistiquement, après une fracture du col du fémur, un tiers des personnes décèdent dans l'année, un tiers garde des séquelles et un tiers seulement récupère ses capacités. La chute a également des conséquences psychologiques telles que la peur de retomber. Cette peur peut engendrer un déclin de l'activité des personnes dans 30% des cas, se terminant par un placement en institution spécialisée. Ainsi, dans le cadre du maintien à domicile, il est primordial de détecter une chute le plus rapidement possible.

2.2 Caractérisation d'une posture

Afin de précisément caractériser la posture de l'individu observé, nous avons défini un réseau de trois capteurs redondants et complémentaires [RIC08a]. L'objectif est de déterminer les trois postures caractéristiques d'un individu : « debout, assis et allongé » ce qui dans le cadre du *Modèle des Croyances*

Transférables (MCI) fournit le cadre de discernement $\Theta_{Posture}$ composé des 3 hypothèses $\{debout, assis, allongé\}$. Il est possible de répartir des masses parmi les éléments de :

$$2^{\Theta_{Posture}} = \{ debout, assis, allongé, debout \cup assis, debout \cup allongé, assis \cup allongé, \Theta_{Posture} \}.$$

Divers travaux présentent des méthodes de reconnaissance de posture à partir d'une caméra placée devant la scène à observer [WAN05][AGA05]. Les résultats obtenus sont probants, néanmoins, les conditions d'éclairage sont parfaites (les essais sont effectués dans une salle éclairée sans aucun bruit externe). En effet, dans le cadre de la vision par ordinateur, l'éclairage de la scène est une source classique de problèmes lors des traitements. Pour renforcer la décision issue du traitement d'image, nous mettons en œuvre un système de capteurs redondants et complémentaires.

2.2.1 Premier capteur retenu : Le capteur de pression mécanique

Les capteurs ont été choisis selon un critère de coût et principalement parce qu'ils permettent de caractériser partiellement la posture d'un individu. Afin de détecter la posture assise de l'individu, nous avons placé des capteurs de pression mécanique (ou *FSR* : *Force Sensitive Resistance*) sous les sièges de chacune des chaises. Ces capteurs sont des dispositifs dont la résistance électrique diminue lorsqu'on applique une pression. Nous utilisons ces capteurs pour détecter lorsqu'une personne s'assoit sur une chaise.

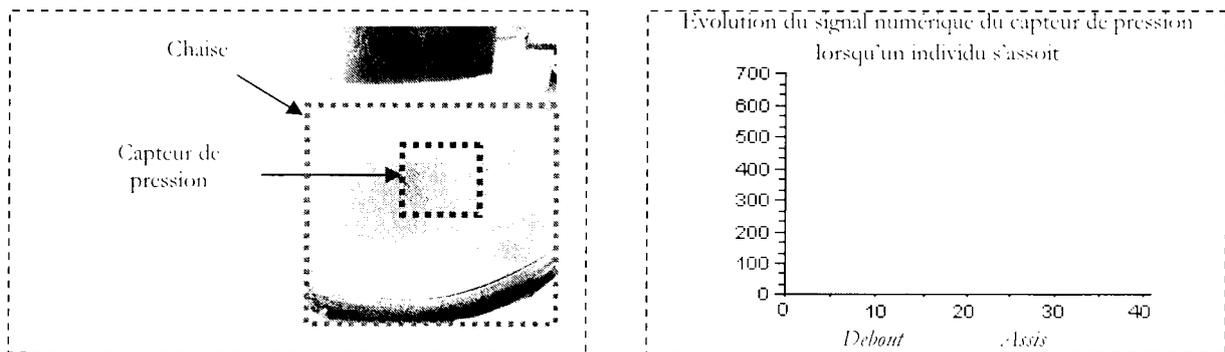


Figure 50 : La chaise équipée du capteur de pression mécanique et un le signal issu du capteur lorsqu'un individu s'assoit.

Sur la Figure 50, nous pouvons observer que l'information de pression issue du capteur est très caractéristique lorsqu'un individu s'assoit. Néanmoins, ce capteur ne peut informer de la posture debout ou allongée de l'individu. Dans le cadre du *MCI*, nous avons expérimentalement déterminé la fonction de masse m_i présentée sur la Figure 51.

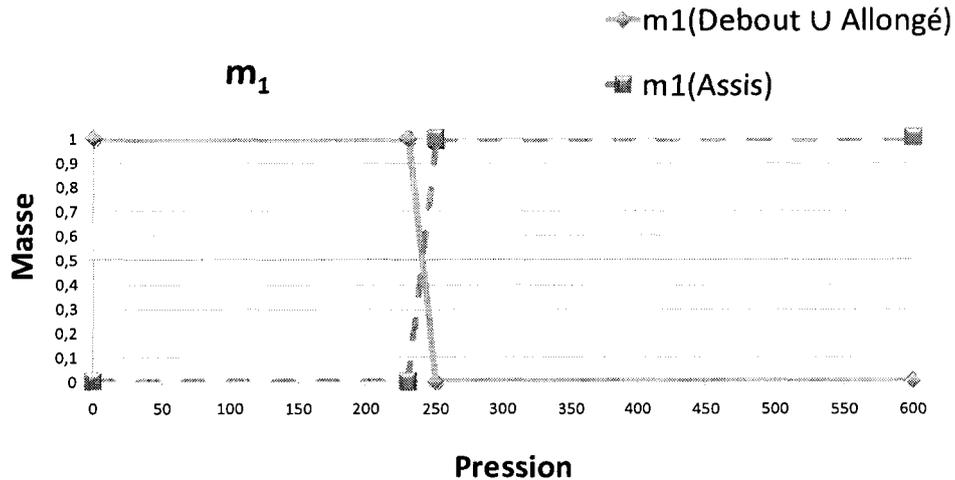


Figure 51 : Fonction de masse du capteur de pression.

Cette première fonction de masse m_1 est à l'image des fonctions d'appartenance dans le domaine de la logique floue. Cette modélisation s'appuie sur des données collectées issues d'une expertise humaine. Cette modélisation est utilisée ici car l'apprentissage n'est pas nécessaire. Les valeurs en abscisse sont ainsi déterminées à partir de l'observation humaine du fonctionnement de ce capteur. Cette modélisation, basée sur les fonctions d'appartenance floue, est utilisée pour la suite de ce chapitre.

Cette première fonction de masse fournit un jeu de masse m_1 portant sur la pression appliquée à une chaise. Cette fonction de masse est directement appliquée à la valeur numérique fournie par le capteur de pression. Nous avons observé expérimentalement que lorsque la pression dépasse 22 Kg (valeur numérique 250), nous pouvons être certains qu'une personne est assise. De la même manière lorsque cette valeur est inférieure à 20 Kg (valeur numérique 230), nous pouvons être certains que la personne n'est pas assise, c'est-à-dire soit debout, soit allongée.

Par exemple, si la pression indiquée est 235, nous obtenons le jeu de masses suivant :

$$\begin{aligned}
 m_1(\text{Assis}) &= 0.25, \\
 m_1(\text{Debout}) &= 0, \\
 m_1(\text{Allongé}) &= 0, \\
 m_1(\text{Assis} \cup \text{Debout}) &= 0, \\
 m_1(\text{Assis} \cup \text{Allongé}) &= 0, \\
 m_1(\text{Debout} \cup \text{Allongé}) &= 0.75, \\
 m_1(\Theta_{\text{Posture}}) &= 0.
 \end{aligned}$$

Ce jeu de masses indique que la personne est dans une phase où elle s'assoit ou s'allonge.

2.2.2 Second capteur retenu : le capteur à vision omnidirectionnelle

Pour compléter les données du capteur de pression, nous avons équipé le plafond de notre pièce d'une Webcam à vision omnidirectionnelle. Ce type de vision s'appuie sur des capteurs composés d'un miroir réfléchissant face à la caméra pour obtenir une image qui est la projection omnidirectionnelle 2D de l'environnement. Ces capteurs omnidirectionnels sont appelés capteurs catadioptriques. Ils présentent l'avantage de fournir une vue globale de l'environnement en une seule acquisition. Autre avantage : ils n'ont aucune partie mobile et donc consomment peu d'énergie (pas de mécanisme de rotation et d'orientation). Par contre, le redressement d'image pose problème. Rappelons que la catadioptrie est la

théorie des systèmes optiques comportant une succession d'éléments réfléchissants et réfractants (dioptries : lentilles...). Dans les systèmes étudiés, il s'agit d'un miroir de révolution convexe associé à une caméra CCD.

Nous utilisons ce capteur afin de connaître tous les éléments d'une pièce en une seule acquisition d'image. De même, les informations pertinentes seront présentes dans une seule image, favorisant les traitements en termes de temps de calcul.

Ce capteur a pour objectif de localiser et de suivre l'individu présent, et ainsi de déterminer sa posture. Afin de détecter et de suivre l'individu dans l'image, nous avons mis en œuvre l'algorithme du MHI (Motion History Images) qui a pour fonction la reconnaissance de silhouette humaine. L'algorithme MHI a initialement été proposé par Davis et Bobick [DAV97]. Ils analysent des flux vidéo contenant une silhouette de corps humain et proposent une représentation spatio-temporelle d'activités humaines [BOB96b][BOB96a][BOB97]. Ils définissent les « images d'énergie de mouvement » (MEI – Motion Energy Image) et les « images de l'historique du mouvement » (MHI – Motion History Image). Les premières, binaires, représentent la position du mouvement dans une séquence d'images. Dans les secondes, les pixels sont des valeurs correspondantes à l'âge du mouvement. Ils sont calculés par un simple remplacement et un opérateur décalage. Si $D(x,y,t)$ est un booléen indiquant si l'intensité lumineuse du point (x, y) à l'instant $t-1$ alors, les pixels de l'image H de l'historique du mouvement à l'instant t sont définis par :

$$H_T(x,y,t) = \begin{cases} T & \text{si } D(x,y,t) = 1 \\ \max(0, H_T(x,y,t-1) - 1) & \text{sinon} \end{cases}$$

Dans cette équation, T est l'âge considéré dans l'historique. L'historique au point (x,y) est affecté à la valeur T si un mouvement est détecté en (x,y) , c'est à dire si $D(x,y,t)=1$. Sinon, l'âge du mouvement augmente et la valeur de $H(x,y,t)$ est décrétementée. L'âge du mouvement en (x,y) est en fait déterminé par :

$$|H_T(x,y,t) - T|$$

L'image E d'énergie de mouvement est alors définie par le seuillage de l'image H à zéro, c'est à dire :

$$E_T(x,y,t) = \bigcup_{i=0}^T D(x,y,t-i) = \begin{cases} 1 & \text{si } H_T(x,y,t) > 0 \\ 0 & \text{sinon} \end{cases}$$

La Figure 52 présente les images d'énergie et les images de l'historique du mouvement d'un geste de la main.

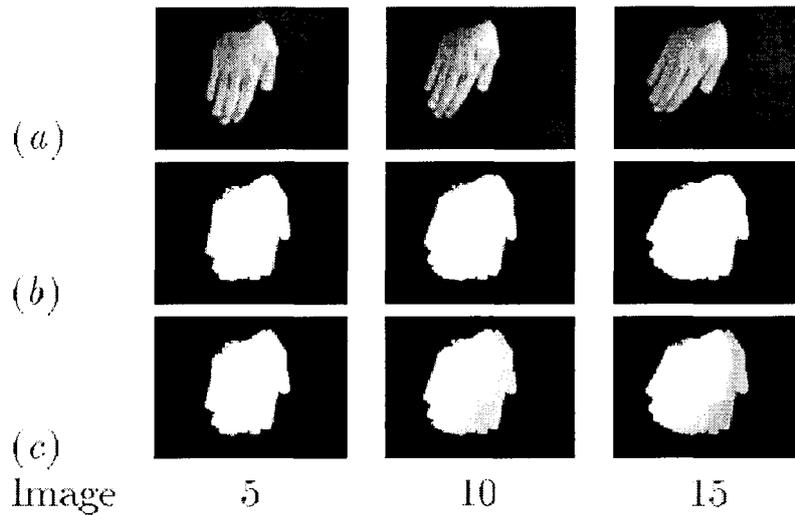


Figure 52 : Exemple d'image d'énergie et d'historique du mouvement
(a) Images originales du geste (b) Images d'énergie (c) Images de l'historique du mouvement.
Figure tirée de [BOB96a].

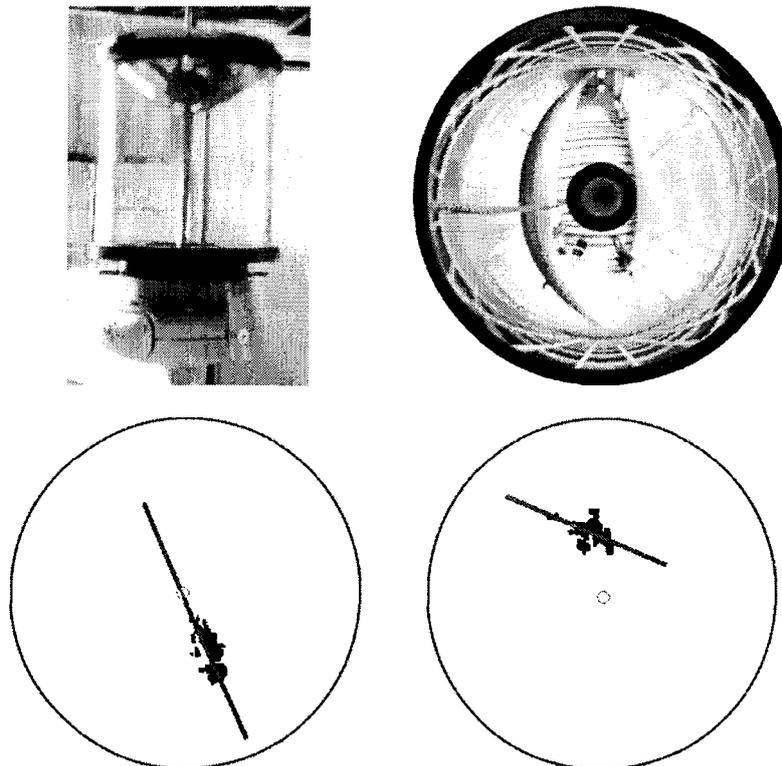


Figure 53 : Le capteur à vision omnidirectionnelle, l'image originale d'une personne debout, son équivalent traité par MHI et le traitement d'une personne allongée.

La Figure 53 présente le montage Webcam et miroir à vision omnidirectionnelle, une image issue de ce montage et deux images traitées par MHI permettant de discerner l'individu et sa posture. Ce type de capteur permet de correctement discerner la posture allongée.

Le traitement MHI permet d'extraire l'ensemble de pixels caractérisant l'individu. Cet amas de n points est ensuite approximé par une droite D au sens des moindres carrés linéaires de la forme : $D: Y = aX + b$

$$\text{avec } a = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \quad \text{et} \quad b = \frac{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}$$

Nous avons observé expérimentalement que lorsque l'individu est debout ou assis, la direction moyenne des points formés par le traitement MHI est orientée vers le centre de l'image omnidirectionnelle (premier cas traité). A contrario, lorsque la personne est allongée, la direction moyenne de l'ensemble des points n'est plus orientée vers le centre de l'image omnidirectionnelle. La distance d en pixel entre la droite approximée et le centre de l'image (voir Figure 54) permettent de définir la fonction de masses m_2 présentée sur la Figure 55. Ainsi, plus la distance entre la droite approximée et le centre de l'image est importante, plus la croyance envers l'hypothèse 'Allongée' est importante.

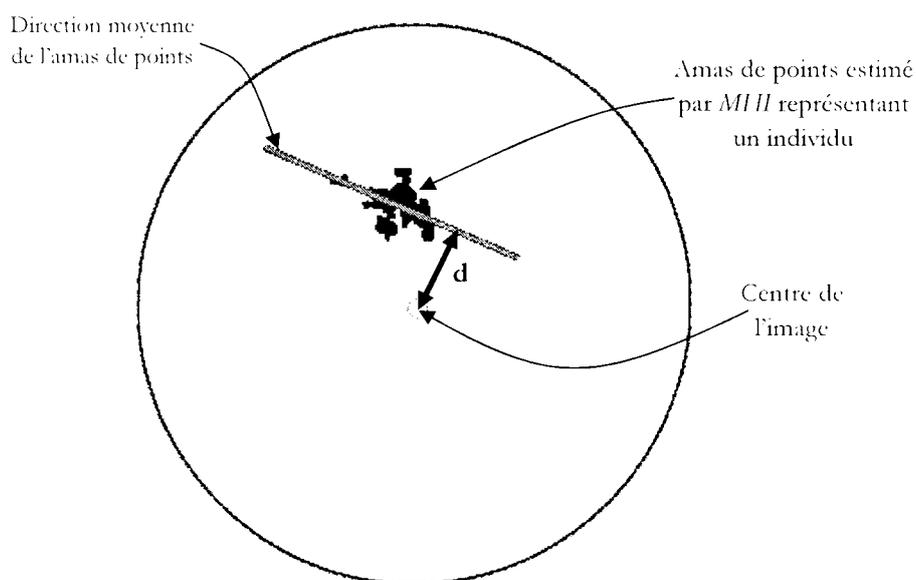


Figure 54 : Estimation de la distance entre la direction moyenne d'un individu et du centre de l'image omnidirectionnelle.

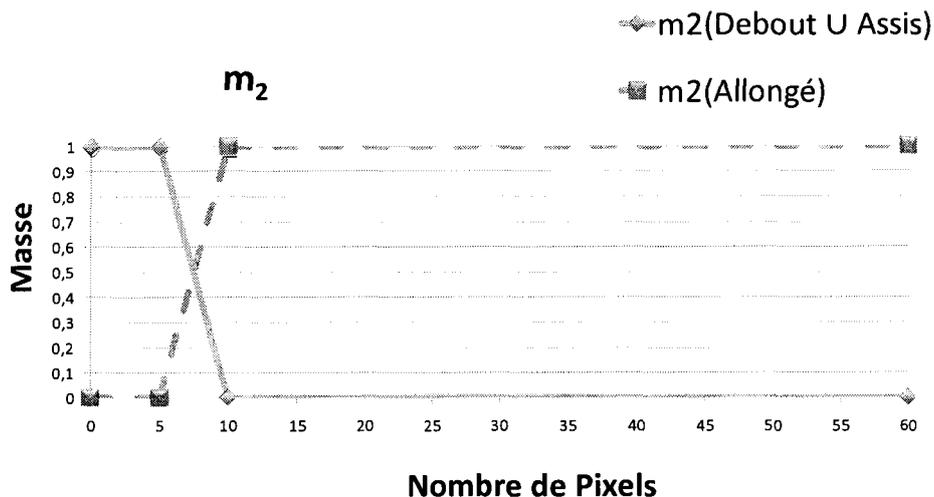


Figure 55 : Fonction de masses du capteur à vision omnidirectionnelle.

2.2.3 Troisième capteur retenu : le capteur accélérométrique

Le troisième capteur est un accéléromètre double axe fixé sur la ceinture de la personne à surveiller. A partir des données brutes issues de l'accéléromètre double axe (accélération relative selon les axes X et Y), il est possible de caractériser les différentes postures d'un individu. Après observation des signaux, nous pouvons apercevoir des changements lors du passage d'une posture à une autre. Lorsque l'individu est debout et/ou en train de marcher, les signaux issus des accéléromètres sont quasi constants (voir Figure 56). Lorsque l'individu s'assoit, nous observons un saut lors du passage de la posture debout à assise (voir Figure 57). De la même manière, lorsque l'individu s'allonge, nous observons un saut brutal dans les signaux d'accéléromètres (voir Figure 58).

Accéléromètre
de marque
Memsic 2125

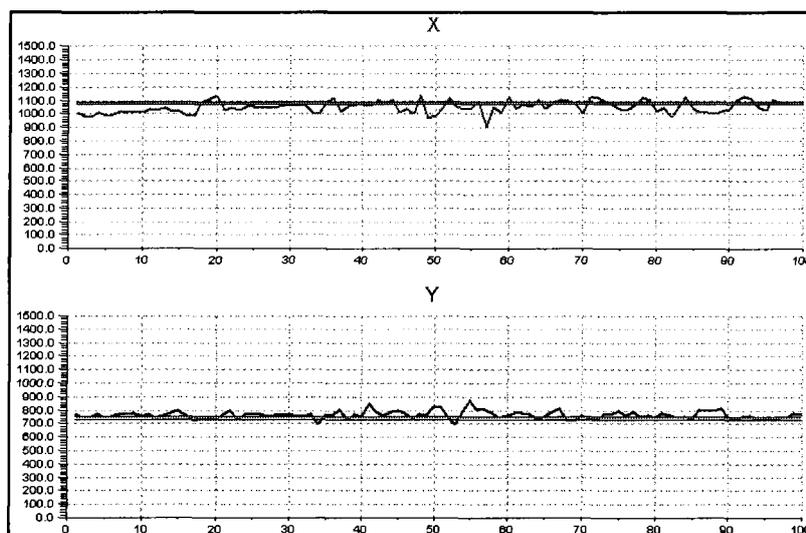


Figure 56 : Données accélérométriques d'un individu marchant.

La stabilité des signaux X et Y caractérise la posture debout.

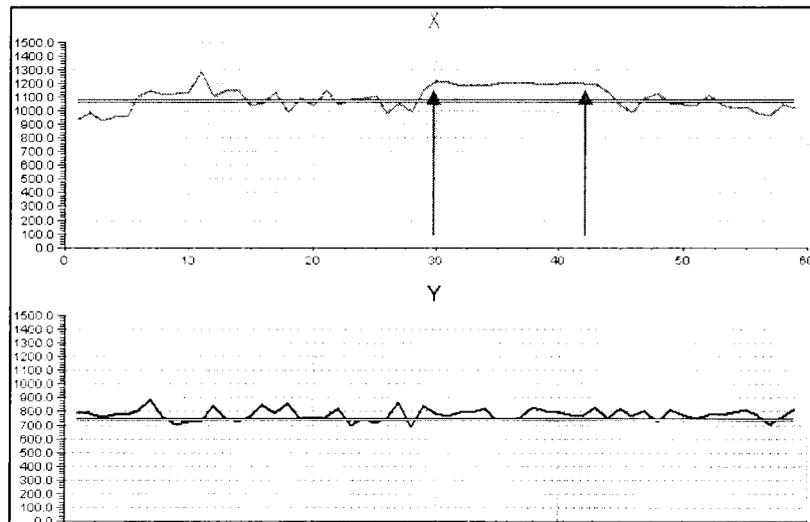


Figure 57 : Données accélérométriques d'un individu qui s'assoit.

Sur le signal X , nous observons que l'individu s'assoit entre les mesures 30 et 42 (au niveau des flèches). Nous voyons alors un petit saut du signal durant cette période qui caractérise la position assise.

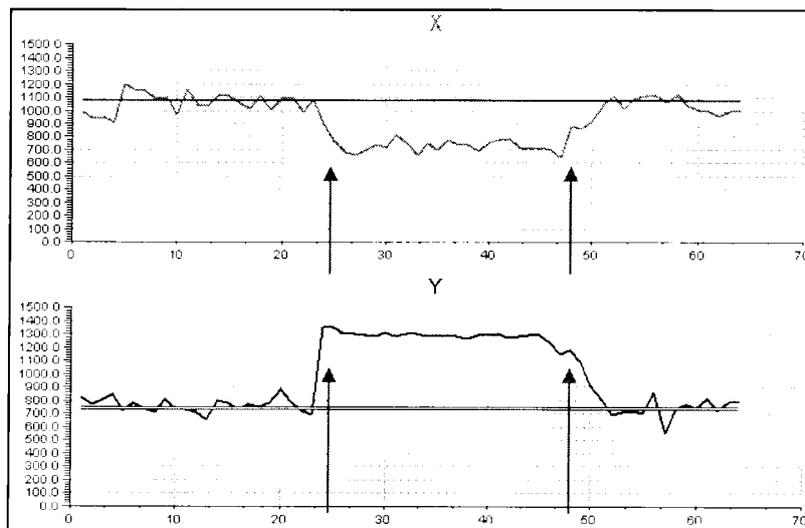


Figure 58 : Données accélérométriques d'un individu qui s'allonge.

Sur les signaux X et Y , nous observons que l'individu s'allonge entre les mesures 25 et 48 (au niveau des flèches). Nous voyons alors un saut important des signaux durant cette période qui caractérise la posture allongée.

Néanmoins, la posture assise est délicate à discerner. La variation d'amplitude n'étant pas assez significative entre une posture debout et une posture assise, nous avons décidé de gérer seulement deux postures grâce à ce capteur. Étant donné que la posture allongée est caractérisée par une variation d'amplitude forte comparée aux postures debout ou assise, ce capteur va permettre de se prononcer sur l'hypothèse allongée et sur la disjonction d'hypothèses *debout* \cup *assis*.

Ainsi, nous utilisons uniquement l'axe Y de l'accéléromètre qui permet de distinguer la posture allongée mais ne peut se prononcer entre les postures debout et assise. Comme précédemment, nous définissons la fonction de masses m_3 permettant de décrire le jeu de masses suivant une posture donnée.

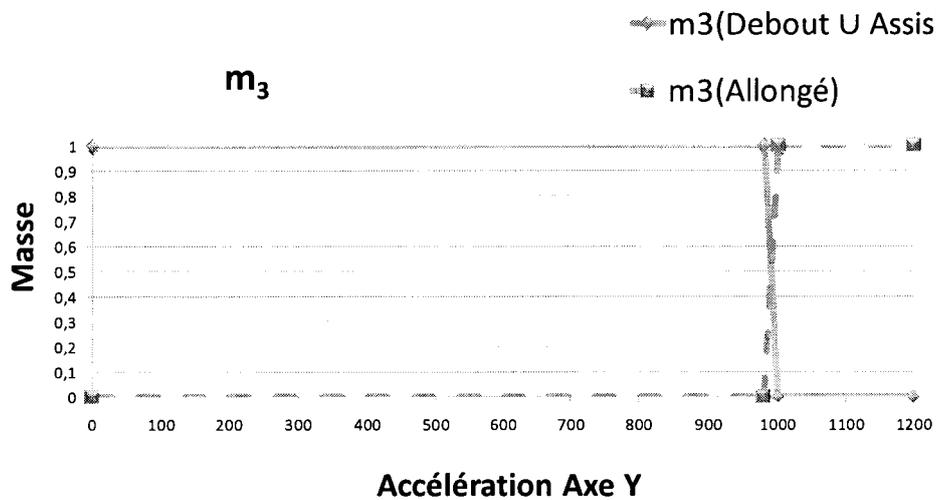


Figure 59 : Fonction de masse de l'accéléromètre placé à la ceinture de l'individu.

Nous venons de présenter les trois capteurs que nous utilisons pour caractériser la posture d'un individu. Les trois capteurs présentés ici sont fortement hétérogènes. A partir d'une phase de prétraitement il est possible d'effectuer un conditionnement des données afin de se restreindre au cadre de discernement global Θ_{Posture} . Une fois ce conditionnement effectué, il est possible de procéder à la fusion des trois jeux de masse m_1 , m_2 et m_3 afin d'obtenir un jeu de masse m_{123} représentant la posture estimée. La fusion est réalisée grâce à l'opérateur de Smets car les capteurs fusionnés sont considérés comme potentiellement non fiables.

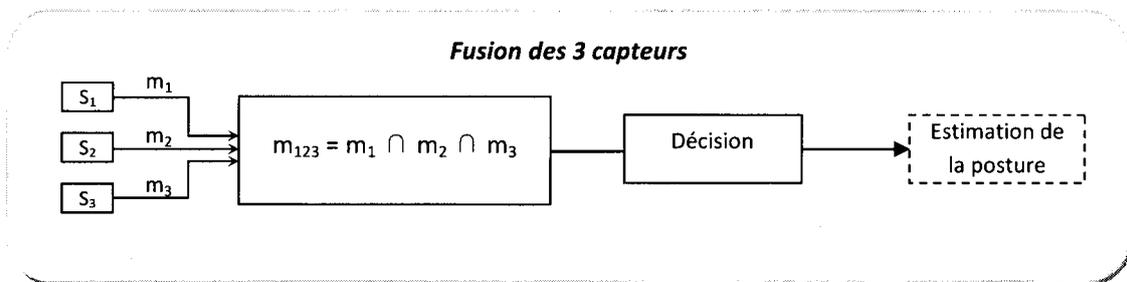


Figure 60 : Détermination de la posture à partir de la fusion de 3 capteurs.

2.2.4 Autres capteurs

Afin de ne pas utiliser inutilement des ressources processeurs et mémoires, l'architecture de fusion de données n'est effective qu'à partir du moment où une personne pénètre à l'intérieur de la pièce équipée. Pour détecter l'entrée d'une personne dans la pièce équipée des trois capteurs vus précédemment, nous avons mis en œuvre deux capteurs sur la porte d'entrée de cette pièce. Etant le point obligatoire de passage de l'individu, ces capteurs permettent de détecter l'entrée ou la sortie d'une personne et autorise ou non la mise en marche de l'architecture de fusion de données.

Le premier capteur que nous avons élaboré est un paillason communicant. Lorsqu'une personne entre ou sort de la pièce, le passage sur le paillason est obligatoire. Le paillason est composé de dalles permettant de détecter l'appui. Cette détection est effectuée grâce à deux feuilles d'aluminium et d'une feuille de

mousse perforée. La dalle est munie d'une résistance de tirage de 10 kOhms et est ensuite reliée à un microcontrôleur. La Figure 61 présente ce principe.

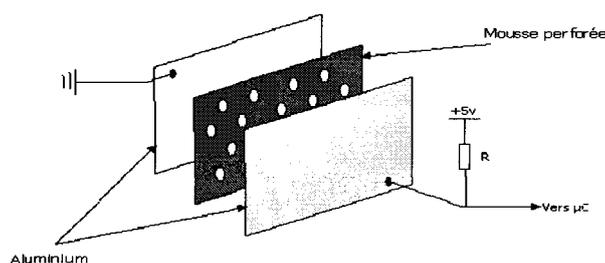


Figure 61 : Principe de fonctionnement d'une dalle du paillason communicant.

Le paillason est composé d'un ensemble de dalles comme celle vue précédemment, ce qui permet de détecter la direction vers laquelle se dirige l'individu (je rentre ou je sors). La direction est déterminée à partir de l'enchaînement temporel de l'activation des dalles du paillason.

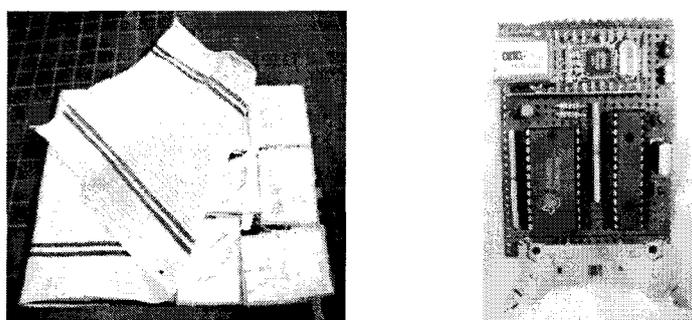


Figure 62 : Le paillason communicant et sa carte de traitement à base de microcontrôleur.

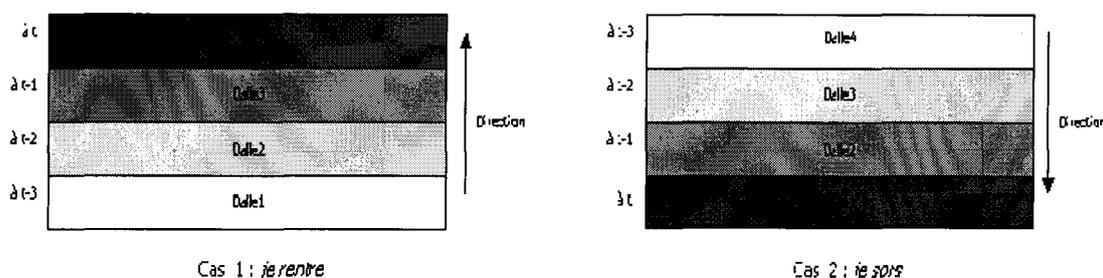


Figure 63 : Principe de détection d'entrée ou sortie dans la pièce à partir du paillason communicant.

Néanmoins, un tel capteur ne peut s'avérer être suffisant. Il arrive que l'individu passe au-dessus du paillason ou que l'enchaînement de l'appui sur les dalles ne permette pas de déterminer la direction de l'individu. Nous avons ainsi mis en œuvre un second capteur permettant de pallier ce problème.

Le second capteur mis en œuvre s'appuie sur une ligne de huit thermopiles capable de détecter l'infrarouge dans la bande de 2 à 22 μm , correspondant aux radiations de chaleur humaine. Les capteurs pyroélectriques habituellement utilisés dans les systèmes anti-intrusions présentent l'inconvénient de ne détecter que des variations de température et ne peuvent faire office que de détecteur de mouvement. La thermopile utilisée ici permet de détecter une source de chaleur statique. Chaque thermopile possède un angle de vue d'environ 12°.

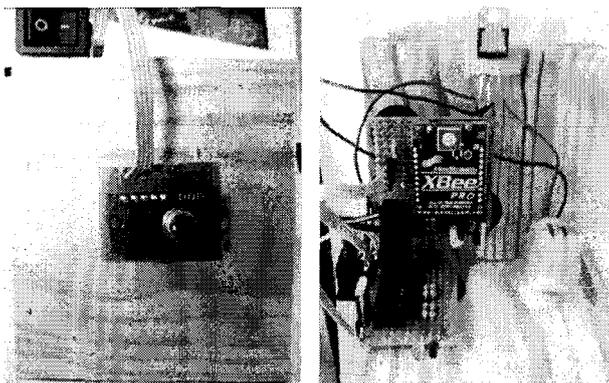


Figure 64 : Le capteur à base de thermopiles et son microcontrôleur chargé du traitement des données.

Ce capteur est placé sur le haut du cadre de la porte d'entrée de la pièce. Orienté vers le bas, ce capteur permet de détecter le sens de passage d'un individu à partir de l'analyse du mouvement de chaleur.

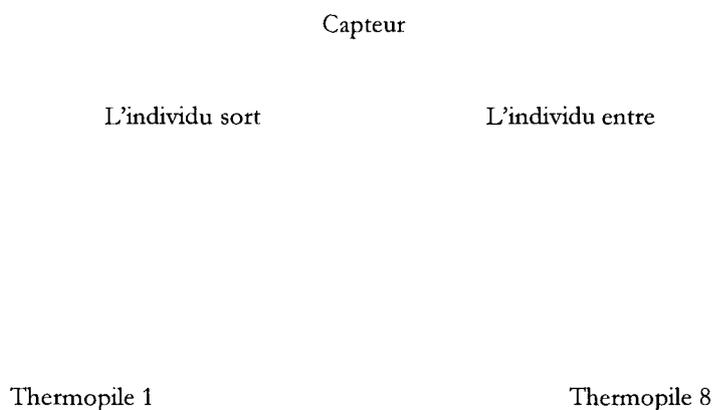


Figure 65 : Principe de détection d'entrée ou sortie dans la pièce à partir du capteur à thermopiles.

Si le sens de variation des thermopiles va de 1 vers 8, alors l'individu entre, sinon il sort.

2.3 Résultats expérimentaux

Nous avons expérimentalement mis en œuvre l'ensemble des capteurs vus précédemment. Les données issues des capteurs sont centralisées sur un même ordinateur chargé de les agréger. Au niveau électronique, nous avons développé une carte d'acquisition de données accélérométriques et de pression basée sur un microcontrôleur. Celui-ci a pour charge d'effectuer la lecture des capteurs et de les transmettre à l'ordinateur par liaison série. La webcam est directement reliée à l'ordinateur par liaison USB.

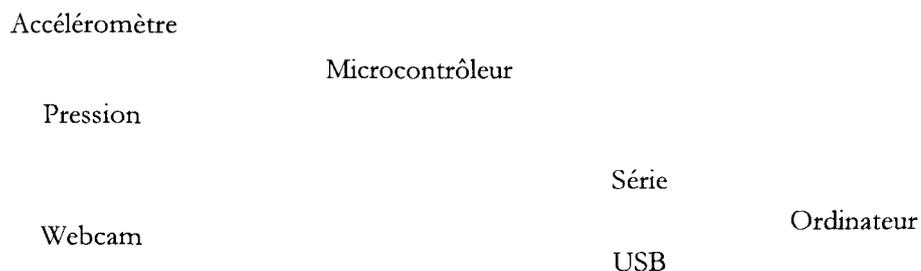


Figure 66 : Architecture matérielle d'acquisition des 3 capteurs.

La carte à base de microcontrôleur est présentée sur la Figure 67. Cette carte permet d'acquérir les données issues du capteur de pression et de l'accéléromètre. Cette carte sert de prétraitement des données brutes capteurs avant de les émettre sur le port série de l'ordinateur.

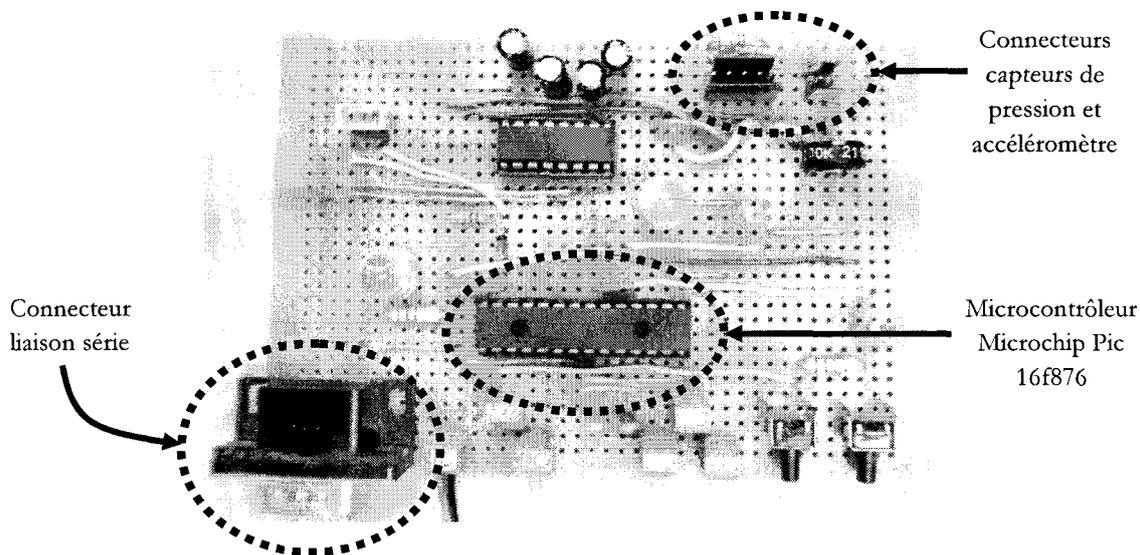


Figure 67 : Carte prototype d'expérimentation pour l'acquisition des données accélérométriques et de pression.

L'architecture logicielle de gestion des données capteurs est présentée sur la Figure 68.

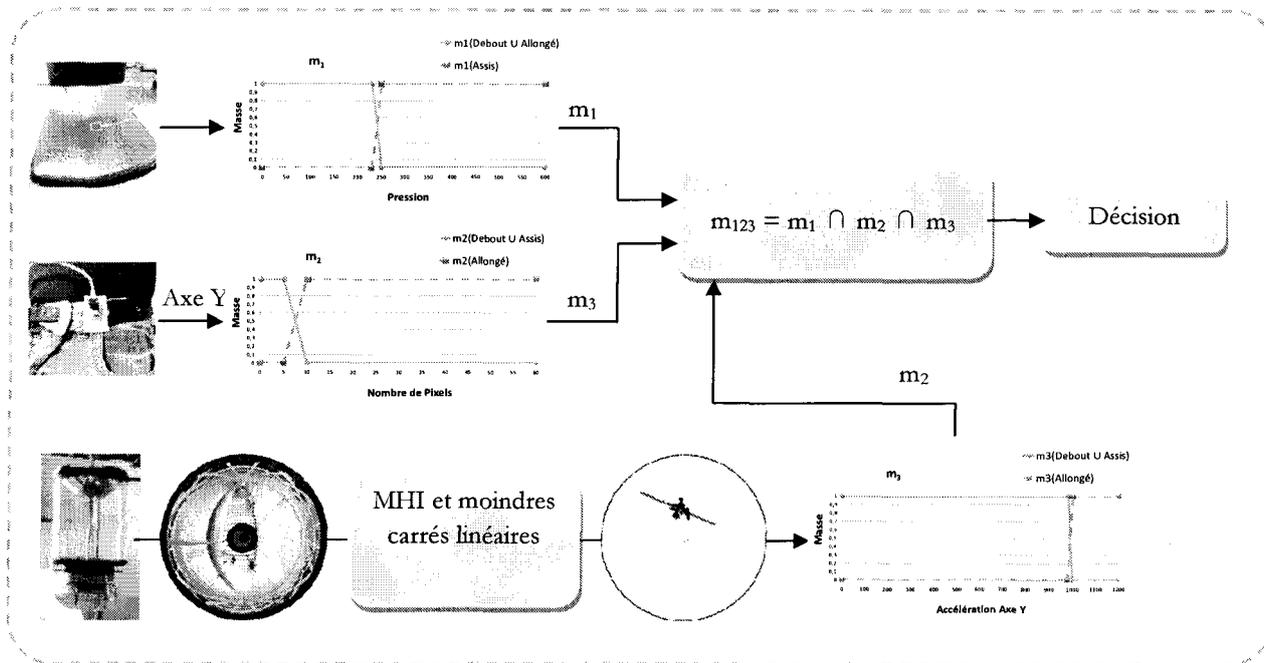


Figure 68 : Architecture logicielle de fusion des 3 capteurs.

Scénario 1 :

Nous avons effectué un ensemble de relevé à partir de chacun des capteurs. Le scénario adopté par l'individu pendant le relevé de mesures est le suivant :

- L'individu est debout pour $t \in [1,10]$,
- L'individu s'assoit pour $t \in [11,17]$,
- L'individu s'allonge pour $t \in [18,46]$,
- L'individu s'assoit pour $t \in [47,61]$,
- L'individu se met debout à nouveau pour $t \in [62,69]$.



Au final, nous obtenons un ensemble de mesures pour chacun des capteurs. Les mesures obtenues sont présentées sur la Figure 69.

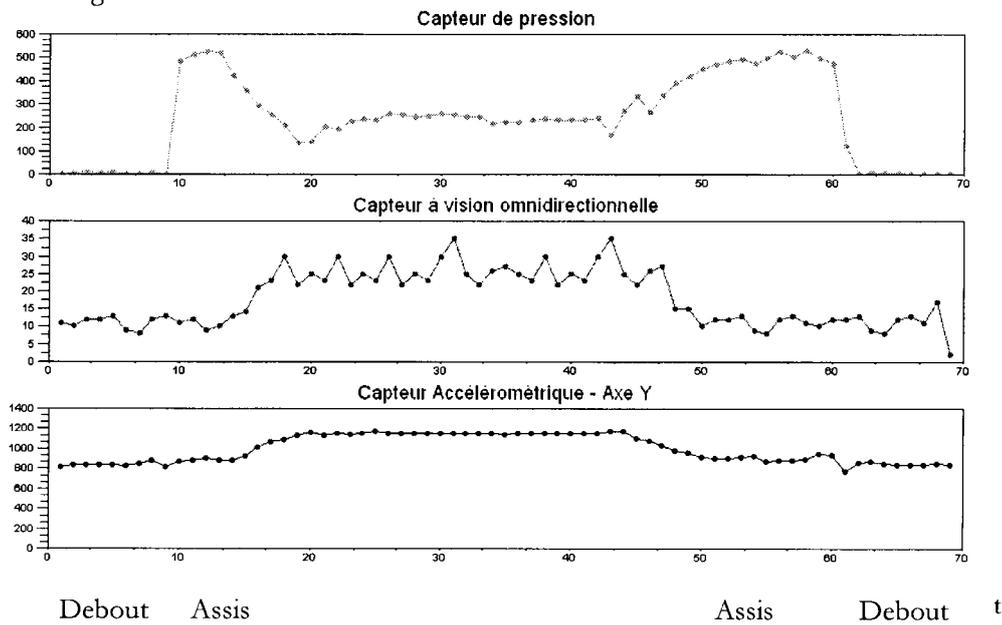


Figure 69 : Signaux expérimentaux obtenus à partir des 3 capteurs pour le premier scénario.

La base expérimentale de mesures est composée de 69 mesures à intervalle régulier. La fréquence d'acquisition est de 3 Hz, environ 3 mesures par seconde pour chacun des capteurs. La durée totale de ce scénario est de 23 secondes.

Conversion numérique vers symbolique

Pour chacune des mesures effectuées, nous réalisons une conversion numérique vers symbolique afin d'obtenir un jeu de masses correspondant.

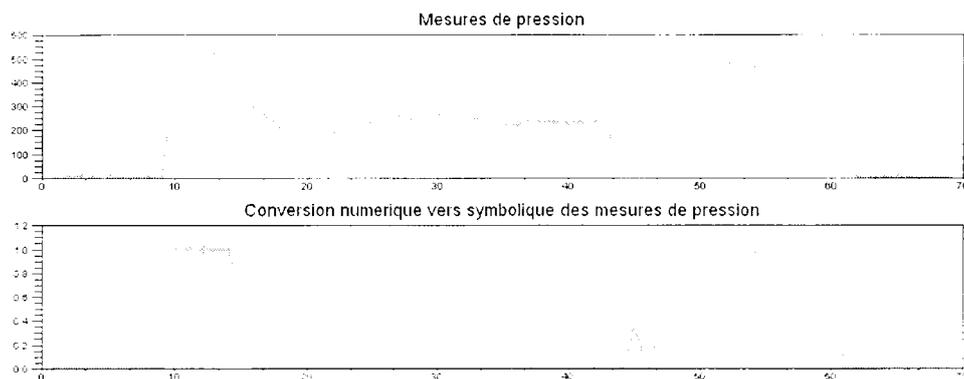


Figure 70 : Conversion numérique vers symbolique des mesures de pression.

La conversion numérique vers symbolique des données issues du capteur de pression permet d'apercevoir que ce capteur ne peut se prononcer que sur une seule hypothèse singleton '*Assis*', et sur la disjonction d'hypothèses '*Debout* \cup *Allongé*' (voir Figure 70). Lorsque nous superposons les mesures expérimentales de ce capteur (première portion de la figure) avec la conversion numérique vers symbolique obtenue pour chacun des points du signal, nous observons que l'enchaînement des postures semble correspondre avec le scénario prédéfini. Sur la Figure 70, la seconde portion correspond au jeu de masses m_1 . Les mesures vertes correspondent à l'hypothèse $m_1(\text{Assis})$ tandis que les mesures jaunes correspondent à la disjonction d'hypothèses $m_1(\text{Debout} \cup \text{Allongé})$. La prise de décision pignistique sur ce jeu de masse m_1 de la Figure 71 permet d'apercevoir plus nettement les hypothèses les plus vraisemblables.

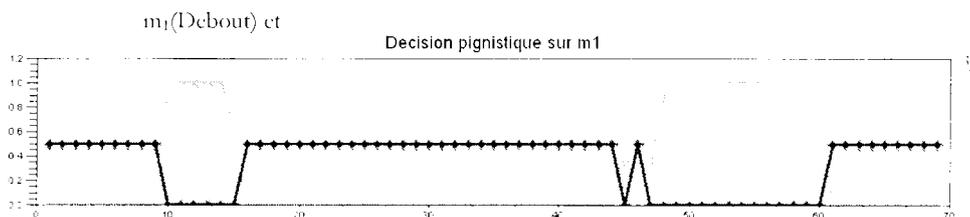


Figure 71 : Prise de décision pignistique sur le jeu de masse m_1 .

La prise de décision pignistique appliquée au jeu de masse m_1 de la Figure 70 est présentée sur la Figure 71. Nous observons que ce capteur permet de décider sur l'hypothèse '*Assis*' (mesures vertes) alors qu'il ne peut choisir entre les hypothèses '*Debout*' et '*Allongé*', qui apparaissent ici en bleu et magenta. Ainsi lorsque la masse $m_1(\text{Debout} \cup \text{Allongé}) = 1$, la décision ne peut être effectuée entre les hypothèses *Debout* et *Allongé*, d'où $P_1(\text{Debout}) = P_1(\text{Allongé}) = 0.5$.

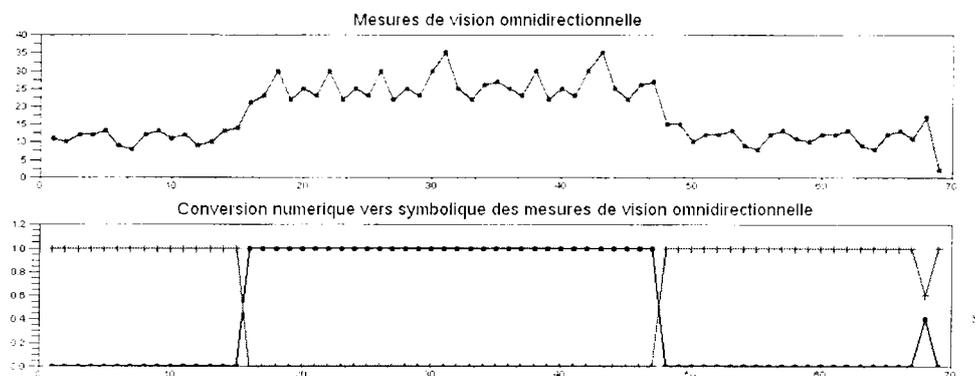


Figure 72 : Conversion numérique vers symbolique des mesures issues de la vision omnidirectionnelle.

De la même manière que pour le capteur de pression, les données issues du capteur à vision omnidirectionnelle et plus particulièrement du traitement MHI donnent lieu à un jeu de masses m_2 . L'objectif est le même, à savoir obtenir un jeu de masse à partir des données expérimentales fournies par ce capteur. L'objectif de ce capteur est de se prononcer soit sur l'hypothèse singleton '*Allongé*' ou sur la disjonction d'hypothèses '*Debout* \cup *Assis*'.

La Figure 72 présente la conversion symbolique des mesures effectuées grâce au capteur à vision omnidirectionnelle. Cette conversion permet d'obtenir le jeu de masse m_2 à partir de la fonction de masse définie Figure 55. Sur la première portion, nous retrouvons les mesures expérimentales effectuées par ce capteur. Sur la seconde portion, nous trouvons la conversion numérique correspondant au jeu de masses m_2 . Les mesures bleues correspondent à l'hypothèse $m_2(\textit{Allongé})$ tandis que les mesures rouges correspondent à la disjonction d'hypothèses $m_2(\textit{Debout} \cup \textit{Assis})$.

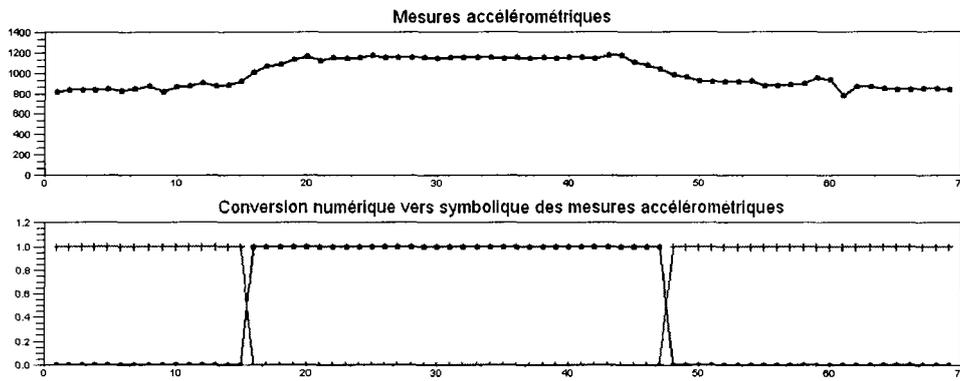


Figure 73 : Conversion numérique vers symbolique des mesures accélérométriques.

Ainsi, la conversion numérique vers symbolique des données issues du traitement MHI permet de correctement caractériser les postures auxquelles sont sensibles ce capteur, c'est-à-dire soit '*Allongé*' soit '*Debout ou Assis*'.

De même, la conversion symbolique des mesures de l'accéléromètre permet d'apercevoir la sensibilité de ce capteur à la posture '*Allongé*' et au doute entre '*Assis*' et '*Debout*'. La Figure 73 présente le résultat de cette conversion. Elle est comparable à la conversion symbolique de données issues du traitement MHI. Sur la première portion, nous retrouvons les mesures expérimentales effectuées par ce capteur. Sur la seconde portion, nous trouvons la conversion numérique correspondant au jeu de masses m_3 . Les mesures bleues correspondent à l'hypothèse $m_3(\textit{Allongé})$ tandis que les mesures rouges correspondent à la disjonction d'hypothèses $m_3(\textit{Debout} \cup \textit{Assis})$.

L'objectif de cet ensemble de capteur est de renforcer la croyance que nous pouvons avoir envers une posture réalisée à un instant t . Ainsi, la fusion de ces trois capteurs, dans le cadre de la théorie de l'Evidence, est réalisable. Les trois jeux de masse correspondant respectivement aux trois capteurs sont obtenus sur le même cadre de discernement $\Theta_{\textit{Posture}}$. Il est désormais possible de procéder à la fusion des jeux de masse m_1 , m_2 et m_3 en utilisant la combinaison de Smets :

$$m_{123} = m_1 \cap m_2 \cap m_3$$

Le jeu de masse m_{123} permet ainsi de quantifier l'incertitude sur la posture estimée à un instant t . Ici, l'objectif de cette fusion est de déterminer d'une manière fiable et quantitative la posture à un instant t .

Aussi, malgré le fait que certains capteurs ont une connaissance partielle des postures recherchées (le capteur de pression ne peut se prononcer précisément que sur la posture 'Assis' par exemple), la fusion doit permettre de négliger cet aspect à partir de la complémentarité des autres capteurs.

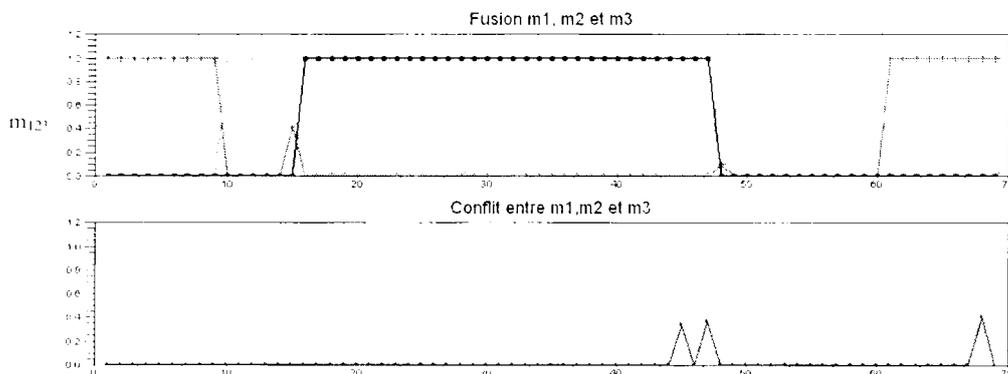


Figure 74 : Fusion des trois jeux de masses m_1 , m_2 et m_3 afin de quantifier l'incertitude sur la posture estimée (première portion du graphique) et progression du conflit lors des fusions (en rouge, seconde portion du graphique).

Il est intéressant de constater que malgré les connaissances partielles de chacun des capteurs, la combinaison des trois capteurs permet d'obtenir une information renforcée sur chacune des hypothèses du cadre de discernement. Pris seuls, les capteurs ne peuvent se prononcer sur la totalité des hypothèses. Nous montrons ainsi l'apport de la fusion multicapteurs lors de l'utilisation de capteurs hétérogènes aux connaissances partielles sur le problème posé.

Il s'avère que le conflit est faible dans la quasi-totalité des fusions (conflit moyen = 0.016), même nul dans certains cas. Ce faible conflit informe que les sources fusionnées sont en parfait accord et que les fonctions de masse ont été correctement établies.

Finalement, la décision retenue est basée sur le calcul de la probabilité pignistique pour chacune des hypothèses.

Postures estimées par la fusion des trois capteurs	Debout		Assis		Assis		Debout		t
	0	10	15	47	61	69			
Postures lors du scénario réellement mis en œuvre	0	10	17	46	61	69			t

Figure 75 : Décision par calcul du maximum de probabilité pignistique et comparaison entre les postures estimées par fusion et le scénario réellement mis en œuvre.

La décision pignistique permet d'extraire la posture la plus crédible à un instant t . Cette décision est effectuée en gardant la posture ayant la plus forte probabilité pignistique.

D'une manière générale, la prise décision pignistique permet de correctement estimer la posture de l'individu à un instant t . Néanmoins, nous observons un décalage temporel lors du passage à la posture 'allongée'.

Ces décalages temporels s'expliquent par le fait que les données capteurs sont désynchronisées. En effet, dans certains cas, nous avons expérimentalement observé que les capteurs peuvent avoir un certain retard lorsqu'ils fournissent leurs données. Le capteur à vision omnidirectionnelle est souvent à l'origine de ces décalages. Ceci est dû aux prétraitements et traitements que subissent les images extraites de la caméra. Ainsi, lorsqu'une nouvelle posture débute, le capteur à vision omnidirectionnelle peut informer soit de la posture précédente soit une phase de doute entre deux postures.

Pour ce premier scénario, nous parvenons à déterminer la posture d'un individu parmi trois postures (debout, assis et allongé) en positionnant ces capteurs sur l'individu et dans l'habitat. Malgré les connaissances partielles de chacun des capteurs, leur combinaison permet d'estimer un ensemble de postures, qui sont difficilement estimables à partir d'un seul des capteurs choisis ici. Il est ainsi possible d'intégrer un nombre important de capteurs au sein d'un habitat communicant afin de rendre plus riche l'information portée par un unique capteur même si ces capteurs fournissent des données fortement hétérogènes : dans ce premier scénario, l'analyse provient des données issues d'une caméra, d'un capteur de pression et d'un accéléromètre. Cette combinaison permet d'apporter une plus grande certitude envers la posture estimée.

Scénario 2 :

Cette seconde partie présente un scénario identique au scénario 1 auquel est ajouté un dysfonctionnement capteur. La Figure 76 présente les mesures expérimentales. Les mesures du troisième capteur ont subi un dysfonctionnement (mesures apparaissant en magenta). L'objectif de ce scénario est de montrer l'impact d'une défaillance sur la fusion des trois capteurs et sur la prise de décision pignistique finale.

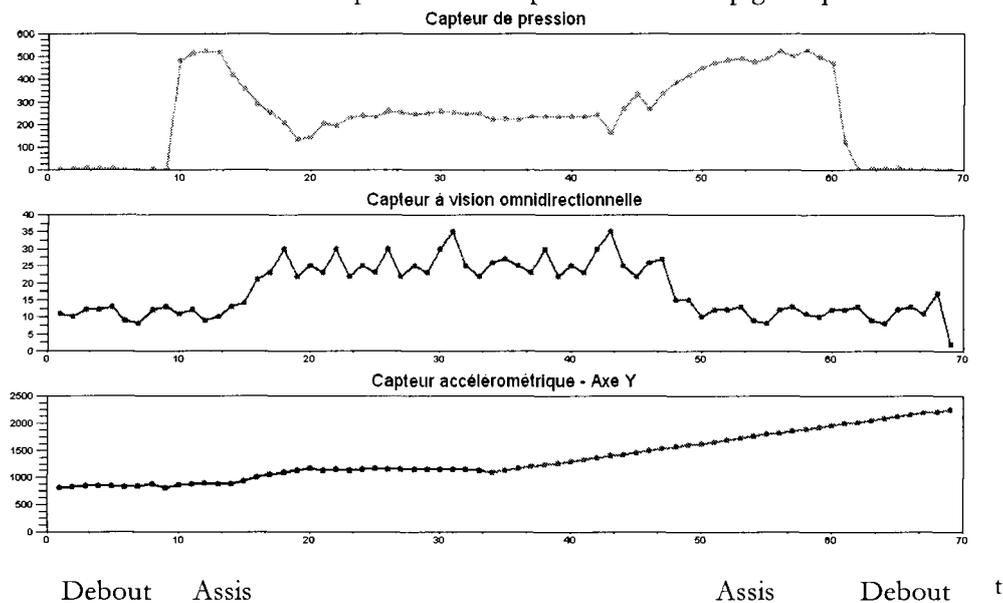


Figure 76 : Signaux expérimentaux avec ajout d'un dysfonctionnement sur le troisième capteur.

Comme précédemment, à partir de ces mesures expérimentales, il est possible d'obtenir les trois jeux de masse m_1 , m_2 et m_3 qui sont fusionnés en utilisant la combinaison de Smets.

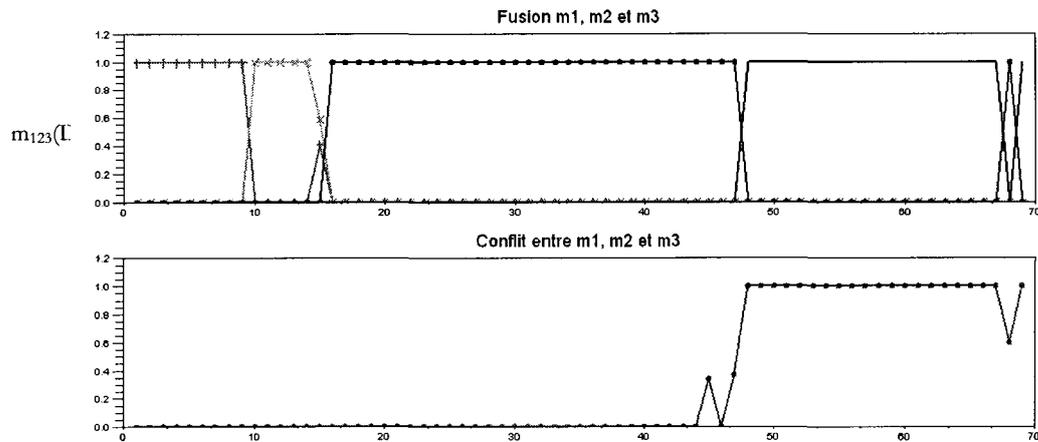


Figure 77 : Fusion des trois jeux de masses m_1 , m_2 et m_3 , la transformation pignistique du jeu de masse fusionné et la prise de décision correspondante.

L'ajout d'un dysfonctionnement a un impact important lors de la fusion des trois capteurs. En effet, nous observons une partie d'ignorance totale (mesures noires sur la première portion de la Figure 77) lorsque les capteurs sont en conflit. Le conflit est alors important durant cette période de dysfonctionnement. Comme le conflit est redistribué sur l'ignorance totale, nous observons à cette période qu'aucune hypothèse singleton n'est plus vraisemblable. La prise de décision pignistique permet d'observer l'impact de ce conflit.

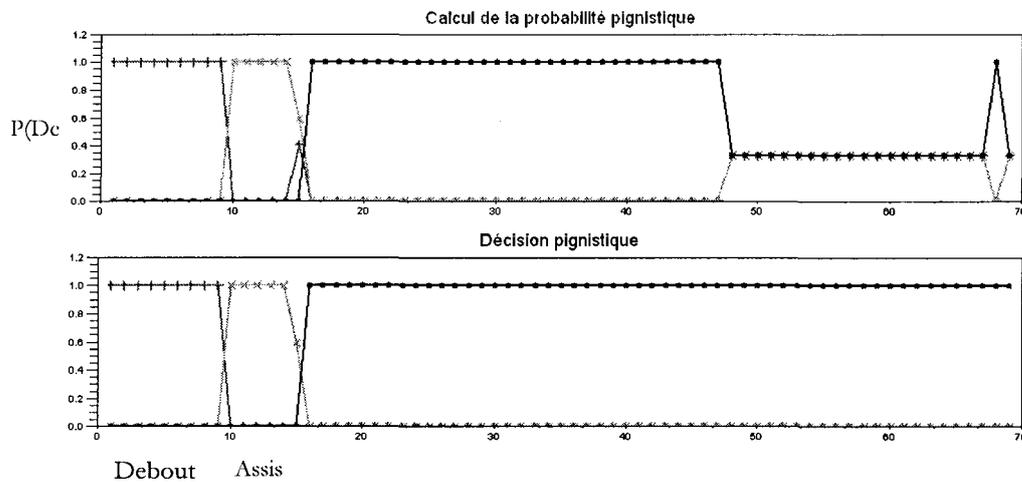


Figure 78 : Décision par calcul du maximum de probabilité pignistique.

La Figure 78 présente dans la partie haute le calcul de la probabilité pignistique sur chacune des hypothèses singletons. Lorsque l'ignorance est totale, le calcul de la probabilité pignistique fournit une équiprobabilité sur les hypothèses singletons ($P(\text{Debout}) = P(\text{Assis}) = P(\text{Allongé}) = 1/3$). Il est alors impossible de choisir l'hypothèse la plus vraisemblable. Dans un tel cas, nous avons choisi par défaut de négliger cette ignorance et pour ne pas prendre de risque d'utiliser la probabilité à l'instant précédent :

Si $P(\text{Debout})(t) = P(\text{Assis})(t) = P(\text{Allongé})(t) = 1/3$ *Alors*
 $\text{DécisionFinale}(t) = \text{DécisionFinale}(t-1)$
Sinon
 $\text{DécisionFinale}(t) = \max(P(\text{Debout})(t), P(\text{Assis})(t), P(\text{Allongé})(t))$
Finsi

Néanmoins, cette solution a l'inconvénient de négliger deux changements de posture. L'estimation des postures est la suivante :

- L'individu est debout pour $t \in [1, 10]$,
- L'individu s'assoit pour $t \in [11, 16]$,
- L'individu s'allonge pour $t \in [17, 69]$.

Il est alors nécessaire de détecter ce dysfonctionnement. Pour cela, nous allons mettre en œuvre la méthode vue au chapitre 3 permettant d'isoler et de pondérer l'influence d'un capteur défectueux au sein d'un processus de fusion.

3 Détection et identification d'un capteur défaillant – Etape E2

L'exemple précédent permet de mettre en avant la difficulté pour un système de prendre une décision lorsqu'un capteur est défaillant. Dans cette section, nous mettons en œuvre notre méthode de détection et d'identification d'un capteur défaillant au sein d'un ensemble de capteurs redondants. Sur des mesures capteurs entachées d'erreurs de fonctionnement, nous appliquons la méthode vue au chapitre 3 s'appuyant sur l'analyse des conflits issus de la fusion deux à deux entre capteurs. Nous présentons ici le résultat de cette méthode appliquée aux mesures de la Figure 76. La chaîne complète appliquée à 3 capteurs est présentée sur la figure suivante :

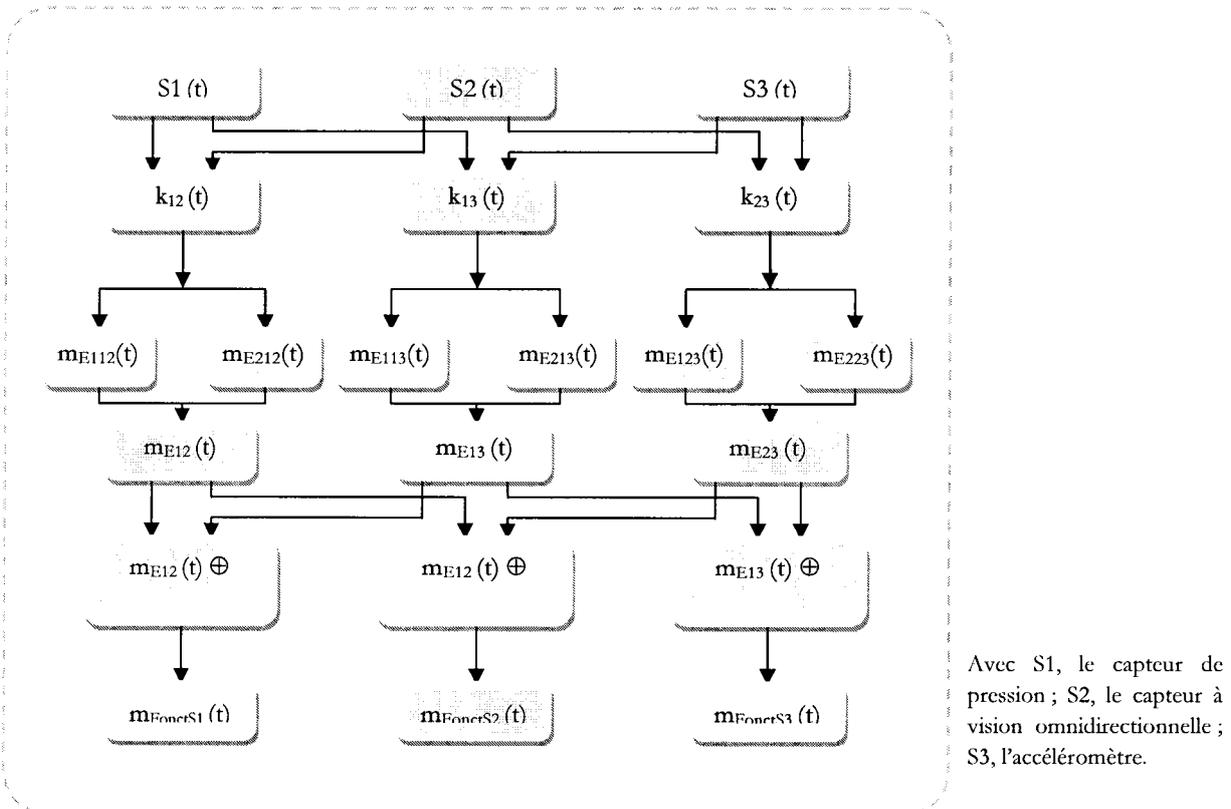


Figure 79 : Détection et identification d'un capteur défaillant parmi les trois capteurs.

3.1 Combinaison deux à deux des capteurs

La première étape de la méthode présentée consiste à effectuer une fusion deux à deux entre chaque capteur afin de vérifier l'existence de conflit entre capteurs. L'objectif de cette étape est de montrer que lorsque les capteurs sont en conflits, il est possible de détecter ces conflits grâce aux mesures de conflit $m_{ii}(O) = k_{ij}$ entre le capteur S_i et S_j . La fusion 2 à 2 permet d'extraire le conflit entre chacun des couples de capteurs.

La fusion 2 à 2 des jeux de masses des N capteurs est assurée par l'opérateur de Smets afin d'isoler le conflit selon :

$$k_{ij} = \sum_{\substack{A \cap B = O \\ i \neq j, 1 \leq i, j \leq N}} m_{S_i}(A) \cdot m_{S_j}(B) \quad \text{with } i=1 \dots N$$

Où k_{ij} représente la mesure de conflit entre les capteurs S_i et S_j . Cet opérateur permet de fusionner 2 à 2 les jeux de masses des N capteurs et offre une valeur de conflit k_{ij} représentative de l'accord entre chacun des capteurs.

Dans le cas de l'utilisation de $N=3$ capteurs, nous obtenons $C^2_3=3$ mesures de conflit :

- k_{12} représente le conflit entre S_1 et S_2 ,
- k_{13} représente le conflit entre S_1 et S_3 ,
- k_{23} représente le conflit entre S_2 et S_3 .

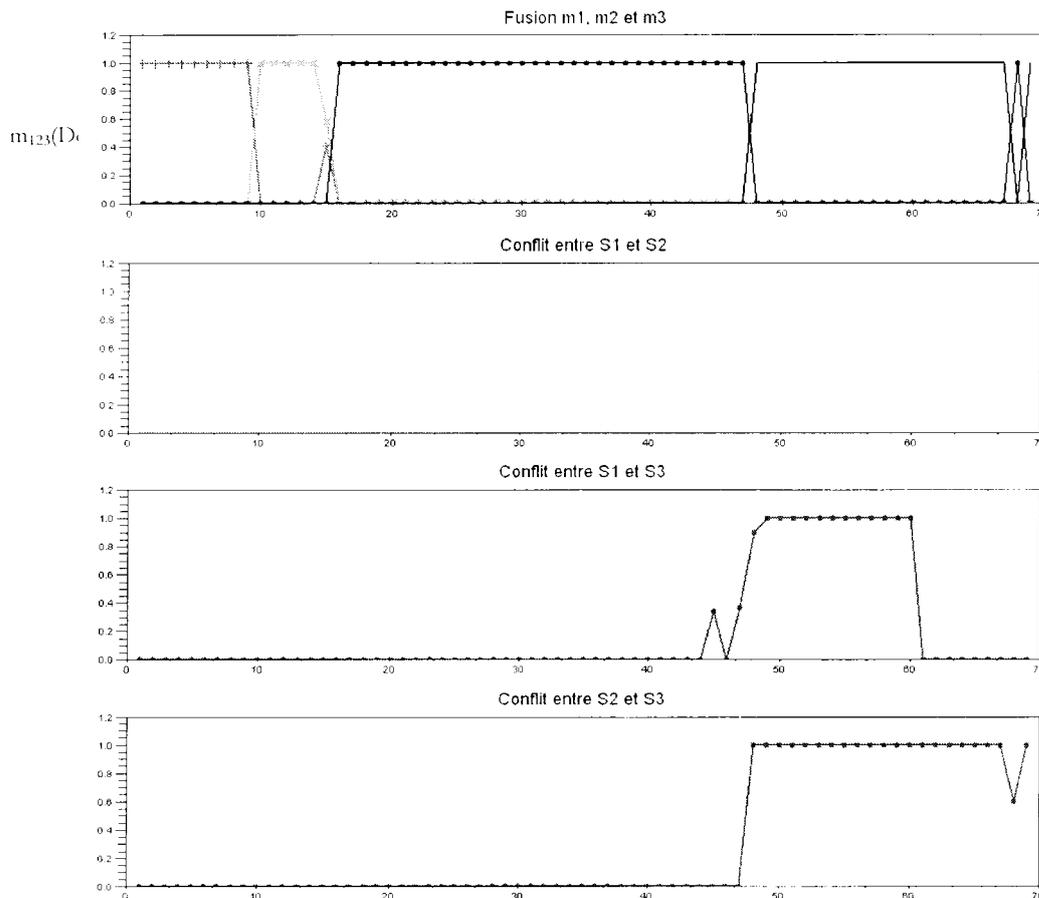


Figure 80 : l'ensemble de conflits résultants de la fusion deux à deux entre les capteurs.

La Figure 80 présente l'ensemble des conflits obtenus à partir de la fusion deux à deux entre capteurs. Il est intéressant de constater que le capteur 3 est à l'origine de deux importants conflits. Ceux-ci apparaissent lorsque le capteur 3 est fusionné avec le capteur 1 et le capteur 2. Nous observons que lorsque ces conflits apparaissent, la fusion entre les trois capteurs fournit une masse sur l'ignorance importante. Ceci est dû au fait que l'opérateur de Smets autorise une gestion plus souple du conflit. Ainsi, nous reportons le conflit sur l'ignorance totale $m_{123}(\Theta_{Posture})$.

3.2 Analyse des conflits

A partir des conflits extraits entre chacun des couples de capteurs, il est possible de déterminer de façon automatique si les capteurs sont en accords ou non. En effet, lorsque le conflit est important entre deux capteurs, cette mesure indique qu'il y a probablement un des deux capteurs qui fournit une information erronée. Cette seconde partie de la méthode permet d'analyser les conflits recueillis lors de la fusion entre deux capteurs. Elle permet d'analyser le comportement d'un conflit dans le but de détecter s'il y a un problème entre deux capteurs. Cette étape est importante puisqu'elle permet de valider ou non un conflit entre deux capteurs afin de déterminer ensuite le capteur réellement défaillant.

Pour réaliser l'analyse des conflits résultant des fusions deux à deux entre capteurs, nous appliquons deux experts chargés de se prononcer sur l'amplitude de chacun des conflits et sur leurs durées respectives. Les experts se prononcent sur le caractère significatif du conflit. Le cadre de discernement utilisé $\Theta_{Fonctionnement}$ est composé de deux éléments *OUI* et *NON*, correspondants respectivement aux assertions « *OUI, le fonctionnement est correct* » et « *NON, le fonctionnement n'est pas correct* »

$$\text{Donc } \Theta_{Fonctionnement} = \{OUI, NON\}$$

Le premier expert est en charge de se prononcer sur l'amplitude du conflit. Il fournit un jeu de masse m_{E1ij} se prononçant sur le caractère significatif de l'amplitude du conflit entre les capteurs S_i et S_j . Le second expert est en charge de se prononcer sur la durée du conflit. Il fournit un jeu de masse m_{E2ij} se prononçant sur le caractère significatif de la durée du conflit entre les capteurs S_i et S_j . Ce second expert a la particularité de s'exprimer sur le conflit filtré à l'aide d'un filtre passe-bas. En effet, un filtre passe-bas appliqué à un signal permet de caractériser l'amplitude du signal. Le filtre passe-bas permet dans notre cas d'extraire les conflits de faibles durées. La figure suivante illustre ce principe en appliquant le filtre passe-bas sur les trois signaux de conflits vus Figure 80.

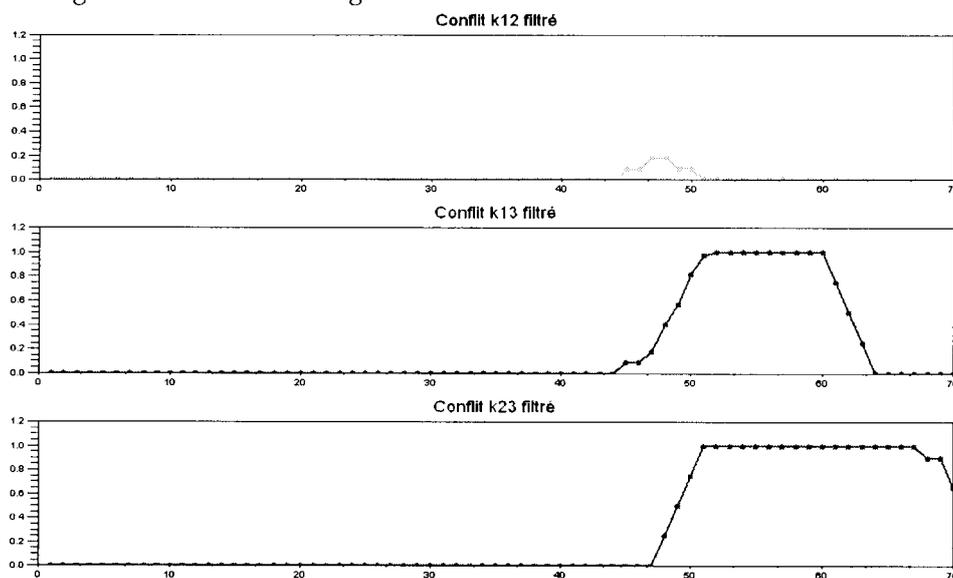


Figure 81 : Ensemble de conflits filtrés par un filtre passe-bas.

Sur la Figure 81, nous observons que les conflits de faibles durées voient leurs amplitudes affaiblies. Ce filtrage permet ainsi de représenter en terme d'amplitude, la durée du conflit, permettant ainsi de négliger les conflits de faible durée. Il est alors possible au second expert de se prononcer en fonction de l'amplitude du signal filtré.

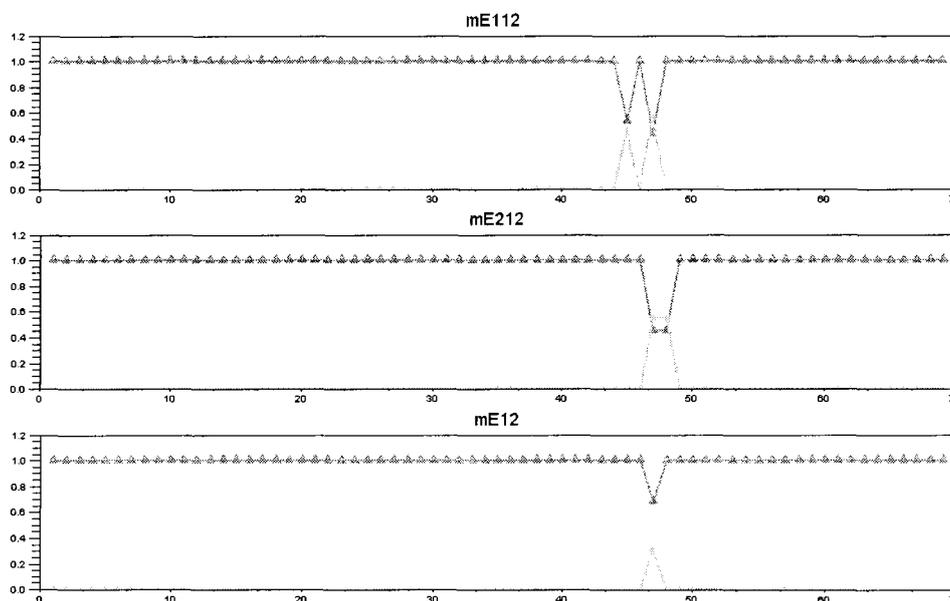


Figure 82 : Application des deux experts sur le conflit k_{12} et leur fusion.

Lorsque ces deux experts sont appliqués sur un conflit faible, les jeux de masse m_{E112} et m_{E212} ont majoritairement une tendance à se prononcer sur l'hypothèse 'OUI, le fonctionnement est correct'. C'est le cas sur la Figure 82. Celle-ci présente le résultat des deux experts appliqués au signal de conflit k_{12} . Le jeu de masse m_{E112} est fourni par le premier expert, avec en vert la masse sur l'hypothèse 'OUI, le fonctionnement est correct' et en jaune l'hypothèse sur le doute $\Theta_{\text{Fonctionnement}}$. Le second jeu de masse m_{E212} est fourni par le second expert. La fusion des deux experts est réalisée grâce à l'opérateur de Smets $m_{E12} = m_{E112} \cap m_{E212}$. Nous constatons que pour la totalité du conflit entre le capteur 1 et le capteur 2, celui-ci n'est pas significatif et permet d'informer qu'il n'y a probablement pas de désaccord entre ces deux capteurs. Il serait alors possible d'arrêter le processus de recherche du capteur défaillant ici, car si les capteurs 1 et 2 ne sont pas défaillants, il ne reste que le troisième capteur. Or, cette méthode peut être appliquée à N capteurs. Il est alors nécessaire de poursuivre l'analyse des conflits afin de précisément déterminer le capteur défaillant.

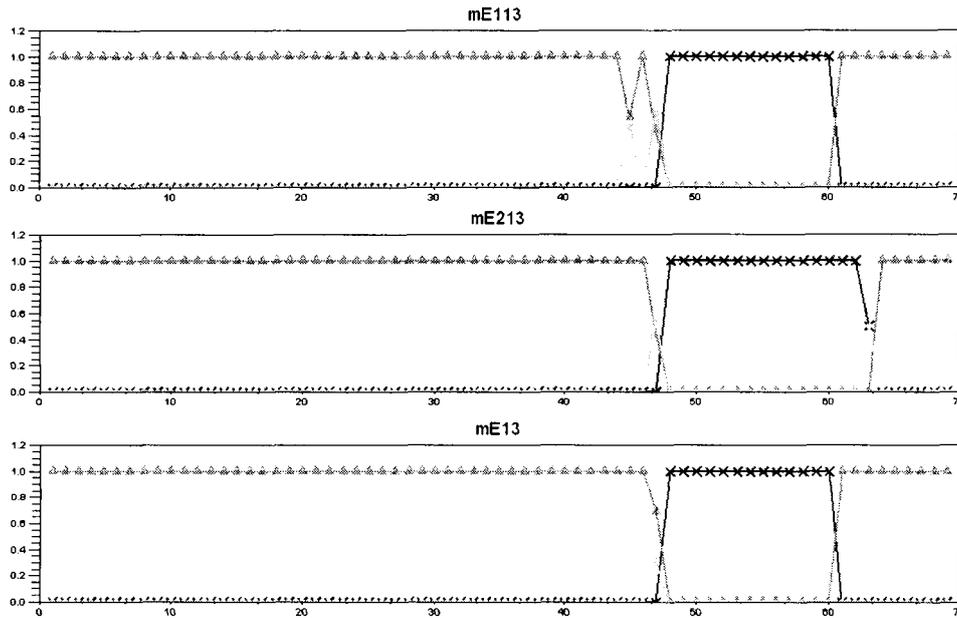


Figure 83 : Application des deux experts sur le conflit k_{13} et leur fusion.

La Figure 83 présente le résultat des deux experts appliqués au signal de conflit k_{13} . Le jeu de masse m_{E113} est fourni par le premier expert, avec en vert la masse sur l'hypothèse 'OUI, le fonctionnement est correct', en rouge la masse sur l'hypothèse 'NON, le fonctionnement n'est pas correct' et en jaune l'hypothèse sur le doute $\Theta_{\text{Fonctionnement}}$. Le second jeu de masse m_{E213} est fourni par le second expert. La fusion des deux experts $m_{E13} \equiv m_{E113} \cap m_{E213}$ permet de constater un conflit élevé entre le capteur 1 et le capteur 3. Ce jeu de masse indique un problème entre les mesures 48 et 60, période durant laquelle la masse sur l'hypothèse NON vaut 1. Cette masse indique un désaccord important entre ces deux capteurs mais ne permet de déterminer quel capteur est fautif.

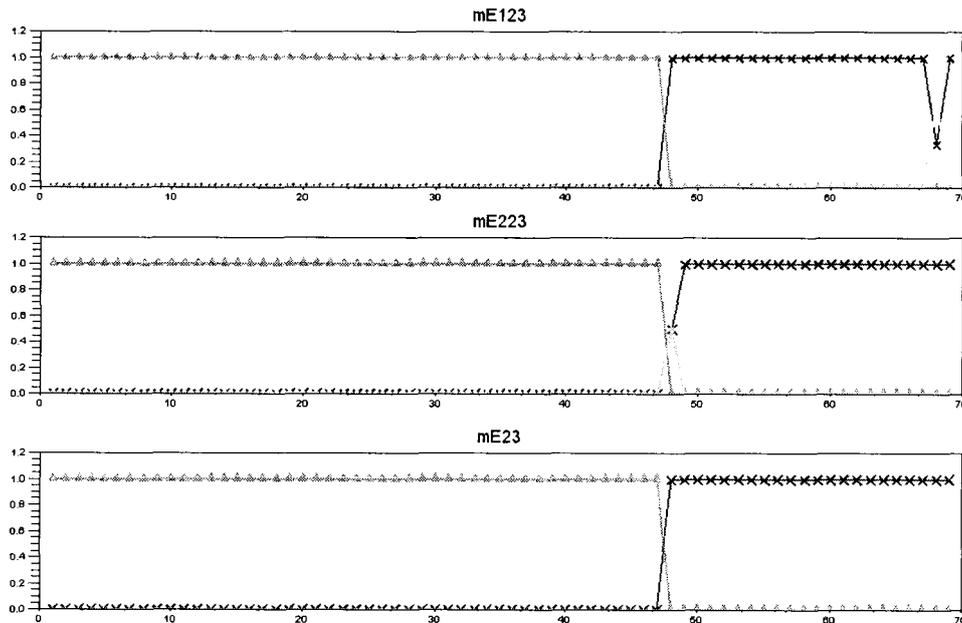


Figure 84 : Application des deux experts sur le conflit k_{23} et leur fusion.

Ainsi, l'analyse du dernier conflit entre les capteurs 2 et 3 va permettre de vérifier que le capteur 3 est défaillant. La Figure 84 présente de même le résultat des deux experts appliqués au signal de conflit k_{23} . Le jeu de masse m_{E113} est fourni par le premier expert selon les mêmes hypothèses vues précédemment. Le second jeu de masse m_{E213} est fourni par le second expert. La fusion des deux experts $m_{E13} = m_{E113} \cap m_{E213}$ permet de constater un conflit élevé entre le capteur 2 et le capteur 3 indiquant un problème entre les mesures 48 et 69, période durant laquelle la masse sur l'hypothèse *NON* vaut 1. Cette masse indique un désaccord important entre ces deux capteurs mais ne permet de déterminer automatiquement quel capteur est fautif. Il est nécessaire d'utiliser les deux jeux de masses pour chacun des conflits afin de les recouper et de déterminer quel capteur est toujours en désaccord avec les autres, synonyme de défaillance.

3.3 Identification du capteur défaillant

L'objectif de cette troisième étape est de déterminer précisément le capteur défaillant. Cette étape consiste à recouper les informations de conflits entre capteurs déterminées précédemment et à fusionner tous les experts se prononçant sur une défaillance potentielle dans laquelle est impliqué un capteur particulier.

D'une manière générale, nous avons vu que pour estimer l'état de fonctionnement d'un capteur S_i , nous utilisons le processus de fusion suivant se basant sur l'opérateur de Smets :

$$m_{FonctS_i} = m_{E1i} \cap m_{E2i} \cap m_{E3i} \cap \dots \cap m_{EiN} \text{ avec } i=1 \dots N$$

Dans notre cas, $N=3$, nous obtenons alors :

$$m_{FonctS1} = m_{E11} \cap m_{E12} \cap m_{E13}$$

$$m_{FonctS2} = m_{E12} \cap m_{E22} \cap m_{E23}$$

$$m_{FonctS3} = m_{E13} \cap m_{E23} \cap m_{E33}$$

avec m_{E11} , m_{E22} , m_{E33} non pris en compte car il ne peut y avoir conflit lors de la fusion de deux jeux de masses identiques.

A partir des jeux de masse m_{E12} , m_{E13} et m_{E23} déterminés précédemment, nous estimons l'état de fonctionnement des trois capteurs. Lorsqu'un capteur est défaillant, les experts permettent d'extraire cette défaillance en se prononçant avec une forte masse sur l'hypothèse '*NON, le fonctionnement n'est pas correct*'.

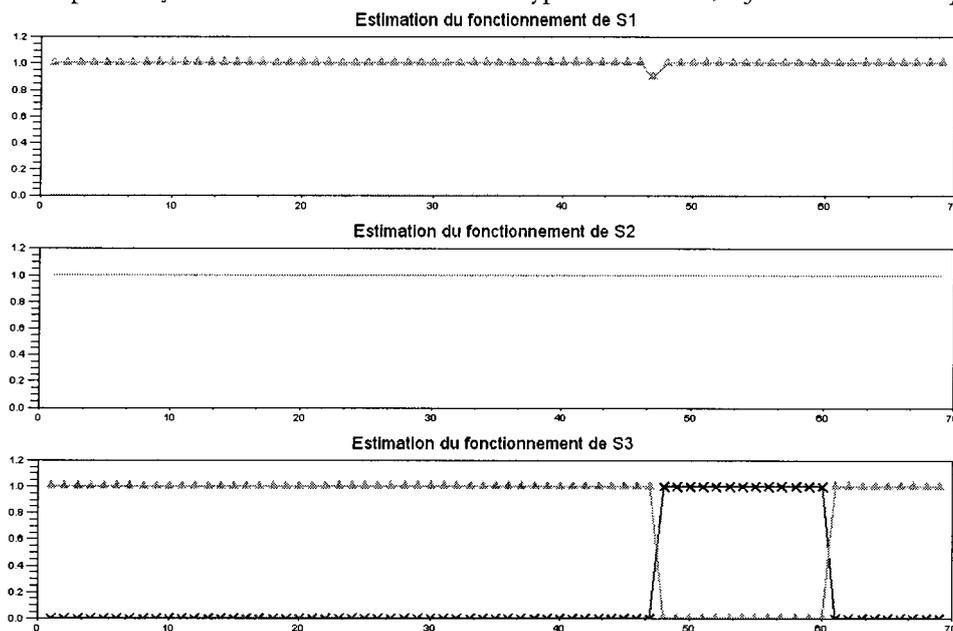


Figure 85 : Estimation de l'état de fonctionnement des 3 capteurs.

La Figure 85 présente les jeux de masse $m_{FonctS1}$, $m_{FonctS2}$, $m_{FonctS3}$ représentant l'état de fonctionnement des 3 capteurs. Nous observons alors que le jeu de masse $m_{FonctS3}$ indique que le fonctionnement du capteur 3 est anormal entre les mesures 48 et 60.

La gestion du conflit à ce niveau est particulière. En effet, lorsqu'un premier capteur est défaillant, il va engendrer du conflit lorsqu'il est fusionné avec un second capteur non-défaillant, c'est à dire que le jeu de masse m_{E12} a une forte masse sur l'hypothèse 'NON'. Ce capteur non-défaillant est lui-même fusionné avec un troisième capteur non-défaillant aussi, ne provoquant alors aucun conflit, c'est à dire que le jeu de masse m_{E23} a une forte masse sur l'hypothèse 'OUI'. Ainsi, lorsque l'on fusionne ces deux jeux de masse afin d'estimer l'état de fonctionnement du second capteur, le jeu de masse $m_{FonctS3} = m_{E12} \cap m_{E23}$ a une forte masse sur le conflit. Nous avons alors choisi de reporter ce conflit $m_{FonctS3}(\emptyset)$ sur $m_{FonctS3}(OUI)$ afin de ne pas générer de fausse alarme et dans ce cas d'identifier le capteur 3 comme défaillant :

$$m_{FonctSi}(OUI) = m_{FonctSi}(OUI) + m_{FonctSi}(\emptyset)$$

Cette manipulation permet de conserver l'une des propriétés élémentaires de la théorie des fonctions de Croyance dans laquelle $\sum_{A \in \Theta} m(A) = 1$.

3.4 Estimation de la fiabilité des capteurs

Lorsqu'un capteur est détecté comme défaillant, il est important de pondérer son influence sur le processus de fusion final afin de ne pas empêcher la prise de décision finale. A ce niveau de la méthode, nous avons identifié le capteur défaillant et la période pendant laquelle celui-ci est défaillant. L'approche ici consiste à pondérer dynamiquement ce capteur pendant le processus de fusion finale en intégrant un coefficient d'affaiblissement reflétant cette défaillance :

$$m_{123}(t) = a_1' \cdot m_{S1}(t) \cap a_2' \cdot m_{S2}(t) \cap a_3' \cdot m_{S3}(t)$$

où a_1' , a_2' , a_3' sont les coefficients d'affaiblissement respectifs aux trois capteurs S_1 , S_2 et S_3 à un instant t . Ces coefficients sont ainsi susceptibles de varier dynamiquement selon l'état de fonctionnement estimé de chacun des capteurs.

Cette dernière étape de la méthode permet d'estimer la fiabilité des capteurs en fonction de leur état de fonctionnement déterminé à l'étape précédente. La fiabilité des capteurs est représentée par un coefficient d'affaiblissement. Ce coefficient d'affaiblissement est estimé à partir de l'état de fonctionnement du capteur testé.

Pour déterminer l'état de fonctionnement d'un capteur S_i , nous nous appuyons sur la probabilité pignistique de sa fonction de masse $m_{FonctSi}$ correspondante. Nous obtenons alors un jeu de probabilité $FonctS_i$ se prononçant sur les 2 hypothèses OUI et NON. La probabilité $FonctS_i(NON)$ reflète d'un état de dysfonctionnement. Ainsi, lorsque $FonctS_i(NON) \neq 0$, le fonctionnement du capteur S_i est anormal. Nous procédons alors à un affaiblissement du jeu de masse produit par ce capteur en fonction de la probabilité $FonctS_i(NON)$ et sur l'état de fonctionnement des $TMax$ mesures voisines. La prise en compte des mesures voisines permet de définir une durée maximale durant laquelle le capteur est considéré totalement défaillant.

Ainsi, le coefficient a_i du capteur S_i se calcule à partir de la formule suivante :

$$\alpha'_i = \frac{TMax_i - D'_i}{TMax_i} \quad \text{with} \quad D'_i = \sum_{k=t}^{TMax_i} FonctS_i(NON)(k)$$

Le coefficient d'affaiblissement a'_i d'un capteur S_i à un instant t est estimé à partir de l'état de dysfonctionnement du capteur à cet instant $FonctS_i(t)$ et sur l'état de dysfonctionnement du capteur aux $TMax$ instants suivants à partir de cet instant t . Cette somme permet d'obtenir $D'_i \in [0, TMax]$.

Lorsque $D'_i = TMax$, le coefficient d'affaiblissement devient nul $a'_i = 0$, le jeu de masse $m_{S_i}(t)$ correspondant au capteur S_i est supprimé de la fusion finale car complètement non fiable.

Lorsque $D'_i = 0$, $a'_i = 1$, le jeu de masse $m_{S_i}(t)$ n'est supprimé de la fusion finale car complètement fiable.

Ainsi, ces coefficients d'affaiblissement permettent de pondérer l'impact d'un capteur défaillant sur le processus de fusion sans le négliger complètement si son état n'est pas totalement défaillant (cas où $0 < a'_i < 1$).

La mise en œuvre de ce calcul du coefficient d'affaiblissement est réalisée sur l'estimation de l'état de fonctionnement des 3 capteurs m_{FonctS_i} vue Figure 85, préalablement convertie en probabilité pignistique $FonctS_i$. L'objectif est ici d'estimer le coefficient d'affaiblissement de chacun des capteurs et d'observer en particulier l'évolution du coefficient d'affaiblissement pour le capteur défaillant.

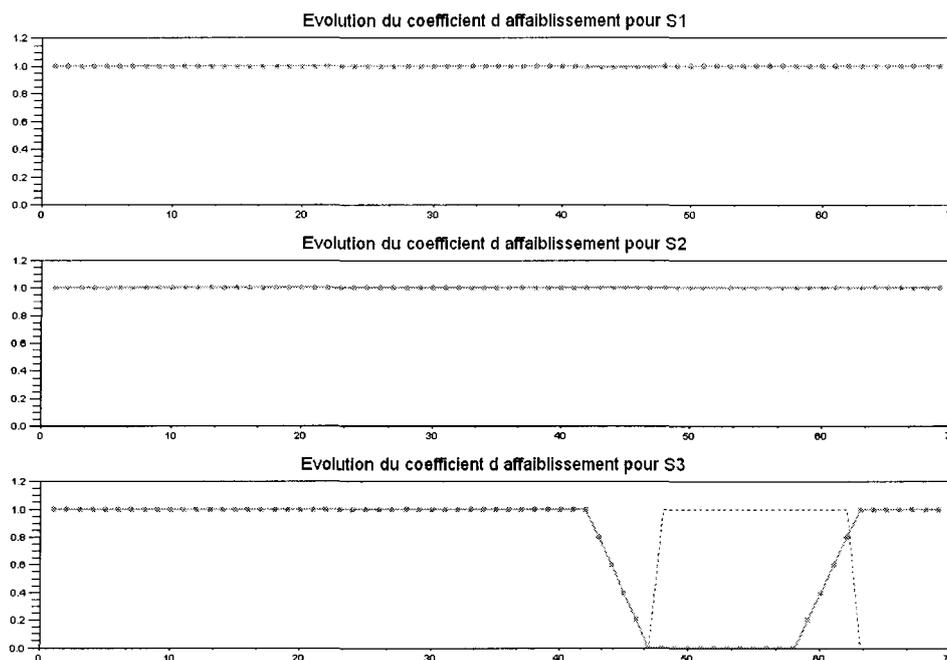


Figure 86 : Estimation du coefficient d'affaiblissement en fonction de l'état de fonctionnement des 3 capteurs.

L'évolution des trois coefficients d'affaiblissement à partir de l'état de fonctionnement estimé des trois capteurs est présentée sur la Figure 86. Lorsque le capteur est détecté comme défaillant, son coefficient d'affaiblissement diminue jusqu'à atteindre 0 (troisième portion de la Figure 86, en vert). Pour comparaison, l'estimation du dysfonctionnement de ce capteur $FonctS_3(NON)$ apparaît en rouge pointillé. Les coefficients d'affaiblissement des capteurs non défaillants restent à 1 comme ils ne sont pas détectés défaillants.

Ces coefficients d'affaiblissement sont injectés dans l'équation de combinaison basée sur l'opérateur de Smets (voir Figure 68), selon :

$$m_{123} = a_1.m_1 \cap a_2.m_2 \cap a_3.m_3$$

Finalement, la fusion des trois capteurs est pondérée dynamiquement en fonction de leurs états respectifs. Cette nouvelle combinaison permet d'observer l'impact du capteur pondéré.

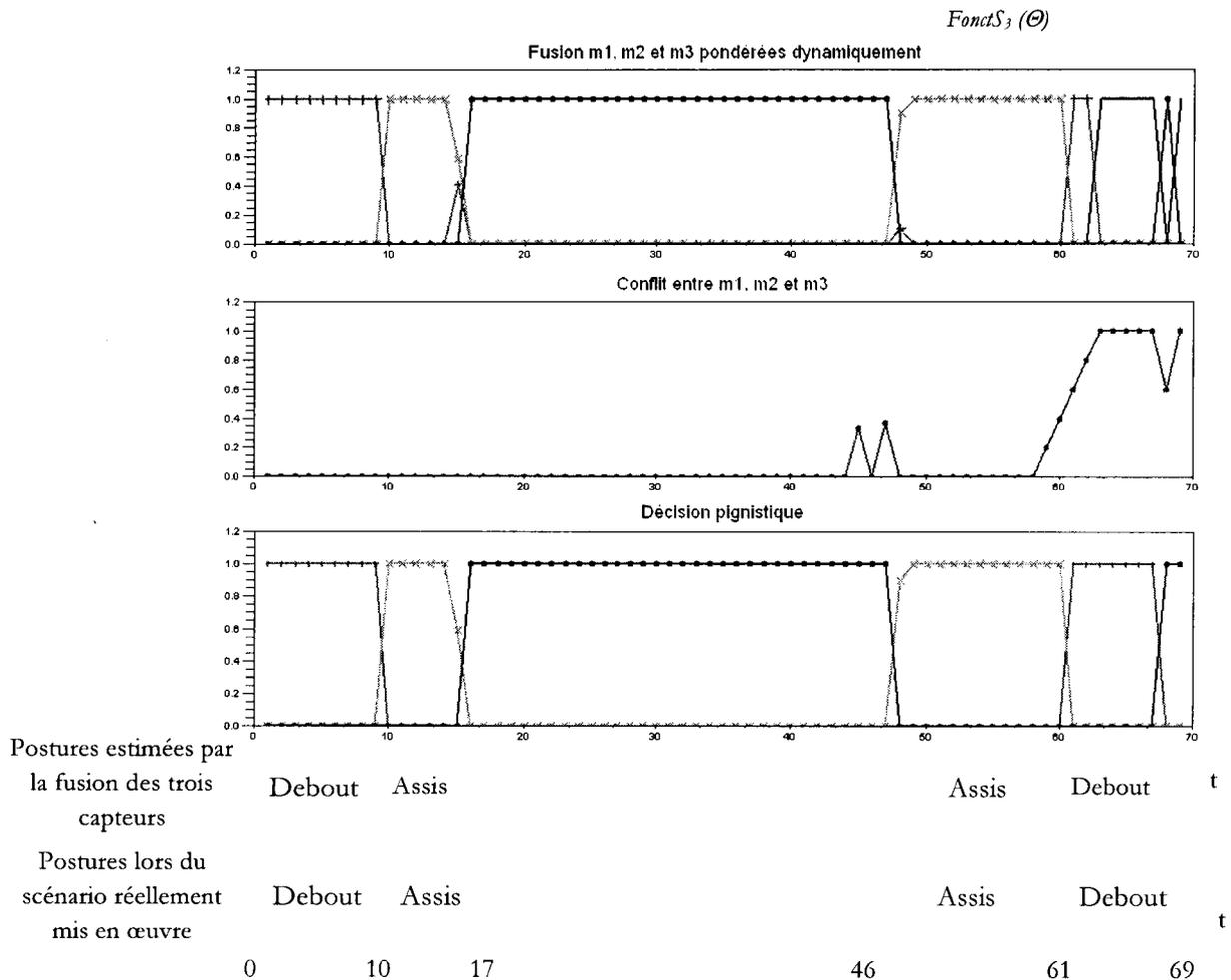


Figure 87 : Fusion des trois jeux de masses m_1 , m_2 et m_3 prenant en compte la fiabilité des capteurs ainsi que la prise de décision pignistique correspondante.

Nous observons que les postures sont correctement estimées et correspondent au scénario prévu à partir des données expérimentales de la Figure 69. A noter l'apparition d'une fausse détection pour $t \in [68,69]$.

Nous observons sur la Figure 87 la fusion des trois capteurs ainsi que la prise de décision pignistique. Une partie importante du conflit (environ 57%) a été supprimée grâce à la pondération du capteur défaillant. Néanmoins, il subsiste du conflit ne permettant pas d'avoir une décision fiable, basée sur les mesures précédentes. Cette non-détection est due au processus de combinaison deux à deux des capteurs. Dans certains cas, le doute entre deux hypothèses peut absorber le conflit entre deux capteurs. Par exemple, à la mesure 65, le premier capteur se prononce fortement sur la disjonction d'hypothèse $Debout \cup Allongé$, c'est-à-dire $m_1(Debout \cup Allongé) = 1$. Le second capteur se prononce sur disjonction d'hypothèse $Debout \cup Assis$, c'est-à-dire $m_2(Debout \cup Assis) = 1$. Le troisième capteur défaillant se prononce sur l'hypothèse Allongé,

c'est-à-dire $m_3(Allongé) = 1$. Ainsi, la fusion entre m_1 et m_3 fournit un conflit nul, la fusion entre m_2 et m_3 fournit un conflit égal à 1. Le capteur 3 se trouve donc bien en conflit avec le capteur 2 mais pas avec le capteur 1, puisque celui-ci se prononce sur une disjonction d'hypothèse, absorbant alors le conflit entre ces deux capteurs. Malgré le fait que le capteur 3 soit défaillant, il est impossible de recouper cette défaillance avec les deux autres capteurs. Il est donc considéré comme fonctionnant correctement.

Cette incapacité à agir face à un tel cas peut être problématique. Aussi, nous avons décidé de considérer cette information de conflit partiel entre capteur. La stratégie adoptée face à cette situation est la suivante : si un capteur est en conflit avec un autre mais pas avec l'intégralité des capteurs, nous considérons les deux capteurs en conflit comme partiellement défaillants. Le second scénario que nous présentons ici est un cas d'application dans lequel ce conflit partiel entre capteur est prépondérant. Il s'appuie également sur les trois capteurs utilisés dans le scénario précédent. Le scénario adopté par l'individu pendant le relevé de mesures est le suivant :

- L'individu est debout pour $t \in [1,10]$,
- L'individu s'assoit pour $t \in [11,17]$,
- L'individu s'allonge pour $t \in [18,46]$,
- L'individu s'assoit pour $t \in [47,61]$,
- L'individu se met debout pour $t \in [62,84]$,
- L'individu s'allonge pour $t \in [85,115]$,
- L'individu s'assoit pour $t \in [116,131]$,
- L'individu se met debout à nouveau pour $t \in [132,138]$.

Au final, nous obtenons un ensemble de 138 mesures pour chacun des capteurs. Les mesures obtenues sont présentées sur la Figure 88.

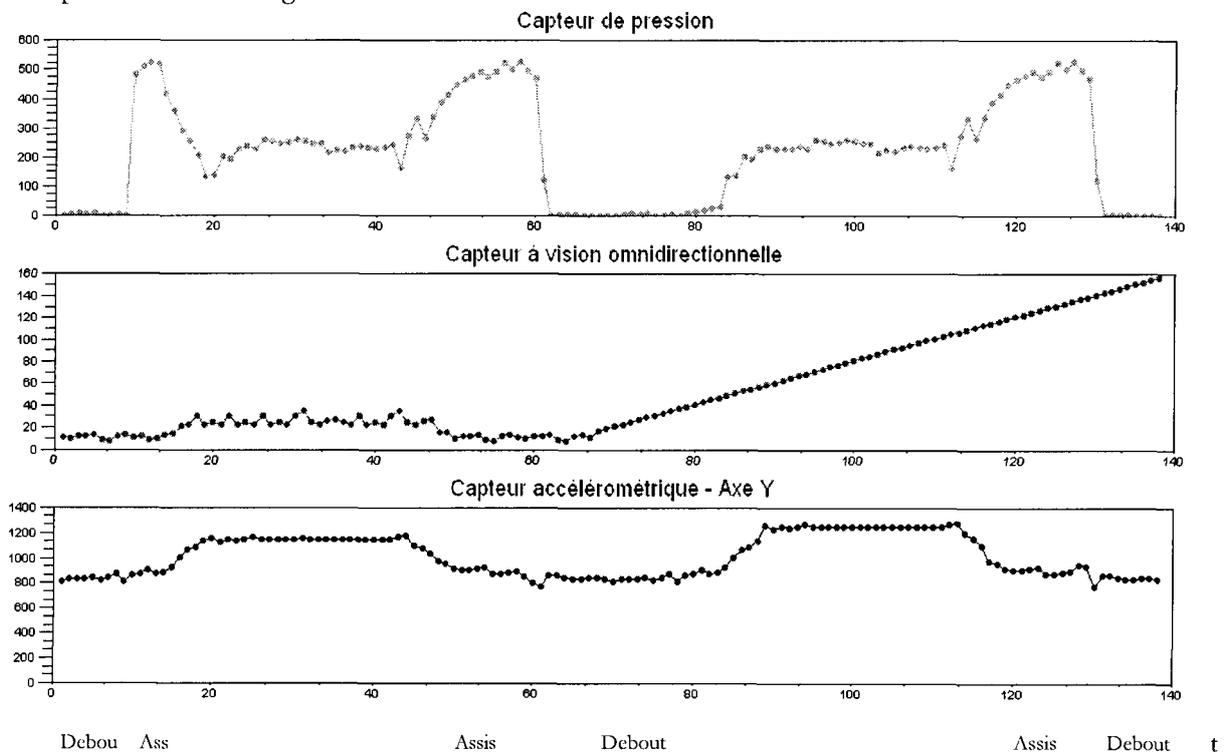
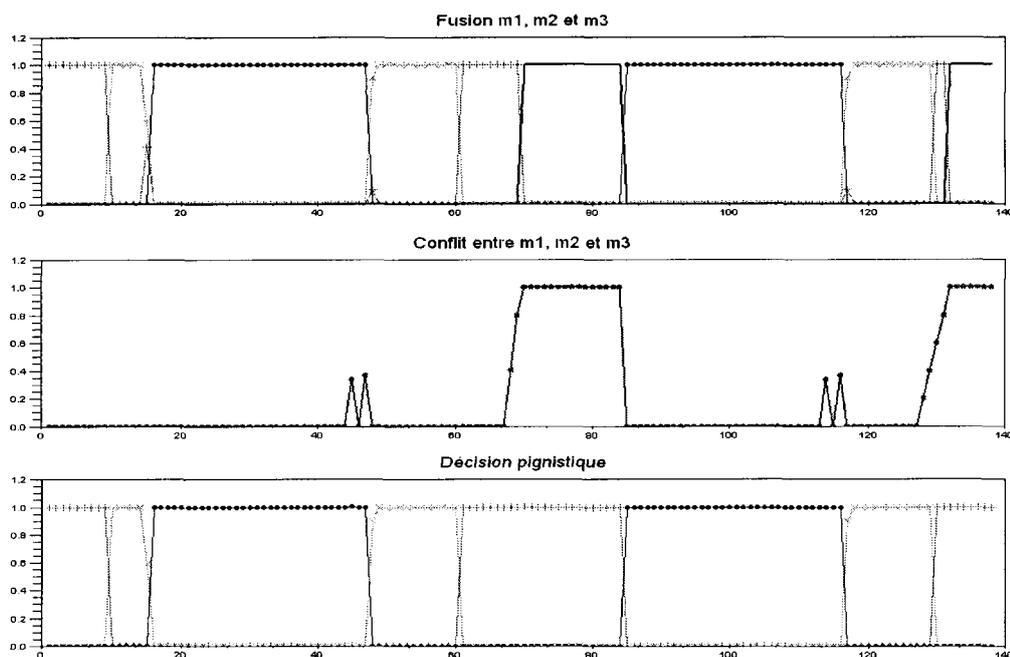


Figure 88 : Signaux expérimentaux obtenus à partir des 3 capteurs pour le second scénario.



Postures estimées
par la fusion des
trois capteurs

Debo Assis Debout Assis Debo t

Postures lors du
scénario réellement
mis en œuvre

Debo Assis Debout Assis Deb t

Figure 90 : Fusion des trois jeux de masses m_1 , m_2 et m_3 prenant en compte la fiabilité des capteurs ainsi que la prise de décision pignistique correspondante pour le second scénario.

Afin de permettre une gestion plus fine de l'information de conflit, nous proposons d'analyser plus finement le conflit entre les capteurs. En nous inspirant de l'opérateur *disjonctif* de Dubois et Prade, nous affaiblissons les deux capteurs qui sont en conflits et dont la défaillance est impossible à isoler. Ainsi, lorsque deux capteurs sont en conflit, mais que le troisième capteur ne peut permettre d'isoler le capteur fautif, les deux capteurs sont pondérés. Cette pondération indique qu'il y a un problème entre ces deux capteurs, soit une mauvaise modélisation des jeux de masse de l'un des capteurs, soit un des deux capteurs est défaillant. La pondération est effectuée à la hauteur de $1/3-2/3$, c'est-à-dire que pour chacun des capteurs, la probabilité pignistique devient :

$$FonctS_i (NON) = 1/3$$

$$FonctS_i (OUI) = 2/3$$

Cette manipulation permet de pondérer légèrement le jeu de masse des deux capteurs dont la défaillance est impossible à isoler. La Figure 91 présente l'application de cette méthode. Alors qu'un conflit existe entre le capteur 2 et le capteur 3, la fusion avec le capteur 1 de ces deux capteurs ne permet pas de définir lequel est défaillant. Lorsque ce cas apparaît, la probabilité pignistique sur l'état de fonctionnement de ces deux capteurs est pondérée. Ainsi, les coefficients affaiblissement en sont modifiés. Nous observons alors une baisse de ce coefficient d'affaiblissement sans toutefois atteindre 0, ce qui signifierait que ces deux capteurs sont défaillants. Néanmoins, leur influence lors de la fusion finale est légèrement pondérée.

conflit entre capteurs. Cette méthode s'inspire de l'opérateur de Dubois et Prade en affaiblissant les deux capteurs en conflit, sans pouvoir déterminer le capteur à l'origine de ce conflit. Grâce à cette méthode, les postures sont correctement estimées avec une baisse probante du conflit lors de la fusion des trois capteurs. Ainsi, la prise de décision pignistique est plus fiable car elle se base sur un conflit beaucoup moins important (la baisse est estimée à 80% des conflits significatifs).

A partir de cette analyse plus fine du conflit, il est alors possible de dresser un état de fonctionnement complet de l'état des capteurs. La figure 93 présente l'état de fonctionnement de chacun des capteurs. L'état vert représente un fonctionnement correct du capteur (ce qui est le cas du premier capteur par exemple). L'état bleu représente un problème indéfini (soit une mauvaise modélisation du jeu de masse du capteur évalué, soit ce capteur est défaillant). L'état rouge représente une défaillance du capteur. Nous remarquons que l'état bleu est indiqué lorsque les capteurs 2 et 3 sont en conflits mais dont la défaillance est impossible à isoler. Par contre, l'état rouge est parfaitement indiqué lorsque le capteur est défaillant.

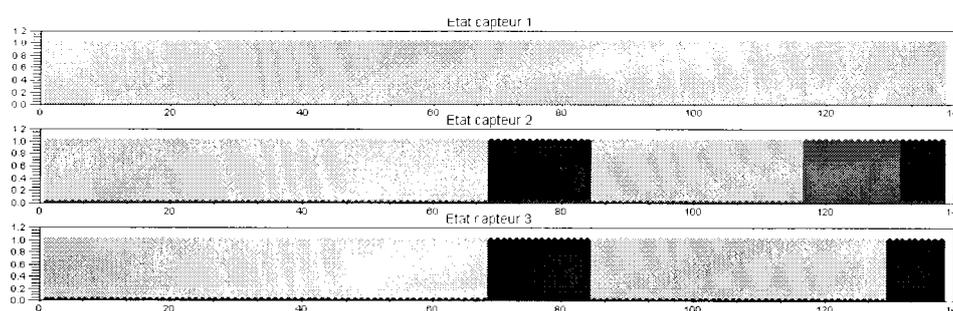


Figure 93 : État de fonctionnement des trois capteurs pour le second scénario.

Cette analyse plus fine du conflit entre capteurs offre un outil de détection de défaillance capteur ainsi qu'un outil d'information. Il permet d'informer qu'un problème est apparu sans pouvoir précisément déterminer la cause du problème (soit une mauvaise modélisation du jeu de masse du capteur évalué, soit ce capteur est défaillant). Néanmoins, il permet d'avertir l'expert humain qu'une maintenance est à effectuer.

4 Détection de dérives comportementales – Etape E3

Dans la section précédente, nous avons validé notre approche de détection et d'identification de capteurs défaillants au sein d'une architecture de fusion de données. Dans cette section, nous mettons en œuvre notre méthode de détection d'anomalies comportementales dans le cadre de la surveillance d'un individu. Ici, nous souhaitons mettre en évidence l'impact que peut avoir un capteur défaillant lors de la surveillance d'un individu. L'objectif est de montrer qu'un capteur défaillant dans la chaîne de fusion de données peut engendrer une omission d'information sur le comportement de l'individu et entraîner une mauvaise surveillance de l'individu [RIC08c].

Dans cette section, nous mettons en œuvre dans un premier temps la méthode de détection d'anomalies comportementales dans le cadre de la théorie du Modèle des Croyances Transférables vue au chapitre 3. La méthode mise en œuvre s'appuie sur la comparaison entre un état observé et un état prédit. L'état prédit se base sur le modèle de comportement du capteur défini au préalable. Deux jeux de masses sont obtenus et fusionnés. Le conflit entre les jeux de masses observés et prédits est analysé grâce à trois experts.

Nous comparons ici les deux approches présentées pour cette méthode. La première se basant sur un modèle comportemental probabiliste à l'aide d'une Chaîne de Markov. La seconde s'appuyant sur une approche à base de fonction de Croyance.

Nous mettons en œuvre cette méthode sur les données issues des trois capteurs présentés précédemment : capteur de pression, capteur à vision omnidirectionnelle et capteur accélérométrique. L'objectif est de détecter des anomalies comportementales par observation de l'enchaînement des postures humaines.

Nous voyons dans un second temps l'impact d'un capteur défaillant sur la détection de dérives comportementales. A partir de la méthode de détection et d'identification d'un capteur défaillant, nous mettons en œuvre l'affaiblissement du capteur défaillant et observons l'impact de cette pondération sur la détection de dérives comportementales.

4.1 Modélisation du comportement humain par Chaîne de Markov et prédiction markovienne

Afin de pouvoir élaborer une prédiction de la posture d'un individu, il est nécessaire d'utiliser un modèle connu, défini au préalable. Ce modèle, nous le définissons par expertise humaine, le travail d'apprentissage n'étant pas celui qui nous incombe ici. Le comportement humain, et plus particulièrement le modèle comportemental de l'enchaînement des postures d'un humain, peut se décrire sous la forme d'une Chaîne de Markov. En effet, les postures sont des états s'enchaînant temporellement et respectant la propriété Markovienne : la distribution conditionnelle de probabilité des postures futures, étant donné l'instant présent, ne dépend que de cette même posture présente et pas des postures passées.

Une telle propriété permet ainsi de modéliser les activités humaines en les représentant sous forme de graphe. Dans le cadre de l'analyse du comportement humain, une Chaîne de Markov permet de décrire l'enchaînement des postures d'un individu. Nous définissons une Chaîne de Markov répondant aux caractéristiques suivantes :

- Quand l'individu est debout, à l'instant suivant l'individu peut rester debout ou être assis,
- Quand l'individu est assis, à l'instant suivant l'individu peut rester assis, être allongé ou être debout,
- Quand l'individu est allongé, à l'instant suivant l'individu peut rester allongé ou être assis.

La Chaîne de Markov correspondant à ces caractéristiques est la suivante :

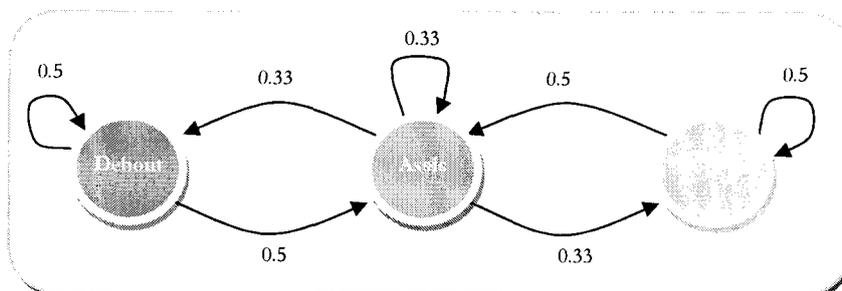


Figure 94 : Chaîne de Markov modélisant un comportement humain.

La matrice de transition correspondante est :

$$T =$$

	<i>Debout</i>	<i>Assis</i>	<i>Allongé</i>
<i>Debout</i>	0.5	0.5	0
<i>Assis</i>	0.33	0.33	0.33
<i>Allongé</i>	0	0.5	0.5

A partir de cette matrice de transition, nous pouvons estimer la posture d'un individu à l'instant suivant selon :

$$V_{Prediction} = V_{Observation} * T$$

Où $V_{Observation}$ est un vecteur d'observation et $V_{Prediction}$ le vecteur estimé. Cette notion représente la prédiction markovienne.

Le vecteur d'observation $V_{Observation}$ est obtenu à partir de la fusion des trois capteurs vus précédemment. La fusion de ces trois capteurs fournit un jeu de masse $m_{Observation}$, nécessitant alors une transformation en jeu de probabilité afin de déterminer le vecteur estimé $V_{Prediction}$. La transformation pignistique est utilisée afin de transformer le jeu de masse $m_{Observation}$ en un jeu de probabilité.

4.2 Application de la prédiction Markovienne à la détection de posture

Nous avons appliqué la prédiction Markovienne sur les mesures de la Figure 69. Le vecteur de probabilités $V_{Prediction}$ obtenu à partir de la prédiction markovienne est ensuite converti en jeu de masse $m_{Prediction}$. Ainsi, les probabilités sur les hypothèses singletons sont placées sur les masses concernant les hypothèses singletons correspondantes :

$$m_{Prediction}(Hi) = P_{Prediction} (Hi)$$

Ainsi, les masses sur les hypothèses non singletons sont nulles.

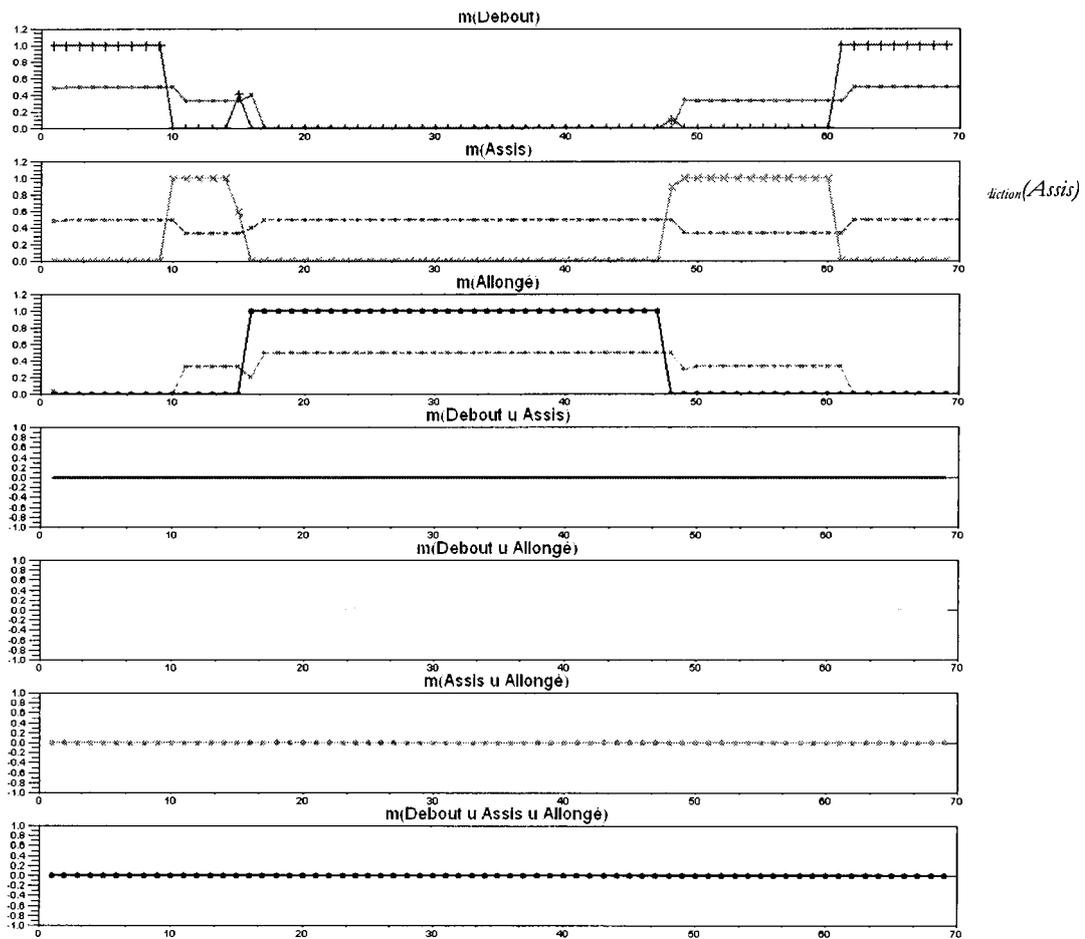


Figure 95 : Application de la prédiction Markovienne aux mesures expérimentales de la Figure 69.

La caractéristique principale de la prédiction Markovienne est qu'elle fournit une tendance sur la posture de l'individu à l'instant suivant. Elle est fortement imprécise dans le sens où les distributions de probabilités sont, dans notre cas, équiprobables. Cette équiprobabilité se traduit par au moins deux possibilités lorsque l'individu se trouve dans une posture particulière. Par exemple, lorsque l'individu se trouve dans la posture « debout », il peut soit y rester, soit passer à la posture « assise ». Cette imprécision ne permet donc pas de déterminer précisément la posture à l'instant suivant mais de simplement proposer une tendance vers une posture.

La Figure 95 présente l'application de la prédiction Markovienne (mesures marron) sur le jeu de masse $m_{\text{Observation}}$ issue de la fusion des signaux de la Figure 69. Nous constatons que la prédiction se prononce uniquement sur les hypothèses singletons. Afin de montrer la nature imprécise de la prédiction Markovienne, nous proposons d'estimer la similarité entre observation et prédiction. Cette estimation de similarité est effectuée à partir d'une mesure de distance entre les deux jeux de masse.

Wartelle [WAR06] propose une mesure de distance basée sur les mesures pignistiques des jeux de masses utilisés appartenant au même cadre de discernement. Appliquée à nos jeux de masse d'observation et de prédiction, nous obtenons la formule suivante :

$$d_{\text{Wartelle}}(m, n) = \frac{1}{2} \sum_{A \in \mathcal{A}} \max(0, |m(A) - n(A)|)$$

Cette distance est comprise dans l'intervalle $[0, 1]$. La valeur minimale de la distance ($d_{\text{OBSER}_\text{Pred}} = 0$) est obtenue entre deux jeux de masse identiques. La distance maximale est obtenue entre deux jeux de masse en conflit maximal.

Par exemple :

Deux jeux de masse identiques :

$$\begin{array}{ll} P^{\text{Pig}}_{\text{Observation}} (\text{OUI}) = 0.9 & P^{\text{Pig}}_{\text{Observation}} (\text{NON}) = 0.1 \\ P^{\text{Pig}}_{\text{Prédiction}} (\text{OUI}) = 0.9 & P^{\text{Pig}}_{\text{Prédiction}} (\text{NON}) = 0.1 \end{array}$$

- $d_{\text{OBSER}_\text{Pred}} = 0$

Deux jeux de masse différents :

$$\begin{array}{ll} P^{\text{Pig}}_{\text{Observation}} (\text{OUI}) = 0.1 & P^{\text{Pig}}_{\text{Observation}} (\text{NON}) = 0.9 \\ P^{\text{Pig}}_{\text{Prédiction}} (\text{OUI}) = 0.9 & P^{\text{Pig}}_{\text{Prédiction}} (\text{NON}) = 0.1 \end{array}$$

- $d_{\text{OBSER}_\text{Pred}} = 0,8$

Deux jeux de masse en conflit maximal :

$$\begin{array}{ll} P^{\text{Pig}}_{\text{Observation}} (\text{OUI}) = 1 & P^{\text{Pig}}_{\text{Observation}} (\text{NON}) = 0 \\ P^{\text{Pig}}_{\text{Prédiction}} (\text{OUI}) = 0 & P^{\text{Pig}}_{\text{Prédiction}} (\text{NON}) = 1 \end{array}$$

- $d_{\text{OBSER}_\text{Pred}} = 1$

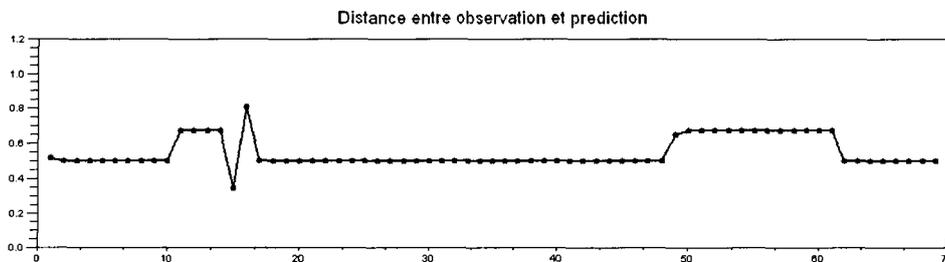


Figure 96 : Calcul de la distance entre observation et prédiction.

La distance entre les jeux de masse observés et prédits permet d'observer l'imprécision de la prédiction Markovienne. La Figure 96 présente la distance entre les jeux de masse observé et prédit à partir des données de la Figure 95. En moyenne, la distance entre ces deux jeux de masse est de 0.5. Cette différence reflète l'imprécision de la prédiction Markovienne. Néanmoins, pour la suite de nos travaux, cette prédiction Markovienne est suffisante car elle permet de fournir une tendance quant à la posture à l'instant suivant. Nous rappelons que l'objectif de cette section est de vérifier que l'observation et la prédiction sont en accords.

Afin de vérifier l'accord entre l'observation et la prédiction, nous analysons le conflit résultant de la fusion entre $m_{\text{Observation}}$ et $m_{\text{Prédiction}}$. L'approche mise en œuvre pour détecter une anomalie de comportement est l'analyse du conflit entre observation et prédiction. L'application de trois experts se prononçant sur l'analyse temporelle de ce conflit permet de déterminer si une anomalie de comportement est apparue. Pour valider cette approche, nous mettons en œuvre cette méthode sur les signaux de la Figure 97. Le

scénario mis en œuvre ici s'appuie sur les trois capteurs utilisés pour caractériser la posture d'un individu. Le comportement adopté par l'individu pendant le relevé de mesures est le suivant :

- L'individu est debout pour $t \in [1,10]$,
- L'individu s'assoit pour $t \in [11,17]$,
- L'individu s'allonge pour $t \in [18,46]$,
- L'individu s'assoit pour $t \in [47,61]$,
- L'individu se met debout pour $t \in [62,84]$,
- L'individu s'allonge pour $t \in [85,115]$,
- L'individu s'assoit pour $t \in [116,131]$,
- L'individu se met debout à nouveau pour $t \in [132,138]$.

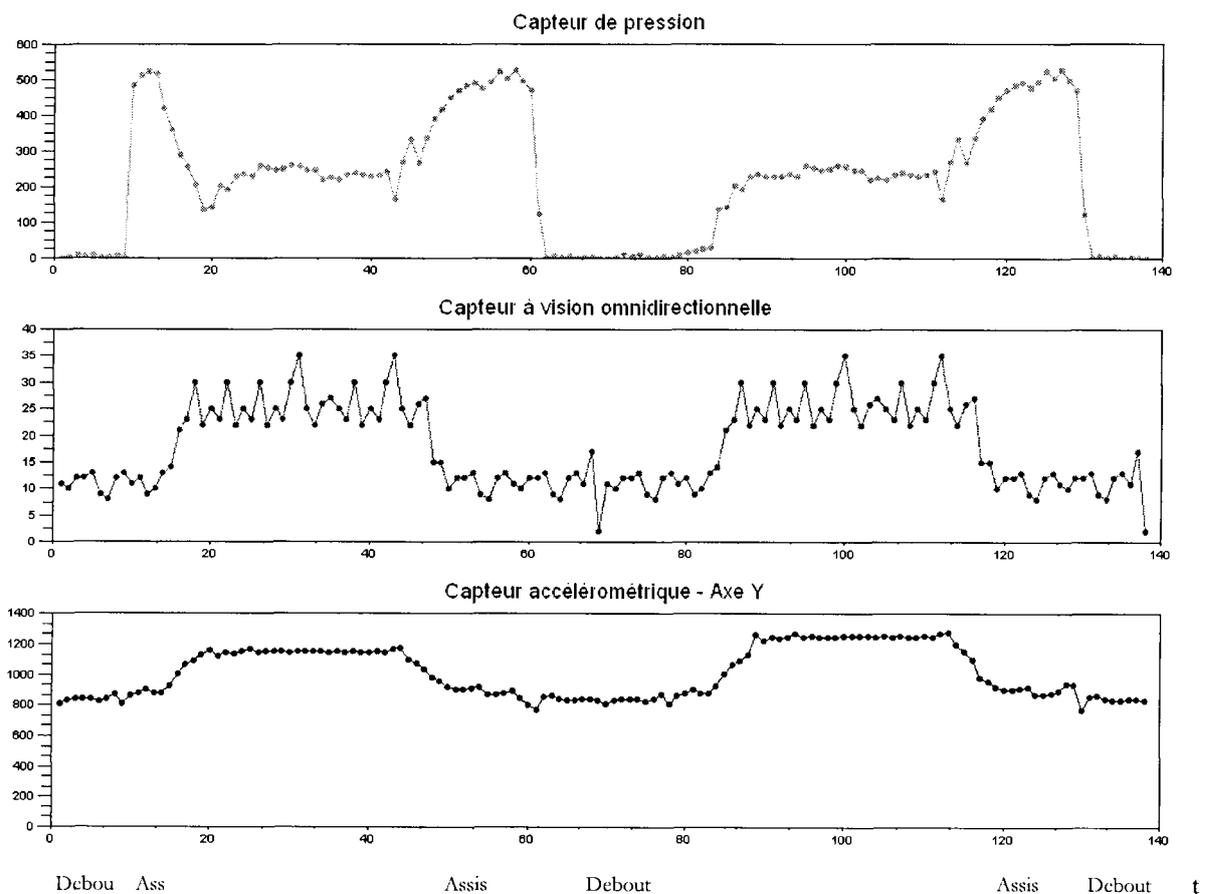


Figure 97 : Signaux expérimentaux obtenus à partir des 3 capteurs possédant une anomalie de comportement.

Sur ces signaux, une anomalie apparaît entre les mesures 84 et 85, où l'individu passe de la posture debout à allongée, ce qui est théoriquement impossible selon la matrice de transition T choisie ($P_{Debout/Allongé} = 0$). La fusion des mesures issues de ces trois capteurs fournissent les résultats attendus (voir Figure 98).

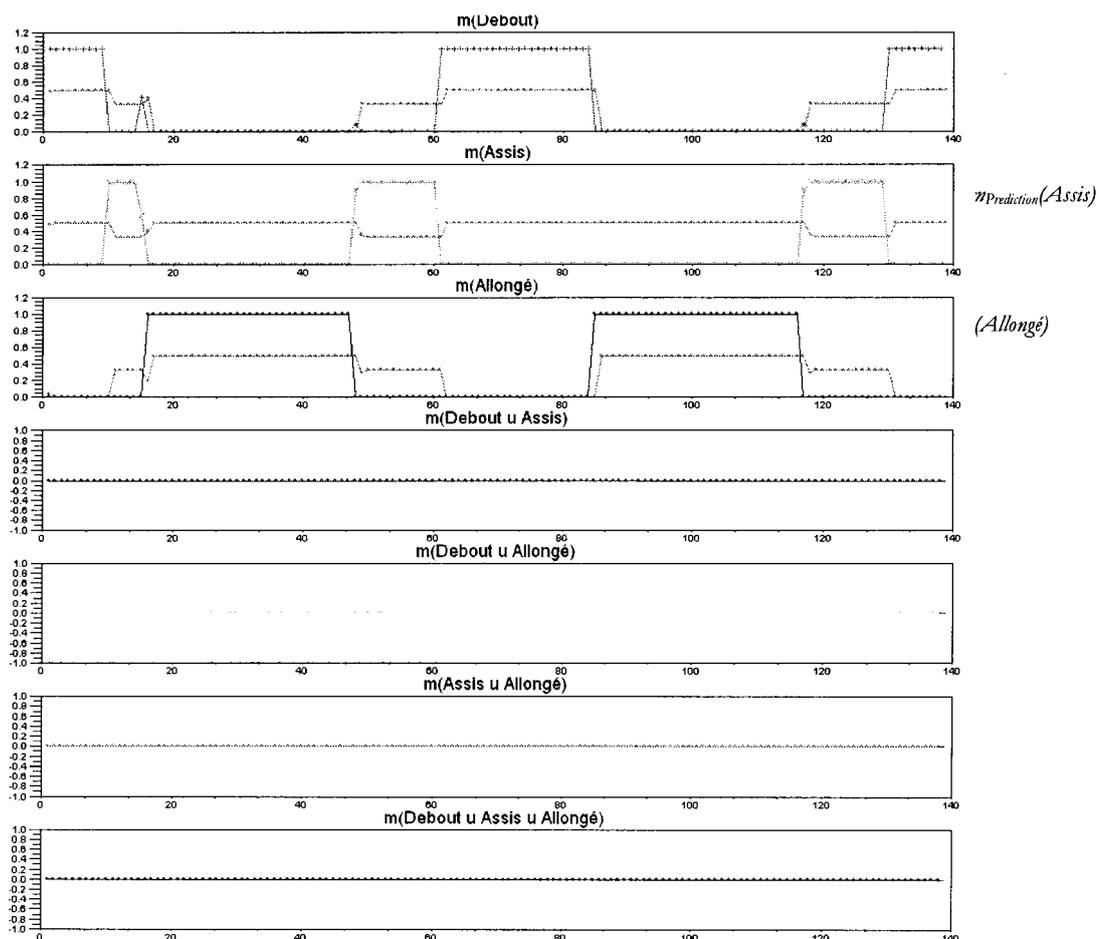


Figure 98 : Application de la prédiction Markovienne aux mesures expérimentales de la Figure 97 sur lesquelles apparaissent une anomalie de comportement (à $t=85$).

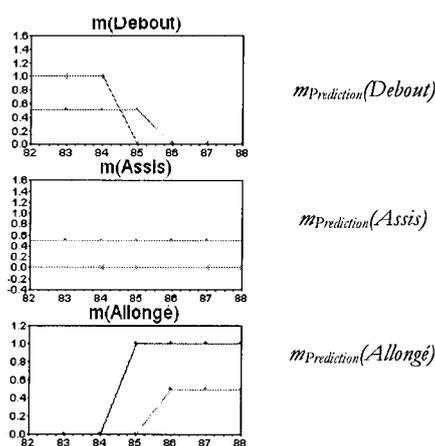


Figure 99 : Zoom sur la période d'anomalie de comportement à $t=85$.

Une dérive de comportement est perceptible grâce à la prédiction Markovienne. En effet, lorsque la prédiction et l'observation diffèrent, la mesure de similarité entre ces deux jeux de masses est importante. C'est le cas à $t=85$. Nous observons que la prédiction indique que la posture sera soit debout ($m(Debout) = 0.5$) ou assise ($m(Assis) = 0.5$) à l'instant suivant. En aucun cas, la posture ne pourra être allongée ($m(Allongé) = 0$). Dans ce cas, nous obtenons une mesure de similarité, basée sur la distance entre $m_{Observation}$ et $m_{Prediction}$ maximale : $d_{OBSER_Pred} = 1$

Cette distance maximale indique une dissimilarité maximale entre ces deux jeux de masse à cet instant précis. Celle-ci est due à une prédiction contraire à l'observation. La valeur de conflit entre observation et prédiction à cet instant met en avant cette dérive. Le conflit $m_{fusion}(e)$ est issue de la fusion entre $m_{Observation}$ et $m_{Prédiction}$ selon l'opérateur de Smets :

$$m_{Fusion} = m_{Prédiction} \cap m_{Observation}$$

Le conflit $m_{fusion}(r)$ est alors isolé et analysé dans l'objectif de détecter une dérive de comportement caractérisé par un saut soudain du conflit. A l'instant précis où la dérive de comportement intervient, le conflit est maximal : $m_{fusion}(r) = 1$.

4.3 Analyse du conflit entre observation et prédiction et identification d'une dérive comportementale

Lorsque le conflit est calculé entre l'observation et la prédiction, il est possible de l'analyser afin de détecter une éventuelle dérive de comportement. Ici, nous présentons l'analyse du conflit issu de la fusion entre les jeux de masse observé et prédit.

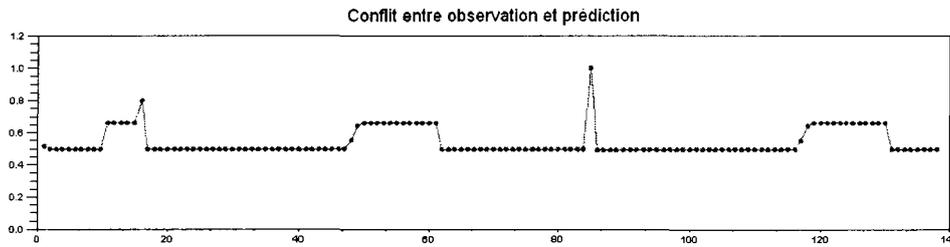


Figure 100 : Conflit $m_{fusion}(e)$ entre observation et prédiction.

Sur la Figure 100, à $t=85$, le conflit subit une hausse soudaine. Cette hausse est due à un désaccord entre observation et prédiction, comme nous l'avons constaté sur la Figure 99. Nous observons que cette anomalie de comportement est caractérisée par une hausse importante et de courte durée du conflit. A partir de trois experts, une analyse de ce signal est alors possible. Le cadre de discernement de ces trois experts est $\Theta_{Comportement} = \{OUI, NON\}$ avec l'hypothèse 'OUI' signifiant 'le comportement du capteur est normal' et l'hypothèse 'NON' signifiant 'le comportement du capteur n'est pas normal'.

Le premier expert est en charge de détecter les hausses de conflit. Il porte sur la dérivée du signal de conflit $m_{fusion}(r)$.

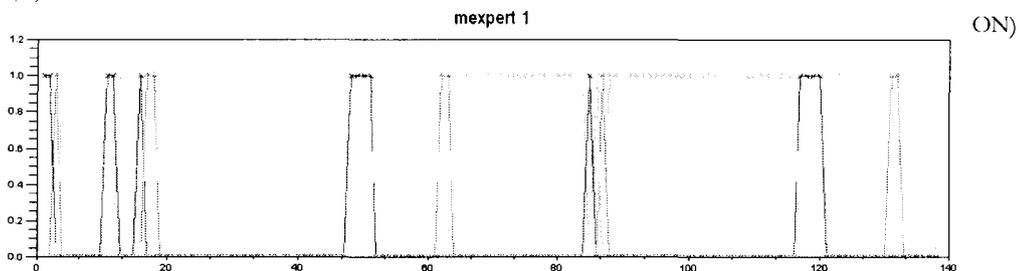


Figure 101 : Application du premier expert sur le conflit $m_{fusion}(e)$ entre observation et prédiction.

La masse sur l'hypothèse 'OUI' apparaît en vert, la masse sur l'hypothèse 'NON' en rouge et la masse sur la disjonction d'hypothèses 'OUI \cup NON' apparaît en jaune. Ce premier expert met en avant les montées de conflit selon l'hypothèse 'NON'. Pour les descentes de conflit, l'opinion de cet expert porte sur

l'hypothèse 'OUP'. Les périodes de conflit stable apparaissent quant à elle selon la disjonction d'hypothèse 'OUI \cup NON'. Le réglage choisi pour la fonction de masse est $Offset1 = 0.001^{20}$.

Cet expert fourni le jeu de masse $m_{Expert1}$.

Le deuxième expert porte sur l'amplitude des pics afin de détecter les sauts soudains de conflits.

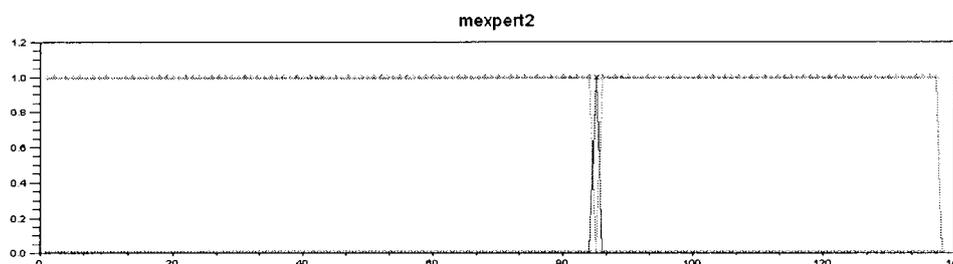


Figure 102 : Application du second expert sur le conflit $m_{fusion}(-)$ entre observation et prédiction.

Cet expert permet d'isoler les forts conflits en attribuant une masse importante sur l'hypothèse 'NON' en cas de fort conflit. Lorsque le conflit est faible, cet expert attribue une masse importante sur l'hypothèse 'OUP'. Les réglages choisis pour la fonction de masse sont $seuil2 = 0.9$ et $Offset2 = 0.5$.

Cet expert fourni le jeu de masse $m_{Expert2}$.

Le troisième expert porte sur la durée des pics de conflit afin de détecter les pics de courte durée.

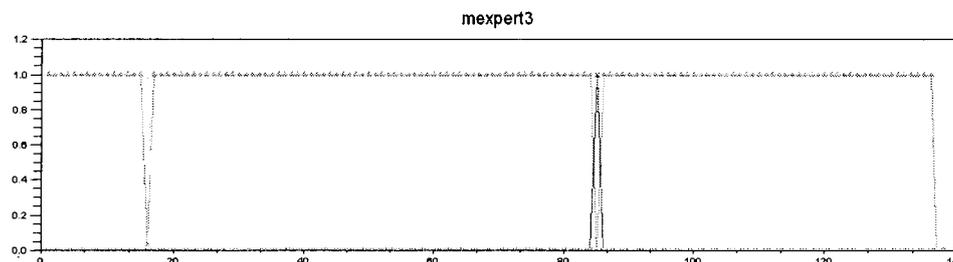


Figure 103 : Application du troisième expert sur le conflit $m_{fusion}(-)$ entre observation et prédiction.

Cet expert permet d'isoler les conflits de courte durée en attribuant une masse importante sur l'hypothèse 'NON' en cas de court conflit. Lorsque le conflit est long, cet expert attribue une masse importante sur l'hypothèse 'OUP'. Les réglages choisis pour la fonction de masse sont $seuil3 = 0.3$, $Offset3 = 0.02$ et un pas de 2.

Cet expert fourni le jeu de masse $m_{Expert3}$.

²⁰ Se reporter au chapitre 3, §4.6 pour la fonction de masse correspondante

A partir de la fusion de ces trois experts, nous parvenons à isoler les pics de conflit correspondant à des sauts d'états impossibles. La fusion est réalisée avec l'opérateur de Smets :

$$m_{Fonct} = m_{Expert1} \cap m_{Expert2} \cap m_{Expert3}$$

La détection d'un problème de comportement se base sur la transformation pignistique de m_{Fonct} . Un problème de comportement est décidé lorsque $P_{Fonct}(NON) > P_{Fonct}(OUI)$.

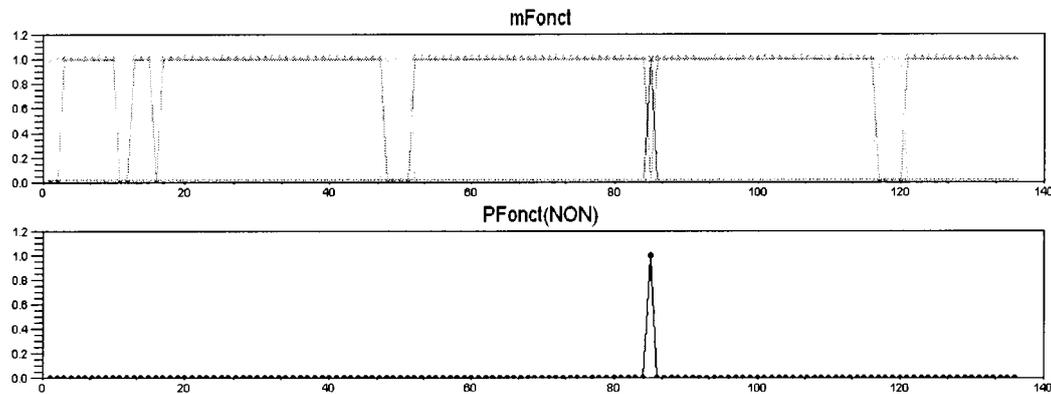


Figure 104 : Fusion des trois experts et prise de décision sur un problème de comportement.

La fusion de ces trois experts et la prise de décision pignistique permettent d'apercevoir une anomalie de comportement à $t=85$. En effet, à $t=85$, un pic de conflit apparaît, dû au passage impossible de la posture debout à la posture allongée. Nous voyons que les trois experts utilisés permettent de caractériser efficacement des conflits importants de courtes durées. A partir de leur fusion, il est possible d'extraire des anomalies de comportement.

Le conflit $m_{fusion}(r)$ que nous traitons ici est ainsi basé sur une prédiction Markovienne classique. Nous avons vu que celle-ci est critiquable puisqu'elle est fortement imprécise. Cette imprécision s'en ressent sur le conflit issu de la fusion entre les jeux de masse $m_{Prédiction}$ et $m_{Observation}$: un conflit moyen d'environ 0.5 est omniprésent même lorsque le comportement de l'individu est correct. L'analyse de ce conflit en est donc complexifiée. Il est nécessaire dans ce cas d'analyser finement le conflit dans l'objectif de détecter un saut soudain, caractéristique d'une dérive de comportement. Certains cas nous ont montré la faiblesse de cette approche lorsque l'observation sur la posture de l'individu est elle-même incertaine. Le conflit est alors beaucoup moins caractéristique et rend l'analyse plus complexe en termes de réglage d'experts.

Afin de rendre plus précise l'analyse du conflit, il est nécessaire de mettre en œuvre un modèle de prédiction prenant en compte la nature incertaine de l'observation et adaptée dans le cadre de l'utilisation de fonctions de Croyance. Nous proposons ainsi d'utiliser dans ce cas un modèle basé sur une Chaîne de Markov au sens crédibiliste permettant de prendre en compte la nature incertaine des états évalués, rendant la prédiction plus précise.

4.4 Amélioration de la prédiction : une approche crédibiliste

Afin de rendre plus précise la prédiction Markovienne, nous mettons en œuvre ici une Chaîne de Markov au sens crédibiliste. Cette Chaîne de Markov « crédibiliste » est modélisée par une matrice de transition T_{Cred} ayant la particularité de modéliser les disjonctions d'états, c'est-à-dire les états intermédiaires de passage d'un état à un autre. Ainsi, si nous redistribuons les probabilités de transition de la matrice de transition T précédente sur les hypothèses et sur les disjonctions d'hypothèses, nous obtenons la nouvelle matrice de transition crédibiliste T_{Cred} suivante :

$$T_{\text{Cred}} =$$

	<i>Debout</i>	<i>Assis</i>	<i>Allongé</i>	<i>Debout</i> \cup <i>Assis</i>	<i>Debout</i> \cup <i>Allongé</i>	<i>Assis</i> \cup <i>Allongé</i>	Θ_{Posture}
<i>Debout</i>	1/3	1/3	0	1/3	0	0	0
<i>Assis</i>	0	0	0	0	0	0	1
<i>Allongé</i>	0	1/3	1/3	0	0	1/3	0
<i>Debout</i> \cup <i>Assis</i>	1/3	1/3	0	1/3	0	0	0
<i>Debout</i> \cup <i>Allongé</i>	0	0	0	0	0	0	0
<i>Assis</i> \cup <i>Allongé</i>	0	1/3	1/3	0	0	1/3	0
Θ_{Posture}	1/7	1/7	1/7	1/7	1/7	1/7	1/7

Cette nouvelle matrice de transition crédibiliste se base sur la matrice de transition T présentée précédemment. A partir d'une expertise humaine, nous redistribuons les probabilités de transition des hypothèses singletons sur l'ensemble des 2^3 sous ensembles du cadre de discernement Θ_{Posture} . Cette matrice de transition modélise les postures singletons ainsi que les disjonctions de postures. Ainsi, lorsque l'individu est en posture debout, à l'instant suivant il peut rester en posture debout, soit être assis ou en train de s'asseoir. Cette nuance est possible grâce à la disjonction *Debout* \cup *Assis*. Lorsque l'individu est assis, l'ignorance est totale puisque à l'instant suivant il peut se trouver dans l'une des trois postures ou être en cours de changement de posture. Comme pour la posture debout, lorsque l'individu est en posture allongée, il peut rester en posture allongée, soit être assis ou en train de s'asseoir à l'instant suivant. Contrairement à la matrice de transition T , (§4.1) où les postures singletons sont uniquement modélisées, la matrice de transition T_{Cred} modélise le doute entre deux postures : *Debout* \cup *Assis*, *Debout* \cup *Allongé*, *Assis* \cup *Allongé*. A partir de ce doute entre deux postures, l'individu peut se trouver dans l'une des deux postures à l'instant suivant ou rester entre ces deux postures. Quand la posture est ignorée totalement $m(\Theta_{\text{Posture}})=1$, à l'instant suivant, l'individu peut se trouver dans l'une des trois postures ou se trouver dans une phase de doute entre deux postures, d'où l'équi-répartition des probabilités de transition vers les postures singletons et les disjonctions de postures.

Cette matrice de transition facilite le travail de prédiction Markovienne puisque la phase de transformation du jeu de masse $m_{\text{Observation}}$ en jeu de probabilité pignistique n'a plus lieu d'être. Ainsi, cette matrice de transition conserve l'un des atouts de la théorie des fonctions de croyances à savoir la modélisation du doute entre hypothèses.

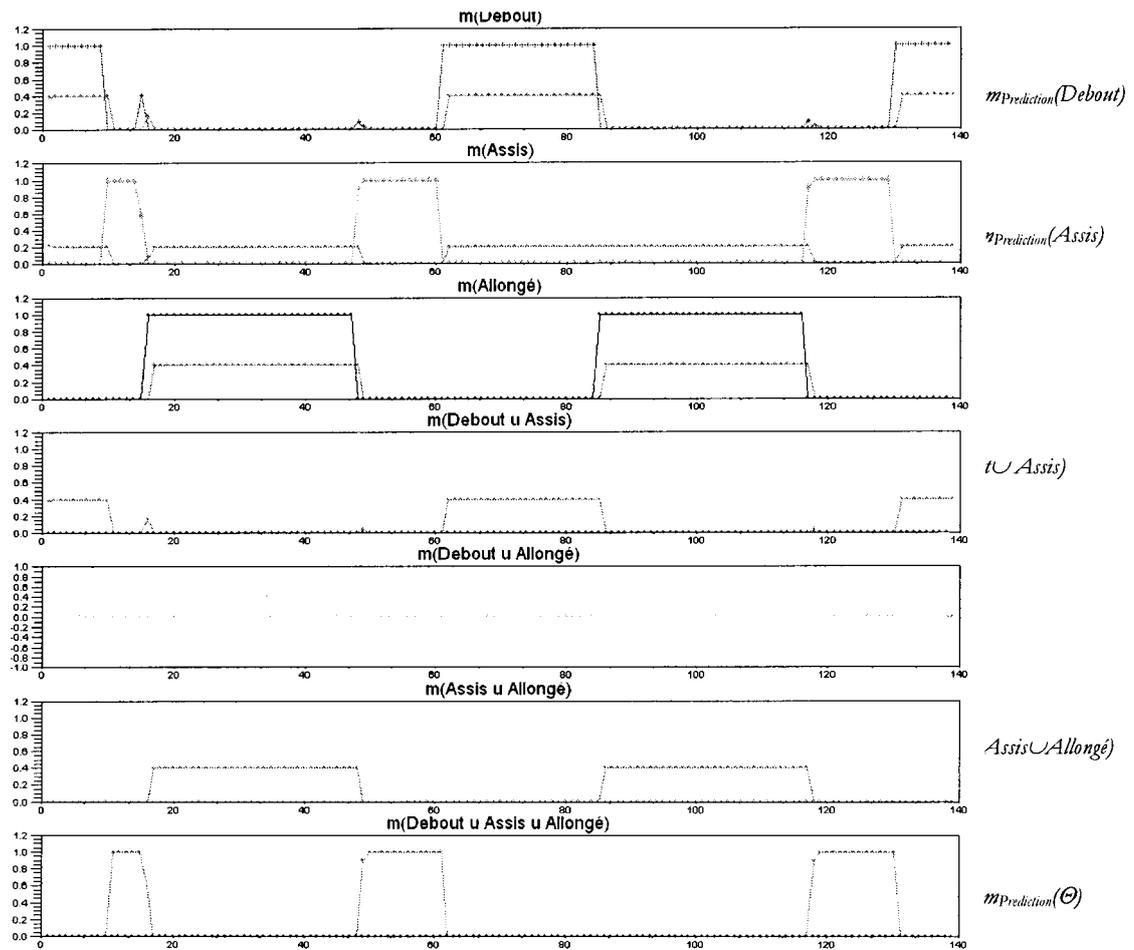


Figure 105 : Application de la prédiction Markovienne basée sur la matrice de transition crédibiliste aux mesures expérimentales de la Figure 97.

La matrice de transition crédibiliste permet de générer une prédiction sur les hypothèses mais aussi sur les disjonctions d'hypothèses du référentiel de définition. La Figure 105 présente l'utilisation de la matrice de transition crédibiliste sur les données expérimentales de la Figure 97. Nous constatons l'apparition de prédictions sur les disjonctions d'hypothèses.

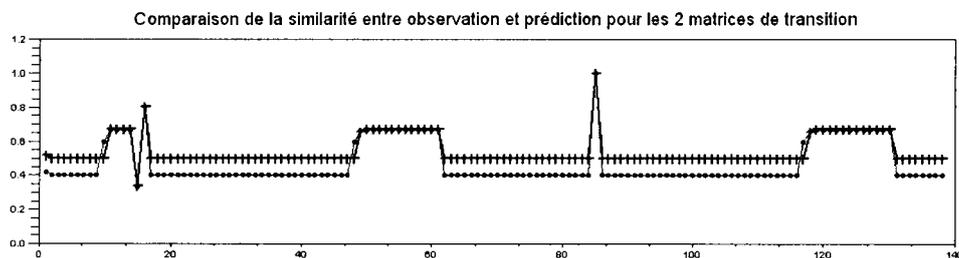


Figure 106 : Comparaison des distances de similarité entre observation et prédiction pour les deux matrices de transition.

A partir des résultats expérimentaux de la Figure 106, nous observons que la matrice de transition crédibiliste fournit une prédiction sensiblement plus précise. En effet, par comparaison avec l'utilisation de la matrice de transition T , la distance entre observation et prédiction est diminuée d'environ 20%. Néanmoins, nous constatons que la variation entre observation et prédiction lors d'une anomalie de

comportement reste importante. Grâce à l'amélioration de la prédiction, le conflit $m_{fusion}(I)$ moyen, résultant de la fusion $m_{fusion} = m_{prédiction} \cap m_{observation}$ est diminué.

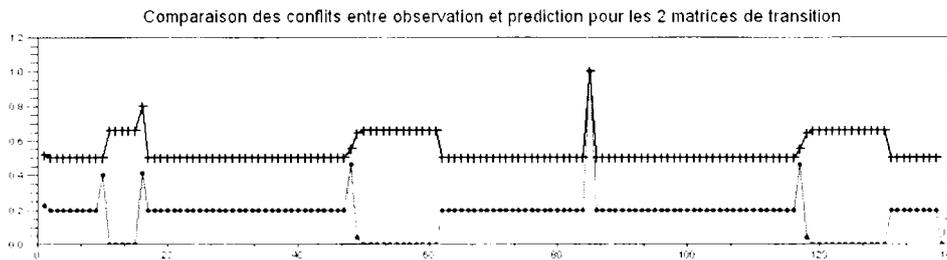


Figure 107 : Comparaison des conflits entre observation et prédiction pour les deux matrices de transition.

Comme pour la distance entre observation et prédiction, le conflit entre observation et prédiction est significativement diminué d'environ 60%. De même, le conflit lors d'une anomalie de comportement reste important. Il est ainsi plus aisé pour un système expert d'isoler une anomalie de comportement.

A partir de ce conflit, les trois experts et leur fusion fournissent sensiblement les mêmes résultats. La seule différence se situe au niveau du réglage des seuils et des écarts. L'utilisation d'une matrice de transition crédibiliste rend le réglage plus simple à réaliser car les pics de conflits apparaissent beaucoup plus clairement. Nous pourrions envisager l'utilisation d'un seul expert se prononçant uniquement sur l'amplitude du conflit.

4.5 Impact d'une défaillance capteur sur la détection de dérives comportementales

Cette section porte sur le couplage des deux approches vues précédemment et présente l'impact d'un capteur défaillant sur la détection de dérives comportementales dans l'analyse de l'activité humaine. Nous montrons ici qu'un capteur défaillant peut engendrer une détection et une caractérisation erronée d'anomalie.

En cas de défaillance capteur, nous avons démontré que la décision pignistique finale peut être faussée, principalement due au fait que le conflit issu de la fusion est reporté sur l'ignorance. Lorsque le conflit est maximal ($k=I$), nous avons vu que l'ignorance est totale. Dans ce cas, nous avons arbitrairement décidé de reporter la posture de l'instant précédent afin de ne prendre aucun risque. Les mesures de la Figure 88 en sont un bon exemple. Le second capteur est défaillant alors qu'une anomalie comportementale intervient : l'individu passe de la posture debout à allongée. Néanmoins, lors de la prise de décision pignistique finale (Figure 89), cette anomalie de comportement est masquée par le report de la posture précédente puisque la décision est impossible à prendre.

Nous avons vu que la détection de dérives comportementales s'appuie sur l'enchaînement des postures et en particulier, le passage anormal d'une posture à une autre. Comme une défaillance capteur induit une perte d'estimation de postures, la détection de dérives comportementales est impossible à réaliser durant cette période de défaillance. Afin de permettre à nouveau cette analyse, il faut procéder à une prise en compte en parallèle de la fiabilité de chacun des capteurs utilisés. Pour cela, la mise en cascade des deux approches précédentes est nécessaire. Elle permet de décomposer l'analyse du conflit en deux parties distinctes mais complémentaires. Lorsque du conflit est révélé lors de la combinaison des capteurs, nous tentons de le diminuer voire de le faire disparaître grâce à l'affaiblissement du jeu de masse du capteur supposé défaillant. Lorsqu'une partie importante du conflit a été retirée, l'ignorance totale n'a plus lieu

d'être et la prise de décision pignistique sur la posture est réalisable. Il est donc possible de détecter une anomalie de comportement en procédant à l'analyse des dérives de comportement.

Sans cette mise en cascade, le conflit inhérent ne peut être analysé finement et il est impossible de déterminer l'origine de ce conflit : soit une défaillance capteur, soit une dérive comportementale.

La Figure 108 présente la mise en cascade de ces deux approches. Elle permet d'observer le rôle de l'affaiblissement des jeux de masse dans l'analyse des dérives de comportement. Grâce à cet affaiblissement, une partie du conflit, par le biais de l'ignorance totale, n'est plus véhiculée dans la chaîne globale d'analyse du comportement.

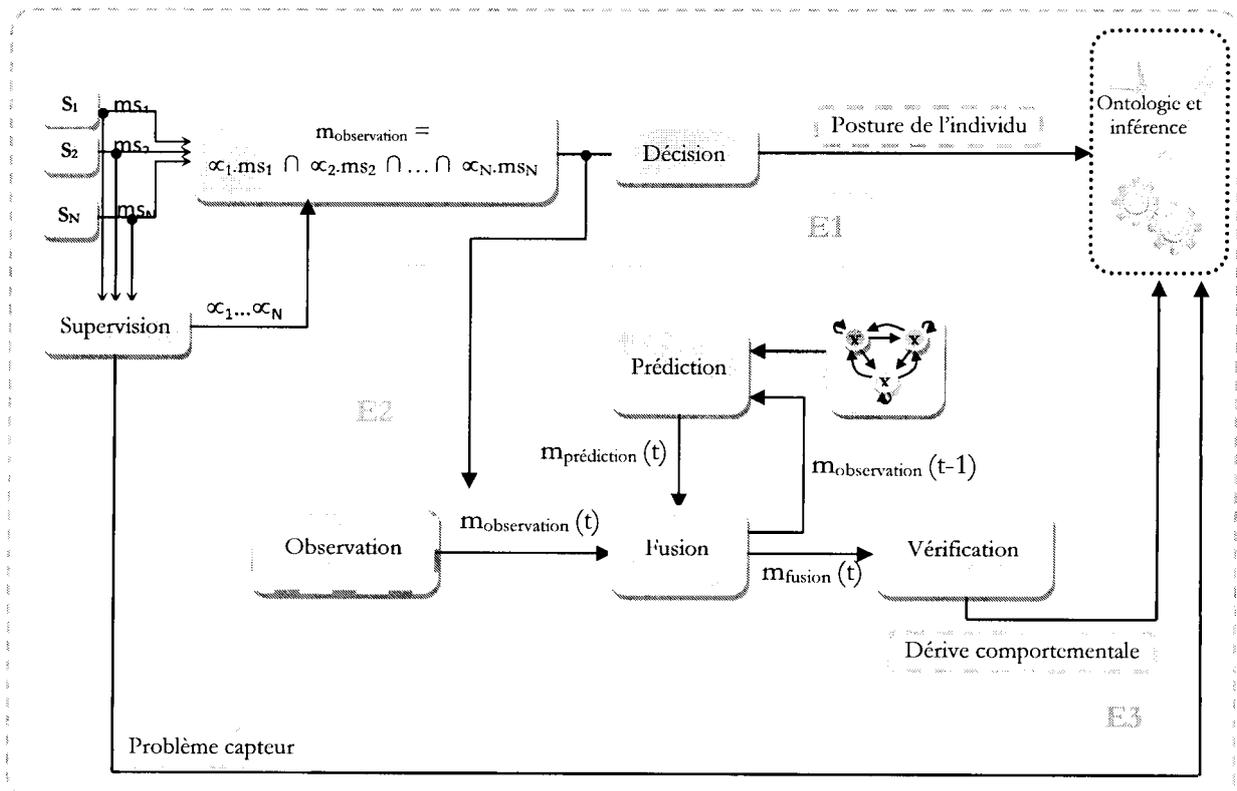


Figure 108 : Mise en cascade des approches présentées précédemment.

Grâce à cette architecture, nous pouvons réaliser l'acquisition de trois données contextuelles de haut niveau sémantique, à savoir la posture d'un individu à un instant précis, une dérive de comportement de cet individu et un problème capteur.

5 Conclusion

Dans ce quatrième chapitre, nous avons présenté et commenté les résultats expérimentaux obtenus concernant la mise en œuvre d'une architecture globale de fusion de données dédiée à la surveillance d'activités humaines. Nous avons focalisé sur trois points principaux.

Le premier porte sur la caractérisation de la posture d'un individu à un instant t à partir de trois capteurs fusionnés grâce au formalisme de la théorie des Croyances. Nous avons montré l'apport d'une approche à base de fusion de données pour déterminer des données contextuelles complexes, impossibles à déterminer à partir d'un capteur pris seul.

Le second point a abordé le problème de la fiabilité des capteurs utilisés. Nous avons montré l'impact d'un capteur défectueux dans un processus de fusion statique de N capteurs. Nous avons observé qu'un capteur défaillant peut fausser la prise de décision finale (obtenue par calcul de la probabilité pignistique) voir même rendre impossible la prise de décision lorsque l'ignorance est totale. L'affaiblissement du jeu de masse du capteur défaillant lors du processus de fusion final permet de diminuer la valeur de conflit et d'accroître la croyance envers la décision prise.

Le dernier point a permis de mettre en œuvre une méthode de détection de dérives comportementales à partir du mécanisme de prédiction Markovien. Nous avons montré la faisabilité de la méthode à partir de données expérimentales et nous avons comparé les deux approches de prédiction. Le couplage d'une prédiction Markovienne probabiliste et d'un diagnostic crédibiliste montre que le traitement temporel du signal de conflit reste complexe à analyser. La seconde approche, basée sur la traduction crédibiliste de notre Chaîne de Markov classique donne des résultats équivalents à la première approche mais fournit un signal de conflit beaucoup plus stable et cohérent, et donc plus interprétable. Le réglage des experts en charge d'analyser le signal de conflit est alors facilité.

Ce troisième point a permis de débattre sur le couplage des deux approches vues précédemment et a présenté l'impact d'un capteur défaillant sur la détection et l'isolation de dérives comportementales dans l'analyse de l'activité humaine. Nous avons montré qu'un capteur défaillant peut engendrer une détection et une caractérisation erronée de dérives de comportement.

Nous avons aussi montré d'une part qu'il est possible d'effectuer une séparation des sources à l'origine du conflit. D'autre part, nous montrons qu'en atténuant une source d'accroissement du conflit, il devient possible de détecter finement et de manière plus fiable des changements d'état par rapport à un flux hétérogène de données capteurs.

Conclusion générale

Rappel de la problématique

Dans le cadre de l'habitat communicant, la prise en compte du contexte est primordiale pour fournir le service le plus adapté possible. C'est en effet à partir d'une analyse issue des données capteurs, que l'informatique centralisée ou distribuée va construire un contexte et décider de l'action à entreprendre. Néanmoins, la complexité des phases de perception et d'interprétation nécessitent de passer par des combinaisons de contextes simples pour alimenter de façon la plus cohérente possible les couches services.

Dans ce mémoire, nous nous sommes attachés à définir une architecture complète pour la prise en compte du contexte dans l'habitat communicant. Nous avons en particulier focalisé notre travail sur le traitement et l'interprétation des données issues des capteurs afin d'alimenter les mécanismes d'inférence. La couche sensorielle permet d'acquérir une mesure numérique d'un phénomène observé. Cette problématique nous a permis de proposer des contributions pour chacun des sous-problèmes sous-jacents à notre gestion de contexte.

Après avoir présenté dans l'état de l'art ce que recouvre la notion d'intelligence ambiante et en particulier la sensibilité au contexte, nous avons positionné notre problématique sur la fusion de données symbolique. L'originalité de notre travail reside dans la prise en compte de l'incertitude par l'utilisation d'un formalisme de fusion crédibiliste. En outre il nous permet de statuer sur l'intégrité du système par une quantification de la notion de conflit à chaque inférence. Ainsi nous produisons une décision accompagnée d'un degré de confiance. Nous avons montré que la combinaison de sources d'information était sensible aux défaillances et aux dérives. L'absence de prise en compte de celles-ci induit inévitablement des constructions erronées de contexte. Elles aboutiront, si elles ne sont pas filtrées sur les niveaux les moins élevés, à une production de services non adaptés.

Le chapitre deux propose une architecture logicielle complète pour la prise en compte du contexte. Nous avons structuré un modèle en couches *PCLA* permettant une gestion complète du contexte allant de la perception de données sensorielles jusqu'à la prise de décision et la fourniture de services adaptés à l'habitant.

Nous avons présenté les enjeux et les difficultés de mise en place pour chacune des couches de ce modèle. La couche *Perception* est basée sur une approche à faible couplage qui utilise un bus logiciel *publish/subscribe* permettant un transit aisé et anonyme des données capteurs. Ces données sont ensuite utilisées par la couche *Contexte*. La principale difficulté rencontrée a été de définir un méta-modèle permettant de représenter le contexte à un instant précis. Un modèle à base d'ontologie écrit en *OWL* a été proposé afin de représenter l'habitat communicant à un instant t et de faciliter le mécanisme d'*Inférence*. Cette dernière est gérée par un langage de règles de la logique du premier ordre *SWRL*, qui interprète facilement une ontologie écrite en *OWL*. Elle offre la possibilité de définir le service le plus adapté. A partir des différentes couches du modèle *PCLA*, nous avons décrit un modèle fonctionnel reposant sur la plateforme logicielle de services *OSGi*. Les tests ont montré la faisabilité de l'architecture mais ont aussi permis de mettre en exergue le problème du temps de réponse du mécanisme d'inférence ainsi que les faiblesses de la logique du premier ordre.

La nature incertaine qui est propre à toute donnée issue des capteurs se doit d'être prise en compte non seulement au niveau de sa quantification et de la possibilité de la propager mais aussi au niveau de l'inférence. Pour ce faire, nous avons utilisé une méthode d'agrégation crédibiliste, à savoir la théorie des *Fonctions de Croyance*. Avant de généraliser cette approche crédibiliste à la totalité de l'architecture, nous nous sommes occupés de son déploiement sur les couches basses, à savoir la couche *perception*.

Le troisième chapitre a permis de répondre à la question de la combinaison de données sensorielles à partir du *Modèle des Croyances Transférables (MCT)*, une variante de la théorie *Fonctions de Croyance*. Nous avons décrit les étapes nécessaires à l'élaboration d'une architecture de fusion crédibiliste de données capteurs. La première étape consiste au prétraitement des données sensorielles et à leur conversion numérique vers symbolique afin d'obtenir des jeux de masse sur le même cadre de discernement Θ , condition principale dans la fusion de données. Les jeux de masse obtenus permettent de quantifier l'incertitude sur les données capteurs. La seconde étape consiste à combiner des jeux de masse pour obtenir un jeu unique et de renforcer la croyance vis-à-vis de cette nouvelle donnée contextuelle. Le conflit issu de la fusion est géré classiquement en reportant la masse conflictuelle sur l'ignorance totale. La troisième étape est la prise de décision. Dans le cadre du *MCT*, elle est réalisée à partir de la transformation pignistique. La décision est prise en choisissant l'hypothèse dont la probabilité pignistique est maximale. Le problème de la défaillance capteur que nous avons positionné dans la présentation de nos objectifs a été géré avec une approche originale intrinsèquement au *MCT*. En effet nous avons proposé des méthodes d'analyse variationnelle du conflit pour diagnostiquer et isoler des défaillances. Nous avons discerné deux types de défaillance : le premier intervient lorsque les données capteurs sont aberrantes à un instant donné; le second porte sur l'interprétation temporelle des données capteur.

Nous avons observé qu'un capteur défaillant fournissant des données aberrantes induit un conflit maximal issu de la fusion. Celui-ci étant reporté sur l'ignorance, nous constatons que la prise de décision devient délicate voire même impossible lorsqu'il est maximal. Nous avons alors proposé une méthode pour la détection et l'identification du capteur défaillant afin de pondérer son influence sur le processus de fusion final. En nous basant sur l'analyse temporelle du conflit résultant de la fusion deux à deux des capteurs, nous réussissons à détecter et à extraire le capteur en faute. À partir de son degré de défaillance, nous estimons un coefficient d'affaiblissement qui est injecté dans le processus de fusion final afin de pondérer l'influence du jeu de masse correspondant. Cette pondération permet de conserver le capteur fautif au sein de l'architecture de fusion de données tout en permettant une prise de décision pignistique.

Nous avons ensuite étudié le second cas portant sur une défaillance capteur au niveau de l'interprétation temporelle des données. Pour ce faire, nous avons proposé un algorithme de détection de dérives comportementales dans le cadre du *MCT*. À partir d'un modèle établi par apprentissage, nous effectuons une prédiction sur l'état à l'instant suivant. Cette prédiction fournit alors un jeu de masse fusionné avec le jeu de masse fourni par le capteur. L'analyse temporelle du conflit entre ces deux jeux de masse permet de détecter une dérive de comportement.

Nous avons défini un modèle de comportement normal du capteur prenant en compte la notion de temporalité et la variation des états symboliques. Un modèle de capteur à partir d'un processus stochastique a été proposé, et plus précisément, une Chaîne de Markov décrite par sa matrice de transition probabiliste. Ce modèle permet de définir un jeu de masse prédit en ayant au préalable converti le jeu de masse observé à l'instant précédent en un jeu de probabilité, autorisant le calcul d'un vecteur de probabilité prédit. Ce vecteur de probabilité prédit est ensuite converti en jeu de masse afin de pouvoir le fusionner avec le jeu de masse observé à l'instant suivant. Le conflit issu de cette fusion est ensuite analysé temporellement grâce à trois experts délivrant chacun un jeu de masse se prononçant sur une caractéristique du conflit. Nous avons observé que lorsqu'une dérive de comportement intervient, un saut

soudain du conflit apparaît. La fusion des trois experts permet de détecter ces sauts de conflits significatifs d'une dérive de comportement. Afin de rendre cette détection plus facile à réaliser, nous avons défini une matrice de transition au sens crédibiliste. Contrairement à l'approche probabiliste, restreignant la prédiction aux hypothèses singletons, cette version crédibiliste offre plus de souplesse dans la description du modèle du capteur. La prédiction est alors fortement améliorée et se prononce sur les hypothèses singletons ainsi que les disjonctions d'hypothèses. Cette nouvelle matrice de transition possède deux atouts. Premièrement, le processus de conversion pignistique du jeu de masse observé afin de calculer le jeu de masse prédit n'a plus lieu d'être. Deuxièmement, l'élaboration des experts est facilitée. En effet, avec la matrice de transition probabiliste, le réglage n'est pas aisé à réaliser et nécessite d'être très précis.

Dans le cadre de l'activité humaine, les deux types de défaillance sont complémentaires. C'est ce que nous montrons dans le quatrième chapitre où les résultats expérimentaux sont présentés. Nous nous sommes intéressés à la caractérisation des postures humaines : *debout*, *assis* et *allongé*. Une première partie présente le réseau de capteurs permettant de caractériser l'ensemble de ces postures. Nous avons ainsi décrit trois capteurs redondants et complémentaires : un capteur de pression mécanique placé sur le siège d'une chaise, une caméra à vision omnidirectionnelle placée au plafond de la pièce équipée et un accéléromètre placé sur la ceinture de la personne observée. Grâce à la combinaison de ces trois capteurs, nous parvenons à caractériser précisément les trois postures énumérées précédemment.

Afin de valider notre première approche pour la détection et l'identification de capteur défaillant, nous avons mis en défaut un capteur et observé le comportement de notre méthode. Un conflit important apparaît lors de la fusion des trois capteurs, empêchant de déterminer correctement une posture. Il s'avère que notre méthode est robuste et permet de détecter précisément le capteur en faute. La pondération de ce capteur permet de limiter son influence lors de la fusion finale et le processus de décision pignistique permet alors de déterminer une posture. D'autres essais montrent les limites de notre approche. En effet, lorsque les capteurs sont temporellement désynchronisés, il est possible que le capteur défaillant ne soit pas en conflit avec les deux autres. Dans ce cas, si deux capteurs sont détectés comme défaillants, ne pouvant pas précisément déterminer lequel est fautif, nous affaiblissons légèrement leur jeu de masse respectifs lors de la fusion finale. Ainsi, cette approche fournit un outil complet pour la détection et l'identification du capteur fautif au sein d'un réseau de capteurs dans le cadre d'un processus de fusion *MCT*.

La mise en cascade de notre deuxième méthode pour la détection de dérives comportementales permet de détecter un problème de comportement humain. Nous avons établi un comportement normal de l'enchaînement des postures humaines que nous avons traduit en une Chaîne de Markov probabiliste. Cette première version permet de détecter des enchaînements de postures théoriquement impossibles relativement à la Chaîne de Markov définie au préalable. L'analyse du conflit entre les jeux de masse observés et prédits est réalisée grâce à trois experts et fournit les résultats attendus, à condition d'un réglage précis de ces trois experts. Nous avons comparé cette première approche avec notre paradigme *MCT* de description de comportement. Il s'avère que cette deuxième version est aussi efficace que la première, sauf que le conflit entre observation et prédiction est beaucoup plus simple à analyser. Le réglage des experts est facilité et un simple seuillage sur l'amplitude du conflit peut, dans certains cas bien prononcés, permettre de détecter une dérive de comportement.

Le couplage de ces deux approches permet de montrer que la séparation des sources à l'origine du conflit est importante pour la surveillance du comportement humain. En effet, nous avons montré finalement qu'une défaillance capteur peut empêcher la détection de dérives comportementales. Lorsqu'une défaillance intervient, la combinaison des trois capteurs ne peut se prononcer et l'analyse de l'enchaînement des postures est alors impossible à réaliser.

Les travaux présentés dans ce mémoire proposant une architecture pour la prise en compte du contexte dans l'habitat communicant offrent plusieurs pistes pour de futurs travaux, aussi bien en terme d'améliorations qu'en terme de terrains d'applications.

Améliorations et perspectives

Les pistes permettant d'améliorer notre architecture sont multiples. Nous avons identifié **cinq** principales améliorations à ce travail.

La première porte sur **Parchitecture logicielle** permettant la prise en compte du contexte. Le principal inconvénient de notre approche à base d'ontologie est le caractère restrictif du langage de règles *SWRL*. Les règles du premier ordre ne permettent pas d'obtenir un raisonnement poussé. Il serait judicieux d'intégrer la notion d'imperfections des données capteurs directement dans l'ontologie et de prendre en compte la crédibilité des données directement au niveau de l'inférence. Une approche reposant sur un raisonnement du premier ordre est alors insuffisante.

La deuxième amélioration porte sur **Putilisation de Pontologie** au sein de notre architecture et en particulier le mode de raisonnement utilisé. Nos travaux ont présenté une méthode de raisonnement permettant de définir une action à partir d'un ensemble de contextes. Néanmoins, cette approche est trop restrictive et oblige la création de règles au préalable. Une approche duale est la logique intuitionniste dirigée par les buts. A partir d'un service voulu, cette méthode recherche les contextes appropriés et fournit l'action semblant la plus en adéquation. La logique intuitionniste permet de raisonner dynamiquement sur des connaissances partielles qui peuvent apparaître lors de défaillances du matériel.

La troisième amélioration porte sur **Pintégration des capteurs**. En effet, le matériel propriétaire oblige le développeur à s'adapter et à créer des ponts entre les différents matériels utilisés. Il serait judicieux d'intégrer le support *IPv6* (*Internet Protocol version 6*) pour le transport des informations, en particulier au niveau des capteurs. Ce standard permettrait de rendre transparents les protocoles propriétaires des constructeurs et faciliterait le travail d'intégration qui porterait simplement sur le contenu. La recherche va dans ce sens. Les protocoles sans fil destinés à un usage domestique, tels que *Zigbee*, se tournent vers *IPv6* actuellement. Ainsi, tout matériel devient interopérable (en théorie) grâce à ce protocole.

Au niveau de l'aspect conceptuel de notre architecture logicielle pour la prise en compte du contexte, nous allons nous heurter à un problème de taille concernant **les instanciations multiples d'un même service**. En effet, l'approche à service que nous avons choisi ne l'autorise pas. Pour pallier ce problème, nous proposons d'étendre le modèle à service en le couplant à un modèle à composant. Nous mêlons ainsi les avantages d'un modèle à composant, notamment la notion de fabrique permettant de créer des instances de composant et les avantages d'un modèle à service permet une gestion plus fine du cycle de vie. Ce couplage fournit alors une dynamique d'exécution accrue par rapport à une approche à composant seule.

Le cas multi-habitants n'est pas traité dans notre approche. En effet, nos travaux se sont bornés à un seul occupant. De même, nous ne considérons qu'une seule pièce de l'habitat communicant sans se soucier des précieuses informations contextuelles que pourraient fournir les autres pièces. Afin d'autoriser une approche multi-habitants au sein d'un habitat global, il faut étayer notre modèle d'ontologie en prenant en compte cette notion de multi-habitants au sein de l'habitat global. Notre ontologie offrant une grande souplesse, la gestion du multi-habitants peut être résolue grâce à une approche *ILS* : *Identification, Localisation et Suivi*. Le suivi de plusieurs personnes est une problématique relativement complexe. Deux méthodes peuvent être envisagées : équiper de manière dense et donc contraignante l'habitat ou utiliser peu de moyens matériels couplés à des algorithmes lourds. Un compromis semble envisageable. Nombre

de travaux sur le suivi multi-cibles existent, principalement dans les domaines de la robotique et du militaire. Il est possible de s'appuyer sur ces méthodes pour les appliquer au problème du suivi multi-habitants.

Les perspectives qu'offrent ces travaux sont aussi nombreuses. La première d'entre elles concerne **la fusion de données contextuelles**. En effet, en l'état actuel, notre processus de fusion est statique et restreint à une configuration initialement figée. Une réorganisation dynamique du processus de fusion en fonction des ressources disponibles et en particulier des capteurs ajoutés en fonction de nouveaux besoins est envisageable. Lorsque la fiabilité d'un capteur est jugée trop faible, celui-ci est écarté momentanément du processus de fusion et le système se met à la recherche d'un capteur qui pourrait remplir les mêmes fonctionnalités ou partiellement. Il est donc nécessaire de connaître l'ensemble des capteurs à disposition ainsi que leurs caractéristiques pour savoir comment ceux-ci pourraient être détournés. Par exemple, pour la détection de présence dans une pièce de l'habitat, si un détecteur de mouvements pyroélectrique devient non fiable, il peut être envisagé de détecter cette présence en observant les indices que laisse la personne : un claquement de porte ou la consommation d'eau, ... Cette perspective implique aussi bien une réflexion sur le déploiement dynamique au sein de l'architecture comme sur reconfiguration dynamique des règles de fusion.

Par rapport aux hypothèses de travail que nous nous sommes fixées, il en est une qui offre des perspectives très intéressantes pour les années à venir. Il s'agit de **l'apprentissage**. Celui-ci se situe à tous les niveaux de notre architecture. En effet, les préférences des habitants sont à prendre en compte au sein de l'ontologie afin de proposer un service adapté au contexte. Il est nécessaire d'observer pendant une période définie les agissements des individus afin de deviner leurs habitudes de vie. Après avoir identifié les paramètres pertinents, l'apprentissage permettra de définir des contextes complexes. De même, l'apprentissage se situe au niveau de l'élaboration des fonctions de masse. Chaque fonction de masse associée à un capteur nécessite de réfléchir au problème de sa distribution de masse. Dans le cadre des fonctions de croyance, l'apprentissage est un problème ouvert. Par rapport à l'aspect description de comportement par Chaîne de Markov, l'apprentissage automatique trouve sa place. En effet, qu'elle soit probabiliste ou crédibiliste, une Chaîne de Markov est décrite à partir de probabilités de transition qui sont, à ce jour, déterminées par expertise humaine.

Nous pensons que l'utilisation des *Séparateurs à Vaste Marge (SVM)* peut s'avérer judicieux. Ils sont aujourd'hui de plus en plus utilisés pour la classification automatique aux vues de leur simplicité et de leur capacité de généralisation. Plus la banque de données est importante, plus la classification sera efficace. Le couplage de ces deux approches permettrait de tenir compte des imperfections des données directement dans l'algorithme de classification. Cette piste semble prometteuse aux vues des premiers résultats dans ce domaine.

L'habitat seul ne peut répondre à toutes les attentes des habitants. Nous pensons que la **robotique domestique** trouvera sa place au sein d'un environnement communicant. Pris séparément, ces domaines ont des usages restreints, couplés, la gamme de services possibles se verrait étendue.

Bibliographie

- [AAR03] E. Aarts, S. Marziano. eds., « The New EveryDay : Visions of Ambient Intelligence », 010 Publishing, The Netherlands, 2003.
- [AGA05] D. Agathangelou, B.P.L. Lo, J.L. Wang, G.-Z. Yang – « Self- Configuring Video-Sensor Networks » - Adjunct Proceedings of the 3rd International Conference on Pervasive Computing (PERVASIVE 2005), pp. 29-32, May 2005.
- [AGN07] C.Anagnostopoulos, P.Pasias, S.Hadjieftymiades – « A Framework for Imprecise Context Reasoning » - in the proceedings of the IEEE International Conference on Pervasive Services, ICPS '07, Istanbul, Turkey, July, 2007.
- [ALM05] J.F. Al-Muhtadi – « An intelligent authentication infrastructure for ubiquitous computing environments » - Thèse de doctorat de l'Université de l'Université d'Illinois, Urbana-Champaign, 2005.
- [APP91] A. Appriou – « Probabilités et incertitudes en fusion de données multi-senseurs » - Revue Scientifique et Technique de la Défense, pages 27-40, 1991.
- [APP98] A. Appriou – « Uncertain data aggregation in classification and tracking processes » - In B. Bouchon-Meunier, editor, Aggregation and Fusion of imperfect information, pages 231-260. Physica-Verlag, Heidelberg, 1998.
- [AUD04] P. Audenino – « Apport de la fusion pour la qualité des Modèles Numériques de Terrain Radar dans un contexte multi modal » - Thèse de doctorat de l'Institut National de Polytechnique de Grenoble, soutenu le 29 octobre 2004.
- [BAR88] Y. Bar-Shalom and T. E. Fortmann – « Tracking and Data Association » - vol. 179 of Mathematics InScience And Engineering, Academic Press, Inc., 1988.
- [BAY1763] T. Bayes – « An Essay towards solving a Problem in the Doctrine of Chances » - Philosophical Transactions of the Royal Society of London, 53, 1763.
- [BEH06] N. B. Behloul – « Ajout de mécanismes de réactivité au contexte dans les intergiciels pour composants dans le cadre d'utilisateurs nomades », Thèse de Doctorat de l'Institut National des Télécommunications à Evry, Novembre 2006.
- [BEN93] E. Benoit – « Capteurs symboliques et capteurs flous : un nouveau pas vers l'intelligence » - Thèse de Doctorat, Université Joseph Fourier, Grenoble I, janvier 1993.
- [BER03] Gérald Berthoud , « A la recherche de l'homme nouveau ». Institut d'anthropologie et de sociologie, Université de Lausanne - Décembre 2003.
- [BEZ81] J.C. Bezdek – « Pattern Recognition with Fuzzy Objective Function Algorithms » - Plenum Press, New York, 1981.
- [BOB96a] A. Bobick, J. Davis. « An appearance-based representation of action ». Proceedings of the 13th International Conference on Pattern Recognition (ICPR'96), vol 1, pages 307-312, Vienna, Austria, Aout 1996
- [BOB96b] A.F. bobick. « Computers seeing action ». Rapport technique MIT, Media Laboratory, Perceptual Computing Section, Septembre 1996
- [BOB97] A. Bobick, J. Davis, S. Intille. « The Kidsroom : An example Application Using a Deep Perceptual Interface ». Dans M. Turk, éditeur, Proceedings of Workshop on Perceptual User Interface (PUI'97), pages 1-4, Banff , Alberta, Canada, Octobre 1997.
- [BOSSAM] « Bossam: A Rule/OWL Reasoner » - <http://projects.semwebcentral.org/projects/bossam/>
- [BRA00] I. Bratko – « PROLOG Programming for Artificial Intelligence », 2000, ISBN 0-201-40375-7.
- [BRA02] G. Bradski and J. Davis. « Motion Segmentation and Pose Recognition with Motion History Gradients ». International Journal of Machine Vision and Applications, Vol. 13, No. 3, 2002, pp. 174-184.
- [BRO85] R. Brooks - « Programming in Common Lisp » - John Wiley and Sons, 1985.
- [BUR00] S. Burbeck – « Evolution of web applications into service-oriented components with web services » - Octobre 2000. <http://www106.ibm.com/developerworks/library/wstao/index.html>
- [CAP01] L. Capra, W. Emmerich, C. Mascolo – « Reflective Middleware Solutions for Context-Aware Applications » - REFLECTION 2001. The Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns, volume 2192 of Incs, pages 126–133, Kyoto, Japan, September 2001.

- [CAP03] L. Capra, W. Emmerich, C. Mascolo – « CARISMA : Context-Aware Reflective mIddleware System for Mobile Applications » - IEEE Transactions on Software Engineering, 29(10) :929–945, oct 2003.
- [CAR06] Caron F., Davy M., Duflos E., Vanheeghe P. – « Fusion de capteurs potentiellement défaillants par filtrage particulière » - Réunion GDR-ISIS: Localisation et Navigation, Paris, 10 février 2006.
- [CHA03b] X. Chalandon. « Situation Awareness en conception système », ÉPIQUE'2003, Deuxièmes Journées d'Étude en Psychologie Ergonomique, 2 et 3 octobre 2003, Institut de Psychologie,, Boulogne Billancourt.
- [CHA05] Chapron L., Truillet P.- « Une taxonomie des capteurs contextuels physiques » - Annexe des actes d'IHM'2005, Toulouse, 27-30 septembre 2005.
- [CHE04a] H. Chen, T. Finin, A. Joshi – « An Ontology for Context-Aware Pervasive Computing Environments » - Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, 18(3) :197–207, May 2004.
- [CHE04b] H. Chen, F. Perich, D. Chakraborty, T. Finin, A. Joshi – « Intelligent Agents Meet Semantic Web in a Smart Meeting Room » - In Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2004), New York City, NY, July 2004.
- [CHO06] T. Choudhury, M. Philipose, D. Wyatt, J. Lester - « Towards Activity Databases: Using Sensors and Statistical Models to Summarize People's Lives » - In IEEE Data Engineering Bulletin, Vol. 29 No. 1, March 2006.
- [CHO08] T. Choudhury, G. Borriello, et al. – « The Mobile Sensing Platform: An Embedded System for Capturing and Recognizing Activities » - To appear in IEEE Pervasive Magazine - Special Issue on Activity-Based Computing, April 2008.
- [CLA04] Clauberg R.- « RFID and Sensor Networks » –RFID workshop, Univ. of St. Gallen, Sept. 2004.
- [CLE02] A. Clémentin, « Localisation d'un robot mobile par coopération multi-capteurs et suivi multi-cibles », Thèse de Doctorat de l'Université de Picardie Jules Vernes à Amiens, 2002.
- [COR02] Cornu J.M. « Les technologies de demain », cahiers de l'Internet– Tome I, FING - février 2002.
- [COU02] J. Coutaz, G. Rey – « Foundation for a Theory of Contextors » - in the Proceedings of the 4th International Conference on Computer-Aided Design of User Interfaces (CADUI02), ACM Publ., Valenciennes, France, May 2002, pp. 283.302.
- [COU05] Coutaz J., Crowley J., Dobson S., Garlan D. - « Context is key » -Commun. ACM, Vol. 48, No. 3. (March 2005)
- [DAN00] M. Daniel - « Distribution of Contradictive Belief Masses in Combination of Belief Functions » - Information, Uncertainty and Fusion, Kluwer Academic Publishers, 431~446, 2000.
- [DAV97] J. Davis and A. Bobick – « The representation and recognition of action using temporal templates » - In Conference on Computer Vision and Pattern Recognition, pages 928-934, Puerto Rico, June 1997.
- [DAV98] J. Davis – « Recognizing movement using motion histograms » - Technical Report 487, MIT Media Lab, April 1998.
- [DAV05] P.C. David – « Développement de composants Fractal adaptatifs : un langage dédié à l'aspect d'adaptation » - Thèse de Doctorat de l'école des mines de Nantes – Juillet 2005.
- [DEL07] F. Delmotte – « Detection of defective sources in the setting of possibility theory » - Fuzzy Sets and Systems, vol. 158, pp. 555–571, 2007.
- [DEM67] A.P. Dempster – « Upper and lower probabilities induced by a multivalued mapping » -Annals of Mathematical Statistics, 38, pp. 325-339, 1967.
- [DEN95] T. Denoeux – « A k-nearest neighbor classification rule based on Dempster-Shafer theory » - IEEE Transactions on Systems, Man and Cybernetics: vol. 25, pp. 804--813, 1995.
- [DEN02] T. Denoeux et A. Ben Yaghlane. Approximating the combination of belief functions using the fast Möbius transform in a coarsened frame. International Journal of Approximate Reasoning, 31(1-2):77–101, October 2002.
- [DEY01a] A. K. Dey, G. D. Abowd et D. Salber, « A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications », Human-Computer Interaction, vol. 16 (2-4), pp. 97-166, 2001.
- [DEY01b] Dey A.K.- « Understanding and using context » – Personal and Ubiquitous Computing, Vol 5, No. 1, pp 4-7, 2001.
- [DEZ02] J. Dezert – « Foundations for a new theory of plausible and paradoxical reasoning » - Information and Security Journal, 2002.

- [DEZ07] J. Dezert, F. Smarandache – « Advances and Applications of DSMT for Information Fusion » - Tutorial at Fusion 2007 Int. Conference on Information Fusion, Québec City, July 2007.
- [DOD99] Dodier, Robert H – « Unified prediction and diagnosis in engineering systems by means of distributed belief networks » - Thèse de doctorat de l'Université du Colorado – 1999.
- [DRO01] Drogoul A. et Servat D., « Intelligence diffuse : vers un monde fait d'agents ? », Actes des JFIADSM'01, A. El Fallah Seghrouchni et L. Magnin éd., pp. 227-238, Hermès, Paris, 2001.
- [DUB80] D. Dubois, H. Prade - « Fuzzy Sets & Systems: Theory and Applications » - Academic Press (APNet), Vol. V.144, 393 p., 1980.
- [DUB88] D. Dubois and H. Prade – « Representation and combination of uncertainty with belief functions and possibility measures » - Comput. Intell., vol. 4, pp. 244–264, 1988.
- [EUZ05] J. Euzenat – « Alignment infrastructure for ontology mediation and other applications », in Proc. Of the 1st ICSOC international workshop on Mediation in semantic web services, Amsterdam (NL), pp81-95, 2005.
- [EUZ06] Euzenat J., Pierson J., Ramparany F.- « Gestion dynamique de contexte pour l'informatique diffuse » - 15e congrès francophone de Reconnaissance des Formes et d'Intelligence Artificielle, Tours - RFIA 2006.
- [FAU89] Faust, N.J., « Reliability Analysis by Use of Markov Modeling » - RAC Technical Brief, March 1989.
- [FLORA] « FLORA-2 : An Object-Oriented Knowledge Base Language » - <http://flora.sourceforge.net/>
- [FOR82] Charles Forgy – « Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem » - in Artificial Intelligence, 19, pp 17-37, 1982.
- [GU04] Gu T., Hang Wang X., Keng Pung H., Da Quing Zhang- « An Ontology-based Context Model in Intelligent Environments » - In Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, San Diego, California, USA, January 2004.
- [GUI07] H. Guillaud – « Notre culture numérique transforme-t-elle notre intelligence ? », InternetActu.net, Enjeux, débats, prospective, Opinions - 16/11/2007. <http://www.internetactu.net/?p=7418>
- [HAL01] D. Hall, J. Llinas – « Handbook of Multisensor Data Fusion » - CRC Press (2001).
- [HEN04] K. Henriksen, J. Indulska – « Modelling and Using Imperfect Context Information » - PerCom Workshops, pages 33–37, 2004.
- [HEN05] K. Henriksen, T. McFadden – « Modelling context information with ORM » - Volume 3762 of Lecture Notes in Computer Science, pages 626–635, 2005.
- [HIG02] J. Hightower, B. Brumitt, G. Borriello – « The location stack: a layered model for location in ubiquitous computing » - in Fourth IEEE Workshop on Mobile Computing Systems and Applications, 2002, pp. 22–28.
- [HOR04] Horrocks I. et al. – « SWRL: A Semantic Web Rule Language - Combining OWL and RuleML » - W3C Member Submission, <http://www.w3.org/Submission/SWRL/> - May 2004.
- [INF06] Infantes G., Ingrand F., Ghallab M. – « Learning stochastic models of activities for planning and execution control » - RFIA 2006, January 25-27 2006, Tours, France.
- [JAN96] Janez F. – « Fusion de sources d'information définies sur des référentiels non exhaustifs différents » - Thèse de doctorat de l'Université d'Angers, Novembre 1996.
- [JEA96] Jean F., Mourot G. – « Estimation d'Etat Robuste: Application à la Détection de Défauts de Capteurs » - AGI'96, Forum des Doctorants Automatique, Génie Informatique, Image, Tours-E3i, Juin 1996, pp. 91-96.
- [JENA] « Jena – A Semantic Web Framework for Java » - <http://jena.sourceforge.net/ontology/index.html>
- [JUR04] Jurdak R., Videira Lopes C., Baldi P.- « A framework for modeling sensor networks » - OOPSLA'04, Vancouver, Canada 2004.
- [JØS03] A. Jøsang, M. Daniel and P. Vannoorenberghe – « Strategies for Combining Conflicting Dogmatic Beliefs » - Proceedings of the 6th International Conference on Information Fusion, Cairns 8-11 July 2003.
- [KAON2] « KAON2 -- Ontology Management for the Semantic Web » - <http://kaon2.semanticweb.org/>
- [KEN90] R. Kennes et Ph. Smets – « Fast algorithms for Dempster-Shafer theory » - In B. Bouchon-Meunier, R.R. Yager et L. A. Zadeh, éditeurs, Uncertainty in knowledge bases, pages 14–23. Springer Verlag, Berlin, 1990.
- [KOH00] Kohler, M.A. Yarlagadda, R.K. – « Markov chain prediction for missing speech frame compensation » - IEEE Workshop on Speech Coding – pp 75-77, Delavan, WI, USA, 2000.

- [KOK06] M. Kokar – « Ontology Based Situation Awareness and High Level Fusion: Methods and Tools » - Workshop on Information Fusion (Fusion06), Florence (Italy), 10-13 July 2006.
- [KOR00] M. Korkea-aho - « Context-Aware Applications survey » - 2000. <http://www.hut.fi/~mkorkeaa/doc/context-aware.html>
- [KOR03] P. Korpipaa, J. Mantjarvi, J. Kela, H. Keranen, E.-Juhani – « Malm, Managing Context Information in Mobile Devices » - IEEE Pervasive Computing, Volume 2 Number 3, July-September 2003, pp. 42-51, ISSN 1536-1268.
- [LAT97] M. Lataste – « Le projet d'hospitalisation à domicile : application à l'Aquitaine » - Bulletin Société de Pharmacie de Bordeaux, 1997, 136, 99-128.
- [LAU00] Launay P.- « Déploiement d'un bus à messages sur un réseau à grande échelle » - Rapport de DEA Informatique : Systèmes et Communication, Grenoble, juin 2000.
- [LAW04] G. Lawton. « Machine-to-Machine Technology Gears Up for Growth ». IEEE Computer, vol. 37:12–15, September 2004.
- [LEE00] Lee K. – « IEEE 1451: A Standard in Support of Smart Transducer Networking » - IEEE Instrumentation and Measurement Technology conference Baltimore, MD USA, May 1-4, 2000.
- [LEE02] D.L. Lee, « Technological and business challenges in pervasive computing », Proceeding of the 2002 Symposium on Applications and the Internet (SAINT'02) – 2002.
- [LEF01] Lefevre E., Colot O., Vannoorenbergh P., De Brucq D. – « Informations et combinaison : les liaisons conflictuelles » - Traitement du signal, vol. 18, n°3; pp 161-177, 2001.
- [LI06] Li Z., Koh B.H., Nagarajaiah S., Phan M. Q. – « Sensor failure detection using interaction matrix formulation » - Smart Structures and Materials 2006: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems. SPIE, Volume 6174, pp. 430-440 (2006).
- [LIC60] Licklider J. « Man-computer symbiosis », IRE Transactions on Human Factors in Electronics, Volume HFE-1, pp. 4-11, 1960.
- [MAN03] K. Manabe, T. Matsui, M. Yamaya, T. Sato-Nakagawa, N. Okamura, H. Ari, and H.Sasaki. « Sleeping patterns and mortality among elderly patients in geriatric hospitals ». Gerontology, 46:318–322, 2000. Cité dans la thèse de Master de E. Tapia (MIT, 2003).
- [MAR05] M. Matin, N. Hussain and R. Shoureshi - "Fiber Bragg sensor for smart bed sheet" - Photonic Devices and Algorithms for Computing VII. Edited by Iftekharuddin, Khan M.; Awwal, Abdul A. S. Proceedings of the SPIE, Volume 5907, pp. 36-43, 08/2005.
- [MAR06] A. Martín et C. Osswald – « A new generalization of the proportional conflict redistribution rule stable in terms of decision » - in Advances and Applications of DSMT for Information Fusion (Collected works) (F. Smarandache et J. Dezert, eds.), vol. 2, pp. 69–88, American Research Press, Rehoboth, 2006.
- [MAS07] S. Maskell – « A Bayesian Approach to Fusing Uncertain, Imprecise and Conflicting Information » - Published online in Information Fusion Journal, doi:10.1016/j.inffus.2007.02.003, 2007.
- [MAU04] J. Mauclair, J. Rinquier – « Fusion de paramètres en classification paroles/musiques » - Proceedings des XXV ièmes Journées d'Etudes sur la Parole (JEP), Maroc, Apr. 2004.
- [MCC43] W. S. McCulloch, W. Pitts – « A logical calculus of ideas immanent in nervous activity » - Bull. Mathematical Biophysics, 5:-133, 1943.
- [MCC93] J. McCarthy - « Notes on Formalizing Contexts » - In T. Kehler and S. Rosenschein, editors, Proceedings of the Fifth National Conference on Artificial Intelligence, pages 555–560, Los Altos, California, 1993. Morgan Kaufmann.
- [MCC94] J. McCarthy, S. Buvac – « Formalizing Context (Expanded Notes) » - Technical report, Stanford, CA, USA, 1994.
- [MER08] D. Mercier, B. Quost, T. Denœux – « Refined modeling of sensor reliability in the belief function framework using contextual discounting » - Information Fusion, vol. 9, pp. 246-258, 2008.
- [MOK07] M. MOKHTARI, M.A. FEKI – « User needs and usage analysis in a smart environment for people requiring assistance » - Topics in Geriatric Rehabilitation, January/March 2007, Volume 23, Number 1, p.52-59.
- [MUS05] D. Musicki, « Limits of Linear Multitarget Tracking » - Proceedings of the 7th International Conference on Information Fusion, July 2005.

- [NAK07] S. Nakano, G. Ueno, and T. Higuchi – « Merging particle filter for sequential data assimilation » - *Nonlinear Processes in Geophysics*, 14, p. 395, 2007.
- [NED03] V. Nedelec, S. Corgnes, V. Dubreuil, L. Hubert-Moy – « Fusion de données de télédétection à l'aide de la théorie des Evidences, Application au Mato Grosso en vue d'une spatialisation de la déforestation » - XI Simposio Brasileiro de Sensoriamento Remoto, Belo Horizonte, 5 a 10 de Abril 2003.
- [NGU01] H. T. Nguyen, M. Worring and R. van den Boomgaard – « Occlusion Robust Adaptive Template Tracking » - In *Proceedings IEEE Conf. Computer Vision 1*, pages 678-683, 2001.
- [NIS00] NIST, « *Pervasive Computing 2000* ». <http://www.nist.gov/pc2000/>
- [NUR06] P. Nurmi, A. Salden, S. Lun Lau, J. Suomela, M. Sutterer, J. Millerat, M. Martin, E. Lagerspetz, R. Poortinga – « A System for Context-Dependent User Modeling » - In *Proceedings of On the Move (OTM) Workshops*, pp. 1894-1903, vol. 4278 in *Series Lecture Notes in Computer Science*, Springer-Verlag, Montpellier, France, Oct. 2006.
- [PAN05] Pan J.Z., Stamou G., Tzouvaras V., Horrocks I. – « f-SWRL: A Fuzzy Extension of SWRL ». In *Proc. of the International Conference on Artificial Neural Networks (ICANN 2005)*, pages 829-834. 2005.
- [PAS97] J. Pascoe – « The stick-e note architecture : extending the interface beyond the user »- IUI '97 : *Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 261–264, New York, NY, USA, 1997. ACM Press.
- [PAS98] Pascoe J.- « Adding Generic Contextual Capabilities to Wearable Computers » - in *Proceedings of 2nd International Symposium on Wearable Computers*, October 1998.
- [PELLET] « Pellet OWL Reasoner » - <http://www.mindswap.org/2003/pellet/index.shtml>
- [PHA06] T. A. Le Pham, N. Le Than – « Raisonement sur des ontologies décomposées » - Rapport de recherche INRIA ISRN I3S/RR-2006-15-FR, Projet Mainline, 2006.
- [PLA06] D.J. Plas , M. Verheijen, H. Zwaal, M. Hutschemaekers - « Manipulating context information with SWRL » - Déliverable D3.12 du projet A- A-MUSE, 2006.
- [PRA04] H. Prade, G. Richard, M. Serrurier – « Enrichir la programmation logique inductive avec des prédicats flous » - 14ème Congrès « Reconnaissance des Formes et Intelligence Artificielle » (RFIA'04), Toulouse, France, 28/01/2004-30/01/2004, p. 319-327, janvier 2004.
- [RAN02] Ranganathan A., Campbell R.H. – « A Middleware for Context-Aware Agents in Ubiquitous Computing Environments » - *USENIX International Middleware Conference*, 2002.
- [RAM06] Ramasso E., Rombaut M., Pellerin D.- « Un filtre temporel crédibiliste pour la reconnaissance d'actions humaines dans les vidéos » - LFA'06, Toulouse, 19-20 octobre 2006.
- [RAM07a] E. Ramasso, M. Rombaut, D. Pellerin – « State filtering and change detection using TBM conflict - Application to human action recognition in athletics videos » - *IEEE Transactions on Circuits and Systems for Video Technology*, volume 17, issue 7, pages 944-949, 2007.
- [RAM07b] E. Ramasso – « Reconnaissance de séquences d'états par le Modèle des Croyances Transférables. Application à l'analyse de vidéos d'athlétisme » Thèse de doctorat de l'université Joseph-Fourier - Grenoble I – 2007.
- [REN07] S. Renouard – « Interaction Homme-Environnement : Modèle et outil » - Thèse de Doctorat en Informatique, INT-Université d'Evry, Avril 2007.
- [REY04] Rey G., Coutaz J.- « Le Contexteur : Capture et distribution Dynamique d'Informations Contextuelles » - *Mobilité et Ubiquité (Ubimob04)*, ACM Publication, 2004.
- [ROG94] G. Rogova – « Combining the results of several neural networks classifiers » - *Neural Networks*, 7(5):777–781, 1994.
- [ROM01] M. Rombaut – « Sur l'utilisation de la théorie de l'évidence pour la perception et la fusion de données » - exposé de la journée GDR ISIS du 18 Janvier 2001, ENST Paris.
- [ROM06] M. Rombaut, E. Ramasso, F. Maussang - « La théorie des fonctions de croyance : une alternative aux probabilités pour la perception et la fusion de données » - *Dossier spécial REE. Revue de l'électricité et de l'électronique : « Du traitement numérique à la gestion des connaissances : de nouvelles voies d'investigation ? »* - ISSN 1265-534 *Revue de l'électricité et de l'électronique* - 2006, n°8, pp. 42-50.

- [RYA05] N. Ryan – « Smart environments for cultural heritage » - In Takao UNO, editor, Reading Historical Spatial Information from around the World: studies of culture and civilization based on geographic information systems data. International Research Center for Japanese Studies, Kyoto, Japan, February 2005.
- [SAM05] N. Samaan, A. Karmouch – « A Mobility Prediction Architecture Based on Contextual Knowledge and Spatial Conceptual Maps » - IEEE Transaction on Mobile Computing, 4, 537-551. 2005.
- [SAT05] Satoh I., « Mobile Agents for Ambient Intelligence », to appear in Postproceedings of International Workshop on Massively Multi-Agent Systems (MMAS'2004), pp.187-201, Lecture Notes in Computer Science (LNCS), vol.3446, Springer, 2005.
- [SAV07] D. Savio, T. Ludwig - « Smart Carpet: A Footstep Tracking Interface » - AINAW '07, Volume 2, Pages 754-760, Niagara Falls, Canada, 21-23 May 2007.
- [SCH93] B. Schilit, M. Theimer, B. Welch - « Customizing Mobile Application » - USENIX Symposium on Mobile and Location-independent Computing, pages 129–138, Cambridge, MA, US, 1993.
- [SCH99] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. V. Laerhoven, W. V. de Velde – « Advanced Interaction in Context » - HUC '99 : Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, pages 89–101, London, UK, 1999. Springer-Verlag.
- [SCH00] Schmidt D.C., Stal M., Rohnert H, Buschmann F. – « Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects » – vol.2, Wiley & Sons, 2000.
- [SEK07] O.Sekkas, C.Anagnostopoulos, S.Hadjiefthymiades – « Context Fusion through Imprecise Reasoning » - in the proceedings of the IEEE International Conference on Pervasive Services, ICPS '07, Istanbul, Turkey, July 2007.
- [SIE04] F. Siegemund - « A Context-Aware Communication Platform for Smart Objects » -Pervasive 2004, Vienna, Austria, April 2004.
- [SIN02] M. Sintek, S. Decker – « TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web » - In ISWC '02 : Proceedings of the First International Semantic Web Conference on The Semantic Web, pages 364–378, London, UK, 2002. Springer-Verlag.
- [SHA76] G.A. Shafer - « A mathematical theory of evidence », Princeton university press, 1976.
- [SHE05] Q. Z. Sheng, B. Benatallah – « ContextUML : A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services ». 4th International Conference on Mobile Business (ICMB'05), IEEE Computer Society. Sydney, Australia., July 11-13 2005.
- [SMA05] F. Smarandache, J. Dezert – « Information Fusion Based on New Proportional Conflict Redistribution Rules » - Presented at the 2005 Int. Conf. Information Fusion, Philadelphia, PA, USA. 2005.
- [SMA06] F. Smarandache et J. Dezert – « Proportional Conflict Redistribution Rules for Information Fusion » - in Advances and Applications of DSMT for Information Fusion (Collected works) (F. Smarandache et J. Dezert, eds.), vol. 2, pp. 3–68, American Research Press, Rehoboth, 2006.
- [SME90] P. Smets – « The combination of evidence in the transferable belief model » -IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 5, pp. 447–458, 1990.
- [SME95] Smets Ph. – « The Transferable Belief Model for uncertainty representation » - Rapport technique, 1995.
- [SME98] Ph. Smets, «The Transferable Belief Model for Quantified Belief Representation», Handbook of Defeasible Reasoning and Uncertainty Management Systems, Kluwer Ed, pp 267-301, 1998.
- [SME99] Ph. Smets – « Theories of Uncertainty » - Handbook for Fuzzy Computation, 1999.
- [SME05] Ph. Smets – « Decision making in the TBM: The necessity of the Pignistic Transformation »- International Journal of Approximate Reasoning, Volume 38, Issue 2, February 2005, Pages 133-147.
- [STI07] M. Stikic, K. Van Laerhoven – « Recording Housekeeping Activities with Situated Tags and Wrist-Worn Sensors: Experiment Setup and Issues Encountered » - in International Workshop on Wireless Sensor Networks for Health Care (WSNHC 2007), Braunschweig, Germany, 2007
- [STI08] M. Stikic, T. Huynh, K. Van Laerhoven, B. Schiele - « ADL Recognition Based on the Combination of RFID and Accelerometer Sensing » - to appear in 2nd International Conference on Pervasive Computing Technologies for Healthcare (Pervasive Health 2008), Tampere, Finland, 2008.
- [SUB07] J. Subercaze, P. Maret, N.M. Dang, K. Sasaki – « Context-aware applications using personal sensors » - Bodynets, Florence, Italie, 2007.

- [SZY98] C. Szyperski – « Component software : beyond object-oriented programming » - ACM Press/Addison-Wesley PublishingCo. – 1998.
- [THO89] H. M. Thoma – « Factorization of belief functions » - Thèse de doctorat, Harvard University, Cambridge, MA, USA, 1989.
- [USD91] U.S. Department of Defense, Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3, "Data fusion lexicon," 1991.
- [VIA07] W. Viana – « Adaptation de services pervasifs et sensibles au contexte » - Forum Jeunes Chercheurs, Inforsid 2007.
- [W3C04a] W3C – « Resource Description Framework (RDF) : Concepts and Abstract Syntax » - <http://www.w3.org/TR/rdf-concepts/>, W3C Recommendation 10 February 2004.
- [W3C04b] W3C – « OWL Web Ontology Language Use Cases and Requirements » - <http://www.w3.org/TR/webont-req/>, W3C Recommendation 10 February 2004.
- [WAL99] L. Wald – « A European proposal for terms of reference in data fusion » - International Archives of Photogrammetry and Remote Sensing, Vol. XXXII, Part 7, 651-654, 1998, or Wald L., Some terms of reference in data fusion. IEEE Transactions on Geosciences and Remote Sensing, 37, 3, 1190-1193, 1999.
- [WAN04] X.H. Wang, T. Gu, D.Q. Zhang, H.K. Pung, « Ontology Based Context Modeling and Reasoning using OWL » - Proceedings of the Workshop on Context Modeling and Reasoning (CoMoRea'04), March 2004.
- [WAN04b] X. H. Wang, D. Q. Zhang, T. Gu, H. K. Pung –« An Ontology Based Context Model in intelligent Environment » - In Communication networks and Distributed System modeling and simulation Conference (CNDS 2004), pages 270–275, San Diego, California, USA, January 2004.
- [WAN05] J.L. Wang, B.L. Lo, G.-Z. Yang « Ubiquitous Sensing For Posture/Behavior Analysis », the 2nd International Workshop on Body Sensor Networks (BSN 2005), April 2005.
- [WAR06] N. Wartelle, F. Delmotte, D. Gacquer– « Classification de scènes à nombre d'objets inconnus » - Actes Conférence MajecStic (MANifestation Des Jeunes Chercheurs STIC, 22-24 Novembre), novembre 2006.
- [WEI91] Weiser M. « The Computer for the Twenty-First Century », Scientific American., pp. 94-104. September 1991.
- [WEI96] Weiser M. & Brown. « Designing Calm Technology », PowerGrid Journal, v1.01 - July 1996.
- [WOO06] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, J. Stankovic - « ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring » -Technical Report CS-2006-11, Department of Computer Science, University of Virginia, 2006.
- [WU02] H. Wu, M. Siegel, R. Stiefelhagen, J. Yang – « Sensor Fusion Using Dempster-Shafer Theory » - IEEE International Measurement Technology Conference (IMTC) 2002, Anchorage AK USA, 2002.
- [WU03] H. Wu, M. Siegel, S. Ablay – « Sensor Fusion using Dempster-Shafer Theory II: Static Weighting and Kalman Filter-like Dynamic Weighting » - IMTC (IEEE annual Instrumentation and Measurement Technology Conference) 2003 proceedings, Vail, CO USA, May 20-22, 2003.
- [WU07] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, J. Rehg – « A Scalable Approach to Activity Recognition Based on Object Use » -ICCV 2007. October 2007, Rio de Janeiro, Brazil.
- [YAN06] Yang H., Jansen E., Helal S., Mann W.- « An IDE for Programmable Pervasive Spaces Based on a Context-Driven Programm » - The 4 th IEEE Int. Conf. on Pervasive Computing and Communications (PerCom), Italy, March 2006.
- [YAG83] R. R. Yager – « Hedging in the Combination of Evidence » - Journal of Information and Optimization Sciences, vol. 4, no. 1, pp. 73–81, 1983.
- [YAG87] R. Yager - « On the Dempster-Shafer framework and new combination rules » - Information Sciences, vol. 41, pp. 93–138, 1987.
- [ZAB05] H. El-Zabadani, A. Helal, B. Abudrazak, and E. Jansen – « Self-Sensing Spaces: Smart Plugs for Smart Environments » - Proceedings of the third International Conference On Smart homes and health Telematic (ICOST), Sherbrooke, Québec, Canada, July 2005.
- [ZAB06] H. El-zabadani, S. Helal, M. Schmalz, and W. Mann, "PerVision: An integrated Pervasive Computing/Computer Vision Approach to Tracking Objects in a Self Sensing Space," Presented in the 4th

International Conference On Smart Homes and Health Telematics (ICOST), June 2006, Belfast, Northern Ireland, UK.

[ZAB07] H. El-Zabadani, A. Helal and H Yang, "A Mobile Sensor Platform Approach to Sensing and Mapping Pervasive Spaces and their Contents," Proceedings of the International Conference on New Technologies of Distributed Systems, Morocco, Marrakesh, June 4-8, 2007.

[ZHU02] Zhu, Jianhan and Hong, Jun and Hughes, John G. – « Using Markov Chains for link prediction in adaptive web sites » - Soft-Ware 2002 Belfast, Northern Ireland, April 8-10, 2002. Lecture Notes in Computer Science, 2311. Springer, pp. 60-73.

[ZIM04] T. Zimmer, « Towards a Better Understanding of Context Attributes » - Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW04), 2004.

Bibliographie de l'auteur

Workshops

[RIC07a] V. Ricquebourg, D. Durand, D. Menga, B. Marhic, L. Delahoche, C. Logé, A.M. Jolly-Desodt – « Context inferring in the Smart Home : An SWRL approach » - The 2007 IEEE International Symposium on Ubiquitous Computing and Intelligence (UCI07) in conjunction with IEEE AINA 2007 - May 21-23 2007 - Niagara Falls, Canada.

[RIC06d] V. Ricquebourg, D. Menga, L. Delahoche, B. Marhic, D. Durand, C. Logé – « Service Oriented Architecture for Context Perception Based on Heterogeneous Sensors Network » - IECON'06 - Special Session Distributed Systems and robotics - IEEE Industrial Electronics Society - November 7-10 2006 - Paris – France.

Congrès internationaux avec comité de lecture

[RIC05a] Ricquebourg V., B. Marhic, L. Delahoche, D. Menga, G. Brisson – « The Robotics/Smarthome duality » - The 11th International Conference on Information Systems Analysis and Synthesis: ISAS 2005 and The 2nd International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2005 - July 14 - 17, 2005 in Orlando, Florida, USA.

[RIC05b] V. Ricquebourg, B. Marhic, L. Delahoche, D. Menga, G. Brisson, E. Brassart – « Robotics/Smarthome duality : Potential application to assistive technologies » - AAATE 2005 - Assistive technology from virtuality to reality 8th European conference for the advancement of assistive technology in Europe Henri Warembourg Faculty of Medicine - Lille - France - September 6-9, 2005.

[RIC06b] V. Ricquebourg, D. Menga, D. Durand, B. Marhic, L. Delahoche, C. Logé – « The Smart Home Concept : our immediate future » - ICELIE'06 - First International Conference on E-Learning in Industrial Electronics - IEEE Industrial Electronics Society - December 18-20 2006 - Hammamet – Tunisia.

[RIC07b] V. Ricquebourg, M. Delafosse, L. Delahoche, B. Marhic, A.M. Jolly-Desodt, D. Menga - « Fault Detection by Combining Redundant Sensors : a Conflict Approach Within the TBM Framework » - COGIS 2007, COGNitive systems with Interactive Sensors - Stanford University , California, USA from 26 - 28 November, 2007.

[RIC07d] V. Ricquebourg, M. Delafosse, L. Delahoche, B. Marhic, A.M. Jolly-Desodt, D. Menga - « A Stochastic Approach for the Sensor Failure Detection within the TBM Framework » - COGIS 2007, COGNitive systems with Interactive Sensors - Stanford University , California, USA from 26 - 28 November, 2007.

[RIC07e] V. Ricquebourg, D. Menga, L. Delahoche, B. Marhic, D. Durand, C. Logé, A.M. Jolly-Desodt – « Service Oriented Architecture For Sensors Based Applications » - Human07 - March 12-17 2007 - Timimoun – Algeria.

[RIC08b] V. Ricquebourg, M. Delafosse, B. Marhic, L. Delahoche, A.M. Jolly-Desodt, D. Menga - « Sensor Failure Detection within the TBM Framework: A Markov Chain Approach » - IPMU 2008, Information Processing and Management of Uncertainty in Knowledge-Based Systems - June 22-27, 2008, Malaga, Spain.

[RIC08c] V. Ricquebourg, L. Delahoche, B. Marhic, M. Delafosse, A.M. Jolly-Desodt, D. Menga – « Anomalies Recognition in a Context Aware Architecture based on TBM Approach » - Fusion 2008, The 11th International Conference on Information Fusion – June 30 - July 03 2008 - Cologne, Germany.

Congrès nationaux avec comité de lecture

[RIC05c] V. Ricquebourg, L. Delahoche, B. Marhic, A.M. Jolly-Desodt – « Suivi de personnes grâce à une grille d'occupation utilisant la théorie de Dempster-Shafer » - Journées SEE - 8-9 décembre 2005, Paris – France.

[RIC06a] V. Ricquebourg, D. Menga, L. Delahoche, B. Marhic, D. Durand, C. Logé – « Architecture de perception de contexte orientée service pour l'habitat communicant » - UBIMOB 2006, ACM Press.5-8 Septembre 2006.

[RIC06c] V. Ricquebourg, B. Marhic, L. Delahoche, D. Menga, C. Logé – « L'habitat intelligent au service du maintien à domicile : Application au suivi d'individus » - HANDICAP 2006 - 4ème Conférence «Nouvelles Technologies au service de l'homme» - 7-9 Juin 2006 - Paris – France.

[RIC07c] V. Ricquebourg, M. Delafosse, L. Delahoche, B. Marhic, A.M. Jolly-Desodt, D. Menga - « Combinaison de sources de données pour la détection de dysfonctionnement capteur » - LFA'07, Rencontres Francophones sur la Logique Floue et ses Applications - 22-23 Novembre 2007 - Nîmes, France.

[RIC08a] V. Ricquebourg, D. Durand, D. Menga, L. Delahoche, B. Marhic, C. Logé, A.M. Jolly-Desodt – « La fusion multi-capteurs dans l'habitat communicant: une approche non-probabiliste » - Conférence UBIMOB 2008, Quatrièmes Journées Francophones: Mobilité et Ubiquité 2008, ACM Press. Pages 9-16, 28 - 30 Mai 2008.

[RIC08d] V. Ricquebourg, L. Delahoche, B. Marhic,, D. Menga, A.M. Jolly-Desodt – « Une approche non-probabiliste pour la fusion multi-capteurs dans l'habitat communicant » - HANDICAP 2008 - 5ème Conférence «Nouvelles Technologies au service de l'homme» - 10-12 Juin 2008 - Paris – France.

[RIC08e] V. Ricquebourg, D. Menga, L. Delahoche, B. Marhic, A.M. Jolly-Desodt – « La fusion de données non-probabiliste : application à la maison communicante » - 6e édition du colloque Capteurs 2008- 5 et 6 Mars 2008 - Bourges, France.

[RIC08f] V. Ricquebourg, L. Delahoche, B. Marhic, A.M. Jolly-Desodt, D. Menga - « Détection d'anomalies de comportement capteur dans une architecture MCT » - LFA'08, Rencontres Francophones sur la Logique Floue et ses Applications - 16-17 Octobre 2008 - Lens, France.

FUSION DE DONNEES CREDIBILISTE DANS LE CADRE DE L'INTELLIGENCE AMBIANTE

Résumé

Le récent concept d'habitat communicant se propose de fournir une offre de services dits contextualisés à l'habitant. Reposant sur un réseau de capteurs disposés au sein de l'environnement, ce nouveau type de « maison intelligente » se base sur l'interprétation de données sensorielles afin de déduire des actions appropriées à réaliser. La première problématique que nous traitons concerne la forte hétérogénéité issue du domaine des capteurs : ceux-ci possèdent leurs propres particularités matérielles, logicielles, et les normes de communications sont faiblement standardisées. Dans cette thèse, nous nous intéressons à la modélisation du contexte, en élaborant une architecture logicielle orientée services combinant des capteurs complémentaires et/ou redondants. Pour pallier la nature incertaine des ces ensembles d'informations et pour les combiner, nous employons le *Modèle des Croyances Transférables (MCT)*, une variante de la théorie de Dempster-Shafer. La fiabilité des données capteurs est également prise en compte au sein du processus de fusion : il nous permet de pondérer l'influence d'un capteur défaillant, qui pourrait empêcher la construction de données contextuelles. Cette fiabilité est estimée à partir de la fusion des données deux à deux des capteurs composant le réseau de perception. Une analyse temporelle des conflits issus des différentes fusions nous aide à détecter et à identifier le capteur en cause. Nous proposons également une seconde approche, consistant à détecter des dérives temporelles de comportement. A partir d'un modèle de comportement connu au préalable, nous réalisons la fusion (au sens du *MCT*) entre un état symbolique *prédit* et l'état symbolique *observé* par le capteur étudié. L'analyse temporelle du conflit issu de cette fusion permet de détecter des dérives comportementales. Nous proposons enfin un cas d'application mettant en cascade les approches présentées dans le cadre de la surveillance de l'activité humaine, en traitant le cas concret de la détection de la chute d'une personne.

Mots-clés

Habitat communicant, Contexte, Ontologie, OSGi, Fusion de données, Modèle des Croyances Transférables, Détection de défaillances.

CREDIBILIST DATA FUSION IN THE AMBIANT INTELLIGENCE FRAMEWORK

Abstract

The Smart Home concept aims at providing contextualized services to its inhabitants. Based on a heterogeneous sensors domestic network, this new kind of Smart Home deduces the most adapted action to realize with sensory data interpretation. The first problem concerns the strong heterogeneity in the sensor domain: sensors have their own hardware and software features, and their communication standards are poorly standardised. In this thesis, our interest is the context modelling and we propose a service oriented software architecture combining complementary and/or redundant sensors. We use the *Transferable Belief Model (TBM)* to merge sensor data and to take into account the uncertain nature of information. This model is a variant of the Dempster-Shafer theory. The sensors reliability is taken into account during the merging process to weight a failing sensor. A sensor failure can prevent context data building. The sensor reliability is estimated with a pairwise sensor fusion. The temporal conflict analysis allows detection and identification of a failing sensor. We present a second method aiming at detecting temporal behaviour drift. A *TBM* fusion between a predicted symbolic state and the observed symbolic state provided by the sensor is achieved. The predicted symbolic state estimation is based on a known model of behaviour. The temporal conflict analysis allows detecting behaviour drifts. Finally, we present a case study where the previous approaches are implemented in cascade in order to detect a falling person.

Keywords

Smart Home, Context, Ontology, OSGi, Data fusion, Transferable Belief Model, Failure detection.