Contribution à la modélisation de produit actif communicant, Spécification et Evaluation d'un protocole de communication orienté sécurité des produits

Ahmed Zouinkhi

▶ **To cite this version:**

Ahmed Zouinkhi. Contribution à la modélisation de produit actif communicant, Spécification et Evaluation d'un protocole de communication orienté sécurité des produits. Informatique [cs]. Université Henri Poincaré - Nancy I, 2011. Français. NNT : . tel-00594402

HAL Id: tel-00594402
https://theses.hal.science/tel-00594402

Submitted on 19 May 2011

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.

École Doctorale SIS, Université de Gabès

U.F.R. Sciences et Techniques Mathématiques, Informatique et Automatique
École Doctorale IAEM Lorraine
Département de Formation Doctorale Automatique

THÈSE

Présentée pour l'obtention du titre de

Docteur de l'Université Henri Poincaré, Nancy I

Spécialité Automatique, Traitement du Signal et Génie Informatique

Et

Docteur de l'École Nationale d'Ingénieurs de Gabès

Spécialité Génie Electrique

Par

Ahmed ZOUINKHI

CONTRIBUTION À LA MODÉLISATION DE PRODUIT ACTIF COMMUNICANT, SPÉCIFICATION ET ÉVALUATION D'UN PROTOCOLE DE COMMUNICATION ORIENTÉ SÉCURITÉ DES PRODUITS

Soutenue le 7 Avril 2011, devant le jury composé de :

M. Ridha BEN ABDENNOUR	**Professeur à l'Université de Gabès**	**Président**
M. Mohamed CHTOUROU	**Professeur à l'Université de Sfax**	**Rapporteur**
M. David ANDREU	**Maître de Conférences à l'Université Montpellier 2**	**Rapporteur**
M. Eddy BAJIC	**Professeur à l'UHP, Nancy**	**Membre**
M. Eric RONDEAU	**Professeur à l'UHP, Nancy**	**Directeur de Thèse**
M. Mohamed Naceur ABDELKRIM	**Professeur à l'Université de Gabès**	**Directeur de Thèse**

À la Mémoire de mon père

À la Mémoire de ma mère

Je dédie cette thèse

Remerciements

Les travaux présentés dans cette thèse ont été effectués en cotutelle au sein de l'Unité de Recherche *Modélisation, Analyse et Commande des Systèmes*, *MACS* de l'Ecole Nationale d'Ingénieurs de Gabès, Tunisie, sous la direction de Monsieur Mohamed Naceur ABDELKRIM et au *Centre de Recherche en Automatique de Nancy*, *CRAN*, France, sous la direction de Monsieur Eric RONDEAU. Avant de présenter ces travaux, je tiens à remercier tous ceux et celles qui ont participé à l'élaboration et à la réussite de ma thèse.

Je tiens à remercier particulièrement mes directeurs de thèse Monsieur Eric RONDEAU, Professeur à l'université Henri Poincaré, Nancy et Monsieur Mohamed Naceur ABDELKRIM, Professeur à l'Ecole Nationale d'Ingénieurs de Gabès pour leur aide inestimable, leur patience et leurs encouragements tout au long de ce travail. Leurs compétences ont été un atout indéniable à la réussite de ces travaux.

Mes remerciements les plus sincères vont à mon codirecteur de thèse Monsieur Eddy BAJIC, Professeur à l'université Henri Poincaré, pour ses directives scientifiques, pédagogiques et même personnelles pour les quelles je lui suis hautement redevable. Ses compétences ont été un atout indéniable à la réussite de ces travaux. De même, je lui suis extrêmement reconnaissant pour son soutien humain et moral et son aide précieuse durant ces années de thèse.

J'adresse mes plus vifs remerciements à Monsieur Mohamed CHTOUROU, professeur à l'Ecole Nationale d'Ingénieurs de Sfax et à Monsieur David ANDREU, Maitre de conférences à l'Université Montpellier 2, pour m'avoir fait l'honneur d'étudier mes travaux de thèse et de les avoir cautionnés en qualités de rapporteurs.

Je remercie également Monsieur Ridha BEN ABDENNOUR, Professeur à l'Ecole Nationale d'Ingénieurs de Gabès de m'avoir fait l'honneur de présider mon jury de thèse.

Je remercie l'ensemble des membres du *CRAN* pour les fructueuses discutions que j'ai pu avoir avec eux, pour leurs conseils et leur soutien.

Ces années de thèse se sont enrichies de fructueux échanges avec mes collègues de l'unité dans une ambiance propice à la récréation autant qu'à la réflexion scientifique. J'exprime ma profonde gratitude envers eux pour leur sympathie et l'ambiance cordiale qu'ils ont su faire régner au sein de l'équipe.

Table des matières

Introduction Générale **1**

CHAPITRE I :

Etat de l'art sur l'intelligence ambiante : concepts, objets et communication des systèmes ambiants

I. Introduction 6
II. Etat de l'art sur l'intelligence ambiante 6
II.1. Introduction 6
II.2. Caractéristiques de l'Intelligence ambiante 6
II.3. Ubiquitous Computing 8
II.4. Pervasive computing 12
II.5. Les systèmes distribués 14
II.5.1. Problématiques de distribution dans les systèmes distribués 14
II.5.2. Les principales caractéristiques d'un système distribué 15
II.6. Conclusion 16
III. Communication pour les systèmes ambiants 16
III.1. Réseaux de capteurs sans fils 16
III.1.1. Architecture d'un nœud de capteur 18
III.1.2. Architecture de communication 19
IV. Objets communicants 26
IV.1. Concept d'objet communicant 26
IV.2. Le concept de Produit Intelligent 28
V. Les projets de recherche majeurs dans les domaines de l'intelligence ambiante 31
V.1. OXYGEN 32
V.2. MediaCup 32
V.3. AITPL 33
V.4. MemoClip 34
V.5. Smart-Its 34
V.6. DigiClip 35
V.7. eSeal 35
V.8. Wisden 36
V.9. WASP 36
V.10. Cobis 37
V.11. Applications Urbaines 37
V.12. Applications robotiques 40
V.13. Applications Navales 40
VI. Problématique du sujet de recherche 41
VII. Sécurité active pour la gestion des produits chimiques dangereux 44
VIII. Conclusion 46

CHAPITRE II :

Spécification du produit actif communicant et de ses services associés

I. Introduction 48
I.1. Concept de Produit actif 48
II. Problèmes de la gestion de la sécurité 49
II.1. Mécanismes d'interaction 49
III. Le modèle interne de produit actif 50
III.1. Modèle fonctionnel de produit actif 50
III.1.1. Modèle de Strohbach 51
III.1.2. Modèle de Quanz 52
III.1.3. Proposition de modèle de produit actif 53
III.2. Règles de sécurité 56
III.2.1. Règles statiques 57
III.2.2. Règles dynamiques 59
III.2.3. Règles communautaires 60
III.2.4. Calcul du niveau de sécurité global 62
IV. Comportement du produit actif 63
IV.1. La structure des paquets transmis 64
IV.2. Les messages échangés 65
IV.3. Annoncement du produit dans la communauté 66
IV.4. Configuration 67
IV.5. Surveillance et Communication 70
IV.6. Surveillance interne 74
IV.6.1. Le scénario de déclenchement d'une alerte d'une règle d'interaction 75
IV.6.2. Le scénario de déclenchement d'une alerte d'une règle interne 76
IV.6.3. Le Scénario de fonctionnement complet 77
IV.7. Le diagramme d'état/transition : 78
V. Conclusion 80

CHAPITRE III :

Modélisation par réseaux de Petri de la coopération des produits actifs

I. Modélisation par réseau de Petri 82
I.1. Introduction 82
I.2. Le formalisme des Réseaux de Petri colorés 83
I.2.1. Définitions de base 83
I.2.2. Réseau de Petri colorés 84
I.3. La Hiérarchie dans les réseaux de Petri 89
II. Outil de modélisation CPN Tools 90
II.1. Introduction CPN Tools : 90
II.2. Couleurs supportées par CPN Tools 91
II.2.1. Integer 91
II.2.2. Enumerated 91

II.2.3. Product 91
II.3. Les Fonctions 92
III. Modélisation par RdP 94
III.1. Modèle de coopération entre produits actifs 94
III.2. Niveau réseau 95
III.2.1. Réseau sans perturbation 95
III.2.2. Réseau avec perturbations 96
III.3. Niveau produit actif 97
III.3.1. Modèle de produit actif 98
III.4. Niveau fonction de dépendance de produit actif 99
III.4.1. Inscription (annoncement) 99
III.4.2. Configuration 100
III.5. Niveau fonction autonome de produit actif 101
III.5.1. Modèle de la surveillance et communication par réseau de Petri 101
III.5.2. Modèle de la surveillance interne 106
III.6. Modèle de gestionnaire 109
IV. Résultats et scénarios 111
IV.1. Règle statique 111
IV.2. Règle dynamique 112
IV.3. Déclenchement d'une alerte dûe à la proximité 113
V. Conclusion 115

CHAPITRE IV :

Simulation des coopérations entre produits actifs

I. Introduction 117
II. Les simulateurs existants 117
II.1. NS2 117
II.2. OPNET 118
II.3. OMNeT++ 118
III. Le simulateur Castalia 119
III.1. Le module radio : 121
III.2. Le module MAC 121
III.3. Le module réseau : 122
III.4. Le module Processus physique : 122
III.5. Module du gestionnaire du dispositif de perception (Sensing device manager) ... 123
III.6. Le module du gestionnaire des ressources : 123
III.7. Le module application 123
III.8. Le module mobilité 123
III.9. Les projets effectués utilisant Castalia : 124
IV. Le diagramme de classe 125
IV.1. Le premier groupe de classes : 125
IV.2. Le deuxième groupe de classes : 126
IV.3. Le troisième groupe de classes : 127

V. Les résultats de simulation 130
V.1. L'environnement de simulation 130
V.2. Les paramètres utilisés dans la simulation 130
V.3. Réglage des paramètres de scrutation 131
V.4. Plans d'expériences 132
V.4.1. Introduction 133
V.4.2. Contexte d'utilisation 133
V.4.3. Objectif 133
V.4.4. Préparation du plan d'expériences 134
V.4.5. Définition de l'objectif de l'étude 134
V.4.6. Description des éléments sur lesquels va porter l'expérimentation 134
V.4.7. Expérimentation 137
V.5. Etude des cas 152
V.5.1. Les scénarios de configuration 152
V.5.2. Scénario 2 155
V.5.3. Scénario 3 156
V.5.4. Scénario 4 158
V.6. Intervalle de confiance 159
V.7. Etude énergétique 160
V.8. Influence du nombre de produits sur la probabilité de perte des paquets 162
VI. Conclusion 162

Conclusion Générale 163

Bibliographie 165

Table des figures

Figure I. 1. Technologies utilisant l'intelligence 7
Figure I. 2. Domaines d'appui de l'Intelligence Ambiante 8
Figure I. 3. Evolution chronologique de la relation (utilisateur, PC) 9
Figure I. 4. Réalité virtuelle vs. Ubiquitous Computing 10
Figure I. 5. Interactions d'éléments clés selon le paradigme ubiquitous computing 12
Figure I. 6. Taxinomie des systèmes informatiques 13
Figure I. 7. Anatomie générale d'un nœud de capteur 19
Figure I. 8. Schéma d'un réseau de capteurs 20
Figure I. 9. Pile protocolaire des réseaux de capteurs 20
Figure I. 10. Apport de l'information du produit dans le processus décisionnel 31
Figure I. 11. MediaCup en communication avec une montre intelligente 32
Figure I. 12. Le Memo Clip avec son dispositif de localisation 34
Figure I. 13. Plateforme Smart-Its 35
Figure I. 14. Le DigiClip 35
Figure I. 15. Conteneur équipé par une particule intelligente 37
Figure I. 16. Estimation de collision par WSN 38
Figure I. 17. Application WSN embarquée pour les véhicules 39
Figure I. 18. Localisation par WSN 40
Figure I. 19. Système de sécurité active 41
Figure I. 20. Diagramme en pieuvre APTE d'un produit actif communicant 43
Figure I. 21. Pictogrammes réglementaires des différents types de dangers 44
Figure I. 22. Matrice d'incompatibilité du catalogue Merck 46

Figure II. 1. Les capacités liées à un Produit Actif 49
Figure II. 2. Deux approches existantes dans le système de gestion de la sécurité proposé 50
Figure II. 3. Modèle du produit selon [Strohbach et al., 2005] 51
Figure II. 4. Illustration de la coopération selon [Strohbach et al., 2005] 52
Figure II. 5. Modèle du produit selon [Quanz and Tsatsoulis, 2008] 52
Figure II. 6. Illustration de la coopération selon [Quanz and Tsatsoulis, 2008] 53
Figure II. 7. Modèle fonctionnel de produit actif 54
Figure II. 8. Comportement autonome d'un produit actif 56
Figure II. 9. Caractéristiques d'un produit actif 57
Figure II. 10. Fonction de la règle statique de température 57
Figure II. 11. Règles statiques 59
Figure II. 12. Illustration des règles dynamiques 60
Figure II. 13. Règles de communauté 62
Figure II. 14. Les états d'un Produit Actif 63
Figure II. 15. La structure de l'entête du paquet générique. 64
Figure II. 16. La structure du paquet générique 65
Figure II. 17. La structure du message CTR 66
Figure II. 18. La structure du message AckCTR 66

Figure II. 19. Diagramme de séquence de l'annoncement d'un produit Actif 67
Figure II. 20. Structure du paquet CMD1 68
Figure II. 21. Structure du paquet CMD3 68
Figure II. 22. Diagramme de séquence de demande configuration et des règles de sécurité .. 69
Figure II. 23. Scénario de demande de configuration 69
Figure II. 24. Scénario de demande de règles de sécurité 70
Figure II. 25. Structure du paquet GRE 71
Figure II. 26. Equivalence RSSI – Distance 71
Figure II. 27. Structure du message RSI 71
Figure II. 28. Scénario de salutation entre deux produits 72
Figure II. 29. Structure du message INA 72
Figure II. 30. Scénario de demande de lecture des valeurs ambiantes 72
Figure II. 31. Structure du paquet CFG 73
Figure II. 32. Scénario de lecture des paramètres de configuration 73
Figure II. 33. Scénario de demande lecture des règles de sécurité 74
Figure II. 34. Structure du paquet ALER 74
Figure II. 35. Scénario de déclenchement d'une alerte de règle d'interaction 75
Figure II. 36. Structure du paquet ALEV 75
Figure II. 37. Scénario de déclenchement d'une alerte de règle interne 76
Figure II. 38. Scénario de fonctionnement complet du réseau 77
Figure II. 39. Diagramme d'états/transitions du modèle interne de produit actif 79

Figure III. 1. Exemple d'un RdP coloré 86
Figure III. 2. Exemple d'un RdP avec couleur composée 87
Figure III. 3. Exemple d'un RdP coloré temporisé 88
Figure III. 4. Réseau de Petri coloré Hiérarchique 90
Figure III. 5. Modèle de coopération entre produits actifs 94
Figure III. 6. Modèle de réseau sans perturbation 95
Figure III. 7. Modèle de réseau présentant une perturbation 96
Figure III. 8. Ligne de transmission avec perte 97
Figure III. 9. Structure du modèle Hiérarchique du produit actif 98
Figure III. 10. Modèle de produit actif 98
Figure III. 11. Modèle inscription de produit actif 99
Figure III. 12. Modèle configuration de produit actif 100
Figure III. 13. Modèle surveillance et communication de produit actif 102
Figure III. 14. Modèle analyse et traitement de messages 103
Figure III. 15. Modèle traitement de la base de connaissance 104
Figure III. 16. Modèle envoi messages 105
Figure III. 17. Règle dynamique 107
Figure III. 18. Modèle surveillance interne de produit actif 108
Figure III. 19. Modèle évaluation des informations capteurs 108
Figure III. 20. Modèle décision de produit actif 109
Figure III. 21. Modèle de gestionnaire 110
Figure III. 22. Scénario illustrant une alerte de la règle statique 111

Figure III. 23. Scénario illustrant une alerte de la règle dynamique 112
Figure III. 24. Scénario illustrant une distance défavorable entre deux produits actifs 113
Figure III. 25. Modèle hiérarchique de la communauté de produits actifs 114

Figure IV. 1. Structure d'un nœud sous Castalia. 120
Figure IV. 2. Code du message CMD1 127
Figure IV. 3. Extrait du code de l'application: traitement du message CMD1 128
Figure IV. 4. Diagramme de classe de l'application implémenté sous Castalia 128
Figure IV. 5. Exemple de code de modèle implémenté sous castalia 129
Figure IV. 6. Disposition des produits actifs dans un entrepôt 131
Figure IV. 7. Représentation du plan d'expérimentation 136
Figure IV. 8. Valeur de la réponse aux points du domaine d'étude. 138
Figure IV. 9. Le coefficient pour la réponse au centre du domaine d'étude. 138
Figure IV. 10. Le coefficient du facteur 1. 140
Figure IV. 11. Illustration de l'effet du facteur 1. 141
Figure IV. 12. Illustration de l'effet de T_{GRE}. 142
Figure IV. 13. Illustration d'une interaction entre deux facteurs. 143
Figure IV. 14. Illustration de l'interaction entre T_{cap} et T_{GRE} 144
Figure IV. 15. Illustration de l'erreur relative en fonction de T_{cap} et T_{GRE}. 146
Figure IV. 16. L'objectif de la réactivité du système. 146
Figure IV. 17. Le coefficient du facteur 1 (Autonomie du système). 147
Figure IV. 18. Illustration de l'effet du facteur 1 (Autonomie du système) 148
Figure IV. 19. Illustration de l'effet de T_{GRE} (Autonomie du système). 148
Figure IV. 20. Illustration de l'interaction entre T_{cap} et T_{GRE} 150
Figure IV. 21. Illustration de l'énergie consommée en fonction de T_{cap} et T_{GRE}. 151
Figure IV. 22. L'objectif de l'autonomie en fonction de T_{cap} et T_{GRE} 151
Figure IV. 23. Les objectifs en fonction de T_{cap} et T_{GRE} 152
Figure IV. 24. Scénario de configuration du PA 1 (aucune configuration installée) 153
Figure IV. 25. Scénario de configuration du PA 1 (Les paramètres installés) 154
Figure IV. 26. Scénario de configuration du PA 1 (Les règles installées) 155
Figure IV. 27. Illustration d'alerte du PA 3 156
Figure IV. 28. Illustration d'un cas d'alerte du PA 3 156
Figure IV. 29. Reconfiguration des produits 3 et 4 157
Figure IV. 30. Illustration de simulation d'envoi d'un message ALERTE 157
Figure IV. 31. Evolution de la distance entre les produits 3 et 4 158
Figure IV. 32. Energie consommée par produit actif 160
Figure IV. 33. Influence du nombre des produits sur la probabilité de perte des paquets 162

Liste des tableaux

Tableau 1. Exemples des protocoles de la couche réseau 23
Tableau 2. Fonctions de services 43
Tableau 3. Base de connaissance du produit actif 55
Tableau 4. Ensemble de messages échangés requêtes/réponses 65
Tableau 5. Facteurs et domaine d'étude 135
Tableau 6. Matrice d'expérimentation. 136
Tableau 7. Matrice d'expériences. 136
Tableau 8. Matrice d'expériences et résultats. 137
Tableau 9. Récapitulation des temps de configuration des PAs 154
Tableau 10. Energie consommée par produit actif en fonction de services 161

Introduction Générale

Les entreprises de production de biens doivent répondre à des impératifs de plus en plus exigeants de qualité, de disponibilité, de réactivité, de traçabilité et de sécurité des produits, et offrir une interaction croissante avec leurs clients.

Dans le cadre des réseaux logistiques, et plus généralement des flux de la chaîne logistique, tels que le transport et le stockage, les systèmes opérants et les produits mobiles sont contraints par des réglementations de sécurité des biens et des personnes, et de respect de l'environnement de plus en plus sévères et exigeants aux niveaux national, européen et mondial. Cela est flagrant dans l'industrie chimique, où le stockage et le transport de produits dangereux sont régis par des directives Européennes de plus en plus sévères (1999/45/EC et 67/548/EEC). Néanmoins, ces réglementations ne réussissent pas à garantir une sécurité certaine des gens et des biens qui évoluent dans un espace de travail hostile notamment pour ceux qui manipulent des substances chimiques dangereuses. Les limitations rencontrées sont majoritairement de nature humaine et proviennent du fait que les procédures classiques de sécurité sont lourdes à gérer par les responsables de la sécurité du personnel qui sont appelés à suivre et à contrôler constamment les détails et les changements pouvant survenir dans leur espace d'activité.

Les points capitaux en matière de sécurité concernent principalement l'identification précise des substances chimiques manipulées, les conditions ambiantes de stockage et de manipulation des produits, les interactions et les compatibilités chimiques entre produits proches.

D'autre part nous observons dans le domaine industriel comme dans le domaine de la recherche scientifique, que selon une approche conceptuelle moderne, le produit tend à devenir un bien intégrant des services ou des fonctionnalités permettant d'optimiser son utilisation, son interopérabilité avec son environnement (usager, système de production, de distribution, autres produits, …), sa traçabilité.

Ces nouveaux services attendus sur les produits nécessitent de disposer d'un continuum informationnel tout au long du cycle de vie du produit. Les NTIC (Nouvelles Technologies de l'Information et de la Communication) maintenant disponibles (étiquettes électroniques RFID HF/UHF, EPCGlobal, Réseaux sans Fils HF/ …), permettent de faire supporter au produit lui-même les informations, voire les traitements qui le concernent pour répondre aux attentes précédemment définies. Le produit peut alors être acteur, et contrôler en partie son usage, sa maintenance, son stockage, sa sécurité, son transport, …

Le produit est transformé en un « objet communicant » avec son environnement système et humain ainsi qu'avec d'autres produits, et il est capable de supporter des « services associés » et de s'insérer intelligemment dans des structures collectives par sa « mobilité ».

Ces notions caractérisent le concept de « système à intelligence ambiante » que nous souhaitons développer dans notre thèse pour prendre en compte la gestion de la sécurité des biens et des personnes dans le domaine du stockage et du transport des produits dangereux dans les industries chimiques.

Dans le paradigme « Produit Communicant », il s'agit de conférer au produit un rôle actif et participatif dans les décisions et les flux d'informations engendrés dans un système de production ou système logistique, selon des objectifs de transformation, de déplacement, de maintenance, de stockage, de transport, d'usage et de recyclage … La communauté scientifique a vu apparaître le terme de « produit intelligent » caractérisant les nouvelles compétences données au produit (« intelligent product », « smart product »), que nous préfèrerons qualifier de « Produit Actif » dans notre discours.

Un produit n'est plus un simple matériau circulant dans un entrepôt, mais un objet capable de réaliser des échanges et des interactions intelligentes avec des systèmes de décision et de pilotage. Ces fonctions de communication et d'interactions avec le produit sont rendues disponibles par la technologie de réseaux de capteurs sans fils (Wireless Sensors Network) annonçant une forme d'intelligence technique permettant d'aller au delà du transfert de données vers le partage et l'échange d'information, la connaissance contextuelle, les services,....

Dans ce contexte, le concept de produit actif permet d'aborder la problématique de la surveillance et de sécurité des produits sous un angle novateur, par un transfert des capacités et responsabilité de surveillance et gestion de la sécurité depuis le système (approche centralisée) vers le produit lui-même (approche atomisée).

Le concept de produit actif consiste à doter un produit de capacités à communiquer, informer, acquérir, décider et réagir aux stimuli et perturbations de son environnement afin de permettre au produit de s'adapter, d'influer, de coopérer, de transformer le comportement de son environnement. L'objectif est de rendre le produit acteur intelligent et proactif dans son environnement afin de gérer et contrôler son niveau de sécurité de façon active autonome et en relation avec les autres produits actifs et son environnement.

L'activité de recherche proposée a pour objectif de formaliser, valider et implémenter le concept de produit actif pour la gestion de sécurité active entre des produits industriels conditionnés, à haute toxicité ou dangerosité (type fûts industriels chimiques) transformés en

Produits Actifs. Toute modification de l'environnement du produit actif enfreignant les règles de sécurité individuelle ou mutuelle doit être détectée, diagnostiquée et doit engendrer des actions externes permettant de recouvrir par des actionnements ou une information à destination de l'environnement ambiant sur le niveau de sécurité actuel des produits.

Le produit actif doit ainsi s'intégrer dans un système d'interactions avec les autres produits et avec son environnement, dans lequel il s'agit d'assurer la sécurité active des produits, des personnes et de l'environnement lui même.

La validation de ce concept nécessite l'instrumentation d'un produit par des systèmes de communication sans fils (de la RFID aux capteurs sans fils WSN, « smart object »), des capteurs et détecteurs locaux (perception environnement T°, luminosité, son, Mouvement-choc, …), des actionneurs et interface de visualisation ou sonores pour la communication humaine et une logique informatique. Ainsi un Produit actif dit à Intelligence Ambiante peut être capable de « sentir » son environnement, de décider et faire un choix d'action / réaction selon des spécifications propres et/ou partagées dans un environnement coopératif, de communiquer avec son environnement.

Notre sujet de recherche recouvre deux communautés scientifiques : la communauté « intelligence ambiante (AI) » portant le concept pour analyser le comportement intelligent des produits, et la communauté « réseaux de capteurs sans fil (WSN) » pour le transport des données et la communication ambiante inter produits.

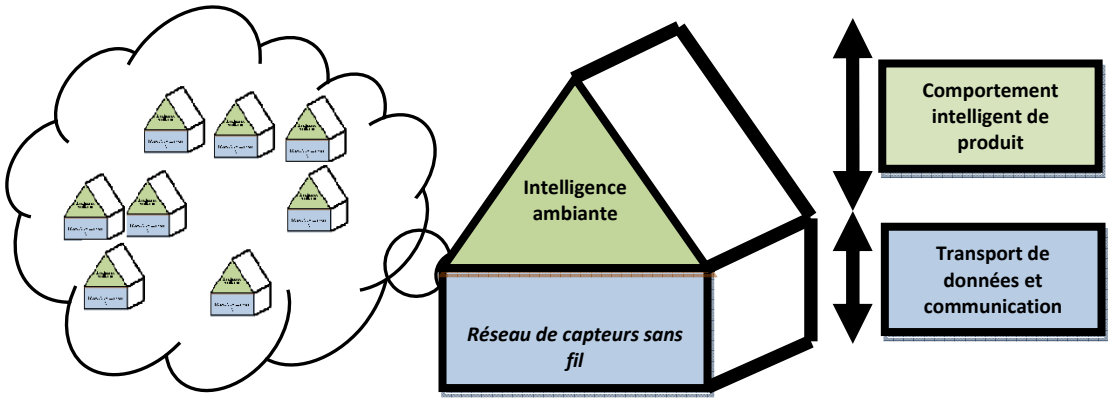

Figure 1. Positionnement du sujet de recherche

Les objectifs du sujet de recherche consistent à analyser, définir, formaliser et mettre en place des mécanismes d'intelligence ambiante dans les environnements de stockage et de logistique industrielle impliquant la gestion et la manipulation de produits dangereux tel que dans le

domaine de la chimie, afin d'améliorer les interactions entre les produits et les acteurs du système de manière à la sécurité active des biens et des personnes.

Le mémoire de thèse est organisé en quatre chapitres suivis d'une conclusion générale. Le positionnement de nos travaux est présenté sur le premier chapitre et nos contributions sont détaillées dans les trois derniers chapitres.

Dans le premier chapitre, nous présentons l'état de l'art sur les notions d'intelligence ambiante en abordant les concepts, les objets et la communication des systèmes ambiants, de façon à expliciter la problématique du sujet de thèse.

Le deuxième chapitre détaille le concept de produit actif et explore les modèles existants référents dans la communauté scientifique. Nous proposons une évolution de ces modèles vers un modèle de produit actif communicant orienté pour la gestion de la sécurité en définissant les éléments de la communication entre produits.

Le troisième chapitre est destiné à formaliser le concept de coopération cité dans le chapitre deux tout en spécifiant le modèle proposé du produit actif et la coopération au sein de la communauté de produits par une modélisation en Réseaux de Petri afin de garantir l'aspect qualitatif.

Dans le quatrième chapitre, nous implémentons les modèles du produit actif dans l'outil de simulation CASTALIA. Le réglage optimal des paramètres de configuration de notre protocole est ensuite réalisé en utilisant les plans d'expériences. Les simulations sous CASTALIA nous permettent d'évaluer nos propositions en montrant ses intérêts et ses limites.

Le mémoire de thèse se termine par une conclusion générale dans laquelle nous synthétisons les apports de nos travaux et nous en dégageons les perspectives.

CHAPITRE I

Etat de l'art sur l'intelligence ambiante : concepts, objets et communication des systèmes ambiants

I. Introduction

L'objectif de l'intelligence ambiante est d'élargir l'interaction entre les utilisateurs et la technologie d'information numérique à travers l'usage des dispositifs relatifs à la notion de « l'informatique omniprésente » ou «l'ubiquitous computing » [Ronzani, 2009]. L'informatique traditionnelle emploie des interfaces utilisateurs comme les souris claviers, écrans… alors que l'espace entourant l'utilisateur n'est pas exploité. L'intelligence ambiante, se sert de tout ce que l'entourage peut fournir comme information (son, mouvement, lumière, forme…) et offre indirectement à l'utilisateur une nouvelle forme d'interface d'interaction par usage d'autres dispositifs distribués dans l'étendue de son domaine d'activité. Le mode de communication de ce type d'interface inclut souvent la technologie sans fil. De ce fait, la combinaison de ces dispositifs intégrant des aptitudes à communiquer mène à augmenter leur degré d'ubiquité.

En fait, l'intelligence ambiante est un domaine interdisciplinaire entrelaçant plusieurs paradigmes similaires tels que « ubiquitous computing », « pervasive computing », objet communicant et ses dérivés dont on développera les concepts dans ce chapitre.

II. Etat de l'art sur l'intelligence ambiante

II.1. Introduction

L'intelligence ambiante (AmI) est un concept défini par Mark Weiser qui était responsable de recherche au Xerox Palto Alto Research Center (PARC) en 1998. AmI est une notion assez vague regroupant un ensemble de technologies partageant des traits communs. Les environnements de l'intelligence ambiante sont basés sur l'éclatement des systèmes informatiques dans une multitude d'objets hétérogènes dans leur nature et leur mode de communication [Weiser, 1991].

II.2. Caractéristiques de l'Intelligence ambiante

Le principe de l'intelligence ambiante est de capturer l'information présente dans l'environnement afin de proposer à un utilisateur des services personnalisés en fonction de sa situation et de son contexte. L'intelligence ambiante est un enjeu majeur des futurs systèmes de télécommunication comme le montre la figure I.1.

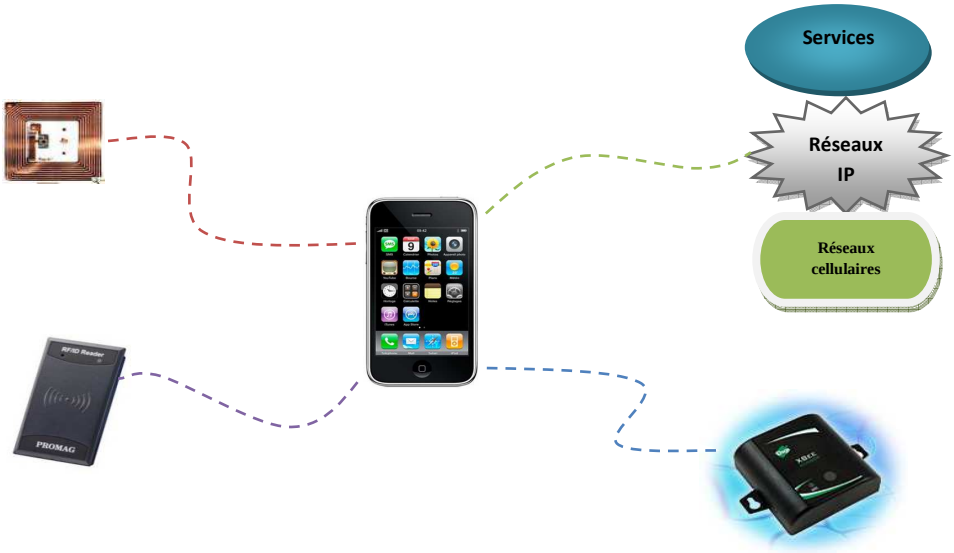

Figure I. 1. Technologies utilisant l'intelligence

L'intelligence ambiante (AmI) met en œuvre quatre éléments de base [Weiser, 1991]:

- ✓ **L'ubiquité :** la capacité pour l'utilisateur d'interagir, n'importe où, avec une multitude d'appareils interconnectés, de capteurs, d'actionneurs et plus globalement avec les systèmes électroniques "enfouis" (Embedded software) autour de lui. Tout cela se fait à travers de réseaux adaptés et d'une architecture informatique très distribuée.

- ✓ **L'attentivité :** la faculté du système à "sentir" en permanence la présence et la localisation des objets, des appareils et des personnes pour prendre en compte le contexte d'usage. Toutes sortes de capteurs sont nécessaires à cette fin : caméras, micros, radars, ainsi que la technologie des puces et lecteurs à radiofréquence (RFID) pour l'identification.

- ✓ **L'interaction naturelle :** l'accès aux services doit pouvoir se faire de la façon la plus naturelle/intuitive possible. Elle s'articule autour de la reconnaissance vocale, gestuelle ou la manipulation d'objets réels.

- ✓ **L'intelligence :** c'est la faculté d'analyse du contexte et l'adaptation dynamique aux situations. Le système doit apprendre en se basant sur les comportements des utilisateurs afin de leur répondre au mieux. Cela implique des capacités de stockage, de traitement et des algorithmes de modélisation.

L'Intelligence ambiante se développe rapidement en tant que approche multidisciplinaire qui permet à beaucoup de domaines de recherche d'avoir un apport remarquable dans notre société [Augusto and Cook, 2007]. Comme l'indique la figure I.2 cette approche est en relation avec beaucoup de secteurs dans l'informatique, tels que les réseaux, les capteurs, les interfaces homme machine et l'informatique omniprésente mais il reste à noter que l'intelligence ambiante alimente tous ces secteurs sans être confondu avec eux, il s'agit de doter un environnement par une technologie de façon à ce qu'il soit capable de prendre des décisions basées sur des informations recueillies en temps réel et des données historiques accumulées.

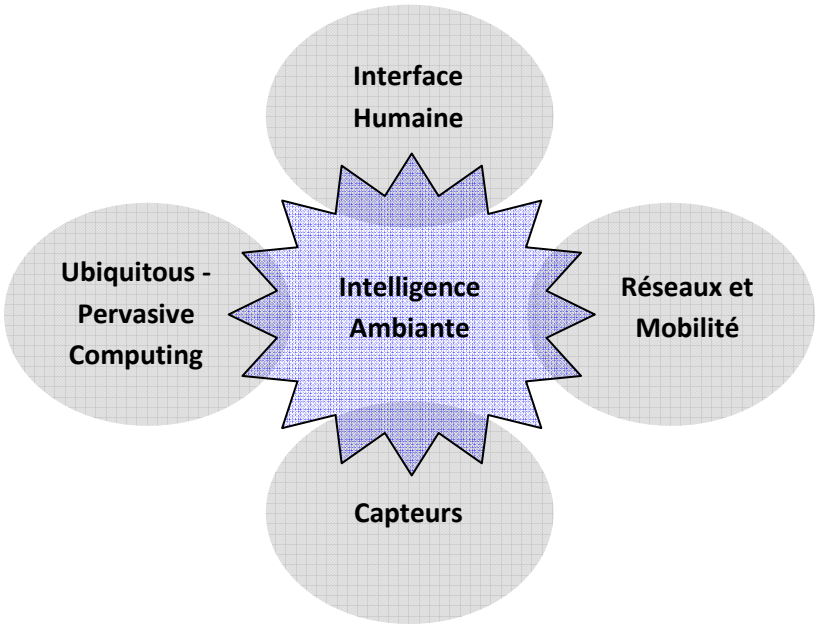

Figure I. 2. Domaines d'appui de l'Intelligence Ambiante

Dans la suite du chapitre nous analysons l'état de l'art des domaines d'appui de l'AmI qui sont : Ubiquitous computing, Pervasive computing et Systèmes distribués pour analyser ensuite le concept d'objet communicant. Nous détaillerons pour chaque domaine les projets de recherche majeurs.

II.3. Ubiquitous Computing

Le concept d'informatique Ubiquiste ou omniprésente résumé par le terme anglais « ubiquitous computing » peut être défini comme le modèle qui suit l'ordinateur de bureau englobant des mécanismes d'interaction homme-machine dans lequel le traitement de l'information a été complètement intégré dans tous les objets des activités journalières. Ceci est explicité par [Wieser, 1993a] en ces termes : "*Ubiquitous computing has as its goal the*

enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the users".

Cette approche est à considérer par opposition au paradigme de bureau, dans lequel un seul utilisateur engage consciemment un dispositif unique dans un but spécialisé.

L'idée de ce paradigme a été inventée par Mark Wieser qui était responsable de recherche au Xerox Palto Alto Research Center (PARC). Le programme de recherche « ubiquitous computing » a débuté vers la fin des années quatre-vingt au laboratoire EIL Electronique et Image de Xerox [Wieser *et al.*, 1999] où l'objectif visait à redéfinir la relation entre les humains, l'environnement de travail et de la technologie informatique par une ère post ordinateur individuel type PC. On peut expliciter ceci en suivant l'évolution chronologique concernant la machine informatique et l'utilisateur, comme l'illustre la figure I.3 : à l'époque il existait la relation centralisée « ordinateur central (mainframe) – plusieurs utilisateurs », puis survint l'ère de l'informatique personnelle « un ordinateur – un utilisateur » puis la troisième ère informatique qui repose sur l'approche distribuée « un utilisateur – plusieurs ordinateurs »

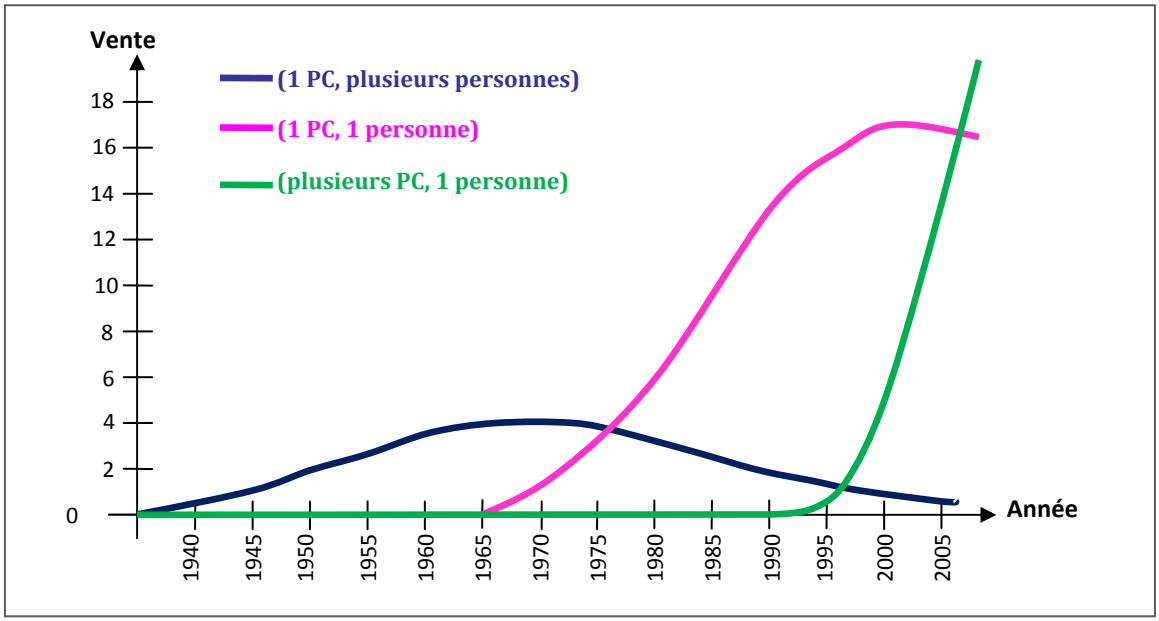

Figure I. 3. Evolution chronologique de la relation (utilisateur, PC) [Ronzani, 2009]

Plus récemment, [Alcaniz and Rey, 2005] caractérisent l'informatique ubiquiste ainsi, montrant la modernité et le côté visionnaire de Marc Weiser, encore vérifiés à ce jour : *"Ubiquitous computing can be defined as the use of computers everywhere. Computers are made available by means of the physical environment, but in an invisible way for the user."*

Une autre description du paradigme d'informatique ubiquiste utilisée dans la communauté informatique le présente en tant qu'approche opposée de « la réalité virtuelle »[1] [Schmidt, 2002]. Cette dernière embarque l'utilisateur dans un monde virtuel généré par les outils informatiques (exemples de ce monde virtuel : simulateurs, jeux vidéo…). L'ubiquitous computing de son côté force les dispositifs informatiques à se répartir autour de l'environnement de l'utilisateur. Cet aspect est illustré dans la figure I.4.

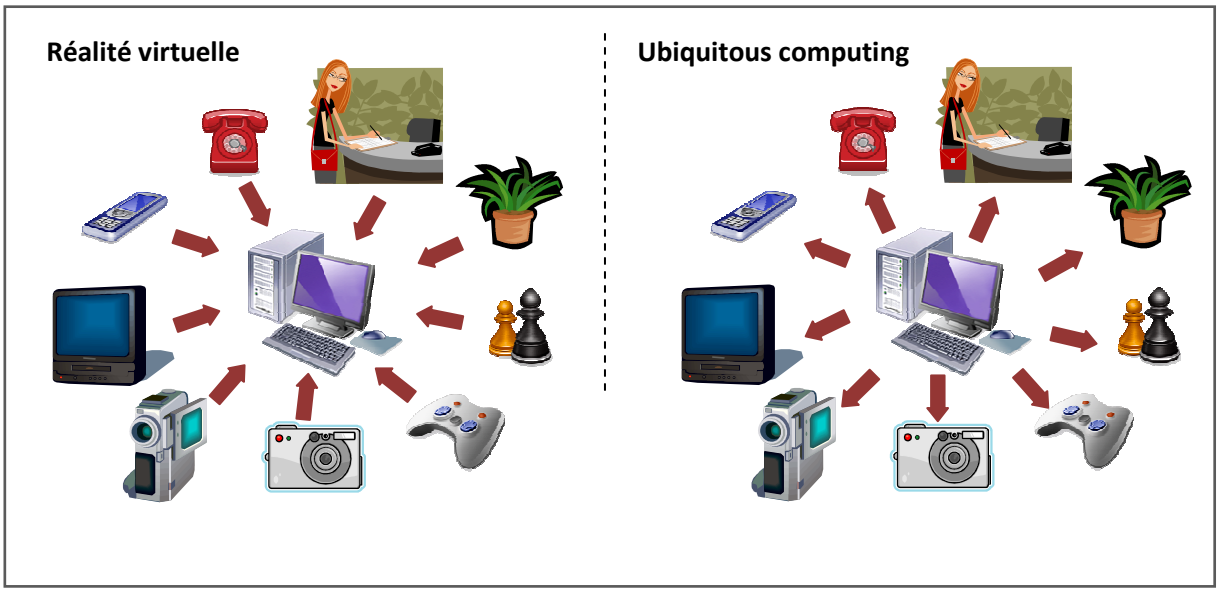

Figure I. 4. Réalité virtuelle vs. Ubiquitous Computing

Dans le domaine de l'intelligence ambiante, ce paradigme se manifeste éventuellement par l'emploi d'outils coopératifs distribués permettant à l'utilisateur d'accéder à différents services. La dissociation de ce paradigme de l'informatique traditionnelle (aspect bureau) vers l'informatique ubiquiste repose sur deux approches [Alcaniz and Rey, 2005] : à savoir fournir à l'utilisateur des dispositifs mobiles et répartir des outils informatiques dans l'environnement.

'Ubiquitous Computing is fundamentally characterized by the connection of things in the world with computation' [Weiser and Brown, 1996]

Selon [Mattern and Sturm, 2003], les tendances et les développements qui permettent de concevoir un monde informatisé tendant vers l'ubiquité sont :

[1] www.ubicomp.com

- Des composants microélectroniques miniaturisés, performants et peu coûteux.
- De nouveaux matériels d'interface homme-machine capables d'afficher et de présenter l'information (LCD, PDA, …)
- De multiples capteurs sans fil consommant moins d'énergie.
- La possibilité d'identification et de localisation des objets permettant d'entrer en relation avec eux.
- Des réseaux de communication filaire et sans fil plus robustes.

De ce fait, le concept ubiquitous computing dépend largement de l'utilisation d'outils électroniques enrichis par des capteurs, actionneurs et d'éléments informatiques intégrés, capables d'échanger de l'information via un réseau. Wieser souligne que la qualité de ces outils embarqués est fortement dépendante de leur aptitude à se dissimuler dans l'environnement *'A good tool is an invisible tool. By invisible, I mean that the tool does not intrude on your consciousness; you focus on the task, not the tool (…) Of course, tools are not invisible in themselves, but as part of a context of use. '* [Wieser, 1993b]. Ceci lui procure un caractère important dans le monde informatique à savoir l'invisibilité (« … *not invisible in themselves, but as part of a context of use").*

Reste que, le domaine d'ubiquitous computing est encore en plein développement et il affronte quelques problématiques d'ordre scientifique tel que le maintien simultané de la simplicité et du contrôle des dispositifs à développer ; aussi bien que d'ordre éthique, puisque l'on s'immisce à terme dans la vie privée des usagers. Autrement dit, avec la présence de ces dispositifs disséminés, les flux de données s'interférent, ce qui rend difficile le contrôle du système global, dans les termes qui collecte l'information, qui l'exploite,…etc [Langheinrich, 2002], [Moncrieff et al., 2007].

Le paradigme «ubiquitous computing» a pour but de fournir des environnements informatisés dépendant du contexte de l'utilisateur pendant que celui-ci se déplace d'un endroit à un autre. La Figure I.5 illustre la symbolisation faite par [Yoo and Kalle, 2005] du concept d'ubiquitous computing. Selon ces auteurs, l'interaction dynamique, continue et simultanée entre :

- la technologie, permettant de faire interagir des ressources (des objets, des applications,…) afin d'offrir des services à un utilisateur ;
- le contexte de l'utilisateur (description de sa situation physique et sociale) ;
- et la mobilité temporelle et spatiale (entre des contextes) d'un utilisateur ;

définit la nature ubiquiste du paradigme ubiquitous computing. La complexité de ce scénario d'interaction est augmentée si l'on considère qu'un utilisateur peut avoir plus d'un rôle à la fois pendant son déplacement physique. A l'heure actuelle, l'application du concept «ubiquitous computing» est seulement partielle. Nous croyons qu'il faut attendre quelques années encore avant de voir l'application entière du concept proprement dit dans les différents domaines de la vie actuelle. Compte tenu de ce fait, l'industrie a adopté le concept «pervasive computing» pour aller de façon plus pragmatique vers une vision technologique invisible centrée sur l'utilisateur [Mattern, 2005].

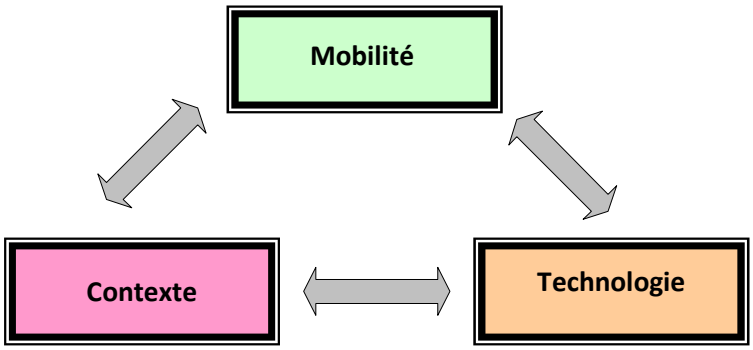

Figure I. 5. Interactions d'éléments clés selon le paradigme ubiquitous computing

II.4. Pervasive computing

L'appellation « pervasive computing » ou « informatique diffuse » représente un paradigme orienté technologie qui s'appuie non plus uniquement sur des PCs mais sur des composants électroniques accessibles au quotidien embarquant des moyens de traitement et de communication de plus en plus petits et de plus en plus puissants [Saha and Mukherjee, 2003]. L'objectif est de pouvoir offrir un panel d'information sur son environnement qui soit toujours accessible et cela de façon transparente. En fait, dans la littérature les termes « pervasive computing » et « ubiquitous computing » sont généralement employés comme synonymes et pourtant ils présentent une nuance conceptuelle. L'informatique diffuse repose sur les aspects de la transparence, de la disponibilité en tout lieu et du couplage entre le monde réel et le monde informationnel. [Agoston *et al.*, 2000] l'exprime comme suit : « l'informatique pervasive rend l'information disponible partout et à tout moment ».

Dans le fait, le terme « pervasive computing » a été introduit par IBM en 1998 pour la création au sein de la société du département « pervasive computing group ».

[Cea, 2006] explique que le concept ubiquitous computing est plus étendu que le pervasive computing car il doit obligatoirement intégrer la notion de mobilité. [Singh et *al.*, 2005] considèrent que le paradigme informatique ubiquiste profite des avantages de « mobile computing » et de « pervasive computing » pour présenter un environnement global informatique. Comme son nom l'indique, le « mobile computing » est basé sur des services intégrés dans des dispositifs mobiles qui utilisent une infrastructure sans fil pour s'affranchir des problèmes de localisation et d'infrastructure. Toutefois, ce paradigme reste simpliste car il n'évoque pas les problèmes liés à l'acquisition des données relativement à leur contexte et à leur évolution dans le temps et dans l'espace. Ces aspects sont pris en compte par contre dans le concept de « pervasive computing » qui doit offrir des informations sélectives en fonction des besoins et des profils des utilisateurs. La figure I.6 illustre une classification des différents paradigmes présentés ci-dessus mettant en évidence les relations qui les associent [Xiang, 2002]. Notons ici que le système distribué est un système englobant plusieurs ordinateurs qui communiquent à travers un réseau dans le but de coordonner le traitement et les actions d'une application [Mattern and Sturm, 2003].

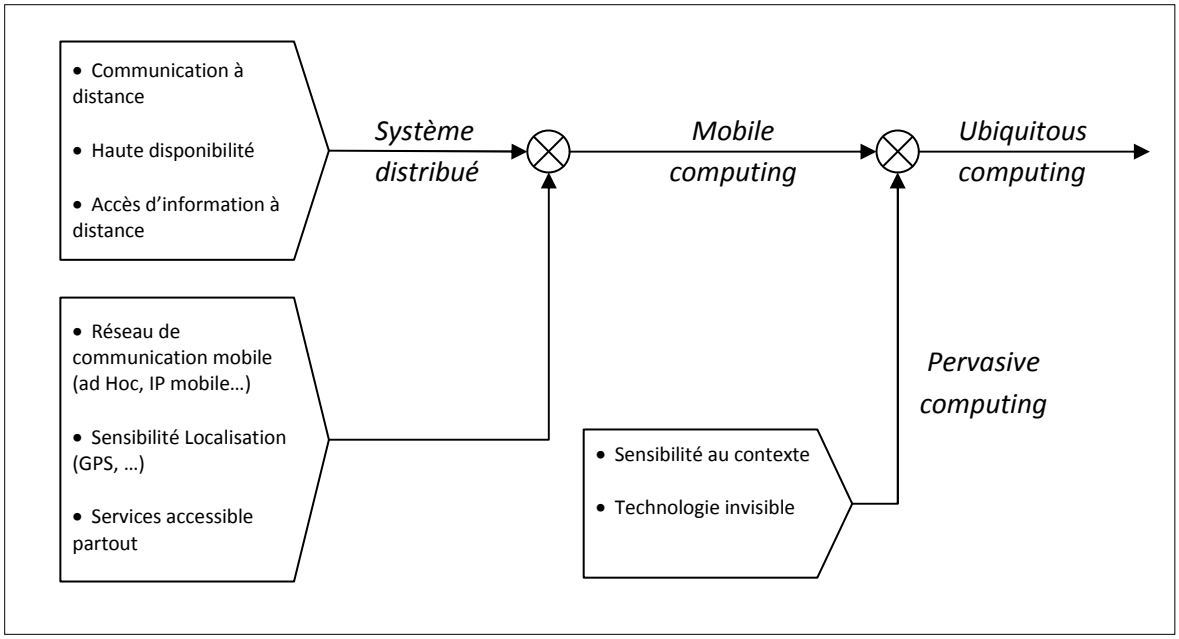

Figure I. 6. Taxinomie des systèmes informatiques

Ainsi nous constatons que les terminologies associées au domaine de l'Intelligence Ambiante sont nombreuses, variées, complexes avec parfois un fort taux de recouvrement, ce qui peut nuire dans une certaine mesure la compréhension et la visibilité du concept défendu.

II.5. Les systèmes distribués

[Thiare, 2007] définit le système distribué comme suit : un système distribué correspond à un ensemble de processus ou nœuds qui s'exécutent sur des sites reliés par un réseau de communication et qui communiquent par envoi de messages. Sur chaque site, il est possible de définir un état local, qui est modifié par l'exécution des processus du site. Les messages, dépendant des contraintes présentes dans le réseau de communication, doivent dans un temps fini mais arbitraire atteindre les processus qui les reçoivent. [Drira, 2005] ajoute à la définition précédente l'aspect coopératif des systèmes distribués qui sont des systèmes distribués multi-utilisateurs dans lesquels il y a un fort degré d'interaction. Les interactions sont liées à la distribution des différents composants du système ainsi qu'à la distribution des utilisateurs. Elles sont aussi liées à la nature coopérative de ces systèmes qui supposent d'une part une forte interactivité pour la consultation et la production de données ou d'informations et d'autre part l'échange et le partage de ces données par les différents utilisateurs.

II.5.1. Problématiques de distribution dans les systèmes distribués

Dans le domaine informatique, un système distribué [Tel, 2001] aussi connu sous le nom de système réparti ou en réseau, est un ensemble d'unités de traitement autonomes interconnectées entre-elles. Ces unités de traitement, aussi appelées processeurs, coopèrent via des échanges d'information pouvant être sous forme de messages dans le but de réaliser une tâche commune [Devisme, 2006]. Les informations transitent via les liens de communications bidirectionnels ou canaux puisque les processeurs peuvent également émettre et recevoir des messages.

A travers ce caractère coopératif des systèmes distribués, certaines contraintes émergent et concernent le manque de connaissance globale sur le système [Devisme, 2006]. En effet les hypothèses les plus faibles, les seules informations que possède une unité sur le réseau se résument à la distinction de ses canaux. L'incertitude temporelle liée au temps de transmission des messages est un autre problème des systèmes coopératifs.
La problématique majeure des systèmes distribués est donc de décrire formellement les mécanismes permettant de résoudre une tâche dont les données sont réparties dans le réseau tout en tenant compte de ces contraintes.

L'étude des systèmes distribués coopératifs soulève des problèmes relatifs aux contraintes et exigences d'ordre comportemental visant les caractéristiques fonctionnelles algorithmiques ou événementielles, mais aussi des problèmes relatifs aux contraintes et exigences d'ordre

structurel concernant les caractéristiques architecturales ou topologiques [Drira, 2005]. Pour les aspects comportementaux, il s'agit par exemple de problèmes algorithmiques liés à la causalité ou à l'ordre des communications de groupe. Il peut s'agir aussi de problèmes fonctionnels relatifs à la nature de l'activité supportée par les dits systèmes (gestion de session, gestion du partage, gestion du workflow). Pour les aspects structurels, il s'agit par exemple de problèmes liés à la gestion dynamique d'une architecture pour optimiser la distribution des composants ou pour s'adapter à la mobilité des utilisateurs ou aux changements dans les groupes ou l'espace de coopération. On peut distinguer notamment la gestion du cycle de vie des composants, leur distribution et déploiement ainsi que la gestion des canaux de diffusion qu'ils utilisent. Il s'agit, dans l'ensemble de ces cas, de problèmes de coordination liés aux interdépendances entres les différents composants des systèmes, les différents utilisateurs de ces systèmes et des actions qu'ils produisent.

II.5.2. Les principales caractéristiques d'un système distribué

Les principales caractéristiques d'un système distribué selon [Devisme, 2006] sont les suivantes:

- **L'échange d'informations**

Une des motivations premières des systèmes distribués est de permettre les échanges de données notamment à grande échelle. L'exemple le plus marquant étant le réseau *Internet* qui permet à des milliards d'individus de communiquer et d'échanger de grandes masses de données.

- **La distribution des données**

Mettre en réseau des machines permet d'obtenir un espace disque conséquent pour un coût raisonnable. Les SAN (Storage Area Network) [Sandeep et al., 2007] [Taisir et al., 2006] se généralisent dans les entreprises et permettent de stocker des quantités d'information très importantes toutes en les rendant facilement accessibles.

Enfin, le fait de pouvoir distribuer sur un réseau les données a permis de faire évoluer les techniques de sauvegarde de données avec l'apparition de méthodes de duplications de données sur le réseau. Par exemple, les méthodes *RAID (Redundant Arrays for Inexpensive Disks)* [David et *al.*, 1988] permettent de reconstituer les informations perdues suite à la panne définitive d'une machine grâce aux duplicata répartis sur le réseau.

- **L'augmentation de la puissance de calcul**

Les systèmes distribués permettent la réalisation de calculs concurrents. Lorsque le temps d'exécution d'un calcul sur une machine monoprocesseur est important, le temps d'exécution de ce même calcul sur un système distribué sera généralement réduit de manière significative. Récemment, des travaux sur les réseaux dédiés au calcul intensif (calcul *out of core*) [Caron, 2000] ont été développés comme par exemple les grilles de calcul (grid computing). Une grille de calcul exploite la puissance de calcul (processeurs, mémoires, ...) de milliers d'ordinateurs interconnectés entre eux en donnant l'illusion d'un ordinateur virtuel très puissant.

- **La Tolérance aux fautes**

Une faute correspond à une défaillance temporaire ou définitive d'un composant du système distribué. Contrairement à un système centralisé, dans un système distribué, une partie des services peut continuer à fonctionner suite à une panne. Cependant, la multiplicité des composants d'un système accroît le risque que l'un d'entre eux tombe en panne. Ces défaillances influent sur le comportement des traitements (envoi et réception de messages et calcul interne). Donc, en cas de fautes, la validité des résultats calculés par ces traitements ne peut pas toujours être assurée.

II.6. Conclusion

Dans cette partie, nous avons présenté et analysé les différents paradigmes supportant le concept d'intelligence ambiante en présentant les avantages qu'ils procurent vis-à-vis des systèmes centralisés, ceci afin de définir le comportement intelligent de produit. Par la suite, nous allons nous intéresser au transport de données, à la communication entre produits, ainsi qu'à l'application de ce concept dans le domaine de la logistique des produits dangereux, afin d'apporter une solution à la problématique de la gestion de la sécurité des produits chimiques dangereux.

III. Communication pour les systèmes ambiants

III.1. Réseaux de capteurs sans fils

L'implantation d'un système distribué là où il y a collecte et échange continuel d'information nécessite un réseau de capture réparti sur les différents points de mesure dans un espace de travail délimité. L'architecture de réseau de capteurs classique repose sur des chemins de

câbles parfois complexes, par conséquent non rentable du point de vue encombrement et coût. De ce fait, suite à l'essor des technologies sans fil, de nouvelles solutions émergentes viennent remplacer cette architecture classique par l'emploi des capteurs capables de récupérer et de communiquer des données sans fil, que l'on nomme Réseaux de Capteurs Sans Fils (RCSF) ou WSN (Wireless Sensor Network). C'est une technologie en plein essor et qui a de très nombreux débouchés.

En fait, ces derniers sont constitués de centaines de milliers de capteurs équipés d'interface de calcul et de communication qui sont capables de se coordonner entre eux. Un réseau de capteurs sans fil possède plusieurs intérêts : il surmonte le problème d'encombrement filaire, et il est extensible.

Un réseau de capteurs peut être aussi vu comme une extension de réseau de communication « ad hoc » où les informations communiquées sont générées par le réseau lui-même [Hoyle, 2005].

Bluetooth et Zigbee se retrouvent parmi les technologies les plus aptes à être exploitées dans les réseaux de capteurs sans fil [Fumolari, 2001]. La technologie Bluetooth a pour but principal de remplacer les câbles sur de petites distances. Malheureusement, le grand défaut de cette technologie est sa trop grande consommation d'énergie. Elle ne peut donc être raisonnablement utilisée par des capteurs qui sont alimentés par une batterie et qui, idéalement, devraient fonctionner durant plusieurs années. Le standard Zigbee offre des caractéristiques qui répondent mieux aux besoins des réseaux de capteurs, en offrant des débits de communication moindres, mais en consommant également nettement moins que Bluetooth. Cependant, cela risque de nuire à son efficacité de transmission de flux de données dans un réseau où la fréquence de transmission est continuelle. Le système de communication RFID est une autre technologie basée sur l'identification automatique par communication radio fréquence. Son intérêt vis-à-vis des réseaux de capteurs sans fils est son aptitude à stocker et à récupérer des données à distance par usage d'étiquettes électroniques appelées Tags, implantées sur les objets.

Selon [Meier et al., 2008], les applications des réseaux de capteurs sont de deux types : des applications de détection des événements « *event detection applications* », et des applications de collecte des données « *gathering applications* ». Alors que pour [Akkaya and Younis, 2005], les réseaux de capteurs sont classés en considérant différents facteurs d'architecture comme le modèle de délivrance des données et la mobilité dans un réseau.

III.1.1. Architecture d'un nœud de capteur

Un nœud capteur est composé de plusieurs éléments ou modules correspondant chacun à une tâche particulière d'acquisition, de traitement, ou de transmission de données, comme l'illustre la Figure I.7. Il comprend également une source d'énergie [Akyildiz et al., 2002b] [David et al., 2004].

L'unité d'acquisition des données : le principe de fonctionnement des détecteurs est souvent le même : il s'agit de répondre à une variation des conditions d'environnement par une variation de certaines caractéristiques électriques (par exemple pour une thermistance, une variation de température entraîne une variation de la résistance). Les variations de tension sont ensuite converties par un convertisseur Analogique-Numérique afin de pouvoir être traitées par l'unité de traitement.

On trouve aussi des structures plus complexes pour détecter d'autres phénomènes : les MEMS (pour MicroElectroMechanical Systems) [Akyildiz et al., 2002a] [Akyildiz et al., 2002b]. Ils sont utilisés pour une grande variété de phénomènes physiques (accélération, concentration chimique...).

L'unité de traitement des données : les microcontrôleurs utilisés dans le cadre de réseaux de capteurs sont à faible consommation d'énergie. Leurs fréquences sont assez faibles, moins de 10 MHz pour une consommation de l'ordre de 1 mW. Une autre caractéristique est la taille de leur mémoire qui est de l'ordre de 10 Ko de RAM pour les données et de 10 Ko de ROM pour les programmes [Karl and Willig, 2005] [Akyildiz et al., 2002b]. Cette mémoire consomme la majeure partie de l'énergie allouée au microcontrôleur, c'est pourquoi on lui adjoint souvent de la mémoire flash moins coûteuse en énergie. Outre le traitement des données, le microcontrôleur commande également toutes les autres unités notamment le système de transmission.

L'unité de transmission de données : les composants utilisés pour réaliser la transmission sont des composants classiques. Ainsi on retrouve les mêmes problèmes que dans tous les réseaux sans fil : la quantité d'énergie nécessaire à la transmission augmente avec la distance. Pour les réseaux sans fil classiques (LAN, GSM) la consommation d'énergie est de l'ordre de plusieurs centaines de milliwatts, et on se repose sur une infrastructure alors que pour les réseaux de capteurs, le système de transmission consomme environ 20 mW et possède une portée de quelques dizaines de mètres. Pour augmenter ces distances tout en préservant l'énergie, le réseau utilise un routage multi-sauts [Wei et al., 2004].

La source d'énergie : pour des réseaux de capteurs sans fil autonomes, l'alimentation est une composante cruciale. Il y a essentiellement deux aspects : premièrement, stocker l'énergie et la fournir sous la forme requise ; deuxièmement, tenter de reconstituer l'énergie consommée par un réapprovisionnement grâce à une source externe au nœud capteur telles les cellules solaires. Le stockage de l'énergie se fait traditionnellement en utilisant ses piles. À titre indicatif, ce sera souvent une pile AA normale d'environ 2.5 mAh fonctionnant sous 1.5 V [Karl and Willig, 2005].

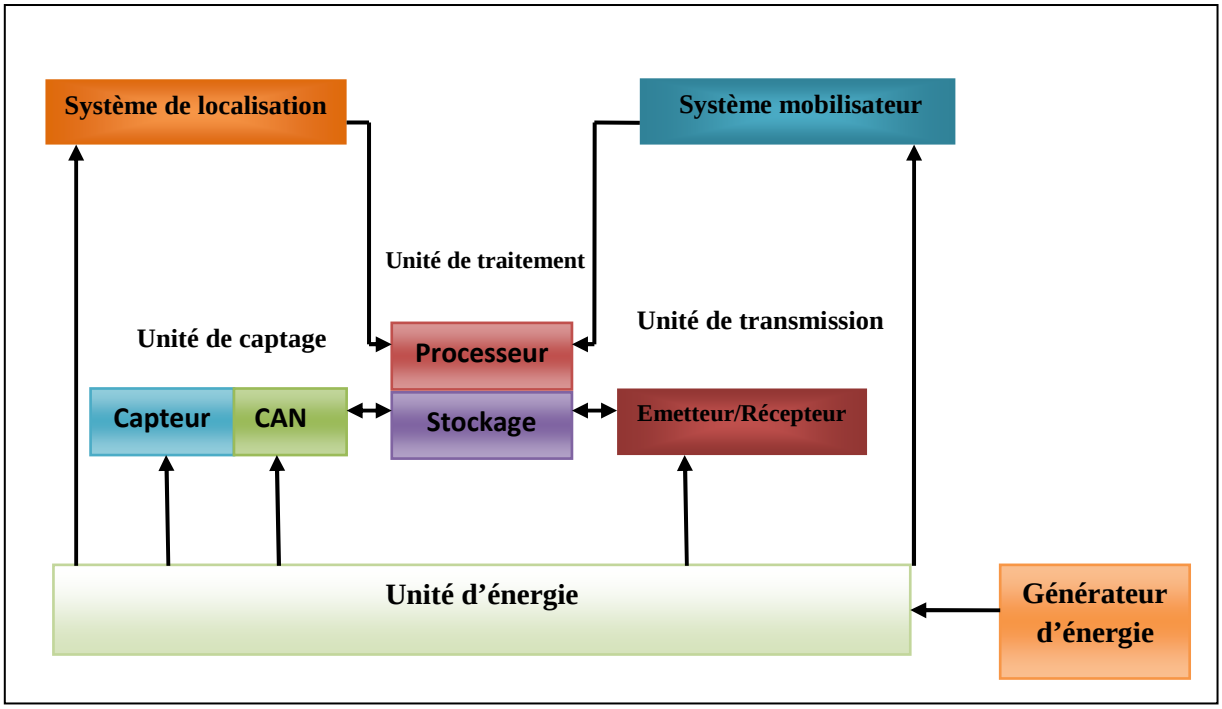

Figure I. 7. Anatomie générale d'un nœud de capteur [Akyildiz et al., 2002a]

III.1.2. Architecture de communication

Dans la grande majorité des domaines, le rôle d'un réseau de capteurs est relativement identique, comme présenté dans la figure I.8. Les nœuds doivent surveiller certains phénomènes grâce à leurs capteurs puis envoient les informations à un puits. Le puits est un nœud particulier doté d'une puissance de calcul supérieure et d'une quantité d'énergie potentiellement infinie. Ce puits peut être connecté à Internet ou possède un lien radio de type GSM ou GPRS qui lui permet d'envoyer les informations (données ou alertes) à un centre de contrôle pour l'utilisateur final. Il peut y avoir plusieurs puits mobiles ou fixes dans un réseau. Mais pour des raisons de coût, il y a beaucoup moins de puits que de nœuds [Samper, 2008].

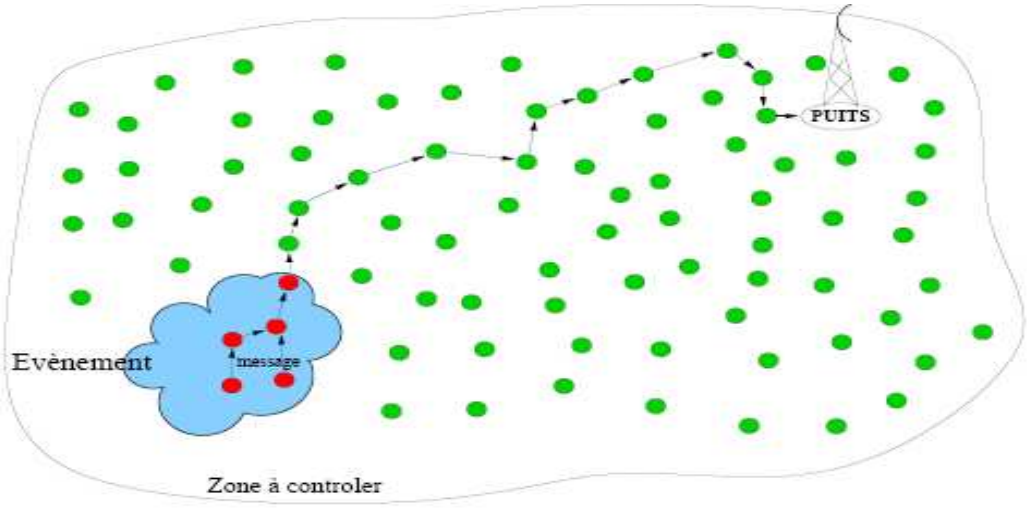

Figure I. 8. Schéma d'un réseau de capteurs [Samper, 08]

La figure I.9 représente un modèle de communication adapté au réseau de capteurs sans fil où l'on trouve une contraction de la traditionnelle pile OSI en 5 couches (couche physique, couche liaison, couche réseau, couche transport, couche Application), avec l'adjonction de trois plans de niveau gestion de l'architecture par la puissance, la mobilité et les tâches.

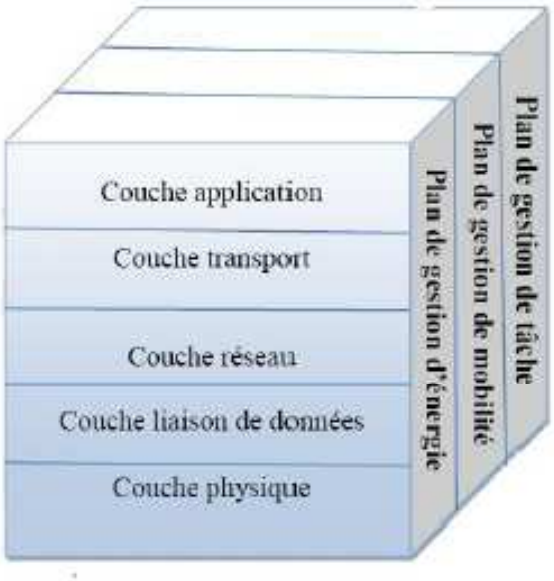

Figure I. 9. Pile protocolaire des réseaux de capteurs [Akyildiz et al., 2002a]

Les cinq couches fonctionnent de la manière suivante [Akkaya and Younis, 2005]:

III.1.2.1. La couche application

Elle permet de rendre transparent les mécanismes de communication dans les couches inférieures, en offrant des interfaces pour la création et la diffusion de requêtes. Selon les activités surveillées, différents types d'applications peuvent être construites et utilisées dans la couche application. Malgré le nombre important des applications proposées pour les réseaux de capteurs, les protocoles de cette couche présentent une région encore peu explorée. On peut examiner trois types de protocoles possibles pour la couche application :

- Le Protocole de gestion des capteurs (SMP : Sensor Management Protocol) [Kulkarni and Pao, 2005] : Il permet la transparence des logiciels et des composants physiques des couches basses vis-à-vis des applications de gestion des réseaux de capteurs. Les administrateurs interagissent avec le réseau de capteurs à travers SMP.
- Le protocole d'affectation des tâches et d'avertissement des données (TADAP : Task Assignment and Data Advertisement Protocol) [Chen et al., 2008]. Les utilisateurs peuvent dialoguer avec un nœud, un sous ensemble de nœuds ou tout le réseau.
- Le protocole de dissémination des données (SQDDP : Sensor Query and Data Dissemination Protocol) [Zhou et al., 2006].

III.1.2.2. La couche transport

Cette couche aide à transporter le flux de données à la couche application si elle en a besoin. La couche transport aide à gérer le flux de données si le réseau de capteurs l'exige. Elle permet de segmenter les données issues de la couche application. A la réception elle réordonne et rassemble les segments venus de la couche réseau avant de les envoyer à la couche application. On utilise le protocole UDP dans la communication entre le nœud collecteur appelé nœud puits (ou sink) et les nœuds de capteurs, parce que chaque nœud de capteur a une mémoire limitée [Pang et al., 2008].

III.1.2.3. La couche réseau

La couche réseau a pour rôle l'acheminement des données fournies par la couche transport. Cette couche dans les réseaux de capteurs est toujours structurée de façon à répondre aux principes suivants:

- Consommation efficace de l'énergie a toujours une considération importante ;
- L'agrégation des données n'est utilisée que lorsqu'elle ne demande pas un effort collaboratif des nœuds de capteurs ;

- Un réseau de capteurs idéal a un adressage basé sur des attributs et ses nœuds sont conscients de leur localisation. Dans un réseau de capteurs l'information est décrite en utilisant des attributs. En effet, l'utilisateur est plus intéressé par l'information fournie par le réseau que par le nœud donnant cette information. Ainsi, il interroge le réseau en utilisant les attributs du phénomène observé.

Cette couche fournit aussi la possibilité de communiquer avec d'autres réseaux de capteurs.

- **Le routage :**

Les protocoles de routage spécifiques aux réseaux de capteurs doivent tenir compte du type de communication induit par l'application. Outre le fait que la quantité de données échangées est très faible par rapport aux applications de types réseaux « ad-hoc », le trafic est particulièrement prévisible puisqu'il va des nœuds vers le puits ou du puits vers les nœuds [Samper, 2008].

Parmi les recherche sur le routage dans les réseaux de capteurs, [Medjiah et al., 2009] proposent un protocole de routage géographique AGEM (Adaptive Greedy-Compas Energy-Aware Multipath Routing Protocol) basé sur le protocole GPSR (Greedy Perimeter Stateless Routing) pour supporter la transmission de flux multimédia dans les WMSNs (Wireless Multimedia Sensor Networks). Le protocole AGEM a deux modes opératoires, un routage glouton intelligent et un routage en marche arrière. Le premier mode est utilisé quand il y a toujours un voisin plus proche de la destination que le nœud actuel, alors que le second est utilisé pour contourner les trous. Pour évaluer les performances du protocole AGEM, OMNeT++ 4.0 a été utilisé comme simulateur de réseaux à événements discrets. Pour prouver l'efficacité de leur protocole, ils ont implémenté également l'algorithme GPSR et ont comparé les résultats des simulations.

Le tableau 1 présente quelques protocoles de routages implémentés dans des réseaux de capteurs.

Les cas du routage multi-chemins pour la transmission de flux vidéo dans les réseaux de capteurs sans fil sont examinés dans [Maimour, 2007]. Les auteurs proposent SLIM (par Simple Lifetime-based Multipath), un protocole de routage multi-chemin conçu pour les couches de transport vidéo sur les réseaux de capteurs de ressources limitées. Un routage multi-chemin est aussi une option intéressante pour répartir spatialement la dépense d'énergie sur un plus grand nombre de noeuds [Maimour, 2007] [Kim et al., 2008].

Protocole de couche réseau	Description
SMECN [Li and Halpern, 2001]	Crée un sous graphe du réseau de capteurs contenant le chemin donnant le minimum de consommation d'énergie
SPIN [Heinzelman et al., 1999]	N'envoie les données aux nœuds que lorsqu'ils sont intéressés. Il a trois types de messages : ADV, REQ et DATA
LEACH [Heinzelman et al., 2000]	Forme des groupements minimisant la dissipation d'énergie.
Directed diffusion [Intanagonwiwat et al., 2000]	Initialise des gradients pour la donnée pour flow de la source vers le puits durant une phase de dissémination.

Tableau 1. Exemples des protocoles de la couche réseau

III.1.2.4. La couche liaison

La couche liaison est responsable du multiplexage des flux des données, du contrôle d'accès au media (effectué par le protocole MAC [Ghosh et al., 2009] [Demirkol et al., 2006]) et du contrôle des erreurs (CRC, …).

[Kuntz and Noël, 2009] se sont intéressés à la problématique de la mobilité dans les RCSFs et à ce que cela implique au niveau de la couche MAC (Medium Access Control) en examinant notamment les problèmes de synchronisation et de congestion qui peuvent survenir. Ils ont exposé les solutions existantes les plus significatives et ont proposé le protocole « Machiavel ». À la différence d'un protocole par échantillonnage classique, Machiavel permet au nœud mobile d'envoyer ses données en étant assuré que ses voisins ont été correctement synchronisés. En réduisant les délais d'accès au médium, Machiavel permet également au nœud de ne pas saturer sa file d'attente. Une évaluation du protocole a permis de démontrer ses bénéfices, notamment en réseaux denses où les pertes du mobile sont significativement réduites

III.1.2.5. La couche physique

La couche physique est responsable du support acheminant les données envoyées entre les nœuds. Ainsi, il existe deux types de médias pouvant être utilisés pour les réseaux de capteurs : les ondes infrarouges et les ondes radiofréquences [Aboelaze and Aloul**,** 2005].

En outre, selon des plans de gestion de l'énergie [Cartron and Sentieys, 2005], et de la mobilité [Al-Obaisat and Braun, 2006], les tâches surveillent respectivement la puissance, le mouvement et la distribution des tâches [Zhou et al., 2006] entre les nœuds capteurs. Ces plans de gestion sont nécessaires, de sorte que les nœuds capteurs puissent fonctionner ensemble d'une manière efficace pour préserver l'énergie destinée à router les données dans un réseau de capteurs mobile et pour partager les ressources entre les nœuds capteurs. Du point de vue global, il est plus efficace d'utiliser des nœuds capteurs pouvant collaborer entre eux. La durée de vie du réseau peut être ainsi prolongée.

- **Problématique de la consommation d'énergie**

Une caractéristique majeure que doit prendre en considération le réseau concerne la gestion de la consommation d'énergie afin d'optimiser sa durée de fonctionnement [Demers et al., 2003]. Cette gestion est encore qualifiée d' « auto organisation» et d' « auto configuration » ne possédant pas une administration centrale. Il répond dynamiquement à la défaillance des nœuds importants du réseau [Englund and Wallin, 2004]. Tout ceci vise à rendre le réseau de capteur aussi autonome que possible, ce qui est très important dans les situations de surveillance automatique ou dans les environnements hostiles.

[Buhrig and Renaudin, 2007] ont proposé une méthode pour gérer la consommation d'énergie dans un contexte de réseau de capteurs. Cette méthode est basée sur l'utilisation de contraintes temps réel pour minimiser la vitesse du processeur et sa consommation d'énergie. Un algorithme d'ordonnancement a été implémenté et permet de réduire la consommation d'énergie du processeur.

[Makkaoui et al., 2010] a étudié les performances d'une chaîne de compression de type JPEG qui intègre une DCT zonale rapide dans le système de transmission d'image dans les réseaux de capteurs sans fil. Cette DCT zonale rapide réduit le nombre de coefficients à calculer, et donc à quantifier et à encoder. Elle entraîne une réduction de la complexité de calcul de la chaîne de compression, et par incidence une réduction de la consommation d'énergie sur le système. Elle est particulièrement intéressante dans le contexte des réseaux de capteurs sans fil où le problème de la consommation d'énergie est dominant.

- **La sécurisation dans les réseaux de capteurs**

Parmi les domaines de recherche ayant pour thème la sécurisation des réseaux de capteurs, nous trouvons le domaine du cryptage des données. [Castelluccia et al., 2005] ont proposé une solution pour la sécurisation des données naviguant sur le réseau. Il s'agit de combiner des

techniques simples de cryptage avec des méthodes simples d'agrégation pour une agrégation efficace des données cryptées. Ils ont proposé une nouvelle méthode de cryptage permettant à des capteurs intermédiaires dits « agrégateurs » d'agréger les données cryptées vers les autres nœuds.

Un autre domaine ayant pris une grande importance dans la communauté de recherche concerne la sécurisation de l'environnement surveillé par les capteurs. [Samper, 2008], dans ses travaux de thèse effectués dans le cadre d'un contrat entre l'entreprise France Télécom R&D et le laboratoire Verimag, a étudié le comportement d'un réseau de capteurs destiné à la surveillance de l'environnement.

III.1.2.6. Les systèmes d'exploitation s'interfaçant avec les réseaux de capteurs

Dans la communauté scientifique, plusieurs projets tel que Smart Dust project[2] s'intéressent aux domaines de réseaux de capteurs et ont publié plusieurs technologies d'implémentation. Nous pouvons citer à titre d'exemple le système TinyOS[3] qui est un système d'exploitation open source conçu pour des réseaux de capteurs sans fil développé et maintenu par l'université de Berkeley. D'autre part, le système TinyDB [Madden et al., 2003] est un système de traitement destiné à extraire les informations depuis un réseau de capteurs TinyOS. Le Régiment [Newton and Welsh, 2004] est un langage de macro programmation fonctionnel basé sur des services qui représentent les états des capteurs spatialement distribués et variables dans le temps.

III.1.2.7. Les standards des réseaux des capteurs sans fil

- **Le standard 802.15.4**

Le 802.15.4[4] est un protocole de communication défini par l'IEEE. Il est destiné aux réseaux sans fil de la famille des LR-WPAN (Low Rate Wireless Personal Area Network) du fait de leur faible consommation, de leur faible portée et du faible débit. Les dispositifs utilisant ce protocole 802.15.4 correspond aux couches basses de Zigbee, à savoir de la couche physique à la couche MAC. [Vaudour and Gauthier, 2006] ont précisé certaines caractéristiques dont on peut citer l'utilisation de CSMA/CA et sa faible consommation d'énergie.

[Tomic, 2006] a étudié le protocole MAC IEEE 802.15.4 apportant une analyse des besoins pour que ce protocole assure des mécanismes de partage des ressources entre plusieurs

[2] http://robotics.eecs.berkeley.edu/~pister/SmartDust/
[3] TinyOS Project: TinyOS (2004). http://www.tinyos.net/
[4] www.wikipedia.fr (2003)

applications. Pour cela, il a défini différents scénarios pour examiner les exigences d'une infrastructure partagée.

- **Le standard Zigbee**

Le standard Zigbee[5] offre des caractéristiques qui répondent mieux aux besoins des réseaux de capteurs, en offrant des débits de données plus bas, mais en consommant également nettement moins que Bluetooth.

IV. Objets communicants

IV.1. Concept d'objet communicant

Un objet communicant regroupe l'ensemble des composants de l'Intelligence Ambiante dans une entité physique tangible (matérielle).

Le concept d' « objet communicant » est apparu au milieu des années 90 et tend à aboutir à une interconnexion complète des objets physiques appartenant à l'environnement quotidien [Senn, 2007] [Mattern, 2000].

Un objet communicant peut être défini comme une entité physique capable de percevoir et de communiquer avec son environnement, avec des utilisateurs et avec d'autres objets quelconques interagissant avec lui au moyen de technologies de communication filaire ou sans fil [Beigl *et al.*, 2003b].

Par conséquent, l'objet communicant ne peut accomplir cet objectif que s'il a les capacités suivantes [Cea, 2006]:

- Mémorisation : stockage des informations caractéristiques de l'objet et de son environnement ;
- Perception : surveillance de son environnement physique et informatique dans le but de s'adapter et de gérer son évolution ;
- Communication : recherche et sélection d'information auprès de son environnement ;
- Action : traitement d'information et exécution des mécanismes afin d'accomplir une tâche précise ;
- Décision : analyse de son état, du comportement de son environnement pour prendre de façon autonome et/ou concertée des décisions relatives à des situations particulières.

[5] www.zigbee.org

Ce concept d'objet communicant qui émerge rapidement, n'est pas seulement à la convergence des deux mondes high tech des terminaux et des objets numériques. Il est en fait appelé à se diffuser largement dans celui des produits de base de tous types: électroménager, HiFi, vêtements, véhicules, objets personnels divers. Dans cet ordre d'idée, la mobilité constitue une caractéristique typique de ce paradigme [Schmidt *et al.,* 1998]. Avec le progrès technologique qui propose des composants de traitement et de mémoire de plus en plus puissants et miniaturisés, les objets communicants passent à un degré de mobilité élevé et peuvent même se porter en tant qu'accessoire vestimentaire (*wearable devices* : lunettes, montre…).

Une autre spécification des objets communicants, en plus de leur caractère transportable, est la capacité de communiquer d'une façon autonome. Ceci les projette dans un aspect où le contexte constitue un élément inévitable pour situer leurs interactions avec le monde externe.

Le terme contexte est défini par Anind Dey [Dey, 2000] par ″Toute information caractérisant la situation des entités″ qui sont dans notre cas les objets communicants. [Schilit *et al.,* 1994] le définit comme étant la collection des objets, des usagers à proximité ainsi que les changements survenus. Cette dernière définition montre l'évolution de la notion du contexte en fonction de la situation où l'objet interagit [Schmidt, 2002]. On parle alors du concept « *context awareness* » où l'objet en question collecte les informations utiles auprès de son environnement puis en génère des décisions. Par exemple un téléphone mobile « *contexte aware* » peut discriminer que son utilisateur est assis dans une salle de réunion et non pas dans sa voiture par exemple. L'appareil pourrait ainsi conclure que son utilisateur est en réunion et rejette tout appel téléphonique inintéressant[6].

En conséquence, l'objet communicant possède une aptitude à interpréter l'environnement au travers de capteurs afin de gérer la relation utilisateur/objet/environnement lui permettant d'avoir une vision structurée et unifiée du monde dans lequel il opère.

Les informations contextuelles aidant l'objet communicant à évoluer dans son environnement reposent selon [Schilit *et al.,* 1994] sur trois aspects importants :

- Qui êtes-vous ? : identifier l'objet à proximité.
- Où êtes-vous ? : localiser l'objet.
- Quelles sont les ressources aux alentours ? : identifier les aptitudes et états des autres objets communicants.

[6] www.wikipedia.org/context_awareness

Ceci nous mène vers une autre caractéristique des objets communicants qui est l'aptitude à être identifié par les autres objets intéressés dans le but de configurer automatiquement les interactions supportées par ces derniers [Cea, 2006]. Parmi les technologies populaires qui visent à atteindre cet objectif, il y a la technologie RFID, qui est une méthode d'identification automatique invoquant le stockage et la récupération des informations à distance au moyen des étiquettes électroniques RFID. Reste aussi une autre caractéristique importante de l'objet communicants qui concerne la gestion de sa propre source d'énergie.

Le domaine d'application habituel des objets communicants est la télécommunication [Privat, 2000]. Mais on peut en citer d'autres comme les applications de la santé (maison médicalisée) et le contrôle de l'environnement (situation à risque, contrôle de l'énergie) [Senn, 2007].

En résumé, les objets communicants se sont concrétisés dans les applications industrielles grâce à l'émergence des nouveaux moyens de communication sans fil qui permettent déjà d'offrir de nouveaux services liés à la localisation physique et à l'environnement immédiat. De ce fait, tout objet physique de notre environnement quotidien (bureau, voiture, magasin…) peut être doté de capacités à percevoir, à analyser son environnement et à interagir avec d'autres objets, avec un système d'information local ou global (notamment le WEB) [Mattern and Sturm, 2003] et avec le ou les utilisateurs.

IV.2. Le concept de Produit Intelligent

Le concept de produit intelligent s'apparente à un objet communicant en étant proche des concepts développés dans la communauté scientifique traitant des systèmes multi-agents. Un produit intelligent est un produit quelconque (au sein d'une chaîne de production) doté d'une capacité de perception (capteur), d'une capacité décisionnelle (unité de traitement) et d'une capacité de communication avec les autres produits (unité de transmission). Un produit peut se définir comme un noyau (milieu) qui présente un certain type d'intelligence générée par l'interaction Produit-Milieu et Produit-Produit. On parle alors d'intelligence ambiante.

Selon Norman [Norman, 1998], l'intégration de ces produits (noyaux) n'est pas vue seulement pour collecter les informations et pour communiquer mais comme un élément naturel dans notre quotidien.

Gershenfeld affirme que la barrière entre le monde numérique et notre monde physique doit être éliminée et il pense qu'il est temps d'établir un système intégrant beaucoup de produits

qui interagissent entre eux. Cette vision a été nommée l'internet des objets (*Internet of Things*) qui a été adoptée par [Gershenfeld et al., 2004], [Huvio et al., 2002]. Dans le contexte de la gestion d'une chaîne de production, le concept de l'internet des objets (IoT) est proche de celui de produit intelligent.

En Angleterre, à l'IfM (Institute for Manufacturing), de l'Université de Cambridge, l'équipe de recherche (Centre for Distributed Automation and Control) a prolongé la notion de produit intelligent dans le domaine manufacturier. Selon [McFarlane et al., 2002] et [Bajic, 2009] un produit intelligent est défini comme une représentation physique et informationnelle d'un objet avec les caractéristiques suivantes :

- Il possède une identification unique ;
- Il est capable de communiquer efficacement avec son environnement ;
- Il peut mémoriser ou stocker des données au sujet de lui-même ;
- Il dispose d'un langage de communication pour transmettre ses caractéristiques et ses besoins pendant son cycle de vie ;
- Il est capable de participer ou de prendre des décisions appropriées à son propre destin de façon continue.

Selon cette définition [Wong et al., 2002] a classé ce produit dans deux catégories. Quand le produit est capable juste de remonter ses variables d'environnement et tout type d'information sur le système, le produit est de 1er niveau. Mais, si ce produit peut prendre la décision et traiter ce flux de données le produit est de deuxième niveau ou produit intelligent.

[Kärkkäinen et al., 2003] affirme aussi que chaque produit intelligent doit présenter quelques qualités :

1. Un code d'identification unique.
2. Une source où il peut s'informer via une hiérarchie bien définie. Cette source fournit la donnée nécessaire telle que la nature du produit, des mises à jour à sa base de connaissance, un code d'identification, et les mécanismes de consultation.
3. Interagir avec son milieu et communiquer avec son entourage en développant ses besoins (même pro-activement).

Selon Ventä [Ventä, 2007] chaque produit intelligent doit effectuer les tâches suivantes :

1. Surveiller continuellement son état et son environnement.
2. Réagir et s'adapter à son environnement et ses conditions de fonctionnement.

3. Garantir un fonctionnement optimal même au changement de circonstances et en cas d'exception.
4. Interagir continuellement avec l'utilisateur, l'environnement ou avec d'autres produits.

L'identification automatique implique d'obtenir automatiquement l'identité d'un objet au moyen d'une technologie donnée. Par exemple, l'accès en temps réel à l'identité d'un produit grâce à l'identification automatique par radiofréquence permet de déterminer la présence d'un produit et, indirectement il est possible de déterminer sa localisation [Finkenzeller, 2003]. Il est important de signaler que dans la définition donnée de produit intelligent il est possible de distinguer deux niveaux de complexité : un produit apportant de l'information dans son environnement et un produit supportant des mécanismes de décision [Wong et al., 2002]. Ce dernier implique que le produit doit avoir une capacité d'analyse intégrée afin d'évaluer et de prendre la meilleure décision en fonction de son état et de son contexte.

Comme exemple concret de l'application du concept de produit intelligent, nous pouvons citer la fonction de traçabilité [Jansen-Vullers et al., 2003] de produits dans son cycle de vie. Le but dans ce cas est d'enregistrer et d'actualiser toute information dynamique associée à un produit (tels que ses états, les opérations qu'il a suivi, …) soit directement sur une étiquette électronique RFID, soit sur une ou plusieurs bases de données distantes afin de relier chaque produit physique (individuellement) avec sa représentation informationnelle. Grâce à cela, il est possible qu'un acteur de la chaîne logistique puisse connaître à tout moment l'histoire détaillée d'un produit. Eventuellement, l'information enregistrée sur une étiquette électronique ou sur des bases de données peut être employée comme entrée d'un processus à posteriori afin d'optimiser une opération donnée.

Comme autre exemple, nous citons l'apport de l'information stockée sur un produit intelligent dans un processus de contrôle d'un système de production. La Figure I.10 [McFarlane, 2003] illustre une boucle de contrôle d'un processus industriel où la donnée est enrichie avec celle contenue dans le produit. L'introduction d'un système d'identification automatique permet au produit physique d'être reconnu et peut ainsi apporter ses informations afin d'influer sur le processus décisionnel. Ceci implique d'attribuer un rôle plus actif à un produit physique. Eventuellement la partie décisionnelle associée au produit peut être incluse dans sa représentation virtuelle lui permettant de prendre des décisions autonomes et ainsi devenir une entité active influençant son cycle de vie [Cea and Bajic, 2004] [Bajic and Cea, 2005]. Dans ce même ordre d'idée, [McFarlane et al., 2002] affirme qu'un produit intelligent est un article

manufacturé qui possède la capacité de surveiller, d'analyser et de raisonner sur son état actuel ou futur et, si nécessaire, d'influencer son destin. Comme avantages de cette approche nous pouvons citer la potentialité qu'a un produit de s'adapter (adaptabilité) face aux opérations aléatoires (commandes urgentes, pannes des machines, ruptures de stock, reconfiguration des lignes, etc.) ou aux réorganisations de la production, du stockage, …, et aux besoins dynamiques et potentiellement évolutifs des clients.

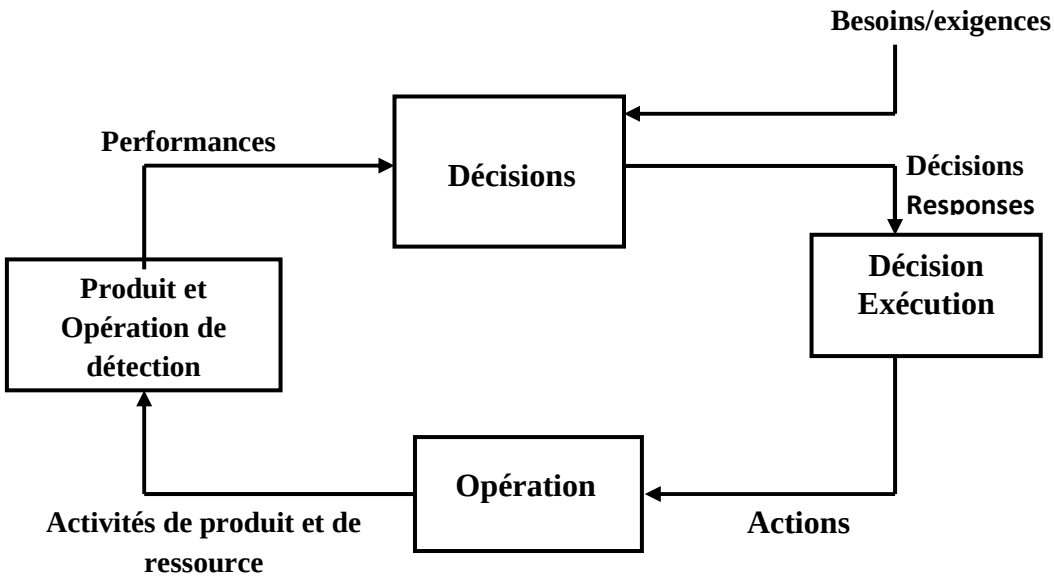

Figure I. 10. Apport de l'information du produit dans le processus décisionnel [McFarlane, 2003].

L'introduction de l'identification automatique fournit la possibilité d'employer l'information concernant chaque produit de façon individuelle (son identité, ses états, son histoire, …) en même temps que les données opérationnelles (tels que la température, position, vitesse, statuts des équipements, …) afin de contrôler le système. La combinaison de ces deux mécanismes peut clairement améliorer les processus de décision et de contrôle [McFarlane, 2003].

V. Les projets de recherche majeurs dans les domaines de l'intelligence ambiante

De nombreux projets de recherche ont vu le jour au niveau international, depuis l'essor des paradigmes précédemment présentés. Nous allons en présenter les plus significatifs en relation avec notre problématique de recherche.

V.1. OXYGEN

Le projet OXYGEN a été développé en 2002 au sein du Laboratoire Media du MIT[7]. Son objectif est de concevoir un monde futur dans lequel la technologie est centrée sur l'utilisateur et hautement répartie au point qu'elle devient disponible partout comme l'oxygène dans l'air [MIT Project Oxygen, 2002]. Cette technologie offre une grande efficience avec un minimum d'effort produit auprès de l'utilisateur (dispositifs adaptables à son environnement et réagissant au langage de communication naturel). La réalisation de ce projet repose sur les aspects suivants :

- La diffusion : technologie embarquée dans n'importe quel dispositif et disponible partout ;
- La mobilité : offrir à l'utilisateur une liberté de déplacement ;
- L'adaptabilité : technologie montrant une accommodation aux changements survenus ;
- L'efficacité : technologie capable d'interagir automatiquement avec les ressources informatiques dans l'environnement ;
- L'éternité : dispositif ne devant jamais s'arrêter et disponible tout le temps.

V.2. MediaCup

Le laboratoire TecO de l'université de Karlsruhe a initié plusieurs projets d'application notamment sur le thème ubiquitous computing et des objets communicants.

Figure I. 11. MediaCup en communication avec une montre intelligente

[7] www.oxygen.lcs.mit.edu/

Le MediaCup [Beigl *et al.*, 2001] constitue une simple tasse de café équipée de capacités d'exploration et de communication avec son environnement comme le montre la figure I.11. La tasse est capable de fournir des états d'information introspectifs, tels que la température de la boisson, le fait qu'elle est en cours de consommation, la liste des autres objets à son voisinage. Contrairement au moyen d'identification commun utilisé dans les réseaux de communication (ex. adresse IP), l'approche introduite dans le MediaCup permet aux objets de communiquer grâce au partage du même environnement d'opération et non parce qu'ils se connaissent déjà.

V.3. AITPL

AITPL (Ambient Intelligence Technologies for the Product Lifecycle) c'est un projet européen qui a été financé par la commission européenne (EC) dans le cadre du 6ème PCRD.
Le projet AITPL mène une réflexion sur la gestion de cycle de vie de produit (PLM : Product life cycle Management). La gestion de cycle de vie d'un produit (PLM) est le procédé qui gère la totalité du cycle de vie d'un produit : conception, production, assistance et mise à jour. PLM doit améliorer de nombreuses caractéristiques d'un produit comme la personnalisation, la configuration par utilisateur, la facilité de maintenance et la mise à jour de service de ce produit et l'auto diagnostique.

Les défis technologiques et de recherche, qui ont été discutés au cours de ce projet peuvent être classés dans les principaux domaines suivants [Frederix et al., 2006] :

1. Développement de produits intelligents, qui possèdent et recueillent les informations au cours de leur «cycle de vie» soutenu les technologies de l'intelligence ambiante ;
2. Elaboration des processus de production intelligents qui s'adaptent en fonction de la demande du client ;
3. Développement de nouveaux services sensibles au contexte (« Context awareness ») ;
4. Développement des technologies et des processus qui conduit à respecter les règles de sécurité.

La nouvelle génération des produits rassemblera des données multiples propres à leurs perceptions et les emploiera pour estimer et calculer les risques potentiels. Une telle intelligence réduira notamment le risque de contrefaçon. En outre, elle facilitera les opérations de maintenance en utilisant les informations de traçabilité du produit à travers des données stockées.

V.4. MemoClip

Le MemoClip [Beigl, 2000] est une forme de PDA portatif qui rappelle à son utilisateur la liste des tâches qu'il a à faire en fonction de sa localisation. L'utilisateur peut associer l'information à se rappeler à une description d'une localisation, en la téléchargeant vers le MemoClip. Par conséquent l'utilisateur peut recevoir ultérieurement des notifications lorsqu'il se trouve à l'endroit spécifié. La particule est équipée par un processeur, des capteurs, des moyens de communication et d'un afficheur LCD comme le montre la figure I.12. Il est conçu de manière qu'il soit transportable par son utilisateur.

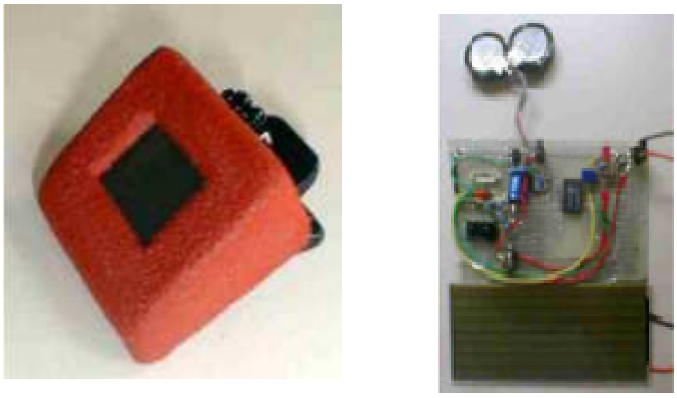

Figure I. 12. Le Memo Clip avec son dispositif de localisation

V.5. Smart-Its

Le projet Smart-Its[8] a vocation à définir des plateformes hardware et software intégrant des capacités d'exploration et de communication dans des objets communicants d'utilisation quotidienne.Smart-Its consiste en une plateforme objet communicant hardware composée de capteurs diversifiés, d'un processeur, d'une mémoire et de communication Radio Fréquence comme le montre la figure I.13 [Beigl et al., 2003a].

Le macro-composant Smart-Its intègre des objets programmables (par leurs utilisateurs) leur permettant de s'adapter au besoin des applications spécifiques et peut s'attacher à n'importe quel objet ordinaire (chaise, bureau, stylo…) pour améliorer leurs fonctionnalités.

[8] smart-its.teco.edu

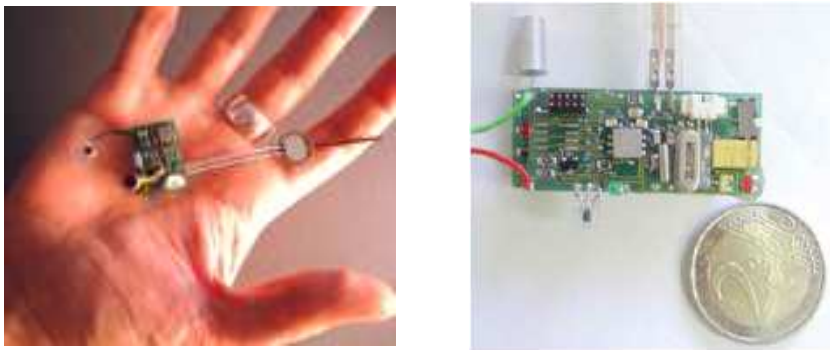

Figure I. 13. Plateforme Smart-Its

V.6. DigiClip

Le projet Digiclip [Decker et al., 2004a] apporte une solution pour permettre la conversion d'un simple document de papier passif en un document physique actif. Il s'agit d'une pince à document, équipée de capteurs ainsi que par des technologies de communication et de traitement de l'information, attachée à un document de papier comme l'illustre la figure I.14. Le dispositif est capable de surveiller l'état du document et de communiquer avec un composant software qui surveille les circonstances du même document. De cette façon, le système DigiClip gère des applications pour contrôler et coordonner des restrictions d'accès et pour garder les traces sur l'évolution du document physique et virtuel.

Figure I. 14. Le DigiClip

V.7. eSeal

Le projet eSeal [Decker et al., 2004b] a pour objet de concevoir un sceau électronique qui s'attache aux articles physiques dans le but de certifier authentique une protection d'objets.

Ce projet est en fait inspiré du sceau de cire classique qui est une méthode archaïque assurant la préservation de la probité des documents de valeur. Le sceau classique est utilisé pour enfermer les documents en question en les marquant par une empreinte de cire. Par conséquent, toute violation de l'intégrité de ces documents provoque la destruction irréversible du sceau. Tout comme le sceau de cire classique, eSeal n'assure pas une protection physique des articles mais fournit des informations et des preuves relatives à l'intégrité et l'authenticité. En fait, eSeal comporte un dispositif de traitement, des services et des protocoles de communication ainsi qu'une plateforme de capteurs attachés à l'objet physique assurant l'exploration de son environnement. Un algorithme logique définit l'état du sceau : intact ou détruit.

V.8. Wisden

Wisden (Wireless Sensor Network for Structural Health Monitoring) est un projet crée par [Chintalapudi et al., 2006]. C'est un réseau de capteurs sans fil basé sur un système d'acquisition des données pour la surveillance d'ouvrage d'art. Wisden recueille en permanence les données sur la structure à partir d'un réseau multi-hop de nœuds de capteurs, affiche et stocke les données à une station de base. Wisden peut être, et a été déployé sur des ouvrages réels pour recueillir des données de vibration. Parmi les principales caractéristiques de Wisden nous citons la livraison fiable des données et la facilité et la souplesse de déploiement.

Wisden est mis en œuvre sur les deux dispositifs MicaZ/Mica2, et il a été déployé et testé dans des environnements réels.

V.9. WASP[9]

Dans le contexte du "scénario de Soins médicaux", Une personne porte un Réseau de capteurs (Body Sensor Network), qui contrôle constamment un ensemble de données spécifiques du corps ou du comportement de la personne [Corroy et al., 2009].

Il est supposé que les patients doivent recevoir l'assistance d'un médecin en cas de problème. Les détecteurs répartis sur le corps sont utilisés pour récupérer les renseignements sur le patient comme la tension, la fréquence cardiaque, etc. Les patients doivent périodiquement

[9] http://www.wasp-project.org: le site web du projet WASP

envoyer ces renseignements à leurs médecins, qui peuvent alors analyser des données et prendre des décisions concernant le traitement des patients.

V.10. Cobis

Une équipe de recherche à l'Université de Lancaster [Strohbach et al., 2005] a lancé un projet CoBis basé sur l'approche des objets communicants pour la manipulation et le stockage des produits chimiques.

CoBis introduit des composants électroniques coopératifs [CoBIs, 2008]. Ce sont des objets physiques intégrants éventuellement les capacités d'exploration, de communication, de traitement et d'action. Contrairement aux autres approches, ces composants ne nécessitent pas une infrastructure externe.

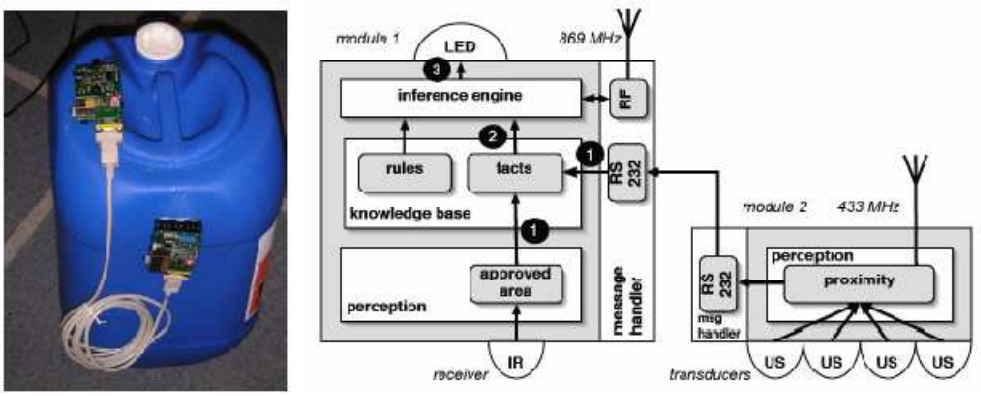

Figure I. 15. Conteneur équipé par une particule intelligente

CoBis a exploité ces dispositifs dans le domaine de l'industrie pétro-chimique avec la société Total. Leur démarche a commencé par identifier quelques scénari menant à des situations de danger. Puis, ils ont équipé des containeurs chimiques par ces composants intelligents pour rassembler les informations nécessaires sur l'environnement externe, et pour pouvoir réagir en cas de dysfonctionnement comme le montre la figure I.15.
Quant à [Strohbach *et al.*, 2005], il a étudié la compatibilité entre les produits et la distance qui les sépare.

V.11. Applications Urbaines

[Chen et al., 2007] a traité la sécurité inter-véhicules dans les autoroutes. Il présente un modèle d'interaction entre agents mobiles en utilisant deux protocoles IEEE et ASTM

destinés aux communications courtes distances. Les performances désirées sont atteintes à condition de minimiser le retard des paquets et de maximiser le nombre des paquets envoyés. Le retard de transmission maximal toléré est de l'ordre de 0.4s car la réaction de conducteur est estimée entre 0.75s et 1.5s. La portée du réseau est de l'ordre de 1000 m. Cette étude permet d'analyser la fiabilité de 802.11a (wifi) dans des applications à fortes contraintes temporelles.

De même [Robinson et al., 2007] décrit l'utilisation de protocole DSRC (dedicated short-rang communication) dans l'échange de données entre produits installés dans une voiture et ceux installés aux feux et panneaux routiers ainsi que dans l'infrastructure (gestionnaire). Les échanges sont subdivisés en 4 types de liaison P2P (point-to-point) pour avertir une voiture d'un risque de collision avec une autre comme l'indique la figure I.16. L'approche P2M (point-to-multipoint) est aussi utilisée pour avertir toutes les voitures de voisinage de la présence d'une violation de feu.

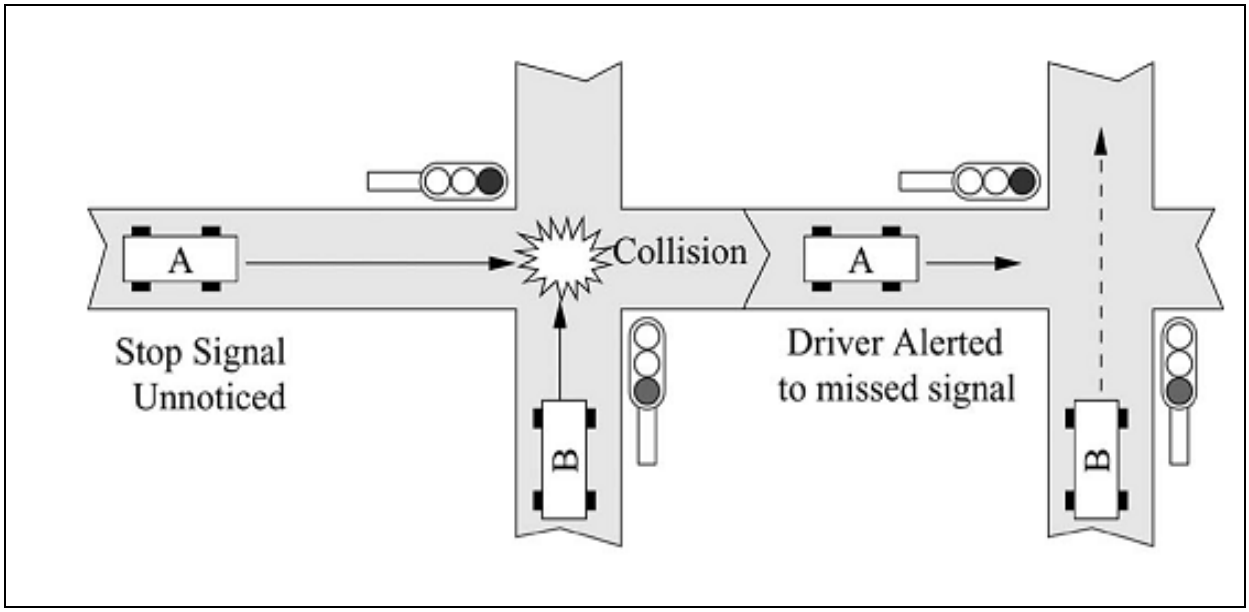

Figure I. 16. Estimation de collision par WSN

Dans ce sens [Hoehmann et al., 2009] a étudié ce même concept de communication entre véhicules (car2car) avec le logiciel de simulation TOSSIM.

Comme l'indique la figure I.17 chaque nœud de capteur indique pour chaque voiture l'altitude, la longitude, la vitesse, l'accélération, l'état du frein, l'angle de déviation des roues directrice… Les scénarios (de collision) sont élaborés a l'aide de module SUMO (simulation

microscopique de trafique routière). A chaque fois on prélève la position, la vitesse et la direction de véhicule (Gi):

$$Gi\begin{pmatrix}Xi\\Yi\end{pmatrix}=\begin{pmatrix}Xi+V(xi).t\\Yi+V(yi).t\end{pmatrix} \qquad (1.1)$$

Et l'estimateur est chargé de prévoir en temps réel les cas de collisions. Ces cas seront détectés entre deux véhicules (0) et (i) si cette équation est satisfaite :

$$\begin{pmatrix}X0+V(x0).t\\Y0+V(y0).t\end{pmatrix}=\begin{pmatrix}Xi+V(xi).t\\Yi+V(yi).t\end{pmatrix} \qquad (1.2)$$

Dans un autre cas d'utilisation, [Kurata et al., 2005] ont étudié la possibilité d'utiliser des réseaux de capteurs sans fil de type MICA2 pour la surveillance des tremblements de terre. Une comparaison avec des résultats obtenus par un accéléromètre de référence montre l'intérêt de cette approche pour surveiller les risques sur les bâtiments.

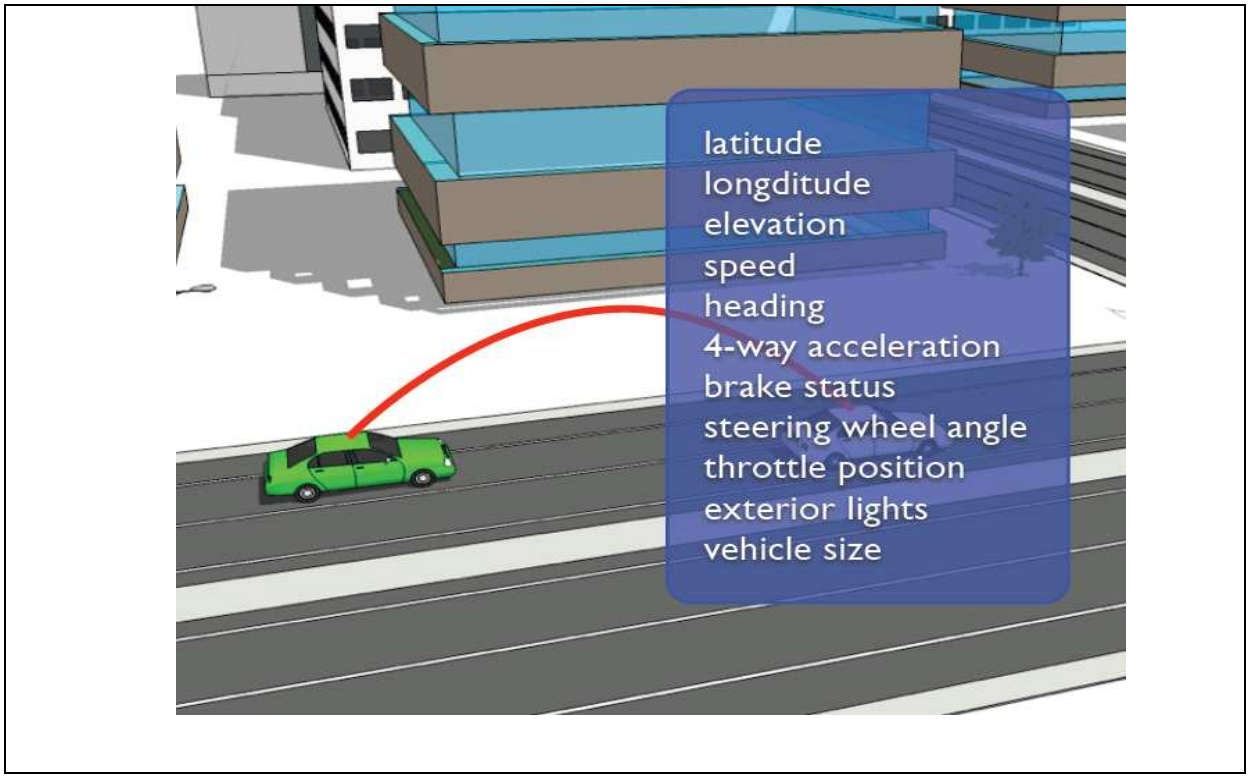

Figure I. 17. Application WSN embarquée pour les véhicules

V.12. Applications robotiques

[Caballero et al., 2008] a étudié la localisation dynamique des nœuds d'un réseau de capteur sans fil (positionnement) en utilisant un robot mobile équipé d'un DGPS[10] adapté au WSN.

Cette localisation est faite à l'aide de la technique RSSI (toutes les produits et le robot sont équipés d'un module RSSI) : à l'initialisation du système les nœuds essaient de se localiser à l'aide de l'échange des messages RSSI (Radio Signal Strength Indicator), ce qui peut s'envisager car le réseau est statique (les positions des nœuds sont fixes) comme l'indique la figure I.18.

Le robot mobile, équipé de DGPS, lui permet de se localiser. Et grâce au module RSSI, le robot peut estimer sa position entre les nœuds.

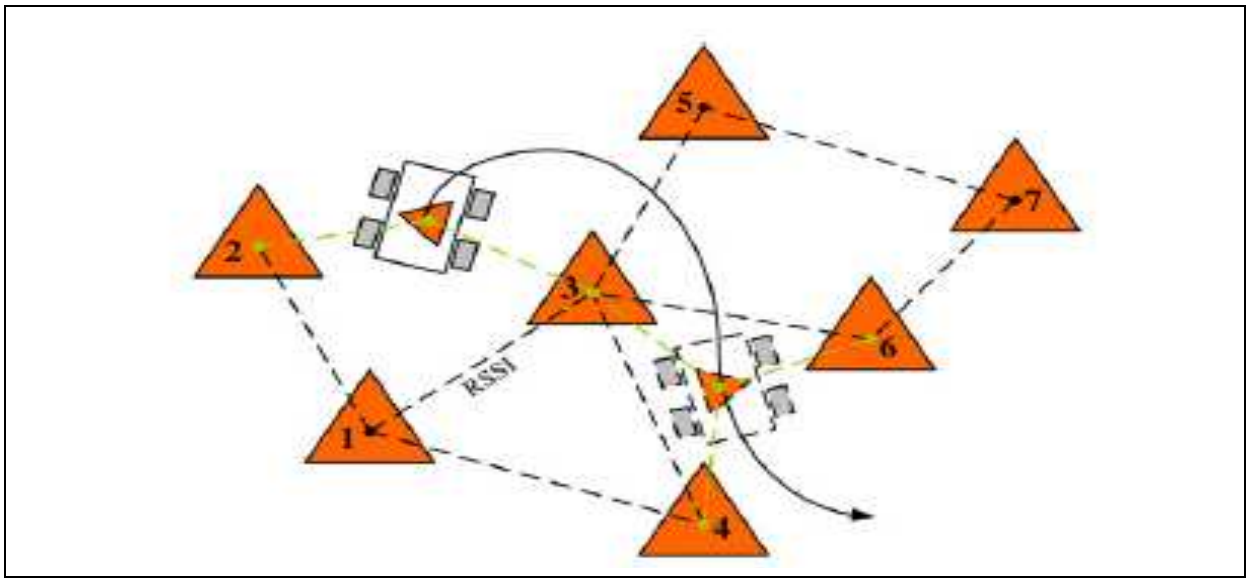

Figure I. 18. Localisation par WSN

V.13. Applications Navales

[Bostwick et al., 2009] a conçu le projet PANDA[11] au sein du département DARPA[12]. Il illustre la coopération entre les nœuds de capteurs distribués sur 100 000 bateaux au sein d'un système de surveillance qui détecte toute anomalie de navigation maritime (vitesse excessive,

[10] Le DGPS : Différentiel Global Positioning System, est une amélioration du GPS.
[11] http://www.darpa.mil/ipto/programs/panda/panda.asp.
[12] The Defense Advanced Research Projects Agency, Information Processing Techniques Office.

changement anormal de trajectoire, arrivé au port…). PANDA[13] supervise aussi les variables environnementales (climat, état de mer…) et les risques de piratage.

VI. Problématique du sujet de recherche

La problématique de recherche que nous développons dans cette thèse concerne la modélisation du comportement interne d'un objet communicant et de ses interactions ainsi que les processus de coopération entre objets communicants. Le contexte est un environnement à intelligence ambiante, intégrant des produits et objets physiques équipés d'intelligence embarquée et de moyen de communication sans fil (RFID, Réseau de capteurs, ...) dans l'objectif d'assurer la sécurité intrinsèque et extrinsèque des produits industriels à caractère chimique dangereux comme l'illustre la figure I.19.

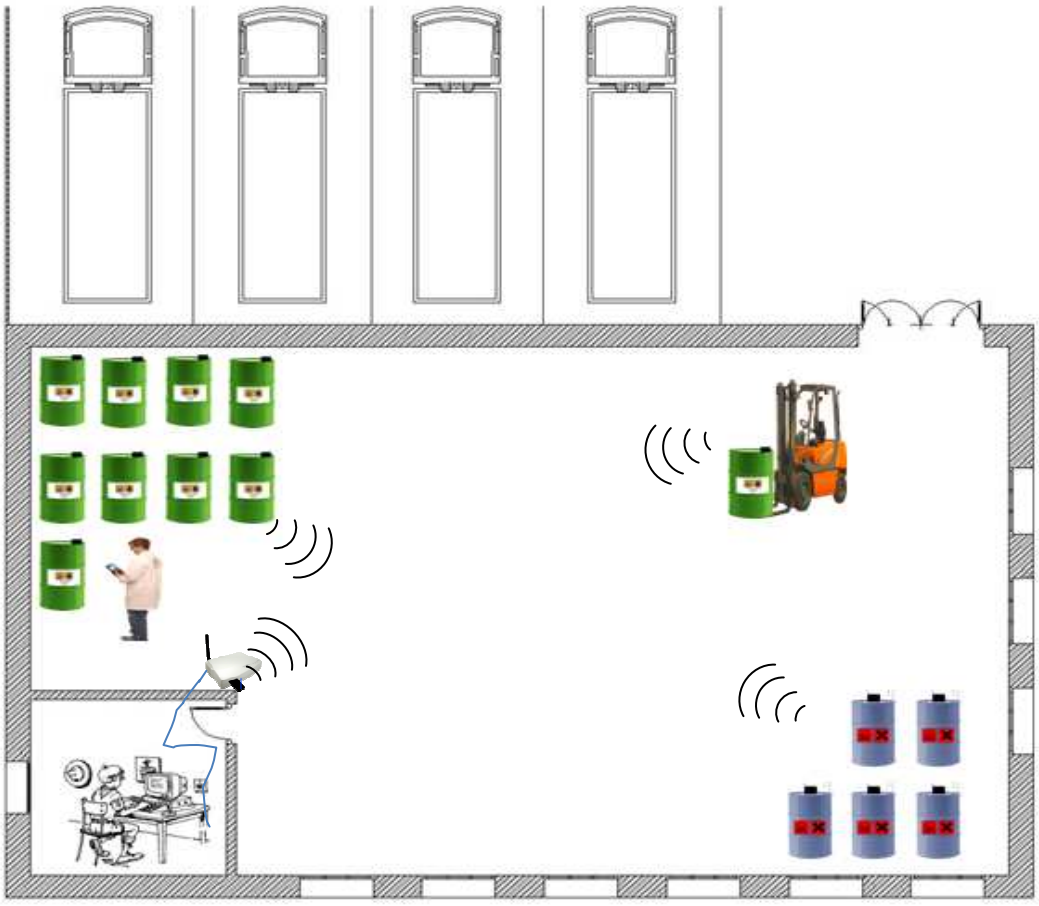

Figure I. 19. Système de sécurité active

[13] http://www.darpa.mil/ipto/programs/panda/panda.asp.

L'objectif de notre thèse consiste à analyser, définir, formaliser et mettre en place des mécanismes d'intelligence ambiante dans les environnements de stockage et de logistique industrielle du domaine chimique, afin d'améliorer les interactions entre les produits et les acteurs du système de manière à assurer la sécurité active des biens et des personnes. Il s'agit de transformer les produits de nature dangereuse en « Produit Communiquant » que nous caractériserons de « Produit Actif » capable de communiquer, informer, acquérir, décider et réagir aux stimulis et perturbations de son environnement afin de permettre au produit de prendre en charge sa sécurité intrinsèque et la sécurité globale dans ses interactions avec d'autres produits ou personnes.

Nous modéliserons le comportement interne de produits dans un cadre d'application de gestion de sécurité active entre des conteneurs industriels (modèle fût chimique) qui seront ainsi transformés en Produits à Intelligence Ambiante.

Nous spécifierons et évaluerons un protocole de communication de niveau application orienté sécurité des produits, permettant de garantir les objectifs de sécurité fixés.

Chaque conteneur - transformé en *Produit Actif* – est capable de surveiller individuellement la conformité de son environnement avec ses spécifications propres (Température, Mouvement-choc, …) et surveille la proximité d'autres conteneurs dans son environnement ambiant (présence d'un conteneur réactif, absence d'un autre tel qu'un extincteur par exemple), tout en surveillant aussi sa compatibilité avec les autres produits proches. Toute modification de son environnement enfreignant les règles de sécurité individuelle ou mutuelle entre produits actifs doit engendrer des actions externes permettant de recouvrir les situations de dangerosité.

Il s'agit alors de s'intéresser à la spécification et à la définition du comportement d'un produit communicant par une modélisation du comportement interne d'un Produit Actif mettant en évidence les fonctions d'information, de communication, de traitement, de décision et d'action. Ce modèle devra permettre la coopération directe Produit à Produit, reposant sur des interactions intelligentes basées sur une communication ambiante sans fil.

La figure I.20 représente le graphe des interactions ou diagramme de Pieuvre, pour analyser les besoins et identifier les fonctions de service du PA. Ce diagramme est utilisé dans la méthode d'analyse fonctionnelle APTE et permet de spécifier les fonctions et les contraintes de produit.

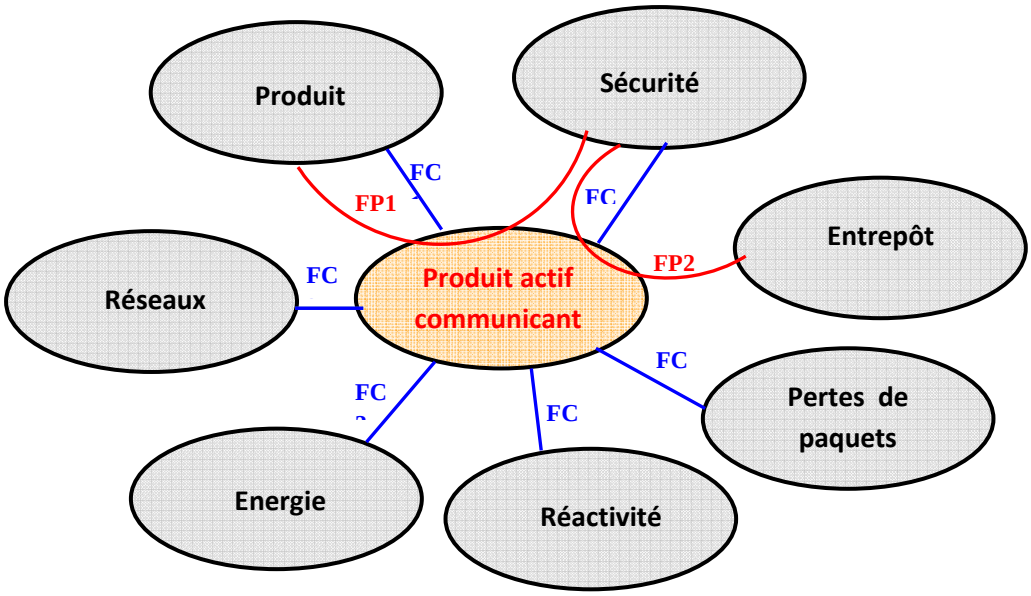

Figure I. 20. Diagramme en pieuvre APTE d'un produit actif communicant

Le tableau 2 indique le type de fonctions de services (Fonction principale FP) ou (Fonction Contrainte FC). Cette approche fonctionnelle vise à définir une liste des solutions techniques non exhaustive.

Afin de résoudre la problématique du sujet il faudra tenir compte de ces fonctions.

FP1	Gérer la sécurité active de produits
FP2	Gérer la sécurité active de l'entrepôt
FC1	Contrôler l'état de produit
FC2	Communiquer via le réseau
FC3	Optimiser l'énergie consommée
FC4	Réagir au changement de son environnement
FC5	Tenir en considération les pertes de paquets
FC6	Adapter à l'organisation de l'entrepôt
FC7	Respecter les règles de sécurité

Tableau 2. Fonctions de services

VII. Sécurité active pour la gestion des produits chimiques dangereux

L'industrie chimique est exposée quotidiennement à des risques affectant la sécurité du matériel et de son personnel travaillant au cœur du système de production. Ainsi, une politique permettant de tracer les mesures de sécurité sur la manipulation de produits dangereux est indispensable à mettre en œuvre. Il est donc impératif de développer des règles de gestion sur les contraintes imposées par la manipulation des produits chimiques dangereux.
L'identification des produits dangereux suit une signalétique définie par des conventions internationales. Le système européen d'étiquetage des substances dangereuses est défini dans la directive 67/548/EEC (27 juin 1967).
A chaque type de danger correspond un symbole international de danger encore appelé pictogramme de risque qui est obligatoirement reporté sur l'étiquette du récipient par le fournisseur lorsque ce danger est connu. Les différents pictogrammes de risques sont reproduits sur la figure I.21.

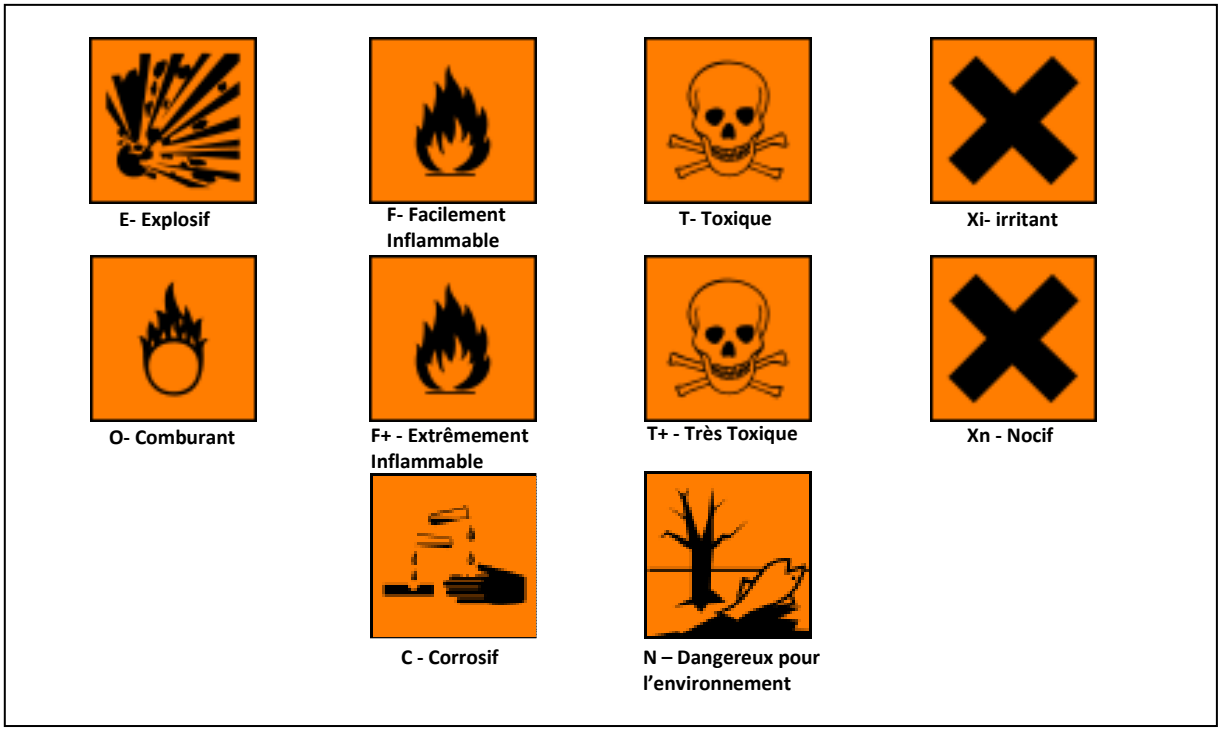

Figure I. 21. Pictogrammes réglementaires des différents types de dangers des substances chimiques

De façon générale, lorsqu'on dispose d'un fût contenant une substance chimique, un système de sécurité doit être implémenté et considérer tous les facteurs qui peuvent aggraver le niveau du risque intrinsèque à cette substance potentiellement toxique, explosive ou bien nocive pour

son environnement. Comme cité dessus, plusieurs règlementations à travers le monde aident à aménager la classification, l'emballage et l'étiquetage les substances dangereuses. Ces règlements couvrent également le domaine de stockage et du transport de ces produits dangereux. Ils doivent respecter des instructions sécurisantes concernant le groupement des produits de natures différentes ainsi que leur disposition dans le milieu de stockage. Ceci évite ainsi les risques d'incidents provenant des incompatibilités potentielles entre substances chimiques ou de l'agencement de certains produits entre eux pouvant déclencher des réactions dangereuses et intempestives. L'arrangement des produits obéit donc aux règles de classement élaborées par divers organisme, repris par le catalogue Merck[14] qui présente une matrice précisant les différentes classes de matières dangereuses et qui indique les compatibilités de stockage en proximité des produits.

Le bulletin de promotion de la conformité (PROCONF 12) en vigueur au Canada, dresse aussi un tableau illustrant la façon de disposer les produits suivant un système de classification des risques issu de la *Loi sur le transport des marchandises dangereuses*[15].

[Hatayama et *al.*, 1980] ont aussi établi un tableau indiquant les incompatibilités des produits chimiques selon leur groupe chimique fonctionnel (Acides minéraux non oxydants, alcools, aldéhydes, cétones, etc.). La figure I.23 montre la matrice d'incompatibilité du catalogue Merck qui fonctionne suivant les symboles de sécurité des substances.

Par conséquent, le niveau de sécurité de chaque produit pris individuellement ou relativement avec la présence d'autres produits est le sujet central de cette thèse. Une gestion de sécurité efficace doit nécessairement être basée sur une approche de modèles de sécurité distribués supportés par les produits eux-mêmes. De ce fait, chaque produit est une entité active de tout l'ensemble du système de sécurité à travers des fonctionnalités embarquées dans les produits telles que le contrôle, la surveillance, la prise des décisions et le déclenchement d'alarmes. Les technologies de communication et l'intelligence ambiante peuvent apporter une nouvelle vision pour créer un système de sécurité fiable où les produits chimiques sont transformés en produits actifs intelligents. Dans ce concept, un produit c'est-à-dire le fût, est capable de se rendre compte des conditions de son stockage, des changements survenus dans son environnement et sur les autres produits situés à sa proximité. Par la suite, le produit peut avoir un comportement réagissant suivant des changements de conditions inappropriés et basés sur les règles de sécurité dont il fait l'objet. Par l'application d'un réseau de capteurs sans fil, monté sur les fûts des produits chimiques, on aboutit à créer un réseau de produits

[14] www.merck.com

[15] www.tc.gc.ca/canutec/fr/menu.htm ,Service correctionnel Canada ANNEXE A

actifs intelligents interagissant ensemble pour former finalement un système de sécurité active.

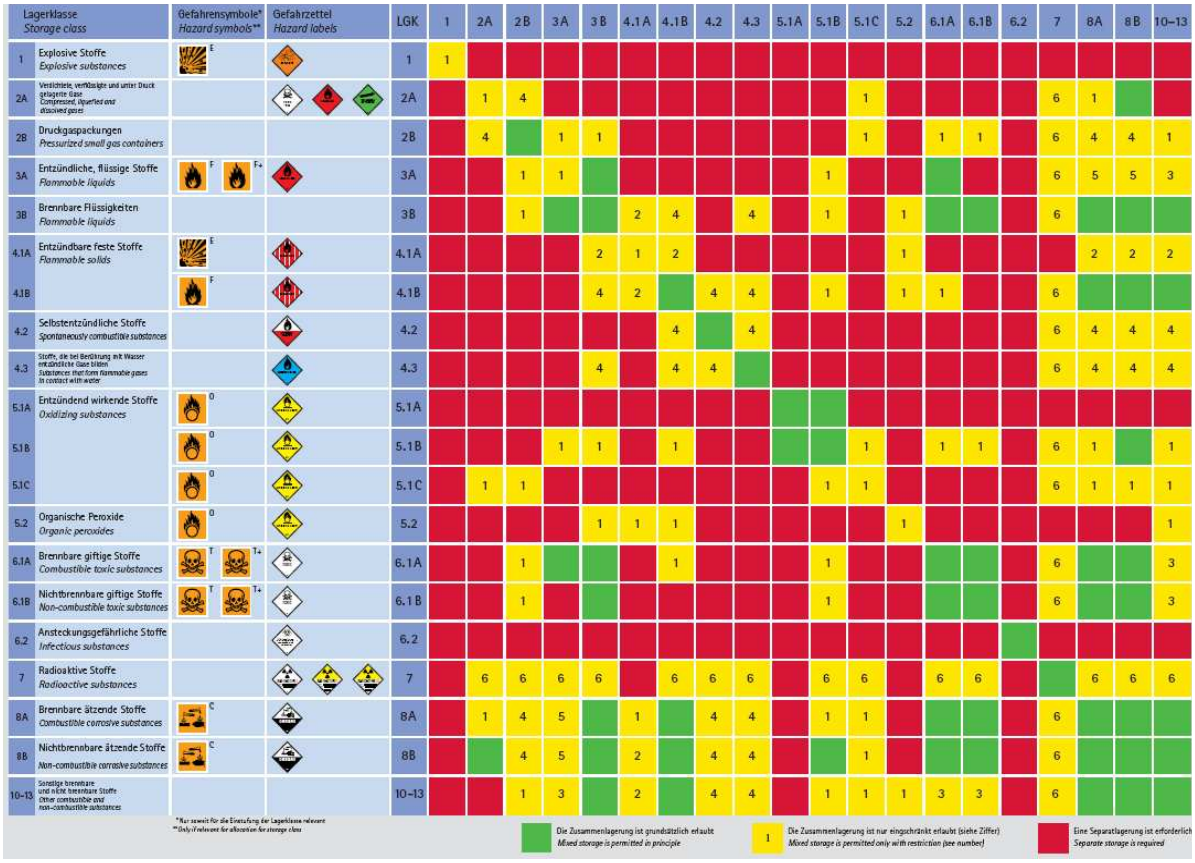

Figure I. 22. Matrice d'incompatibilité du catalogue Merck

VIII. Conclusion

Dans ce chapitre les différents paradigmes sur l'intelligence ambiante on été présentés. L'intérêt de l'intelligence ambiante a été illustré à travers plusieurs champs applicatifs. Le but de cette thèse est de l'instancier à la gestion de la sécurité de produits chimiques durant ses phases de stockage. Dans la suite de ce mémoire, nous allons nous attacher à proposer des modèles de gestion de sécurité qui seront directement embarqués sur les produits. Le comportement de ces produits devenant ainsi actifs sera ensuite simulé pour évaluer les performances globales du système.

CHAPITRE II

Spécification du produit actif communicant et de ses services associés

I. Introduction

Après avoir vu les travaux sur l'intelligence ambiante et sur les réseaux de capteurs, on va se pencher sur son intégration pour former des produits actifs. Ce concept « Produit Actif » est défini par [Zouinkhi et al., 2007] [Dobre and Bajic, 2007]. Chaque produit constitue un nœud actif d'un système de sécurité global au moyen d'un modèle réactif embarqué. Dans ce chapitre, nous exposons en premier lieu certains problèmes liés à la sécurité de l'environnement. Et dans une deuxième partie, nous proposons le modèle interne du produit actif en définissant ce concept et son comportement relativement aux changements de son environnement (modification des mesures ambiantes, modification des distances entre les produits chimiques, …) et en présentant les entêtes des messages échangés entre produits actifs afin de gérer la sécurité active.

I.1. Concept de Produit actif

Le concept de produit dit actif consiste à doter un produit de capacités à communiquer, informer, acquérir, décider et réagir aux stimuli et perturbations de son environnement afin de permettre au produit de s'adapter, d'influer, de coopérer, de transformer le comportement de son environnement. Le produit est ainsi un acteur intelligent et proactif dans son environnement ambiant avec lequel il interagit au moyen de communication sans fil. Dans l'industrie chimique, on peut utiliser ce concept sur un produit industriel de type conteneur pour l'administration de sécurité des biens et des personnes. Cet objet est composé d'un fût contenant une substance chimique auquel est attaché un dispositif microélectronique qui peut communiquer avec d'autres dispositifs qui sont attachés aux autres fûts. Ainsi un Produit à Intelligence Ambiante est capable de « sentir » son environnement, à travers des capteurs embarqués, de décider et faire un choix d'action/réaction selon des spécifications propres et/ou partager dans un environnement coopératif, de communiquer avec son environnement [Dobre et al., 2009] [Zouinkhi et al., 2007].

Un produit est un objet qui joue deux rôles : un rôle « passif » de capture de mesure (des valeurs ambiantes) et un rôle « actif » de action/réaction (réaliser des échanges et des interactions intelligentes avec les autres produits) à travers des capteurs embarqués [Zouinkhi et al., 2009b]. Toute modification de son environnement enfreignant les règles de sécurité individuelle ou mutuelle doit être détectée, diagnostiquée et doit engendrer des actions externes permettant de recouvrir par des actionnements le niveau de sécurité fixé.

Les capacités liées à un produit actif sont illustrées dans la figure II.1.

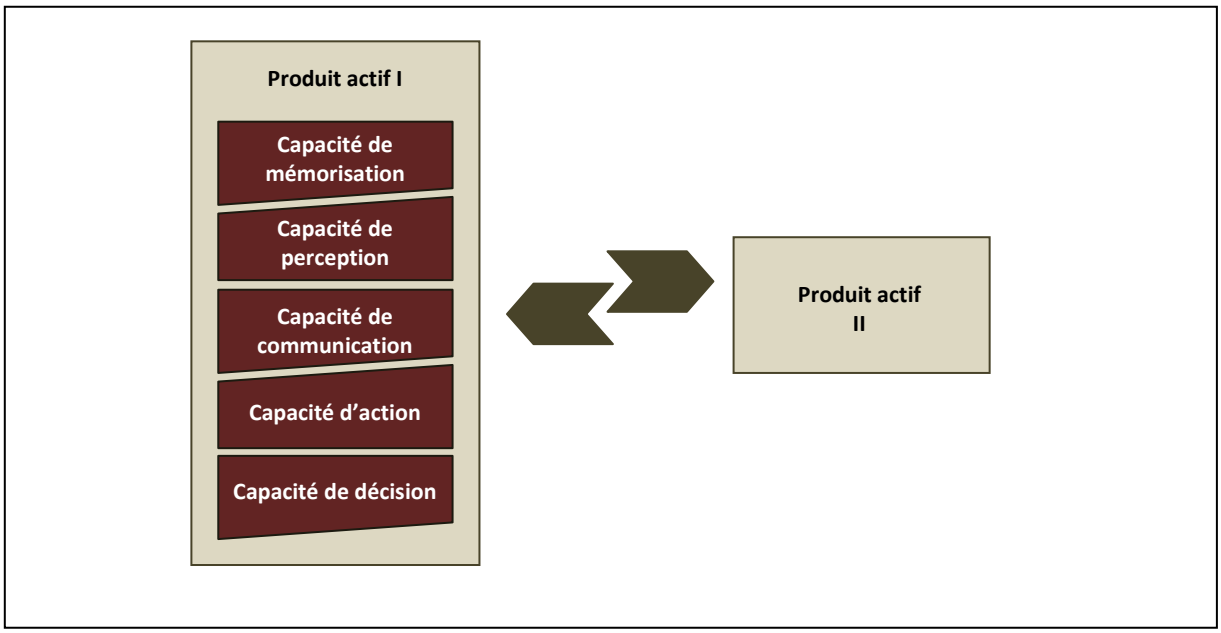

Figure II. 1. Les capacités liées à un Produit Actif

II. Problèmes de la gestion de la sécurité

II.1. Mécanismes d'interaction

L'organisation des systèmes autonomes repose sur une maîtrise de son système d'information. Cette maîtrise englobe les aspects contrôle, sécurité, administration de façon à pouvoir gérer les interactions entre les différents éléments constituant le système autonome. Cette problématique est très bien illustrée par [Garate et al., 2005] qui ont développé un réseau rassemblant des appareils domotiques (TV, réfrigérateur, etc.) connectés et gérés par un contrôleur central. Le réseau donne à l'utilisateur les moyens de communiquer avec ces appareils et d'interagir avec eux à l'aide de services et fonctions en langage naturelle. Cependant, ces interactions passent obligatoirement par un contrôleur central et n'intègre pas dans les appareils domotiques des capacités décisionnelles. Un autre exemple développé par [Rajeet et al., 2005] montre une application de la technologie pervasive RFID pour l'antivol de vélos dans un campus universitaires. Une étiquette RFID est placée sur chaque vélo et les informations sont contrôlées par un PC situé dans une salle de contrôle. L'approche est donc centralisée et en cas du dysfonctionnement du gestionnaire central, le système n'est plus opérationnel.

Notre objectif consiste donc à intégrer des mécanismes d'interaction dans les objets pour qu'ils soient capables de communiquer, acquérir des informations, décider et réagir aux stimuli et perturbations de son environnement. Le but est que le produit puisse prendre en charge sa sécurité intrinsèque et la sécurité globale dans ses interactions avec d'autres produits ou personnes.

Notre système de gestion de sécurité active comprend premièrement un ensemble de produits actifs interagissant mutuellement et se partageant de l'information, et deuxièmement d'un gestionnaire qui se charge d'initialiser et de rassembler les données provenant de chaque produit actif. C'est donc une architecture mixte où l'approche centralisée traite principalement des aspects administratifs pour gérer globalement la communauté des produits actifs en termes de configuration, d'acceptation de nouveaux produits. Par contre au sein de cette communauté, l'approche est entièrement décentralisée.

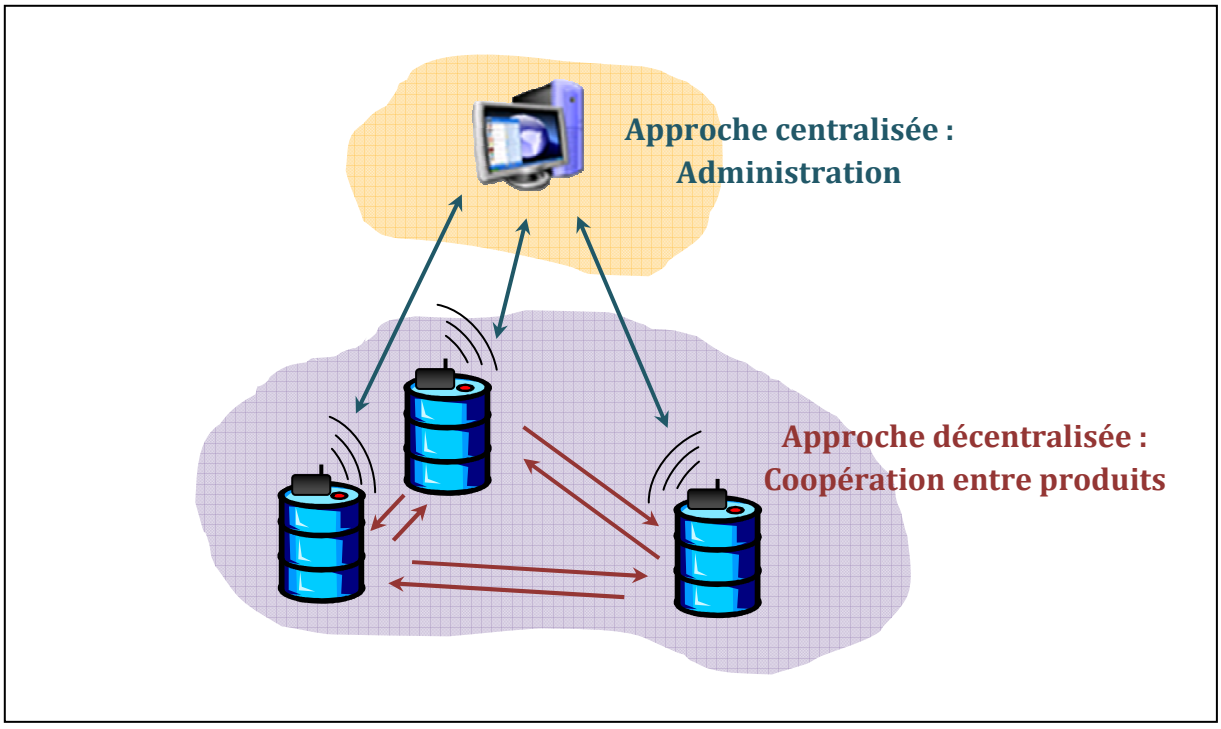

Figure II. 2. Deux approches existantes dans le système de gestion de la sécurité proposé

III. Le modèle interne de produit actif

III.1. Modèle fonctionnel de produit actif

La communauté de recherche a proposé deux modèles de produits intelligents dans le contexte de sécurité des produits.

III.1.1. Modèle de Strohbach

Quant à [Strohbach et al., 2005], il a étudié la compatibilité entre les produits. Pour cela il a développé un modèle de produit intelligent qui est représenté par la figure II.3 :

Dans ce modèle la détection de danger se fait par traitement d'une base de connaissance dans laquelle des règles d'incompatibilité et de proximité sont stockées. (La détection de menace se fait par proximité). Ce modèle ne tient pas compte des règles statiques ambiantes des produits et il ne fait pas une classification de degré de dangerosité.

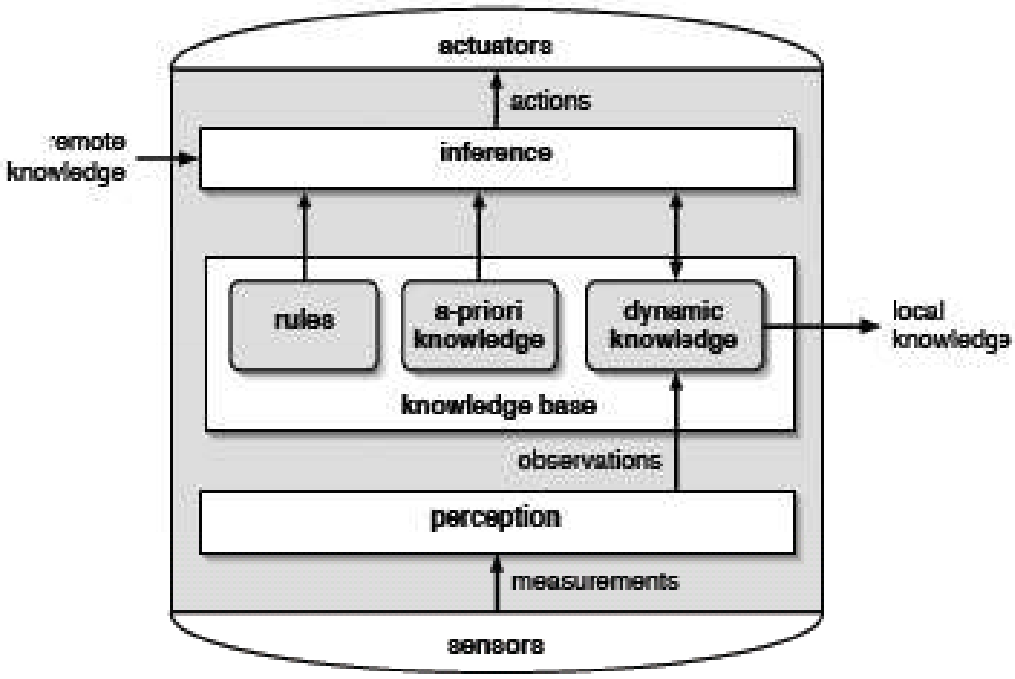

Figure II. 3. Modèle du produit selon [Strohbach et al., 2005]

Les mesures de capteurs sont traitées par le composant « perception » qui associe à ces données une signification produisant la connaissance d'observation qui est significative en termes de domaine d'application. Par exemple, un produit peut être identifié si d'autres produits sont dans la proximité.

Des observations sont stockées et maintenues dans une base de connaissance « Knowledge base » qui reflète les connaissances actuelles de l'objet sur son environnement.

Le composant « Inférence » englobe la connaissance en tenant compte des objets à proximité et l'action nécessaire à prendre par le produit en utilisant les actionneurs (Actuators).

D'autre part le mécanisme de coopération entre produits est décentralisé. En effet les produits partagent la connaissance. Par conséquent, si une menace est détectée les produits déclenchent une alarme comme le montre la figure II.4.

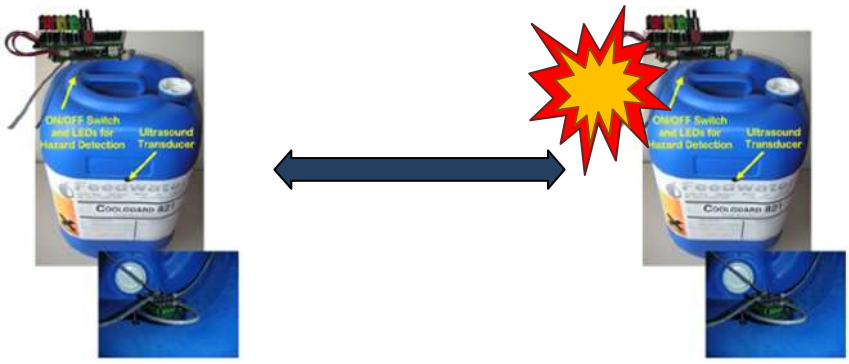

Figure II. 4. Illustration de la coopération selon [Strohbach et al., 2005]

III.1.2. Modèle de Quanz

Le deuxième modèle a été introduit par [Quanz and Tsatsoulis, 2008]. Ce dernier a représenté chaque produit par un agent « an object safety agent ». L'agent traite les informations de l'environnement à partir des capteurs, selon une formule appelée situation.

Le modèle spécifié par [Quanz and Tsatsoulis, 2008] est représenté par la figure suivante :

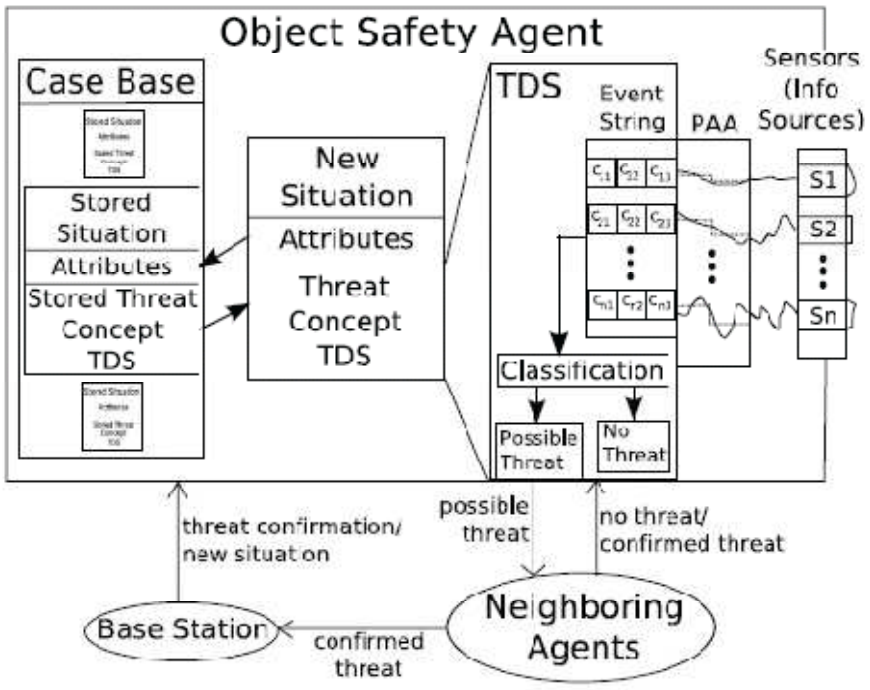

Figure II. 5. Modèle du produit selon [Quanz and Tsatsoulis, 2008]

Les informations capteurs sont traitées par un système de détection de menace « Threat Detection System (TDS) ». Quand un agent détecte une menace, il propage les informations capteurs à ses voisins, pour les tester avec leurs propres systèmes de détection de menaces (TDS). Quand un taux de détections positives est atteint, une confirmation de la nouvelle situation est envoyée par la station de base vers l'agent qui détecte la menace.

Une situation est représentée par des attributs qui décrivent l'environnement et le comportement normal de l'agent dans cette situation.

La notion de menace dans [Quanz and Tsatsoulis, 2008] ne tient pas compte des règles de communauté (compatibilité, ….) ni de règles dynamiques (évolution de situation en fonction du temps), car le dialogue entre les agents se fait seulement par les données de capteurs. En plus il ne fait pas une classification de degré de dangerosité.

La confirmation de menace se fait finalement par confirmation d'une station de base. La station de base va donner au produit une nouvelle situation. Donc, le mécanisme de coopération dans [Quanz and Tsatsoulis, 2008] est centralisé et peut être représenté par la figure II.6.

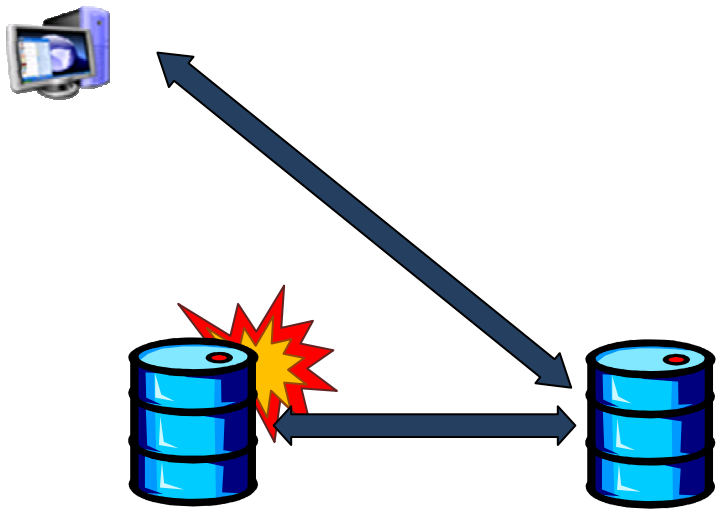

Figure II. 6. Illustration de la coopération selon [Quanz and Tsatsoulis, 2008]

III.1.3. Proposition de modèle de produit actif

Notre modèle fonctionnel proposé du produit actif est représenté dans la figure II.7. Il se compose de plusieurs rubriques. Un bloc de gestionnaire des dispositifs perception qui est chargé comme interface entre la base de connaissance et les capteurs et actionneurs. Une base de connaissances qui transforme les informations perçues par ses propres capteurs et par

celles du voisinage en action de sécurité relativement à ses règles décisionnelles qui sont de trois types : statique (ambiantes), dynamique et communautaire.

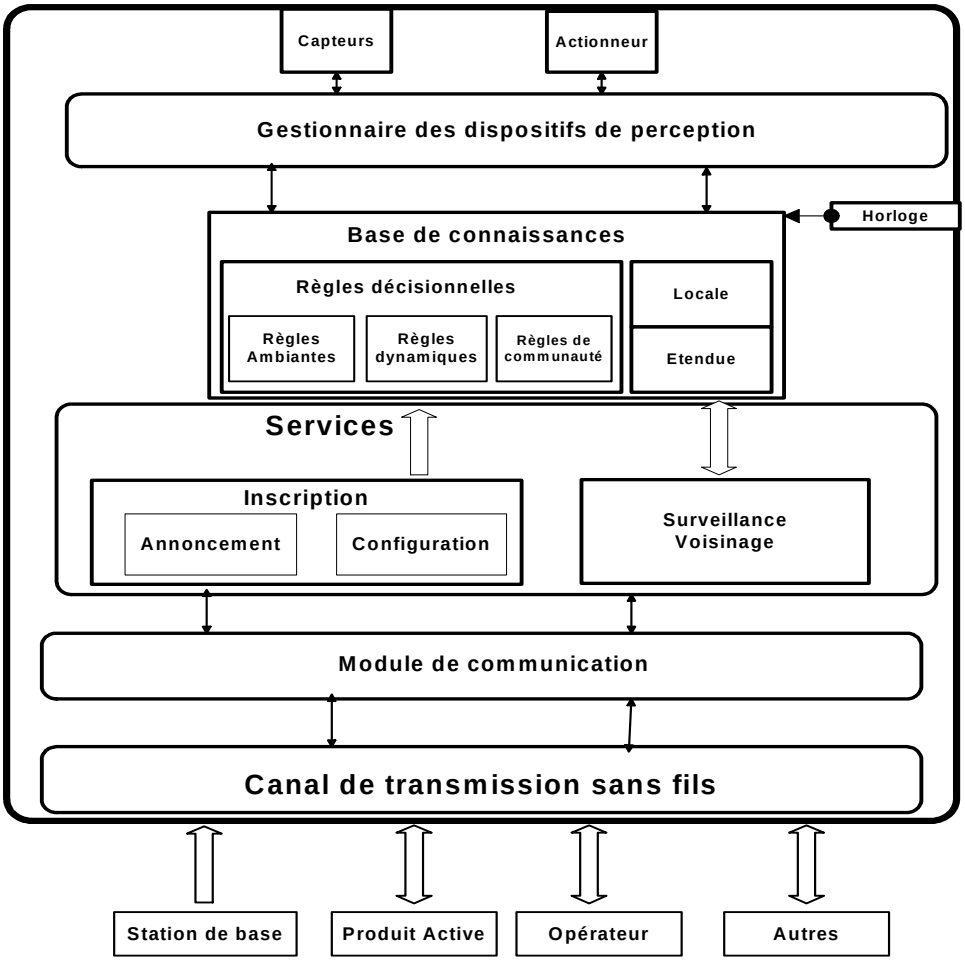

Figure II. 7. Modèle fonctionnel de produit actif

La base de connaissance d'un produit actif représentée par le tableau 3, traite les informations des différents capteurs en se basant sur des règles décisionnelles qui reflètent la situation actuelle de produit.

Un bloc de services dans lequel on trouve les blocs d'inscription qui est lié aux informations de la base de connaissance. Un module de surveillance de voisinage qui collecte les informations des voisinages et les transmet à la base de connaissance pour être traitées.

Notre modèle peut gérer la sécurité de son environnement en coopérant avec son entourage (Produit Actif, Opérateur, Ressources, Autres).

Quand un produit détecte une menace provenant des informations capteurs ou des services offerts, il la classe selon un niveau de dangerosité. Selon la valeur de ce dernier un actionneur est déclenché et une station de base est informée.

A l'entrée d'un produit actif dans la communauté, une station de base offre la situation actuelle de produit, et ensuite il laisse le produit à gérer sa sécurité intrinsèque en coopération avec les autres.

Dotées d'une base de connaissance (règles et symboles de sécurité) et d'une capacité de captage et de décision, le produit actif peut effectuer deux tâches essentielles pour son bien être. La première tâche est la surveillance interne où il est capable de surveiller son voisinage. La violation des règles de sécurité individuelles ou mutuelles doit être détectée et analysée. Un écart entre les variables d'environnement et celles de sa base de connaissance, induit des réactions tel que l'envoi d'un Rapp_D au gestionnaire signalant un état de danger.

Domaine de connaissance	Produit (Type Produit, ID, Symbole)
Règles statiques	D=Température (<LimInf ou >LimSup) B=Température (in [LimInf + Δ, LimSup - Δ]) M=Température (in [LimInf, LimInf + Δ])
Règles dynamiques	Danger=Délai (Mauvais)>Période critique
Règles de communauté	Distance (PA, X) Symbole Produit % Table de compatibilité
Actionneurs	Message= « Incompatible & Distance » ou « M » Alerte= « Incompatible & Distance<Dmin » ou « D »

Tableau 3. Base de connaissance du produit actif

Le caractère X dans la base de connaissance indique un autre produit actif.
La base de connaissance de produit, est composée des faits[16] et des règles. Par exemple un produit devra connaître la substance à laquelle est liée une liste des produits incompatibles.
La deuxième tâche est la surveillance et la communication avec d'autres produits actifs. La communication entre produits actifs se fait par le message de salutation GRE. C'est un message transmis automatiquement et périodiquement entre les produits dans leur état normal de fonctionnement. Il représente un message de salutation (Greeting) portant les informations

[16] La base pour n'importe quelle prise de décision

propre du produit (nom, symboles de sécurité,…), son niveau de sécurité actuel, et a pour rôle ultérieurement de contribuer au processus de calcul de la distance séparant deux produits actifs.

Dès qu'un produit reçoit un message GRE, il va émettre un Message RSI : L'information de ce type de message contient principalement la différence de puissance du signal. Cette technique de localisation s'appelle RSSI (Received Signal Strength Indicator) et se base sur une équivalence entre la valeur de la puissance du signal et la valeur de la distance séparant les deux produits. Elle est utilisée pour estimer la compatibilité en terme de distance minimale entre produits intelligents.

Le mécanisme de coopération intègre deux approches : une centralisée côté administration et une décentralisée coté coopération entre produits actifs.

Les produits actifs coopèrent ensemble par échanges d'informations sur l'environnement et calculent son niveau de sécurité en temps réel, par conséquent un produit actif peut gérer sa sécurité et la sécurité de son environnement.

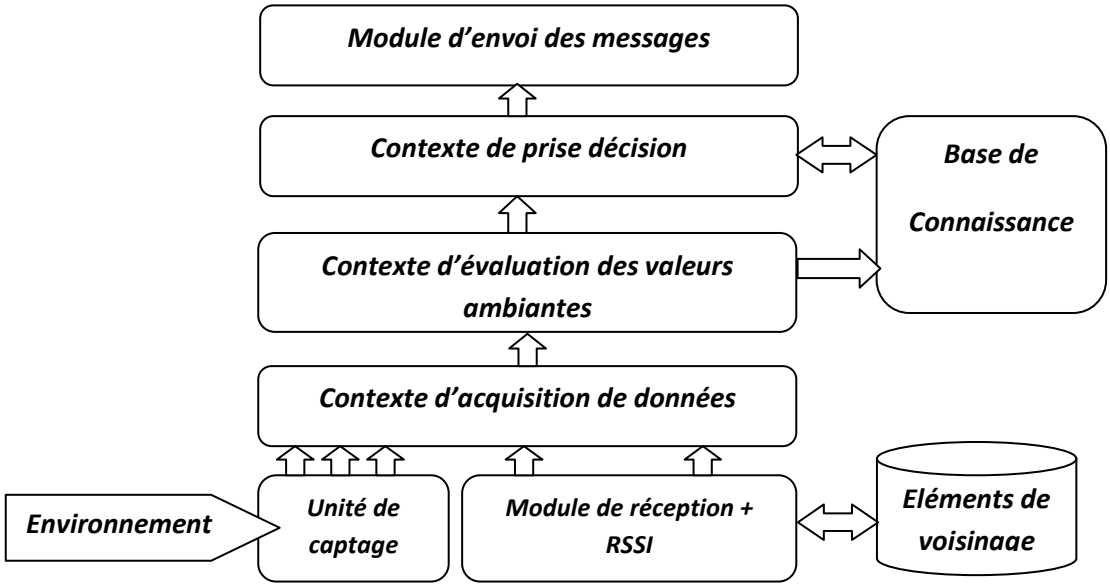

Figure II. 8. Comportement autonome d'un produit actif

III.2. Règles de sécurité

Les règles de sécurité sont basées sur des informations statiques caractérisant le produit de façon intemporelle, des informations dynamiques indiquant les limites du produit en fonction

de la fréquence, la durée de certaines anomalies et les informations communautaires montrant le comportement du produit par rapport à ses voisins.

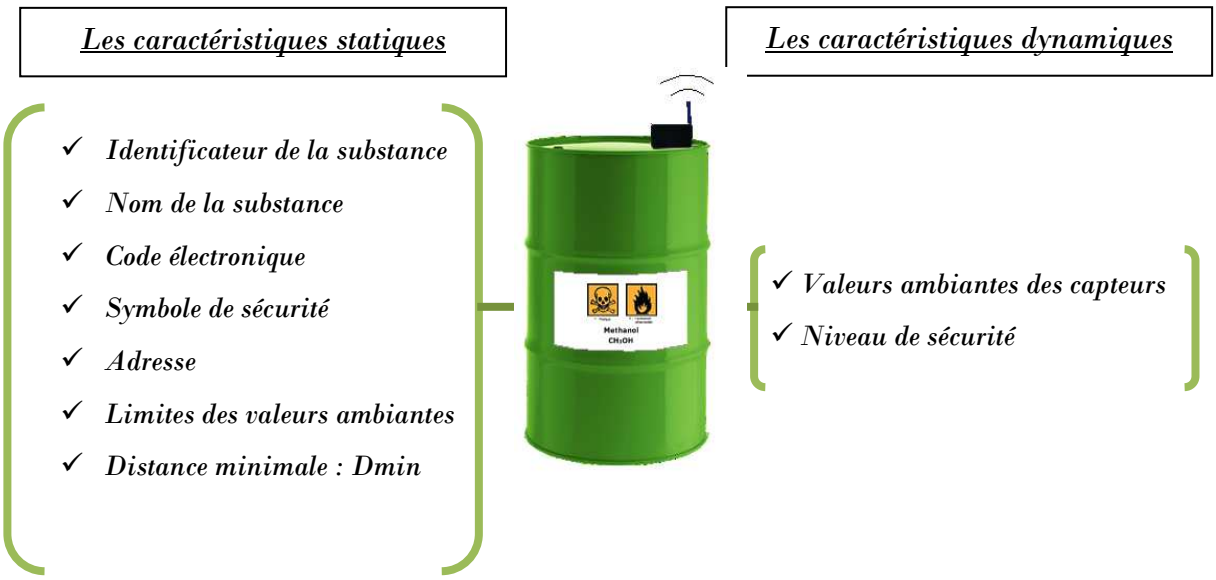

Figure II. 9. Caractéristiques d'un produit actif

III.2.1. Règles statiques

Ces règles sont utilisées pour maintenir le niveau de sécurité intrinsèque de chaque produit. Elles définissent par exemple des limites min ou/et max qui garantissent la pérennité du produit. Cet état est valide tant que ces limites ne se trouvent pas en dehors des valeurs mesurées.

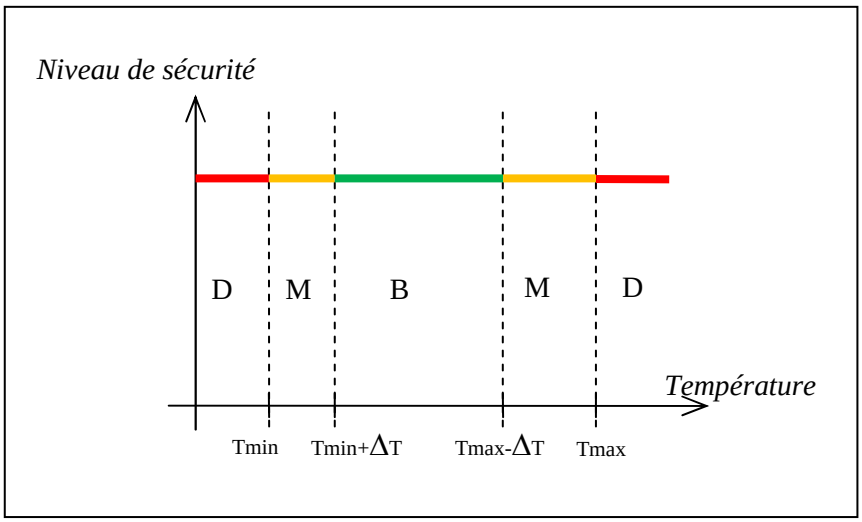

Figure II. 10. Fonction de la règle statique de température

L'histogramme de la figure II.10 donne les caractéristiques de température d'un produit. On distingue trois zones possibles pour la valeur de la température à capter :

- la zone en rouge : c'est la zone de danger. Si la température de la substance contrôlée est dans cette zone alors le produit est dans un état critique. Il doit y avoir un traitement particulier exécuté par ce produit actif (déclenchement d'une alerte vers le superviseur et les autres produits).
- La zone jaune : c'est aussi une zone critique mais moins sensible (état mauvais). Une ou plusieurs valeurs des capteurs sont dans la proximité d'une valeur de restriction (± ΔT de valeur restrictive).
- La zone verte : La valeur mesurée est bonne. Le produit peut continuer à réagir avec son environnement.

Chaque produit actif possède des variables d'environnement (température, lumière, humidité, …). La règle statique exige que ces variables ne doivent pas dépasser des valeurs critiques (min ou max), suite aux contraintes environnementales de produit (caractéristiques chimiques), afin d'éviter des mauvaises réactions. Pour cela, les données de capteurs doivent être mémorisées ensuite comparer à un ensemble de règles propres au produit. Les règles que nous avons définies sont fondées sur un ensemble de limites pour chaque grandeur à mesurer.

i = {Température, Humidité, Lumière, ….}

Chaque valeur de capteur V_{si} : est caractérisée par deux valeurs critiques $V_{si\ min}$ et $V_{si\ max}$ associés à une marge de sécurité ΔV_{si} .

Pour chaque capteur i on évalue son niveau de sécurité S_i qui varie entre 3 états ; S_{Bi} si la valeur de capteur i définit un état bon, S_{Mi} si cette valeur signale un état moyen ou mauvais et S_{Di} si la valeur de capteur indique un d'état dangereux.

D'où

$$S_i = \begin{cases} S_{Bi} \ \ si \ \ V_{si} \in]V_{si\,min} + \Delta V_{si}\ , V_{si\,max} - \Delta V_{si}[\\ S_{Mi} \ \ si \ \ V_{si} \in [V_{si\,min}\ , V_{si\,min} + \Delta V_{si}] \cup [V_{si\,max} - \Delta V_{si}, V_{si\,max}\ ] \\ S_{Di} \ \ si\ V_{si} \in \ ]-\infty, V_{si\,min}\ [\cup]\ V_{si\,max}, +\infty\ [\end{cases} \qquad (2.1)$$

S_{sr} est l'état à en conclure à partir des règles statiques, cet état varie entre bon (B), moyen (M) ou dangereux (D).

$$S_{sr} = \begin{cases} D \;\; si \;\; \exists \;\; i \;\; / \;\; S_i = S_{Di} \\ M \;\; si \;\; \begin{cases} \forall \; i \;\; S_i \neq S_{Di} \\ \exists \;\; i \;\; / \; S_i = S_{Mi} \end{cases} \\ B \;\; si \;\; \forall \; i \begin{cases} S_i \neq S_{Di} \\ S_i \neq S_{Mi} \end{cases} \end{cases} \qquad (2.2)$$

La température possède une limite haute et basse définissant ainsi les intervalles de sécurité. Comme le montre la figure II.11, pour la température, on définit deux états de danger (T<10°C et T>40°C), deux états mauvais ([10°C,20°C] et [30°C,40°C]) et un état bon ([20°C,30°C]). Pour la lumière, on définit un état de danger (>80 cd), un état mauvais ([80 cd,60 cd]) et un état bon (<60 cd). Et enfin, pour l'humidité un état de danger (>70%), un état mauvais (entre 50% et 70%) et un état bon (<50%). La Figure II.11 illustre ces règles.

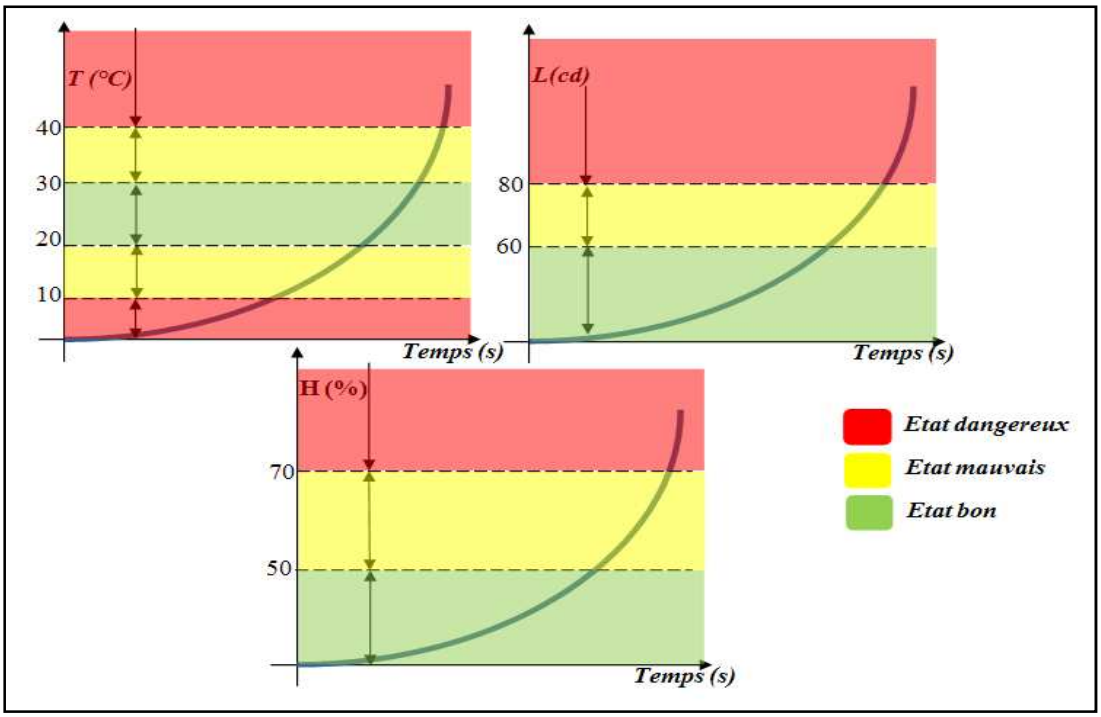

Figure II. 11. Règles statiques

III.2.2. Règles dynamiques

Ces règles sont liées au produit tout en prenant en considération l'évolution de son état. Certains événements ne sont pénalisants pour le produit qu'en fonction de la distribution de leur apparition : une température qui dépasse furtivement un seuil n'est pas forcément critique pour le produit, des chocs non répétés sur un fût peuvent être sans conséquence.

Ces règles ont pour but de valoriser la dimension temporelle, inexistante dans les règles statiques. Car, si un état mauvais persiste pour une période considérable, cet état doit être signalé comme état dangereux.

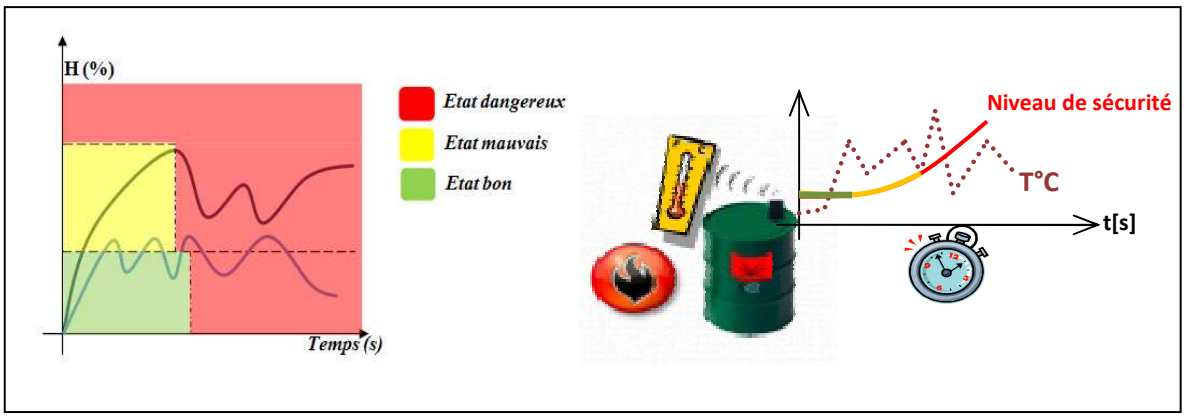

Figure II. 12. Illustration des règles dynamiques

Par exemple, si une variable d'environnement la température (température, lumière, humidité, …) atteint l'intervalle de l'état mauvais, comme l'indique la figure II.12, et y reste pendant une période T_{cr} , un état dangereux doit être signalé. Aussi si un basculement entre état bon et état mauvais, un compteur doit être présent pour signaler un état de danger si le nombre de basculements dépasse une limite d'occurrence n_C.

D'où les règles dynamiques S_{dr} peuvent conclure à un état dangereux décrit par :

$$S_{dr} = D \ \ si \begin{cases} \exists\, t_1 \,/\, \forall\, t \in [t_1, t_1 + T_{cr}] \quad S_{sr}(t) = M \\ Ou \\ Occur\big(S_{sr}(t)\big) \geq n_C \end{cases} \qquad (2.3)$$

Où T_{cr} est une période critique fixée selon le produit à ne pas dépasser dans un état moyen (Mauvais) et n_c est le nombre de basculement de S_{sr} autorisé entre les états B et M.
Avec Occur est la fonction qui s'incrémente lorsque S_{sr} passe de l'état B à l'état M.

III.2.3. Règles communautaires

Ces règles implantées dans le produit concernent les contraintes de compatibilité avec d'autres règles de produits de même nature ou non. Elle s'appuie sur des matrices

d'incompatibilité et repose sur les symboles de sécurité et les directives européennes 67/548/EEC[17].

Chaque substance chimique peut avoir des réactions dangereuses face à d'autres substances. Le produit actif doit être capable de surveiller ces incompatibilités en fonction d'une base de connaissance contenant des fiches techniques.

Dans un environnement de sécurité, Ainsi, un produit actif envoie des informations sur la substance stockée à l'intérieur du fût comme l'identificateur l'ID de la substance et les symboles de sécurité. Un autre produit actif reçoit ces informations puis calcule la valeur de RSSI (Received Signal Strength Indicator) du message et renvoie cette valeur avec son symbole de sécurité. Le premier produit actif reçoit ce message, lit la valeur de RSSI, calcule la distance, et vérifie s'il est compatible avec le deuxième produit actif. Enfin le niveau de sécurité est calculé.

Selon leurs caractéristiques chimiques, certains produits peuvent avoir des contraintes de compatibilité avec d'autres produits de natures différentes selon la matrice de compatibilité entre produits chimiques dangereux comme l'indique la figure II.13. D'où la nécessité d'une procédure qui vise à déterminer le niveau de compatibilité entre produits stockés dans le même entrepôt. En revanche, la compatibilité de quelques produits est proportionnelle à la distance qui les sépare.

De même les règles de communauté S_{cr} peuvent en conclure un des trois états cités précédemment par :

$$S_{cr} = \begin{cases} D \quad si \quad \begin{cases} \exists\, S_{yi} \in \{symboles\ de\ produit\ i\} \\ \exists\, S_{yj} \in \{symboles\ de\ produit\ j\} \end{cases} / \begin{cases} F_{Comp}(S_{yi}, S_{yj}) = \ Incompatible \\ D(produit\ i, produit\ j) < D_{min} \end{cases} \\ M \quad si \quad \begin{cases} \exists\, S_{yi} \in \{symboles\ de\ produit\ i\} \\ \exists\ \ S_{yj} \in \{symboles\ de\ produit\ j\} \end{cases} / \begin{cases} F_{Comp}(S_{yi}, S_{yj}) = \ Incompatible \\ D(produit\ i, produit\ j) \in [D_{min}\ , D_{min} + \Delta D] \end{cases} \\ B \quad si \quad \begin{cases} \begin{cases} \exists\, S_{yi} \in \{symboles\ de\ produit\ i\} \\ \exists\, S_{yj} \in \{symboles\ de\ produit\ j\} \end{cases} / \begin{cases} F_{Comp}(S_{yi}, S_{yj}) = \ Incompatible \\ D(produit\ i, produit\ j) > D_{min} + \Delta D \end{cases} \\ \qquad\qquad Ou \\ \begin{cases} \exists\, S_{yi} \in \{symboles\ de\ produit\ i\} \\ \forall\ \ S_{yj} \in \{symboles\ de\ produit\ j\} \end{cases} / F_{Comp}(S_{yi}, S_{yj}) = \ Compatible \end{cases} \end{cases} \qquad (2.4)$$

Où :

$F_{Comp}(S_{yi}, S_{yj})$: est la fonction qui étudie la compatibilité entre les symboles de sécurité de deux produits i et j.

$D(produit\ i, produit\ j)$: est la distance qui sépare ces deux produits.

[17] Directive 2006/121/CE du Parlement européen et du Conseil, Journal officiel de l'Union européenne

D_{min} : est une distance critique à respecter lors d'incompatibilité entre produits.

ΔD : est une marge de distance fixée par la nature de produit.

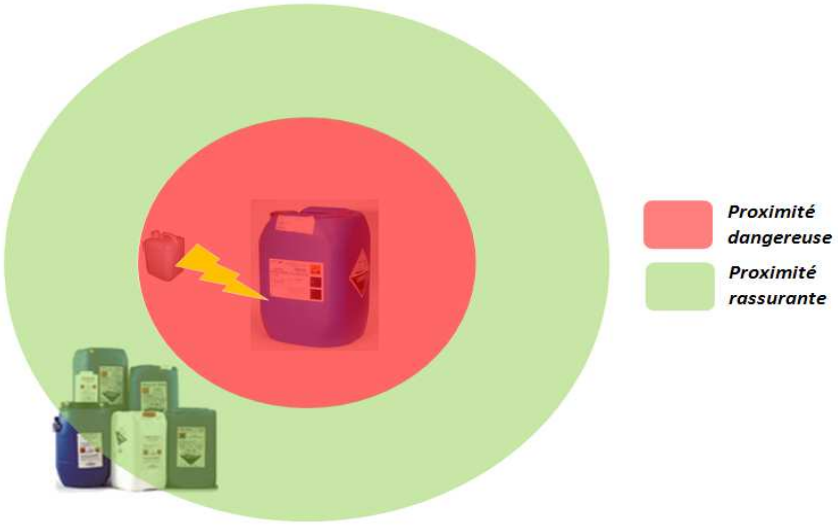

Figure II. 13. Règles de communauté : proximité d'incompatibilité

Dans ce qui suit, nous définissons les règles de sécurité comme étant les variables critiques ambiantes propres au produit (mentionnées dans les règles statiques et dynamiques) et nous définissons aussi les symboles de sécurité comme l'ensemble des produits chimiques qui présentent une menace d'incompatibilité complète ou partielle au produit en question.

Chaque produit intelligent (niveau inférieur), doit demander au gestionnaire (niveau supérieur), ses propres règles et symboles de sécurité après avoir été identifié par ce dernier. Ces règles et ces symboles illustrent la base de connaissance qui aide le produit à prendre des décisions à partir des valeurs des variables d'environnement.

III.2.4. Calcul du niveau de sécurité global

Chaque produit actif détermine son niveau de sécurité de chaque règle, mais il est nécessaire pour apprécier globalement le niveau de sécurité du produit d'avoir un indicateur de sécurité général. Si on suppose que les états de sécurité sont classés selon l'ordre croissant suivant : Bon, Mauvais et Danger. Nous proposons que ce niveau de sécurité global soit calculé en prenant le maximum de tous les niveaux de sécurité et en appliquant la formule suivante :

Niveau de sécurité global = Max (niveau de sécurité déduit de la règle de communauté, niveau de sécurité déduit de la règle statique, niveau de sécurité déduit de la règle dynamique)

Enfin après avoir recueilli les états de chaque règle (statique, dynamique et communauté) il est nécessaire de formaliser un état global S_G qui décrit la situation absolue du produit.

$S_G = f(S_{sr}, S_{dr}, S_{cr})$ (2.5)

Soit $\alpha \in \{sr, dr, cr\}$

$$S_G = \begin{cases} D \ si \ \exists \ \alpha \ / S_\alpha = D \\ M \ si \ \forall \ \alpha \begin{cases} S_\alpha \neq D \\ \exists \ \alpha / S_\alpha = M \end{cases} \\ B \ si \ \forall \ \alpha \begin{cases} S_\alpha \neq D \\ S_\alpha \neq M \end{cases} \end{cases} \quad (2.6)$$

IV. Comportement du produit actif

Pour passer vers l'état Actif le produit doit subir une stratégie lui permettant de s'introduire correctement afin de gérer sa propre sécurité active dans un entrepôt [Zouinkhi et al., 2008]. Il passe obligatoirement dans des états précis à savoir l'inscription, la configuration, la surveillance et la communication et finalement la surveillance interne :

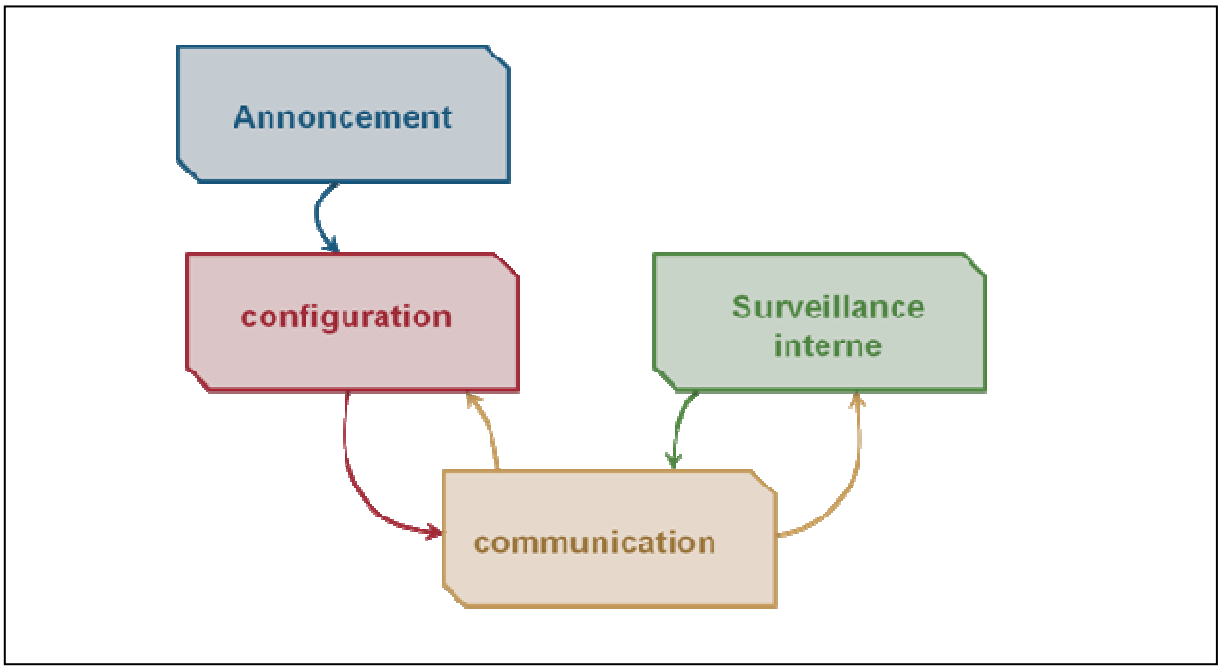

Figure II. 14. Les états d'un Produit Actif

La stratégie proposée comporte les différentes étapes mentionnées ci dessus. Dans le but d'implémenter le modèle interne des produits actifs, il est nécessaire de définir les structures des paquets utilisées pour les échanges de données. La section suivante décrit ces paquets.

IV.1. La structure des paquets transmis

Afin de gérer la sécurité du système, le produit actif échange les informations via des messages. Les messages envoyés doivent être courts et doivent contenir uniquement les informations pertinentes pour une certaine situation.

Le protocole de communication proposé a été conçu à partir de celui développé par TecO et utilise le formalisme de communication par tuples (ConCom). Les messages doivent permettre de transporter le plus grand nombre d'informations pertinentes dans des délais les plus courts de façon à accommoder le plus efficacement possible le comportement interne du produit. Cet impératif oblige le Produit Actif à répondre le plus rapidement possible aux changements de l'environnement et à agir immédiatement pour délivrer les alertes.

En référence au formalisme de communication par tuples (ConCom) [Krohn et al., 2004], nous décrivons dans la figure II.16, la structure générique du paquet. Les paquets correspondants aux messages du modèle vont être encapsulés dans ce paquet. L'entête de ce paquet est défini dans la figure II.15 :

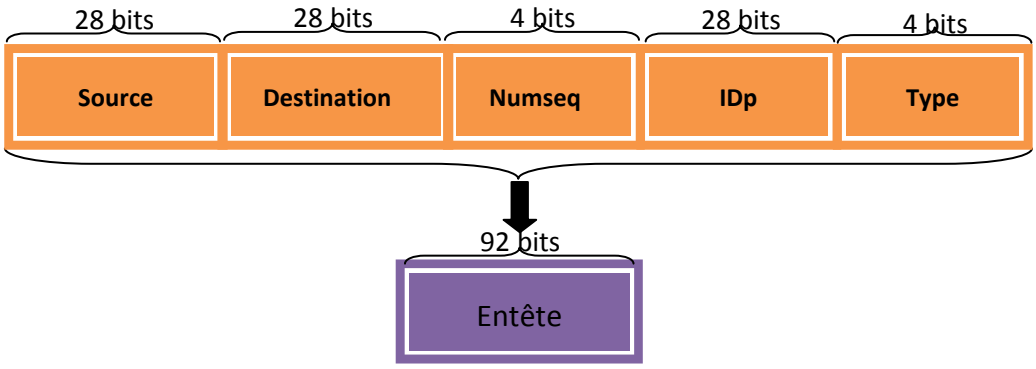

Figure II. 15. La structure de l'entête du paquet générique.

Avec :

Source : l'adresse du nœud source

Destination : l'adresse du nœud destination

NumSeq : numéro de séquence du paquet

IDp : identificateur de l'application

Type : type du message

La structure du paquet générique est donnée par la figure II.16.

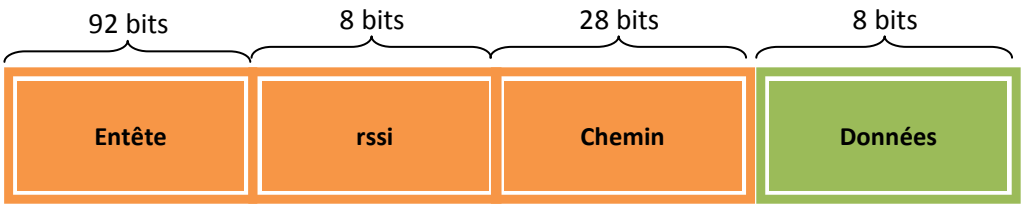

Figure II. 16. La structure du paquet générique

Avec

Entête : c'est l'entête du paquet

rssi : indicateur de la puissance du signal du paquet reçu (Received Signal Strength Indicator)

Chemin : le chemin courant à partir de la source

Données : le contenu du paquet appartenant au modèle interne de produit actif

IV.2. Les messages échangés

Le tableau 4 illustre les différentes natures des messages émis ainsi que leurs désignations. Ces messages sont présentés en couple requête/réponse.

Requête		Réponse	
Nom	**Désignation**	**Nom**	**Désignation**
CTR	Demande d'inscription d'un produit actif	Ack_CTR	Acquittement du gestionnaire
NCF0	Demande suite à un manque des règles et des symboles de sécurité	CMD1+CMD3	
NCF1	Demande suite à un manque des règles de sécurité	CMD3	Règles de sécurité appropriées envoyé par le gestionnaire
NCF2	Demande suite à un manque des symboles de sécurité	CMD1	Symboles de sécurité appropriée envoyé par le gestionnaire
GRE	Salutation entre produits actifs	RSI	Puissance de signal reçu équivalent à la distance qui sépare les Produits actifs
CMD2	Demande de gestionnaire sur l'état de configuration d'un produit actif	CFG	Réponse d'un produit actif annonçant sa configuration
CMD4	Demande de gestionnaire des règles de sécurité d'un produit actif	SER	Réponse d'un produit actif annonçant ses règles de sécurité
CMD5	Demande de gestionnaire des variables ambiantes d'un produit actif	INA	Réponse d'un produit actif contenant ses variables ambiantes
Rapp_D	Rapport état danger vers le gestionnaire	Ack_Rapp_D	Acquittement sur état Danger renvoyé par le gestionnaire
Rapp_M	Rapport état mauvais vers le gestionnaire	Ack_Rapp_M	Acquittement sur état mauvais renvoyé par le gestionnaire

Tableau 4. Ensemble de messages échangés requêtes/réponses

IV.3. Annoncement du produit dans la communauté

L'étape d'inscription est décisive pour qu'un nouveau produit s'introduise dans une communauté. L'inscription se fait chez le gestionnaire pour que ce dernier le détecte et l'ajoute dans sa liste des produits déjà existants. Le produit transmettra continuellement le message de découverte (CTR) jusqu'à ce qu'elle reçoive un acquittement du superviseur (AckCTR) comme le représente le diagramme de séquence de la figure II.19. Le message est formé d'un seul tuple qui n'a aucune information supplémentaire.

Pour aboutir à cette action destinée à inscrire le produit dans la communauté, on a proposé deux types de messages :

- message CTR qui est un message vide. Il est envoyé continuellement au superviseur jusqu'à la réception d'un acquittement.

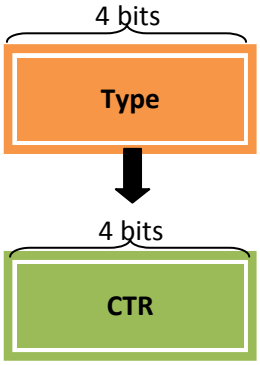

Figure II. 17. La structure du message CTR

- message **AckCTR** qui représente l'acquittement du gestionnaire après la réception du **CTR**. De même, ce message ne contient pas d'information. Le champ **type** prend alors dans ce cas la valeur **AckCTR.**

-

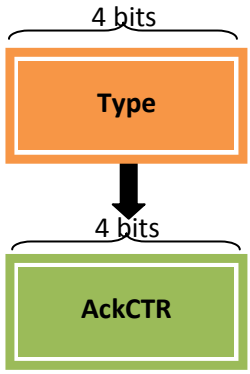

Figure II. 18. La structure du message AckCTR

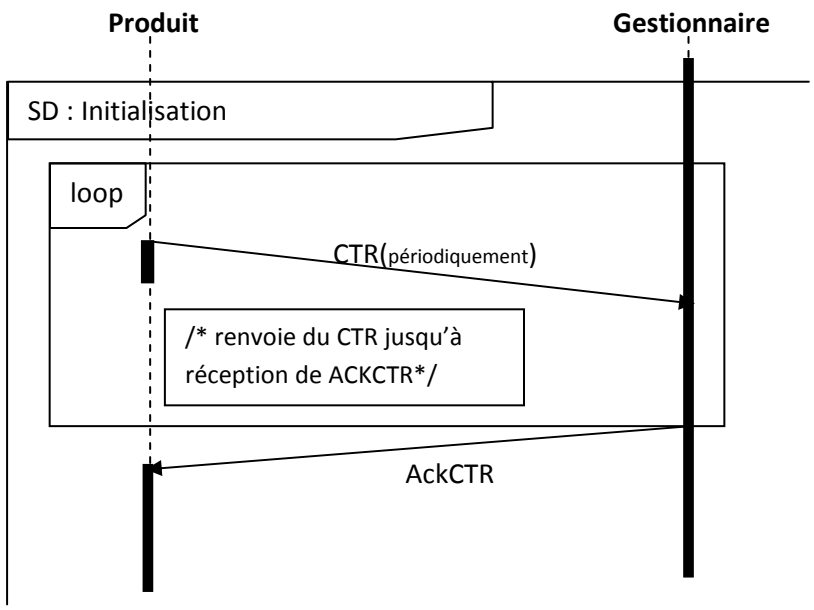

Figure II. 19. Diagramme de séquence de l'annoncement d'un produit Actif

IV.4. Configuration

Chaque élément de la communauté doit avoir une liste de paramètres pour qu'il soit identifié. Cette liste se compose de deux groupes : les paramètres de configuration générale et les règles de sécurité. Pour cela, un produit actif peut se trouver dans trois cas possibles :

Premier cas :

Il n'admet aucune configuration préinstallée. Dans ce cas, il demande l'installation d'une configuration complète par envoi d'un message **NCF0**.

Deuxième cas :

Il a déjà les règles de sécurité mais il reste à demander les paramètres de configuration générale. Et cela se fait par envoi du message **NCF1**.

Troisième cas :

Dans ce dernier cas, le produit n'admet que les paramètres de configuration générale. Donc il envoie un message **NCF2** pour qu'il reçoive les règles de sécurité correspondantes.
Dans ces trois messages, le champ **type** dans l'entête prend la valeur **NCF** et la séquence **données** se compose d'un seul champ : le **type** prend la valeur **NCF0**, **NCF1** ou **NCF2**.

Ces derniers seront envoyés périodiquement au superviseur. De son côté, ce dernier répond par les messages de commande de type configuration respectifs :

- **CMD1** : message supportant la configuration de la classification du produit. Ce paquet comporte les champs suivants :

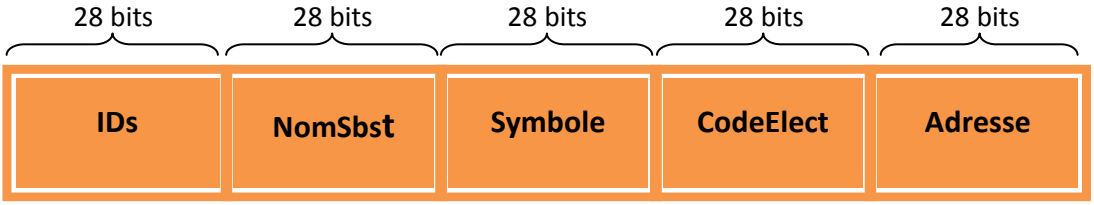

Figure II. 20. Structure du paquet CMD1

Avec :

IDs : l'identificateur de la substance

nomSbst : le nom de substance

Symbole : le symbole de sécurité de la substance, avec la valeur de ce champ, on détermine la compatibilité

CodeElect : le code électronique du produit.

Adresse : l'adresse du produit

- **CMD3** : message supportant la configuration des différentes règles de sécurité.

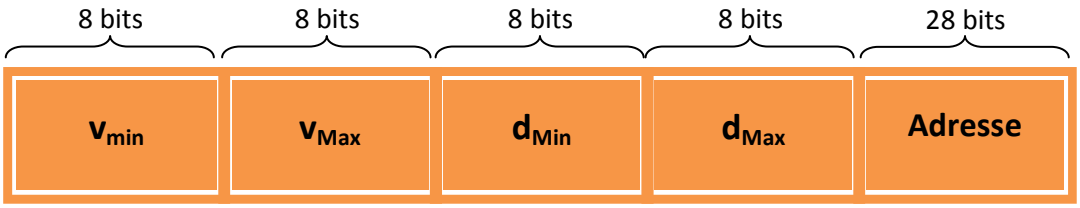

Figure II. 21. Structure du paquet CMD3

Le produit devient actif lorsqu'il reçoit la réponse convenable à son message NCF. A ce moment, il commence à exécuter ses rôles dans la communauté (communication et surveillance).

La figure II.22 modélise le cas où le produit n'est pas configuré et n'a pas les règles de sécurité. Donc il envoie au superviseur un message NCF0 et il attend jusqu'à l'arrivée des messages CMD1 et CMD3 pour qu'il devienne actif, puis il entre en communication avec les autres produits actifs.

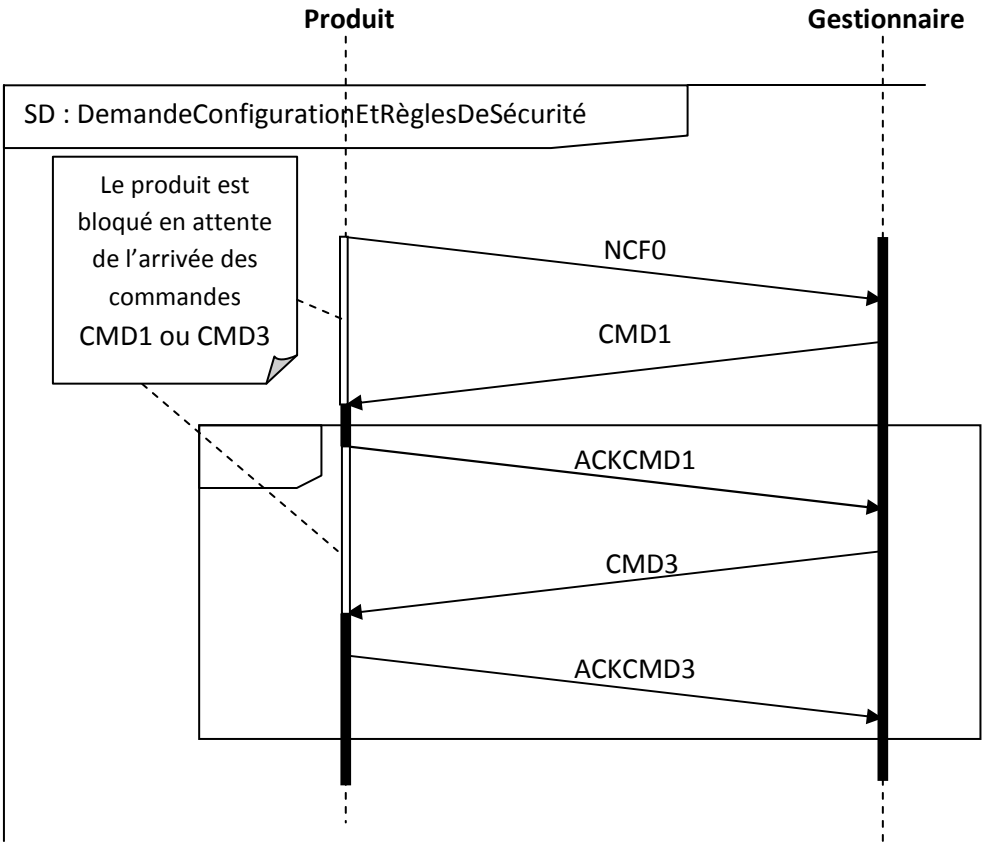

Figure II. 22. Diagramme de séquence de demande configuration et des règles de sécurité

Le produit actif peut avoir les règles de sécurité avant qu'il soit actif. Dans ce cas, il est sensé de ne demander que les paramètres de configuration (nom de la substance, code électronique du produit, symbole, identificateur de la substance, adresse). La figure II.23 schématise ce cas.

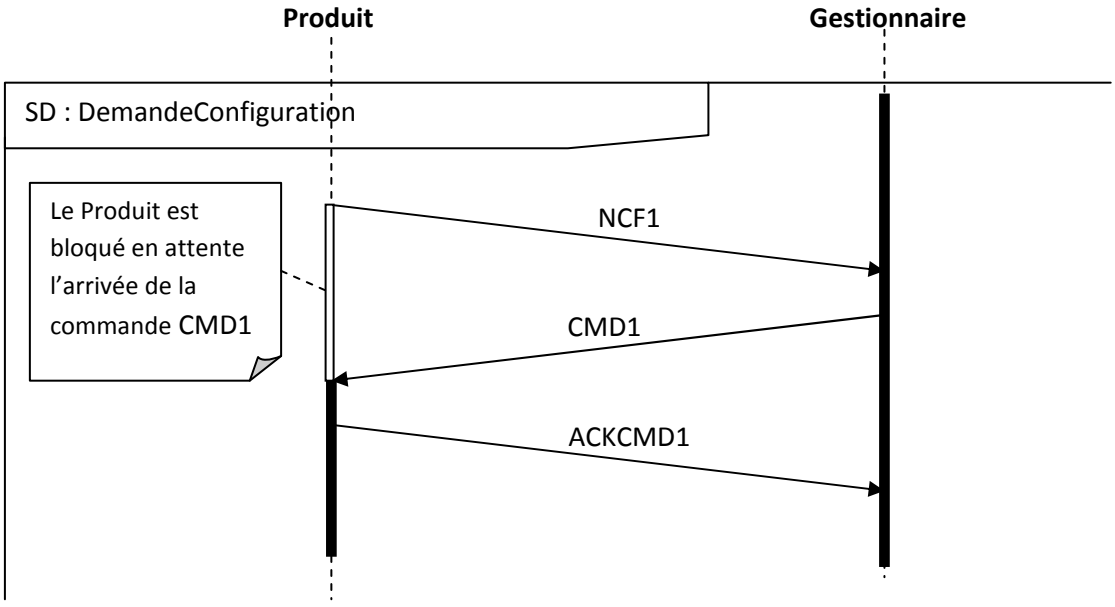

Figure II. 23. Scénario de demande de configuration

De la même façon, le produit peut être configuré avant qu'il soit actif. Alors, il ne demande que les règles de sécurité (les règles internes, les règles d'interaction, l'adresse). La figure II.24 schématise ce cas.

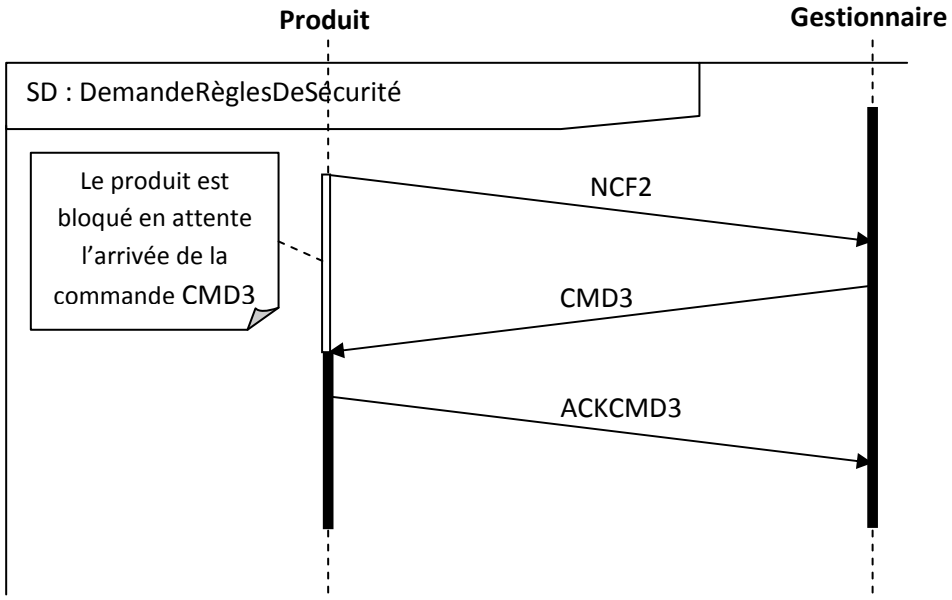

Figure II. 24. Scénario de demande de règles de sécurité

IV.5. Surveillance et Communication

Le produit actif, doit interagir avec son environnement. Il doit être sensible à toutes modifications de son environnement violant les règles de sécurité individuelles ou réciproques, pour qu'il réagisse en temps réel en transmettant ces anomalies vers le superviseur.

Ces interactions s'effectuent par le biais des messages suivants :

- ***Message de salutation GRE*** : Ce message est transmis périodiquement entre les produits déjà configurés. C'est un message de salutation (GREeting) portant les informations propres du produit (identificateur, symboles de sécurité,…), son niveau de sécurité actuel, et a pour rôle ultérieurement de contribuer au processus de calcul de la distance séparant deux produits actifs. Le message de salutation est important pour la surveillance et la communication entre produits parce qu'il notifie au produit actif ses caractéristiques principales.

La structure du message de salutation est donnée par la figure II.25.

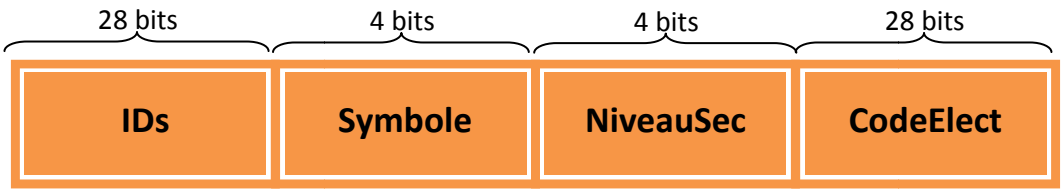

Figure II. 25. Structure du paquet GRE

- ***Message RSI*** : L'information de ce type de message contient principalement la différence de puissance du signal du message GRE reçu. La valeur de la RSSI (n'admettant pas d'unité) est équivalente à celle de la distance. Pour cela elle est utilisée pour estimer la compatibilité avec la distance minimale entre produits actifs. La figure II.26 schématise cette dépendance par une courbe quasi linéaire.

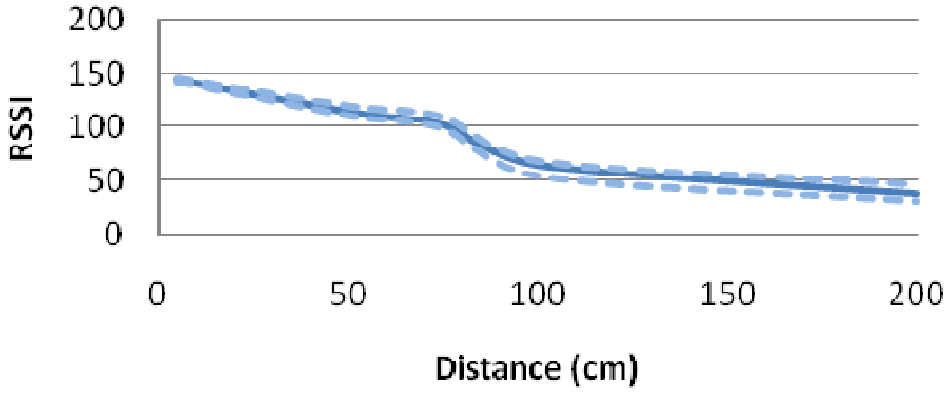

Figure II. 26. Equivalence RSSI – Distance [Zouinkhi et al, 2007]

Après avoir reçu un message GRE, le produit actif effectue donc son calcul RSSI (Received Signal Strength Indicator) de différence de puissance du signal entre les deux produits émetteur et récepteur, puis envoie le résultat à travers le message RSI comme le montre le diagramme de séquence de la figure II.28. Le premier produit actif recevant ce message, lit la valeur de RSSI, vérifie si la valeur de la distance a dépassé la valeur acceptable ou non.
Voici la structure du message RSI suivant le formalisme ConCom.

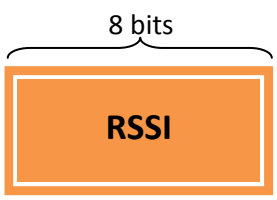

Figure II. 27. Structure du message RSI

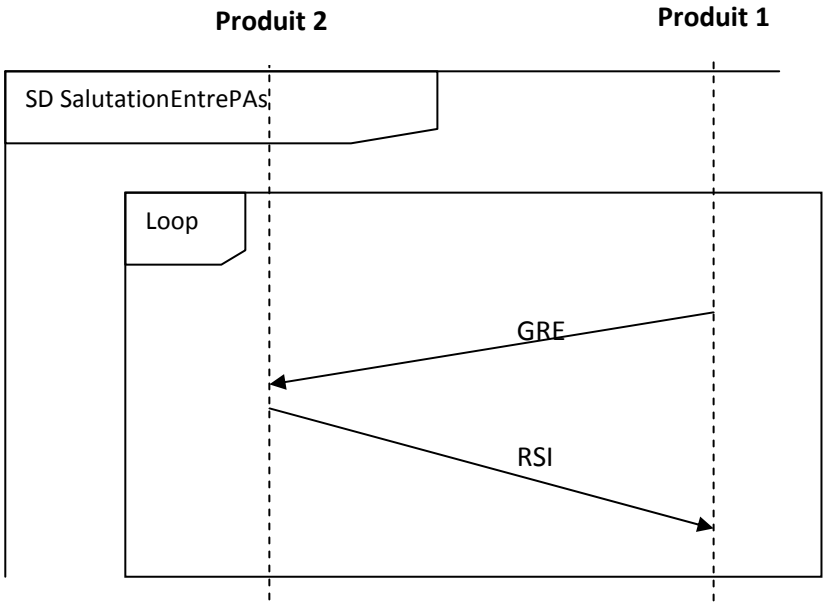

Figure II. 28. Scénario de salutation entre deux produits

- ***Message des valeurs ambiantes INA*** : ce message contient les valeurs ambiantes mesurées par les capteurs du produit actif. Le diagramme de séquence de la figure II.30 montre comment ce message est envoyé par le produit actif après avoir reçu une requête depuis le superviseur (gestionnaire) pour lui fournir ces informations.

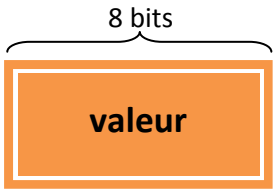

Figure II. 29. Structure du message INA

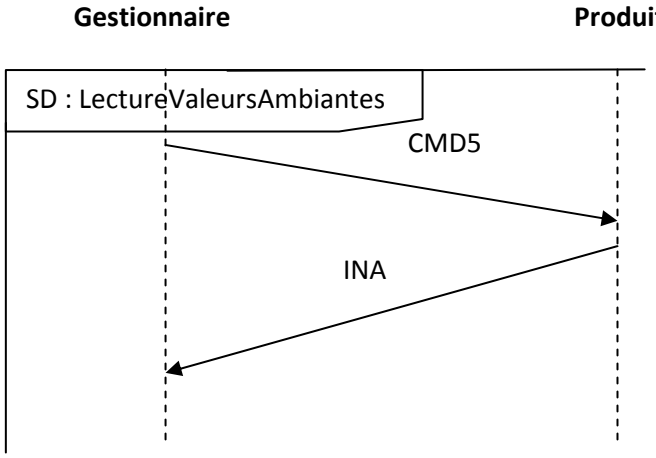

Figure II. 30. Scénario de demande de lecture des valeurs ambiantes (pour le gestionnaire)

La figure II.30, illustre une commande particulière (CMD5). Elle est utilisée par le superviseur s'il a besoin de lire les dernières valeurs ambiantes détectées par le capteur.

- ***Message CFG*** : il comporte la configuration spécifique au produit actif. Ce message est émis par ce dernier après une requête du gestionnaire.

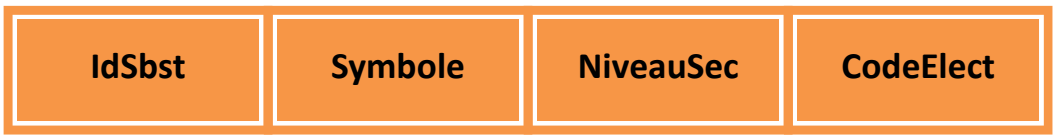

Figure II. 31. Structure du paquet CFG

Dans le scénario de la figure II.32, le superviseur envoie la commande CMD2 pour lire les paramètres de configuration d'un produit actif particulier.

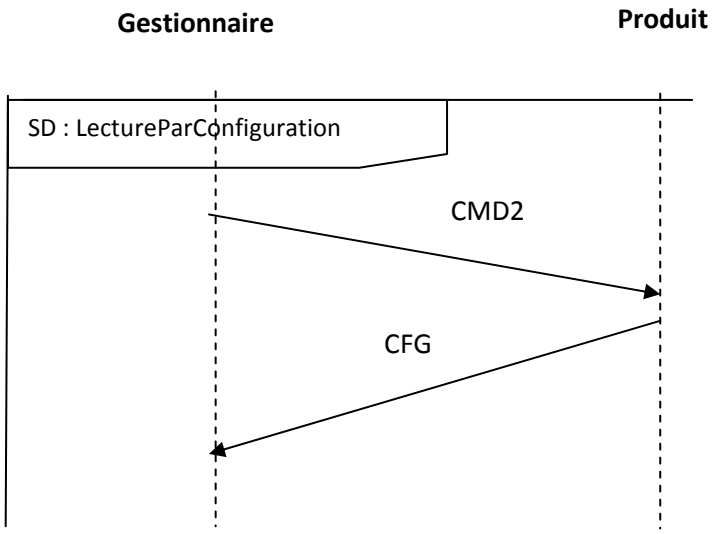

Figure II. 32. Scénario de lecture des paramètres de configuration

- ***Message SER*** : ce message comprend les configurations des règles de sécurité. Il est aussi envoyé après une demande du gestionnaire. Ce message admet la même structure que le message CMD3.

Le superviseur peut aussi lire les règles de sécurité d'un produit actif en envoyant une commande CMD4. Le produit répond par le message SER comme schématisé sur la figure II.33.

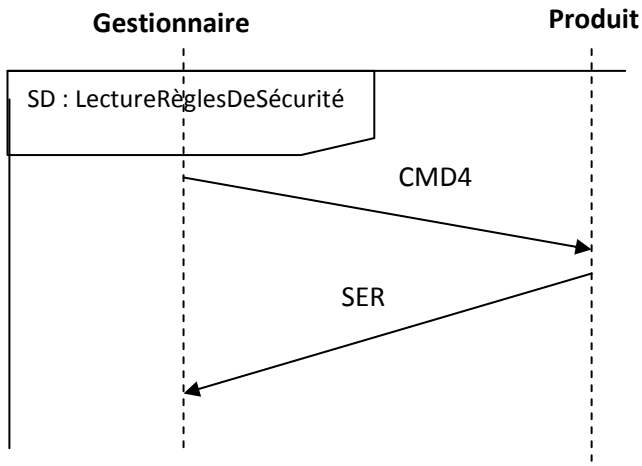

Figure II. 33. Scénario de demande lecture des règles de sécurité

Le superviseur peut demander à des produits la lecture de certains paramètres tels que les paramètres de configuration ou les valeurs ambiantes lues par les capteurs associés. Les commandes utilisées pour lancer ces demandes sont :

- ***Message CMD2*** : le gestionnaire requiert la configuration du produit actif à travers ce message.
- ***Message CMD4*** : de même, les règles de sécurité sont demandées via ce message.

Le paquet de message CMD4 est vide.

- ***Message CMD5*** : Avec ce message le gestionnaire collecte les diverses informations ambiantes spécifiques au produit. Le paquet correspondant à ce message est aussi vide.

IV.6. Surveillance interne

La surveillance est la procédure où le produit actif applique ses règles ambiantes et commence à traiter les mesures générées depuis sa plateforme pour en décider finalement le niveau de sécurité courant. Les messages qui peuvent émettre dans ce cas sont les messages d'alerte annonçant un état de sécurité menaçant et nécessitant une intervention immédiate.

Messages d'alerte ALE : il est envoyé en cas d'alerte. Ce message reporte au superviseur l'état de sécurité défaillant avec les mesures qui ont provoqué cet état. Dans notre modèle nous avons subdivisé cette situation d'alerte en deux catégories avec deux types de messages

- **ALER** généré s'il y a incompatibilité entre produits actifs proches (suivant la valeur du RSSI).

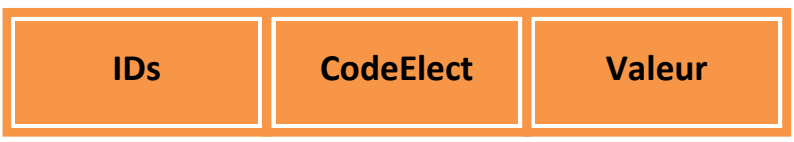

Figure II. 34. Structure du paquet ALER

IV.6.1. Le scénario de déclenchement d'une alerte de dépassement d'une règle d'interaction

La figure II.35, illustre un deuxième cas possible d'alerte. Deux produits appartenant à la même communauté subissent des règles d'interaction entre elles. Si la distance entre deux produits dépasse une limite critique, le produit qui a détectée le premier ce dépassement déclenche une alerte.

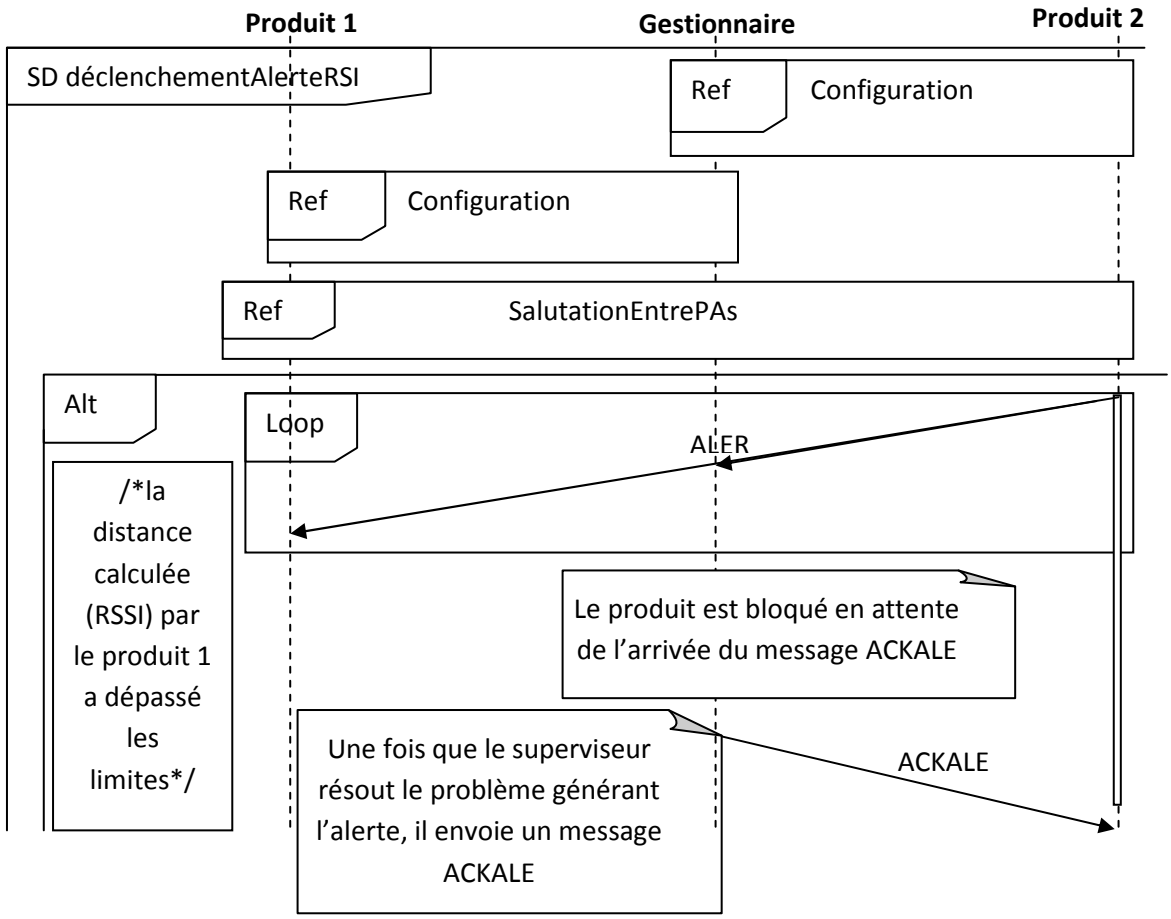

Figure II. 35. Scénario de déclenchement d'une alerte après dépassement d'une règle d'interaction

- **ALEV** reporte l'état déficient des mesures des valeurs ambiantes.

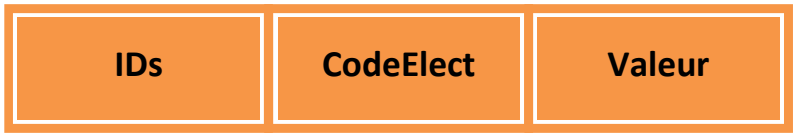

Figure II. 36. Structure du paquet ALEV

IV.6.2. Le scénario de déclenchement d'une alerte de dépassement d'une règle interne

A part le fonctionnement normal du système, il y a des cas exceptionnels où le produit se trouve dans un état de dysfonctionnement. Dans ce cas, le produit doit envoyer un message d'alerte ALE en diffusion afin d'avertir les autres qu'il y a un problème dans ce produit. La figure II.37, présente un exemple de cet état : un nœud de capteur détecte un dépassement des valeurs limites fixées dans les règles internes de sécurité. Il envoie donc un message ALE en diffusion (au superviseur et aux autres produits). Le produit reste bloqué en attente du message ACKALE de la part du superviseur. Ce dernier n'envoie ce message que lorsque le problème est résolu.

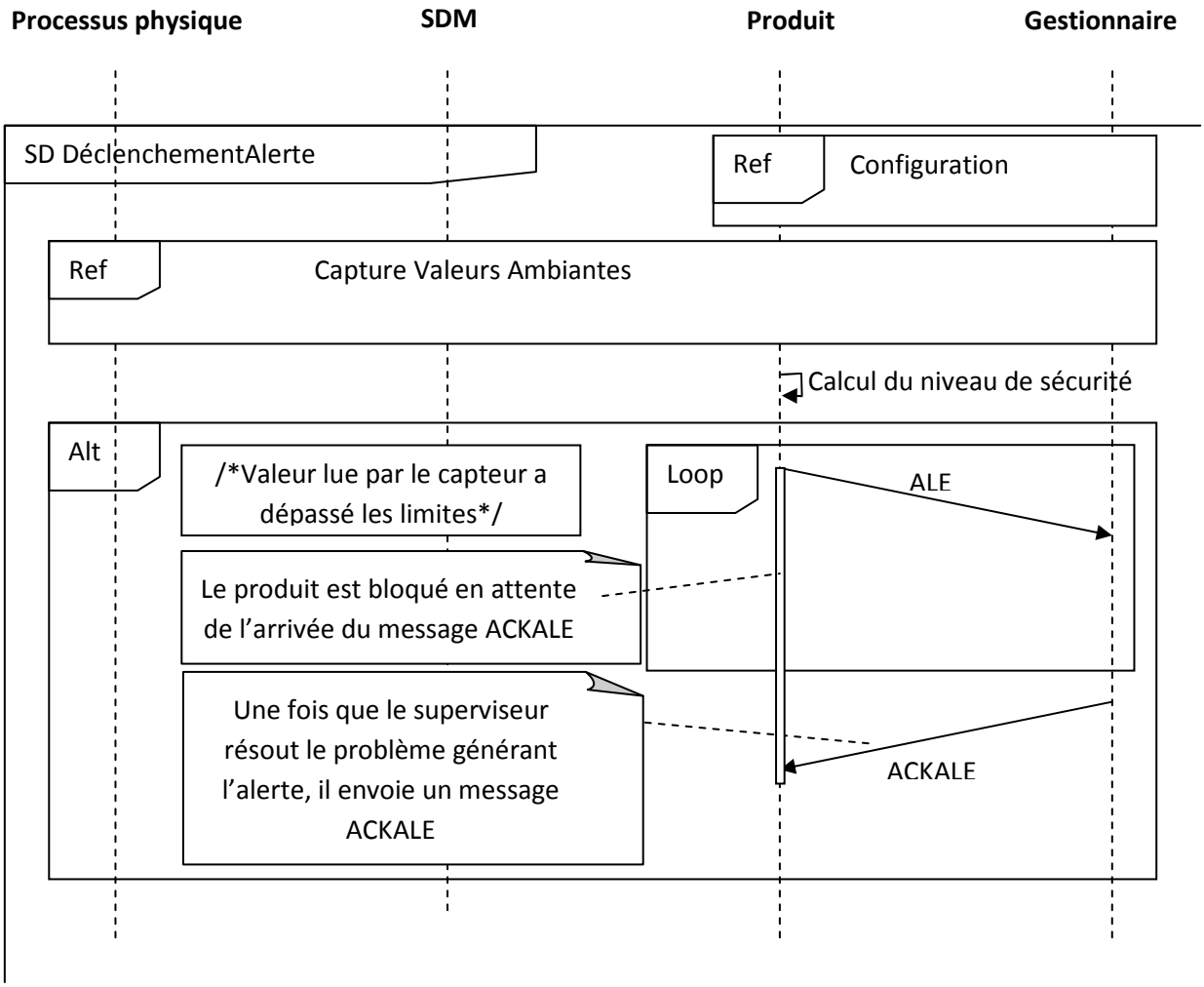

Figure II. 37. Scénario de déclenchement d'une alerte après dépassement d'une règle interne

IV.6.3. Le Scénario de fonctionnement complet

La figure II.38, illustre le fonctionnement normal du réseau dans le cas favorable qui ne correspond à aucune 'alerte. Les produits configurés entre en communication entre elles selon le modèle du scénario SalutationEntreProduits. Et chacun réalise sa fonction principale, qui est la détection périodique de la valeur du processus contrôlé, selon le scénario CaptureValeursAmbiantes.

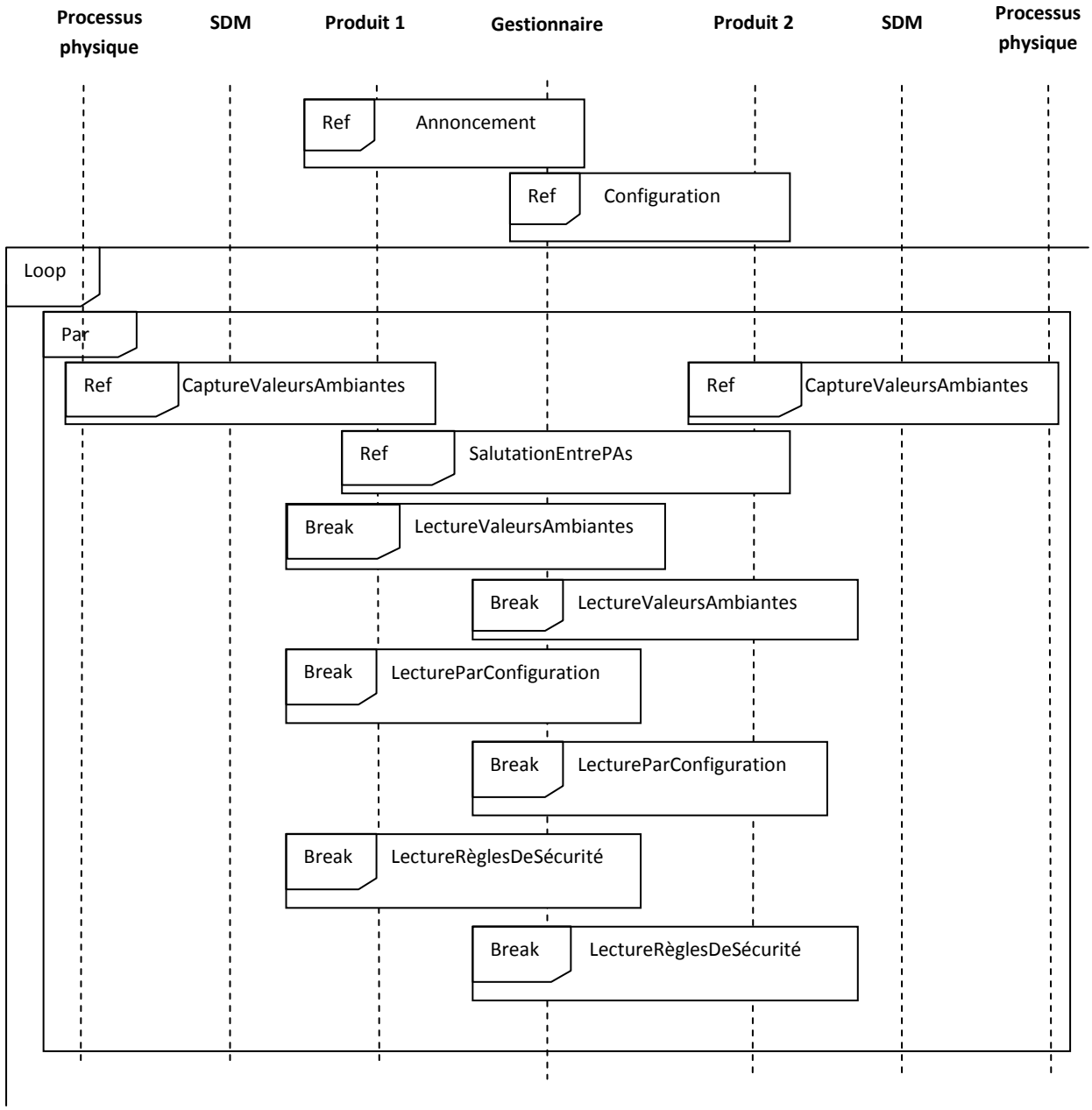

Figure II. 38. Scénario de fonctionnement complet du réseau

Après l'étape de configuration, le produit entre dans une boucle définie par deux activités qui sont :

- la salutation entre les produits.
- la capture des valeurs ambiantes.

Ces deux activités se répètent périodiquement. Mais il y a d'autres activités qui se déclenchent de façon événementielle suivant les besoins du superviseur comme :

- la demande de lecture des valeurs ambiantes,
- la demande de lecture des paramètres de configuration,
- et la demande de lecture des règles de sécurité.

IV.7. Le diagramme d'état/transition :

Le diagramme d'état/transition nous a permis de schématiser globalement le modèle interne du produit actif. Dans ce diagramme, un état est configuré par un rectangle et une transition par une flèche. L'état initial est représenté par un petit cercle plein et l'état final par un cercle vide. Ce modèle comporte un nombre bien défini d'états. La première étape consiste à envoyer le message CTR. Le produit actif passe donc à un état d'attente d'un acquittement de la part du gestionnaire. A la réception de l'acquittement, le produit actif peut avoir trois cas possibles selon sa configuration initiale. Si le produit actif n'a ni les paramètres de configuration ni les règles de sécurité, il envoie un message NCF0. Si le produit actif n'a pas les paramètres de configuration mais qu'il a les règles de sécurité, il envoie un message NCF1. Et si le produit actif a les paramètres de configuration mais qu'il n'a pas les règles de sécurité, il envoie un message NCF2. Après l'envoi d'un NCF0, lorsqu'il reçoit un message CMD1, il passe à l'état de configuration des paramètres généraux. Lorsqu'il reçoit un message CMD3, il passe à l'état de configuration des règles de sécurité. Mais après l'envoi d'un message NCF1 ou NCF2, il doit recevoir un seul message qui est respectivement le message CMD1 ou le message CMD3. Dans les trois cas précédents, s'il n'y a aucun message reçu, le produit actif continu à envoyer le message NCF0, NCF1 ou NCF2 périodiquement. Une fois que l'étape de configuration est achevée, le produit devient actif et il reste dans cet état jusqu'à la fin de sa durée de vie. L'état actif peut être traité et découpé en d'autres états dans un autre diagramme d'état/transition.

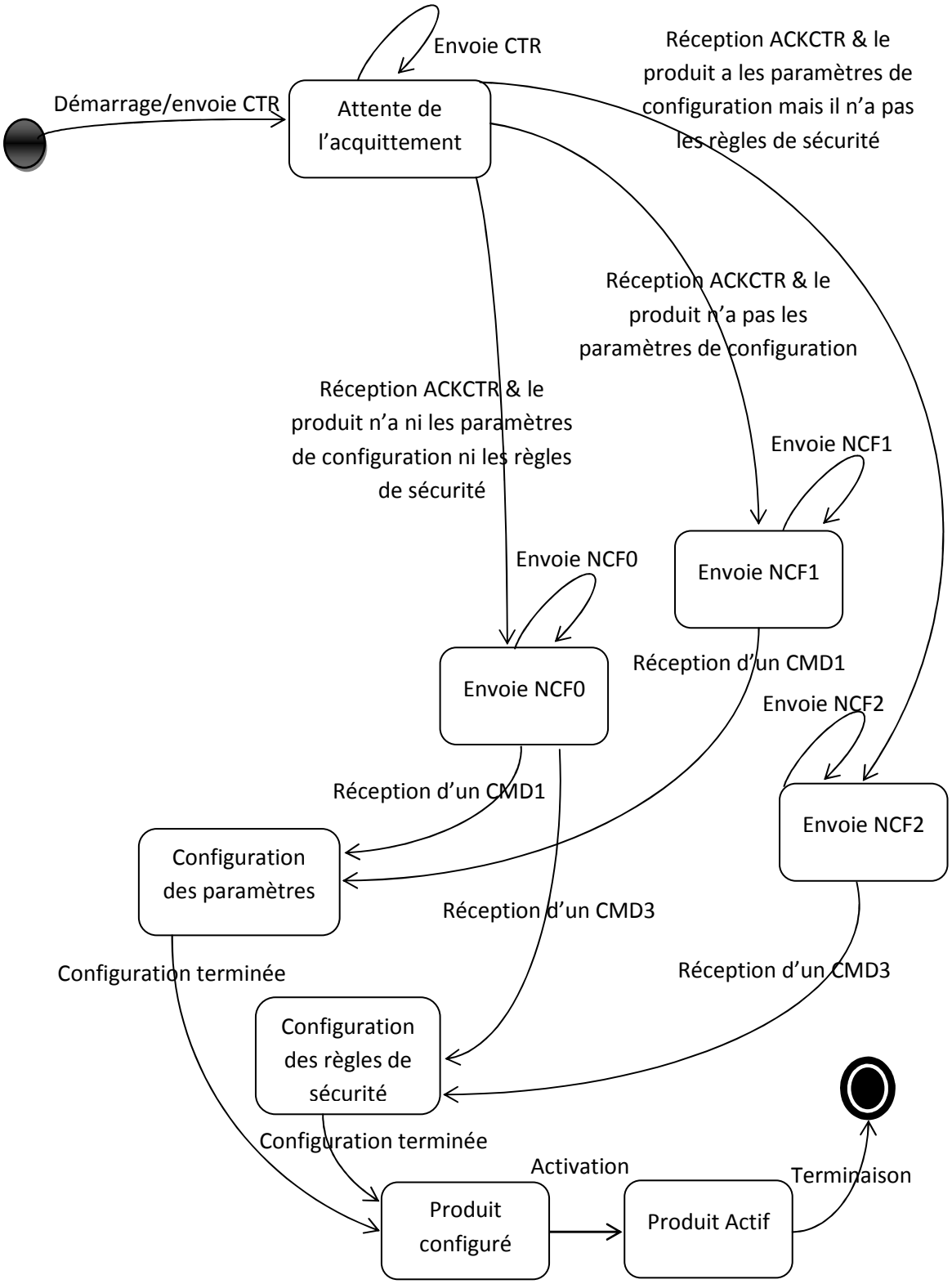

Figure II. 39. Diagramme d'états/transitions correspondant au modèle interne de produit actif

V. Conclusion

Le système de sécurité établi dans ce chapitre propose une approche novatrice que permet de résoudre les problèmes de stockage des produits conditionnés. Les modèles proposés permettent à se comporter individuellement dans son entourage en se servant des algorithmes de surveillance et de communication. Ce modèle a permis de concevoir une structure complexe détaillant l'aspect comportemental du produit actif. Dans le chapitre suivant nous allons établir un modèle par un outil de modélisation mathématique destiné aux systèmes industriels complexes à savoir les réseaux de Petri.

CHAPITRE III

Modélisation par réseaux de Petri de la coopération des produits actifs

I. Modélisation par réseau de Petri

I.1. Introduction

La complexité grandissante des systèmes industriels requiert de plus en plus de méthodes de représentation et de techniques d'analyse, permettant de tenir compte de ses différentes fonctionnalités, ainsi que de ses caractéristiques temporelles.

Cet impératif conduit inéluctablement à la nécessité de pouvoir disposer de méthodes formelles permettant de vérifier un certain nombre de propriétés du système modélisé. Les réseaux de Petri, parmi l'ensemble des formalismes existants, répondent à ces besoins.

En effet, les réseaux de Petri sont largement utilisés pour la modélisation et l'analyse de systèmes à événements discrets. Ce succès est dû à de nombreux facteurs. Parmi ceux-ci nous pouvons noter leur simplicité de compréhension, leur représentation graphique permettant sans grande difficulté à la modélisation de phénomènes complexes.

Plusieurs chercheurs ont utilisé le réseau de Petri comme étant un moyen de modélisation des comportements des réseaux tel que [Brahimi, 2007] qui a proposé une simulation par réseau de Petri d'un système contrôlé en réseau : elle a utilisé le réseau de Petri coloré pour modéliser le mécanisme de prise de décision ainsi que réseau de Petri coloré hiérarchique pour modéliser Ethernet en décrivant les canaux de transmission, les buffers de mémorisation et le modèle de commutateur. Ce modèle intègre les caractéristiques générales d'un réseau comme l'ordonnancement (gestion de priorité), la classification, la commutation, et les retards d'émission et de transmission.

[Bitam, 2005] et [Bitam and Alla, 2006a] utilisent le réseau de Petri pour modéliser et étudier les performances d'une ligne de transmission TCP/IP. Cette modélisation comprend la modélisation du routeur, du buffer et des canaux de transmission. Elle comprend aussi les pertes occasionnées lorsque le buffer est plein. Ce modèle illustre aussi le phénomène de congestion, suite à l'encombrement du réseau. [Bitam and Alla, 2006b] a présenté un modèle d'une ligne de transmission avec perte utilisée ultérieurement dans le cadre de coopération entre produits actifs.

[AbdulRauf and Jeyakumar, 2008] aborde le sujet de la modélisation d'un réseau 802.11 WLAN par réseau de Petri en mettant en jeu le débit du réseau et le retard du à la transmission. Pour cela il évoque trois modèles possibles en utilisant le réseau de Petri:

Modèle de commutateur : chaque entrée a ses propres buffers et le retard de transmission est fixe.

Modèle de point d'accès : c'est le même que celui du commutateur mais avec un retard de transmission variable.

Modèle d'un terminal : au sein d'un réseau, les terminaux réagissent entre eux (échange des paquets) ce qui permet d'estimer le retard du réseau LAN entre l'instant d'émission d'un terminal et l'instant de réception d'un autre.

[Khoukhi and Cherkaoui, 2010] remarque que le RdP classique est incapable de modéliser les systèmes incertain ce qui a poussé les chercheurs à combiner le RdP et la logique floue pour aboutir à un réseau de Petri flou. Le mécanisme décisionnel flou est situé dans l'inter couche (MAC). Cette étude cherche à utiliser le retard récolté de réseau pour améliorer le bon acheminement des paquets et apporter une solution au phénomène de congestion cité par [Bitam and Alla, 2006b].

[Hsieh, 2009] présente un concept de vérification et de résolution de problème lié au mécanisme de coopération et d'interaction des systèmes multi-agents. Ces systèmes sont souvent modélisés en RdP et l'approche consiste à contrôler la vivacité du réseau.

[Song et al, 2008] définit le réseau de Petri comme étant un outil de modélisation des événements qui nécessitent une synchronisation spéciale tel que le réseau de capteurs sans fil. Le système modélisé est un système de sécurité (évacuation) utilisé dans les mines du coke qui localise la positon des mineurs en cas d'accident. Dans la modélisation il est passé par deux phases à savoir (1) une modélisation d'un produit (service et communication) et (2) une généralisation du modèle à grande échelle.

I.2. Le formalisme des Réseaux de Petri colorés

Avant de passer aux réseaux de Petri colorés commençons par la présentation de la définition de base du formalisme des réseaux de Petri qui ont été introduits par Carl Adam Petri dans sa thèse intitulée « Communication avec Automates » en 1962.

I.2.1. Définitions de base

Un réseau de Petri ou RdP peut être considéré comme un système composé de deux parties distinctes [Bonhomme, 2001]. Une partie statique ou structurelle qui présente le graphe du réseau et sa structure même à travers les places et les transitions et une partie dynamique ou comportementale qui montre l'évolution du réseau et le déplacement de ses jetons au cours de franchissement des transitions en question.

Les Réseaux de Petri sont des graphes orientés bipartis se composant du quadruplet < P, T, Pré, Post > avec:

P, un ensemble fini de places, {p1, p2,..., pn}.

T, un ensemble fini de transitions, {t1, t2,..., tm}.

Ces deux ensembles forment les sommets du réseau,

Pré : P x T $\rightarrow$IN est l'application d'incidence avant, correspondant aux arcs directs reliant les places aux transitions.

Post : P x T $\rightarrow$IN est l'application d'incidence arrière, correspondant aux arcs directs reliant les transitions aux places.

Chaque Réseau de Petri est qualifié par son marquage. Le marquage correspond à un nombre entier positif ou nul de marques ou de jetons contenus dans chaque place du réseau. Le marquage à un certain instant reflète l'état du RdP qui est en fait l'état du système modélisé. L'évolution du marquage correspond donc à l'évolution de l'état du système. Cette évolution prend place par le franchissement des transitions.

Le franchissement d'une transition se réalise si chacune des places en amont de cette transition contient au moins une marque. On obtient donc une transition franchissable ou validée. Le franchissement d'une transition conduit à l'élimination d'un jeton sur toutes les places en amont de la transition et l'ajout d'un jeton sur toutes les places en aval.

I.2.2. Réseau de Petri colorés

Les réseaux de Petri offrent un cadre bien adapté et progressif pour la représentation et l'analyse des systèmes complexes notamment ceux de la communication [Brahimi et al., 2006] et des chaînes logistiques [Long, 1993]. Les RdP sont bien appropriés pour décrire les systèmes dans lesquels interviennent les problèmes de concurrences, de synchronisation, de partage de ressources et de parallélisme [Juanole et al., 2004], [David and Alla, 1989], [Jensen, 1982]. Il arrive souvent que dans un système, on rencontre plusieurs parties qui ont une description identique. Elles sont alors modélisées par le même RdP autant de fois qu'il y a de parties identiques. Cela a pour conséquence d'aboutir à des modèles de taille trop importante et donc inexploitable. D'où la nécessité de réduire la taille des modèles en associant des couleurs pour des modèles identiques qui sont dupliqués [Kristensen et al., 1998].

De plus, dans un RdP l'information est portée par la place. Par exemple, la présence d'un jeton (ou marque) dans une place peut signifier une ressource disponible. Si elle est vide, cela signifie que la machine est occupée. Si on veut enrichir l'information apportée par une place d'un RdP, il faut être en mesure de distinguer deux jetons entre eux au sein de la même place. On associe alors un identificateur ou une couleur à chaque jeton de la place.

L'information est représentée par un ensemble place-couleur. Ainsi est défini un nouvel outil de modélisation : les réseaux de Petri colorés.

I.2.2.1. Présentation intuitive des Réseaux de Petri colorés

Le comportement d'un système peut être vu en termes de conditions et d'actions ou d'événements : une action est réalisée quand un certain nombre de conditions d'entrée à l'action existent. La réalisation d'une action induit la génération de nouvelles conditions à savoir les conditions de sortie de l'action qui elles mêmes vont être les conditions d'entrée d'une autre action etc. [Juanole et al., 2004] [David and Alla, 1989]. Le modèle RdP places-transitions est très bien adapté pour représenter cet aspect comportemental d'un système.

Prenons deux systèmes identiques modélisés chacun par un RdP, on peut représenter le fonctionnement de ces deux systèmes par un seul RdP en attribuant une couleur à chaque jeton associé au RdP représentant chaque système. Pour réaliser une action, un jeton passe d'une place à une autre en franchissant une transition. Les places et les transitions sont reliées par des arcs associés à des fonctions. Les fonctions correspondantes aux arcs en entrée permettent d'indiquer les conditions d'activation d'une transition alors que les fonctions des arcs en sortie déterminent l'effet de l'action représentée par la transition. Ces fonctions représentent la relation entre les couleurs de franchissement et le marquage coloré concerné.

I.2.2.2. Présentation formelle

Comme présenté dans [Marsal, 2006], [Jensen, 1981], [Jensen, 1982], [David and Alla, 1989], un réseau de Petri coloré est un sextuplet (P, T, Pré, Post, M_0, C) :

Où,

P est un ensemble fini de places,

T est un ensemble fini de transitions disjoint de P,

C est l'ensemble des couleurs, C= {c1, c2, c3,…} appelé Type. A chaque place est associé un type ou un ensemble de couleurs.

Pré, Post : P x T $\rightarrow$IN où Pré, Post sont respectivement des fonctions d'incidence avant, et des fonctions d'incidence arrière. Ces fonctions sont associées à chaque arc pour traduire la relation qui existe entre la couleur et la transition associée : celle choisie pour franchir cette transition (couleur de franchissement) et celle associée à la place correspondante (couleur du jeton).

M_0 est un ensemble multiple appelé marquage initial du réseau.

Dans la suite deux exemples illustratifs présentent les notions citées ci-dessus. Ces exemples sont introduits dans les ouvrages de [David and Alla, 1989] et [Jensen, 1997] :

Exemple d'un RdP coloré :

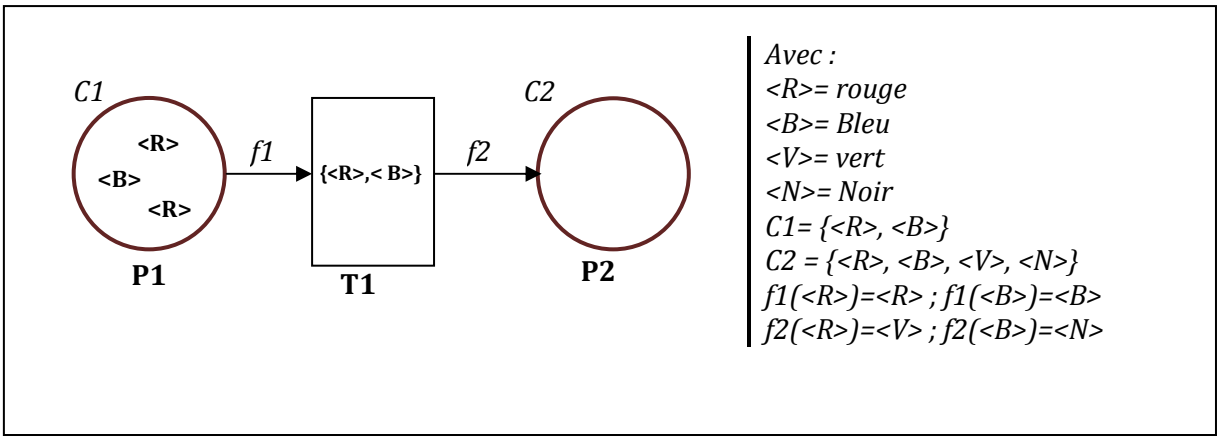

Figure III. 1. Exemple d'un RdP coloré

La place P1 de la figure III.1 contient trois jetons, un jeton de couleur bleu, et deux jetons de couleur rouge. Ces couleurs font partie des types ou des ensembles de couleurs C1 et C2.
Une transition est représentée par un rectangle. A chaque transition est associée un ensemble de couleurs. Chacune de ces couleurs indique une possibilité distincte de franchissement. La transition T1 peut être franchie par rapport à la couleur <rouge> et à la couleur <bleu>.
Un arc orienté relie une place à une transition, ou une transition à une place. Le poids d'un arc est une fonction Pré (appelée ici f1) ou Post (appelée ici f2) qui établit une correspondance entre chaque couleur de la transition et les couleurs de la place. Ainsi le franchissement de T1 par rapport à la couleur <rouge> retire la couleur <rouge> qui se traduit par la fonction [f1(rouge)= rouge]. Elle ajoute un jeton de couleur <vert> qui se traduit par la fonction [f2(rouge)=vert]. Donc les fonctions Pré et Post s'écrivent comme suit :

Pré (P1, T1/<R>)= *f1* (rouge)= rouge,
Pré (P1, T1/<B>)= *f1* (bleu)= bleu.
Post (P2, T1/<R>)= *f2* (rouge)= vert,
Post (P2, T1/<B>)= *f2* (bleu)= noir.

La fonction f1 est appelée fonction identité puisqu'elle renvoie la même couleur du jeton d'entrée.

Exemple Couleurs complexes :

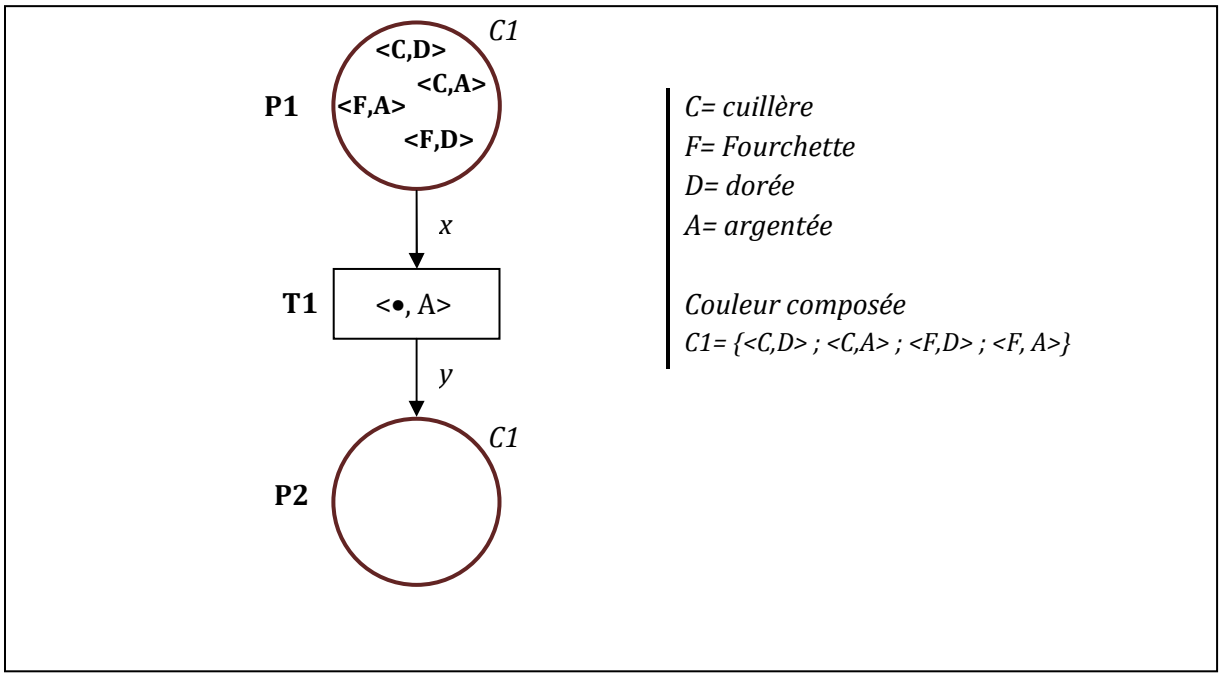

Figure III. 2. Exemple d'un RdP avec couleur composée

Imaginons dans l'exemple que nous disposons d'un ensemble de cuillères et de fourchettes de teinture dorée et argentée. Dans l'exemple de la figure III.2 le RdP contient deux places P1 et P2, auxquelles est associé un type de couleur complexe. C'est-à-dire, une couleur comporte en elle d'autres informations supplémentaires. Ici nous avons deux informations par jeton, la première concerne la nature du couvert et la deuxième concerne sa teinture, soient :

<C,D>= Cuillère Dorée
<C,A>= Cuillère Argentée
<F,D>= Fourchette dorée
<F,A>= Fourchette Argentée

Ce type complexe est défini dans la déclaration du RdP, à savoir : C = {<C,D> ; <C,A> ; <F,D> ; <F,A>}. Ici les couleurs de franchissement de la transition sont définies implicitement. La place P1 contient les 4 jetons du Type complexe C. Quand la transition T1 est franchie, un jeton appelé 'x' associé à l'arc amont de T1, est retiré de la place P1, et un jeton appelé 'y' associée à l'arc aval de T1 à P2. Le garde '<•,A>' associé à T1 définit la condition de franchissement de la transition T1, à savoir pour que T1 soit franchie il faut que la deuxième information du jeton x soit 'A' (Argentée) quelque soit le contenu de la première information. Autrement dit, la transition T1 ne laisse passer que les couverts de teinture argentée.

En faisant une analogie aux définitions de [David and Alla, 1989] cela peut être assimilé aux fonctions suivantes :

$$\begin{cases} x: x(< C, A >) =< C, A > \\ x: x(< C, D >) =< C, D > \\ x: x(< F, A >) =< F, A > \\ x: x(< F, D >) =< F, D > \end{cases}$$

C'est la fonction identité associée à l'arc amont de T1.

Le garde et 'y' peuvent être définis par la fonction suivante :

$$\begin{cases} y: y(< C, A >) =< C, A > \\ y: y(< C, D >) = \emptyset \\ y: y(< F, A >) =< F, A > \\ y: y(< F, D >) = \emptyset \end{cases}$$

Dans cet exemple, la condition de franchissement de la transition dépend de la présence d'une information particulière dans le jeton. Cette propriété est utile dans la modélisation des échanges des messages tenant compte d'autres critères en plus du type du message lui-même telle que l'identification de l'émetteur et du récepteur.

I.2.2.3. Les réseaux de Petri colorés Temporisés

La notion de délai dans les RdP est liée soit aux places soit aux transitions. Les RdP associant le délai aux transitions sont appelés aussi RdP T-temporisés. Nous allons ici introduire la notion du temps selon la sémantique de [David and Alla, 1989] et [Ramchandani, 1974] qui est aussi utilisé par l'outil de simulation CPN Tools. Donc un RdP Coloré T-temporisé est défini comme un n-tuplet : (RdPC, tempo)

- RdPC est le réseau de Petri coloré,
- Tempo est une application de l'ensemble des transitions dans l'ensemble des rationnels positifs ou nuls.

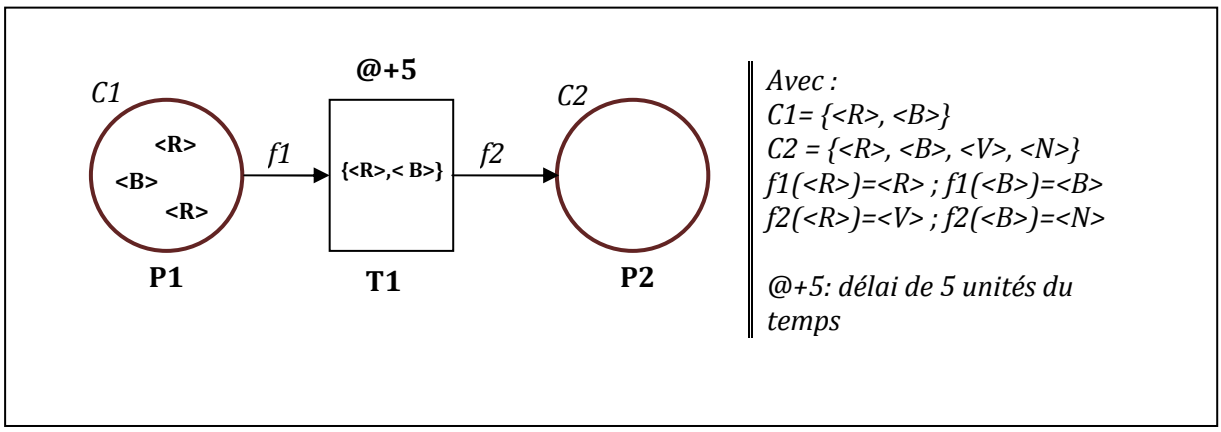

Figure III. 3. Exemple d'un RdP coloré temporisé

I.3. La Hiérarchie dans les réseaux de Petri

Le réseau de Petri Hiérarchique est un modèle dans lequel une partie peut être représentée par une transition de substitution qui est une abstraction d'un autre modèle. C'est à dire, que cette partie est détaillée à part. La hiérarchie sert à subdiviser un modèle en différentes parties ce qui autorise une modélisation modulaire.

Dans ce qui va suivre, nous allons définir la sémantique utilisée par [Jensen, 1997]. Donc un réseau de Petri coloré hiérarchique (RdPCH) est un n-tuplet : RdPCH = (S, SN, SA, PN, PT, PA) où :

S est un ensemble fini non-vide de pages où chaque page représente un réseau de Petri coloré non-hiérarchique,

SN est un ensemble de nœuds de substitution appartenant à l'ensemble T (ensemble de transitions),

SA est une fonction d'assignation des pages définies à partir de SN dans S. Cela veut dire que le nom (ou l'assignation) du nœud de substitution correspond au nom (ou l'assignation) de la sous-page qui le représente par un RdPC non-hiérarchique.

PN est un ensemble de nœuds appelés Port appartenant à l'ensemble P.

PT est le type de fonctions associé au nœud 'Port' des éléments PN. Ce type de fonctions peut être : in, out, i/o (in/out) comme indiqué dans la figure III.4.

PA est une fonction d'assignation du port qui définit la relation entre les nœuds, et entre les pages et les sous-pages. Le type d'interconnexion entre les pages et les sous pages s'effectue à partir des transitions. Une transition de substitution remplace une page complète.

Pour mieux illustrer ces définitions un exemple introduisant ces notions est représenté par la figure III.4.

Dans l'exemple de la figure III.4 nous avons les mêmes déclarations que dans l'exemple de la Figure III.1, seulement ici nous avons ajouté une transition T2 qui est une transition puits. Elle retire un jeton y de P2. La transition T1 est une transition de substitution à laquelle est associée une sous-page du même nom. Cette sous-page représente un RdP coloré (i.e. la transition T1 est une abstraction du réseau de Petri coloré représenté par la sous-page T1) possédant deux places : P1-1, P2-1. P1-1 et P2-1 qui doivent être du même type que P1 et P2 respectivement et qui doivent avoir le même nombre de jetons. Les places P1 et P1-1, (respectivement P2 et P2-1) sont appelées des places fusionnées.

Les places P1 et P2 représentent les ports d'entrées et de sorties représentées par les étiquettes in et out respectivement de la transition de substitution T1 donc du RdPC représenté par la sous-page T1.

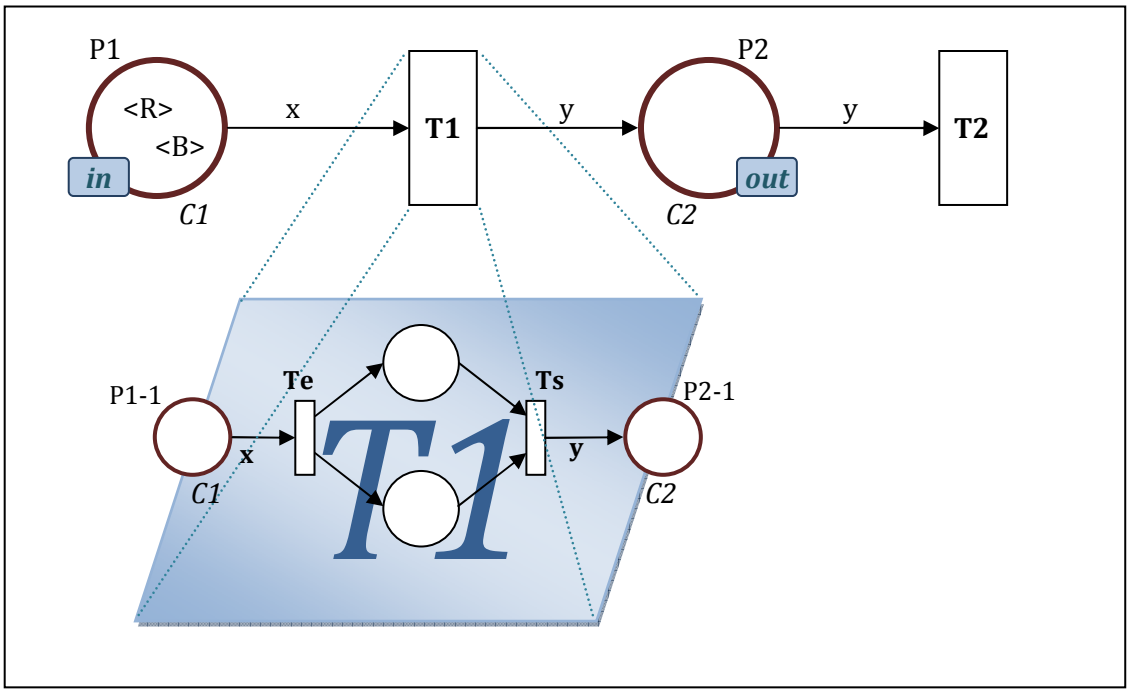

Figure III. 4. Réseau de Petri coloré Hiérarchique

T1 remplace le sous Réseau de Petri coloré représenté entre les transitions Te et Ts. En fait, un sous réseau de Petri doit être délimité par deux transitions pour qu'il puisse être substitué par une transition de substitution.

S est la sous-page de la transition de substitution T1. SN est la transition de substitution elle même.

PN représente les places ports P1 et P2, PT représente les fonctions in et out associées à P1 et P2. PA représente la fonction d'interconnexion entre les places fusionnées et la transition de substitution ainsi que la page qu'elle représente.

II. Outil de modélisation CPN Tools

II.1. Introduction CPN Tools :

La modélisation de la coopération entre produits actifs a été réalisée suivant un modèle hiérarchique de réseau de Petri coloré en utilisant l'outil de modélisation CPN Tools[18]. C'est

[18] CPN Tools :Colored Petri Net Modeling Tools

un outil de modélisation pour le formalisme des réseaux de Petri colorés [Jensen, 1981]. Il permet d'éditer des réseaux, de vérifier la syntaxe et la cohérence (en pas à pas ou automatiquement avec pour critère d'arrêt une date ou un nombre de tirs de transitions). Nous présentons ici quelques caractéristiques de cet outil liées à la modélisation.

II.2. Couleurs supportées par CPN Tools

La déclaration des couleurs et des variables est effectuée à l'aide du langage CPN ML basé sur le langage de programmation ML [Ullman, 1998]. Grâce au logiciel CPN Tools, il est possible d'utiliser des couleurs complexes et des fonctions. Les couleurs simples existantes dans le CPN ML, sont : unité (couleur noir –sans couleur-), entier, booléen, string (chaîne de caractères) et énuméré.

Notre modèle est essentiellement basé sur le type complexe "product". Ce type ressemble au type record (enregistrement) du Turbo Pascal qui possède plusieurs champs de même type ou de types différents et les regroupe en un seul type complexe. Cette caractéristique nous aide à associer plusieurs critères dans un seul type. Par conséquent un jeton du type complexe "MESSAGE" peut supporter plusieurs informations telles que la nature du message, son origine (de quel émetteur) et sa destination (vers quel récepteur).

II.2.1. Integer

Integer est le type des entiers naturels. CPN Tools nous offre la possibilité de réduire la largeur de cette couleur en un intervalle plus restreint. Voici un exemple de déclaration de ce type dans CPN Tools:

Colorset douzaine= int 1..12;

Nous avons ainsi déclaré l'ensemble de couleur appelé douzaine qui comporte 12 éléments.

II.2.2. Enumerated

Enumerated est le type 'énuméré'. Il permet de regrouper explicitement des éléments identifiés. Voici un exemple de déclaration :

Colorset semaine = with Lun| Mar| Mer| Jeu| Ven;

Où nous avons défini l'ensemble de couleur semaine qui comporte comme élément les jours de la semaine.

II.2.3. Product

Comme cité ci dessus le type 'Product' ou 'produit' est l'équivalent du type 'record' dans le langage de programmation Turbo Pascal. 'Product' permet de combiner des ensembles de

couleur simple. Dans notre modèle, nous avons défini un ensemble de messages destinés à être échangés entre produits actifs et aussi entre produits actifs et gestionnaire. Voici donc un exemple de déclaration :

Colset Msg = with CTR|GRE|RSI|NCF0|NCF1|NCF2|CMD1|CMD2|CMD3|rapp_M|rapp_D|CMD4|CMD5|INA|SER|CFG|AckCTR|Ack_CMD3|Ack_CMD1|Ack_rapp_D|Ack_rapp_M;

colset Psend=with P1|P2|P3|OP|GES|RES;

colset Prec=Psend;

Msg est alors un type 'énuméré' permettant de définir un ensemble de couleurs comportant les différents champs des messages échangés entre produits actifs.

Psend et Prec sont aussi de type énuméré mais désignent respectivement les produits actifs émetteurs et récepteurs des messages.

colset MESSAGE = product Psend*Prec*Msg;

MESSAGE est un type qui comporte trois champs. Les deux premiers champs sont de type Psend ou Prec pour identifier respectivement l'émetteur et le récepteur. Le troisième champ est de type Msg il porte l'information et la nature du message.

Par exemple (P1,P2,GRE) est une couleur de l'ensemble de couleur MESSAGE. Le premier champ P1 est un Psend se référant à l'émetteur qui est le produit actif numéro 1. Le deuxième champ P2 est un Prec se référant au récepteur qui est le produit actif numéro 2. Le troisième champ se réfère au message transporté à savoir le message GRE (greating).

II.3. Les Fonctions

Le logiciel CPN Tools fournit un ensemble de structures et de syntaxes permettant de construire des fonctions. Un tel avantage donne la possibilité de construire un RdP paramétrable dont on peut obtenir une instanciation en changeant seulement les déclarations.

Exemple :

Fun comparaison(x) = Si x<3

Alors « x est inférieur à trois »

Sinon « x est supérieur ou égal à 3 »

Cette expression vérifie si la valeur de x est inférieure à 3 et retourne le caractère approprié à la réponse.

Exemple :

Fun Date(jour) = case jour of

Lun => « aujourd'hui est Lundi »

|Mar => « aujourd'hui est Mardi »

|Mer => « aujourd'hui est Mercredi »

|Jeu => « aujourd'hui est jeudi »

|Ven => « aujourd'hui est Vendredi »

| _ => « c'est un jour de weekend »

Cette expression retourne le caractère qui dépend de la valeur de 'jour', où jour est de type semaine présentée ci-dessus dans l'exemple des types 'énuméré'. Le blanc souligné '_' indique que l'expression ne retourne le caractère «c'est un jour de weekend» que si la variable 'jour' est égale à un caractère autre que Lun, Mar, Mer, Jeu, Ven.

Dans notre cadre on a utilisé plusieurs fonction tel que

fun Prc()=Psend.ran() ;

Cette fonction choisie au hasard une variable de type Psend expliqué précédemment. Dans ce qui suit toute fonction qui présente la structure type1.ran désigne une fonction de choix au hasard d'un élément de type1 tel que :

fun compatibilite()=comp.ran() ;

fun distance()=dist.ran();

fun config()=conf.ran();

fun val_T()=Temperature.ran();

fun val_L()=Lumiere.ran();

fun val_H()=Humidite.ran();

fun eval()=etat.ran();

Les trois fonctions qui suivent présentent les fonctions d'évaluation des règles de sécurité des variables ambiantes (température, humidité et lumière UV). Et à partir des variables recueillies on évaluera l'état de système.

fun eval_T(i)= if i> LimSup_T orelse i< LimInf_T then 1`danger else if i> LimSup_T-ΔT orelse i< LimInf_T +ΔT then 1`mauvais else 1`bon; (pour la temperature).

fun eval_L(i)= if i> LimSup_L then 1`danger else if i> LimSup_L-ΔL then 1`mauvais else 1`bon;(pour la lumière).

fun eval_H(i)= if i> LimSup_H then 1`danger else if i> LimSup_H-ΔH then 1`mauvais else 1`bon; (pour l'humidité)

Et la dernière fonction, c'est la fonction qui, à partir des états des trois variables, indique l'état global du système.

fun etat_rslt(a1,b1,c1)= if a1=danger orelse b1=danger orelse c1=danger then 1`danger else if a1=mauvais orelse b1=mauvais orelse c1=mauvais then 1`mauvais else 1`bon;

III. Modélisation par RdP

Notre étude s'oriente vers la représentation par réseaux de Petri d'un modèle de comportement, des interactions et des processus de coopération entre produits communicants dans un environnement à intelligence ambiante.

III.1. Modèle de coopération entre produits actifs

Le modèle de coopération est doté de six éléments (P1, P2, P3, GEST, OP, ressource) qui communiquent entre eux, en formant une communauté de coopération sans fil.

Pi représente un produit actif numéro i, GEST est le gestionnaire, OP est un opérateur et ressource par exemple un chariot.

Chaque élément, est représenté par une transition (hiérarchique) qui présente les services et les tâches appropriées citées en détail dans ce qui suit. Comme l'indique la figure.III.5, chaque nœud présente deux places : Net Input et Net Output qui représentent respectivement les buffers de sortie et d'entrée de chaque élément. Ces buffers ont pour rôle de mémoriser temporairement les messages reçus à partir du réseau avant d'être traités (dans l'unité de traitement) et ceux émis dans le réseau.

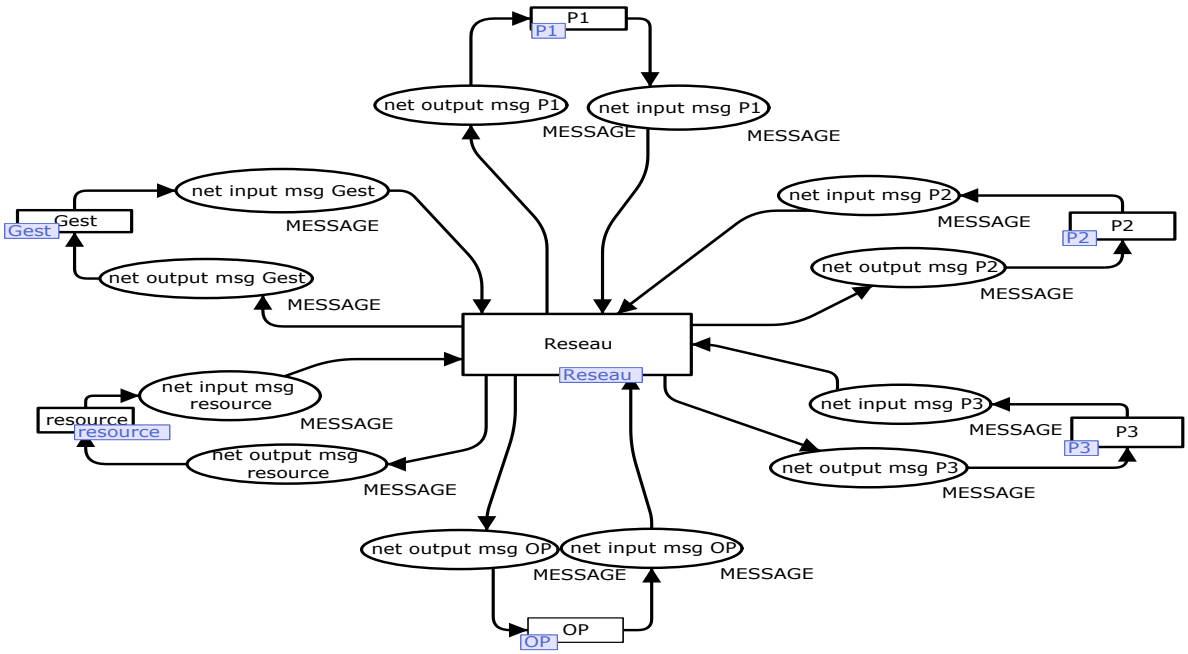

Figure III. 5. Modèle de coopération entre produits actifs

III.2. Niveau réseau

Dans cette partie on va présenter le modèle de coopération entre les produits actifs en construisant le modèle du réseau représenté par la transition de substitution « Reseau » de la figure III.5. Le réseau est considéré dans un premier temps comme parfait et dans un second avec pertes de paquets.

III.2.1. Réseau sans perturbation

La figure.III.6 indique le niveau hiérarchique inférieur de la transition de substitution «Réseau». Les places supérieures net input désignent les buffers des sorties des produits actifs où les messages ont été stockés avant d'être émis dans le réseau. Ces messages passent par une étape correspondant à la phase d'accès au medium.

Le réseau étant parfait (ne présent pas de perturbation), alors tous les messages vont passer directement à travers la transition réseau vers les buffers de sortie du réseau messages reçus puis redirigés vers leur destinataire.

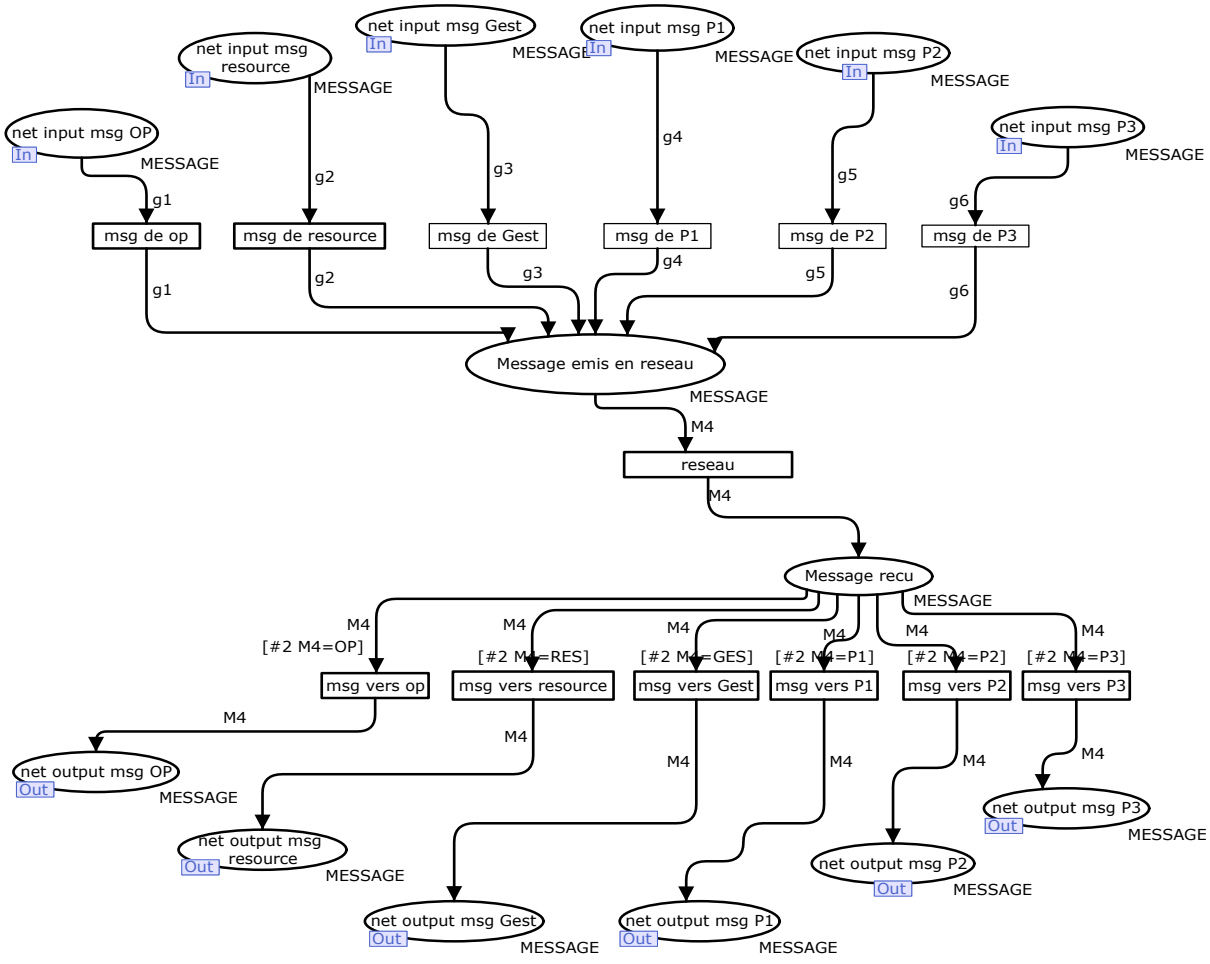

Figure III. 6. Modèle de réseau sans perturbation

III.2.2. Réseau avec perturbations

Le réseau présenté dans la figure.III.7 défini un réseau perturbé où il y a risque de pertes de paquets lorsque celui-ci passe de buffer d'entrée réseau vers le buffer de sortie. Chaque jeton (message), qui se présente dans la place ("message émis en réseau") doit franchir la transition en aval où il sera affecté à une autre place. A cet instant, ce paquet sera perdu (franchissement de la transition *Perdu*) ou passé (franchissement de la transition *Passé*). Si le paquet n'est pas perdu, il entre dans un buffer d'entrée et après entre dans un buffer de sortie pour être enfin dans la place "message reçu". Ce modèle de perturbation réseau a été présenté par [Bitam and Alla, 2006a] où ils ont modélisé une ligne de transmission avec pertes.

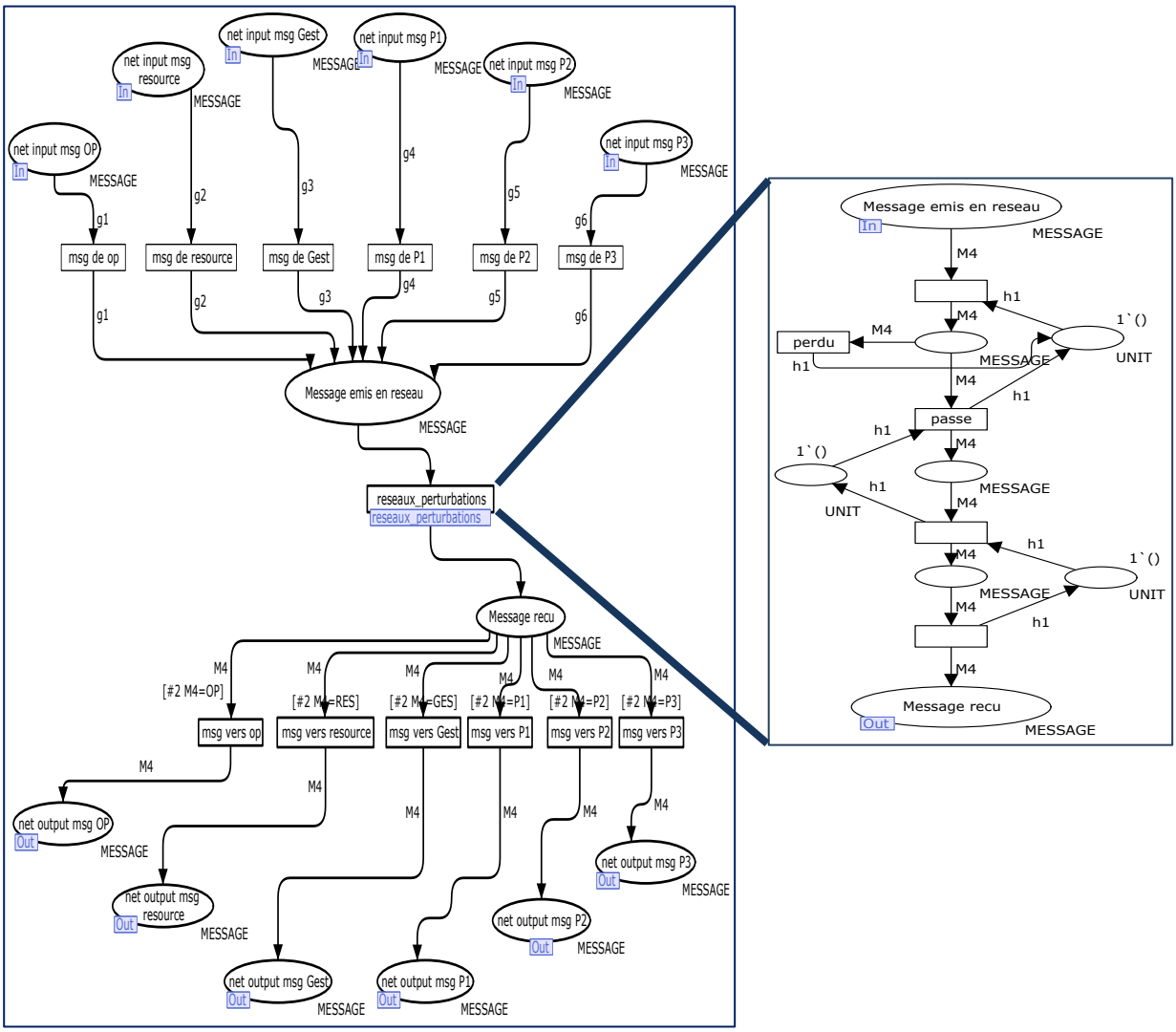

Figure III. 7. Modèle de réseau présentant une perturbation

La modélisation de perte dans une ligne de communication indiquée en figure.III.8 se présente comme suit: lorsque un message (jeton) se trouve dans la place P1 et un autre se trouve dans la place P2 la transition T'1 est franchissable, donc ce message va passer à la place P3, à cet instant il y a deux directions : franchissement de T''1 (ce message sera perdu) ou franchissement de T1 (le message va être émis) et après il passe au buffer d'entrée ensuite à celui de sortie (après franchissement de T2). Ce phénomène s'explique comme suit: parfois la vitesse de traitement d'un nœud est beaucoup moins rapide que celle de réception. Donc quelques messages n'auront pas la chance d'être traiter par le nœud du capteur du à la limitation de la place d'entrée E.

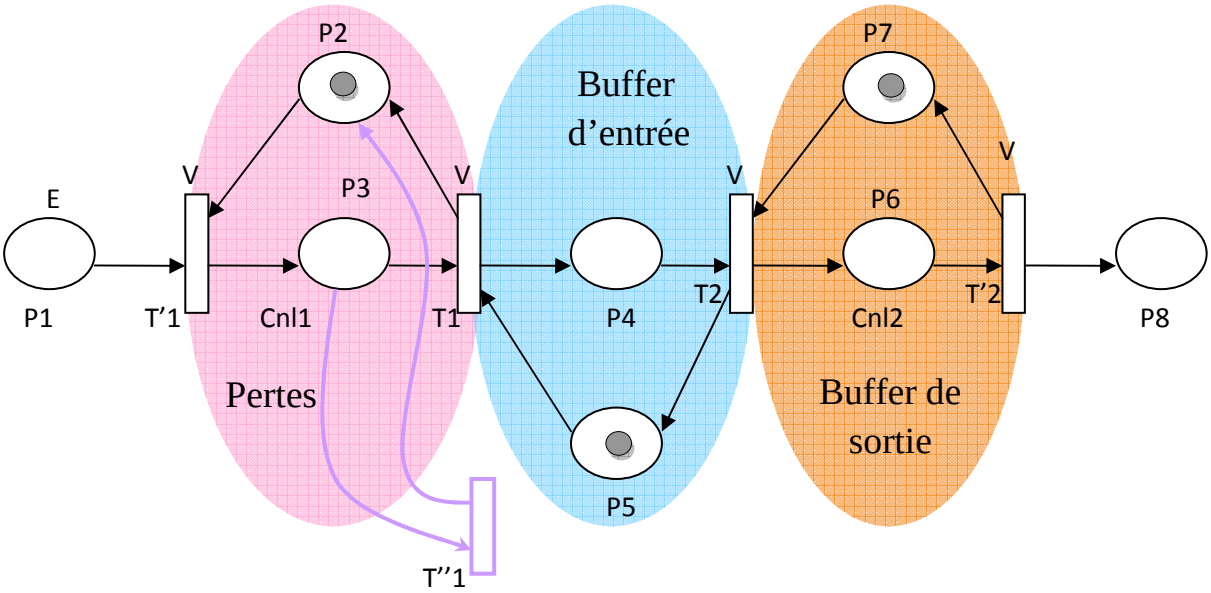

Figure III. 8. Ligne de transmission avec perte [Bitam, 2005]

III.3. Niveau produit actif

Comme l'indique la figure III.9, la représentation du produit actif est décomposée en hiérarchie de modèles qui correspondent aux fonctions d'enregistrement, de configuration, de surveillance interne et de surveillance et communication (surveillance globale).

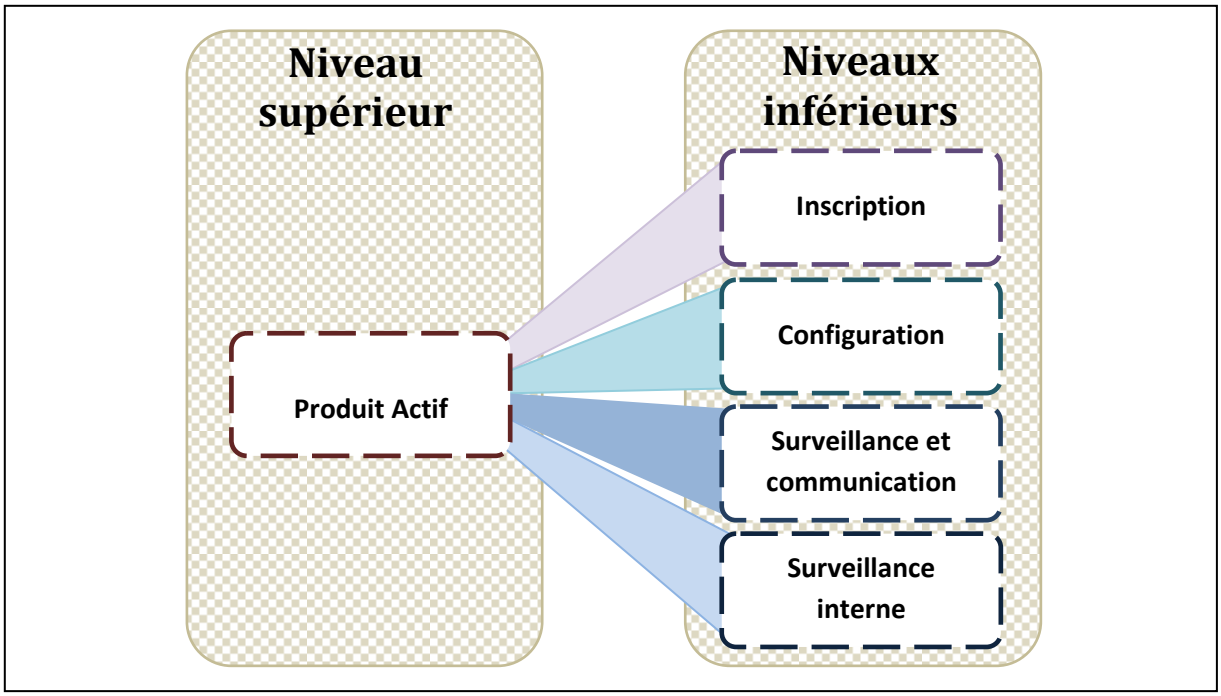

Figure III. 9. Structure du modèle Hiérarchique du produit actif

III.3.1. Modèle de produit actif

Comme l'indique la figure.III.9, chaque produit actif présente des tâches internes et externes dont certaines obéissent à l'approche centralisée alors que les autres suivent l'approche d'omniprésence. Chaque transition dans ce réseau présente une structure hiérarchique décrite explicitement dans la figure III.10.

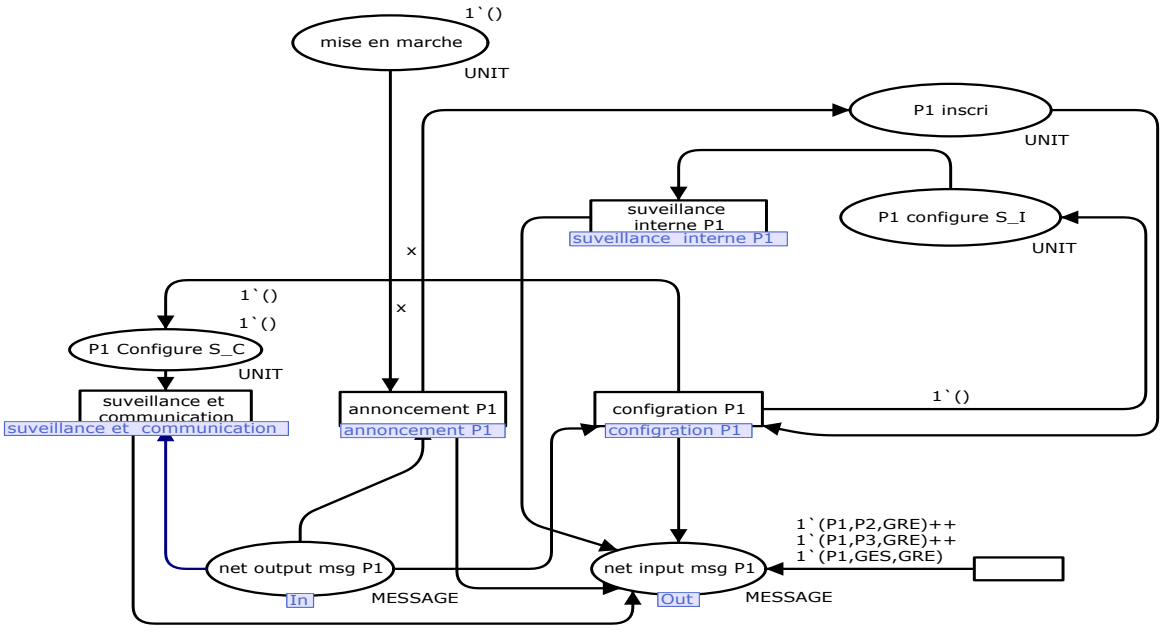

Figure III. 10. Modèle de produit actif

III.4. Niveau fonction de dépendance de produit actif

Les deux tâches représentées en transition hiérarchique : annoncement P1 et configuration P1, illustrent l'approche centralisée où chaque produit actif doit se référer initialement au gestionnaire pour s'annoncer (entrer dans le réseau et avoir un ID) et aussi pour se configurer (demander les règles de sécurité au gestionnaire).

III.4.1. Inscription (annoncement)

L'inscription sert à introduire un produit étranger dans l'entourage des autres produits actifs. Le besoin de se lancer dans cette communauté nécessite une inscription vers le gestionnaire pour que ce dernier le détecte et l'ajoute dans sa base de données des produits déjà existant.

Dans le modèle qui gère l'inscription, la place « net input msg P1» est une place de sortie pour diriger le message vers la place « Reseau ». La réception des messages se manifeste à travers la place d'entrée « net output msg P1 ». Après la mise sous tension, le Produit Actif s'annonce dans le réseau par envoi d'un message de découverte qui est détecté par le gestionnaire. Ceci est explicité à travers le jeton (P1,GES,CTR) déposé dans la place NET INPUT MSG P1. La transition « ACK » ne sera franchie que si un jeton (GES,P1,AckCTR) figure dans la place « net output msg P1 ». L'absence d'un jeton d'acquittement entraînera la validation de la transition « ACK_bar » et le même processus se répètera. La fonctionnalité de ce RdP permet d'assurer une inscription du produit dans le réseau et son accomplissement nous mène vers la partie suivante : modèle de la configuration.

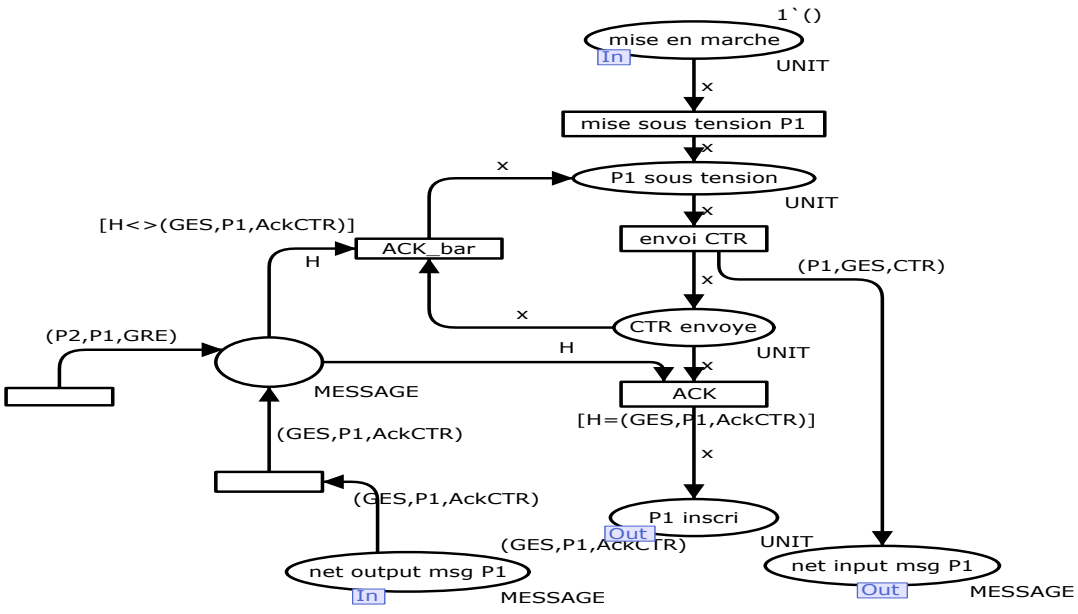

Figure III. 11. Modèle Inscription de produit actif

III.4.2. Configuration

La configuration a pour rôle de fournir au produit actif les configurations nécessaires lui permettant d'interagir correctement au sein de la même communauté de produits actifs.

Similairement au modèle précédent la place « net input msg P1» est une place de sortie et «net output msg P1» est la place d'entrée renvoyant au modèle globale du produit actif.

Les transitions « nc_ns », « nc_s », « c_ns » et « c_s » définissent l'état de configuration propre au produit. Selon le type de configuration nécessaire, la transition correspondante sera franchie et un jeton NCF approprié passe à la place « net input msg P1». Si le bon jeton CMD apparaît à la place «net output msg P1», un jeton passe à la place « P1 configuré ». Le réseau de Petri correspondant à cette partie est donné par la figure III.12.

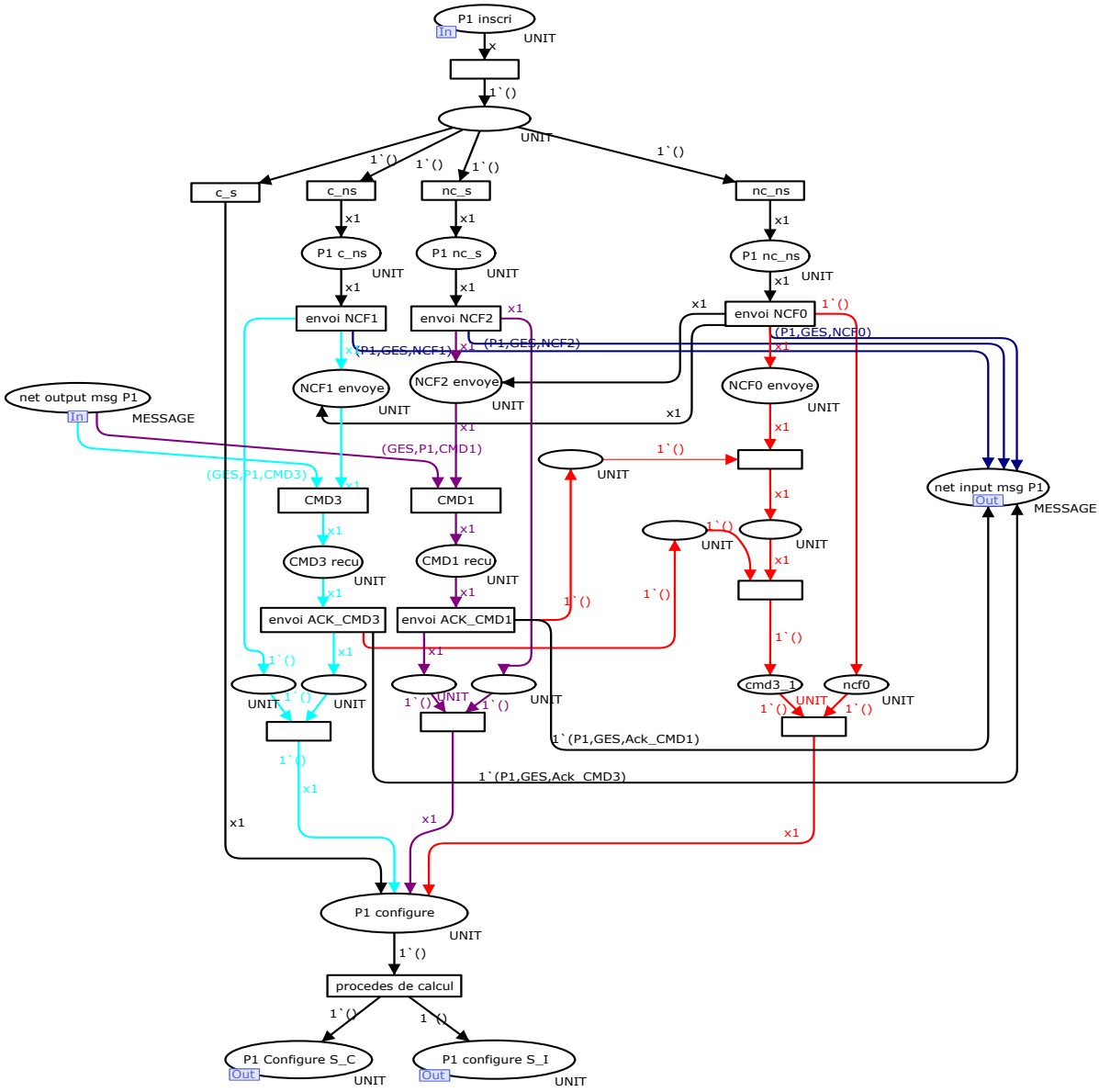

Figure III. 12. Modèle configuration de produit actif

III.5. Niveau fonction autonome de produit actif

Les deux autres transitions hiérarchiques : *surveillance et communication* et *surveillance interne*, obéissent à l'approche distribuée où chaque produit actif est doté d'une capacité de décision qui illustre la notion de réactivité. Pour la surveillance interne, le produit actif collecte à chaque fois les informations issues du capteur (température, lumière, humidité, …) et évalue pour chaque information le niveau de la sécurité. Si un niveau dangereux est atteint le produit actif envoi un message rapp_D (rapport danger) au gestionnaire afin de l'informer que l'une de ses variables a atteint un niveau critique [Zouinkhi et al., 2009a].

La transition *surveillance et communication* illustre aussi l'intelligence distribuée par la notion de sociabilité. Dans cette place les messages acceptés sont les messages reçus de gestionnaire CMD2, CMD4 et CMD5 (notion proactive) et les messages GRE et RSI reçus des autres produits actifs de voisinage.

Les messages RSI illustrent la collaboration entre produits actifs : à chaque fois qu'un produit actif reçoit un message GRE causé par le module RSSI (Received Strength Signal Indicator), ce produit actif va estimer la distance qui la sépare de l'émetteur. Cette distance sera comparée par la suite à deux valeurs L_inf et L_sup (reçus lors de la configuration) :

- Si distance > L_supp : cela indique que la distance est une distance de confort entre les deux produits
- Si L_inf < distance < L_sup : cela désigne que le produit est en mauvaise position qui se traduit par l'envoi d'un message rapp_M (rapport mauvais)
- Si D < L_inf : cela illustre un état de danger car la distance entre les deux produits est une distance critique où on a un risque d'une réaction chimique dangereuse. Donc un message rapp_D va être envoyé au gestionnaire.

Distance désigne une fonction de choix aléatoire d'un type énuméré ***dist***, et ***dist*** désigne un type qui englobe les intervalles de distance.

III.5.1. Modèle de la surveillance et communication par réseau de Petri

Le modèle de la *surveillance et communication* représenté par la figure III.13 dirige le flux des messages reçus et envoyés par rapport à un produit actif. L'aspect analyse et interaction se manifestent particulièrement dans ce modèle qui selon la nature des messages reçus et suivant leur contenu informatif, le produit actif réagit et génère des messages appropriés. La structure de ce réseau comporte des branches introduites par des transitions qui décrivent le comportement du produit.

A la réception d'un message les différentes étapes s'exécutent successivement. Tout d'abord l'analyse et le traitement de message par la transition correspondante qui sera détaillée dans la figure III.14 ensuite l'étape envoi message.

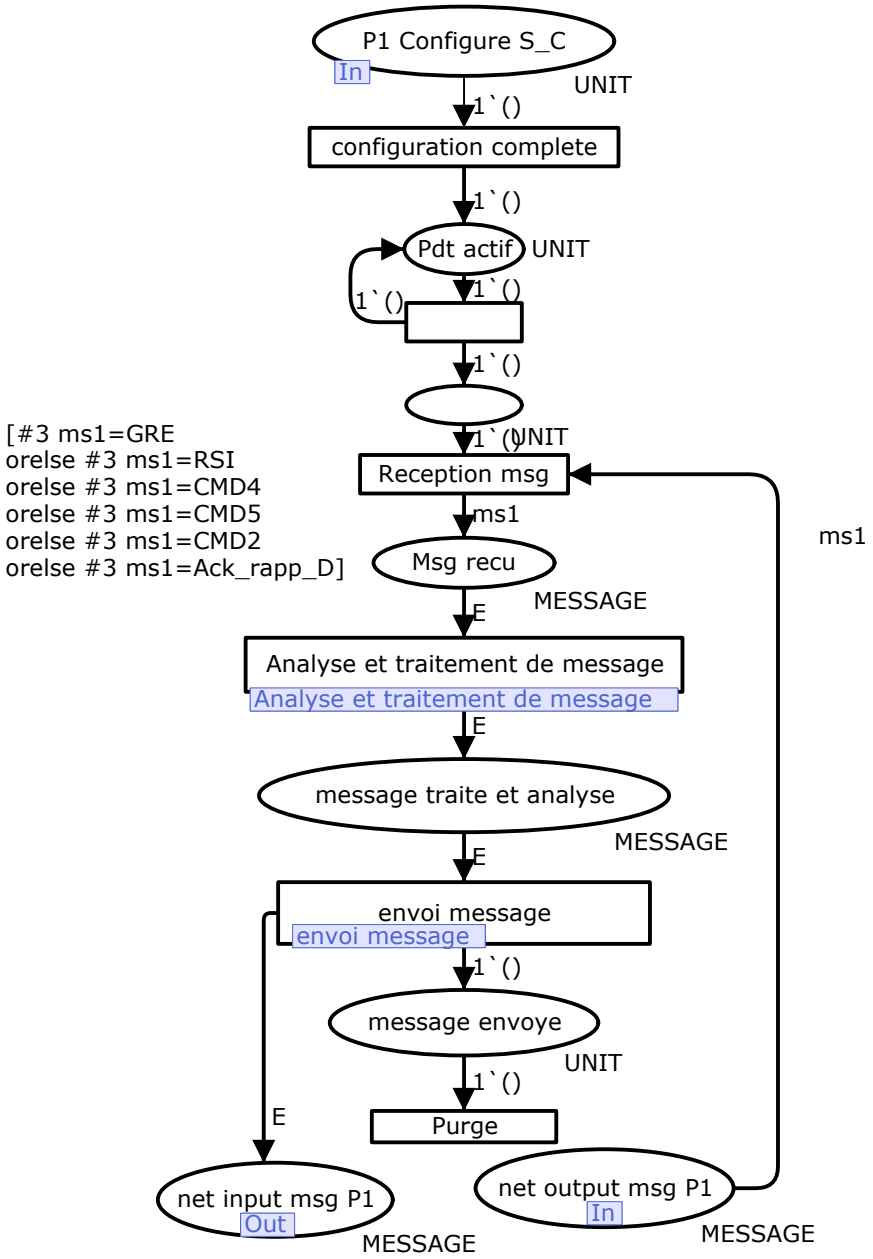

Figure III. 13. Modèle surveillance et communication de produit actif

III.5.1.1. Analyse et traitement de messages :

Le modèle hiérarchique de la transition « analyse et traitement des messages » est représenté par la figure III.14. Tous les messages reçus sont traités dans la base de connaissance.

Chaque jeton reçu symbolisant un message reçu franchira sa transition correspondante :

Transition GRE : suivant les lois incarnées par les matrices d'incompatibilité, le produit actif décide de la présence d'un état d'incompatibilité exprimé par les deux transitions « comp » (produit compatible) et « Incomp » (produit incompatible) comme l'indique la figure III .15 et qui respectivement :

- enclenche l'envoi du message RSI précisément vers le produit qui a envoyé le message GRE en utilisant les informations contenues dans les champs émetteur et récepteur figurant dans le jeton reçu (p,ds,GRE). ds est la variable référant à l'émetteur.
- entraîne le déclenchement d'une alarme et l'émission d'un message Rapp_D vers le gestionnaire en cas d'incompatibilité et distance critique entre produits.

Transition RSI : l'information que porte le message RSI sera analysée pour en déduire l'état de sécurité dépendant de la valeur de la distance séparant les deux produits actifs en question. Dans le cas où la distance touche les limites des valeurs dangereuses, un message Rapp_D vers le gestionnaire sera transmis.

Transitions CMD2, CMD4 et CMD5 : chaque transition traite une commande spécifique. Le franchissement de chacune déclenche l'envoi de message correspondant vers le gestionnaire à savoir CFG ou INA ou SER.

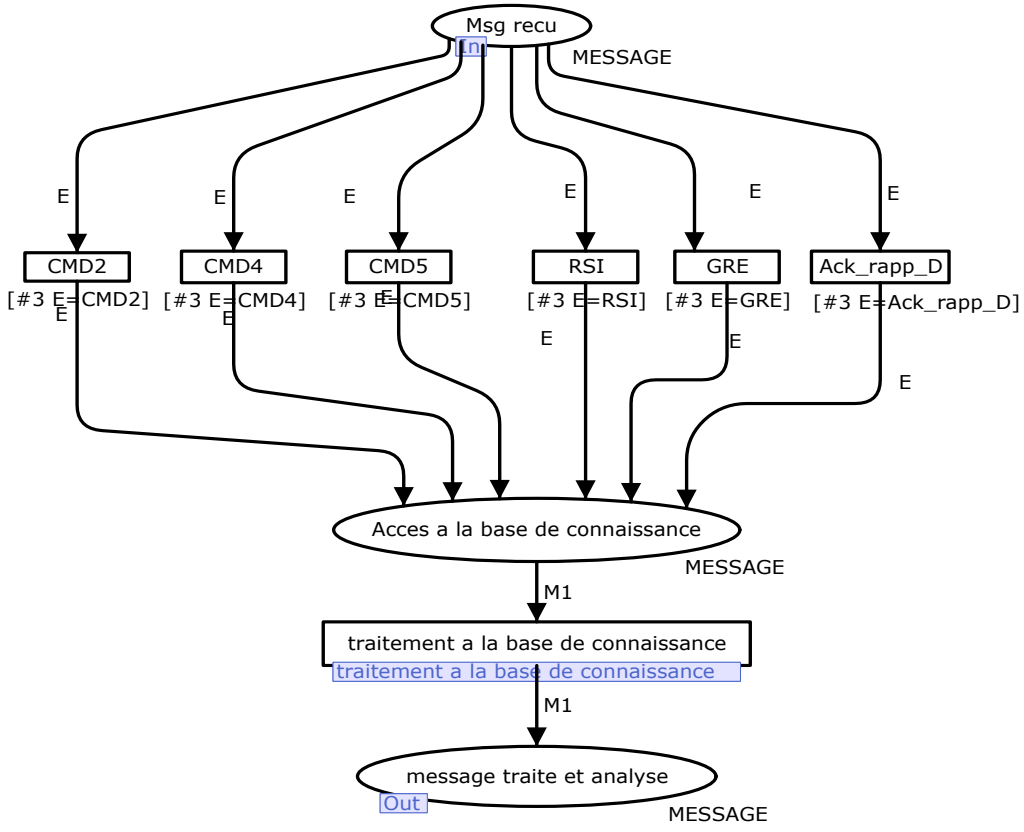

Figure III. 14. Modèle analyse et traitement de messages

III.5.1.2. Traitement à la base de connaissance :

La transition traitement à la base de connaissance représente un modèle hiérarchique qui est représenté par la figure III.15 :

Après le classement de la nature de message reçu le produit actif décide la nature de réponse. A la réception de message RSI qui indique la distance qui sépare ce produit avec son voisin qui lui est incompatible, l'état du produit sera évalué.

Cette distance sera comparée par la suite à une valeur inférieure L_inf et une valeur supérieure L_sup reçus lors de l'étape de configuration du produit.

- Si distance > L_supp : cela indique que la distance est une distance de confort entre les deux produits
- Si L_inf < distance < L_sup : cela désigne que le produit est en mauvaise position qui se traduit par l'envoi d'un message rapp_M (rapport mauvais)
- Si D < L_inf : cela illustre un état de danger car la distance entre les deux produits est une distance critique où on a un risque d'une réaction chimique dangereuse. Donc un message rapp_D va être envoyé au gestionnaire.

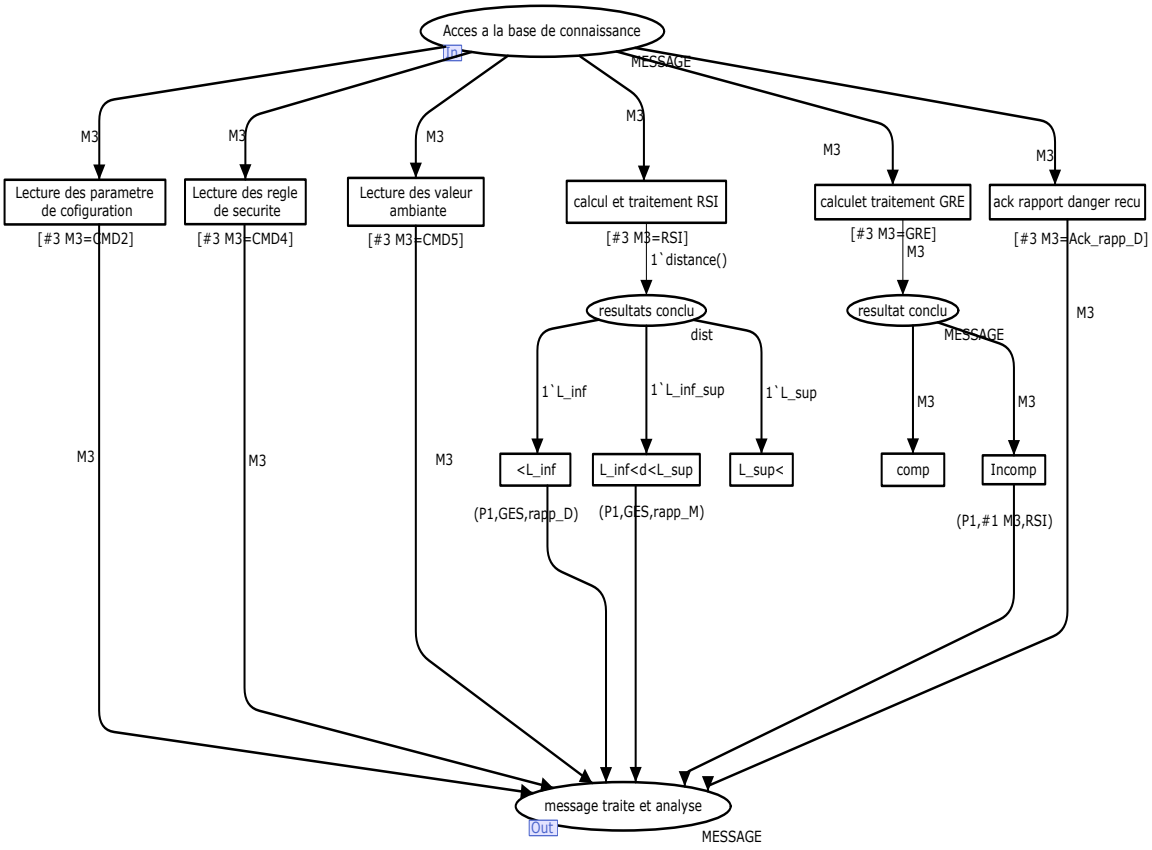

Figure III. 15. Modèle traitement de la base de connaissance

A la réception d'un message de salutation GRE, ce message sera traité et deux cas peuvent être distingués. Le produit envoyant ce message est compatible avec ce produit ou incompatible. Dans le cas d'incompatibilité, le produit actif va envoyer un message RSI afin d'évaluer la distance que les sépare.

III.5.1.3. Envoi messages :

La transition Envoi messages de la figure III.13 est détaillée par la figure suivante :

Après l'analyse de différents messages reçus, le produit actif réagit par l'envoi d'un message correspondant ;

- à la réception d'un message CMD2, le produit actif envoie un message CFG.

- Si le message reçu est de type CMD4, un message SER est envoyé.

- Si le produit actif reçoit un message CMD5, il répond par un message de type INA qui contient les informations ambiantes.

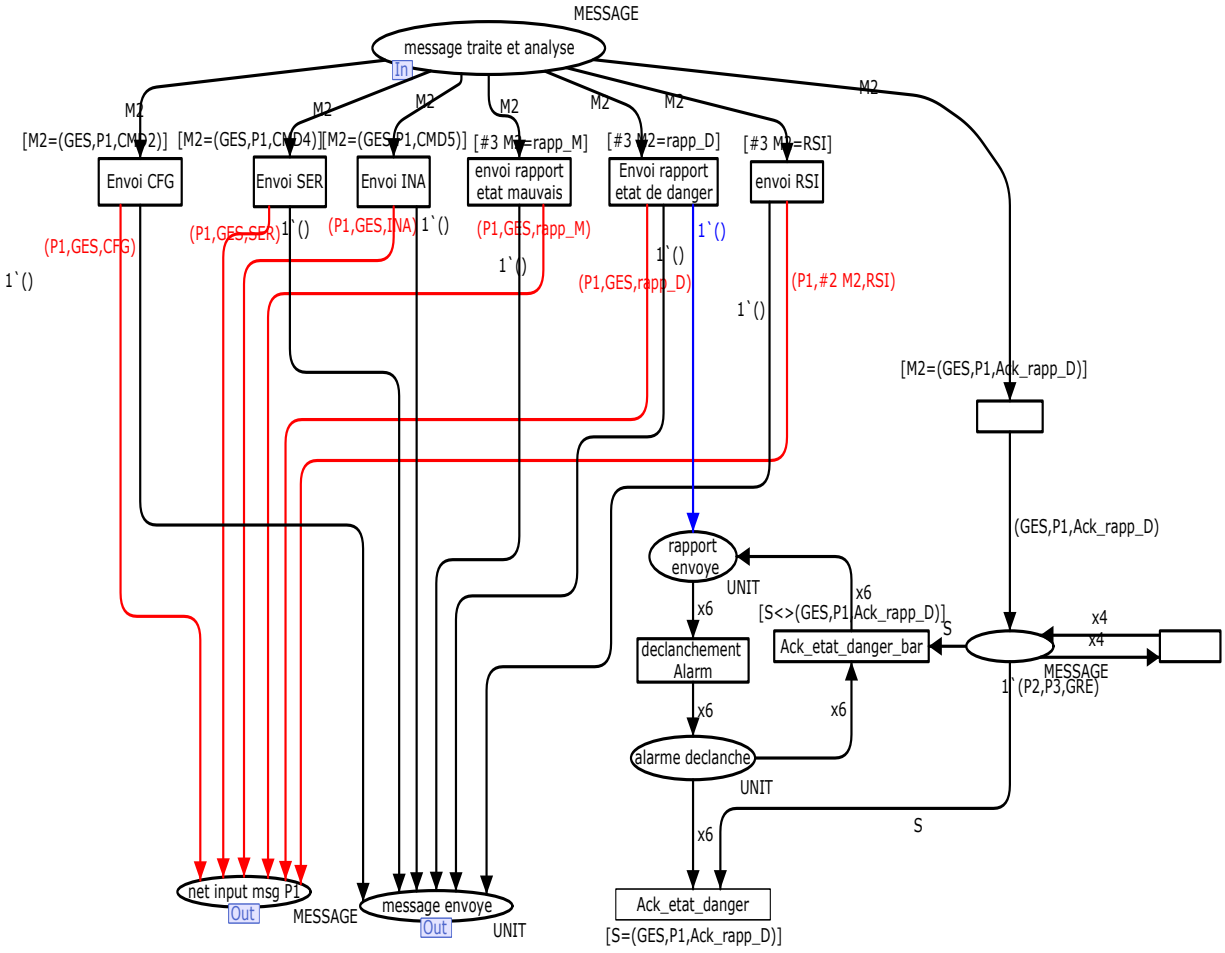

Figure III. 16. Modèle envoi messages

III.5.2. Modèle de la surveillance interne

La surveillance interne permet au produit actif de récupérer ses paramètres physiques, de les traiter et d'en déterminer le niveau de sécurité correspondant. Chaque capteur est représenté dans le modèle du réseau de Petri par une transition spécifique délivrant une valeur ambiante mesurée continuellement. La place d'entrée "calcul et diagnostique" reçoit un jeton dès que l'opération de configuration est achevée. Par suite les transitions représentant les capteurs deviennent toujours actives.

Le produit actif doit se procurer des valeurs ambiantes de façon continue. Pour ce faire les transitions se référant aux capteurs sont marquées par une temporisation permettant de générer des valeurs dans des intervalles de temps égaux. En fait, Val_ambient est une fonction permettant de générer une valeur entière aléatoire. Elle représente les valeurs mesurées par le capteur de température.

Ces mesures passent par une partie d'évaluation exprimée par les règles statiques et dynamiques dans le but d'estimer l'état de sécurité globale du produit actif. Pour cela on définit le type *ambientval* qui regroupe des valeurs entières représentant la marge de la température mesurée. Comme la représente la figure III.17, on a pris une température mesurée comprise entre 0 et 45°C.

Aussi on définit le type *etat* qui représente les trois états de sécurité qui peuvent qualifier un produit actif à savoir 'B' pour Bon, 'M' pour Mauvais et 'D' pour Dangereux.

Les règles de sécurité statiques permettent de repérer la valeur récupérée du capteur et de la faire correspondre à un niveau de sécurité prédéfini. Voici un exemple de règle statique de température :

Si la température est <10°C ou >40°C
Alors le niveau de la sécurité du produit est Dangereux (D)
Sinon Si la température est < 18°C ou > 30°C
Alors le niveau de sécurité du produit est Mauvais (M)
Sinon le niveau de sécurité est Bon (B).

Cette règle est exprimée au moyen d'une fonction en CPNTools à savoir Etat_val illustrant la règle statique relative au critère de la température. Elle juge l'état de sécurité du produit actif s'il est Bon, Moyen ou Dangereux.

Concernant les règles de sécurité dynamiques, elles décrivent l'évolution temporelle de l'état de sécurité du produit actif. La figure III.17 présente le modèle d'application d'une règle dynamique qui renvoie un état de sécurité dangereux du produit actif si une valeur de température critique est atteinte plusieurs fois au cours une période du temps.

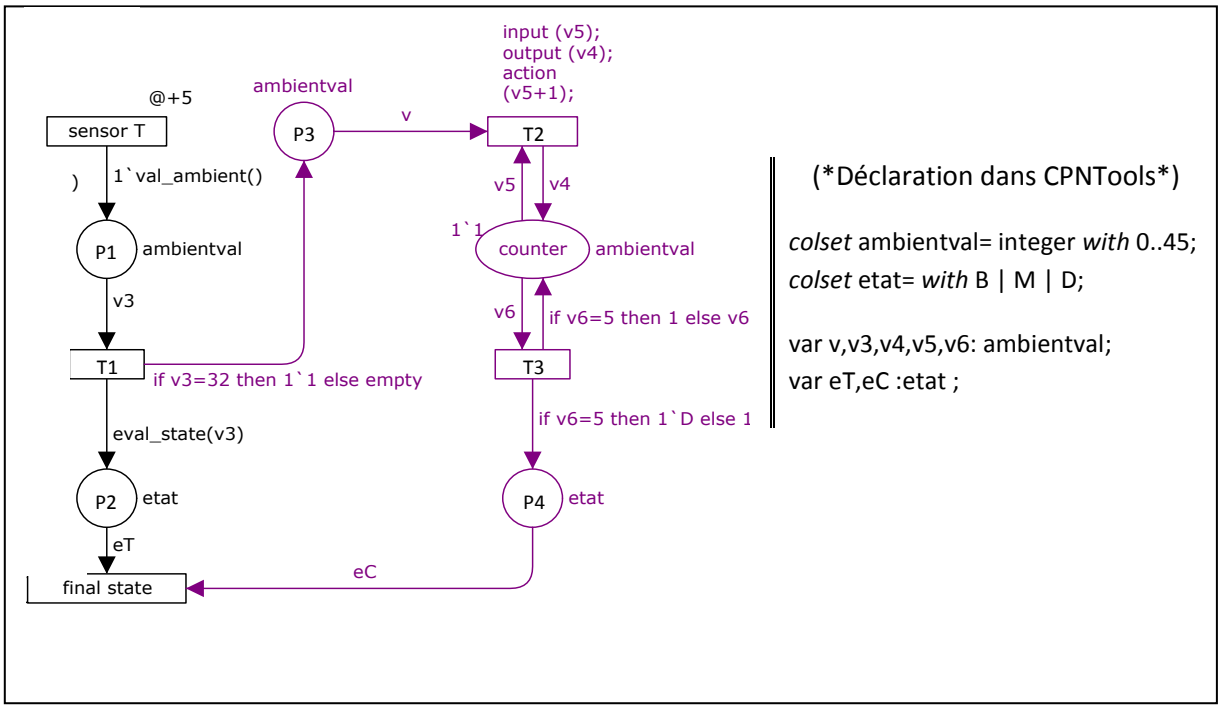

Figure III. 17. Règle dynamique

L'objectif est de compter le nombre de fois une valeur critique de température est atteinte et d'en déduire l'état de sécurité engendré. Cette opération est accomplie par le segment de code de la transition T2. Il incrémente le compteur lorsqu'un jeton de la valeur de température définie apparaît à la place P3. Le franchissement de la transition T3 dépose soit un jeton (B) soit un jeton (D). Si le compteur dépasse une valeur présélectionnée un jeton (D) figure dans la place état exprimant un niveau de sécurité dangereux.

Finalement, tous les jetons passent dans la transition état final qui a pour rôle d'analyser tous les états de sécurité des capteurs générés depuis les règles de sécurité statiques et dynamiques. Le but est de synthétiser ces états pour en estimer un état global du produit actif. La décision opère comme suit : La présence d'un seul jeton (D) donne un état global D, sinon la présence d'un seul jeton (M) donne l'état (D), sinon l'état est (B). Pour aboutir à ce résultat on a défini la fonction etat_fin.

Après la détermination du niveau de sécurité du produit actif des messages appropriés sont transmis, à savoir : un message GRE est émis si l'état du produit est en état bon (B) ou mauvais (M) puisque ce sont des états non dangereux et un message alerte ALE est envoyé si l'état est danger (D).

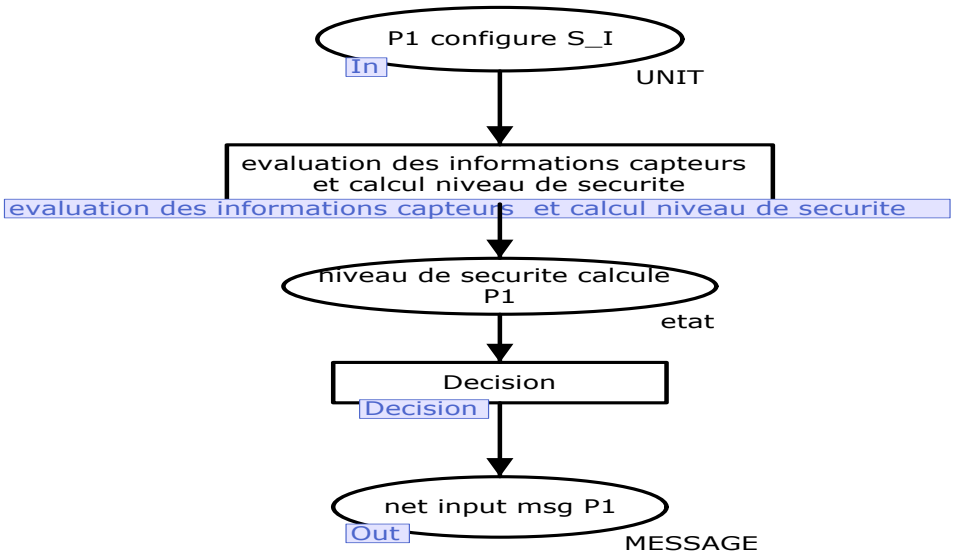

Figure III. 18. Modèle surveillance interne de produit actif

III.5.2.1. Evaluation des informations capteurs et calcul niveau de sécurité :

Le modèle inférieur de la transition « Evaluation des informations capteurs et calcul niveau de sécurité » indiquée dans la figure III.18 est représentée par la figure III.19. Chaque valeur de capteur lue sera évaluée en donnant son état bon, mauvais ou danger. Finalement le niveau de sécurité statique de produit actif dépend de l'état des valeurs captées à partir des différents capteurs.

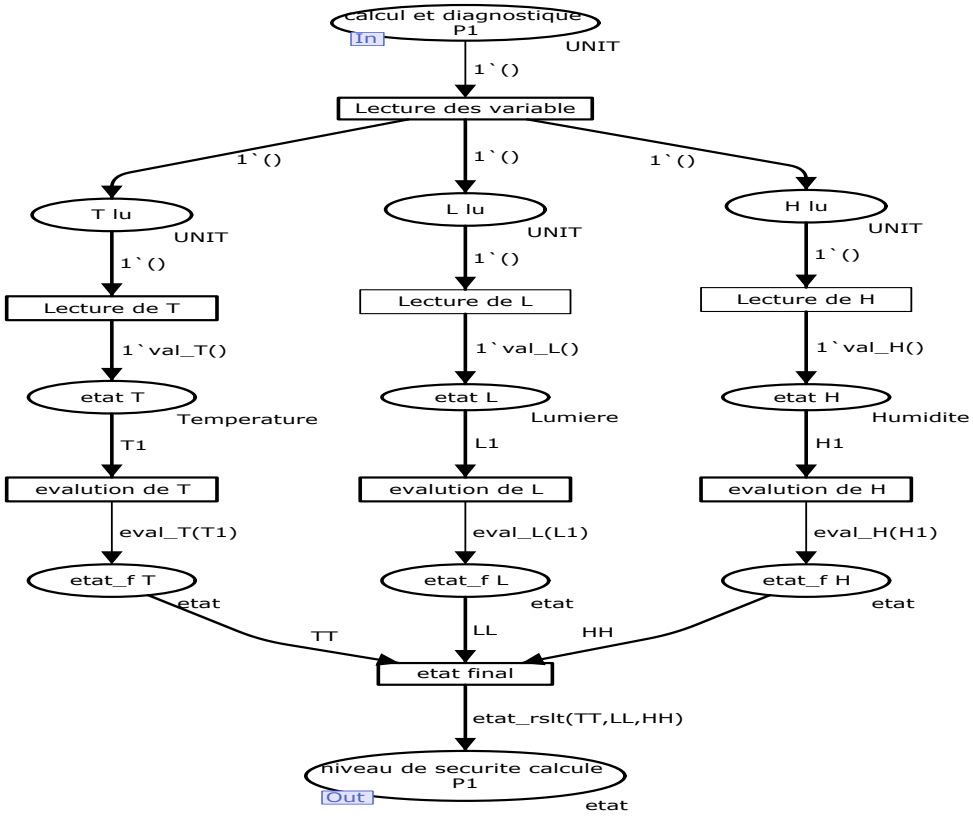

Figure III. 19. Modèle évaluation des informations capteurs

III.5.2.2. Décision :

La transition Décision de la figure III.18 représente un niveau supérieur de la figure suivante :

Selon l'état Bon, Mauvais ou Danger une décision est prise. Dans le cas de l'état danger le produit actif va envoyer une alarme de danger au gestionnaire (rapp_D) . Dans le deux autres cas le produit actif va continuer l'envoi de message de salutation (GRE) dans lequel il indique son état actuel.

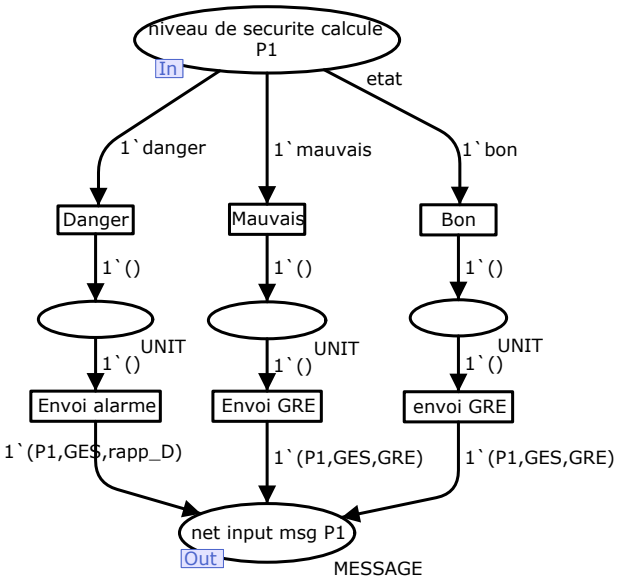

Figure III. 20. Modèle décision de produit actif

III.6. Modèle de gestionnaire

Le gestionnaire joue le rôle de serveur central, comme l'indique la figure III.21. Il est constitué d'un étage de classification, et ensuite selon le type de message, il réagie. Ce modèle peut être subdivisé en deux parties selon le concept caractérisant les systèmes multi-agents réactif ou proactif.

- Réactivité du gestionnaire [Strohbach et al., 2005]: Dès l'arrivée d'un jeton dans le buffer d'entrée du gestionnaire, une étape de classification est réalisée selon la nature du message (INA, CFG, GRE, NCF0, NCF1, Ack_CMD1, NCF2, Ack_CMD3, CTR, RAPP_D, RAPP_M, NCFOP, CMP). Suivant le message reçue, le gestionnaire doit réagir soit en mettant à jour sa base de données soit en envoyant des messages pour servir les autres produits actifs (règle de sécurité, les acquittements des rapports reçus…).

- Pro-activité de gestionnaire [Kashif et al., 2009]: Comme l'indique la figure III.21, le gestionnaire anticipe parfois en interrogeant aléatoirement (via les messages CMD5, CMD4 et CMD2) les produit actifs voisins pour connaître leurs états.

La partie proactive de la figure III.21 permet au gestionnaire de connaître les informations liées aux produits à un instant bien déterminé.

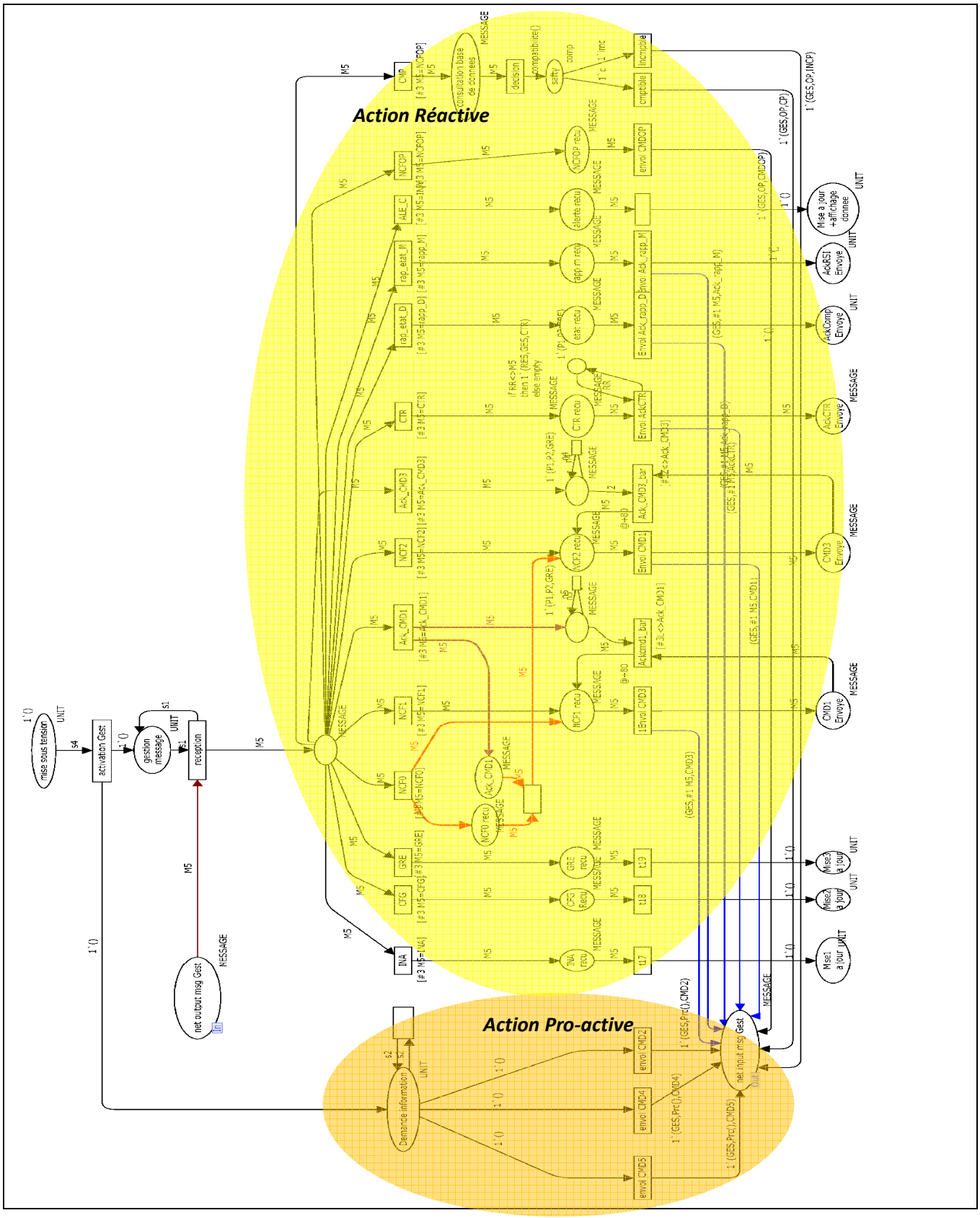

Figure III. 21. Modèle de gestionnaire

IV. Résultats et scénarios

Le RdP permet de faire l'analyse qualitative. On a donc pu vérifier que notre modèle développé fonctionne bien en termes de blocage et d'états finis à l'aide de l'outil CPN-Tools. Dans cette partie, on va créer quelques scénarios selon les règles détaillées dans le chapitre II pour voir le si comportement de nos modèles réagit bien.

IV.1. Règle statique

Dans cette partie ont va simuler la présence d'un état de danger dû à l'augmentation de la température. Comme l'indique la figure III.22, on choisit une température égale à 45°C. La valeur minimale de la température de l'état danger est évaluée à 40°C. Donc la variable état_RSLT est en état de danger et un message (P1, GES, RAPP_D) est envoyé au gestionnaire qui va lui répondre par un acquittement ACK_RAPP_D.

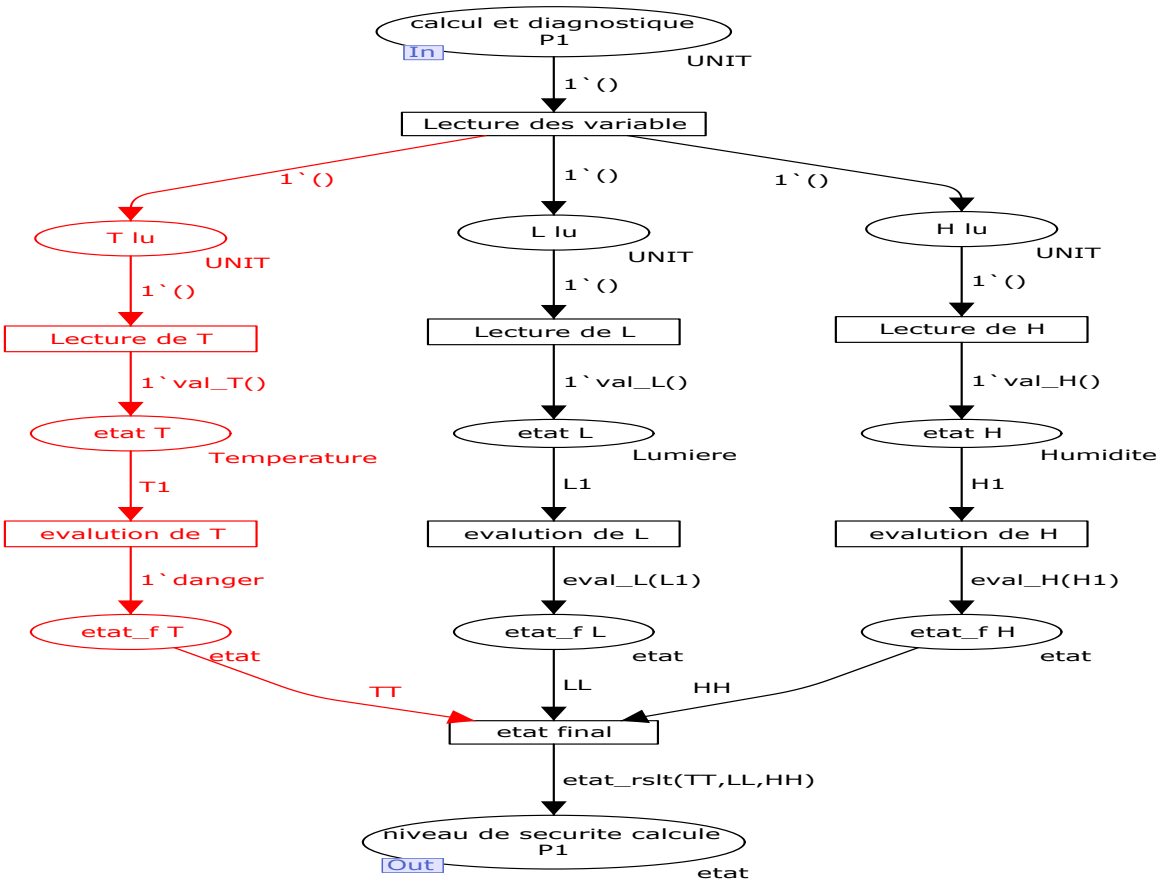

Figure III. 22. Scénario illustrant une alerte de la règle statique

IV.2. Règle dynamique

Comme l'indique la figure III.23 et à l'aide d'une fonction mémoire, chaque mauvais état qui persiste pendant 3 itérations est considéré comme un état dangereux. Dans la simulation, on suppose que la température garde la valeur 35°C et un compteur va compter la persistance de cette valeur (T1=35°>30° mauvais état de température). Si cette température reste pendant 3 itérations cela positionne la variable état_rslt en état danger. Donc le message (P1, GES, RAPP_D) est de nouveau envoyé au gestionnaire.

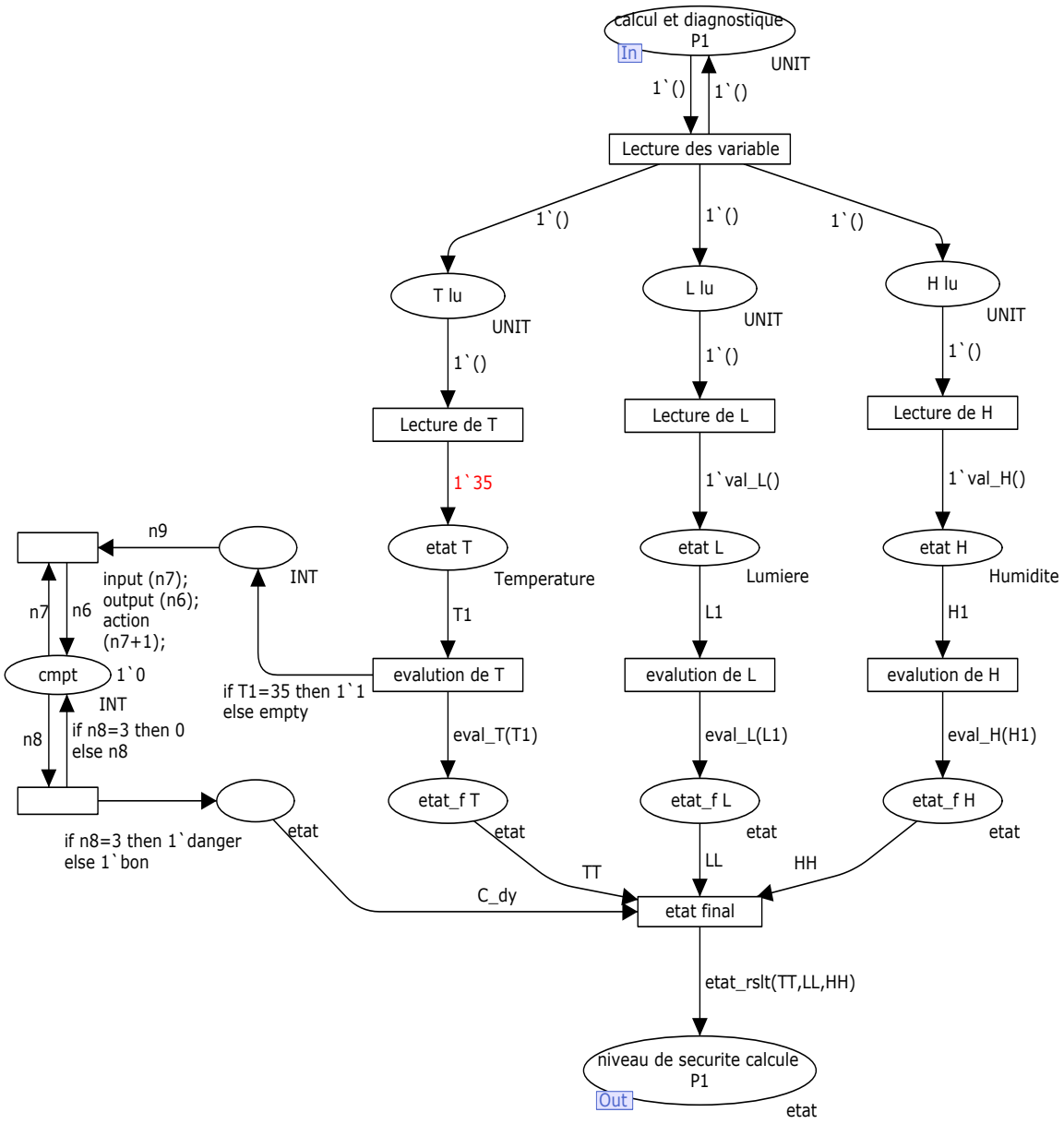

Figure III. 23. Scénario illustrant une alerte de la règle dynamique

IV.3. Déclenchement d'une alerte dûe à la proximité

Chaque produit envoie en continue des messages GRE (Greating) en mode broadcast. Ces messages sont reçus par les autres produits. Selon l'amplitude du signal, les produits vont estimer la distance qui les sépare en échangeant des messages RSSI (Received Strength Signal Indicator). Cette distance est analysée par les produits actifs. Si elle est critique, les produits actifs informent le gestionnaire par un message (P1, GES, RAPP_D).

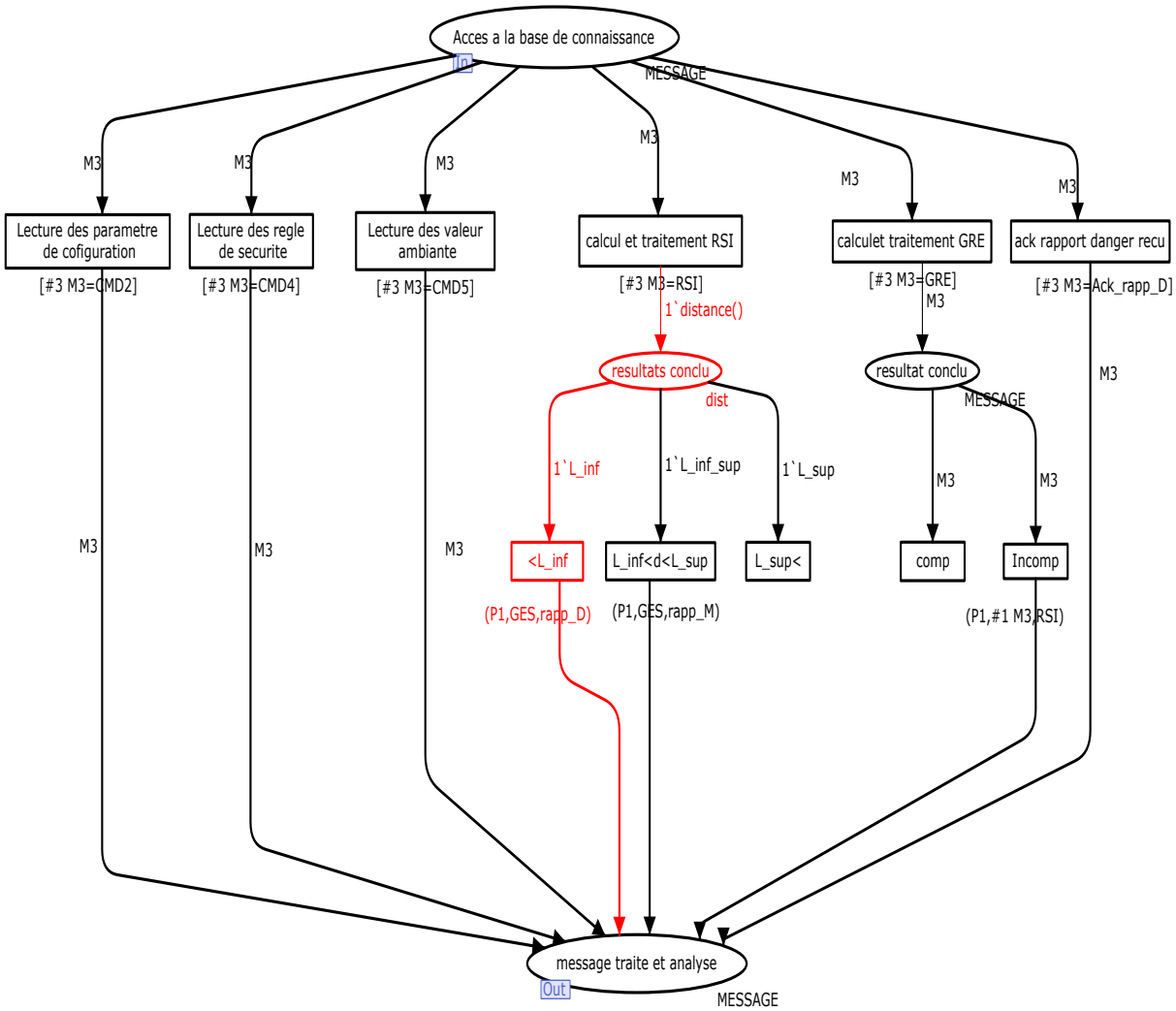

Figure III. 24. Scénario illustrant une distance défavorable entre deux produits actifs

La figure III.25, présente le modèle de coopération entre produits avec les différents niveaux de fonctions.

Cette figure illustre une vue d'ensemble des modèles et sous modèles.

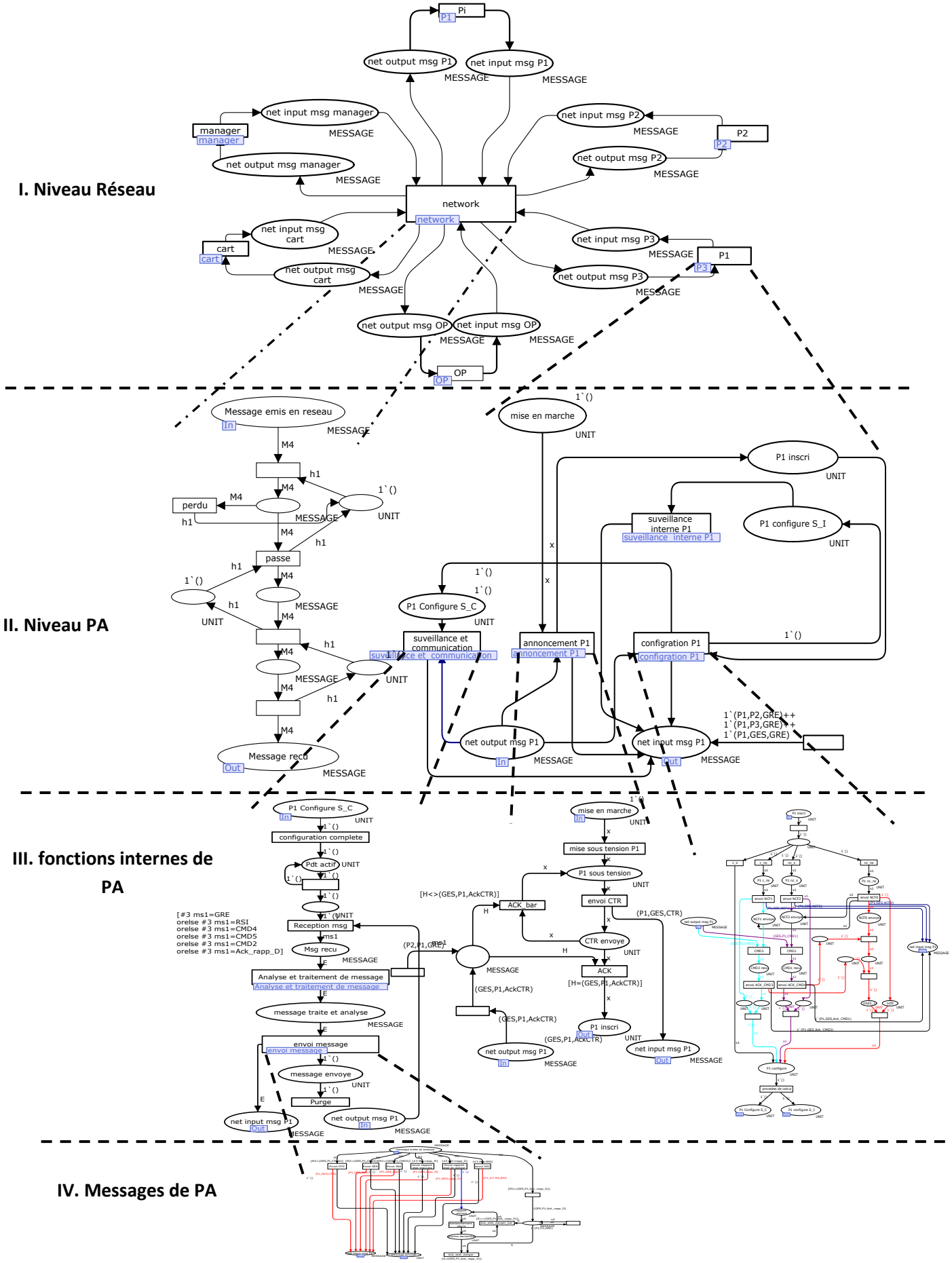

Figure III. 25. Modèle hiérarchique de la communauté de produits actifs

V. Conclusion

Dans ce chapitre on a donné une présentation générale du formalisme de modélisation par réseau de Petri ainsi qu'un aperçu sur la sémantique particulière du logiciel ouvert CPN-Tools. Ces deux outils nous ont permis de concevoir une structure complexe détaillant l'aspect comportemental du produit actif. La validité du modèle réalisé a été testée par CPN-Tools.

La validité du modèle réalisé doit être maintenant illustrée par la simulation dans un autre environnement spécifique aux réseaux de capteurs. Dans le chapitre suivant nous allons commencer par la description du travail que nous avons fait sous le simulateur Castalia/OMNeT++ et nous finissant par l'exposition des différents résultats obtenus.

CHAPITRE IV

Simulation des coopérations entre produits actifs

I. Introduction

La simulation de réseau de capteurs est un thème de recherche difficile et important. Beaucoup de compagnies et d'Universités y travaillent en utilisant des simulateurs réseaux différents qui répondent à des besoins bien spécifiques. Parmi les simulateurs existants, nous avons choisi le simulateur Castalia/OMNeT++ pour tester le fonctionnement et les performances de notre modèle de Produit Actif. Pour rappel, nos spécifications protocolaires ont été précédemment modélisées par réseaux de Petri. Cependant, il est difficile de faire l'évaluation quantitative sous Réseaux de Petri, lorsque les systèmes à étudier comprennent des centaines ou des milliers de composants. Ainsi, pour tester la robustesse de nos propositions relativement au facteur d'échelle, nous implantons notre protocole dans le simulateur Castalia/OMNET++. Celui-ci nous permet alors d'éprouver notre protocole dans des situations plus conformes à la réalité et nous permet aussi d'analyser plus précisément d'autres critères comme la consommation d'énergie.

Dans ce chapitre, nous donnons les raisons qui nous ont encouragées dans le choix de ce simulateur. Nous présentons le simulateur et nous donnons les résultats d'évaluation de notre protocole.

II. Les simulateurs existants

Un point commun entre tous les simulateurs est qu'ils permettent d'évaluer la performance de systèmes avant son déploiement. Dans le cas de la simulation des réseaux de capteurs sans fil, les objectifs de base sont très diversifiés tels que la validation des protocoles de routage ou des protocoles de sécurité.

Dans notre cas, nous sommes intéressés à valider le comportement des capteurs sur lesquels on a installé le modèle interne de produit actif.

De nombreux outils de simulation sont utilisés dans les recherches portant sur les réseaux de capteurs. [Giel et al., 2008]. Il existe les outils d'émulations comme Avrora [Titzer et al., 2005] et TOSSIM[19] et les environnements de simulation comme OMNeT++, OPNET et NS2.

II.1. NS2

C'est un simulateur à événements discrets, écrit en C++ avec une interface TCL. Il est proposé dans le domaine des recherches sur les réseaux. Il est gratuit et open-source. Des améliorations sont en cours sur ns2 avec en parallèle le développement de NS3.

[19] http://www.tinyos.net

Les nœuds dans le simulateur NS2 modélisent des piles protocolaires OSI complètes qui ne sont pas nécessaires pour représenter des nœuds de capteurs. En effet, un ensemble de protocoles de routage fixe, des protocoles de transport et des modèles d'application (comme des services web) sont fournis mais ils ne sont pas utilisés par les réseaux de capteurs. Enfin, la simulation des réseaux de capteurs n'est pas facilement supportée par NS2 même si plusieurs travaux sont actuellement en cours pour que NS2 supporte mieux la simulation des réseaux sans fil. Notamment, [Xue et al., 2007] ont évalué la performance du simulateur NS2 dans le cadre des RCSF. Ils ont montré que NS2 ne simule pas très bien les RCSFs. Aussi, ils ont proposé des modifications pour améliorer les modèles RCSFs sous NS2.

II.2. OPNET

OPNET[20] Modeler est un simulateur de réseaux commercialisé par Opnet Technologies, Inc. Une interface utilisateur graphique supporte la configuration des scénarii et le développement des modèles de réseau. Trois niveaux hiérarchiques sont définis pour la configuration : le niveau réseau qui crée la topologie du réseau à simuler, le niveau nœud qui définit le comportement du nœud et qui contrôle le passage des données entre les différents éléments fonctionnels à l'intérieur du nœud, et le niveau processus, qui décrit les protocoles qui sont représentés par des machines à état finis. Le code source est C/C++. L'analyse des données simulées est effectuée avec un ensemble de fonctions intégrées.

Le simulateur OPNeT comporte plusieurs fonctionnalités adaptées et performantes pour la simulation des réseaux de capteurs..

II.3. OMNeT++

OMNeT++ [Varga, 2001] [Varga et al., 2008] est un simulateur inspiré d'OPNet. C'est un composant modulaire et un environnement de simulation avec un support GUI. OMNeT++ fournit une architecture hiérarchique. Les modules sont programmés en C++, le GUI est crée en utilisant la librairie Tk. Les modules sont assemblés sous forme de composants et de modèles en utilisant un langage de haut niveau. Ils communiquent entre eux en envoyant des messages. La configuration d'une simulation est gérée par des fichiers .ini.

Aujourd'hui, il y a plusieurs modèles de réseaux de capteurs basés sur OMNeT++. Le composant mobile (Mobility Framework) implémente un support pour la mobilité des nœuds,

[20] http://www.opnet.com

la gestion dynamique des connexions et un modèle de canal sans fil. Maintenant, il ne fournit que des modèles pour l'IEEE 802.11. Un modèle pour le 802.15.4 a été développé mais il n'a pas été pour le moment publié.

III. Le simulateur Castalia

Castalia est un simulateur de réseaux de capteurs sans fil (RCSF), de réseaux BANs (Body Area Networks) et généralement de réseaux de composants à puissance limitée. Il est basé sur la plateforme OMNeT++ [Yesid et al., 2009] [Boulis, 2009]. Il est utilisé par les chercheurs et les développeurs pour tester des algorithmes distribués et des protocoles avec des composants réels de réseaux de capteurs comme le canal sans fil, le modèle radio, et le comportement des nœuds lié à l'accès au canal radio. Castalia a été développé à NICTA[21] (National ICT Australia) en 2006. En 2007, il est devenu publique en tant qu'un projet open source sous la licence publique académique. Notre implémentation du modèle interne des Produits actifs a nécessité une compréhension approfondie des différents modules qui le constituent. L'étude du code source de Castalia a été nécessaire pour l'adapter à nos besoins. Toutes les informations données dans cette partie sont obtenues à partir du manuel d'utilisation du simulateur Castalia.

La structure du code source de CASTALIA est hiérarchique. Chaque module est un répertoire qui contient des sous répertoires. Dans le cas d'un module simple, le répertoire contient du code C++ pour décrire le comportement du module, un fichier *.ned pour définir la structure du module et un fichier *.msg pour spécifier les paquets que génèrent ce module. Un nœud dans un réseau de capteurs sous CASTALIA est un module composé essentiellement d'un canal sans fil, de la couche physique radio et de la couche MAC, comme l'indique la figure IV.1. Les flèches pleines représentent les messages échangés. Et les flèches hachées sont des appels de fonction du module.

Les développeurs de Castalia ont implémenté toutes les caractéristiques et les détails des composants d'un réseau de capteurs dans leur simulateur.

[21] http://www.nicta.com

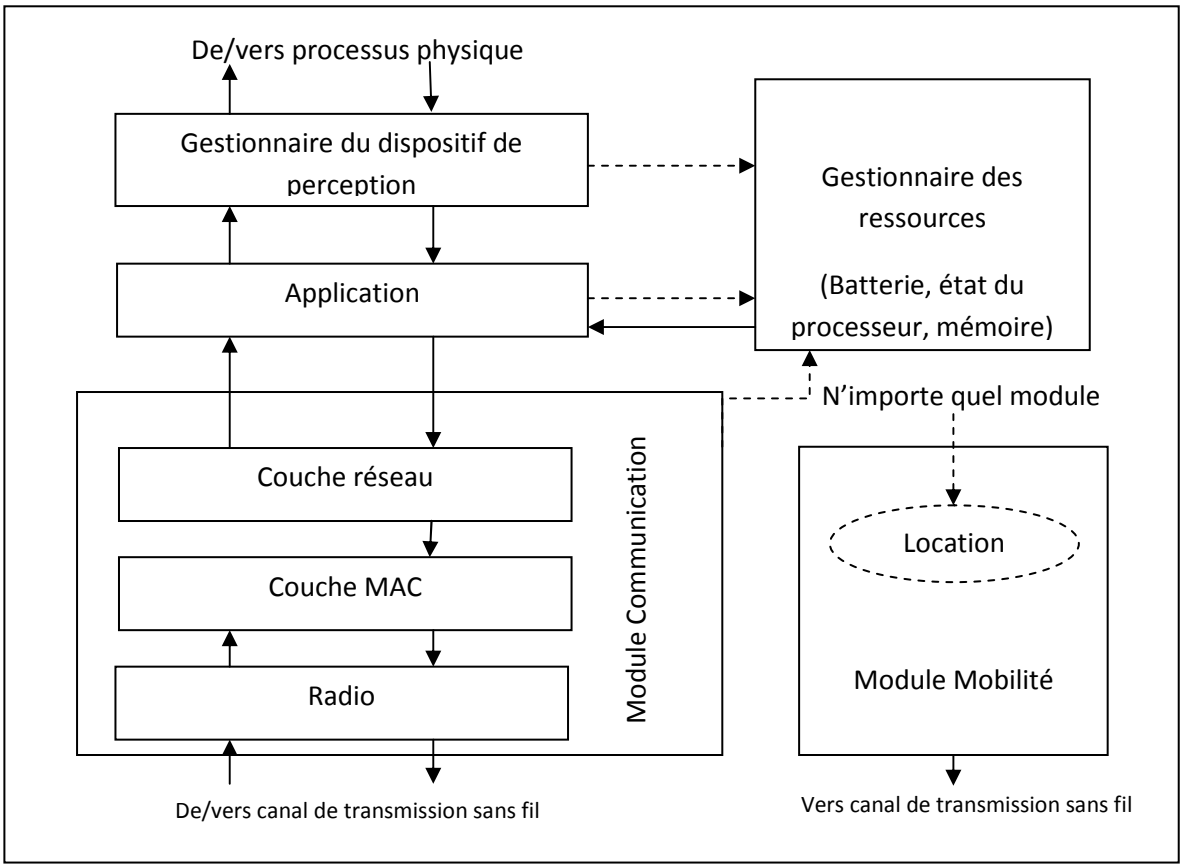

Figure IV. 1. Structure d'un nœud sous Castalia.

Le modèle du canal sans fil est basé sur le travail de [Zuniga and Krishnamachari, 2004]. L'équation qui donne l'affaiblissement de parcours PL(d) (Path Loss) à la distance d est :

$$PL(d) = PL(d_0) + 10 * n * \log(\frac{d}{d_0}) + X_\sigma \qquad \text{(IV.1)}$$

Avec $PL(d_0)$: affaiblissement du parcours connu pour une distance d_0

σ : exposant de l'affaiblissement du parcours

X_σ : variable gaussienne avec une déviation standard σ.

L'interférence est calculée dynamiquement à partir des différents nœuds transmetteurs. Et de la même façon, on peut calculer les probabilités de réception des paquets.

Avec le simulateur Castalia, on peut choisir le modèle de collision parmi trois types :

- Pas de collision,
- Modèle simple de collision : si deux nœuds ont transmis des paquets, il y a occurrence d'une collision chez le récepteur

- Modèle additif de collision : les transmissions à partir des autres nœuds sont calculées en tant qu'interférence.

Nous avons choisi le troisième type du modèle de collision pour avoir un trafic de transmission plus proche de la réalité.

III.1. Le module radio :

Ce module supporte plusieurs états : en transmission, en réception/attente, en veille. Il est possible de faire varier les puissances et les délais de transmission pour les différents états.

III.2. Le module MAC

Le module MAC (Medium Access Control) est une partie importante dans le comportement du nœud. La première motivation dans le développement de Castalia a été de tester le réglage du protocole MAC dans des conditions réalistes du canal radio. Pour le moment, il n'y a que ce protocole (Tunable_MAC) qui est implémenté. D'autres protocoles MAC (exp : SMAC, TMAC, BMAC) pourront être implémentés dans le futur.

Les paramètres les plus importants de cette couche sont :

- Cycle d'utilisation : c'est la fraction du temps que le nœud passe dans l'écoute du canal. La valeur (1 – cycle d'utilisation) est la fraction de temps de veille. Le réglage de ce paramètre est important pour minimiser la consommation d'énergie.
- Intervalle d'écoute : c'est la période d'écoute du canal pour un nœud.

Pour limiter le risque, on utilise le protocole CSMA/CA : Le protocole CSMA/CA utilise un mécanisme d'esquive de collision basé sur un principe d'accusé de réception réciproque entre l'émetteur et le récepteur :

La station voulant émettre écoute le réseau. Si le réseau est encombré, la transmission est différée. Dans le cas contraire, si le média est libre pendant un temps donné (appelé DIFS pour *Distributed Inter Frame Space*), alors la station peut émettre. La station transmet un message appelé Ready To Send (ou *Request To Send*, noté RTS signifiant prêt à émettre) contenant des informations sur le volume des données qu'elle souhaite émettre et sa vitesse de transmission. Le récepteur (généralement un point d'accès) répond par un *Clear To Send* (CTS, signifiant *Le champ est libre pour émettre*), puis la station commence l'émission des données.

À la réception de toutes les données émises par la station, le récepteur envoie un accusé de réception (ACK). Toutes les stations avoisinantes patientent alors pendant un temps qu'elles considèrent être celui nécessaire à la transmission du volume d'information à émettre à la vitesse annoncée.

III.3. Le module réseau :

À partir de la version 1.2, le simulateur introduit le module réseau (routage). Au début, ils n'ont pas donné une importance à cet élément, donc il n'y avait pas un module pour le routage. L'utilisateur doit alors traiter le routage des paquets dans le module application s'il en a besoin.

III.4. Le module Processus physique :

Pour avoir un environnement de simulation proche de la réalité, on a besoin d'avoir des modèles flexibles de processus physique (corrélation spatiale des données, variabilité dans le temps, …). Pour ce besoin, le simulateur offre un module générique pour le processus physique pour fournir les données aux nœuds de capteurs. La base de ce modèle est les valeurs des sources des dont l'influence est diffusée dans l'espace. Les sources peuvent changer leurs positions et leurs valeurs. L'effet des sources multiples dans un point est additif.

Le modèle qui détermine la valeur du processus physique dans un emplacement et un temps bien défini est :

$$V(p,t) = \sum_{tous\_les\_sources\,i} \frac{V_i(t)}{(K.d_i(t)+1)^a} \qquad \text{(IV.2)}$$

Avec : V(p, t) donne la valeur du processus physique dans un point (p) et un temps (t),

$V_i(t)$: la valeur de l'i^{ème} source à temps t,
$d_i(t)$: la distance du point p de la i^{ème} source
K, a : paramètres qui déterminent la manière de diffusion de la valeur de la source
Dans notre cas, il est intéressant d'avoir une méthode plus simple pour contrôler plus efficacement les valeurs fournies aux capteurs comme le fait de pouvoir faire évoluer une valeur de capteur qui va enclencher une alerte suivant son état.

III.5. Module du gestionnaire du dispositif de perception (Sensing device manager)

C'est un module intermédiaire entre le module application et le module processus physique (mesure captée). L'application envoie les requêtes de lecture des valeurs ambiantes vers ce module, et ce dernier accède au processus pour lire la valeur. Ce module permet de simuler la réception de données (événements). Il faut savoir que dans le cas pratique, les valeurs proposées par le dispositif physique (détaillé dans la section suivante) sont faussées par l'inexactitude du dispositif de perception. Castalia propose un ensemble de paramètres pour modéliser cette inexactitude. Dans Castalia, il y a une seule correspondance entre le processus physique et le dispositif de perception. En pratique, un unique processus physique peut déclencher plusieurs dispositifs de perception. Par exemple : une alarme d'explosion peut être déclenchée par des microphones, des capteurs sensoriels et des capteurs de température. Mais encore, un capteur peut être affecté par les conditions environnementales du champ de captage. C'est à dire que plusieurs événements (changement climatique ou incendie de forêt) peuvent changer un processus physique comme par exemple la température. Dans le modèle que je vais utiliser, je considère que le dispositif de perception intercepte un seul type de valeur. C'est-à-dire que le dispositif de perception est connecté à un seul module processus physique (la température).

III.6. Le module du gestionnaire des ressources :

Ce module sauvegarde des traces des ressources des différents nœuds, et surtout l'énergie.
Il soustrait linéairement l'énergie demandée par les différents modules, et il effectue des statistiques sur l'utilisation de la mémoire. Des modèles gèrent les charges des batteries.

III.7. Le module application

C'est le module dans lequel, l'utilisateur effectue normalement beaucoup de changements pour implémenter de nouveaux algorithmes. Dans ce module, j'ai défini les messages du modèle du produit actif.

III.8. Le module mobilité

Le module de mobilité spécifie comment les nœuds bougent dans l'espace. Les autres modules peuvent y accéder à n'importe quel moment. Des notifications périodiques sont envoyées au canal sans fil pour donner la position des nœuds.

III.9. Les projets effectués utilisant Castalia :

[Tschirner et al., 2008] ont utilisé Castalia/OMNET++ pour démontrer que les techniques de base de la modélisation (comme le modèle testeur de UPPAAL[22]) peuvent être considérées comme une approche complémentaire pour la conception et l'analyse des réseaux de capteurs. Leur but était de comparer des résultats données par le simulateur Castalia/OMNET++ avec ceux obtenus sous UPPAAL. Le réseau étudié comporte 9 nœuds dont un est le puits. Ces capteurs sont de type CC2420. Ils ont utilisé UPPAAL pour chercher les paramètres temporels et pour valider les propriétés de la qualité de services. Le simulateur et le modèle de test sur UPPAAL sont utilisés pour analyser le comportement du réseau. Pour démontrer la faisabilité de la technique, ils ont étudié le ratio de la distribution des paquets et la qualité de la connexion du réseau. Le simulateur est configuré avec le protocole 802.15.4. Et ils ont analysé une application médicale composée d'un Électrocardiogramme et d'un thermomètre qui envoient périodiquement des messages sur le réseau sans-fil. La comparaison faite démontre que les résultats trouvés avec UPPAAL coïncident avec ceux trouvés en utilisant le simulateur OMNeT++. Leurs expériences démontrent qu'UPPAAL peut être aussi utilisé comme un outil complémentaire dans la simulation et le réglage de paramètres.

[Meier et al., 2008] ont utilisé le simulateur Castalia pour la simulation de réseaux de capteurs pour une application de surveillance, appelé DiMo (Distributed Node Monitoring in Wireless Sensor Networks). DiMo est la première solution pour le contrôle d'un réseau de capteurs destinée aux applications avec des détections d'événements. Ils ont comparé cette solution avec d'autres en observant l'occurrence de fausses alertes. C'est-à-dire, on suppose qu'un nœud est en échec alors qu'il est en marche et que cette erreur d'interprétation est liée aux pertes de paquets. DiMo utilise les nœuds « relais » dans l'observation. Ils ont utilisé d'autres mesures de performance comme la consommation d'énergie.

[Pham et al., 2007] ont essayé de valider le simulateur Castalia en comparant les résultats trouvés par simulation avec ceux trouvés expérimentalement. Le réseau simulé se compose de 9 capteurs de type TelosB (radio CC2420) sur une surface fermée 70mx90m. Pour gérer le comportement des nœuds et générer les résultats, ils les ont programmés de façon à recevoir des commandes en communication radio (wirelessly). Les commandes peuvent changer les paramètres MAC des nœuds, et aussi commander un nœud à envoyer et recevoir des données venant d'autres nœuds. Dans chaque simulation, ils ont élaboré un plan d'expérience autour

[22] http://www.uppaal.com

de 8 paramètres (cycle d'utilisation, état veille, état écoute, intervalle d'écoute, retransmissions, puissance de transmission, probabilité de transmission, intervalle de retransmission). La conclusion de leurs travaux est que les résultats expérimentaux sont différents de ceux obtenus par simulation. Leur prochaine étape est alors l'analyse et l'explication de ces différences et de corriger les anomalies dans le code du simulateur Castalia. Selon [Pham et al., 2007], ce type de travail doit se poursuivre de façon à aboutir à un simulateur acceptable pour les réseaux de capteurs.

Pour conclure, malgré ces défauts, on a choisi comme simulateur Castalia sous plateforme OMNET++ car il est le plus avancé et est le plus utilisé dans la communauté scientifique pour la simulation de réseaux de capteurs sans fil.

IV. Le diagramme de classe

Le simulateur Castalia utilise un diagramme de classe pour modéliser ses composants. Aussi, avant de commencer à implémenter le modèle interne du produit actif, nous avons décomposé notre modèle de produit actif en classes. La figure 33 est le diagramme de classes de l'application à implémenter sous Castalia. On différencie dans nos travaux trois groupes de classes dans ce diagramme :

IV.1. Le premier groupe de classes :

Ce sont les classes qui existent déjà sous Castalia et qui n'ont pas été modifiées :

- BypassRoutingModule : cette classe implémente les fonctionnalités de la couche réseau. Sous Castalia 2.0, on a le choix entre trois types de routage :
  - BypassRouting : c'est un algorithme de routage trivial. Il y a deux destinations possibles pour un paquet : BROADCAST (diffusion) ou l'adresse d'un nœud destinataire MultipathRingsRouting : le protocole sauvegarde une trace du chemin traversé (les adresses des nœuds traversés).
  - SimpleTreeRouting : dans cette implémentation et celle du multipathRingsRouting, les messages sont retransmis d'un nœud à l'autre jusqu'à ce qu'il atteigne le puits. Les nœuds sont reliés de façon à former un arbre : chaque nœud a des nœuds parents et à travers eux, il communique avec les autres.

On a choisi d'utiliser la classe BypassRouting puisqu'elle contient les fonctionnalités de base de routage nécessaires à notre application.

- TunableMacModule : elle implémente la couche Mac (couche liaison), on peut choisir entre trois protocoles :
  - BypassMac : cette classe n'implémente que les fonctionnalités de base d'une couche MAC.
  - TunableMac : ils ont ajouté dans cette classe le mécanisme du protocole CSMA : le temps est composé en intervalles égaux. Le noeud actif ne peut accéder au media que dans des fractions d'intervalles bien définies selon la capacité d'énergie dédiée à chaque produit. Cette fraction est définie par le paramètre dutyCycle.
  - TMac : TMac est un protocole très utilisé dans les réseaux de capteurs.

- RadioModule : ce module a pour intérêt de réaliser les caractéristiques radios de faible capacité comme celui utilisé dans les plateformes réseaux de capteurs. Il supporte différents états (transmission, réception/écoute, veille), avec des puissances de consommation et de délais différents d'une transmission entre deux nœuds.

- RessourceGenericManager : garde des traces de différentes ressources des nœuds. Les autres classes l'utilisent lorsqu'ils ont besoin des ressources (exp. l'énergie).

IV.2. Le deuxième groupe de classes :

Ce sont les classes qui existent sous Castalia mais qui ont été modifiées :

- WirelessChannel : Castalia[23] est le simulateur le plus proche d'un fonctionnement de canal sans fil. Cette classe contient les méthodes et les propriétés nécessaires pour qu'elle se comporte comme un vrai canal sans fil. Cette classe traite des statistiques sur les paquets envoyés et les paquets perdus provoqués notamment par des interférences. On a ajouté à la fonction `finish() de cette classe` l'affichage de la probabilité de pertes des paquets (totalRxFailedNodes/transmissions) pour un nombre de nœuds donnés (numNodes).

- SensorDevMgrModule : la classe applicationModule utilise cette classe pour accéder aux phénomènes surveillés (Physical Process). Généralement, elle reçoit des requêtes de lecture des valeurs ambiantes à partir de la classe applicationModule sous forme des messages de type (`APP_2_SDM_SAMPLE_REQUEST`), et elle répond aussi par des messages de type (`SDM_2_APP_SENSOR_READING`). Dans le reste de ce chapitre, on utilise l'abréviation « SDM » pour désigner ce module.

- CustomizablePhysicalProcess : cette classe correspond au composant physique qui surveille l'environnement. Ce composant est le capteur. Pour cela, ce composant doit avoir

[23] www.castalia.npc.nicta.com.au

toutes les caractéristiques souhaitées liées à un capteur. On a ajouté ici plusieurs attributs décrits comme des paramètres de configuration dans le modèle interne des produits actifs. La liste des attributs ajoutés sont les suivants : securityLevelTable, electronicCodeTable, symbolTable, substanceNameTable, substanceIdTable, addressTable, vMaxTable, vMinTable, dMaxTable, dMinTable. Ce module est appelé « Processus physique ».

IV.3. Le troisième groupe de classes :

Ce sont les classes ajoutées. Dans notre cas, on a ajouté une seule classe :

- **ApplicationModule :** cette classe représente la partie essentielle de l'implémentation du modèle. Cette classe contient les mêmes attributs ajoutés à CustomizablePhysicalProcess avec le paramètre theRSSI (Received Signal Strength Indicator) [Bahl and Padmanabhan, 2000] [Niculescu and Nath, 2001] [Savarese et al., 2002] qui correspond à la valeur du RSSI envoyé avec chaque message. La figure IV.2 donne un extrait du fichier application_CMD1Packet.msg qui contient la structure du paquet CMD1.

```
message application_CMD1Packet extends App_GenericDataPacket
{
fields:
        string substanceId;
        string substanceName;
        int symbol;
        string electronicCode;
        string address;
};
```

Figure IV. 2. Code du message CMD1

Dans la méthode handleMessage, on a ajouté le traitement approprié à chaque message reçu.
Ex : après la réception d'un message CMD1, le produit actif doit mettre à jour ses paramètres de configuration en utilisant ceux envoyés dans le paquet CMD1. S'il possède aussi les règles de sécurité, il peut alors commencer à lire les valeurs ambiantes et envoyer le message de salutation GRE (voir figure IV.3).

```
case CMD1:
{
  printf("node: %d -> received APP_DATA_PACKET(CMD1) from %d at %f\n", self,
  atoi(msgSender.c_str()), simTime());
  if (self == 0)
    printf("ERROR: CMD1 must be sent by the supervisor to the node\n");
  else
  {
    cmd1Received = true;
    valuePropagation_CMD1Packet *dataPacket;
    dataPacket = check_and_cast<valuePropagation_CMD1Packet *>(msg);
    theSymbol = dataPacket->getSymbol();
    theSubstanceName = dataPacket->getSubstanceName();
    theSubstanceId = dataPacket->getSubstanceId();
    theElectronicCode = dataPacket->getElectronicCode();
    theAddress = dataPacket->getAddress();
    isConfigured = true;
    send2NetworkACK(msgSender.c_str() ,ACKCMD1, 1);
    if (hasSecurityRules == true)
    {
      printf("node: %d is now Configured and it has SecurityRules at %f\n", self,
      simTime());
      scheduleAt(simTime()+ DRIFTED_TIME(APP_SAMPLE_INTERVAL)/* to avoid missed
      receptions due to not yet started-up nodes */, new
      App_ControlMessage("Application self message (request sample)",
      APP_SELF_REQUEST_SAMPLE));
      scheduleAt(simTime() + DRIFTED_TIME(GRE_DELAY), new
      App_ControlMessage("Application --> Application (self)", APP_NODE_GRE));
    }
  }
  break;
}
```

Figure IV. 3. Extrait du code de l'application: traitement du message CMD1

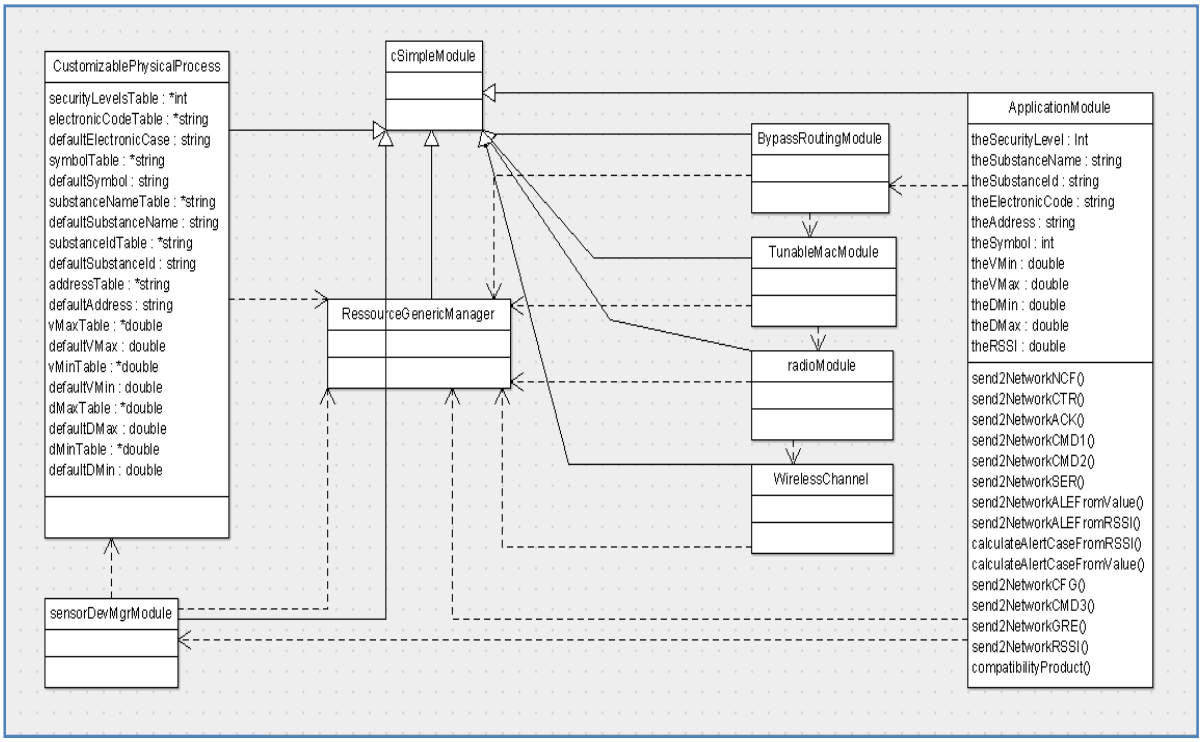

Figure IV. 4. Diagramme de classe de l'application implémenté sous Castalia

Dans la figure IV.4, les flèches hachées signifie qu'il y a des relations de dépendance (utilisation entre les deux classes liées), et les flèches en continu désignent des relations d'héritage.

A des fins d'évaluation, nous avons implémenté le modèle des produits actifs développé avec les réseaux de Petri sous le simulateur Castalia. L'implémentation du modèle Réseaux de Petri sous Castalia se fait comme suit :

La figure IV.5 décrit un exemple traduit sous Castalia du modèle RdP de la figure III.12 qui modélise le passage de l'état « attente d'acquittement » à l'état « envoie NCF0 », « envoie NCF1» ou « envoie NCF2». La transition correspond à la réception du message ACKCTR, qui se traduit sous Castalia au « case AKCTR ».

```
case ACKCTR:
{
	if (self == 0)
		printf("ERROR: ACK must be sent by the supervisor to the node\n");
	else
	{
		isAckCtr = 1;
		theSecurityLevel = 0;
		if (!isConfigured && !hasSecurityRules)
		{
			printf("node: spentEnergy before sending NCF0 %f\n",
			resMgrModule->getSpentEnergy());
			send2NetworkNCF(msgSender.c_str(),0, 1);
		}
		else
		if (isConfigured && !hasSecurityRules)
		{
			printf("node: spentEnergy before sending NCF1 %f\n",
			resMgrModule->getSpentEnergy());
			send2NetworkNCF(msgSender.c_str(),1, 1);
		}
		else
		if (!isConfigured && hasSecurityRules)
		{
			printf("node: spentEnergy before sending NCF2 %f\n",
			resMgrModule->getSpentEnergy());
			send2NetworkNCF(msgSender.c_str(),2, 1);
		}
	}
	break;
}
```

Figure IV. 5. Exemple de code de modèle implémenté sous castalia

Le traitement consiste à vérifier le type de configuration du produit actif. On a 3 possibilités de transition comme le montre le diagramme d'état/transition de la figure II.39 ou le modèle de la figure III.12, et selon le type de configuration, on passe à l'état « Envoie NCF0 »,

« envoie NCF1» ou « envoie NCF2». Sous Castalia, cela correspond à la méthode « send2NetworkNCF » qui envoie NCF0, NCF1 ou NCF2.
Ce travail d'implémentation sous Castalia a été réalisé pour tous les RdP présentés dans le chapitre précédent.

V. Les résultats de simulation

V.1. L'environnement de simulation

Les caractéristiques du PC sur lequel les simulations ont été effectuées sont :

- Processeur : Intel® Core™2 Duo CPU T5250 1,50 GHz
- RAM : 2046 MB
- OS Version : Windows XP
- plateforme de développement : Microsoft Visual C++ 2008 Express Edition
- plateforme de simulation : Cygwin

V.2. Les paramètres utilisés dans la simulation

Pour valider le modèle interne de produit actif, nous avons choisi d'appliquer ce modèle au stockage et logistique industriel de produits chimiques. La surface simulée est un entrepôt avec une surface de 25m x 25m qui contient 8 produits actifs et un gestionnaire (Chaque conteneur disposant d'un capteur). L'organisation spatiale de ces produits est donnée sur la figure IV.6.

Les produits sont organisés suivant une matrice 6 produits actifs en ligne x 6 produits actifs en colonne, et la distance qui sépare deux produits voisins sur un même axe est de 5m.

Avant de lancer la simulation, plusieurs paramètres doivent être fixés selon plusieurs facteurs comme le type de réseau de capteurs à simuler (surface, nombre de nœuds, …), le protocole simulé (nombre de paquets transmis, longueurs des paquets, …), le type de capteur choisi (TelosB CC2420 ou Mica2 CC1000).
Castalia propose une méthode simple pour que l'utilisateur puisse accéder et/ou modifier ces paramètres.

L'outil de simulation utilise un ensemble de fichiers de configuration (.ini), pour chaque type de module.

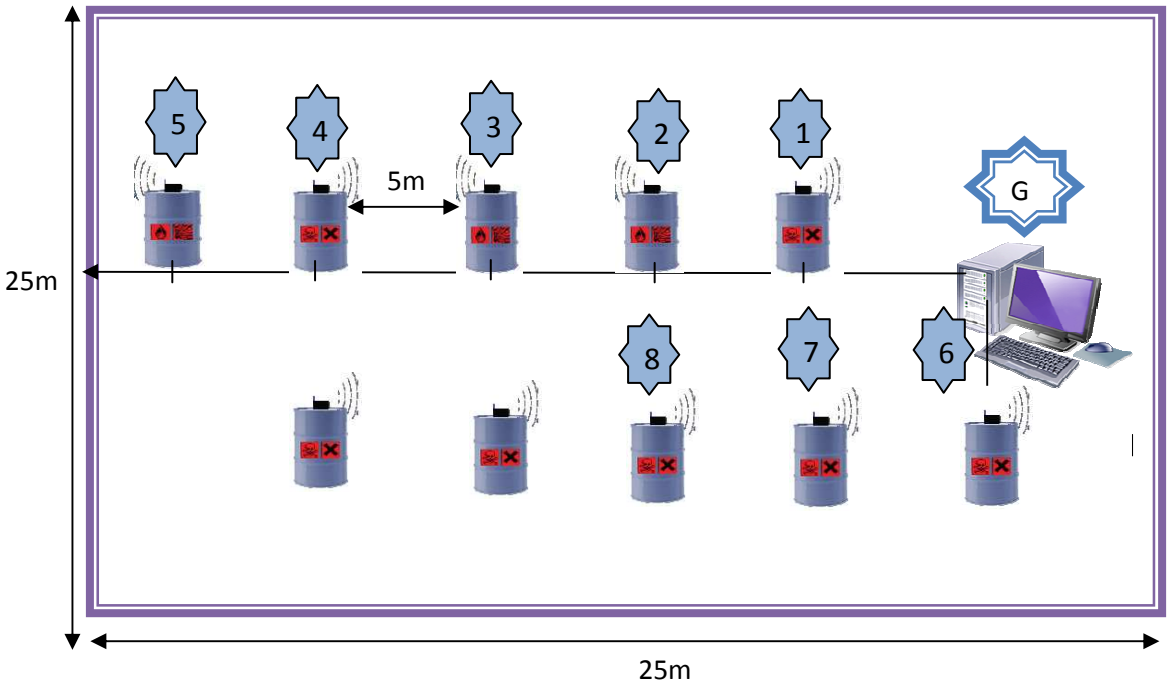

Figure IV. 6. Disposition des produits actifs dans un entrepôt 25m x 25m

Les paramètres utilisés dans les simulations sont les suivants:

Les paramètres du canal réseau sans fil :

- Sigma = 4 (la déviation standard du chemin)
- Exposant de l'affaiblissement de parcours = 2.4
- PLd_0 = 55 (dBm l'affaiblissement de parcours pour d0)
- d_0 = 1 (m)
- modèle de collision = interférence additive

Les paramètres MAC :

- cycle d'utilisation = 1 (= intervalle d'écoute / (intervalle de veille + intervalle d'écoute))
- intervalle d'écoute = 100 ms

Les paramètres du Radio :

- Conforme aux composants Telos (CC2420).

V.3. Réglage des paramètres de scrutation

Certaines applications des réseaux de capteurs sans fil et essentiellement dans le domaine de surveillance exigent que les données perçues doivent atteindre la station de base durant un

temps limite pour que la donnée soit utile et acceptable [Sohraby et al., 2007]. D'autre part un nœud de capteurs consomme, généralement, la majorité de son énergie durant la communication des données [Pottie and Kaiser, 2000]. Pour cela nous consacrons notre étude essentiellement sur la réactivité du système et la consommation d'énergie.

Afin de bien régler les valeurs de la période d'envoi des messages de salutation (T_{GRE}) et de la période de lecture des valeurs ambiantes (T_{cap}), nous avons fait une série de simulations.

Nous avons plusieurs contraintes et pour cela nous avons appliqué la méthode des plans d'expériences.

Pour mesurer la réactivité du système, nous avons crée un scénario. Ce scénario représente un PA numéro 3 de la communauté (PA 3) qui a pour valeur critique une température de 200°C. Nous avons programmé la valeur de ce paramètre de sorte qu'elle dépasse cette limite à l'instant t=200s. Par la suite nous allons mesurer l'instant de détection de danger par le gestionnaire après la transmission de l'état critique du produit 3. Ensuite nous allons mesurer l'erreur relative en % (E_r) qui est défini par :

$$E_r(\%) = \frac{t_m - t_r}{t_r} * 100 \qquad \text{(IV.3)}$$

Avec t_m l'instant de détection de l'alerte par le gestionnaire et t_r l'instant de l'occurrence de l'alerte qui est égale dans notre scénario à 200s.

Les résultats des simulations dépendent des périodes de salutation et de lecture des valeurs ambiantes. Pour cela nous avons fixé comme domaine d'étude l'intervalle [0.5s , 5s] pour les périodes T_{cap} et T_{GRE} qui sont considérés comme des valeurs extrêmes.
Le but est d'évaluer de façon combinée le rapport T_{cap}/T_{GRE} en fonction de l'erreur relative et de la consommation d'énergie. Pour limiter les essais et les temps de simulation, la section suivante propose le recours aux plans d'expériences pour résoudre ce problème.

V.4. Plans d'expériences

Les plans d'expériences sont très utilisés dans la communauté scientifique. [Hajjaji et al., 2010] ont utilisé la méthode de plan d'expériences pour décrire le comportement énergétique d'un procédé de production d'hydrogène par Reformage du Méthane à la Vapeur (RMV). Les résultats de la simulation, par le logiciel Aspen Plus™, ont été exploités pour calculer les

performances énergétiques d'une unité de RMV. Ainsi un modèle mathématique a été établi, à partir de la méthode par plans d'expériences, corrélant le rendement énergétique avec les paramètres opératoires de l'unité de RMV.

L'objectif de la méthode appliquée à la simulation est d'obtenir un maximum d'information avec un minimum de simulations. En effet, pour un problème défini, différentes solutions peuvent être testées, évaluées et comparées. Une organisation systématique des simulations accélère l'obtention des résultats en diminuant en même temps le risque d'erreurs [Sila, 2006].

V.4.1. Introduction

L'origine des plans d'expériences remonte au début du siècle dernier pour des recherches agronomiques. Ils reposent essentiellement sur des expérimentations multi-facteurs et sur un traitement des résultats à l'aide de régressions multiples et d'analyse de la variance. Ils sont restés du domaine de quelques spécialistes et leurs applications industrielles ont été réduites du fait de la complexité des calculs qu'ils nécessitent, jusqu'à l'arrivée des moyens informatiques puissants. [Souvay, 1995]

V.4.2. Contexte d'utilisation

Le contexte d'utilisation des plans d'expérience recouvre les phénomènes de type boite noire, que l'on cherche à éclaircir pour mieux en comprendre le fonctionnement et en optimiser les performances. Cela impose une connaissance minimale sur le phénomène étudié avant d'entreprendre les essais. On doit être en mesure de lister les paramètres susceptibles d'agir sur le fonctionnement de la boite noire. Ces paramètres sont les entrées appelées par la suite facteurs ou variables. Les performances obtenues sont les sorties appelées par la suite réponses. [Benoist et al., 1994]

V.4.3. Objectif

A partir des résultats d'essais et/ou de simulation réalisés, on cherche à expliquer le fonctionnement de la boite noire en estimant les réponses pour des combinaisons non réalisées des paramètres d'entrées. Le but est de trouver une configuration des paramètres d'entrée qui résolve le problème.

Nous avons appliqué la méthode du plan d'expérience sur la réactivité et l'autonomie du système. On va prendre comme réponses du système l'erreur relative en % et l'énergie consommée en Joules.

V.4.4. Préparation du plan d'expériences

Cette étape se décompose en plusieurs parties dont les principales sont décrites dans les paragraphes suivants.

V.4.5. Définition de l'objectif de l'étude

Ici le but est de montrer une méthodologie qui permet de régler au mieux les paramètres de notre protocole de communication en fonction d'un cahier des charges données. Ce cahier des charges est le suivant :

- Le système doit avoir un temps de détection de la faute inférieure à 4 s. Cela signifie que les valeurs des erreurs relatives inférieures à 2%.
- Le système doit avoir une autonomie de 9.3 jours signifie que l'énergie consommée doit être inférieure à 61,827957 Joules.

L'étude sera réussie si on peut donner des consignes de réglages des facteurs pour atteindre les deux objectifs.

V.4.6. Description des éléments sur lesquels va porter l'expérimentation

Nous décrivons maintenant les simulations réalisées.

- **Choix des réponses permettant d'atteindre l'objectif**

La réponse est la réactivité (Erreur relative en %) du système et la consommation énergétique

- **Recherche des facteurs qui pourraient être influents sur les réponses**

Il y a bien sûr les deux facteurs que nous allons étudier, la période de lecture capteurs (T_{cap}) et la période d'envoi des messages de salutation (T_{GRE}). Mais il y en a d'autres.

Les facteurs qui ne seront pas étudiés dans le plan d'expériences seront, fixés à un niveau constant pendant toutes les simulations

- **Définition des niveaux des facteurs**

Il s'agit de choisir les niveaux haut et bas de chaque facteur comme l'indique le tableau 5.

Facteur1 : La période de lecture capteurs ne devra pas être trop faible et elle ne devra pas dépasser les limitations. Dans notre cas, le niveau bas sera de 0.5s et le niveau haut de 5s.

Facteur 2 : Le niveau bas du facteur période d'envoi des messages de salutation sera 0.5s et le niveau haut de 5s.

Facteur	Niveau bas (-)	Niveau haut (+)
T_{cap} (s)	0.5	5
T_{GRE} (s)	0.5	5

Tableau 5. Facteurs et domaine d'étude

Les conclusions de l'expérimentation ne seront valables qu'à l'intérieur de ce domaine d'étude.

- **Choix du plan d'expériences**

Nous savons qu'il y a deux facteurs à étudier. Les niveaux bas et haut de chaque facteur ont été définis. Les autres facteurs sont conservés constants pendant la simulation.

Ayant deux facteurs prenant chacun deux niveaux, le plus simple est de choisir un plan d'expériences factoriel complet 22. La dénomination 22 a la signification suivante : le 2 en exposant indique le nombre de facteurs, l'autre 2 indique le nombre de niveaux des facteurs. Ce plan est bien adapté à notre problème puisqu'il correspond exactement à deux facteurs prenant chacun deux niveaux. Les points d'expériences ont pour coordonnées les niveaux bas et les niveaux hauts des facteurs.

Nous avons dessiné le domaine d'étude dans l'espace expérimental, puis nous avons ajouté les points d'expériences en tenant compte de leurs coordonnées (Tableau 5).

Nous pouvons aussi représenter les expériences à faire sous forme de tableaux, en utilisant soit la grandeur habituelle ou légale (s), soit les grandeurs codées. Avec les grandeurs légales, le tableau prend le nom de tableau d'expérimentation ou de matrice d'expérimentation. Avec les grandeurs codées, le tableau prend le nom de plan d'expériences ou de matrice d'expériences.

La première colonne de la matrice d'expérimentation est utilisée pour indiquer les noms des essais (Tableau 6). Nous pouvons soit les numéroter, soit leur donner un nom. La deuxième colonne est celle du premier facteur, qui indique successivement les niveaux qu'il faut donner à ce facteur. La troisième colonne est celle du deuxième facteur et indique également les niveaux de ce facteur. Le premier essai, essai n° 1 ou essai A, sera exécuté avec une période T_{cap} =0.5s et T_{GRE}=0.5s. Le deuxième essai, essai n° 2 ou essai B, sera exécuté avec une période T_{cap}=5s et T_{GRE}=0.5s. Le troisième essai, essai n° 3 ou essai C, sera exécuté avec une période T_{cap}=0.5s et T_{GRE}=5s. Enfin le quatrième essai, essai n° 4 ou essai D, sera exécuté avec une période T_{cap}=5s et T_{GRE}=5s. Ce tableau est très utile pour l'exécution des essais.

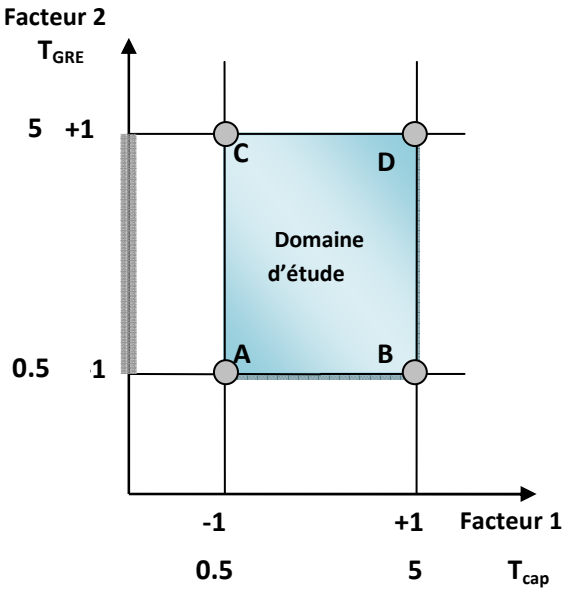

Figure IV. 7. Représentation du plan d'expérimentation

Essai N°	T_{cap} (Facteur 1)	T_{GRE} (Facteur2)
1 (A)	0.5 s	0.5 s
2 (B)	5 s	0.5 s
3 (C)	0.5 s	5 s
4 (D)	5 s	5 s

Tableau 6. Matrice d'expérimentation.

Essai N°	T_{cap} (Facteur 1)	T_{GRE} (Facteur2)
1 (A)	-1	-1
2 (B)	+1	-1
3 (C)	-1	+1
4 (D)	+1	+1
Niveau -1	0.5 s	0.5 s
Niveau +1	5 s	5 s

Tableau 7. Matrice d'expériences.

La première colonne de la matrice d'expériences (Tableau 7) est utilisée de la même manière pour indiquer les noms des essais. La deuxième colonne est celle du premier facteur. Elle indique successivement les niveaux qu'il faut donner à ce facteur mais cette fois sous forme codée c'est-à-dire avec des -1 et des +1. La troisième colonne est celle des niveaux codés du deuxième facteur. Les deux lignes en bas du tableau indiquent la signification des niveaux -1 et +1 de chaque facteur. Ce tableau est utilisé lors de la construction du plan.

Pour l'interprétation des résultats d'essais, nous utilisons soit la matrice d'expérimentation, soit la matrice d'expériences selon les faits que l'on veut mettre en évidence.

V.4.7. Expérimentation

C'est la partie technique de l'étude. Nous exécutons la simulation pour l'essai n° 1 et nous notons la valeur de l'erreur relative du système et la valeur de la consommation d'énergie. Nous répétons de même pour les autres essais. Les résultats sont consignés dans une quatrième et cinquième colonnes de la matrice d'expériences (Tableau 8).

Essai N°	T_{cap} (Facteur 1)	T_{GRE} (Facteur2)	Réactivité: erreur relative (%)	Energie consommée (Joule)
1 (A)	-1	-1	0.1529345	61.98652644
2 (B)	+1	-1	2.403695	61.97554556
3 (C)	-1	+1	0.155441	61.74362867
4 (D)	+1	+1	2.4042855	61.73451489
Niveau -1	0.5 s	0.5 s		
Niveau +1	5 s	5 s		

Tableau 8. Matrice d'expériences et résultats.

Le modèle des plans factoriels complets 22 est :

$$y = a_0 + a_1x_1 + a_2x_2 + a_{12}x_1x_2 \qquad \text{(IV.4)}$$

Où :

– y est la réponse, dans notre cas, la réactivité (Erreur relative) du système ;

– x_1 représente le niveau du facteur 1 (la période de T_{cap}), dans notre cas 0.5s (ou –1) et 5s (ou +1) selon les essais ;

– x_2 représente le niveau du facteur 2 (la période d'envoi des messages de salutation), dans cet exemple 0.5s (ou –1) et 5s (ou +1) selon les essais ;

– x_1x_2 est le produit des niveaux des facteurs 1 et 2 ; dans notre cas, en unités codées, ce produit est égal à – 1 (x_1x_2 = –1*+1 = +1*–1 = – 1) ou à +1 (x_1x_2 = –1*–1 = +1*+1 = +1) ;

- a_0 est le coefficient constant du modèle ;

– a_1 est le coefficient du facteur 1 ;

– a_2 est le coefficient du facteur 2 ;

– a_{12} est le coefficient du terme x_1x_2.

Ce modèle est appelé modèle polynomial du premier degré avec interactions ou modèle PDAI et nous allons examiner la signification de ses coefficients.

Les réponses sont étudiées l'une après l'autre.

V.4.7.1. Réactivité :

Nous avons ensuite reporté les résultats de simulations sur le domaine d'étude (Figure IV.7).

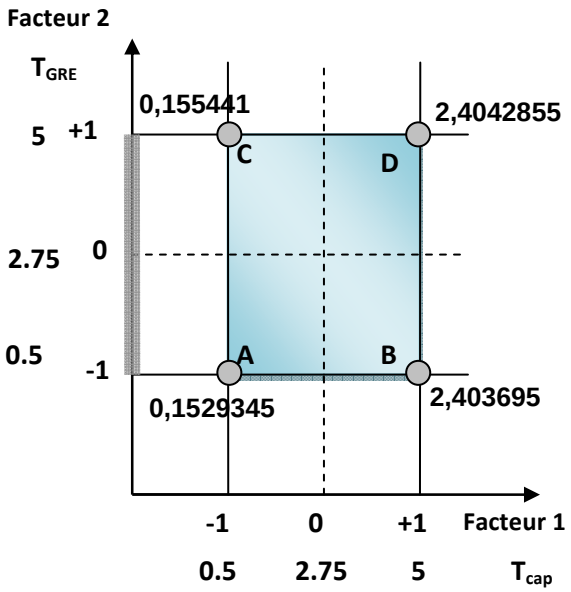

Figure IV. 8. Valeur de la réponse aux points du domaine d'étude.

- **Signification du coefficient constant**

Pour trouver la signification du coefficient constant a_0, il suffit de donner la valeur 0 (unités codées) aux niveaux des deux facteurs. Le point représentatif de l'expérience correspondante est alors au centre du domaine d'étude (Figure IV.9) et la réponse en ce point a pour valeur y_0.

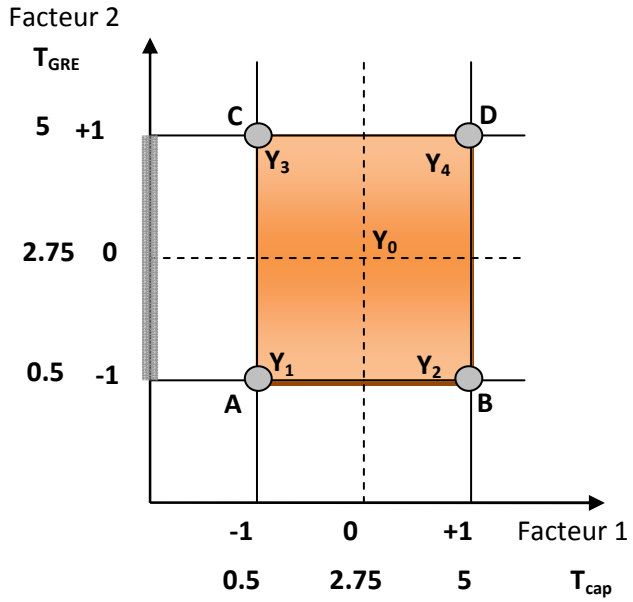

Figure IV. 9. Le coefficient pour la réponse au centre du domaine d'étude.

La relation (IV.4) devient :

$y = a_0 + a_1 * 0 + a_2 * 0 + a_{12} * 0 * 0$ par suite $y_0 = a_0$

La valeur du coefficient constant a_0 est égale à la réponse au centre du domaine d'étude.

- **Signification du coefficient du facteur 1**

Considérons les deux points B et D qui se trouvent au niveau haut du facteur 1.

Les coordonnées de ces points sont, en unités codées :

B : $x_1=+1$ et $x_2= -1$ et **D** : $x_1=+1$ et $x_2=+1$

La réponse au point B est y_2. En remplaçant les niveaux par leurs valeurs en unités codées cette réponse peut s'écrire:

$y_2 = a_0 + a_1*(+1) + a_2*(-1) + a_{12}*(+1)*(-1) = a_0 + a_1 - a_2 - a_{12}$

La réponse au point D est y_4, que nous pouvons écrire en remplaçant les niveaux par leurs valeurs en unités codées :

$y_4 = a_0 + a_1*(+1) + a_2*(+1) + a_{12}*(+1)*(+1) = a_0 + a_1 + a_2 + a_{12}$

Additionnons les deux réponses y_2 et y_4 : $y_2 + y_4 = 2\ (a_0 + a_1)$

Faisons le même calcul pour les points A et C qui se trouvent au niveau bas du facteur 1 et où les réponses sont respectivement y_1 et y_3.

Nous obtenons : $y_1 + y_3 = 2(a_0 - a_1)$

Si nous faisons la soustraction de ces deux dernières relations, nous avons :

$4a_1 = -y1 + y2 - y3 + y4$

Nous obtenons : $a_1 = \frac{1}{2}[\frac{y_2 + y_4}{2} - \frac{y_1 + y_3}{2}]$

Or $\frac{y_2 + y_4}{2}$ est la moyenne des réponses au niveau haut du facteur 1. Nous nommons cette moyenne $\overline{y_+}$. Quant à l'expression $\frac{y_1 + y_3}{2}$, c'est la moyenne des réponses au niveau bas du facteur 1, soit $\overline{y_-}$. Nous pouvons écrire : $a_1 = \frac{1}{2}[\overline{y_+} - \overline{y_-}]$

Nous avons les quatre réponses, Nous pouvons donc calculer facilement le coefficient :

a_1=1,12490125.

Le coefficient a_1 est donc la demi-différence entre la moyenne des réponses au niveau haut du facteur 1 et la moyenne des réponses au niveau bas du même facteur 1.

Quand nous passons du niveau bas au niveau haut, la réponse varie, en moyenne, comme la différence $[\overline{y_+} + \overline{y_-}]$. Si cette différence est grande, la réponse varie beaucoup, si cette différence est faible, la réponse varie peu. Nous avons donc là un moyen de savoir comment la réponse varie en fonction du facteur 1. C'est la raison pour laquelle nous appelons le coefficient a_1 l'effet du facteur 1.

- **Représentation du coefficient du facteur 1**

La moyenne des réponses au niveau haut du facteur 1, $\overline{y_+}$, est située sur la surface de réponse et se trouve à l'aplomb du point M+, milieu du segment BD (Figure IV.10).

Il a donc pour coordonnées : **M+** : x_1=+1 et x_2=0

La moyenne des réponses au niveau bas du facteur 1 est située sur la surface de réponse et se trouve à l'aplomb du point M-, milieu du segment AC. Il a donc pour coordonnées : **M-** : x_1= -1 et x_2=0.

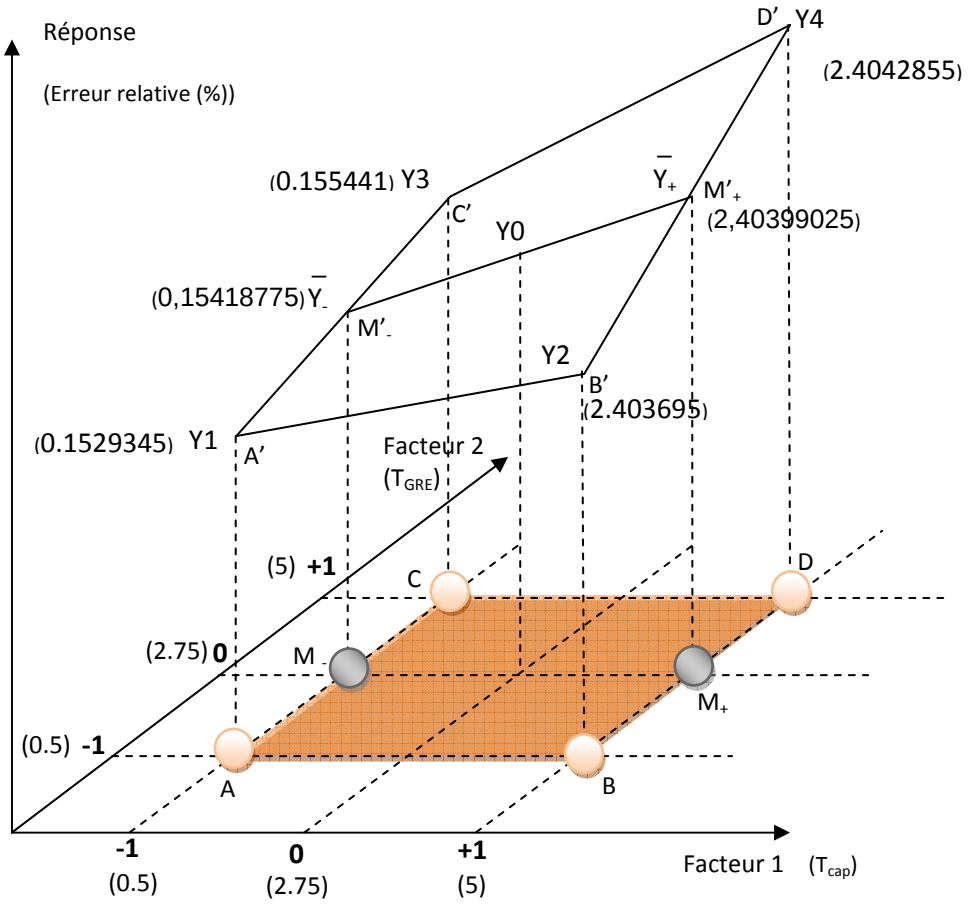

Figure IV. 10. Le coefficient du facteur 1 est la pente de la droite qui joint les deux réponses moyennes $\bar{y}_-$ et $\bar{y}_+$.

La variation de la réponse entre la moyenne des réponses au niveau haut du facteur 1, $\overline{y_+}$, et la moyenne des réponses au niveau bas de ce même facteur $\overline{y_-}$ est $\overline{y_+}$ - $\overline{y_-}$, c'est-à-dire deux fois le coefficient a_1.

Le coefficient a_1 est donc égal à la variation de la réponse entre y_0, réponse au centre du domaine d'étude, et $\overline{y_+}$, moyenne des réponses au niveau haut du facteur 1. Nous pouvons également regarder le coefficient a1 comme la pente de la droite M'-M'+.

Nous pouvons dire aussi que le coefficient a_1 est égal à la variation moyenne de la réponse quand le facteur 1 passe du niveau zéro au niveau haut. Il représente donc l'influence du facteur 1 dans le domaine d'étude.

- **Illustration de l'effet du facteur 1**

Pour faire apparaître clairement la droite et illustrer l'effet du facteur 1, nous avons extrait le plan de la figure IV.10. Nous obtenons la figure IV.11 qui est beaucoup plus facile à lire.

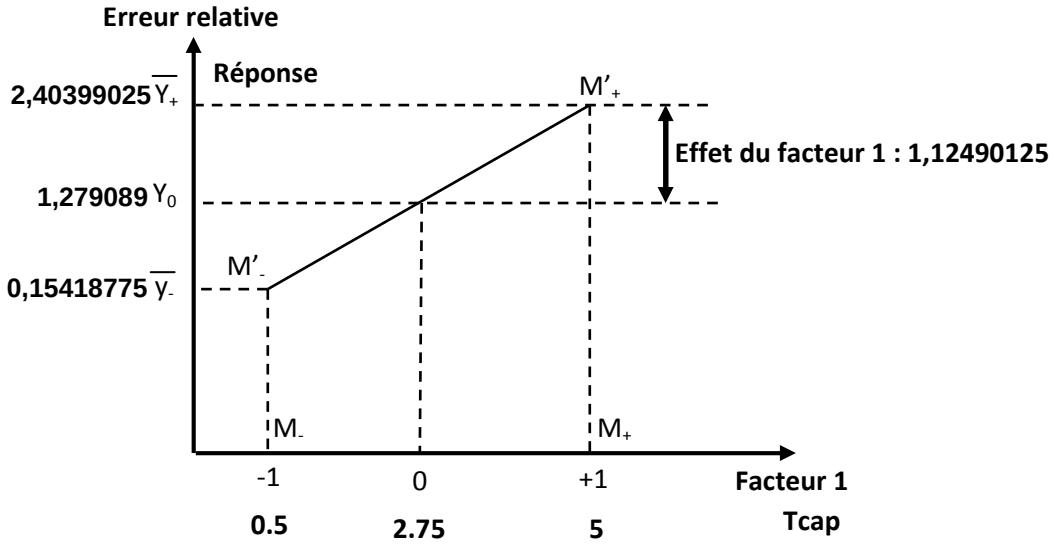

Figure IV. 11. Illustration de l'effet du facteur 1.

L'erreur relative passe, en moyenne, de 0,15418775% à 2,40399025% quand la période de lecture capteurs T_{cap} passe de 0.5 à 5s. L'erreur relative au centre est la moitié des deux moyennes, soit Y_0=1,279089s.

L'erreur relative passe, en moyenne, de 1,279089% à 2,40399025% quand la période de lecture capteurs T_{cap} passe de 2.75s à 5s. Cette augmentation de 1,12490125 représente l'effet du facteur T_{cap} (Figure IV.11).

- **Signification du coefficient du facteur 2**

De la même manière, on montre que le coefficient a_2 est égal à la variation moyenne de la réponse quand le facteur 2 passe du niveau zéro au niveau haut. Il représente l'influence du facteur 2 dans le domaine d'étude. On l'appelle « effet du facteur 2 ».

D'une manière générale, quand le modèle choisi est un polynôme, les coefficients des termes du premier degré sont les effets des facteurs.[Demonsant, 1996]

Nous avons les quatre réponses, nous pouvons donc calculer facilement le coefficient a2 :

$a_1 = \frac{1}{4}[-y_1 - y_2 + y_3 + y_4]$

a_2 = 0,00077425

L'effet de la période d'envoi de messages de salutation (facteur 2) est de 0,00077425. Cela signifie que, si la période d'envoi de messages de salutation passe de 0.5s à 2.75s, l'erreur relative augmente en moyenne de 0,00077425. Si la période d'envoi de messages de salutation passe de 0.5s à 5s l'erreur relative augmente de 0,0015485.

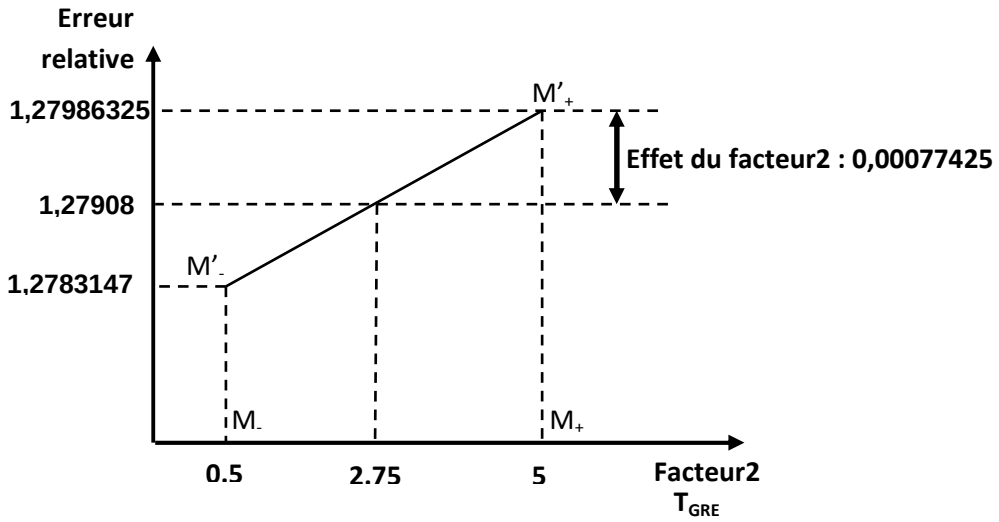

Figure IV. 12. Illustration de l'effet de T_{GRE}.

La moyenne des erreurs relatives au niveau haut de la période d'envoi de messages de salutations entre PA (facteur 2) est :

$$\overline{y_+} = \frac{y_3 + y_4}{2} = \frac{0.155441 + 2.4042855}{2} = 1{,}27986325\ \%$$

La moyenne des consommations au niveau bas de la période d'envoi de messages de salutations entre PA (facteur 2) est :

$$\overline{y_-} = \frac{y_1 + y_2}{2} = \frac{0.1529345 + 2.403695}{2} = 1{,}27831475\ \%$$

L'erreur relative passe, en moyenne, de 1,27831475 à 1,27986325 quand la période d'envoi de messages de salutations entre PA (facteur 2) passe de 0.5s à 5s. L'erreur relative au centre est la moitié des deux moyennes, soit 1,279089%.

L'erreur relative passe, en moyenne, de 1,279089 à 1,27986325 quand la période d'envoi de messages de salutations entre PA passe de 2.75 à 5s. Cette augmentation de 0,00077425 est l'effet du facteur période d'envoi de messages de salutations entre PA (T_{GRE}) (Figure IV.12).

Signification du coefficient a_{12}

Nous avons calculé le coefficient a_{12} par une méthode analogue à celle qui a été utilisée pour les coefficients a_1 et a_2. Nous trouvons que le coefficient a_{12} est égal à :

$$a_{12} = \frac{1}{2}\left[\frac{y_4 - y_3}{2} - \frac{y_2 - y_1}{2}\right]$$

$$a_{12} = \frac{1}{4}\left[+ y_1 - y_2 - y_3 + y_4\right]$$

a_{12}= - 0,001916

Or a_{12} est l'effet du facteur 1 lorsque le facteur 2 est au niveau haut. C'est la moitié de la variation de la réponse entre y_4 et y_3. Cet effet est illustré par la pente de la droite C'D' (Figure IV.10 et Figure IV.13).

L'interaction entre les facteurs 1 et 2 est de - 0,001916%. Cela signifie que l'effet de la période de lecture de capteurs (T_{cap}) est un peu plus petit quand on se trouve en période d'envoi de message (T_{GRE}) plus grand. Quand la T_{cap} est de 0.5s, l'effet de T_{GRE} est de 0,00269025 %. Quand T_{cap} est de 5s, l'effet T_{GRE} est de -0,00114175 %.

Cela signifie aussi que l'effet de T_{GRE} est plus petit quand T_{cap} est grand. Quand T_{GRE} est de 0.5s, l'effet de T_{cap} est de 1,12681725 %. Quand T_{GRE} est de 5s, l'effet de T_{cap} est de 1,12298525 % (Figure IV.14).

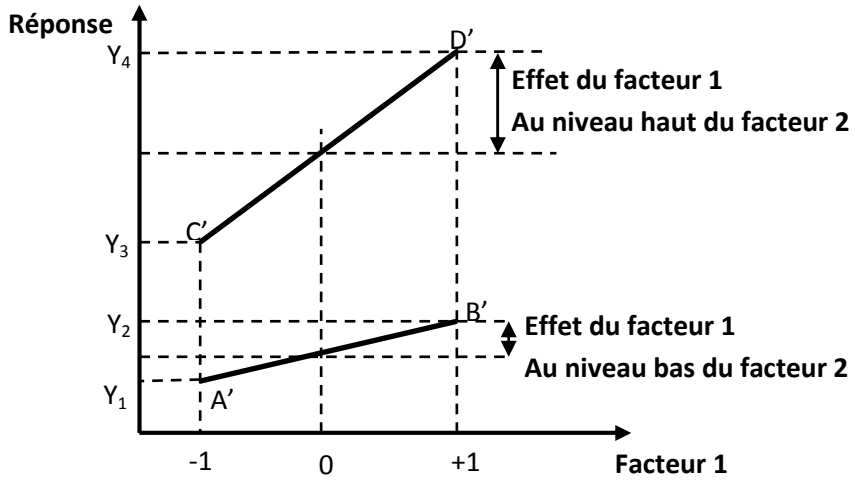

Figure IV. 13. Illustration d'une interaction entre deux facteurs.

L'expression est l'effet du facteur 1 lorsque le facteur 2 est au niveau bas.

C'est la moitié de la variation de la réponse entre y_2 et y_1. Cet effet est illustré par la pente de la droite A′B′ (Figure IV.13).

Le coefficient a_{12} est la moitié de la différence entre ces deux effets.

Le coefficient a_{12} mesure donc la variation de l'effet du facteur 1 quand le niveau du facteur 2 est modifié. Nous pouvons aussi montrer que le même coefficient a_{12} mesure également la variation de l'effet du facteur 2 quand le niveau du facteur 1 est, lui aussi, modifié.

Le coefficient a_{12} est appelé l'interaction entre les facteurs 1 et 2.

Nous pouvons illustrer une interaction entre deux facteurs en extrayant de la figure IV.10 les plans ABA′B′ (niveau bas du facteur 2), et CDC′D′ (niveau haut du facteur 2) et en projetant ces plans sur un même plan (Figure IV.13).

S'il n'y a pas d'interaction entre deux facteurs, les pentes des droites A′B′ et C′D′ sont les mêmes.

S'il y a interaction entre deux facteurs, les pentes des deux droites précédentes ne sont pas les mêmes. L'interaction est d'autant plus forte que les pentes sont différentes.

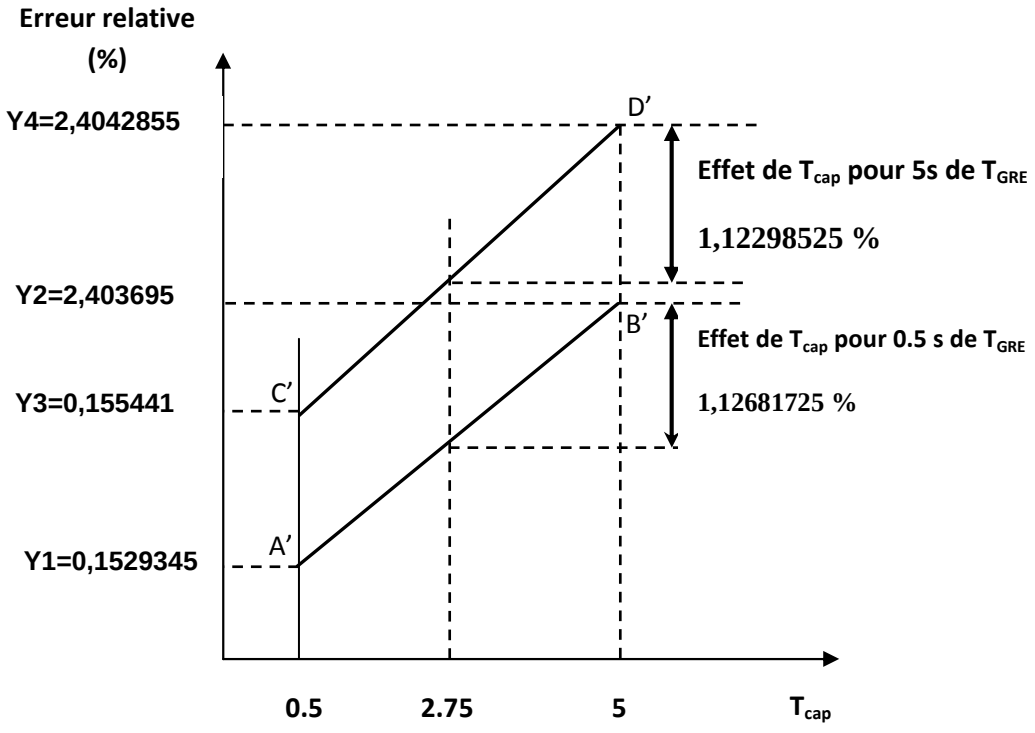

Figure IV. 14. Illustration de l'interaction entre T_{cap} et T_{GRE}.

Interprétation des résultats des calculs

Nous avons maintenant, les valeurs :

– du coefficient constant : a_0 =1,279089;

– du coefficient du facteur 1 (T_{cap}) : a_1=1,12490125;

– du coefficient du facteur 2 (T_{GRE}) : a_2=0,00077425;

– de l'interaction a_{12} entre T_{cap} et T_{GRE} : a_{12} = - 0,001916.

Nous pouvons reporter ces valeurs dans la relation (IV.4) du modèle postulé :

$$y = 1{,}279089 + 1{,}12490125\, x_1 + 0{,}00077425\, x_2 + (-\,0{,}001916)\, x_1 x_2 \qquad \text{(IV.5)}$$

Avec ce modèle, nous pouvons calculer toutes les réponses dans le domaine d'étude. Il suffit d'attribuer des valeurs aux niveaux x_1 et x_2 pour obtenir immédiatement l'erreur relative. Le modèle étant en unités centrées réduites, il faut faire les calculs dans ces unités et transformer ensuite les résultats obtenus en unités légales. Si nous voulons utiliser les unités légales directement, il faut transformer la relation (IV.5). Dans ce cas, il suffit d'appliquer la relation suivante [Goupy and Creighton, 2006]:

$$x = \frac{A - A_0}{Pas} \qquad \text{(IV.6)}$$

Avec A_0 est la valeur centrale en unité courante $A_0 = \frac{A_{+1} + A_{-1}}{2}$

Et $Pas = \frac{A_{+1} - A_{-1}}{2}$

Pour les deux facteurs :

$Pas = \frac{A_{+1} - A_{-1}}{2} = \frac{5 - 0.5}{2} = 2.25$ et

$A_0 = \frac{A_{+1} + A_{-1}}{2} = \frac{5 + 0.5}{2} = 2.75$

Facteur 1 (T_{cap}) : $x_1 = \frac{A_1 - A_0}{Pas} = \frac{A_1 - 2.75}{2.25}$

Facteur 2 (T_{GRE}) : $x_2 = \frac{A_2 - A_0}{Pas} = \frac{A_2 - 2.75}{2.25}$

$$y = 1{,}279089 + 1{,}12490125(\frac{A_1 - 2.75}{2.25}) + 0{,}00077425(\frac{A_2 - 2.75}{2.25}) + (-0{,}001916)(\frac{A_1 - 2.75}{2.25})(\frac{A_2 - 2.75}{2.25})$$

$$y = -\,0{,}09959878 + 0{,}5009969\, A_1 + 0{,}0013849\, A_2 - 0{,}00037847\, A_1 A_2 \qquad \text{(IV.7)}$$

La figure IV.15 présente la réponse du modèle dans l'espace

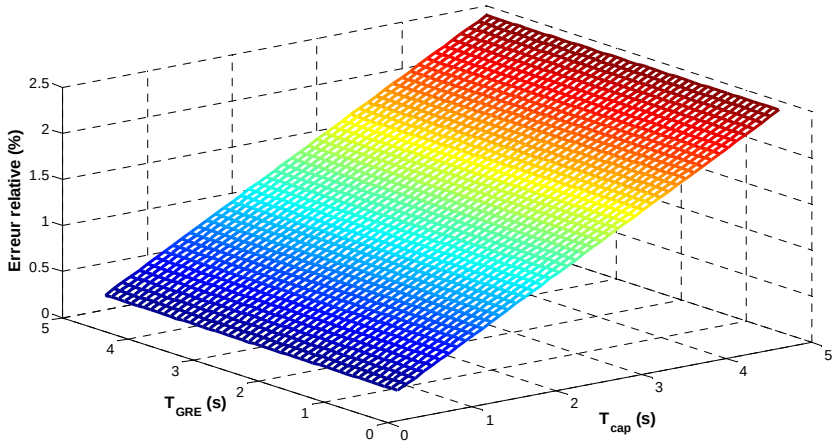

Figure IV. 15. Illustration de l'erreur relative en fonction de T_{cap} et T_{GRE}.

- **Recherche de conditions opératoires respectant les objectifs**

Le premier objectif est que la réactivité ne dépasse pas 4s donc on interdit les réactivités supérieures à 4s.

Les valeurs de réactivité inférieures à 4s donnent les valeurs des erreurs relatives inférieures à 2%. La figure suivante illustre toutes les combinaisons appartenant aux zones foncées bleus permettent de répondre aux objectifs de la réactivité du système.

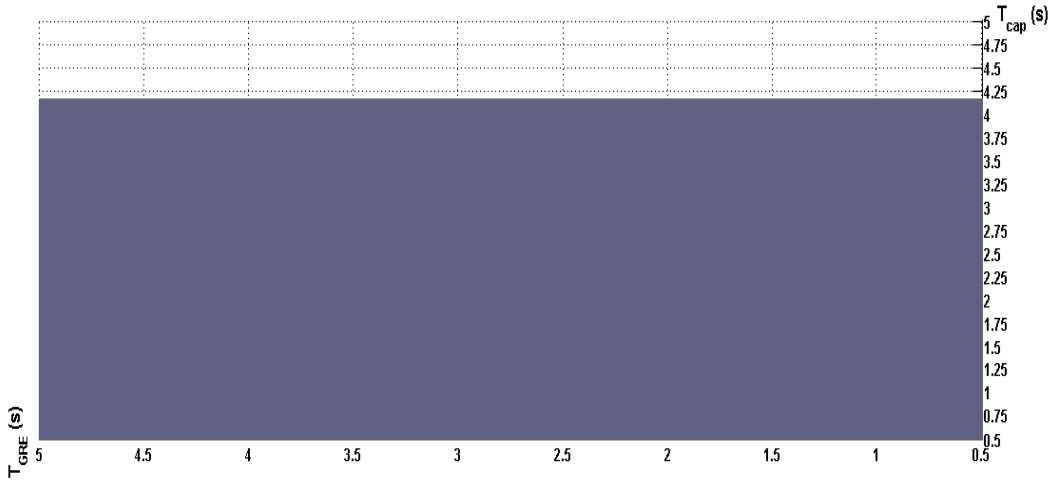

Figure IV. 16. L'objectif de la réactivité du système est atteint pour la zone grisée.

D'après la figure IV.16, l'objectif de la réactivité du système est atteint pour T_{cap} appartient à l'intervalle [0.5s, 4.1s] et T_{GRE} appartient à l'intervalle [0.5s, 5s].

V.4.7.2. Consommation énergétique :

De la même manière nous déterminons les coefficients du modèle (IV.4)

- **Signification du coefficient constant**

Nous avons les quatre réponses, Nous pouvons donc calculer facilement le coefficient.

Nous trouvons : a_1= - 0,00502366 Joules

- . **Représentation du coefficient du facteur 1**

De la même manière Nous représentons[2] la figure suivante :

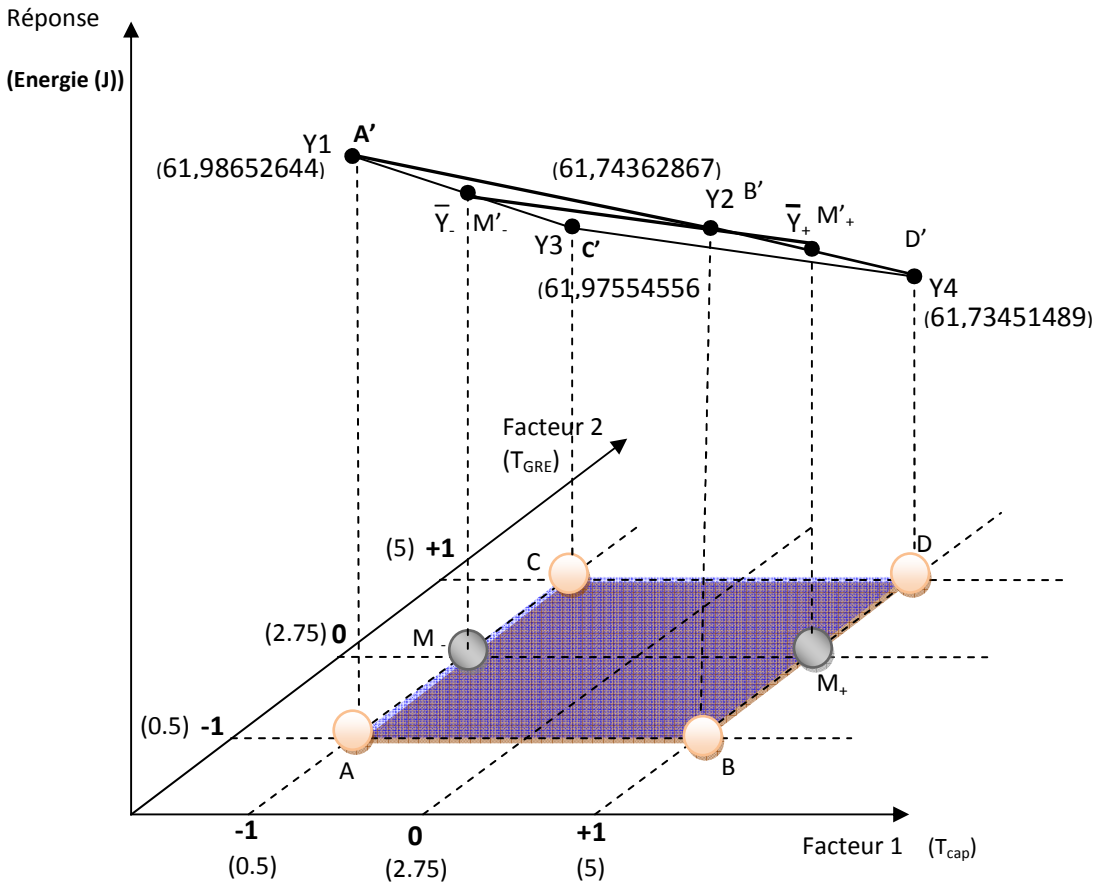

Figure IV. 17. Le coefficient du facteur 1 est la pente de la droite qui joint les deux réponses moyennes $\bar{y}_-$ et $\bar{y}_+$.

- **Illustration de l'effet du facteur 1**

La figure IV.18 illustre l'effet du facteur 1.

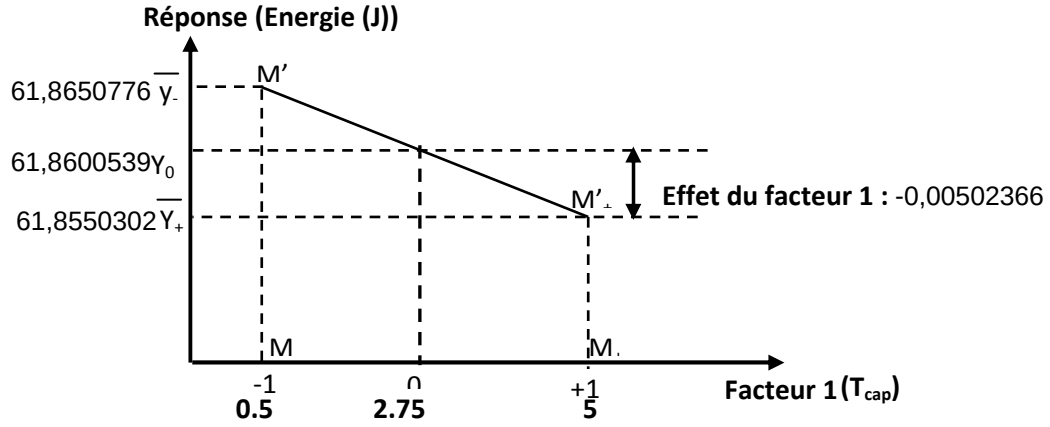

Figure IV. 18. Illustration de l'effet du facteur 1.

La consommation d'énergie passe, en moyenne, de 61,8650776 J à 61,8550302 J quand la période de lecture capteurs T_{cap} passe de 0.5 à 5s. La consommation d'énergie au centre est la moitié des deux moyennes, soit 61,8600539 J**.**

La consommation d'énergie passe, en moyenne, de 61,8600539 J à 61,8550302 J quand la période de lecture capteurs T_{cap} passe de 2.75s à 5s. Cette diminution de 0,00502366 J représente l'effet du facteur « T_{cap} » (Figure IV.10).

- **Signification du coefficient du facteur 2**

Nous avons les quatre réponses, nous pouvons donc calculer facilement le coefficient a_2 :

$a_2 = ¼[-y_1 - y_2 + y_3 + y_4]$

$a_2 = -0,12098211$

L'effet de la période d'envoi de messages de salutation (facteur 2) est de - 0,12098211 J. Cela signifie que, si la période d'envoi de messages de salutation passe de 0.5s à 2.75s, la consommation d'énergie diminue en moyenne de 0,12098211 J. Si la période d'envoi de messages de salutation passe de 0.5 s à 5 s la consommation d'énergie diminue de 0,24196422 Joules.

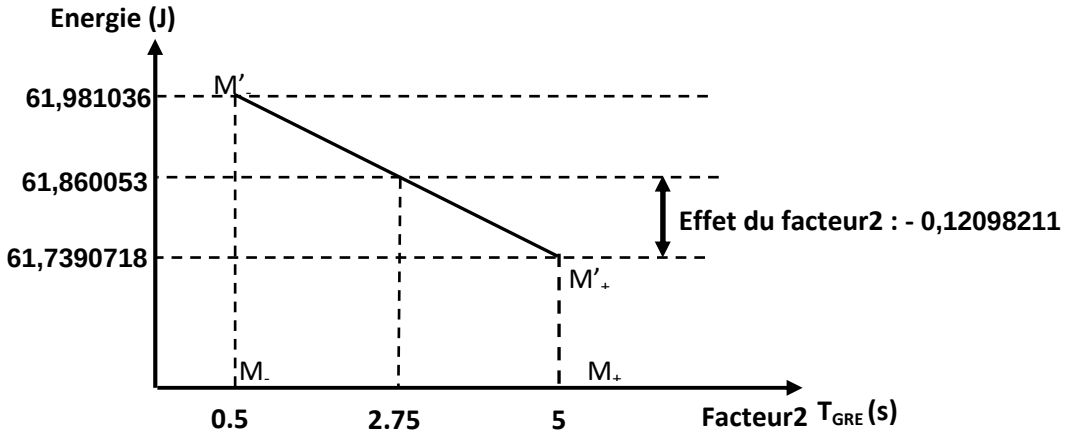

Figure IV. 19. Illustration de l'effet de T_{GRE}.

La moyenne des erreurs relatives au niveau haut de la période d'envoi de messages de salutations entre PA (facteur 2) est :

$$\overline{y_+} = \frac{y_3 + y_4}{2} = \frac{61{,}9865264 + 61{,}9755456}{2} = 61{,}7390718 \text{ Joules}$$

La moyenne des consommations au niveau bas de la période d'envoi de messages de salutations entre PA (facteur 2) est :

$$\overline{y_-} = \frac{y_1 + y_2}{2} = \frac{61{,}7436287 + 61{,}7345149}{2} = 61{,}981036 \text{ Joules}$$

La consommation d'énergie passe, en moyenne, de 61,981036 Joules à 61,7390718 Joules quand la période d'envoi de messages de salutations entre Produits Actifs (facteur 2) passe de 0.5s à 5s. La consommation d'énergie au centre est la moitié des deux moyennes, soit 61,8600539 Joules.

La consommation d'énergie passe, en moyenne, de 61,8600539 Joules à 61,7390718 Joules quand la période d'envoi de messages de salutations entre Produits Actifs passe de 2.75 à 5s. Cette diminution de 0,12098211 Joules est l'effet du facteur période d'envoi de messages de salutations entre Produits Actifs (T_{GRE}) (Figure IV.11).

- **Signification du coefficient a_{12}**

Nous avons calculé le coefficient a_{12} par une méthode analogue à celle qui a été utilisée pour les coefficients a_1 et a_2. Nous trouvons que le coefficient a_{12} est égal à :

$$a_{12} = \frac{1}{2}\left[\frac{y_4 - y_3}{2} - \frac{y_2 - y_1}{2}\right] = \frac{1}{4}\left[+ y_1 - y_2 - y_3 + y_4\right]$$

a_{12}= 0,0018671 Joules.

S'il n'y a pas d'interaction entre deux facteurs, les pentes des droites A′B′ et C′D′ représentées par la figure IV.20 sont les mêmes.

S'il y a interaction entre deux facteurs, les pentes des deux droites précédentes ne sont pas les mêmes. L'interaction est d'autant plus forte que les pentes sont différentes.

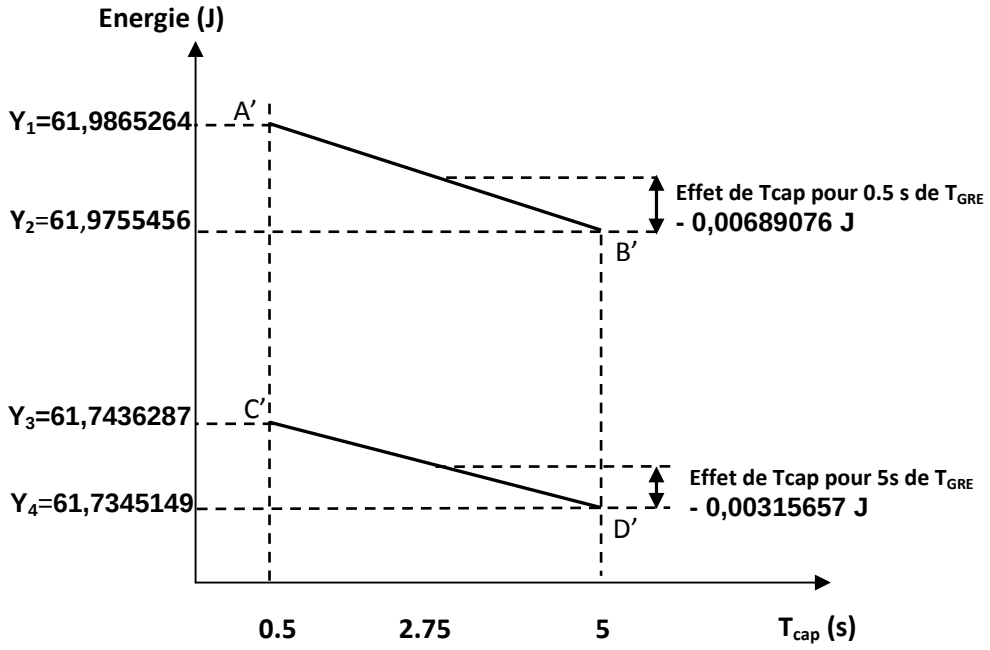

Figure IV. 20. Illustration de l'interaction entre T_{cap} et T_{GRE}.

- **Interprétation des résultats des calculs**

Nous avons les valeurs :

– du coefficient constant : a_0 = 61,8600539;

– du coefficient du facteur 1 (T_{cap}) : a_1=- 0,00502366;

– du coefficient du facteur 2 (T_{GRE}) : a_2= - 0,12098211;

– de l'interaction a_{12} entre T_{cap} et T_{GRE} : a_{12} = 0,0018671.

Nous pouvons reporter ces valeurs dans la relation (IV.4) du modèle postulé :

$$y=61,8600539 + (- 0,00502366)\, x_1 - 0,12098211\, x_2 + 0,0018671\, x_1\, x_2 \qquad \text{(IV.8)}$$

Avec ce modèle, nous pouvons calculer toutes les réponses dans le domaine d'étude. Il suffit d'attribuer des valeurs aux niveaux x_1 et x_2 pour obtenir immédiatement la consommation d'énergie. Le modèle étant en unités centrées réduites, il faut faire les calculs dans ces unités et transformer ensuite les résultats obtenus en unités légales. Si nous voulons utiliser les unités légales directement, il faut transformer la relation (IV.8). Dans ce cas, il suffit d'appliquer la même démarche que pour la réponse 1.

Nous obtenons :

$$y = 62,0112718 - 0,00121851\, A_1 - 0,0527556\, A_2 - 0,00036881\, A_1\, A_2 \qquad \text{(IV.9)}$$

La figure IV.21 présente la réponse en consommation énergétique dans l'espace.

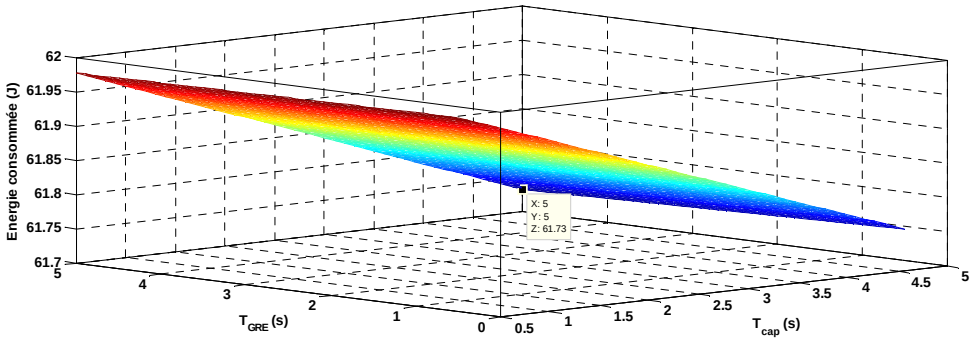

Figure IV. 21. Illustration de l'énergie consommée en fonction de T_{cap} et T_{GRE}.

- **Recherche de conditions opératoires respectant les objectifs**

Une autonomie <9.3 jours signifie que l'énergie consommée <61,827957 Joules.

Pour l'objectif de l'autonomie du système, la figure IV.22 représente toutes les combinaisons appartenant aux zones foncées noires permettent de répondre aux objectifs de l'autonomie du système.

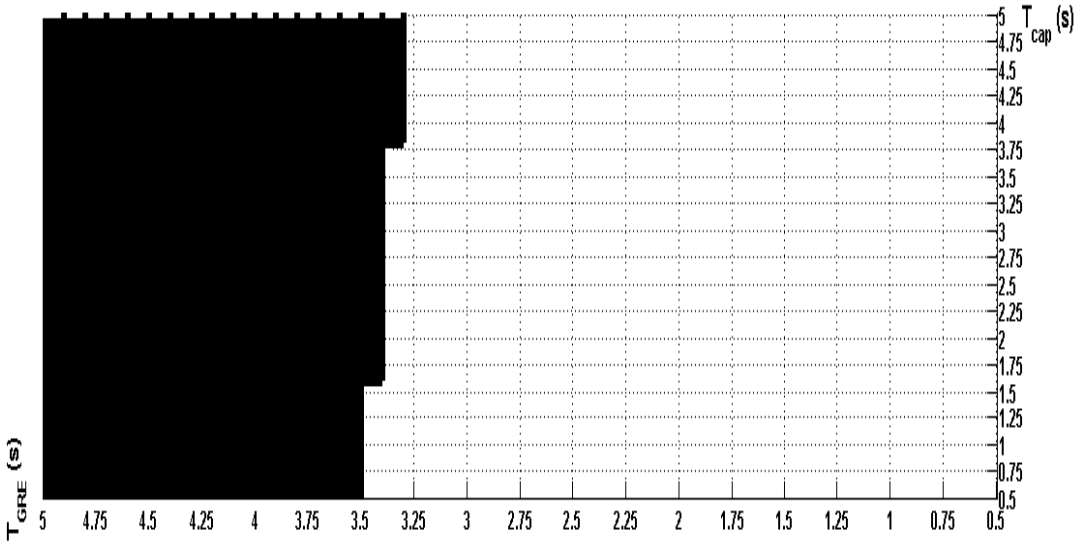

Figure IV. 22. L'objectif de l'autonomie en fonction de T_{cap} et T_{GRE}

V.4.7.3. Recherche de conditions opératoires respectant les objectifs

Comme il y a deux objectifs, une réactivité ne dépasse pas 4s et le système doit avoir une autonomie égale à 9.3 jours, nous allons superposer les résultats de la réactivité à ceux de la consommation d'énergie (Figure IV.23).

On interdit les réactivités inférieures à 4s et l'autonomie inférieure à 9.3 jours.
Toutes les combinaisons T_{cap}/T_{GRE} appartenant à la zone d'intersection grisée des deux courbes permettent de répondre aux objectifs. Donc, on a un large éventail de solutions.

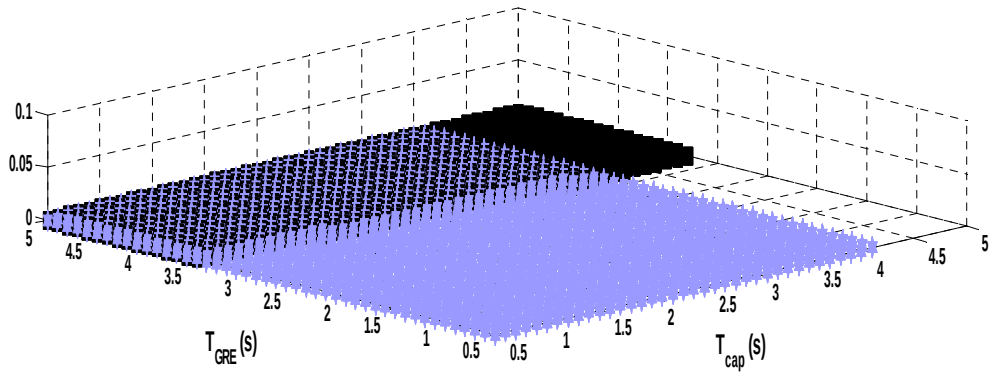

Figure IV. 23. Les objectifs en fonction de T_{cap} et T_{GRE}

Dans l'intervalle respectant les objectifs, la réactivité est minimale pour T_{cap}=0.5s et T_{GRE}=3.3s et a pour valeur 0.1548 %. La consommation d'énergie est minimale pour T_{cap}=4.1s et T_{GRE}=5s et a pour valeur 61.7349 Joules.
Nous allons prendre par la suite les valeurs suivantes T_{cap}=0.5s et T_{GRE}=5s. Pour ces valeurs les objectifs sont bien satisfaits.

V.5. Etude des cas

V.5.1. Les scénarios de configuration

Chaque produit actif doit être configuré pour entrer dans la communauté. Premièrement, pour entrer dans la communauté, le produit actif doit envoyer le message CTR au superviseur et attendre pour être acquitté. Lorsque le produit actif reçoit un acquittement (message ACK), il se met en mode configuration en envoyant les messages NCF0 ou NCF1 ou NCF2 selon l'état initial du produit actif comme cela a déjà montré dans le modèle RdP de la figure III.12. Après cette étape, le produit actif est considéré comme configuré.

Afin de voir les résultats de simulation de ces étapes, nous présentons les scénarios suivants :

Scénario1 : Le produit actif qui entre dans la communauté n'admet ni les paramètres de configuration générale ni les règles de sécurité.

```
Admin: 0 -> received APP_NODE_STARTUP at 0.000000
AP: 1 -> received APP_NODE_STARTUP at 0.050996
AP: 1 -> sent CTR to Admin at 0.050996
…
AP: 0 -> received APP_DATA_PACKET(CTR) from 1 at 0.190010
AP: 0 -> sent ACK(1028) to 1 at 0.190010
AP: 1 -> received APP_DATA_PACKET(ACKCTR) from Admin at 0.195583
AP: 1 -> sent NCF(0) to Admin at 0.195583
…
AP: 0 -> received APP_DATA_PACKET(NCF 0) from 1
AP: 0 -> sent CMD1 to 1 at 0.404516
AP: 0 -> sent CMD3 to 1 at 0.404516
AP: 1 -> received APP_DATA_PACKET(CMD1) from Admin at 0.414901
AP: 1 -> sent ACK(1037) to Admin at 0.414901
AP: 0 -> received APP_DATA_PACKET(ACKCMD1) from 1 at 0.423653
…
AP: 1 -> sent ACK(1038) to Admin at 0.826218
…
AP: 1 is now Configured and it has SecurityRules at 0.826218
```

Figure IV. 24. Scénario de configuration du PA 1 (aucune configuration installé)

Le résultat de simulation montre que le produit actif 1 envoi un message CTR vers le gestionnaire à l'instant t_1=0.050996s et il a reçu un Acquittement de la station de base (Administrateur) à t_2=0.195583s et comme le produit actif n'admet ni les paramètres de configuration générale ni les règles de sécurité, il a envoyé au gestionnaire un message NCF0 à l'instant t=0.195583s. Le Gestionnaire a répondu par le message CMD1 qui contient les paramètres de configuration et par le message CMD3 qui contient les règles de sécurité correspondent à ce produit actif. Le produit actif 1 devient configuré à l'instant t_3=0.826218s.

Ce résultat de simulation nous montre que le temps de configuration du produit actif 1 est égal à $t_3 - t_1$= 0,775222 s.

Les autres produits actifs vont être configurés de la même façon. Le tableau IV.5 comprend le temps de début et de fin de configuration de 8 produits actifs du scénario précédent.
On a défini une période égale à 0.5s (APP_NODE_STARTUP_DELAY) entre l'initialisation de deux nœuds successifs qui entre simultanément dans la communauté. Cette période permet de minimiser le nombre de messages dans l'étape de configuration en décalant la phase d'initialisation d'un produit actif numéro I+1 par 0.5 de celle du produit actif numéro I.

PA	Temps de connexion (en ms)	Temps de fin de configuration (en ms)
Gestionnaire	0.000000	--
PA 1	0.050996	0.826218
PA 2	0.100994	1.199466
PA 3	0.151011	1.634322
PA 4	0.201001	2.004465
PA 5	0.250982	2.422116
PA 6	0.301031	2.810260
PA 7	0.351023	3.226666
PA 8	0.401016	3.613599

Tableau 9. Récapitulation des temps de configuration des PAs

Les temps de configuration des Produits actifs donnés dans le tableau précédent montre un décalage de presque de 0.5s entre les différents produits actifs.

Comme nous l'avons remarqué précédemment, il y a deux autres cas possibles pour le scénario de configuration. Dans un premier cas le PA admet les paramètres de configuration générale mais n'admet pas les règles de sécurité et le résultat de simulation est donné par la figure IV.25.

```
Admin: 0 -> received APP_NODE_STARTUP at 0.000000
AP: 1 -> received APP_NODE_STARTUP at 0.050996
AP: 1 -> sent CTR to Admin at 0.050996
…
Admin: 0 -> received APP_DATA_PACKET(CTR) from 1 at 0.190010
Admin: 0 -> sent ACK(1028) to 1 at 0.190010
AP: 1 -> received APP_DATA_PACKET(ACKCTR) from Admin at 0.195583
AP: 1 -> sent NCF(1) to Admin at 0.195583
….

Admin: 0 -> received APP_DATA_PACKET(NCF 1) from 1
Admin: 0 -> sent CMD3 to 1 at 0.404516
…
AP: 1 -> received APP_DATA_PACKET(CMD3) from 0 at 0.822044
AP: 1 -> sent ACK(1038) to Admin at 0.822044
AP: 1 is now Configured and it has SecurityRules at 0.822044
```

Figure IV. 25. Scénario de configuration du PA 1 (seule la configuration des paramètres généraux est déjà installée)

Le résultat de simulation représenté par cette figure montre que le PA 1 envoie un message d'annoncement CTR vers le gestionnaire à l'instant t1=0.050996s et il a reçu un acquittement à l'instant t2=0.195583s. Ensuite vu que le PA admet les paramètres de configuration générale mais n'admet pas les règles de sécurité, il a envoyé un message de type NCF1 au

gestionnaire à l'instant t3=0.195583s. Dans ce cas le temps de configuration du produit actif est de T=t3-t1=0.195583s - 0.050996s=0,144587s et est inférieur au cas précédent car PA n'admet que les règles de sécurité.

Le scénario où le produit n'admet pas les paramètres de configuration générale mais admet les règles de sécurité est donné dans la figure IV.26.

```
Admin: 0 -> received APP_NODE_STARTUP at 0.000000
AP: 1 -> received APP_NODE_STARTUP at 0.050996
AP: 1 -> sent CTR to Admin at 0.050996
…
Admin: 0 -> received APP_DATA_PACKET(CTR) from 1 at 0.190010
Admin: 0 -> sent ACK(1028) to 1 at 0.190010
AP: 1 -> received APP_DATA_PACKET(ACKCTR) from Admin at 0.195583
…
Admin: 0 -> received APP_DATA_PACKET(NCF 2) from 1
Admin: 0 -> sent CMD1 to 1 at 0.404516
AP: 1 -> received APP_DATA_PACKET(CMD1) from Admin at 0.414901
AP: 1 -> sent ACK(1037) to Admin at 0.414901
AP: 1 is now Configured and it has SecurityRules at 0.414901
Admin: 0 -> received APP_DATA_PACKET(ACKCMD1) from 1 at 0.423653
```

Figure IV. 26. Scénario de configuration du PA 1 (seules les règles de sécurité sont déjà installées)

Dans ce cas le temps de configuration du PA est de T=0.414901s - 0.050996s=0,363905s.

Nous remarquons que dans ce cas, le temps de configuration du PA est supérieur à la durée du cas précédent. Cela est dû à la taille des informations envoyée par le PA qui est plus grande.

V.5.2. Scénario 2 :

Lorsque le produit est configuré, et qu'il possède les règles de sécurité, il commence à envoyer périodiquement le message de salutation (message GRE) pour une période égale à T_{GRE}, et le message de lecture de valeurs des capteurs pour une période égale à T_{cap}.

Ce scénario montre le comportement du réseau lorsque le produit actif est dans un cas d'alerte. Pour cette raison, on a configuré le produit actif 3 de façon à affecter la valeur 50°C au paramètre V_{Max} (la valeur maximale à respecter de la valeur de température ambiante). La valeur détectée initialement par le produit actif était de 7°C, et cette valeur va être incrémentée par 2 à chaque période de lecture capteurs T_{cap}.

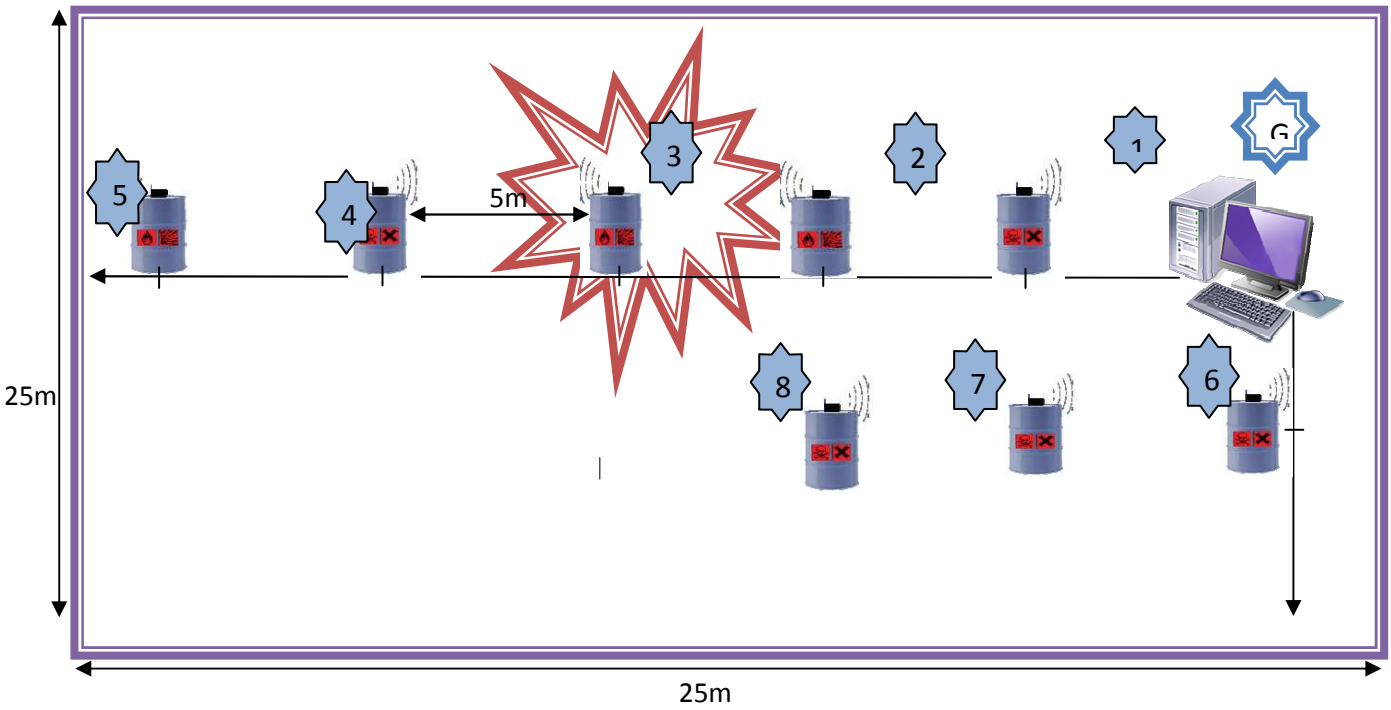

Figure IV. 27. Illustration d'alerte du PA 3

Les résultats de simulation montre que le produit actif numéro 3 de l'architecture de la figure IV.27 détecte le dépassement de la valeur ambiante et qu'il a envoyé une alerte en diffusion.

```
…
AP: self 3 calculateAlertCaseFromValue theValue 43.038390 theVMax 50.000000
…
AP: self 3 calculateAlertCaseFromValue theValue 43.840240 theVMax 50.000000
…
AP: self 3 calculateAlertCaseFromValue theValue 44.806861 theVMax 50.000000
…
AP: self 3 calculateAlertCaseFromValue theValue 46.787591 theVMax 50.000000
…
AP: self 3 calculateAlertCaseFromValue theValue 47.902659 theVMax 50.000000
…
AP: self 3 calculateAlertCaseFromValue theValue 48.927368 theVMax 50.000000
…
AP: self 3 calculateAlertCaseFromValue theValue 51.015800 theVMax 50.000000
AP: 3 -> sent ALEV from Value on BROADCAST at 41.637175
…
```

Figure IV. 28. Illustration d'un cas d'alerte du PA 3

V.5.3. Scénario 3 :

Dans ce nouveau scénario, nous avons créé une situation de danger entre les nœuds 3 et 4. Nous avons configuré ces deux produits actifs de façon à ce qu'ils soient incompatibles et par conséquent ils devront respecter une distance de proximité :

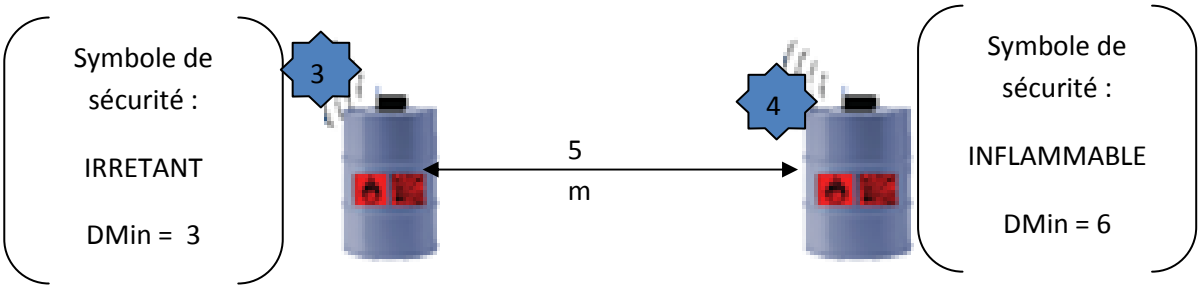

Figure IV. 29. Reconfiguration des produits 3 et 4

Les autres produits prennent le symbole de sécurité : « DANGER POUR ENVIRONNEMENT » qui est compatible avec tous les autres.

```
…
AP: 3 -> sent GRE on BROADCAST at 6.634678
AP: 1 -> received APP_DATA_PACKET(GRE)
AP: 2 -> received APP_DATA_PACKET(GRE)
AP: 4 -> received APP_DATA_PACKET(GRE)
AP: 4 -> sent ALER (from RSSI) with node 3 on BROADCAST at 6.640934
…
```

Figure IV. 30. Illustration de résultat de simulation d'envoi d'un message ALERTE dû à l'incompatibilité entre PAs

Les produits actifs 3 et 4 contiennent deux produits incompatibles selon la matrice d'incompatibilité du catalogue Merck. Lorsque le produit actif 4 reçoit le premier paquet GRE venant du PA 3, il extrait du message la valeur de la distance et il recalcule son niveau de sécurité. Il trouve alors que la distance (= 5m) avec le PA 3 est inférieure à la valeur de la distance minimale fixée à ce nœud par rapport à ses règles de sécurité. Son niveau de sécurité est alors 2 (Danger).

On remarque d'après le résultat de simulation qu'à l'instant t_1=6.634678s le produit actif 3 a envoyé un message de salutation qui contient les différents paramètres de sécurité. Dès que le produit actif 4 a reçu ce message, il calcule son niveau de sécurité qui devient égale à 2 puisqu'il est incompatible avec le produit actif 3 situé à une distance de danger. Il a donc envoyé une alerte à l'instant t_2=6.640934s. La valeur de t_2-t_1=0,006256s est une durée très faible et donc elle est acceptable pour la réactivité de notre système.

V.5.4. Scénario 4 :

Dans ce nouveau scénario, on reprend le scénario précédent mais le produit actif numéro 3 est mobile (il se déplace dans l'entrepôt). Au départ les deux produits actifs (PA3 et PA4) communiquent entre eux en utilisant le message de salutation (GRE) et à chaque réception de ce message provenant du produit actif 4, le produit actif 3 (PA3) calcule son niveau de sécurité.

La figure IV.31, illustre la variation de la distance entre les produits actifs 3 et 4, de même elle présente le niveau de sécurité de ces produits actifs en fonction de la distance.

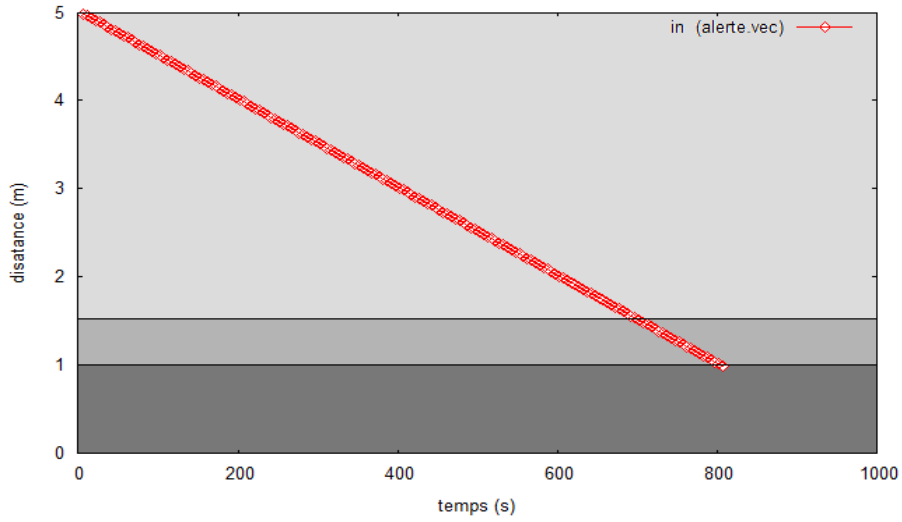

Figure IV. 31. Evolution de la distance entre les produits 3 et 4 avec le temps

Le niveau de sécurité du nœud 3 est égal à 0, son état est **Bon**

Le niveau de sécurité du nœud 3 est égal à 1, son état est **Mauvais**

Le niveau de sécurité du nœud 3 est égal à 2, il est en état de **Danger**

En utilisant la règle de communauté avec un réseau contenant un nœud mobile qui s'approche de son voisin qui présente une substance incompatible avec la sienne, on a trouvé qu'une alerte est signalée à l'instant : 807.017420s. À cet instant la distance entre les deux PAs a dépassé la distance minimale précisée dans la règle de sécurité qui est égale à 1m. Depuis cet instant, le PA 3 arrête la communication avec les autres PAs, et il continue à envoyer les messages d'alerte ALE jusqu'à la réception d'un message d'acquittement (AckALE) par la station de base (Administrateur).

Le résultat de simulation obtenue montre que le modèle spécifié fonctionne bien avec la règle de communauté.

V.6. Intervalle de confiance

En statistiques, lorsqu'on cherche à estimer la valeur d'un paramètre, on parle d'intervalle de confiance lorsque l'on donne un intervalle qui contient, avec un certain degré de confiance, la valeur à estimer. Le degré de confiance est en principe exprimé sous la forme d'une probabilité. Par exemple, un intervalle de confiance à 95% (ou au seuil de risque de 5%) a une probabilité égale à 0,95 de contenir la valeur du paramètre que l'on cherche à estimer.

Plus l'intervalle de confiance est petit, plus l'incertitude sur la valeur estimée est petite.

L'usage le plus simple des intervalles de confiance concerne les populations à distribution normale dont on cherche à estimer la moyenne $\overline{X}$. Si on connaît l'écart type σ(X) (ou si on en connaît une estimation assez fiable) de cette distribution, et si on mesure la moyenne $\bar{x}$ sur un échantillon de taille n pris au hasard, alors l'intervalle de confiance mesure le degré de précision que l'on a sur les estimations issues de l'échantillon. Il y a deux sources principales de variations sur les données qui peuvent être la cause d'un manque de précision dans l'estimation d'une grandeur.

Soit $\{E_1, E_2,\ldots, E_K\}$ l'ensemble des valeurs trouvées correspondant à K simulations. On calcule la valeur moyenne :

$$E = \frac{1}{K}\sum_{i=1}^{K} E_i \qquad \text{(IV.10)}$$

Ainsi que la variance de ces valeurs:

$$\sigma^2 = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(E_i - E)^2} \qquad \text{(IV.11)}$$

En utilisant une valeur de K supérieure à 30, l'intervalle de confiance IC [Saporta, 2006] de 95% de la valeur moyenne E sera estimé comme suit:

$$IC = E \pm 1{,}96 * \sqrt{\frac{\sigma^2}{K}} \qquad \text{(IV.12)}$$

Ce dernier représente une mesure de confiance des résultats obtenus: plus il est faible, plus les résultats sont fiables.

Les résultats présentés par la suite proviennent de simulations avec un intervalle de confiance de 95%.

Dans la suite nous allons étudier l'influence de nombre de produits actifs sur la consommation d'énergie et les pertes de paquets avec un intervalle de confiance de 95%.
Les paramètres de simulation suivants :
Surface de simulation : 25m x 25m
Temps de simulation : 1000s
Période de lecture capteurs : 0.5s
Période d'envoi de messages de salutation : 5s

V.7. Etude énergétique

Afin d'étudier la consommation d'énergie pour les différents produits actifs, on a mesuré l'énergie consommée en fonction du nombre de produits actifs dans l'entrepôt.
La courbe moyenne et l'intervalle de confiance de l'énergie consommée pour 50 simulations sont tracés sur la figure IV.32. La concentration des 50 courbes est liée à la largeur de l'intervalle de confiance.

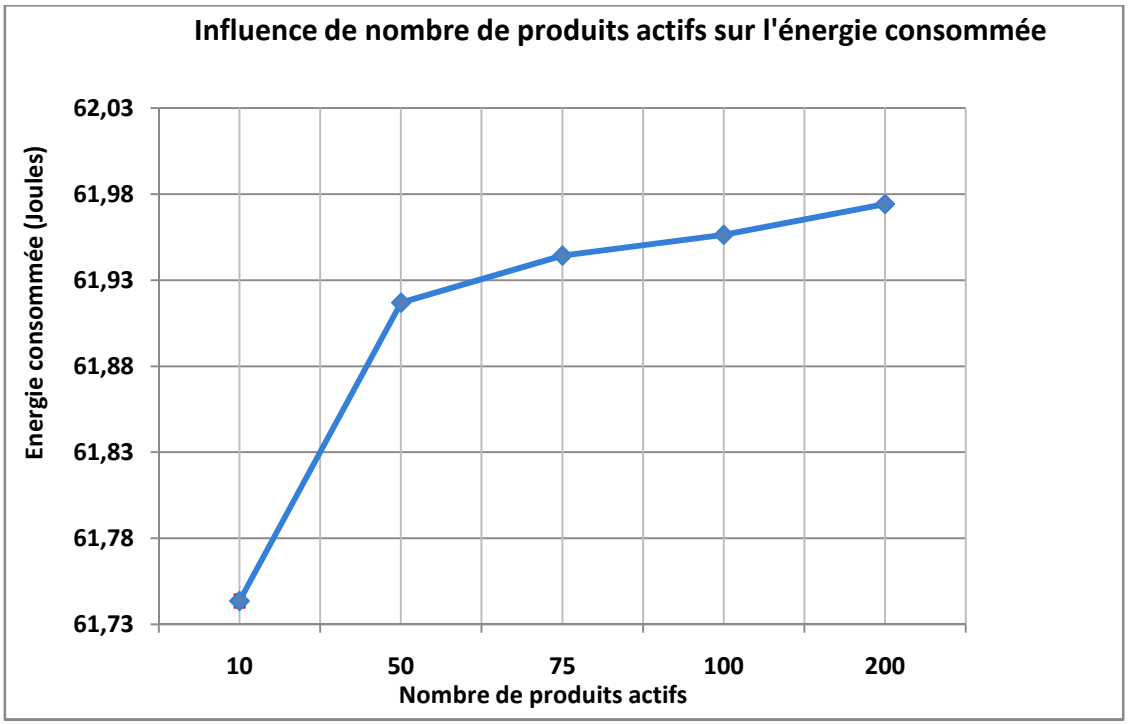

Figure IV. 32. Energie consommée par produit actif

Nous constatons que pour dix produits actifs dans l'entrepôt l'énergie moyenne consommée est égale à 61,74362867 Joules et l'intervalle de confiance de 95% de la valeur moyenne est égal à 0,00657061 Joules. Lorsque nous avons calculé le pourcentage de consommation de l'énergie pour un nombre de produits actifs égale à 10, nous avons trouvé que chaque nœud consomme en moyenne 0,124282666 % de l'énergie initiale (49680 Joules) pour une simulation qui a duré 1000s. De même pour 100 produits actifs l'énergie moyenne consommée est égale à 0,124711078 % de l'énergie initiale.

D'après [Shih et al., 2001], l'énergie liée a la communication est déterminée par la quantité des données à transmettre et la distance de transmission. Il serait souhaitable que les nœuds utilisent des paquets de données de taille réduite car plus la taille des paquets est réduite plus le nœud capteur économise de l'énergie durant la transmission.

Afin de bien étudier l'énergie consommée on a simulé l'énergie consommée pour une transmission de message. Les résultats sont donnés dans le tableau 10.

La consommation d'énergie dépend de la nature du message envoyé. Elle est faible pour les messages de taille faible par contre elle est grande pour les messages de taille grande.

Messages	**Energies consommés (Joules)**
Initialisation (CTR/ ACKCTR)	0.011764
Configuration (NCF0/ CMD1/ CMD3)	0,013863
Lecture des règles de sécurité (CMD4/SER)	0,05175
Lecture de paramètres de configuration (CMD2/CFG)	0,05175
Lecture informations ambiantes (CMD5/INA)	0,05175

Tableau 10. Energie consommée par produit actif en fonction de services

La conservation d'énergie peut être réalisée en réduisant la taille des paquets de données ainsi que le nombre de paquets transmis dans le réseau. Le protocole doit être le plus simple possible afin de consommer le minimum d'énergie.

Donc pour minimiser la consommation d'énergie il faut que la taille des messages échangés soient de taille faible par exemple nous pouvons coder les informations pour minimiser ses tailles.

V.8. Influence du nombre de produits sur la probabilité de perte des paquets

Dans le but de vérifier l'influence de l'ajout de produits actifs dans le comportement du réseau, on a lancé plusieurs fois la simulation en modifiant chaque fois le nombre de produits actifs dans l'entrepôt. Après, on extrait la probabilité des paquets perdus à partir des fichiers générés pour chaque simulation.

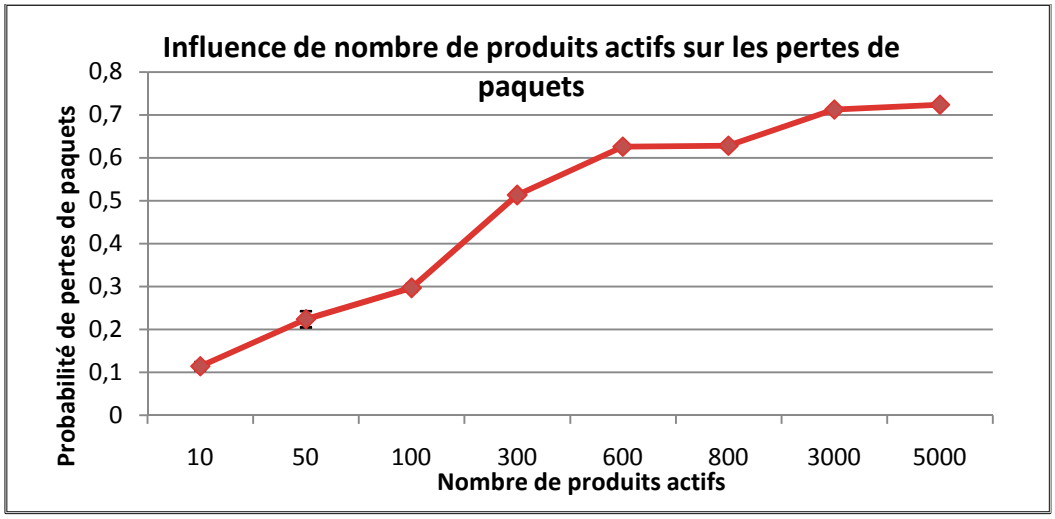

Figure IV. 33. Influence de l'incrémentation du nombre des produits sur la probabilité de perte des paquets

La figure IV.33 montre que la probabilité des paquets perdus dépasse 50% à partir d'un nombre de PAs égal à 300. Et elle est 22,390967% lorsque le nombre des PAs est égale à 50 avec un intervalle de confiance de 95% est égale à 3.76297%. A cet effet on remarque que la probabilité de perte de paquets avec cette architecture est une lacune de notre application.

VI. Conclusion

Dans ce dernier chapitre, on a exposé certains résultats de simulation obtenus après avoir implémenté et simulé le modèle de produits actifs sous Castalia. Ce simulateur nous a permis de vérifier la validité de ce modèle sous une architecture dont les caractéristiques sont très proches de la réalité tout en étudiant les critères de performances tels que la réactivité, l'étude énergétique et la mise de système en grande échelle. Mais, une implémentation de ce modèle dans un environnement réel reste une étape primordiale pour être certain de sa validité.

Conclusion Générale

L'objectif de cette thèse est la modélisation et la simulation du comportement, des interactions et des processus de coopération entre produits communicants dans un environnement à intelligence ambiante, intégrants des produits et objets physiques équipés d'intelligence embarquée et de moyen de communication sans fil (Réseau de capteurs, ...).

Les travaux menés dans ce mémoire se situent à l'intersection des domaines des réseaux de capteurs sans fil et l'intelligence ambiante. Nous avons étudié les principaux travaux de recherche dans le domaine des réseaux de capteurs et de l'intelligence ambiante, en regard plus particulièrement des applications de sécurisation. Cette étude fait l'objet du premier chapitre.

L'originalité de ce travail consiste en la définition d'un modèle adapté à une application avec des contraintes spécifiques en termes d'énergie et de délai de réponse. Nous proposons de limiter les traitements à une gestion/coordination des tâches au niveau de la couche application du modèle OSI.

Au cours de ce mémoire, Nous avons introduit le concept de produit actif et nous avons démontré la faisabilité d'accomplir une gestion de la sécurité active d'une zone de stockage pour les produits dangereux, basée sur les objets intelligents et la technologie des capteurs sans fil, en utilisant la simulation du comportement d'un produit actif orienté sécurité des biens et des personnes. Ce concept a pour idée d'équiper chaque produit par un dispositif physique admettant des capacités sensitives locales de surveillance ambiante, des capacités de communication et des règles pour prendre des décisions. Ainsi, un Produit à Intelligence Ambiante est capable de « sentir » son environnement, de décider de façon autonome et faire un choix d'action/réaction selon des spécifications propres et/ou partagées dans un environnement coopératif et de communiquer avec son environnement.

La hiérarchie de la modélisation par réseaux de Petri permet d'étaler l'évolution de chaque étape du fonctionnement de produit actif tout en représentant l'influence du réseau de communication. Ce genre de réseau de Petri devient un langage de programmation à niveau élevé et puissant, qui laisse répartir la complexité du code au niveau de la structure nette, des inscriptions et des conditions préalables des arcs.

Pour démontrer la faisabilité du concept de produit actif, nous avons fait le choix entre plusieurs simulateurs et nous avons choisi le simulateur Castalia/OMNeT++. Ce simulateur nous a permis de tester le fonctionnement d'un produit actif dans un environnement très proche des réseaux de capteurs sans fil.

Pour accomplir l'objectif de la thèse, on a implémenté le modèle de produit actif suivant des algorithmes définissant le comportement interne d'un produit actif dans son environnement. Ce comportement se manifeste dans l'interaction entre les différents produits actifs voisins et la détection des situations dangereuses qui peuvent se réaliser et la réaction en déclenchant des alertes appropriées pour prévenir les situations catastrophiques dans les zones dangereuses, tel qu'un domaine chimique. Nous avons cherché à déterminer les meilleures combinaisons de solutions permettant de réduire le temps de réaction de système. Disposant de tous les éléments pour lancer les simulations, le problème majeur est alors, le nombre de simulations à effectuer et le traitement des résultats obtenus. Cela a été résolu par l'emploi de plans d'expériences. Ces plans d'expériences permettent de limiter le nombre de simulations tout en gardant les paramètres cruciaux.

Ensuite, nous avons étudié l'efficacité du protocole suivant les métriques suivantes : réactivité, autonomie et facteur d'échelle. Les résultats de simulations ont montré une réactivité avec un taux d'erreur voisin de zéro. D'autre part, une consommation énergétique très faible a été observée pour un réseau de capteurs TelosB. De même, nous avons décris le comportement du système lors d'une apparition de panne, et lorsque le nœud est mobile.

Le concept de Sécurité Active proposé dans ce travail est innovant et ouvre de grandes perspectives et de nombreux axes de recherche dans les interactions produits à produits et produits à homme. Ainsi nous envisageons d'étendre le modèle de produit actif par la conception de l'interactivité avec un utilisateur en ajoutant les commandes nécessaires lui permettant de configurer/reconfigurer les produits. Cette modification augmente l'efficacité du modèle pour être plus adéquat aux cas pratiques.

La connaissance des facteurs ambiants d'un produit pourra être renforcée en utilisant la connaissance d'autres produits environnants qui possèdent d'autres capteurs sensoriels. Chaque produit connaît son état mais aussi, l'état des produits environnants et il peut utiliser ses règles pour arbitrer une situation sans avoir besoin de l'intervention d'un tiers. La modélisation du comportement interne du produit pour s'adapter à diverses situations sera donc poursuivie, et cela en l'étendant à une plus vaste communauté de produits actifs.

Une autre perspective est de proposer une technique de mise en veille des produits actifs qui peuvent être non concernées par le fonctionnement du système à un certain moment. Pour cela, il est indispensable de mettre les produits actifs de ces zones en mode veille afin de conserver leur énergie et par suite prolonger la vie du système. De même, une étude de l'impact de contrôle d'accès au medium par le protocole MAC pourra améliorer les réponses du système.

BIBLIOGRAPHIE

[AbdulRauf and Jeyakumar, 2008] H. AbdulRauf and A. E. Jeyakumar, "*Colored Petri Net modeling and throughput analysis for wireless infrastructure networks*", Ubiquitous Computing and Communication Journal, Vol. 3, No. 3, pp. 1-6, 2008.

[Aboelaze and Aloul, 2005] M. Aboelaze and F. Aloul, "*Current and future trends in sensor networks: a survey*", Second IFIP International Conference on Wireless and Optical Communications Networks (WOCN), March 2005, Dubai, UAE, pp. 551– 555, 2005.

[Agoston et al., 2000] T. Agoston, T. Ueda and Y. Nishimura, "*Pervasive computing in the networked world*", Proceedings of INET 2000, Yokohama, Japan, Oct.3-5, 2000, pp. 101-120, 2000.

[Akkaya and Younis, 2005] K. Akkaya and M. Younis, "*A Survey of Routing Protocols in Wireless Sensor Networks*", Ad Hoc Network Journal, Elsevier, Vol. 3, Issue 3, pp. 325-349, 2005.

[Akyildiz et al., 2002a] I. F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam and Eredal Cayirci, "*A Survey on Sensor Networks*", IEEE Communications Magazine, Vol. 40, Issue 8, pp. 102-114, 2002.

[Akyildiz et al., 2002b] I. F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam and Eredal Cayirci, "*Wireless sensor networks: a survey*", Computer Networks, Vol. 38, Issue 4, pp. 393-422, 2002.

[Alcaniz and Rey, 2005] M. Alcaniz, B. Rey, "*New technologies for ambient intelligence*", Ambient intelligence, section 1, chapter1, IOS Press, 2005.

[Al-Obaisat and Braun, 2006] Y. Al-Obaisat and R. Braun, "*On Wireless Sensor Networks: Architectures, Protocols, Applications and Management*", International Conference on Wireless Broadband and Ultra Wideband Communications (Auswireless 2006), March 13-16, Sydney, Australia, 2006.

[Augusto and Cook, 2007] J.C. Augusto and D. Cook. "*Ambient Intelligence: applications in society and opportunities for AI*", 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India, January 2007.

[Bahl and Padmanabhan, 2000] P. Bahl and V. N. Padmanabhan, "*RADAR: An In-Building RF-based User Location and Tracking System*", in IEEE Infocom, Vol. 2, Tel Aviv, Israel, March 2000, pp. 775–784, 2000.

[Bajic and Cea, 2005] E. Bajic, A. Cea, "*Smart objects and services modeling in the supply chain*", 16th Triennial IFAC World Congress, Prague, Czech Republic, 2005.

[Bajic, 2009] Eddy Bajic, "*A Service-Based Methodology for RFID-Smart Objects Interactions in Supply Chain*", International Journal of Multimedia and Ubiquitous Engineering, Vol. 4, No. 3, pp. 37-54, 2009.

[Beigl et al., 2001] M. Beigl, H. W. Gellersen and A. Schmidt, "*MediaCups: Experience with Design and Use of Computer-Augmented Everyday Artefacts*", Computer Networks, Special Issue on Pervasive Computing, Elsevier, Vol. 35, No. 4, pp. 401-409, 2001.

[Beigl et al., 2003a] M. Beigl, T. Zimmer, A. Krohn, C. Decker et P. Robinson, "*Smart-Its – Communication and Sensing Technology for UbiComp Environments*", Technical Report ISSN 1432-7864 2003/2, 2003.

[Beigl et al., 2003b] M. Beigl, A. Krohn, T. Zimmer, Christian Decker and P. Robinson, "*AwareCon: Situation Aware Context Communication*", International conference on Ubiquitous Computing (Ubicomp), October 12-15, Seattle, USA, 2003.

[Beigl, 2000] M. Beigl, "*Memoclip: A location based Remembrance Applicance*", Personal and Ubiquitous Computing, Volume 4, Issue 4, Springer-Verlag, 2000.

[Benoist et al., 1994] D. Benoist, Y. Tourrbier et S. Germain-Tourbier, "*Plan d'expériences : Construction et analyse*", coll. Tec & Doc, Lavoisier, Paris, 693 pp., 1994.

[Bitam and Alla, 2006a] Bitam Melha et Alla Hassane, "*L'outil réseaux de Petri hybrides dans les réseaux de communication : Dynamique des transmissions et étude de comportement*", Journal européen des systèmes automatisés, JESA, vol. 40, pp. 73-94, 2006.

[Bitam and Alla, 2006b] M. Bitam and H. Alla, "*Performance evaluation of communication networks for distributed systems*", International Journal of Computer Applications in Technology, Vol. 25, Number 4, pp. 218-226, 2006.

[Bitam, 2005] Melha Bitam, "*Modélisation et étude de comportement d'une ligne de communication TCP/IP*", Thèse de doctorat de l'Université Joseph-Fourier - Grenoble I, France, juin 2005.

[Bonhomme, 2001] P. Bonhomme, "*Réseaux de Petri P-temporel contribution à la commande robuste*", Thèse de Doctorat, Université de Savoie, France, 2001.

[Bostwick et al., 2009] Daniel Bostwick, Jacob Goldstein, Thomas Stephenson, Sean Stromsten, Jorge Tierno, Michelle Torrelli, and James White "*PARSEC, an Application of Probabilistic Case Based Reasoning to Maritime Surveillance*", IEEE International Conference on Technologies for Homeland Security (HST 2009), May 11 - 12, Boston, MA, pp. 73 – 79, 2009.

[Boulis, 2009] A. Boulis, "*Castalia, a simulator for wireless sensor networks and body area networks, version 2.2*", User's manual, NICTA, August 2009.

[Brahimi et al., 2006] B. Brahimi, C. Aubrun and E. Rondeau, "*Modelling and Simulation of Scheduling Policies Implemented in Ethernet Switch by Using Coloured Petri Nets*", 11th IEEE International Conference on Emerging Technologies and Factory Automation, Czech Republic, 2006.

[Brahimi, 2007] B. Brahimi, "*Proposition d'une approche intégrée basée sur les réseaux de Petri de Haut Niveau pour simuler et évaluer les systèmes contrôlés en réseau*", Thèse de doctorat, l'Université Henri Poincaré, Nancy I, France, 2006.

[Buhrig and Renaudin, 2007] A. Buhrig et M. Renaudin, "*Gestion de la consommation des nœuds de réseau de capteurs sans fil*", 10ème édition des Journées Nationales du Réseau Doctoral en Microélectronique (JNRDM'07), Lille, 2007.

[Caballero et al., 2008] F. Caballero, L. Merino, P. Gil, I. Maza and A. Ollero, "*A probabilistic framework for entire WSN localization using a mobile robot*", Robotics and Autonomous Systems Journal, Vol. 56, pp. 798–806, 2008.

[Caron, 2000] E. Caron, "*Calcul numérique sur données de grande taille*", Thèse de Doctorat, Université de Picardie Jules Verne, France, 2000.

[Cartron and Sentieys, 2005] M. Cartron et O. Sentieys, "*Optimisation énergétique d'un système de communication dédié à un réseau de capteurs*", 20ième Colloque sur le traitement du signal et des images, Lannion, France, pp 1100-1103, 2005.

[Castelluccia et al., 2005] C. Castelluccia, E. Mykletun etand G. Tsudik, "*Efficient Aggregation of Encrypted Data in Wireless Sensor Networks*", ACM/IEEE Mobiquitous Conference, July 2005, San Diego, USA, 2005.

[Cea and Bajic, 2004] A. Cea, E. Bajic, "*Ambient Services for Smart Objects in the Supply Chain Based on RFID and UPnP Technology*", Third Conference on Management and Control of Production and Logistics MCPL'04, Santiago, Chile, November 3–5, 2004.

[Cea, 2006] Aldo Alexis CEA RAMIREZ, "*Contribution à la Modélisation et à la Gestion des Interactions Produit-Processus dans la Chaîne Logistique par l'Approche Produits Communicants*", Thèse de Doctorat, Université Henri Poincaré, Nancy I, France, Juillet 2006.

[Cerf and Kahn, 1974] V. G. Cerf and R. E. Kahn, "*A protocol for packet network interconnection*", IEEE Transactions on Communications Technology, Vol. 22, Issue 5, pp. 627–641, 1974.

[Chen et al., 2007] X. Chen, H. Refai and X. Ma, "*Quantitative Approach to Evaluate DSRC Highway Intervehicle Safety Communication*", In IEEE Global communications Conference, GLOBECOM 2007, November 26-30, Washington, DC, USA, pp. 151–155, 2007.

[Chen et al., 2008] J. Chen, J. Zhang, W. Xu, L. Shu and Y. Sun, "*The Development of a Realistic Simulation Framework with OMNeT++*", Second International Conference on Future Generation Communication and Networking (FGCN '08), December 13-15, 2008, Horizon Resort, Sanya, Hainan Island, China, pp. 497-500, 2008.

[Chintalapudi et al., 2006] K. Chintalapudi, T. Fu, J. Paek, N. Kothari, S. Rangwala, J. affrey, R. Govindan, E. Johnson and S. Masri, "*Monitoring Civil Structures with a Wireless Sensor Network*", IEEE Computer Society, Washington, DC, USA, pp. 1-9, 2006.

[CoBIs, 2008] CoBIs. "*Collaborative business items*", European Community FP6 STREP Project, IST 004270, Technical report, 2OO0082008. (www.cobis-online.de).

[Corroy et al., 2009] S. Corroy, J. Beiten, J. Ansari, H. Baldus and P. Mähönen, "*Selection of Computing Elements for Energy Efficiency in Wireless Sensor Networks using a Statistical Estimation Method*", International Journal On Advances in Networks and Services. Vol. 2, No. 3, 2009.

[David and Alla, 1989] R. David and H. Alla, "*Du grafcet aux réseaux de Petri*", Edition Hermès, Paris, 1989. (ISBN: 2-86601-325-5).

[David et al., 1988] A. P. David, G. Gibson and R. H. Katz, "*A case for Redundant Arrays of Inexpensive Disks*", ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, June 1-3, pp. 109-116, 1988.

[David et al., 2004] David Culler, Deborah Estrin, and Mani Srivastava, "*Overview of sensor networks*", IEEE Computer Society, Vol. 37, Issue 8, August 2004, pp. 41-49, 2004.

[Decker et al., 2004a] C. Decker, M. Beigl, A. Eames and U. Kubach, "*DigiClip: Activating Physical Documents*", 24th International Conference on Distributed Computing Systems Workshops (ICD-CSW'04), March 23-26, Tokyo, Japan, pp. 388 – 393, 2004.

[Decker et al., 2004b] C. Decker, M. Beigl, A. Krohn, P. Robinson and U. Kubach, "*eSeal - A System for Enhanced Electronic Assertion of Authenticity and Integrity of Sealed Items*", Second International Conference on Pervasive Computing, April 18-23, Linz, Vienna, Austria, 2004.

[Demers et al., 2003] A. Demers, J. Gehrke, R. Rajaraman, N. Trigoni, and Y. Yao, "*Energy-Efficient Data Management for Sensor Networks, A Work-In-Progress Report*", 2nd IEEE Upstate New York Workshop on Sensor Networks, Syracuse, NY, October 2003.

[Demirkol et al., 2006] I. Demirkol, C. Ersoy and F. Alagöz, "*MAC Protocols for Wireless Sensor Networks: A Survey*", IEEE Communications Magazine, Vol. 44, Issue 4, pp. 115-121, 2006.

[Demonsant, 1996] Jacques Demonsant, "*Comprendre et mener des plans d'expériences*", édition AFNOR, 1996. (ISBN : 2-12-475032-1).

[Devisme, 2006] Stéphane Devisme, "*Quelques Contributions à la Stabilisation Instantanée*", Technical Report 2006-09, LaRIA, CNRS FRE 2733, 2006.

[Dey, 2000] A. Dey, "*Providing Architectural Support for Building Context-Aware Applications*", PhD Thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2000.

[Dobre and Bajic, 2007] D. Dobre, E. Bajic, "*Smart object design for active security management of hazardous products*", International conference on Ubiquitous Computing (Ubicomp 2007), Sbruck, Austria, Septembre 16-18, 2007.

[Dobre et al., 2009] Dragos Dobre, Eddy Bajic, Ahmed Zouinkhi, "*Active Product Modeling based on Smart Object Concept: Application to Chemical Security Management*", Journal Européen des Systèmes Automatisés, JESA, N° 4-5/2009, pp 559-578, 2009.

[Drira, 2005] K. Drira, "*Contribution à la conception des architectures logicielles et des protocoles de coordination pour les systèmes distribués coopératifs*", Thèse de Doctorat, Université Paul Sabatier Toulouse III, France, 2005.

[Englund and Wallin, 2004] C. Englund, and H. Wallin, "*RFID in Wireless Sensor Network*", Technical Report, Department of Signals and Systems, Chalmers University of Technology, Sweden, 2004.

[Finkenzeller, 2003] K. Finkenzeller, "*RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*", Second Edition, 2003, John Wiley & Sons, 2003. (ISBN: 0-470-84402-7).

[Frederix et al., 2006] F. Frederix, , P. Jaronski , P. Friess, "*Workshop on Ambient Intelligence Technologies to Enhance the Product Lifecycle*", Workshop report, February 27, Brussels, 2006.

[Fumolari, 2001] D. Fumolari, "*Link performance of an embedded Bluetooth personal area network*", In Proceedings of IEEE International Conference on Communications (IEEE ICC'01), vol. 8, pp. 2573–2577, 2001.

[Garate et al., 2005] A. Gárate, N. Herrasti , A. López, "*GENIO: an Ambient Intelligence Application in Home Automation and Entertainment environment*", Joint Conference on Smart Objects and Ambient Intelligence, 2005.

[Gershenfeld et al., 2004] N. Gershenfeld, R. Krikorian, D. Cohen, "*The internet of things*", Scientific American 291 (4), pp.76–81, 2004.

[Ghosh et al., 2009] Somnath Ghosh, Prakash Veeraraghavan, Samar Singh and Lei Zhang, "*Performance of a Wireless Sensor Network MAC Protocol with a Global Sleep Schedule*", International Journal of Multimedia and Ubiquitous Engineering, Vol. 4, No. 2, 2009.

[Giel et al, 2008] A. Timm-Giel, K. Murrah, M. Becker, C. Lynch, C. Gorg and D. Pesch, "*Comparative Simulations of WSN*", ICT-Mobile Summit, Stockholm, Sweden, June 10-12, 2008.

[Goupy and Creighton, 2006] Jacques Goupy, Lee Creighton, "*Introductions aux plans d'expériences*", 3ième édition Dunod, Paris, 2001, 2006. (ISBN : 2 10 0497448).

[Hajjej et al., 2010] Noureddine Hajjaji, Viviane Renaudin, Ammar Houas and Marie Noëlle Pons, "*Factorial design of experiment (DOE) for parametric energetic investigation of a steam methane reforming process for hydrogen production*", Chemical Engineering and Processing, Vol. 49, pp. 500–507, 2010.

[Hatayama et al., 1980] H. K. Hatayama, E.R. de Vera, B.P. Simmons, R.D. Stephens, and D.L. Storm, "*Hazardous waste compatibility*", In: Proc. Sixth Ann. Research Symposium on

Disposal of Hazardous Waste, EPA/600/9-80-010, U.S. EPA, Cincinnati, OH, pp. 21-28, 1980.

[Heinzelman et al., 1999] W. R. Heinzelman, J. Kulik, H. Balakrishnan, "*Adaptive protocols for information dissemination in wireless sensor networks*", Proceedings of the ACM MobiCom'99, Seattle, Washington, pp. 174–185, 1999.

[Heinzelman et al., 2000] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "*Energy-efficient communication protocol for wireless microsensor networks*", IEEE Proceedings of the Hawaii International Conference on System Sciences, pp. 1–10, 2000.

[Hoehmann et al., 2009] L. Hoehmann and A. Kummert, "*Mobility Support for Wireless Sensor Networks Simulations for Road Intersection Safety Applications*", In 52nd IEEE International Midwest Symposium on Circuits and Systems, August 2-5, Cancun, Mexico , pp. 260-263, 2009.

[Hoyle, 2005] B. S. Hoyle, "*Cooperating sensor nodes in spatially critical networks for smart measurement and monitoring systems*", Proc. Smart Object Systems Workshop, UbiComp 2005, Tokyo, Japan, pp. 77-82, 2005.

[Hsieh, 2009] F. S. Hsieh, "*Developing cooperation mechanism for multi-agent systems with Petri nets*", Engineering Applications of Artificial Intelligence, Vol. 22, No. 4, pp. 616–627, 2009..

[Huvio et al., 2002] E. Huvio, J. Gronvall, K. Framling, "*Tracking and tracing parcels using a distributed computing approach*", in Proceedings of NOFOMA 2002 Conference, June 13-14, Trondheim, Norway, pp. 29–43, 2002.

[Intanagonwiwat et al., 2000] C. Intanagonwiwat, R. Govindan and D. Estrin, "*Directed diffusion: a scalable and robust communication paradigm for sensor networks*", Proceedings of the ACM Mobi-Com'2000, Boston, MA, pp. 56–67, 2000.

[Jansen-Vullers et al., 2003] M. H. Jansen-Vullers, C. A. Van Dorp and A. J. M Beulensb, "*Managing traceability information in manufacture*", International Journal of Information Management, Vol. 23, pp. 395–413, 2003.

[Jensen, 1981] K. Jensen, "*How to find invariants for Coloured Petri Nets*", 10th symposium on Mathematical foundations of computer science 1981, Springer-Verlag, pp 327-338, 1981.

[Jensen, 1982] K. Jensen, "*High-level Petri nets*", Third European workshop on applications and theory of Petri Nets, Varenna, Italy, september1982, pp. 261-276, 1982.

[Jensen, 1997] K. Jensen, "*Coloured Petri Nets: Basic Concepts, Analysis Methods and Practice Use*", Volumes 1-3, Monographs in Theoretical Computer Science, Springer-Verlag, Berlin, 1997.

[Juanole et al., 2004] G.Juanole, M.Diaz, et F.Vernadat, "*Les réseaux de Petri étendus et méthodologie pour l'analyse de performances*", In the book 'Méthodes exactes d'analyse de performance des réseaux'. IC2 Réseaux et Télécoms (Information –Commande – Communication), Edition Hermès science, 2004.

[Kärkkäinen et al., 2003] M. Kärkkäinen, J. Holmström, K. Främling, K. Artto, "*Intelligent products-a step towards a more effective project delivery chain*", Computer in Industry, Vol. 50, Elsevier, pp.141-151, 2003.

[Karl and Willig, 2005] Holger Karl and Andreas Willig, "*Protocols and Architectures for Wireless Sensor Networks*", Jhon Wiley & Sons, 2005. (ISBN: 0470095105).

[Kashif et al ., 2009] Saleem Kashif, Norsheila Fisal, Sharifah Hafizah, Sharifah Kamilah and Rozeha Rashid, "*Autonomously Intelligent WSN Routing Protocol based on Ant Colony Optimization*", 7th International Conference on Robotics, Vision, Signal Processing & Power Applications (RoViSP 2009), Awana Porto Malai, Langkawi, Kedah, Malaysia, 2009.

[Khoukhi and Cherkaoui, 2010] L. Khoukhi and S. Cherkaoui, "*Intelligent QoS management for multimedia services support in wireless mobile ad hoc networks*", Journal of Computer Networks, Elsevier Edition, Vol 54, No.10, pp. 1692-1706, 2010.

[Kim et al., 2008] M. Kim, E. Jeong, Y. Bang, S. Hwang, C. Shin, G. Jin and B. Kim, "*An energy-aware multipath routing algorithm in wireless sensor networks*", IEICE Transactions on Information and Systems, Vol. E91-D, Issue 10, pp. 2419–2427, 2008.

[Kristensen et al., 1998] L. M Kristensen, S. Christensen and K. Jensen, "*The Practitioner's Guide to Coloured Petri Nets*", International Journal on software Tools for technology Transfer, Vol. 2, Issue 2, pp. 98–132, 1998.

[Krohn et al., 2004] A. Krohn, M. Beigl, C. Decker ,T. Zimmer, "*ConCom: A language and protocol for communication of context*", Technical Report, ISSN: 1432-7864 2004/19 University of Karlsruhe, 2004.

[Kulkarni and Pao, 2005] V. Kulkarni and L. Y. Pao, "*A Sensor Management Protocol for Tracking with Diverse Sensors*", American Control Conference 2005, Portland, OR, USA, Vol. 7, pp. 5015- 5020, 2005.

[Kuntz and Noël, 2009] R. Kuntz, T. Noël, "*Un protocole d'accès au médium orienté mobilité et réseaux de capteurs*", Colloque francophone sur l'ingénierie des protocoles (CFIP'09), Octobre 2009, France, pp. 75-86, 2009.

[Kurata et al., 2005] N. Kurata, B. F. Spencer Jr. and M. Ruiz-Sandoval, "*Risk monitoring of buildings with wireless sensor networks*", Structural Control and Health Monitoring, Special Issue: Advanced Sensors and Health Monitoring, pp. 315-327, 2005.

[Langheinrich, 2002] M. Langheinrich, "*Privacy Invasions in Ubiquitous Computing*", Workshop on Socially-informed Design of Privacy-enhancing Solutions, Goteborg, Sweden Ubicomp, 2002.

[Li and Halpern, 2001] L. Li et J.Y. Halpern, "*Minimum-energy mobile wireless networks revisited*", IEEE International Conference on Communications ICC'01, Helsinki, Finland, 2001.

[Long, 1993] J. Long, "*Sur la conduite hiérarchisée des systèmes flexibles de production*", Thèse de Doctorat à l'Institut National Polytechnique de Grenoble, France, 1993.

[Madden et al., 2003] S. Madden, M. Franklin, J. Hellersterin, W. Hong, "*The design of an acquisitional query processor for sensor networks*", Proceedings of SIGMOD Conference, San Diego, California, USA, 2003.

[Maimour, 2007] Moufida Maimour, "*Multipath routing protocol for layered video transport in wireless sensor networks*", 7th International Conference on New Technologies of Distributed Systems, Marrakesh, Marocco, 2007.

[Makkaoui et al., 2010] Leila Makkaoui, Vincent Lecuire and Jean-Marie Moureaux, "*Efficacité énergétique d'une DCT zonale rapide dans le contexte de la compression d'image dans les réseaux de capteurs sans fil*", Conférence Compression et Représentation des Signaux Audiovisuels CORESA 2010, 26 et 27 octobre 2010, Lyon, France, 2010.

[Marsal, 2006] G. Marsal, "*Evaluation of time performances of Ethernet-based Automation System by simulation of High Level Petri Nets*", Thèse de Doctorat, Ecole Normale Supérieure de Cachan, Université de Kaiserslautern, 2006.

[Mattern and Sturm, 2003] F. Mattern and P. Sturm, "*From Distributed Systems to Ubiquitous Computing – The State of the Art, Trends, and Prospects of Future Networked Systems*", In Klaus Irmscher and Klaus-Peter Fähnrich, editors, Proceedings KIVS, pp. 3-25, Springer-Verlag, February 2003.

[Mattern, 2000] F. Mattern, "*State of the art and future trends in distributed systems and ubiquitous computing*", Vontobel TeKnoBase, 2000.

[Mattern, 2005] F. Mattern, "*Ubiquitous Computing: Scenarios from an informatized world*", E-Merging Media-Communication and the Media Economy of the Future, Springer-Verlag, pp. 145-163, 2005.

[McFarlane et al., 2002] D. McFarlane, S. Sarma, C. J. Lung, C. Y. Wong, K. Ashton, "*The intelligent product in manufacturing control and management*", 15th Triennial World Congress, Barcelona, Spain. 2002.

[McFarlane et al., 2003] D. McFarlane, S. Sarma, J. L. Chirn, C.Y.. Wong, K. Ashton, "*Auto id systems and intelligent manufacturing control*", Engineering Applications of Artificial Intelligence, Vol. 16, Issue 4, pp. 365–376, 2003.

[McFarlane, 2003] D. McFarlane, "*Product Identity and Its Impact on Discrete Event Observability*", European Control Conference ECC 2003, Cambridge, Septembre 2003.

[Medjiah et al., 2009] S. Medjiah, T. Ahmed, F. Krief, P. Gélard, "*AGEM : Un Protocole de Routage Géographique angulaire Adaptatif*", Colloque Francophone sur l'Ingénierie des Protocoles, CFIP'09, 12-15 Octobre 2009, Strasbourg, France, 2009.

[Meier et al., 2008] A. Meier, M. Motani, H. Siquan and S. Künzli, DiMo, "*Distributed Node Monitoring in Wireless Sensor Networks*", In MSWiM 2008, October 27-31, 2008, Vancouver, BC, Canada, 2008.

[MIT Project Oxygen, 2002] MIT Project Oxygen, "*Pervasive, Human - centered computing*", MIT Project Oxygen, MIT Laboratory For Computer Science, MIT Artificial Intelligence Laboratory, Second Printing 2002.

[Moncrieff et al., 2007] S. Moncrieff, S. Venkatesh and G. West, "Dynamic privacy in a smart house environment", IEEE International Conference Multimedia and Expo, Beijing, China, 2007.

[Newton and Welsh, 2004] R. Newton and M. Welsh, "*Region streams: Functional macro programming for sensor networks*", Proceedings of the First Workshop on Data Management for Sensor Network, (DMSN'04), Toronto, Canada, 2004.

[Niculescu and Nath, 2001] D. Niculescu and B. Nath, "*Ad hoc positioning system*", in IEEE Globecom, San Antonio, TX, USA, November 2001, pp. 2926–2931, 2001.

[Norman, 1998] D.A. Norman, "*The Invisible Computer: Why Good Products can Fail, the Personal Computer is so Complex, and Information Appliances are the Solution*", MIT Press, 1998.

[Pang et al., 2008] Q. Pang, V. W.S. Wong and Victor C.M. Leung, "*Reliable data transport and congestion control in wireless sensor networks*", International journal of sensor networks, Vol. 3, No. 1, pp. 16-24, 2008.

[Pham et al., 2007] H. N. Pham, D. Pediaditakis, and A. Boulis, "*From Simulation to Real Deployments in WSN and Back*", IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2007, 18-21 June 2007, Helsinki, Finland, pp. 1-6, 2007.

[Pottie and Kaiser, 2000] G.J. Pottie and W. J. Kaiser, "*Wireless Integrated Network Sensors*", Communications of the ACM, Vol. 43, No. 5, pp. 51-58, 2000.

[Privat, 2000] G. Privat, "*A system architecture view point on smart networked devices*", Microelectronic Engineering, Vol. 54, pp. 193-197, 2000.

[Quanz and Tsatsoulis, 2008] B. Quanz and C. Tsatsoulis, "*Determining Object Safety using a Multiagent Collaborative System*", Workshop at Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems ECOSOA 2008, Venice, Italy, October 20-24, 2008.

[Rajeet et al., 2005] C. Rajeet, P. Wolfe, S. Park, J. Choi, "*Evaluation of using RFID passive tags for monitoring product location / ownership*", In Proceedings of the 2004 IIE Annual Conference, Houston, TX, 2005.

[Ramchandani, 1974] C. Ramchandani, "*Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*", Thèse de Doctorat, Cambridge, MIT, 1974.

[Robinson et al., 2007] C. L. Robinson, D. Caveney, L. Caminiti, G. Baliga, K. Laberteaux and R. Kumar, "*Effecient Message Composition and Coding for Cooperative Vehicular Safety Applications*", IEEE Transactions on Vehicular Technology, Vol. 56, Issue 6, 2007, pp. 3244-3255, 2007.

[Ronzani, 2009] Daniel Ronzani, "*The battle of concepts: Ubiquitous Computing, pervasive computing and ambient intelligence in Mass Media*", Ubiquitous Computing and communication journal, Vol. 4, No. 2, 2009.

[Roy, 2004] D. Roy, D. Anciaux, T. Monteiro and L. Ouzizi, "*Multi-agents architecture for supply chain management*", Journal of Manufacturing Technology Management, Vol. 15, Issue 8, pp. 745-755, 2004.

[Saha and Mukherjee, 2003] Debashis and Amitava Mukherjee, "*Pervasive computing: a paradigm for the 21st century*", IEEE Computer journal, Vol. 36, No. 3, pp. 25-31, 2003.

[Samper, 2008] Ludovic Samper, "*Modélisations et Analyses de Réseaux de Capteurs*", Thèse de Doctorat, Institut National Polytechnique de Grenoble, France, 2008.

[Sandeep et al., 2007] Sandeep P. Abhang, and Girish V. Chowdhay, "*WDM-Based Storage Area Network (SAN) for Disaster Recovery Operations*", International Journal of Computer and Information Engineering, Vol. 1, Issue 4, pp. 249-252, 2007.

[Saporta, 2006] Gilbert Saporta, "*Probabilités, analyse des données et statistique*", Edition Technip, 2e édition, 2006. (ISBN : 978-2-7108-0814-5).

[Savarese et al., 2002] C. Savarese, J. M. Rabaey, and K. Langendoen, "*Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks*", in USENIX Annual Technical Conference, Philadelphia, PA, USA, June 2002, pp. 317–327, 2002.

[Schilit et al., 1994] B. Shilit, N. Adams and R. Want, "*Context aware computing application*", IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, USA, pp. 85-90, 1994.

[Schmidt et al., 1998] A. Schmidt, M. Beigl and H.W. Gellersen, "*There is more to context than location*", in International workshop of Interactive Applications of Mobile Computing, Rostock, 1998.

[Schmidt, 2002] A. Schmidt, "*Ubiquitous Computing – Computing Context*", Thèse de Doctorat, Computing Departement Lancaster Universtity, United Kingdom, 2002.

[Senn, 2007] P. Senn, "*Objets communicants et nanotechnologie*", Journées Scientifiques du CNRFS Nanosciences et Radioélectricité, 2007.

[Shih et al., 2001] E. Shih, S. H. Cho, N. Ickes, R. Min, A. Sinha, A.Wang, A.Chandrakasan, "*Physical Layer driven protocol and algorithm design for energy-efficient wireless sensors networks*", in Proceeding of international Conference on Mobile computing and networking (Mobicom), July 16-21, Rome, Italy 2001.

[Sila, 2006] Sila Filfli, "*Optimisation Bâtiment/Système pour minimiser les consommations dues à la climatisation*", Thèse de Doctorat, Ecole de Mines de Paris, 2006.

[Singh et al., 2005] S. Singh, S. Puradkar et Y. Lee, "*Ubiquitous Computing: connecting pervasive computing through semantic web*", school of Computing and engineering, University of Missouri-Kansas, 2005.

[Sohraby et al., 2007] K. Sohraby, D. Minoli, T. Znati, "*Wireless Sensor Networks Technology, Protocols, and Applications*", John Wiley & Sons, Inc. Printed in the United States of America, 2007. (ISBN: 978-0-471-74300-2).

[Song et al., 2008] C. Song, G. Qi-Wei, S. Qian-Ming and Z. Qian, "*Modeling and performance analysis of wireless sensor network systems using Petri nets*", 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2008), July 6-9, 2008, Kaikyo Messe Shimonoseki, Shimonoseki City, Yamaguchi, Japan, pp. 1689-1692, 2008.

[Souvay, 1995] Pierre Souvay, "*Les plans d'expériences, Méthode Taguchi*", Afnor collection- A savoir, 1995. (ISBN: 2-12-475028-3).

[Strohbach et al., 2005] M. Strobach, G. Kotuem and H. Gellersen, "*Cooperative Artefacts - A Framework for embedding knowledge in real world object*", International Workshop on Smart Object Systems, UbiComp, pp. 91–99, Tokyo, Japan, 2005.

[Taisir et al., 2006] E. Taisir, H. El-Gorashi, B. Pranggono, and M. H. Elmirghani, "WDM Metropolitan Sectioned Ring for Storage Area Networks Extension with Symmetrical and Asymmetrical Traffic", IEEE International Conference on Communications (IEEE ICC 2006), Istanbul, 2006.

[Tel, 2001] Gerard Tel, "*Introduction to distributed algorithms*", Cambridge University, Press New York, NY, USA, Second edition, 2001. (ISBN: 0521794838).

[Thiare, 2007] O. Thiare, "*Exclusion mutuelle de groupe dans les systèmes distribués*", Thèse de Doctorat, Université de Cergy Pontoise, Laboratoire Informatique de Cergy-Pontoise, 2007.

[Titzer et al., 2005] B. L. Titzer, D. K. Lee, J. Palsberg, "*Avrora: scalable sensor network simulation with precise timing*", 4th international symposium on Information processing in sensor networks (IPSN2005). Piscataway, NJ, USA, 2005.

[Tomic, 2006] S. Tomic, "*Network-Growing Scenarios in IEEE 802.15.4 Wireless Sensor Networks*", 25th Conference on Computer Communications, IEEE INFOCOM 2006, Barcelona, Catalunya, Spain, 2006.

[Tschirner et al, 2008] S. Tschirner, L. Xuedong and W. Yi, "*Model-Based Validation of QoS Properties of Biomedical Sensor Networks*", Proceedings of the 8th ACM international conference on Embedded software Atlanta, GA, USA, pp 69-78, 2008.

[Ullman, 1998] Jeffery-D Ullman, "*Elements of ML programming*", Pearson Education, 2nd edition, 1998. (ISBN: 9780137903870).

[Varga et al., 2008] A. Varga, R. Hornig, "*An overview of the OMNeT++ simulation environment*", Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Marseille, France, 2008.

[Varga, 2001] András Varga, "*The OMNeT++ Discrete Event Simulation System*", the Proceedings of the European Simulation Multiconference (ESM'2001). Prague, Czech Republic, June 6-9, 2001. pp 319-324, 2001.

[Vaudour and Gauthier, 2006] J. Vaudour et V. Gauthier, "*Comparaison de différentes couches MAC pour les réseaux de capteurs*", Rapport de Recherche INT N_06006RST, 2006.

[Ventä, 2007] O. Ventä, "Intelligent products and systems", Technical Report, VTT, 2007.

[Wei et al., 2004] Ye Wei, John Heidemann, and Deborah Estrin, "*Medium access control with coordinated adaptive sleeping for wireless sensor networks*". IEEE/ACM Transactions on Networking, Vol. 12, No.3, pp. 493-506, 2004.

[Weiser and Brown, 1996] M. Wieser and J. S. Brown, "Design *Calm technology*", PowerGrid Journal, Vol. 1, Issue 1, pp. 75-85, 1996.

[Weiser, 1991] M Weiser, "*The computer for the 21st century*", Scientific American, Vol. 265, Issue 3, pp. 94–104, 1991.

[Wieser et al., 1999] M. Wieser, R. Gold and S. Brown, "*The origins of ubiquitous computing research at PARC in the late 1980s*", IBM system journal, Pervasive Computing, Vol. 38, Issue 4, 1999.

[Wieser, 1993a] M. Wieser, "*Ubiquitous Computing*", IEEE Computer, Vol. 26, No. 10, October 1993, pp. 71-72, 1993.

[Wieser, 1993b] M. Wieser, "*The world is not a desktop*", ACM Transactions, 1993.

[Wong et al., 2002] C.Y. Wong, D. McFarlane, A. Zaharudin and V. Agarwal, "*The Intelligent Product Driven Supply Chain*", IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia, 2002.

[Xiang, 2002] L. Xiang, "*Pervasive (ubiquitous) computing: Challenges to Embedded System Engineering*", School of Software, Peking University, 2002.

[Xue et al., 2007] Y. Xue, H. Sung Lee, M. Y. Kumarawadu, P. Ghenniwa and H.H. Weiming Shen, "*Performance Evaluation of NS-2 Simulator for Wireless Sensor Networks*", Canadian Conference on Electrical and Computer Engineering (CCECE 2007), pp. 1372-1375, 2007.

[Yesid et al., 2009] Yesid Jarma, Golnaz Karbaschi, Marcelo Dias de Amorim, Farid Benbadis and Guillaume Chelius, "*VAPS: Positioning with Spatial Constraints*", 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia networks (IEEE WoWMoM), Kos, Greece - June 2009.

[Yoo and Kalle, 2005] Youngjin Yoo and Lyytinen Kalle, "*Editorial Social Impacts of ubiquitous computing: exploring critical interactions between mobility, context and technology*", A special issue for Information and Organization, Vol. 15, pp. 91-94, 2005.

[Zhou et al., 2006] Z. Zhou, X. Xiang and X. Wang, "*An Energy-Efficient Data-Dissemination Protocol in Wireless Sensor Networks*", IEEE Computer Society, Washington, DC, USA, pp. 13 – 22, 2006.

[Zouinkhi et al., 2007] A. Zouinkhi, E. Bajic, M. Ben Gayed et M. N. Abdelkrim, "*Modèle de Produits Actifs et Communication Ambiante pour la Gestion de la Sécurité de Produits Dangereux*", STA'2007, 05-07 Novembre, Monastir, Tunisie, 2007.

[Zouinkhi et al., 2008] A. Zouinkhi, E. Bajic, R. Zidi, M.K. BenGayed, M.N. Abdelkrim and E. Rondeau, "*Modelling by Petri Nets of an active product for the Management of the Security of Dangerous Products*", The Ninth international conference on Sciences and Techniques of Automatic control & computer engineering (STA'2008), December 20-23, 2008, Sousse, Tunisia, 2008.

[Zouinkhi et al., 2009a] A. Zouinkhi, E. Baijc, R. Zidi, M.K. Bengayed, M.N. Abdelkrim and E. Rondeau, "*Petri Nets Modelling of active products cooperation for active security management*", International Multi conference on systems, Signal & Devices, conference

sensors, circuits & Instrumentation system, SSD'2009, 23-26 March 2009, Djerba –Tunisia, 2009.

[Zouinkhi et al., 2009b] Ahmed Zouinkhi, Eddy Bajic, Raja ZIDI, Mohamed Ben Gayed, Eric RONDEAU et Mohamed Naceur Abdelkrim, "*Modèle d'Interaction entre Produits Actifs pour la Gestion de la Sécurité*",5th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, SETIT 2009, March 22-26, 2009 – Tunisia.

[Zuniga and Krishnamachari, 2004] M. Zuniga and B. Krishnamachari, "*Analyzing the transitional region in low power wireless links*", IEEE SECON: First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, pp. 517–526, 2004.

RÉSUMÉ

La surveillance d'entrepôts de produits chimiques est une opération délicate dans le sens où elle passe par une connaissance de la nature de chaque produit stocké, sur leur localisation, sur leur possible interaction et sur les actions à mettre en œuvre en cas d'alerte. Pour faciliter cette gestion de stockage, cette thèse propose d'utiliser le concept de l'ambiant où le produit possède son propre système d'information et de communication sans fil de façon à le rendre intelligent et autonome. Cette thèse propose et développe un modèle de comportement interne de produit actif permettant une approche distribuée de sécurité active. Celui-ci permet d'aboutir à un protocole de communication de niveau applicatif à embarquer dans les produits actifs. Ce protocole est évalué de façon formelle en utilisant les Réseaux de Petri colorés hiérarchiques. Finalement, ce protocole est implémenté dans le simulateur Castalia/Omnet++ pour l'analyser dans plusieurs scénarii et aussi pour l'éprouver lors du passage à l'échelle. Les résultats montrent l'intérêt et la faisabilité du concept de produit actif.

Mots clés: ***Produit actif, Réseaux de Petri, Sécurité active, Coopération, Réseaux de capteurs sans fil, Simulation.***

ABSTRACT

Monitoring of chemical product storage is a delicate operation in the sense that it requires knowledge of the nature of each stored product, their location, their interaction and possible actions to be implemented in case of emergency. To facilitate the storage management, this thesis proposes to use the concept of ambient where the product has its own information system and wireless communication so as to make it intelligent and autonomous. This thesis proposes and develops a model of internal behavior of active product that allows a distributed approach of active security. This can lead to a communication protocol of application level to embed the active products. This protocol is formally assessed using hierarchical colored Petri nets. Finally, this protocol is implemented in the simulator Castalia/Omnet++ to analyze it in several scenarios and also for the experience when going to scale. The results show the usefulness and feasibility of the concept of active product.

Key words: ***Active product, Petri Nets, Active Security, Cooperation, Wireless Sensor Network, Simulation.***