

Thèse de Doctorat de l'Université Pierre et Marie Curie PARIS VI

Spécialité

INFORMATIQUE

présentée par

Philippe DARCHE

pour l'obtention du titre de

DOCTEUR DE L'UNIVERSITE PARIS VI

Le Paradigme Acteur appliqué aux Systèmes Embarqués Communicants :

ActNet, un Réseau d'Acteurs Robotiques

Soutenue le 2 Mars 1994

Devant le jury composé de :

Claude Girault	Président
Hong Hoang Quang	Rapporteur
Martial Vivet	Rapporteur
Philippe Coiffet	Examineur
Pascal Estrailier	Examineur
Marie-France Le Roch	Examineur
Jean-Daniel Nicoud	Examineur

A la mémoire de mon Père

A ma Mère

A ma Famille

A Fawzia

A mes Amis

Il faudrait faire en sorte que tout soit aussi simple
que possible, mais pas plus simple.

Albert Einstein

(citation non référencée, lue dans [**Minsky 88**])

Remerciements

Par cette longue liste, j'aimerais remercier toutes les personnes qui sont intervenues dans mon projet, et qui m'ont surtout fait confiance pendant ces quatre années de travail. J'ai pu vivre une expérience passionnante et sans eux, je n'aurais sans doute pas abouti dans ce vaste projet. En espérant n'avoir oublié personne, je remercie donc :

- M. Claude Girault, Professeur à l'Université Paris VI, qui m'a fait l'honneur de diriger ma thèse et de présider ce jury,
- M^{me} Marie-France Le Roch, Maître de Conférences et examinateur, et M. Gérard Nowak, tous deux enseignants à l'Université Paris VI avec qui j'ai collaboré et qui ont suivi mon travail et mes écrits. Merci pour leur disponibilité et leur gentillesse,
- MM. Hong Hoang Quang et Martial Vivet, Professeurs respectivement à l'Université Paris V et à l'Université du Maine, pour avoir accepté d'être rapporteurs de ma thèse malgré leur charge de travail. Leurs remarques constructives ont contribué à améliorer la qualité de ce travail,
- MM. Philippe Coiffet, Pascal Estrailier et Jean-Daniel Nicoud, Professeurs respectivement à l'INSTN, à l'Université Paris VI et à l'école Polytechnique de Lausanne, pour avoir accepté d'être examinateurs de ma thèse,
- M. Michel Rozenberg, qui a réalisé la CAO du robot mobile. Sans ses compétences et son professionnalisme, la réalisation mécanique ne présenterait pas cette qualité,
- M. Louis Audoire, Ingénieur de Recherche à l'INRIA, pour ses compétences et son aide en électronique,
- M^{me} Jacqueline Narboni, qui m'a mis en relation avec le Lycée Polyvalent de Sens,
- M. Taraud, Chef des travaux et M. Kryzmanic, Professeur de mécanique, au lycée polyvalent régional de Sens, avec qui nous avons collaboré pour la fabrication mécanique du premier prototype de robot mobile,
- M^{me} Rault, Proviseur du lycée technique industriel René Cassin de Noisiel, MM. Boucard, Chenavier et Mathieu, chefs des travaux respectivement aux lycées de Meaux, Chelles et Noisiel pour leur collaboration lors de la fabrication des autres prototypes,
- M. José Martinez du Lycée technique d'Aubervilliers et ses collègues du département mécanique,
- MM. Pierre-Guillaume Raverdy, Jean-Michel Couvreur, Bertil Folliot, Karim Fougali et Emmanuel Paviot-Adet pour leur aide lors de la finalisation du document.
- M. le Professeur Auvray avec la collaboration de M. Patrick Noël de l'IST qui m'ont accueilli dans leurs locaux,
- M. Jean-Marc Raibaud du CNET pour m'avoir aidé lors de la conception des modems HF,
- M. Magueur pour sa participation active lors de la réalisation des prototypes mécaniques,
- M^{lles} Emmanuelle Encrenaz et Christiane Austermann pour leur aide spontanée de pâtissière,
- Pour finir, tous ceux qui m'ont côtoyé et surtout supporté (aux sens propre et figuré !) durant ces années au laboratoire, en particulier les ZoZios de la 206, François Bréant, Rémy Card, Hubert Le Van Gong et Pierre Sens. L'ambiance qui règne dans ce bureau est géniale !

Résumé

Notre étude concerne un réseau d'acteurs matériels et logiciels. La notion d'acteur est une approche récente du parallélisme empruntée au modèle des organisations sociales, qui permet de résoudre naturellement les problèmes classiques des architectures distribuées en combinant les notions d'objet, de message, et d'activité. Nous avons appliqué ce concept à la robotique avec un réseau d'acteurs robotiques mobiles et sédentaires communicants : le système **ActNet** (Actors Network ou réseau d'acteurs).

Nous proposons une réalisation de ce système en vue d'applications dans le domaine de l'enseignement sous la forme d'un outil pédagogique pour l'apprentissage des notions du parallélisme et de la communication. Ce système pourrait aussi servir comme plate-forme de développement pour la robotique mobile, appliquée particulièrement à l'Intelligence Artificielle Décentralisée, car nous offrons les couches matérielles et logicielles nécessaires pour supporter un développement système et/ou langage en Lisp, Smalltalk, ou Prolog. Une carte générique de contrôle à base du transputer et un robot mobile prototype "le Tatou" avec toutes ses cartes électroniques ont été développés.

Mots-clés : acteur, robotique mobile, robotique pédagogique, réseaux locaux radio-fréquences, transputer

Abstract

Our work concerns a hardware and software actors network. The notion of actor is a recent parallelism approach based on the model of social organisations, which allows one to naturally resolve classical problems of distributed architectures by combining the notions of object, message and activity. We have applied this concept to robotics with a network of communicating mobile and sedentary robotic actors : the **ActNet** network.

We propose an implementation of this system to be used as a pedagogical tool to learn the notions of parallelism and communication. This system could also be used as a framework for the development of mobile robotics, especially applied to Decentralized Artificial Intelligence, as we provide hardware and software layers necessary to support the development of systems and/or languages in Lisp, Smalltalk or Prolog. A transputer based generic control board and a mobile robot prototype, "Tatou", with all its electronic boards have been developed.

Mots-clés : actor, mobile robotics , pedagogical robotics, wireless local network, transputer

Nous avons désigné notre système **ActNet** pour "Actors Network". Le nom ActNet a déjà été utilisé dans un article de N.J. Nilsson intitulé "Actnet : an Action-Network language and its interpreter" pour désigner un langage basé sur le concept de réseaux d'actions. Il n'y a aucune ambiguïté possible dans la mesure où ce nom concerne un sujet et un domaine, l'Intelligence Artificielle, totalement différents de nos préoccupations.

TABLE DES MATIERES

Introduction	1
1. L'enseignement du parallélisme	1
2. La robotique mobile et réactive	3
3. ActNet	4
4. Organisation de la thèse	4
Chapitre I : ActNet	7
1. Introduction	9
2. Paradigme acteur.....	9
2.1. Modèle de base d'Hewitt	10
2.2. Modèle de base d'Agha.....	11
2.3. Langages d'acteurs	12
2.3.1. Plasma	12
2.3.2. Le modèle ABCL/1	12
2.3.3. Actalk	12
2.3.4. Mering IV.....	13
2.4. Caractéristiques générales	13
2.5. Implémentation Système	14
3. Modèle acteur matériel.....	14
3.1. Définition	14
3.2. Comportement	15
3.3. Spécificités de l'acteur matériel	17
3.4. Communication	18
3.5. Architecture matérielle.....	24
4. Réseau virtuel	24
4.1. Définition	24
4.2. ActNet comme un micro-monde.....	26
4.3. Groupe de communication	25
4.4. Topologie.....	30
5. Conclusion	32
Chapitre II : Architecture matérielle des acteurs robotiques	33
1. Introduction	35
2. Robotique mobile	35
2.1. Définition	36
2.2. Architecture de contrôle	38
2.3. Coopération	39
2.4. Exemples	40
3. Robot et robotique pédagogique.....	42
3.1. Définition de la robotique pédagogique	42
3.2. Définition d'un robot pédagogique.....	42
3.3. Etude de l'interface de commande	43
3.3.1. Classification des robots pédagogiques	43
3.3.2. Liaison interface-robot	44

2.1.4. Conclusion	95
2.2. Principe d'émission-réception.....	95
2.2.1. Emission.....	95
2.2.2. Réception.....	97
2.2.3. Antenne.....	99
2.2.4. Duplexeur	100
2.3. Synthèse digitale de fréquence.....	101
2.3.1. Boucle à verrouillage de phase.....	102
2.3.2. L'oscillateur contrôlé en fréquence.....	104
2.3.3. Principe de la synthèse de fréquence	105
2.3.4. Synthèse numérique directe.....	106
3. Etat de l'art des réseaux radio-fréquences	106
3.1. Les normes	108
3.2. Présentation de réseaux	110
3.2.1. Le réseau DECT	110
3.2.2. Le réseau cellulaire digital	110
3.2.3. La norme Tetra	111
3.2.4. Le réseau domotique CEBus.....	111
4. Sous-réseau d'acteurs mobiles	111
4.1. Spécificités d'ActNet	111
4.2. Interface d'entrée-sortie	114
4.3. Choix technologiques	117
4.4. Types de communication.....	118
4.4.1. Communication duplex à une antenne	118
4.4.2. Communication duplex à deux antennes.....	122
4.4.3. Communication simplex ou à l'alternat	125
5. Sous-réseau d'acteurs sédentaires	127
6. Passerelle	128
7. Conclusion	128
Chapitre IV : Architecture logicielle	131
1. Introduction	133
2. Etat de l'art des systèmes	133
2.1. Le temps réel	133
2.2. L'approche noyau monolithique	135
2.3. L'approche micro-noyau multi-serveurs	135
3. Architecture logicielle de l'acteur matériel	136
3.1. Couche entrée/sortie.....	138
3.1.1. Sous-couche matériel.....	138
3.1.2. Sous-couche réflexe.....	138
3.2. Couche Système	138
3.2.1. Sous-couche interruption	138
3.2.2. Sous-couche micro-noyau	139
3.2.3. Sous-couche serveur	139
3.3. Communication.....	140
3.3.1. Réseau	140
3.3.2. ATM	141
3.3.3. Graphe de liaison	143
3.4. Couche application : Méta-Vlisp.....	145
3.4.1. Caractéristiques.....	147
3.4.2. Implémentation de Méta-Vlisp sur Transputer	148

4. Conclusion	148
Conclusions et Perspectives	151
Evolutions futures	154
Annexes : Schémas électroniques et Mécaniques	157
Références bibliographiques	225
Glossaire	237

TABLE DES TABLEAUX

Tableau II-1 : Les batteries	70
Tableau II-2 : Correspondance grandeur physique / capteur	82
Tableau II-3 : Correspondance grandeur physique / transducteur	82
Tableau III-1 : Classes courantes d'amplificateurs radiofréquences	96
Tableau III-2 : Type de communication en radiofréquences	108
Tableau III-3 : Caractéristiques des principales normes de radio-transmission de données	109
Tableau III-4 : Caractéristiques des principales normes de radio-transmission de données (suite)	109

TABLE DES FIGURES

Figure I-1 : Comportement interne d'un acteur X	11
Figure I-2 : Graphe d'état d'un acteur matériel	16
Figure I-3 : Niveaux d'interruption	18
Figure I-4 : Communication de type passé	19
Figure I-5 : Communication de type présent	20
Figure I-6 : Communication de type futur	21
Figure I-7 : Organisation interne d'un acteur matériel	23
Figure I-8 : Les entités d'ActNet.	25
Figure I-9 : Les groupes de communication	27
Figure I-10 : Topologie logique d'ActNet.	28
Figure I-11 : Topologie physique d'ActNet	29
Figure I-12 : Topologie générale d'ActNet	30
Figure I-13 : Topologie détaillée d'ActNet	31
Figure II-1 : Chaîne de contrôle d'un robot pédagogique	44
Figure II-2 : Spécialisation de la carte générique par type d'acteur	48
Figure II-3 : Deuxième proposition d'architecture	50
Figure II-4 : Gestion des interruptions	52
Figure II-5 : Groupe contrôle	57
Figure II-6 : Groupe entrée-sortie	58
Figure II-7 : Module gestion du temps	59
Figure II-8 : Module interface série	59
Figure II-9 : Module interface parallèle	59
Figure II-10 : Groupe propulsion	61
Figure II-11 : Groupe capteur	62
Figure II-12 : Groupe impression	65
Figure II-13 : Asservissement	66
Figure II-14 : Groupe alimentation	68
Figure II-15 : Gestion de l'énergie	74
Figure II-16 : Le Model_Unit	76
Figure II-17 : Organisation interne d'un acteur robotique	77
Figure II-18 : Modélisation de l'acteur robotique mobile "tatou"	78
Figure II-19 : Chaîne de contrôle d'un robot pédagogique	83
Figure III-1 : Modélisation de la communication	89
Figure III-2 : La modulation	92
Figure III-3 : Synoptique de la chaîne émission	95
Figure III-4 : Synoptique de la chaîne réception version changement de fréquence	97
Figure III-5 : Synoptique de la chaîne réception version conversion directe	99
Figure III-6 : Synoptique d'une communication radio-fréquence	99
Figure III-7 : Duplexeur d'antenne	101
Figure III-8 : Schéma de base d'une PLL	102
Figure III-9 : Schéma de base d'une PLL à diviseur	104
Figure III-10 : Canal de secours (1 ^{er} cas)	115
Figure III-11 : Canal de secours (2 ^{ème} cas)	116
Figure III-12 : Communication en duplex à une antenne	119

Figure III-13 : Diffusion en duplex à une antenne.....	119
Figure III-14 : Bus en duplex à une antenne	120
Figure III-15 : Etoile en duplex à une antenne	121
Figure III-16 : Anneau en duplex à une antenne	121
Figure III-17 : Arborescence binaire en duplex à une antenne	122
Figure III-18 : Anneau en duplex à deux antennes	122
Figure III-19 : Bus en duplex à deux antennes	123
Figure III-20 : Etoile en duplex à deux antennes	123
Figure III-21 : Anneau en duplex à deux antennes	124
Figure III-22 : Arbre binaire en duplex à deux antennes	124
Figure III-23 : Anneau en alternat	125
Figure III-24 : Bus en simplex	125
Figure III-25 : Etoile en alternat.....	126
Figure III-26 : Anneau en alternat	126
Figure III-27 : Arborescence en alternat	127
Figure IV-1 : Architecture logicielle d'ActNet	137
Figure IV-2 : Les couches communication	142
Figure IV-3 : Graphe des capacités de liaisons physiques.....	144
Figure IV-4 : Graphe des liaisons physiques	144
Figure IV-5 : Graphe des capacités de liaisons logiques.....	145
Figure IV-6 : Graphe des liaisons logiques	145
Figure IV-7 : Couche application	146

TABLE DES PHOTOGRAPHIES

Photographie II-1 : Carte générique prototype	49
Photographie II-2 : Robot mobile - vue de face	53
Photographie II-3 : Robot mobile - vue de côté.....	54
Photographie II-4 : Carte générique nue	57
Photographie II-5 : Carte contrôleur moteur pas à pas	60
Photographie II-6 : Carte capteur de collision	63
Photographie II-7 : Capteur de collision en situation.....	64
Photographie II-8 : Groupe impression.....	65
Photographie II-9 : Carte contrôleur stylo	66
Photographie II-10 : Carte détection de fins de course.....	67
Photographie II-11 : Carte sélecteur d'alimentation	72
Photographie II-12 : Carte chargeur en situation	73
Photographie II-13 : Carte convertisseur DC/DC en situation	75
Photographie II-14 : Le robot sédentaire "Youpi"	80
Photographie II-15 : Carte d'adaptation robot-E/S	81
Photographie III-1 : Récepteur	117

Chapitre I

ActNet

- 1. Introduction**
- 2. Paradigme acteur**
- 3. Modèle acteur matériel**
- 4. Réseau virtuel**
- 5. Conclusion**

1. Introduction

Après l'approche fonctionnelle, une nouvelle tendance est apparue où code et données étaient encapsulés dans la même entité : l'approche objet. Nous définissons dans cette partie le concept d'objet. Pour passer à la notion d'acteur, il nous faut parler d'activité et d'objet actif.

Un objet est une entité informatique passive activée par un envoi de message. Il est constitué d'une partie attributs qui est la partie variable interne et d'un ensemble de fonctions et procédures applicables aux attributs appelées méthodes. L'objet peut être vu comme une entité passive car seul un message peut activer un objet. La communication est donc implicite.

Pour ajouter de l'activité, il faut passer à la notion d'objet actif ou d'acteur. Les modèles d'objets concurrents proposent de conserver le principe de **réactivité** des objets séquentiels où l'objet ne peut être activé que par transmission de messages. D'autres modèles proposent la notion d'**activité** autonome complète où l'objet possède une activité autonome dès sa création (analogue à un processus) et indépendamment des requêtes qui lui sont adressées. L'objet peut contrôler directement sa propre exécution, ainsi que les messages qu'il décide de traiter. La communication est explicitée. C'est dans cette catégorie que se trouve l'acteur.

Nous présentons le modèle de base d'Hewitt [**Hewitt et al. 73**] et le modèle de calcul d'Agha [**Agha 86**], ainsi que les principaux langages développés à partir de ces deux modèles.

Un acteur logiciel peut avoir à gérer des périphériques matériels. Pour notre système, il s'agit de capteurs et d'actionneurs équipant des robots mobiles et sédentaires. Le modèle acteur logiciel doit être étendu. Nous introduisons donc la notion d'acteur matériel. Pour rester cohérent avec le modèle logiciel, un acteur matériel communique avec ses congénères par messages via un réseau local. L'ensemble du système se nomme **ActNet** pour **Actors Network**.

Nous détaillons les acteurs d'ActNet et introduisons la notion de groupe de communication. Les acteurs peuvent appartenir à deux sous-familles en fonction de leur mobilité : les acteurs matériels mobiles et les acteurs matériels sédentaires. A chaque type d'acteur correspondra un sous-réseau : le sous-réseau d'acteurs mobiles et le sous-réseau d'acteurs sédentaires dont nous détaillerons la topologie. Un sous-réseau utilisateur permet de développer et de contrôler des applications.

2. Paradigme acteur

A l'origine, le paradigme acteur a été défini dans [**Hewitt et al. 73**]. Ce modèle est inspiré des modèles des organisations sociales. Après avoir présenté ce modèle et celui de

Gul Agha [**Agha 86**], nous étudions les langages les plus représentatifs à base de ce modèle.

2.1. Modèle de base d'Hewitt

Un acteur est une **entité active autonome et coopérante** ou expert qui communique avec ses congénères et lui-même par messages [**Hewitt 77**]. Rappelons que la coopération est l'ensemble des actions permettant de réaliser un but commun. Ce modèle respecte le principe de l'encapsulation du paradigme objet mais la notion d'activité y est ajoutée [**Masini et al. 89**]. Un acteur n'a qu'une connaissance locale et limitée de son environnement : une liste des autres acteurs (**accointances**) à qui il peut envoyer des messages et un comportement à calculer en fonction des messages reçus. Ce comportement est exprimé sous la forme d'un **script**. Ce script est un ensemble de listes ayant le format (<nom_commande> <actions>), ce nom de commande est ensuite interprété sous la forme d'actions à exécuter.

La notion de séquentialité n'existe pas; on ne présuppose rien sur l'ordre d'envoi des messages, ni sur l'ordre de création des acteurs. Le modèle 'acteur' est un modèle local : il n'y a pas d'actions à distance et l'utilisateur ne peut pas connaître, à un instant donné, l'état global du système.

L'acteur est connu de l'extérieur par l'adresse d'une boîte aux lettres; à celle-ci est associée une file d'attente où les messages sont stockés jusqu'à leur évaluation. Lors de la prise en compte d'un message de la file, l'acteur a le comportement suivant :

- envoyer d'éventuels messages à d'autres acteurs selon ses accointances ou à lui-même (self-adressage),
- créer d'autres acteurs,
- évaluer les expressions de la méthode invoquée.

La réponse à l'évaluation d'un message n'est pas forcément renvoyée à l'émetteur de ce message, mais à un récepteur tiers. Ce mécanisme est appelé la **continuation**. Pour schématiser le fonctionnement de l'acteur, l'unité de traitement de l'acteur reçoit un flot d'entrée de messages et émet, si nécessaire, un flot de messages en sortie. Pour garder la cohérence du modèle, tout message arrive au destinataire (obligation de réception); par contre il n'y a pas obligation de traitement.

La durée de vie d'un acteur logiciel est infinie et le nombre d'acteurs peut être infini (en pratique limité par les capacités du système : taille mémoire, etc.).

Chaque acteur a sa propre horloge appelée horloge locale et aucune synchronisation entre horloges locales n'est réalisée.

2.2. Modèle de base d'Agha

Le modèle d'Agha peut être considéré comme un modèle de calcul concurrent et distribué à approche 'objet' [Agha 86].

La communication est asynchrone par envoi de messages. Les avantages sont principalement :

- ce type de communication permet à un acteur de s'envoyer des messages sans phénomène d'interblocage,
- l'envoyeur n'est pas bloqué en attendant que le message soit traité par le receveur.

La notion de séquentialité n'existe pas : on ne présuppose rien sur l'ordre temporel d'envoi par plusieurs acteurs, et sur le respect de celui-ci lors de l'évaluation. L'ordre d'arrivée des communications envoyées est l'ordre séquentiel. Une communication synchrone peut être ramenée à une communication asynchrone à capacité de bufferisation nulle.

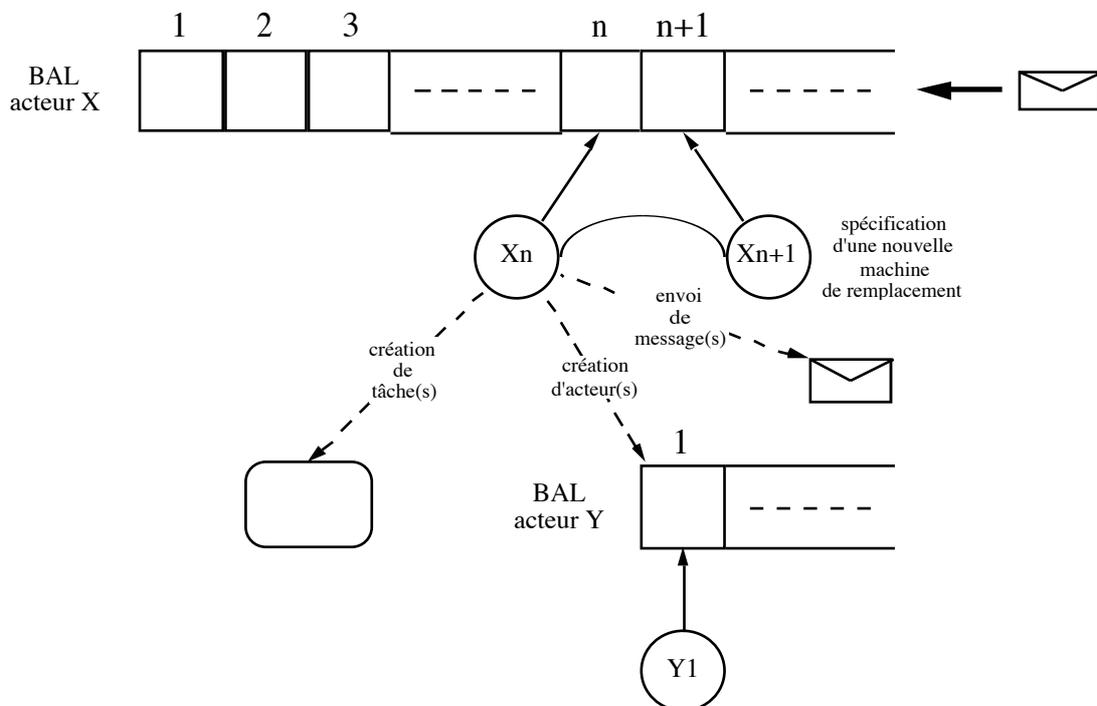


Figure I-1 : Comportement interne d'un acteur X

Un acteur peut être décrit en spécifiant son adresse de boîte aux lettres, et son comportement, qui est une fonction des messages reçus. Un événement est l'évaluation d'un message par l'acteur. A chaque évaluation d'un message n , l'acteur X_n associe un quadruplet

composé des éléments suivants (figure I-1 inspirée de [Agha 86]) :

- un ensemble fini de messages envoyés à d'autres acteurs ou à lui-même,
- un nouveau comportement associé au prochain message pris en compte, encapsulé dans un nouvel acteur X_{n+1} ,
- un ensemble fini d'acteurs créés,
- et un ensemble de tâches à exécuter.

On ne présuppose rien sur l'ordre de création des acteurs. Le changement d'état est ici remplacé par un changement de comportement, comportement qui spécifie les actions à entreprendre en fonction des messages reçus, et ces comportements sont susceptibles d'être exécutés simultanément. Les structures de contrôle itération et synchronisation dans ce modèle sont donc le passage de messages.

2.3. Langages d'acteurs

A partir des travaux fondateurs d'Hewitt, puis d'Agha, plusieurs langages d'acteurs ont été développés. Il en existe de nombreux, nous en retiendrons principalement quatre : PLASMA [Hewitt 77], ABCL/1 [Yonezawa 90], Actalk [Briot 89] et MERING IV [Ferber 89b] [Carle 92]. Nous allons présenter maintenant les caractéristiques principales de chacun de ces quatre langages.

2.3.1. Plasma

Plasma (PLAnner-like Modeled on Actors) [Smith and Hewitt 75] est le premier langage acteur issu des travaux d'Hewitt. Un script est un programme PLASMA qui définit le comportement de l'acteur, c'est-à-dire les actions qu'il doit exécuter à la réception d'un message. Son successeur a été la série des langages ActX qui ont intégré le parallélisme.

2.3.2. Le modèle ABCL/1

Dans le langage ABCL/1 [Yonezawa et al. 90] qui repose sur CommonLisp, l'acteur est dit sérialisé car l'évaluation de chaque message est faite de manière séquentielle. Ceci peut être vu comme un moniteur [Hoare 87]. Deux types de messages sont possibles : ordinaire ou express, avec quatre types de transmission : 'past', 'now', 'future' et 'call'. Le type 'past' est une transmission asynchrone, le message est envoyé sans attente de réponse de l'acteur récepteur, ni de résultat. Le type 'now' est une transmission synchrone, l'acteur émetteur attend explicitement une réponse. Le type 'future' permet à l'acteur de ne pas rester en attente d'une réponse immédiate, cette attente étant reportée ultérieurement. Le dernier type a été créé pour unifier les procédures, les fonctions et les objets. L'acteur est bloqué, en attente du résultat de l'exécution de l'objet procédure ou fonction demandée.

2.3.3. Actalk

Actalk [Briot 89] est une plate-forme de test pour réaliser et étudier des langages de type acteur. C'est un système acteur minimal, développé dans l'environnement Smalltalk-80.

Originellement, le seul type de messages était de type ‘passé’. Mais grâce à la modularité de la plate-forme, les types ‘futur’ et ‘présent’ ont été implémentés sous la forme d’une nouvelle classe. Actalk a permis de simuler le comportement du modèle d’Agha ou d’autres comportements comme celui du langage ABCL/1.

2.3.4. Mering IV

Ce langage reposant sur Lisp fait la distinction entre objet et acteur. Un acteur dans Mering IV [Ferber and Carle 90] [Carle 92] peut avoir trois états : passif, attente et actif. Un seul acteur est actif à la fois. Un seul type de message existe : le type ‘past’. Un acteur n’est jamais bloqué par la réponse à un calcul. Il précise que la réponse du calcul doit être envoyée à un autre acteur : le *customer*.

2.4. Caractéristiques générales

A partir des modèles pré-cités, la famille des acteurs se subdivise en deux sous-familles : les acteurs **sériels** et les acteurs **non sériels**. Un acteur sériel exécute un seul message à la fois et spécifie explicitement son nouvel état afin de traiter le message suivant, alors que l’acteur non sériel a la possibilité de traiter plusieurs messages à la fois.

L’acteur selon Agha peut être considéré comme non sériel, car ses comportements sont exécutés en parallèle. Pour le modèle d’Hewitt, l’acteur est de type sériel, c’est-à-dire que les messages sont interprétés séquentiellement.

De la présentation des deux principaux modèles et des quatre langages, nous pouvons maintenant dégager les caractéristiques générales du modèle acteur :

- Parallélisme implicite : le modèle acteur permet d’exprimer “naturellement” des activités concurrentes. Si un calcul doit s’exécuter séquentiellement, la séquentialité s’obtiendra par les acteurs eux-mêmes.
- Dynamisme et reconfigurabilité : le modèle permet de créer d’autres acteurs.
- Du point de vue de la communication :
 - Asynchronisme : les acteurs communiquent de manière asynchrone par envoi de messages. C’est la seule interaction possible entre acteurs.
 - Bufferisation : les messages reçus sont rangés dans une boîte aux lettres.
 - Garantie de livraison.
- Continuation : la suite d’un calcul peut être exécutée par un autre acteur.
- Comportement et script : un script décrit un comportement qui est exécuté à la réception d’un message.
- Localité : l’acteur n’a qu’une vue locale de son environnement par sa liste d’acointances. Il ne peut donc pas envoyer un message à un acteur qui n’est pas dans sa liste.
- Non déterminisme pour le modèle Agha, si l’ordre de traitement des messages n’est pas le même, il peut y avoir différentes exécutions.
- Modèle permanent : un acteur a une vie infinie.

2.5. Implémentation Système

Une mise en oeuvre du concept acteur a été réalisé au niveau système. Il s'agit du **CAC** ou **Communicating Active Component [Courtrai et al. 92]**. Le CAC est une entité destinée à la construction d'applications concurrentes sur architectures distribuées. Tout comme un acteur, il se compose d'une activité ou processus, d'un environnement local, et d'une boîte aux lettres. Ces CACs sont rassemblés en modules. Un module représente le code partageable pour un sous-ensemble de CACs, il est l'entité de distribution du code et des objets à l'exécution. La différence entre un acteur et un CAC est qu'un acteur peut utiliser plusieurs scripts durant sa vie, un CAC n'a qu'un comportement unique.

3. Modèle acteur matériel

Dans la section précédente, nous avons vu que la notion d'acteur était inspirée du modèle des experts communicants des modèles des organisations sociales [Hewitt 77]. Le modèle acteur peut être considéré comme un modèle de calcul distribué et coopérant avec une approche objet [Agha 86]. Dans le contexte du micro-monde, les robots peuvent être vus comme des acteurs matériels réels. La dimension matérielle du projet nécessite donc une extension du modèle acteur, l'acteur matériel, qui adaptera le modèle aux contraintes du matériel, en particulier robotique.

Ce nouveau modèle doit être cohérent avec le modèle acteur logiciel, pour que la structure de contrôle soit toujours le passage de messages. Or de nouveaux mécanismes comme les interruptions doivent être pris en compte. Nous proposons l'encapsulation des interruptions sous la forme d'un message. Pour respecter les contraintes temporelles d'exécution, nous proposons un traitement différent en fonction du type de message.

3.1. Définition

Nous avons développé le concept d'acteur matériel. A l'origine, le modèle a été défini dans [Darche 90-91]. Un acteur matériel est constitué d'une carte générique de commande, d'une interface réseau et éventuellement d'interfaces spécifiques comme un robot pédagogique. Ce sont les interfaces qui spécifient le type de l'acteur : dans le cas d'une interface robotique, nous parlerons d'**acteur robotique**. Nous avons voulu que notre modèle s'adapte aux principaux modèles d'acteurs logiciels, c'est pour cela qu'il présente des caractéristiques modulaires qui reprennent celles des modèles les plus répandus et détaillés dans la première partie de ce chapitre.

Un acteur matériel est caractérisé par :

- son identité,
- son état, c'est-à-dire l'ensemble des variables internes et l'état interne dans lequel il se trouve,
- son comportement, c'est-à-dire l'ensemble des actions possibles en fonction des messages qu'il reçoit.

Un acteur matériel ne peut créer réellement d'autres acteurs matériels, comme peuvent le faire les acteurs logiciels. Nous considérons que nous disposons d'un ensemble fini de robots, et que l'acteur désirant créer un congénère, pourra en activer un parmi l'ensemble (non vide) des acteurs libres. L'activation d'un acteur matériel est donc équivalent à la création d'un acteur logiciel.

Sa durée de vie est finie car un défaut d'alimentation volontaire ou involontaire fait mourir celui-ci. De plus, un acteur peut se désactiver, ou se faire désactiver. L'acteur interagit avec son environnement par l'émission et la réception de messages, pour garder la cohérence du modèle standard. Un système à communication par messages renforce la modularité et est donc adapté à la distribution.

Chaque acteur matériel possède une horloge locale et aucune synchronisation n'existe entre horloges locales. La mise en place d'une horloge logique est indispensable car les messages doivent être exécutés selon l'ordre d'émission. Ceci se retrouve en robotique où l'ordre d'activation des tâches est important.

3.2. Comportement

Certains périphériques doivent être gérés en section critique. Le modèle sériel traite les messages de manière séquentielle et s'adapte donc bien à la contrainte de gestion atomique de certaines tâches. C'est pour cette raison que nous l'avons retenu. Lors de l'évaluation d'un message et en fonction de son script, l'acteur matériel a le comportement suivant :

- il envoie d'éventuels messages à d'autres acteurs, à lui-même, ou aux périphériques de sortie ou d'entrée-sortie,
- il crée un nombre fini d'acteurs logiciels et active un nombre fini d'acteurs matériels,
- il évalue les expressions de la méthode invoquée.

La nature de l'information est donc soit un message provenant d'acteurs, soit une information provenant des périphériques d'entrée, soit une information destinée aux périphériques de sortie, ou soit un compte-rendu d'entrée-sortie.

Les périphériques d'entrée sont principalement des capteurs. Les informations issues de ceux-ci nous renseignent, soit sur l'état interne de l'acteur, (la tension batterie pour un mobile par exemple), soit sur le monde extérieur (la collision avec un autre acteur ou le décor par exemple). A ce niveau, l'information est de nature analogique ou logique.

L'acteur possède cinq états (figure I-2) :

- Mort : l'acteur n'est pas alimenté.
- Initialisation : l'acteur initialise son unité centrale et ses périphériques.
- Attente passive : les files d'attente sont vides et l'acteur est dormant.
- Actif : l'acteur évalue un message.
- Attente active : l'acteur est en attente explicite d'un message.

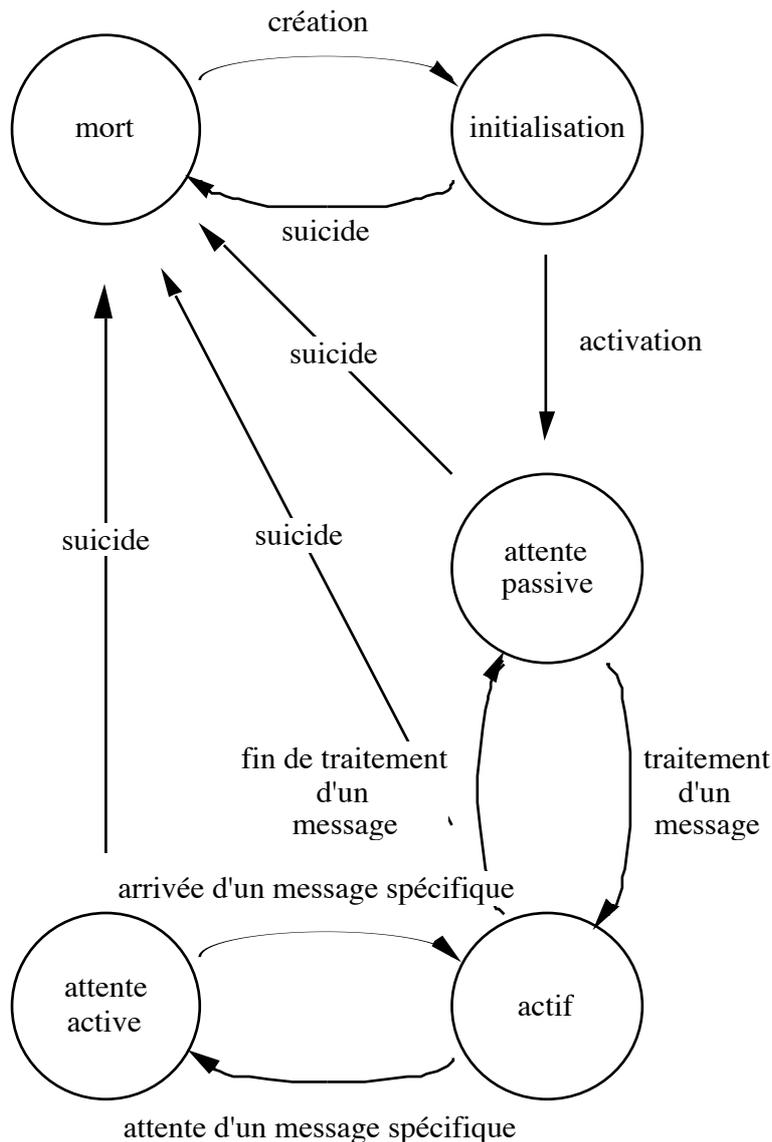


Figure I-2 : Graphe d'état d'un acteur matériel

Lorsque l'acteur matériel n'est pas alimenté, il est considéré comme *mort*. A sa mise sous tension, il passe à l'état *initialisation*. Cette phase permet d'initialiser l'unité centrale et les périphériques. Après un temps t_0 , l'acteur passe à l'état *attente passive* où il devient prêt à recevoir un message et à le traiter. La réception d'un message fait passer l'acteur de l'état *attente passive* à l'état *actif*. Après l'évaluation de celui-ci, il repasse à l'état *attente passive*. Il peut se mettre en attente d'un message particulier provenant d'un acteur défini ou d'un périphérique particulier (état *attente active*). A sa réception, il repasse à l'état *actif* pour pouvoir l'évaluer. A tout instant, l'acteur peut se suicider, c'est-à-dire peut agir sur son alimentation en lui envoyant un signal de désactivation ou être tué.

3.3. Spécificités de l'acteur matériel

L'acteur matériel gère ses périphériques suivant des événements. Généralement, les événements dans les systèmes distribués sont classés dans trois catégories :

- les événements envoyés,
- les événements reçus,
- et les événements internes.

Nous avons ramené cette classification à deux catégories : les événements internes, et les événements externes. L'événement étant encapsulé dans un message, les événements sont envoyés ou reçus.

Certains événements doivent être gérés avec des contraintes temporelles. Tout système temps réel pose le problème du temps de réponse du système, en particulier pour la sécurité de l'utilisateur (arrêt d'urgence). Le mécanisme d'interruption semble être le plus adapté à cette contrainte. L'interruption étant encapsulée dans un message, il faut absolument l'acheminer dans les plus brefs délais car elle doit être traitée très rapidement (contrainte de vitesse d'exécution et de priorité). Il est nécessaire donc d'utiliser un nouveau type de message : le télégramme, type le plus prioritaire. Les temps de communication étant incompatibles avec la rapidité de traitement, il nous faut délocaliser le traitement. Nous proposons de traiter certains événements rapides au niveau matériel, par exemple les événements relevant de la sécurité, et d'informer par message les niveaux supérieurs.

Nous avons défini trois niveaux de traitement (figure I-3) :

- le niveau 1 dit niveau matériel : l'interruption agit directement à ce niveau; un exemple : un capteur de fin de course coupe l'alimentation d'un moteur d'entraînement. L'information remonte ensuite au niveau 2.
- le niveau 2 dit niveau entrée-sortie : l'interruption est prise en compte et encapsulée sous la forme d'un message. A réception, un comportement spécifique est calculé. Si nécessaire, l'information remonte au dernier niveau.
- le niveau 3 dit niveau acteur : l'acteur reçoit sous la forme d'un télégramme l'information interruption et en fonction de la loi de priorité, un comportement est calculé.

Les interruptions logicielles sont générées par les acteurs. Une file d'attente dite file interruption, qui traite uniquement les télégrammes, est rajoutée à notre modèle, ainsi que deux politiques de gestion, spécifiques à cette file, pour traiter les interruptions : priorités tournantes et fixes. Cette file est donc hautement prioritaire.

Le modèle sériel semble bien adapté à la robotique puisqu'il se comporte comme un moniteur [Hoare 87]. En effet, un tel mécanisme évite, par exclusion mutuelle, des problèmes d'incohérence. Par exemple, si un acteur gère une ressource en section critique, le traitement en parallèle de messages pourrait violer l'accès unique. Ce problème se pose en particulier au niveau matériel pour la gestion d'un périphérique, par exemple un moteur. Un

message donnant une consigne de vitesse pendant un temps t_1 pourrait être contredit par un autre message traité en parallèle donnant une consigne différente pendant un temps t_2 .

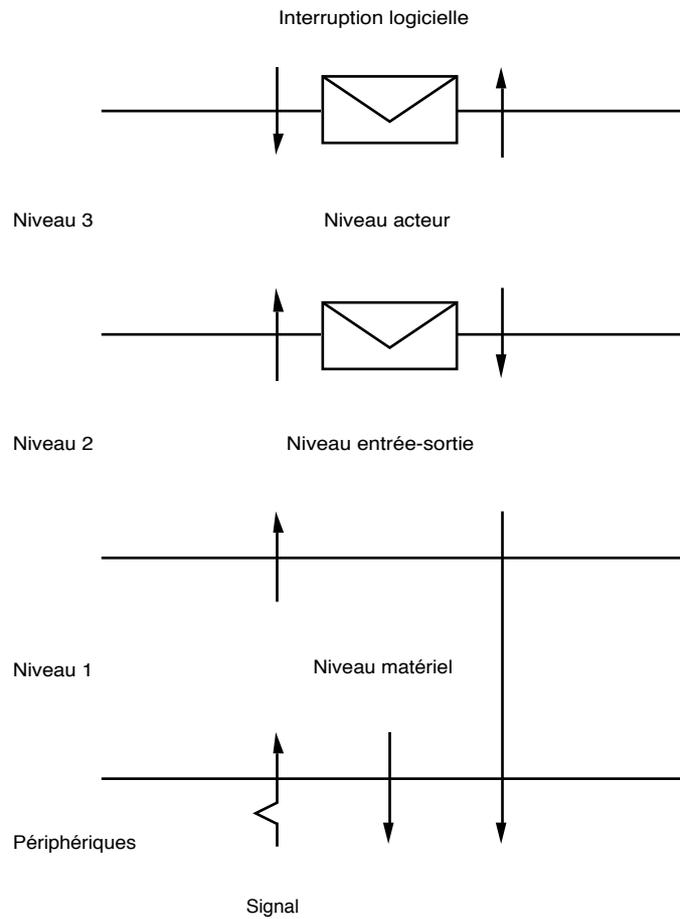


Figure I-3 : Niveaux d'interruption

3.4. Communication

La communication est la mise en relation d'une ou plusieurs entités via un ou plusieurs canaux. Pour deux entités en communication, l'une sera nommée émetteur et l'autre récepteur. Il existe d'autres types de communicants : l'écouteur autorisé ou pirate et l'émetteur anonyme. S'il existe plusieurs entités utilisant un canal, un conflit d'accès au canal peut avoir lieu lorsque plusieurs émetteurs se mettent à communiquer à un seul récepteur. La communication peut être point à point ou multi-point. Dans le dernier cas, il est possible alors de faire de la diffusion.

L'acteur matériel peut communiquer avec trois catégories de messages : la lettre, le paquet et le télégramme. Le traitement peut être urgent, normal (non urgent) pour la lettre ou le paquet. L'acteur possède au niveau réception trois files d'attente à politique variable : la file

application pour les lettres et paquets normaux, la file système pour les lettres et paquets urgents et la file interruption pour les télégrammes.

Plusieurs schémas de communication sont mis en oeuvre, caractérisés par un des trois types de messages qui sont :

- le message sans attente de réponse du ou des receveurs (type passé) (figure I-4),
- le message avec attente présente de réponse du ou des receveurs (type présent) (figure I-5),
- le message avec attente future de réponse du ou des receveurs (type futur) (figure I-6).

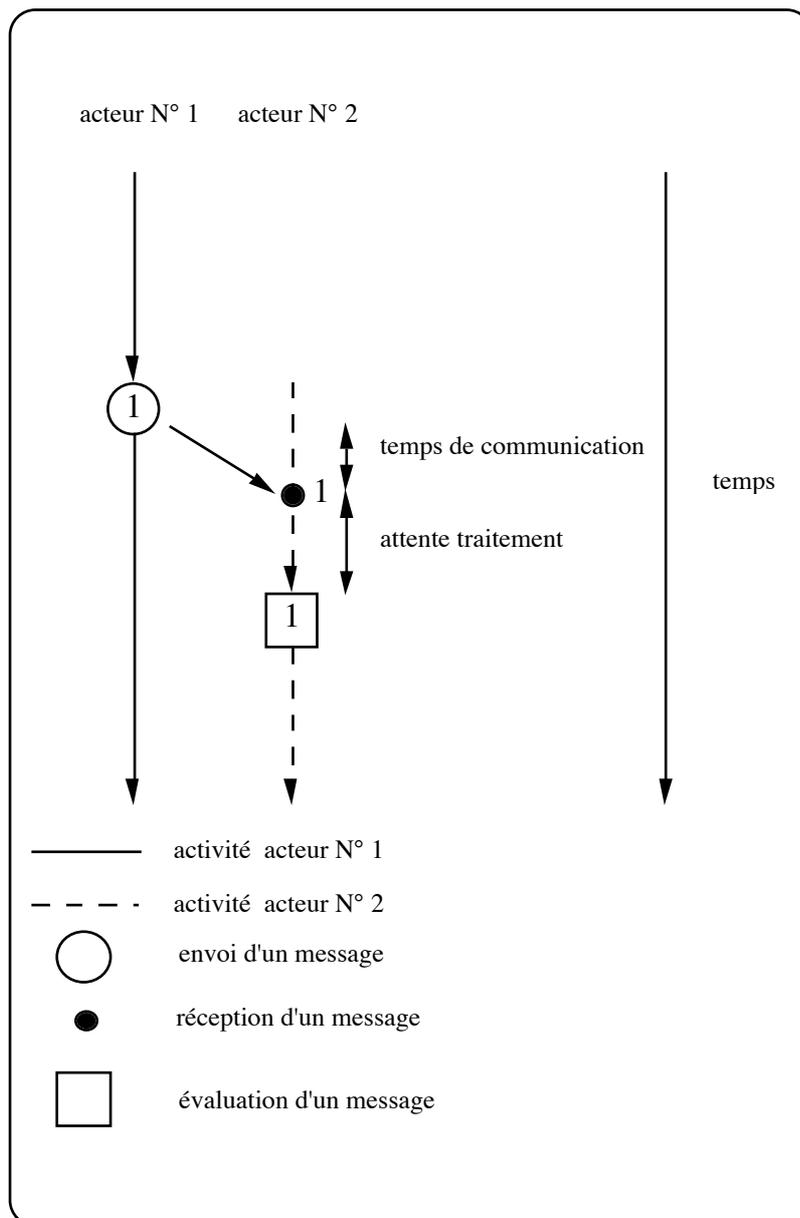


Figure I-4 : Communication de type passé

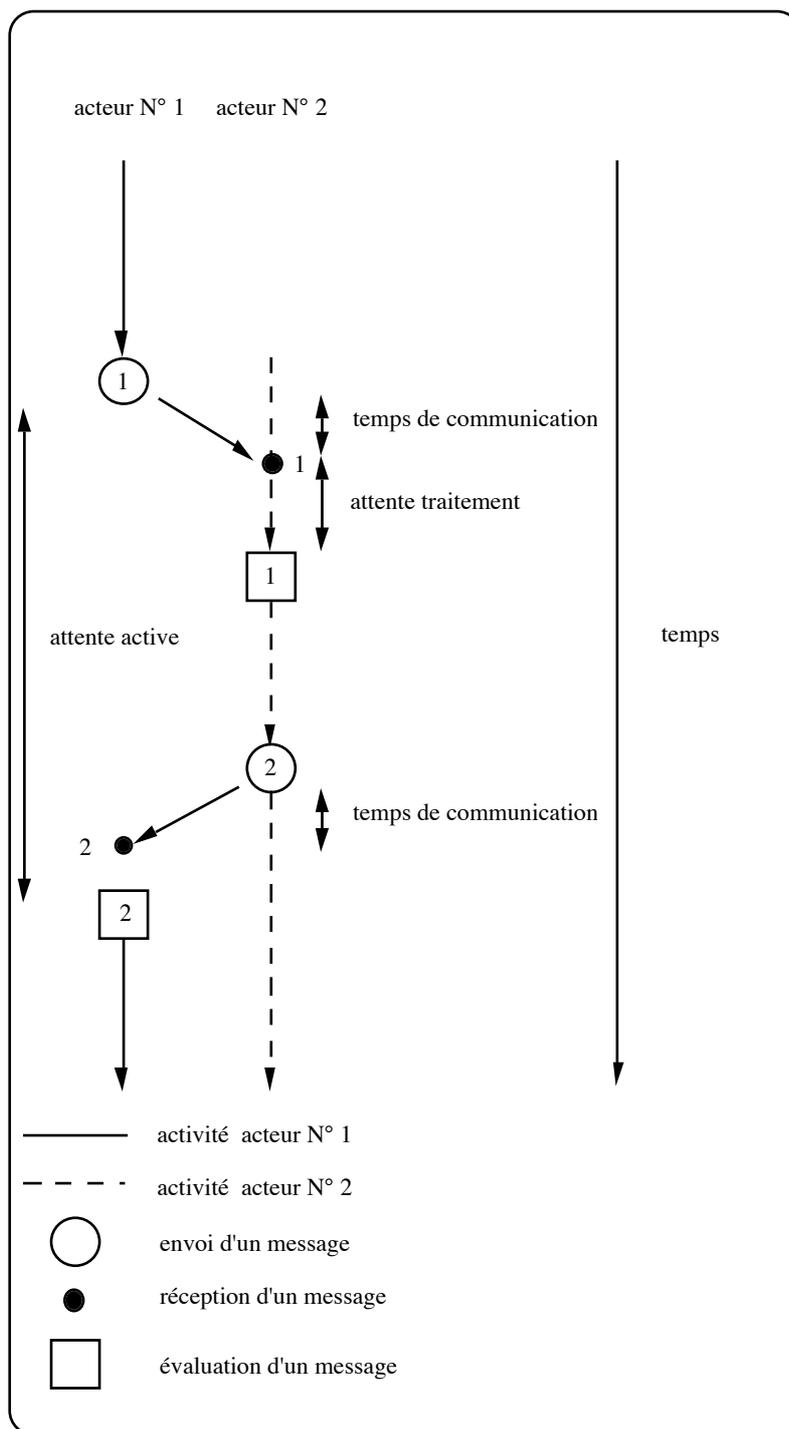


Figure I-5 : Communication de type présent

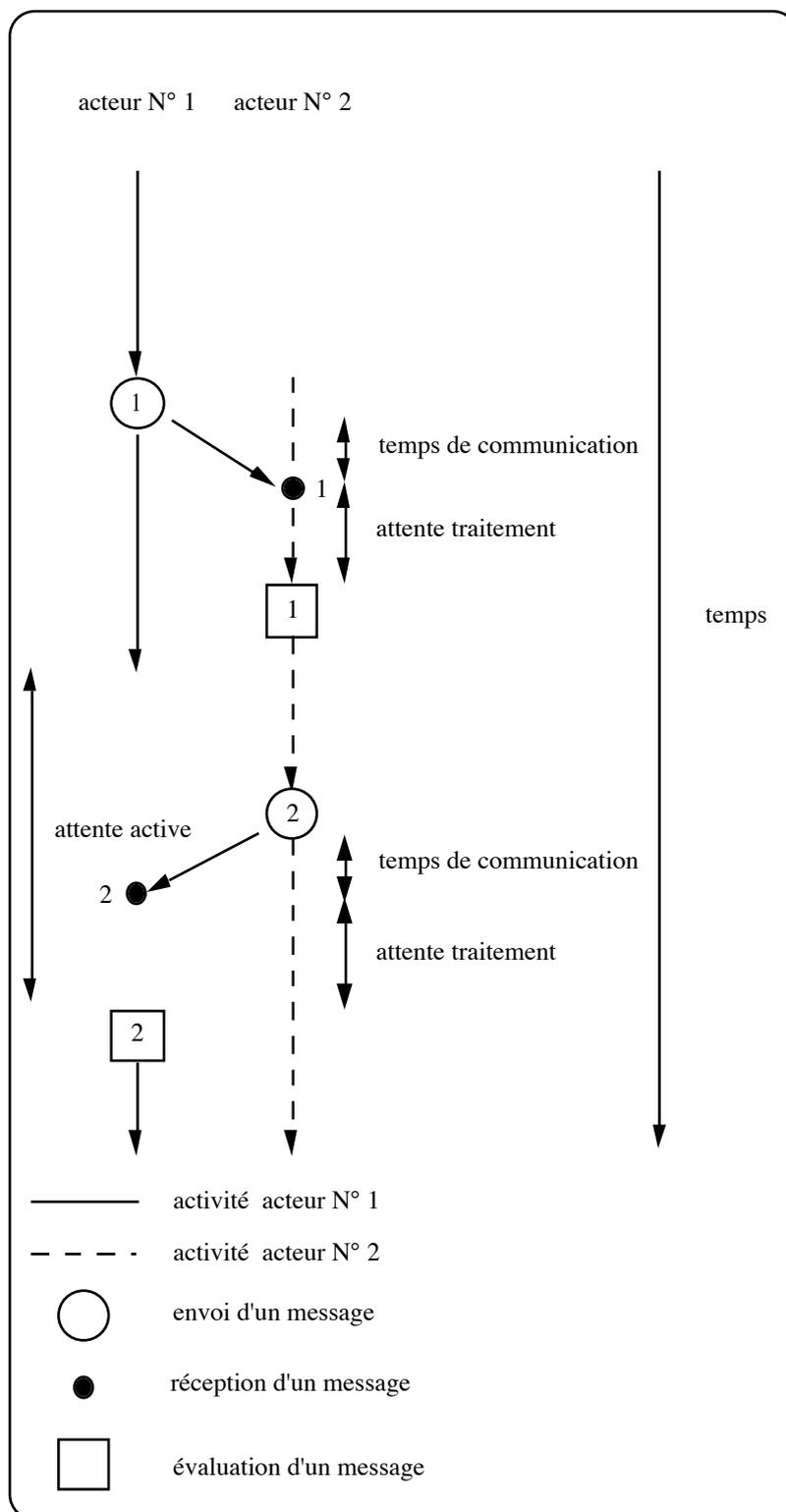


Figure I-6 : Communication de type futur

Pour le premier type de messages, la communication est dite non-bloquante ou asynchrone. Dans les autres types de messages, la communication est dite synchrone. Pour caractériser un calcul synchrone, nous reprenons la définition donnée dans **[Bougé 87]** : “Un système possède des communications synchrones si aucun des messages d’un type donné ne peut être envoyé avant que le receveur soit prêt à recevoir (c’est-à-dire, dans un état où la prochaine action serait une réception d’un message de ce type. Pour un observateur extérieur, la transmission peut ainsi être instantanée et atomique. Envoyer ou recevoir un message correspond en fait au même événement”.

La communication peut donc être asynchrone, avec ou sans accusé-réception, ou synchrone. Il existe trois files d'attente, une par type de messages. Les politiques de gestion de file sont au nombre de quatre : FIFO (First In First Out), LIFO (Last In First Out), random, ordre temporel d'émission des messages par catégorie de message; à celles-ci, il faut rajouter deux autres politiques : priorités fixe et tournante, pour la file interruption. Elles sont choisies par l'utilisateur en fonction du modèle acteur implémenté et de l'application.

Un gestionnaire de priorité sélectionne les messages à traiter. La priorité entre files par ordre croissant est la suivante : file application, file système et file interruption (figure I-7). L'acteur a une gestion préemptive des messages. Il y a garantie de réception et de maintien de l'ordre d'émission de plusieurs messages par un acteur donné.

En ce qui concerne la politique de réception, un message urgent interrompt le traitement d'un envoi non urgent. Un télégramme interrompt le traitement des messages urgents et non urgents ou, selon la politique de gestion adoptée par l'utilisateur, un télégramme de priorité inférieure.

Pour la politique en émission, une demande d'émission de type urgent interrompt le traitement d'un message non urgent. Un télégramme interrompt le traitement des envois urgents et non urgents ou, selon la politique de gestion adoptée par l'utilisateur, un télégramme de priorité inférieure.

La taille d'un télégramme est variable, mais ne dépasse pas celle d'une lettre. Une lettre a une taille fixe (256 ou 512 octets). Un paquet a une taille variable multiple de la taille d'une lettre.

Un message d'un acteur matériel se compose d'un en-tête (header) et de l'information, c'est-à-dire l'expression à évaluer. L'en-tête comprend :

- la liste des émetteurs,
- la liste des receveurs,
- la liste des écouteurs connus des receveurs,
- la liste des écouteurs non connus des receveurs,
- la liste des receveurs du message réponse,
- le format de message,
- la priorité dans le type de message (un octet = 256 priorités possibles).

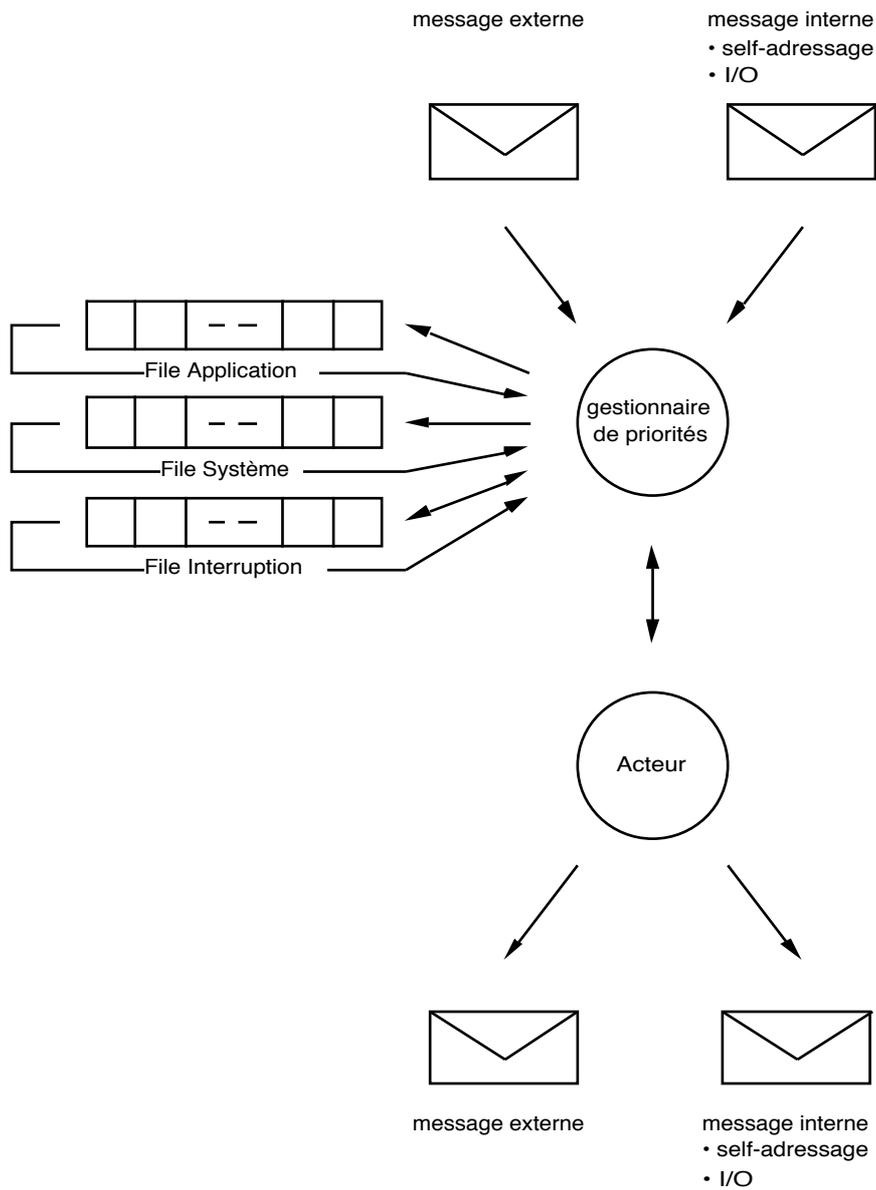


Figure I-7 : Organisation interne d'un acteur matériel

Un groupe est un ensemble d'acteurs. Un identificateur d'émetteur ou de receveur est un identificateur de groupe de communication et un identificateur d'acteur, ou un identificateur de groupe, ou un identificateur d'acteur. Un identificateur de groupe caractérise un ensemble de communicants. La liste des receveurs peut être de type "all" (diffusion totale) ou un identificateur de groupe de communication ou une liste de communicants (diffusion partielle).

3.5. Architecture matérielle

Un acteur matériel se compose :

- d'un ensemble de périphériques d'entrée,
- d'un ensemble de périphériques de sortie,
- de périphériques d'entrée-sortie (principalement des interfaces de communication série, parallèle et réseau),
- d'une carte de commande mono ou multi-processeurs.

La lecture des périphériques d'entrée ou d'entrée-sortie peut être faite aléatoirement, cycliquement (polling), ou sous interruption.

Pour un acteur robotique, les périphériques d'entrée sont essentiellement des capteurs. Les périphériques de sortie sont principalement constitués d'actionneurs.

Nous décrivons de manière détaillée dans le chapitre II ces ensembles.

4. Réseau virtuel

Nous avons défini le modèle acteur matériel. Ce type d'acteur, comme pour le modèle logiciel, communique par messages. Les entités étant matériellement indépendantes, il est nécessaire de les relier via un réseau de communication.

4.1. Définition

ActNet est une plate-forme d'expérimentation robotique pour de nouveaux modèles d'acteurs logiciels et matériels. Ce système propose un environnement de développement et d'exécution d'applications à base d'acteurs [Marsala 92]. Il s'adresse aussi bien aux laboratoires de recherche en IAD qu'au monde éducatif [Darche and Nowak 92 93].

Comme la structure de contrôle dans le modèle acteur est le passage de messages, les acteurs matériels et logiciels communiquent par l'intermédiaire d'un réseau de communication (figure I-10) appelé "ActNet" pour "Network of Actors". Matériellement, ActNet est un réseau d'acteurs hétérogènes, qui se subdivisent en une sous-famille d'acteurs matériels (AM) et en une sous-famille d'acteurs logiciels (AL) qui sont 'hébergés' par ces acteurs matériels. En prenant comme critère de sélection celui de la mobilité, deux types d'acteurs ont été inventoriés : les acteurs robotiques sédentaires (ARS), et les acteurs robotiques mobiles (ARM). Les acteurs matériels peuvent posséder des interfaces spécifiques qui les spécialisent. Nous avons caractérisé trois AM particuliers : l'acteur robotique mobile, l'acteur robotique sédentaire, et l'acteur passerelle.

L'acteur logiciel est une entité communicante qui n'agit physiquement en aucune manière sur son environnement.

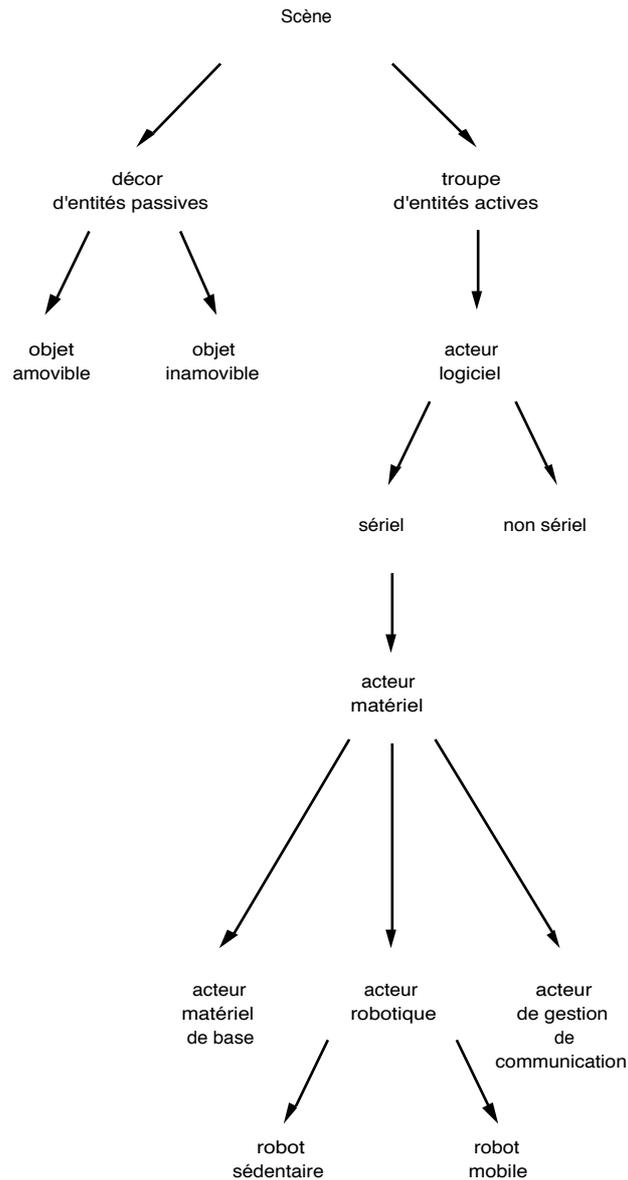


Figure I-8 : Les entités d'ActNet

Nous spécifions et synthétisons la notion d'acteur dans notre projet à l'aide d'une grammaire. Un acteur peut ainsi se définir par les règles suivantes :

- ActNet \rightarrow [acteur]⁺
- acteur \rightarrow acteur_logiciel | acteur_matériel
- acteur_matériel \rightarrow acteur_matériel_de_base | acteur_spécialisé
- acteur_spécialisé \rightarrow acteur_robotique | acteur_gérant_de_communication | autre_acteur
- acteur_robotique \rightarrow acteur_robotique_mobile | acteur_robotique_sédentaire

4.2. ActNet comme un micro-monde

ActNet peut être considéré comme un micro-monde [**Papert 80**], c'est-à-dire un monde en réduction. L'utilisateur se trouve devant un micro-monde nommé scène dans notre application se composant d'un ensemble d'objets passifs amovibles et inamovibles appelé le décor et d'un ensemble d'objets actifs nommés acteurs. Cet ensemble est nommé troupe. Un acteur peut avoir une réalité physique ou être virtuel. Nous avons affaire dans le premier cas à un acteur matériel mobile ou sédentaire, et dans le deuxième cas, à un acteur logiciel sans réalité matérielle. A partir de cette définition, et celle du paragraphe précédent, nous construisons l'arbre des entités d'ActNet. La figure I-8 résume notre propos.

4.3. Groupe de communication

Les aspects techniques de la communication seront exposés dans le chapitre III, nous développons uniquement les principes régissant la communication dans ActNet. Notre préoccupation a été de proposer des principes de communication, et les primitives en découlant, suffisamment généraux pour pouvoir s'adapter aux deux domaines d'utilisation, c'est-à-dire la robotique pédagogique et l'IAD.

La communication dans le système peut être de deux types : active ou passive. La communication est active lorsqu'un acteur communique via les moyens de communication avec un ou plusieurs locuteurs. Elle est passive lorsqu'un acteur déduit une information par observation du monde extérieur; c'est le cas lors d'une collision, les capteurs informent l'acteur de ce fait et il peut en déduire en fonction de sa position dans l'environnement s'il est rentré en collision avec un de ses congénères ou un objet du décor.

La communication à l'intérieur d'un réseau local peut s'effectuer sous la forme d'un monologue de l'acteur lui-même, d'un dialogue entre deux acteurs ou d'un colloque entre plus de deux entités.

Un acteur doit être rattaché à un groupe de communication [**Verissimo and Rodrigues 93**] que nous nommerons troupe. Les communications peuvent être privées ou publiques, uni ou bi-directionnelles comme le montre la figure I-9. Une communication est dite privée lorsqu'un ensemble d'acteurs décident de communiquer entre eux, et personne d'autre ne peut accéder à cette conversation. Une communication peut être publique et tout acteur peut participer à la conversation. Un acteur peut faire partie de plusieurs conversations privées.

Chaque groupe est repéré par un identificateur de groupe de communication. La structure des messages permet d'effectuer des communications du type :

- diffusion totale,
- diffusion partielle selon une liste particulière d'acteurs ou selon un groupe,
- diffusion point-à-point.

Un groupe d'acteurs gèrent un robot. Un seul acteur gère un périphérique.

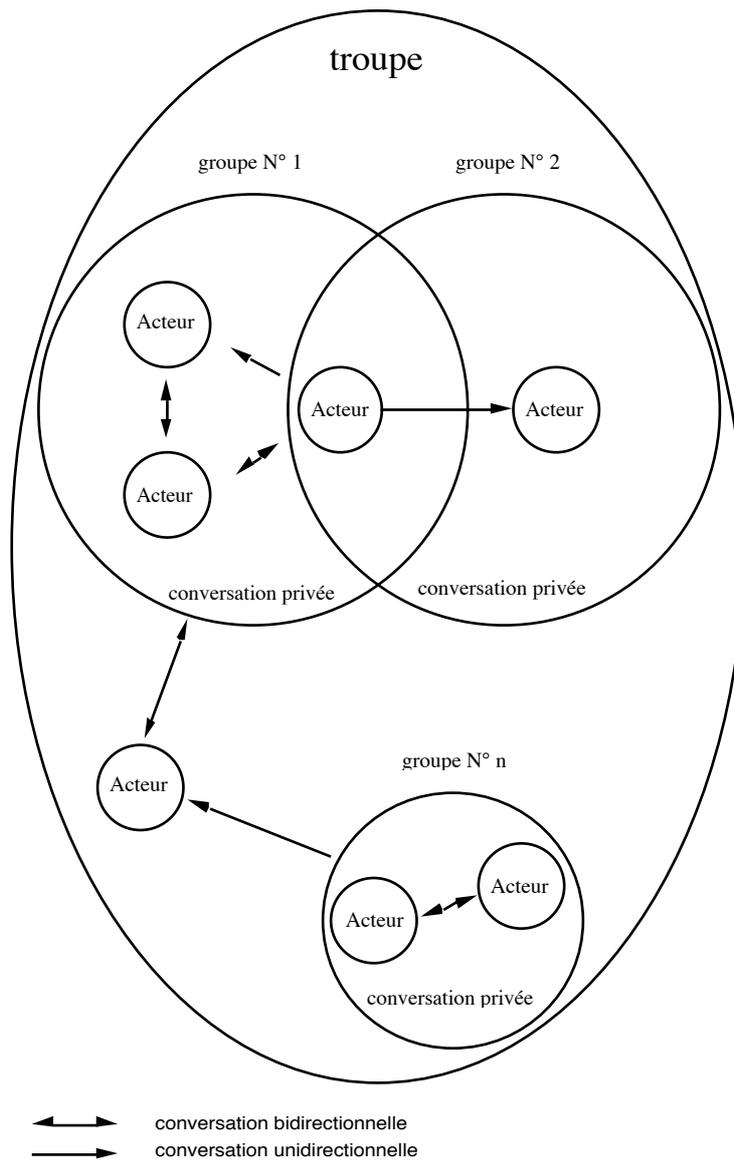


Figure I-9 : Les groupes de communication

Logiquement, un ensemble d'acteurs matériels et logiciels communiquent via un réseau de communication (Figure I-10). Physiquement, le réseau externe de communication relie un ensemble de noeuds de communication. Un noeud de communication héberge un et un seul acteur matériel. Il possède donc une adresse réseau physique. Un acteur matériel peut posséder n modules processeurs P_0 à P_n ($n \geq 1$) et chaque module processeur héberge p acteurs logiciels A_0 à A_p ($p \geq 1$). Un processeur est un mono-processeur à mémoire locale qui communique par messages. La figure I-11 résume notre propos.

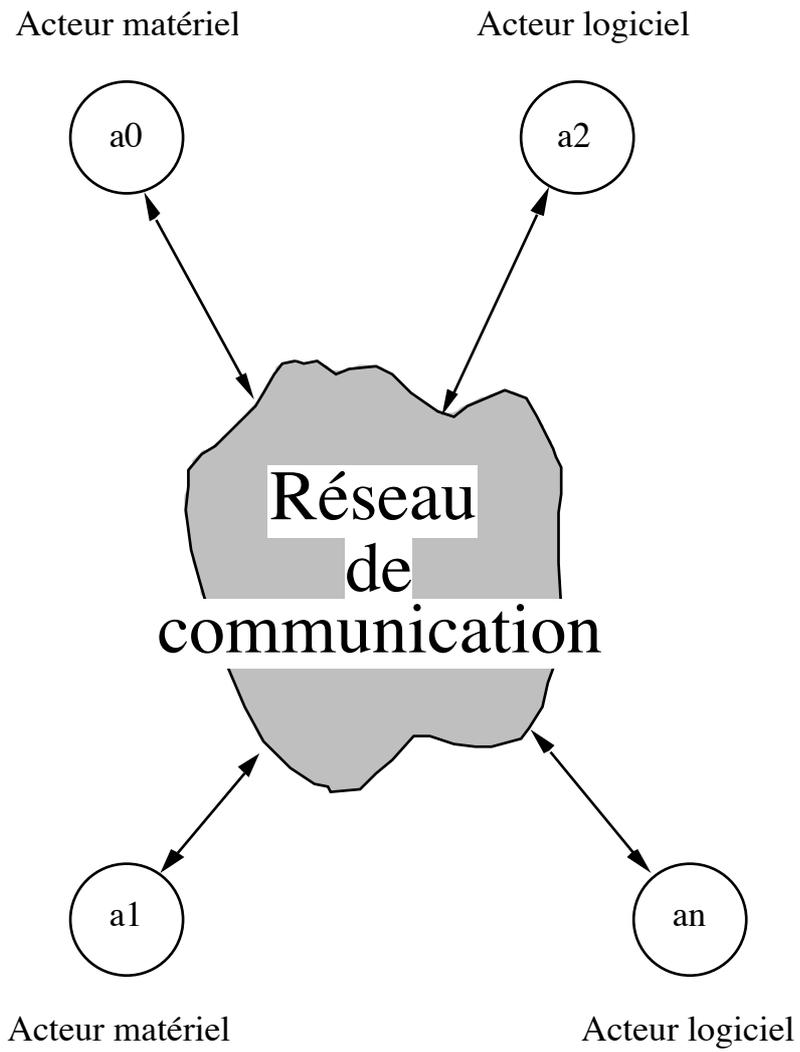


Figure I-10 : Topologie logique d'ActNet

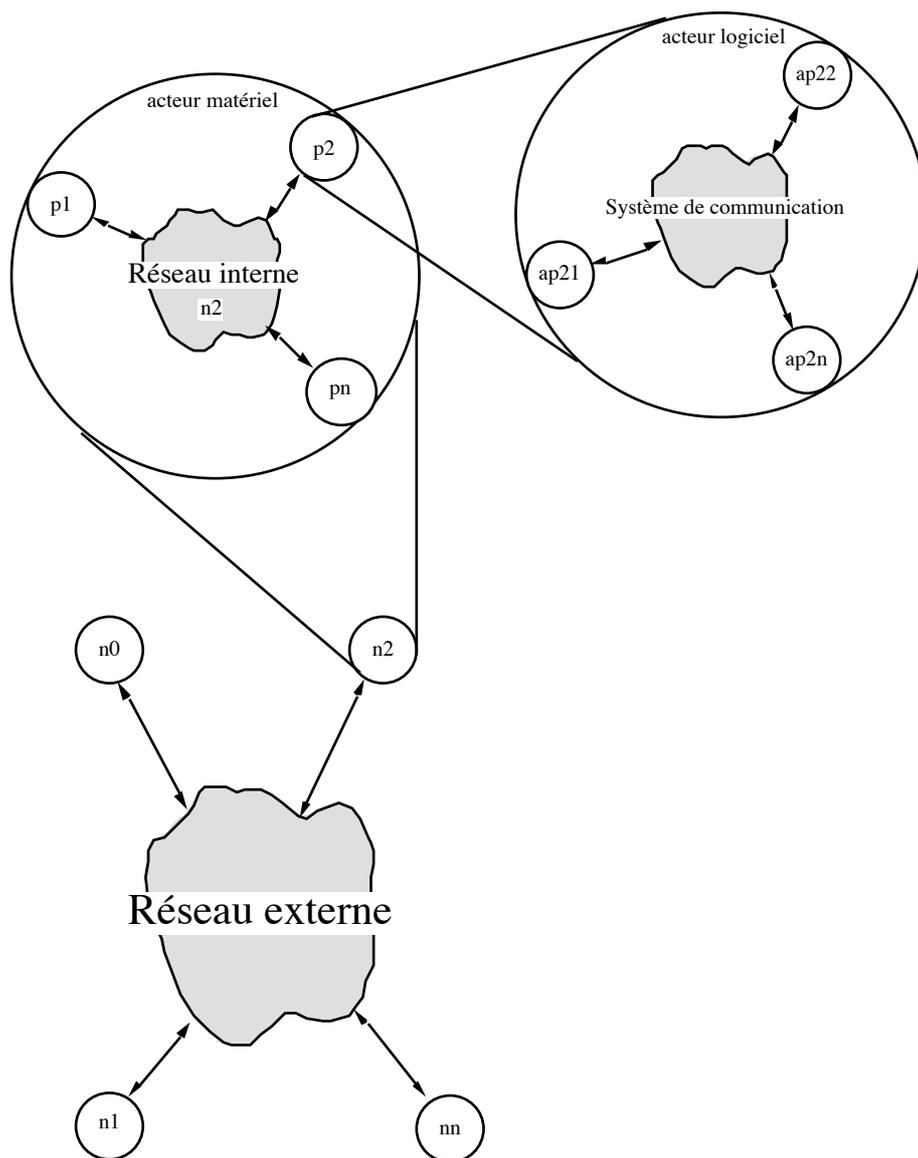


Figure I-11 : Topologie physique d'ActNet

4.4. Topologie

Les acteurs matériels se décomposent du point de vue de la mobilité en des entités statiques et des entités nomades. A chaque type correspondra un sous-réseau de communication :

- pour les acteurs matériels sédentaires, un réseau filaire de type anneau à débit élevé. Il est nommé Sous-Réseau d'Acteurs Sédentaires (SRAS). Cet anneau est du type double anneau contre-rotatif, car il permet de fiabiliser le fonctionnement du sous-réseau,
- pour les acteurs mobiles, le respect de la contrainte d'autonomie d'un point de vue communication fait que le milieu ambiant (éther) sera utilisé comme support de transmission. Ces éléments forment le Sous-Réseau d'Acteurs Mobiles (SRAM).

Ces deux sous-réseaux forment le Sous-Réseau Exécutif (SRE). A ces deux sous-réseaux, il faut ajouter le Sous-Réseau Utilisateur (SRU), dédié au développement et au contrôle des applications. Notre propos est résumé en figure I-12.

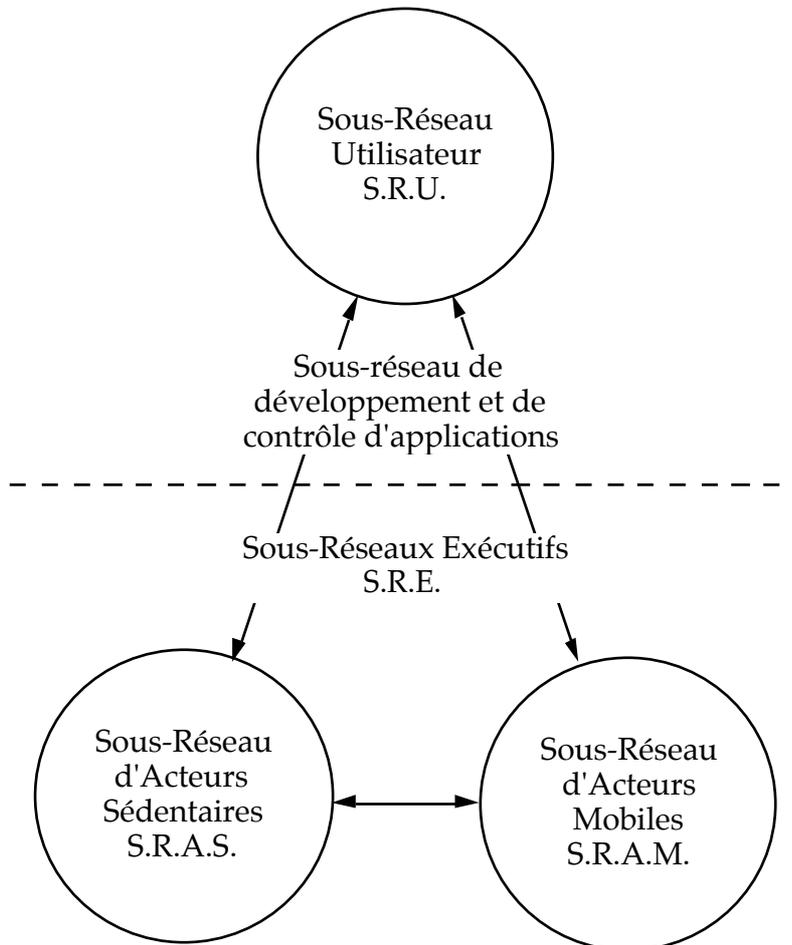


Figure I-12 : Topologie générale d'ActNet

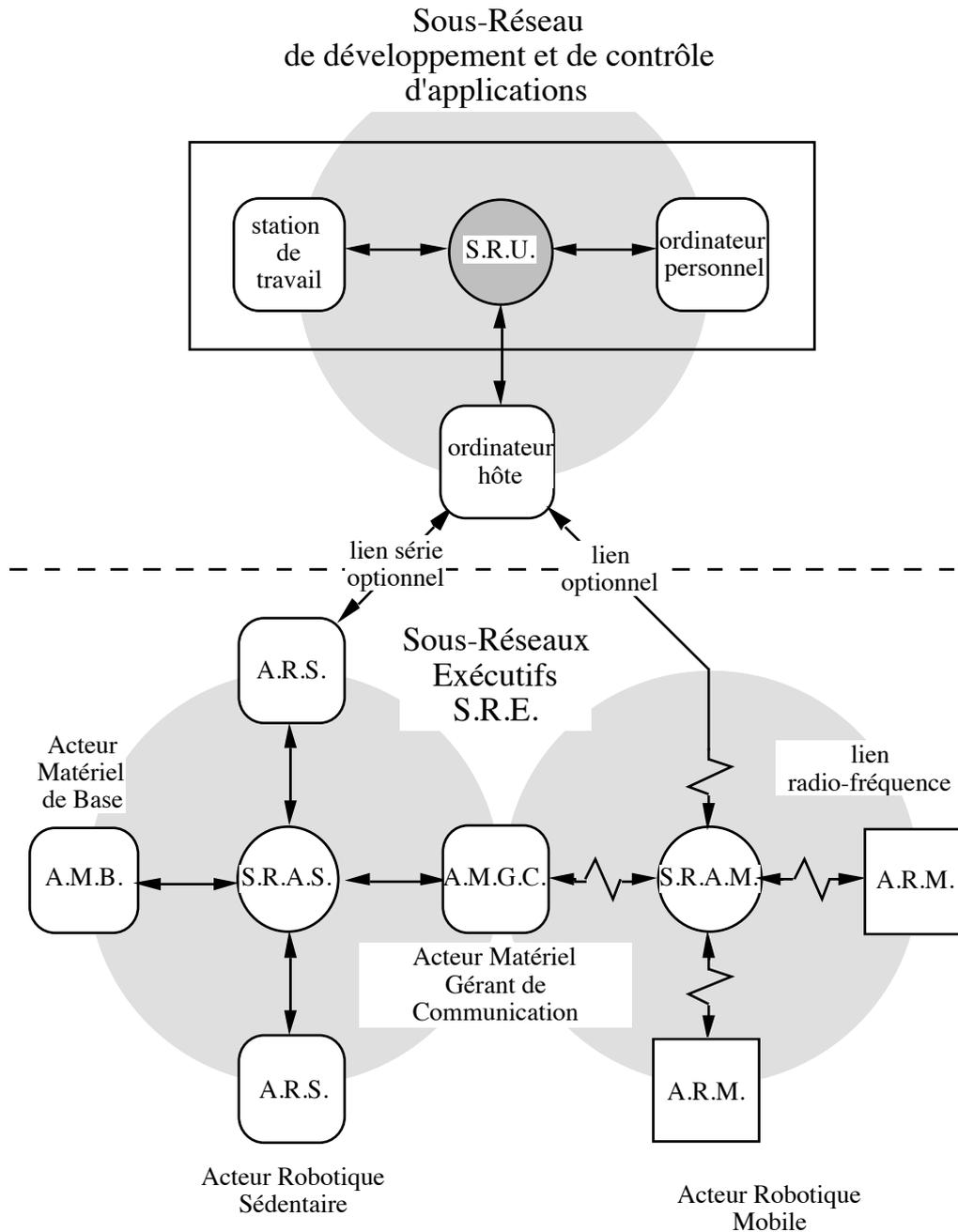


Figure I-13 : Topologie détaillée d'ActNet

Le réseau est hétérogène par sa topologie programmable (SRAM, réseau ou bus), mais aussi par ses types de processeurs équipant chaque noeud. Le SRU est un réseau local commercial avec une topologie variable définie par les ordinateurs supportant les développements des utilisateurs. Il se réduit pour le développement à un micro-ordinateur équipé d'une carte d'interface pour le SRE. Le SRAS est un double anneau contre-rotatif pour augmenter la fiabilité et assurer une vitesse garantie et élevée pour le flot de données.

Des passerelles permettent de faire communiquer les deux sous-réseaux. Le réseau utilisateur sert de serveur disque pour le système (figure. I-13).

De la description topologique, nous déduisons les rôles spécialisés des acteurs :

- l'acteur robotique, mobile (ARM) ou sédentaire (ARS), qui est une entité autonome capable d'interagir avec son environnement (objets passifs et autres acteurs),
- l'acteur matériel de base (AMB), dont la présence est requise si nécessaire pour coordonner le développement d'une application,
- l'acteur matériel gestionnaire de communication (AMGC), effectuant la passerelle entre le SRAM et le SRAS, et
- l'ordinateur hôte (un PC actuellement), pont entre le SRU et le SRAS.

En ce qui concerne les Acteurs Robotiques Mobiles, la télécommunication par canal filaire est envisagée uniquement dans les cas suivants :

- lors d'une défaillance du système embarqué fournisseur d'énergie, un apport d'énergie extérieur est nécessaire :
 - pour alimenter les systèmes électroniques,
 - pour recharger les batteries,la ligne d'alimentation générale du réseau remplirait alors ce rôle;
- en phase de développement pour un téléchargement rapide de logiciels prototypes ou pour une mise au point des applications;
- lors d'une défaillance du système radio-fréquence de télécommunication.

5. Conclusion

Après une présentation des modèles acteur de base et des principaux langages d'acteurs, nous avons décrit le modèle acteur matériel. Ensuite, nous avons défini la notion d'acteur robotique qui est un acteur sériel. Son comportement et ses spécificités ont été explicités.

La communication a ensuite été détaillée avec en particulier les trois types de messages, passé, présent et futur, que l'acteur matériel peut envoyer ou recevoir. La notion de réseau virtuel a été introduite et nous avons décrit le système ActNet composé de trois sous-réseaux : sous-réseau utilisateur (SRU), sous-réseau d'acteurs mobiles (SRAM) et sous-réseau d'acteurs sédentaires (SRAS).

Le prochain chapitre détaillera l'architecture matérielle des acteurs.

Chapitre II

Architecture matérielle des acteurs robotiques

- 1. Introduction**
- 2. Robotique mobile**
- 3. Robot et robotique pédagogique**
- 4. Contrôle des acteurs matériels**
- 5. Le robot mobile “Tatou”**
- 6. Le robot bras**
- 7. Uniformisation des E/S**
- 8. Conclusion**

1. Introduction

Nous présentons dans ce chapitre l'architecture matérielle des acteurs matériels et en particulier celle des acteurs robotiques intervenant dans notre projet. Nous commençons par un état de l'art de la robotique mobile et nous détaillons en particulier dans les architectures de contrôle, l'architecture comportementale de Brooks [**Brooks 85**]. Quelques réalisations de robots sont décrites. Nous donnons ensuite une définition de la robotique pédagogique et nous présentons une proposition de classification des robots pédagogiques. Nous étudions les interfaces existantes des robots pédagogiques et nous proposons une classification de celles-ci.

Nous avons développé une carte générique de commande qui permet de s'adapter à tout type de robots pédagogiques. Nous avons sélectionné pour la réalisation le transputer et proposons une amélioration matérielle pour prendre en compte plus d'une interruption.

ActNet est un micro-monde reposant sur le concept d'acteur matériel. Nous avons utilisé des robots pédagogiques pour construire nos acteurs robotiques et valider nos idées. L'utilisation de robots pédagogiques dans notre projet se justifie par le fait qu'ils sont robustes, d'un bon rapport qualité/prix, et que les performances (précision, répétitivité, etc.), en dépit du facteur d'échelle, sont comparables à celles de leurs homologues industriels.

Nous avons sélectionné un robot de chaque sous-famille d'acteurs matériels : un robot mobile prototype "le tatou" développé au laboratoire et un robot sédentaire commercial de type bras. Nous détaillons d'un point de vue architecture matérielle les modules matériels les constituant. A chaque module correspondent des primitives de commande qui sont présentées dans le chapitre IV décrivant la partie logicielle du projet. Lors de la description du robot mobile prototype, nous proposons un modèle de description d'un acteur matériel, l'Actor_Unit, que nous appliquons à celui-ci.

L'étude de la connexion d'un robot pédagogique avec un système de contrôle a soulevé le problème de l'interface. Nous proposons d'uniformiser l'interface afin de ne plus être tributaire du matériel, sous la forme d'un module adaptateur, le système NIOS.

2. Robotique mobile

A l'origine vers les années 50, les robots étaient de simples manipulateurs. Ces robots étaient de première génération [**Coiffet 86**]. Les robots de la deuxième génération dans les années 60/70 intégrèrent des capacités de perception du monde les environnant. Ils étaient commandés par des ordinateurs programmés en langages structurés classiques avec des instructions spécialisées dans la gestion des entrées-sorties. La génération actuelle (troisième génération) est dite intelligente. Elle intègre des fonctions de raisonnement et de décision sur les tâches à accomplir. Les techniques de l'IA sont largement mises à contribution (planification, résolution de problèmes).

En robotique classique, il existe essentiellement trois grandes classifications [Coiffet 86] :

- la classification de la JIRA (Japan Industrial Robotic Association),
- la classification de la RIA (Robot Industrial Association) aux USA,
- la classification de l'AFRI (Association Française de Robotique Industrielle).

Selon l'AFRI, il convient de distinguer :

- le manipulateur manuel, matériel de manipulation motorisé commandé par l'homme ayant au moins quatre degrés de liberté (un télémanipulateur par exemple),
- le manipulateur automatique, matériel de manipulation ayant au moins deux axes non asservis à cycle automatique (manipulateur de chargement/déchargement par exemple),
- le robot programmable : manipulateur à trois axes minimum dont deux au moins sont programmables par apprentissage et/ou langage symbolique,
- le robot *intelligent*, manipulateur automatique capable d'analyser son environnement et de réagir aux modifications de celui-ci par rapport aux tâches à accomplir.

Seules les deux dernières catégories entrent dans la catégorie des robots industriels.

Au niveau applicatif, les différents types de robots actuels en fonction du domaine d'utilisation sont les systèmes industriels, les systèmes de maintenance, les systèmes d'automatisation bureautique et les systèmes de sécurité [Meystel 91]. Depuis 1980, la robotique est associée à la productique c'est-à-dire à l'ensemble des techniques d'automatisation de la production : conception (CAO), fabrication (CFAO) et gestion de la production (GPAO).

Le domaine qui nous intéresse, à savoir la robotique mobile, a constitué un pas important dans l'évolution de la robotique. Ce domaine a permis de faire évoluer l'intelligence en robotique et la technique au niveau des capteurs et de la gestion de l'énergie embarquée.

2.1. Définition

Un robot mobile est un robot qui est capable de se déplacer, contrairement aux robots à poste fixe qui ne possèdent pas de base mobile. Nous nous situons dans le domaine de la Robotique Mobile Autonome (RMA) ou Autonomous Mobile Robots (AMR).

Un robot mobile se caractérise par son autonomie. L'autonomie se définit comme la capacité que possède un système vivant ou artificiel à maintenir sa viabilité dans des environnements variés, structurés ou non structurés, et changeants, sans contrôle extérieur. L'autonomie est donc la capacité que possède le robot à accomplir sa ou ses missions dans un environnement donné sans intervention externe.

Le robot mobile remplit une ou plusieurs des fonctions suivantes :

- transport, principalement en milieu industriel,
- sécuritaire,

- exploration de zones dangereuses, comme les réacteurs nucléaires par exemple, ou exploration planétaire avec le projet VAP du CNES.

Un robot mobile se compose de quatre sous-ensembles : le système de locomotion, le système de contrôle, le système énergétique (alimentation, source d'énergie, etc.) et la charge utile (bras manipulateur, caméra, etc.).

Le robot mobile est généralement caractérisé par trois grandes fonctions **[Coiffet 86]** :

- la fonction locomotion,
- la fonction perception, décision,
- la fonction localisation.

Au niveau locomotion, selon **[Hirose 82]** repris dans **[Hirose and Morishima 90]**, les robots mobiles appartiennent à trois sous-familles ou configurations :

- les roues et chenilles,
- les pattes,
- les corps articulés.

Un robot mobile peut appartenir à une ou plusieurs de ces trois sous-familles. Les roues et les chenilles, inventions humaines, sont les principaux moyens de locomotion. Elles présentent une faible capacité d'adaptation au terrain, mais une nouvelle tendance utilise les pattes. Les pattes présentent la meilleure adaptabilité, pourvu que la commande soit bien faite car les problèmes de stabilité doivent alors être pris en compte. Les corps articulés sont rarement utilisés en robotique mobile. Cette configuration a souvent été inspirée du serpent. Nous retrouvons ce type de robot en particulier pour les inspections de conduits.

La fonction perception s'effectue grâce aux capteurs qui renseignent la partie contrôle sur l'état interne du robot et sur le monde extérieur. Toujours d'après **[Coiffet 86]**, les capteurs se subdivisent en trois grandes classes : les capteurs proprioceptifs, les capteurs extéroceptifs et les capteurs de sécurité. Un capteur proprioceptif permet de renseigner le système sur l'état interne des éléments constitutifs du robot (température, tension(s) d'alimentation, état de charge de la source d'énergie embarquée, etc.). Un capteur extéroceptif permet d'appréhender le monde extérieur par la mesure de grandeurs physiques comme par exemple la température ou la lumière. Les capteurs de sécurité permettent de sauvegarder les éléments mécaniques et électroniques du robot, et aussi la vie des utilisateurs. Grâce à ces informations, le robot est à même de prendre des décisions, en fonction des tâches et missions qui lui sont assignées.

Lorsque le nombre de capteurs augmente, le problème de l'acheminement des données se pose, à cause de l'encombrement dû à la multiplication des connecteurs et surtout des câbles, le coût intervenant aussi de ce fait. La solution est d'organiser ces capteurs en réseau, les performances de celui-ci (vitesse, fiabilité, distance maximale, etc.) étant fonction de l'application. Des bus à trois fils comme le I2C de Philips ou à deux fils comme le Meter-Bus (M-Bus) permettent de relier des sous-ensembles capteurs. L'acquisition peut se faire périodiquement par scrutation (polling) ou sous interruption. Le contrôle logiciel peut inhiber certains d'entre eux.

La fonction localisation permet de renseigner le robot sur sa position par rapport aux trois axes. Elle s'effectue grâce à des capteurs spécifiques de type infra-rouge ou ultra-sonique par mesures de distance et d'angles par rapport à des objets à position connue.

2.2. Architecture de contrôle

Selon [Cox and Gehani 89], la programmation des robots est caractérisée par le besoin :

- d'attendre des événements internes ou externes,
- de gérer les activités concurrentes,
- de synchroniser des actions avec des événements,
- et de communiquer de manière interne (inter-processus) ou externe (entre robots).

Tous ces processus doivent être évidemment traités en parallèle, ou du moins en quasi-parallélisme. A cela s'ajoute la nécessité de gérer en temps réel la synchronisation des tâches et les périphériques d'entrée/sortie.

Deux tendances s'affrontent actuellement :

- l'architecture classique centralisée [Narasimhan et al. 88] avec la décomposition fonctionnelle suivante : perception, planification, et action,
- ou l'approche comportementaliste ou réactive de type Brooks par décomposition comportementale [Brooks 85] [Brooks 86a].

Cette deuxième approche découle du constat suivant : les robots actuels sont trop complexes à concevoir, la robotique classique impose des besoins informatiques souvent incompatibles avec les missions temps réel que les robots doivent remplir, en particulier en robotique mobile. L'électronique sous-jacente est souvent incompatible avec les contraintes d'autonomie et/ou impose des dimensions physiques trop importantes.

L'architecture de Brooks

En robotique classique, la simulation est de plus en plus employée pour valider les concepts. L'idée de base est donc de refuser la simulation car il existe une différence de perception entre le monde simulé et le monde réel [Brooks 92], cette différence étant due aux erreurs de la chaîne de mesure (capteurs, conversion, etc.). Il existe aussi une différence dans l'action, différence due à la chaîne d'actions (commande, actionneurs, etc.). Il est en fait très difficile de modéliser et simuler la dynamique du monde réel. Il s'agit aussi de faire le plus simple pour la plus grande efficacité. L'analyse revient à décomposer par couches successives les comportements en sachant qu'une couche inférieure peut être inhibée par une couche supérieure.

Brooks propose l'**Architecture de subsomption** [Brooks 85] [Brooks 86a]. Cette architecture est à contre courant de la robotique classique. Elle repose sur la réactivité permettant d'obtenir des temps de réponse rapides, et sur la multiplicité des robots. La

simplicité de l'architecture permet de remplir des missions élémentaires, qui associées à un nombre élevé de robots, fait émerger une intelligence collective. L'activité du robot est la somme de multiples activités élémentaires s'exécutant en parallèle. C'est une architecture verticale où une action est la somme des réponses pondérées de tâches verticales. Il n'y a pas de communication directe entre entités, seule la communication indirecte via l'environnement existe. Cette approche permet de câbler ces comportements et elle s'adapte donc très bien aux nouvelles architectures comme les réseaux neuronaux. Cette approche par construction successive de couches comportementales présente l'inconvénient qu'il faille tenir compte des couches inférieures pour construire les couches au-dessus. De plus, la notion de temps réel n'est pas prise en compte (assurance des temps de réponse). Il n'y a pas de planification des actions.

Le domaine d'utilisation est l'IAD avec les agents réactifs. Le découpage comportemental a évolué et une réalisation étonnante avec ce type d'architecture est le robot Herbert **[Brooks et al. 88]** **[Connell 89]** qui est un robot mobile cylindrique de 18 pouces de diamètre ($\approx 46\text{cm}$) et 4 pieds de hauteur ($\approx 1,22\text{ m}$). Son unique tâche est de collecter les canettes de soda ! Trois roues servent à sa mobilité. Pour la détection des obstacles, il y a deux anneaux de 16 capteurs de proximité infra-rouges. Un bras à deux degrés de liberté équipé de ses capteurs permet de collecter les canettes. L'architecture de commande est un raffinement de la "subsumption architecture". Le système est organisée en couches et contrairement aux couches comportementales de Brooks qui définit un ordre total sur les comportements du robot, ici elles définissent un ordre partiel en forme d'arbre. Huit micro-processeurs d'architecture huit bits pilotent l'ensemble. Chaque contrôleur est indépendant et la communication se fait par une liaison série 300 bauds.

Mais comme dans tout domaine scientifique, Brooks a des détracteurs et des chercheurs **[Etzioni 93]** pensent que la robotique n'est pas la seule issue pour l'IAD. Les environnements logiciels permettent de développer, pour un coût financier et temporel moins élevé, des applications à base d'agents. De plus, la mise au point est moins fastidieuse.

2.3. Coopération

Le robot ne travaille plus de manière isolée. Il doit maintenant coopérer pour mener sa mission à terme. Le système de **[Premvuti and Yuta 93]** fait fonctionner des robots autonomes à comportement coopératif modeste, c'est-à-dire qu'ils partagent des informations sans contrôle centralisé et qu'ils utilisent des informations locales.

La coopération est composée de deux phases : la collaboration et la coordination **[Noreils 93]**. Il peut s'agir d'un ensemble de robots hétérogènes ou homogènes.

Les interactions que peuvent avoir les RMA avec le monde extérieur se manifestent sous plusieurs formes : communication, coordination, coopération, coadaptation, coévolution, etc.

Une recherche a lieu pour mettre à la disposition des utilisateurs un groupe de robots mobiles comme s'il s'agissait d'ordinateurs [**Yakoh and Anzai 93**]. Le système s'appelle Prime et est destiné au domaine de l'IA particulièrement.

2.4. Exemples

Au niveau de la recherche appliquée, nous retrouvons deux familles de robots mobiles caractérisées par leur encombrement et leur poids. La première famille concerne les gros systèmes qui embarquent une source d'énergie volumineuse, des systèmes industriels de contrôle sous forme de racks et des systèmes d'asservissement de puissance et la famille des petits robots mobiles embarquant un micro-contrôleur et une source d'énergie limitée. Ces petits robots ne permettent de valider en général qu'un concept et possèdent des performances restreintes (autonomie, etc.). Les systèmes de développement intermédiaires n'existent pratiquement pas. Nous présentons quelques réalisations représentatives du domaine.

Parmi les gros robots, un des premiers robots mobiles (fin des années soixante) fût le robot "Shakey" de l'Institut de Recherche de Stanford (Stanford Research Institute) [**Meystel 91**]. Ce robot a été développé dans le but d'étudier des applications pour l'Intelligence Artificielle. Il y eu aussi "Hexapod" de l'Ohio State University. En France, le projet Hilare (Heuristiques Intégrées au Logiciel et aux Automatismes dans un Robot Evolutif) au Laboratoire d'Automatique et d'Analyse des Systèmes de Toulouse initialise, en septembre 1977, la recherche en robotique mobile [**Giralt et al. 79**].

Plus récemment, un représentant de la fonction sécuritaire est "ROBART II" qui est un robot mobile autonome. Il est utilisé par le Naval Ocean Systems Center à San Diego (USA) comme plate-forme de test pour les systèmes de sécurité. Il possède des capteurs de détection et de vérification de différentes natures, qui, après traitement "intelligent", doivent être capables de remplir une mission sécuritaire. C'est donc un système automatique qui fournit une haute capacité de détection avec la capacité de faire la distinction entre les alarmes vraies et fausses [**Everett and Gilbreath 88**]. Il est équipé d'une multitude de capteurs pour mesurer la température, la pression, le taux d'humidité, la lumière, les infra-rouges, les distances entre autres, et d'un système vidéo stéréoscopique. Son unité de commande est équipée de huit systèmes à micro-processeur communiquant via plusieurs réseaux locaux. Le planificateur n'est pas embarqué et communique avec le robot grâce à un lien radio-fréquence à 9600 bauds.

En ce qui concerne les petits robots, "Alder" [**Nehmzow et al. 89**] issu du projet RUR (Really Useful Project), à base de pièces Fischertechnik est un robot mobile qui sert de plate-forme d'expérimentation pour développer des schémas de contrôle adaptatif robustes et flexibles. Sa carte de contrôle est basée sur le micro-contrôleur 8 bits Intel 8052 AH Basic qui contient, comme sa référence le laissait présager, un interpréteur basic. Une carte d'interface gère deux moteurs en boucle ouverte et huit capteurs tactiles binaires pour appréhender son environnement. Les mêmes chercheurs ont construit "Caingorm", le troisième robot du projet RUR avec la même structure mécanique mais qui est équipé d'une

carte à base de micro-processeur Motorola 68000 avec 128 Ko de mémoire. Ce robot est programmé en langage C. Ces robots sont capables d'éviter des obstacles ou de suivre les murs grâce à un réseau neuronal.

La tendance dans les petits robots est la miniaturisation. Deux réalisations remarquables utilisées en IAD sont le robot suisse "Khepera" [**Mondada and Verschure 93**] et le robot américain "Squirt" [**Flynn et al. 89**].

Le robot "Khepera" est un robot mobile miniature, spécialement conçu pour tester des algorithmes de contrôle. Il est de forme cylindrique d'une hauteur de 30 mm et d'un diamètre de 55 mm. Il est constitué de deux plate-formes : la base et la carte CPU qui s'emboîtent selon le modèle *piggy-back*. Les connecteurs électriques servent en même temps d'armatures mécaniques. La plate-forme de base supporte deux moteurs courant continu, des détecteurs d'obstacles basés sur la réflexion de signaux infra-rouges modulés et des détecteurs de lumière. Un codeur incrémental couplé à l'axe de rotation des roues permet de déterminer la position de celles-ci. La carte CPU est composée d'un micro-contrôleur Motorola 68331 cadencé à 16 MHz, de 256 Ko de RAM, de 256 Ko de ROM, d'un convertisseur A/N 10 bits à 6 entrées et un port série de 38400 bauds. Des plate-formes supplémentaires peuvent être rajoutées comme le module vision qui comporte deux caméras CCD 64 x 1 pixels. L'autonomie de fonctionnement est de 20 mn. Sa petite taille s'adapte bien aux besoins de l'IAD.

Le robot "Squirt" est un robot mobile d'un coût très faible (\approx \$25) et de dimensions réduites (l'ensemble tient dans un cube d'environ 3 cm d'arête). Son poids est d'environ 50 g. Ces caractéristiques le destinent particulièrement aux expérimentations en IAD. Il est équipé d'un micro-contrôleur 68HC811 cadencé à 8 MHz avec 2 Ko d'EEPROM et 256 octets de RAM intégrés. Deux moteurs à courant continu suisses assurent la propulsion. Au niveau capteurs, il est équipé de deux microphones, d'une photo-cellule. Deux piles au lithium forment la source d'énergie. Son architecture logicielle est basée sur l'architecture à subsomption de Brooks.

Tous les robots présentés ci-dessus sont à roues. Pour terminer, nous présentons un robot marcheur à six pattes "Ghenghis" [**Brooks 89**]. Ce robot a une longueur de 35 cm, une longueur de patte de 25 cm et pèse approximativement 1 kg. Malgré sa taille, il réussit à embarquer une électronique et une informatique capables de gérer six pattes. Chaque patte possède deux degrés de liberté et est animée par un servo-mécanisme de type modélisme. Quatre micro-contrôleurs 8 bits Motorola MC68HC11 avec 1Ko de RAM et 10 Ko d'EPRM, un par paire de patte et un pour gérer les capteurs, sont reliés par un anneau à jeton à 62,5 Kbits/s. Un logiciel basé sur l'architecture de subsomption composé de 8 couches comportementales gère la marche, la poursuite de personnes à démarche lente et l'évitement d'obstacles. Les 8 couches sont composées de 57 machines à états finis. Cinq sont spécialisées dans la marche et sont dupliquées pour chaque patte. Deux sont chargées de la coordination totale des pattes. Les 25 machines restantes gèrent les collisions, l'évitement d'obstacles et la poursuite de cibles. L'énergie embarquée est constituée de trois piles zinc/argent.

3. Robot et robotique pédagogique

ActNet utilise des acteurs robotiques dont la base électro-mécanique est un robot pédagogique. Avant de définir ce qu'est un robot pédagogique, donnons une définition du domaine robotique pédagogique.

3.1. Définition de la robotique pédagogique

Une tentative de définition de la robotique pédagogique a été donnée dans [Vivet 90]. La micro-robotique pédagogique est une activité de conception, création/mise en oeuvre, à des fins pédagogiques, d'objets techniques physiques qui sont des réductions aussi fidèles et significatives que possible de procédés/outils robotiques réellement utilisés dans la vie courante, en particulier en milieu industriel. Souvent Logo est associé aux activités micro-robotiques [Vivet et al. 91].

3.2. Définition d'un robot pédagogique

Un robot pédagogique est un robot de taille réduite (< 1 m sur les trois axes). Il est une copie physique conforme ou non et reproduit plus ou moins les fonctionnalités des modèles réels. Ses capacités techniques (précision, répétabilité, puissance, ...) sont inférieures à celles des robots industriels. Le robot pédagogique est principalement utilisé comme outil pour l'apprentissage de la robotique ou tout autre domaine.

Il n'y a pas de classification en robotique pédagogique. Nous avons détaillé dans la première section la classification de l'AFRI concernant la robotique industrielle. Nous avons préféré classer les robots pédagogiques en fonction de leur aspect extérieur, et non par leurs fonctionnalités ou leurs contrôles, bien que ces classifications puissent s'y appliquer. A partir de cette définition, nous distinguons trois catégories :

- les robots de type 'industriel', complètement fonctionnels, reproduction à échelle réduite de robots industriels existants; leur conception est robuste,
- les robots de type 'jouet éducatif', comme par exemple ceux des firmes Lego ou Fischertechnik, dont l'aspect rappelle les formes des robots industriels et reproduit une partie des fonctionnalités (préhension, mouvement de déplacement, ...); leur conception est fragile,
- les robots issus de la recherche.

Ces derniers existent souvent sous la forme de modèle prototype, nous pensons par exemple à la tortue de sol Logo, premier objet informatique à vocation pédagogique, du domaine de la psycho-pédagogie, dont la coccinelle [Tanguy 87] est une réalisation.

Ce concept a donné lieu à des réalisations commerciales. Citons principalement quatre robots : le robot PIP de la société Swallow Systems, le robot mobile Roamer de la société Valiant Technology Ltd et le robot T3 de la société Jeulin. Ils sont équipés d'un stylo pour dessiner au sol et, pour les deux premiers, d'un clavier pour pouvoir programmer le déplacement. Une interface sonore et lumineuse est aussi disponible. Ils sont destinés aux jeunes enfants. Mais des robots pédagogiques plus évolués peuvent être destinés à des

fonctions plus complexes, comme le robot KitBorg de la société Aleph Technologies utilisé pour l'apprentissage de l'automatisme avec la possibilité de programmer en Grafcet et en C. Tous ces robots mobiles sont équipés d'un micro-processeur ou pour une solution plus intégrée, d'un micro-contrôleur 8 bits.

Les constructeurs que nous pourrions qualifier de représentatifs dans le domaine de la robotique pédagogique et ludique, sont au nombre de trois principalement : Fischertechnik®, Lego technic® et peut-être Meccano®. Ces robots ont une structure mécanique composée d'éléments pré-définis. L'avantage de cette solution est bien évidemment le coût. Mais les éléments de construction sont prédéfinis et les formes et fonctions reproduites ne peuvent être qu'approximatives par rapport à un robot conçu spécifiquement.

3.3. Etude de l'interface de commande

Avant de détailler l'interface de commande d'un robot pédagogique, il nous faut parler des différents types de robots pédagogiques. Ensuite, nous étudierons dans cette section la chaîne de commande d'un robot pédagogique et les différents types d'interfaces existants.

3.3.1. Classification des robots pédagogiques

Les robots pédagogiques sont en général pilotés par un micro-ordinateur. Il existe deux types de robots :

- le robot à "commande évoluée",
- le robot à "commande de bas niveau".

Le robot est dit 'à commande évoluée' lorsqu'il est capable d'interpréter les commandes de haut niveau provenant de son interface de communication et d'exécuter celles-ci. Il fournit ou non en retour un compte-rendu d'exécution. Le robot est dit à 'commande de bas niveau' s'il reçoit directement les signaux électriques de commande des actionneurs et fournit une lecture directe des capteurs sans interprétation de mesure.

La liaison robot-ordinateur n'est pas directe, mais s'effectue par l'intermédiaire d'une interface (figure II-1). Une interface robotique a pour fonction d'interpréter les commandes et de fournir des signaux adaptés au robot. L'interface peut être incluse dans le micro-ordinateur, dans le robot, ou externe. L'intelligence de commande est au niveau du micro-ordinateur et facultativement aux niveaux interface et robot. Les interfaces peuvent donc être classifiées de la manière suivante :

- intégrées ou externes,
- à 'commande évoluée' ou à 'commande de bas niveau',
- dédiées ou non à une machine ou à un type de machine : Macintosh®, PC/AT, BBC®, etc.,
- dédiées ou non à un type de matériel robotique : Lego®, Fischertechnik®, etc.

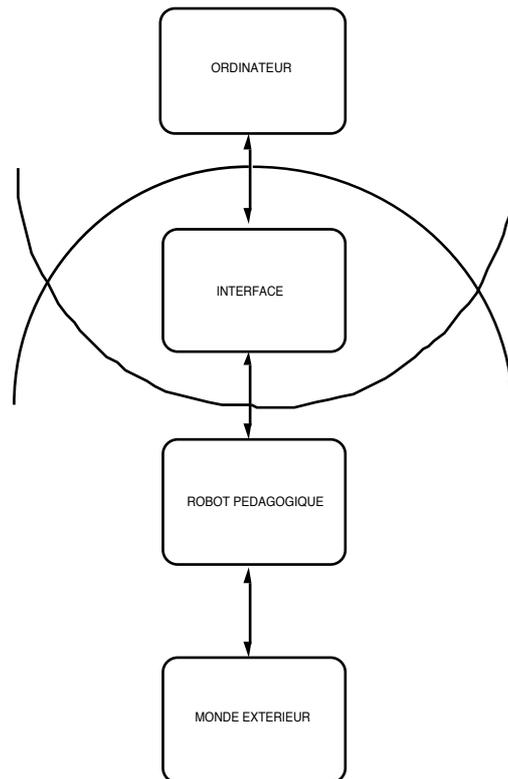


Figure II-1 : Chaîne de contrôle d'un robot pédagogique

Comme ces interfaces s'intercalent entre la machine de contrôle, en général un micro-ordinateur, et le robot, il faut alors considérer deux liaisons : la liaison ordinateur-interface et la liaison interface-robot.

3.3.2. Liaison interface-robot

En ce qui concerne la liaison interface-robot, quatre cas sont à considérer :

- Aucune 'intelligence' aux niveaux interface et robot. L'interface sert donc d'adaptateur de signal. L'interprétation est faite au niveau du logiciel de commande. Au retour, il y a adaptation des signaux provenant principalement des capteurs sans traitement spécifique. C'est le cas du robot YOUPI [J.D. Productique 86] (interface type 1).
- 'L'intelligence' de commande est au niveau de l'interface. L'interface interprète une commande de haut niveau envoyée par l'ordinateur et la traduit en un signal de puissance qui commande directement les effecteurs. En retour, elle renvoie une donnée numérique, résultat direct de la conversion du signal provenant de la lecture des capteurs (interface type 2).
- 'L'intelligence' de commande est au niveau robot. L'interface sert uniquement d'adaptateur de communication entre le robot et l'ordinateur (interface type 3).

- acteur_robotique_mobile —> paquetage_logiciel carte_acteur_robotique_mobile
interface_SRAM [interface_SRAS]
robot_pédagogique_mobile
- acteur_robotique_sédentaire —> paquetage_logiciel
carte_acteur_robotique_sédentaire interface_SRAS
robot_pédagogique_sédentaire
- acteur_gérant_de_communication —> paquetage_logiciel
carte_gérant_de_communication
interface_SRAS interface_SRAM

Un acteur robotique peut se connecter au réseau filaire, mais un acteur sédentaire n'a pas la possibilité d'accéder directement au SRAM. Il est obligé de passer par la passerelle.

La carte générique est caractérisée par un microprocesseur avec une architecture 16 ou 32 bits , une capacité RAM et ROM de 512Ko à 4 Mo par banque de 512 Ko et par de multiples entrées-sorties (contrôleurs d'interfaces parallèles de trois ports chacun, contrôleurs d'interface série full-duplex, et gérants de temps programmables permettant, par exemple, de gérer une horloge). Cette généralité est traduite par les règles suivantes :

- carte_générique —> CPU_16-32bits RAM_externe ROM_externe [E/S_externe]
[interface_spécifique]
- interface_spécifique —> interface_robotique | interface_communication |
autre_interface
- interface_robotique —> interface_robot_mobile | interface_robot_sédentaire

La carte générique peut être instanciée pour les besoins de notre projet en (figure II-2) :

- carte_acteur_matériel_de_base —> CPU_16-32bits EEPROM RAM_dynamique
E/S_externe
- carte_acteur_robotique_mobile —> CPU_16-32bits EEPROM RAM_statique
E/S_externe interface_robot_mobile
- carte_acteur_robotique_sédentaire —> CPU_16-32bits EEPROM RAM_dynamique
E/S_externe interface_robot_sédentaire
- carte_acteur_gestionnaire_de_communication —> CPU_16-32bits EEPROM
RAM_dynamique E/S_externe

Pour le robot sédentaire, la carte de contrôle peut être équipée de circuits RAM statiques ou dynamiques (avec un facteur d'économie de six, contre une perte de temps d'accès mémoire de 30%). Par contre, la carte de l'ARM est impérativement équipée de RAM statiques pour répondre à la contrainte de consommation associée à l'autonomie énergétique.

Le cas où la mémoire morte est optionnelle a été écarté. En effet, des processeurs comme ceux de la famille transputer possède une possibilité de démarrer en chargeant un programme par les liens série haut débit, ce qui évite d'avoir un programme en mémoire morte. Mais ce cas a été écarté pour les acteurs robotiques car cela nécessiterait une liaison filaire temporaire. Ce serait incompatible avec le critère d'autonomie des mobiles, et contraignant pour les autres types d'acteurs. La ROM est composée de mémoires de type EPROM pour le prototype,

mais les modèles standards peuvent être équipés de mémoires de type EEPROM ou FlashEPROM permettant de garder les mêmes principes physiques (électriques) pour la lecture et l'écriture, ce qui par conséquent assure une plus grande flexibilité de développement.

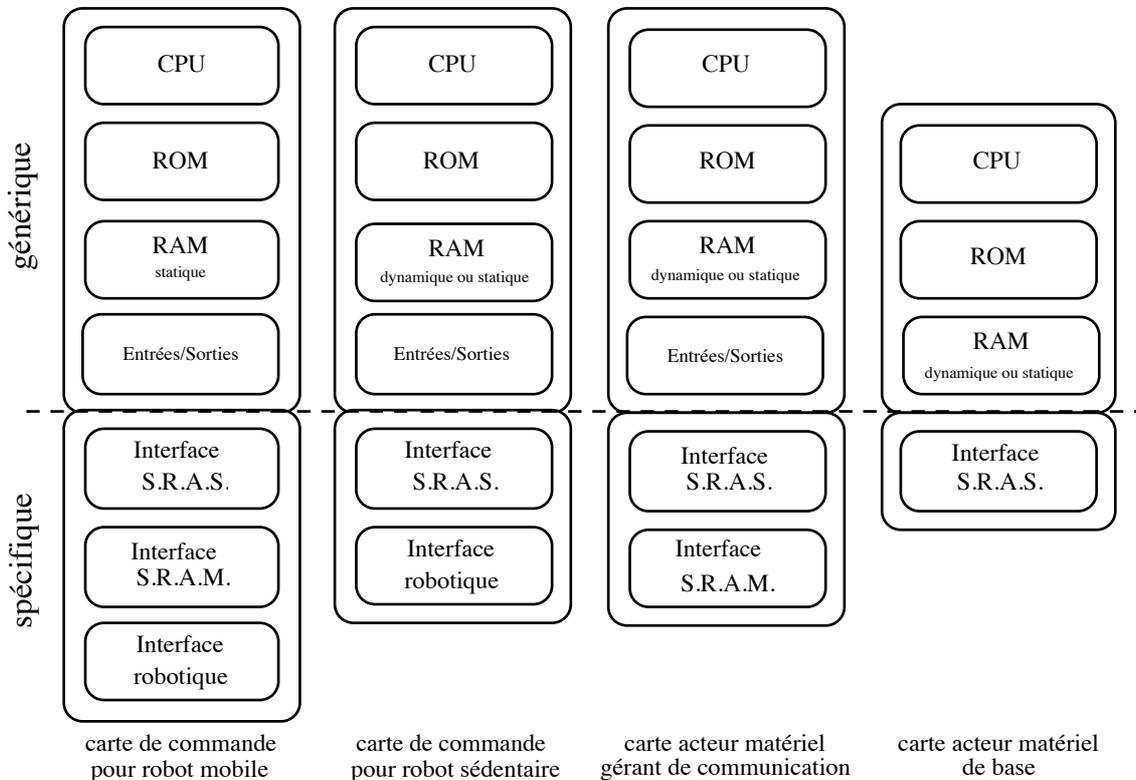


Figure II-2 : Spécialisation de la carte générique par type d'acteur

Un prototype de carte générique pour piloter un robot mobile a été réalisé (photographie II-1) sur lequel un interpréteur Méta-Vlisp a été implémenté.

En ce qui concerne les entrées-sorties, nous proposons deux architectures de contrôle :

- la première, conventionnelle,
- la seconde dont le principe est de faire gérer les entrées-sorties par un contrôleur spécialisé.

Le réseau étant constitué d'acteurs hétérogènes, l'expérience a montré qu'il est trop coûteux en temps et en argent de développer des cartes en fonction des robots pédagogiques à contrôler et des micro-processeurs à étudier, l'évolution technique étant trop rapide. Nous proposons donc la structure N° 2 présentée dans [Darche 90]. Il s'agit de séparer la partie traitement de la partie entrée-sortie et interfaces spécifiques, cette dernière étant gérée par un processeur spécialisé qui communique par une liaison rapide de type série ou parallèle avec l'unité centrale. Cette proposition se justifie par deux constats :

- Les entrée-sorties sont lentes par rapport à la vitesse des processeurs actuels et futurs. Il existe, certes, des mécanismes qui permettent d'accélérer celles-ci (interruptions, mode canal, etc.), mais ils ne sont pas suffisants.
- Les processeurs évoluent très rapidement. Si nous voulons faire évoluer l'architecture, il faut sans cesse redéfinir la partie entrée-sortie. Il faut donc par conséquent libérer la CPU de la gestion des entrées-sorties. Les entrées-sorties seront gérées par un contrôleur d'entrées-sorties spécifiques. Ceci est résumé par la figure II-3.



Photographie II-1 : Carte générique prototype

Cette deuxième architecture améliore la genericité et l'hétérogénéité : les cartes contrôlant les acteurs robotiques peuvent être différentes, seules les entrées-sorties restent inchangées. La flexibilité de l'architecture permet ainsi un prototypage matériel rapide.

Cette approche a été mise en oeuvre dans **[Barnes et al. 91]**. Un micro-contrôleur Intel 80C39 gère les entrées-sorties d'un robot de type hexapode, et dialogue via les liens avec un réseau de transputers **[Inmos 89]** montés sur des cartes au standard constructeur TRAM.

Nous terminons notre description par les règles qui caractérisent les composants de la carte générique :

- CPU_16-32bits → T414 | T425 | T800 | T805
- RAM_externe → RAM_statique | RAM_dynamique
- ROM_externe → EPROM | EEPROM | Flash_EEPROM

- E/S_externe → E/S_discrètes | processeur_E/S
- E/S_discrètes → contrôleur_interface_parallèle | contrôleur_interface_série
| contrôleur_gérant_de_temps
- processeur_E/S_ → CPU_8bits RAM ROM E/S_discrètes | micro-contrôleur
[RAM] [ROM] [E/S_discrètes]
- CPU_8bits → 680x | 80xx | Z80
- micro-contrôleur → 68HC05 | 68HC11 | 80Cxxx

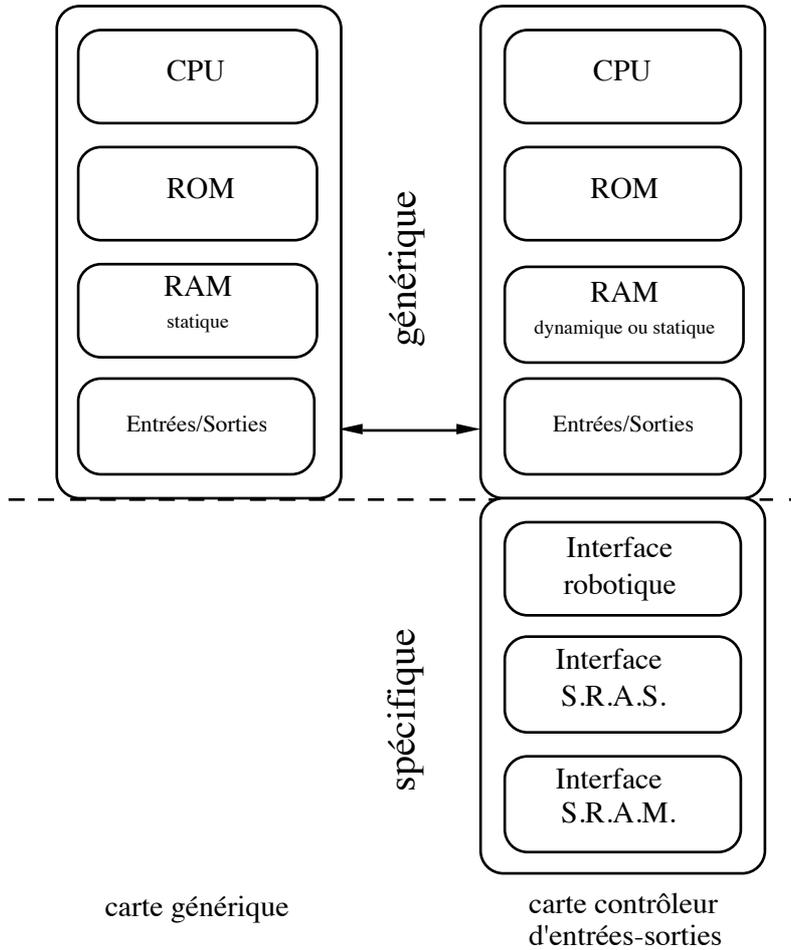


Figure II-3 : Deuxième proposition d'architecture

Grâce à cette approche, les nouveaux modèles de microprocesseurs comme le T9000 d'INMOS [**Inmos 91**] ou des microprocesseurs d'autres familles (Intel 80x86, Motorola 680x0, etc.), sont à considérer pour contrôler les cartes processeur avec un ajustement matériel et logiciel minimal.

4.2. Le choix du micro-processeur

Notre application peut être classée dans la catégorie des systèmes de type "temps réel", puisqu'elle devra réagir à des événements extérieurs, comme les impulsions d'une horloge

ou un changement d'état d'un capteur, par exemple un capteur de collision et ceci dans un temps suffisamment court de l'ordre de la milli-seconde.

La famille des transputers de la société INMOS a été sélectionnée dans notre application pour les caractéristiques suivantes :

- le transputer est un composant spécialement développé pour les architectures distribuées,
- l'architecture d'inspiration RISC laisse espérer des performances élevées, même en utilisant le langage évolué OCCAM,
- le quasi-parallélisme, grâce à l'enchaînement de tâches micro-programmées, facilite l'implémentation du modèle concurrent acteur. La gestion des processus (files, priorités, quantum de temps, changement de contexte, etc.) est extrêmement rapide,
- les quatre interfaces série rapides (de 5 à 20 Mbits/s) appelées liens, sont bien adaptées au réseau filaire, avec un coût minimal pour l'interface réseau car le protocole est intégré, et une liaison directe est possible (longueur maximale : cinq mètres sans amplificateur),
- la mémoire rapide intégrée de 2 à 4 Ko augmente la vitesse d'exécution en optimisant les accès mémoire,
- l'unité de calcul virgule flottante intégrée (pour le T800) améliore la puissance de calcul en accélérant le calcul local des coordonnées du robot (le bras dans l'espace et la position du mobile sur un plan),
- la technologie C-MOS permet une consommation en courant raisonnable pour un système embarqué et inférieure en tout cas à une solution en technologie bipolaire, en regard de la vitesse du processeur,
- l'initialisation, le téléchargement et la mise au point peuvent être mis en oeuvre avec flexibilité et efficacité. L'initialisation peut être faite par les liens ou par ROM et un signal de contrôle 'analyse' permet d'analyser la mémoire du système de manière post-mortem.

Notre choix s'est porté sur le circuit IMS T800 [**Inmos 89**] qui intègre une unité de calcul en virgule flottante, outil indispensable pour la connaissance en temps réel des coordonnées de la pince et des mobiles et sur l'IMS T425 (T800 sans coprocesseur arithmétique) pour les acteurs matériels de base et l'acteur passerelle.

Le transputer est souvent présenté comme un processeur utilisable pour les applications temps réel mais sa non prédisposition à gérer les interruptions l'écarte souvent de celles-ci. Il ne dispose, en effet, que d'une seule entrée pour les événements extérieurs. Nous proposons ici plusieurs solutions matérielles et logicielles.

4.3. Analyse d'une interruption

Une interruption est un signal logiciel ou matériel qui permet d'interrompre un processeur et d'effectuer un traitement par un processus spécialisé.

Les interruptions logicielles sont des instructions spécifiques qui interrompent le processeur (par exemple Int pour le 80X86). Une interruption matérielle est un signal électrique logique extérieur au processeur qui interrompt celui-ci. Un signal électrique logique se caractérise par deux fronts, montant et descendant, et par deux états : haut et bas. Un déclenchement d'interruption s'effectue sur une des caractéristiques précédentes du signal ou sur une combinaison logique de ces caractéristiques. Lors d'un déclenchement sur état, plusieurs opérations sont exécutées :

- le signalement,
- la prise en compte selon une loi de priorité et le déroutement vers une routine dite d'interruption,
- la fin de l'interruption.

Il peut y avoir perte d'interruption lorsqu'une interruption a été signalée et prise ou non en compte et qu'une ou plusieurs autres arrivent ensuite. Une détection de perte doit être mise en oeuvre, avec mise en place d'un compteur indiquant le nombre d'interruptions perdues pendant l'interruption courante. Lorsqu'il existe plusieurs entrées d'interruption, différentes politiques de gestion peuvent être mises en oeuvre : fixe, tournante, FIFO, LIFO, et aléatoire.

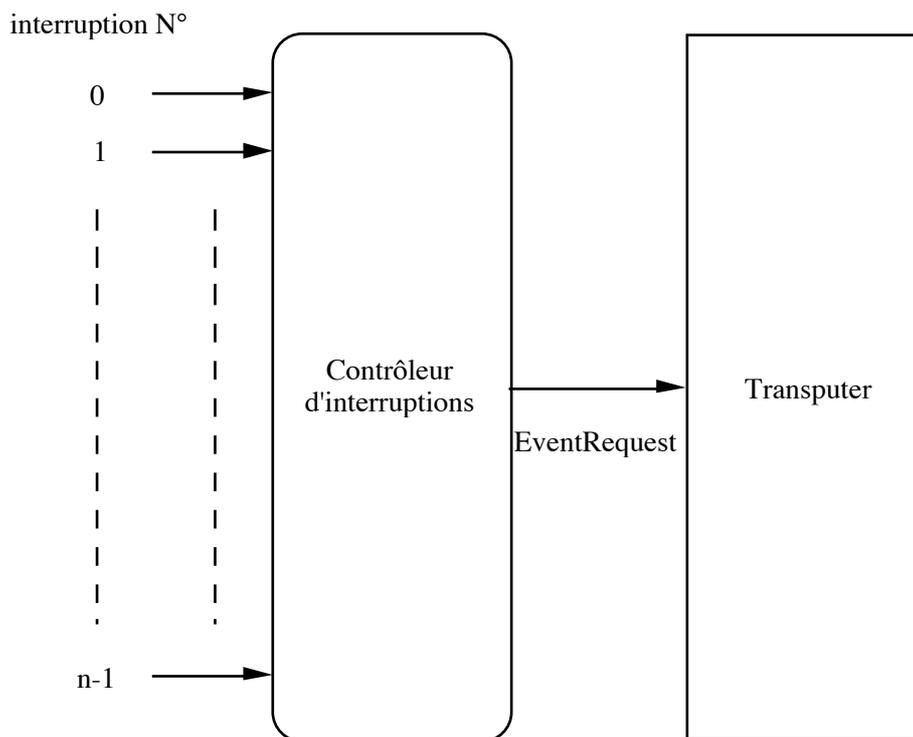


Figure II-4 : Gestion des interruptions

La famille transputer dispose de trois signaux qui sont utilisables pour les interruptions :

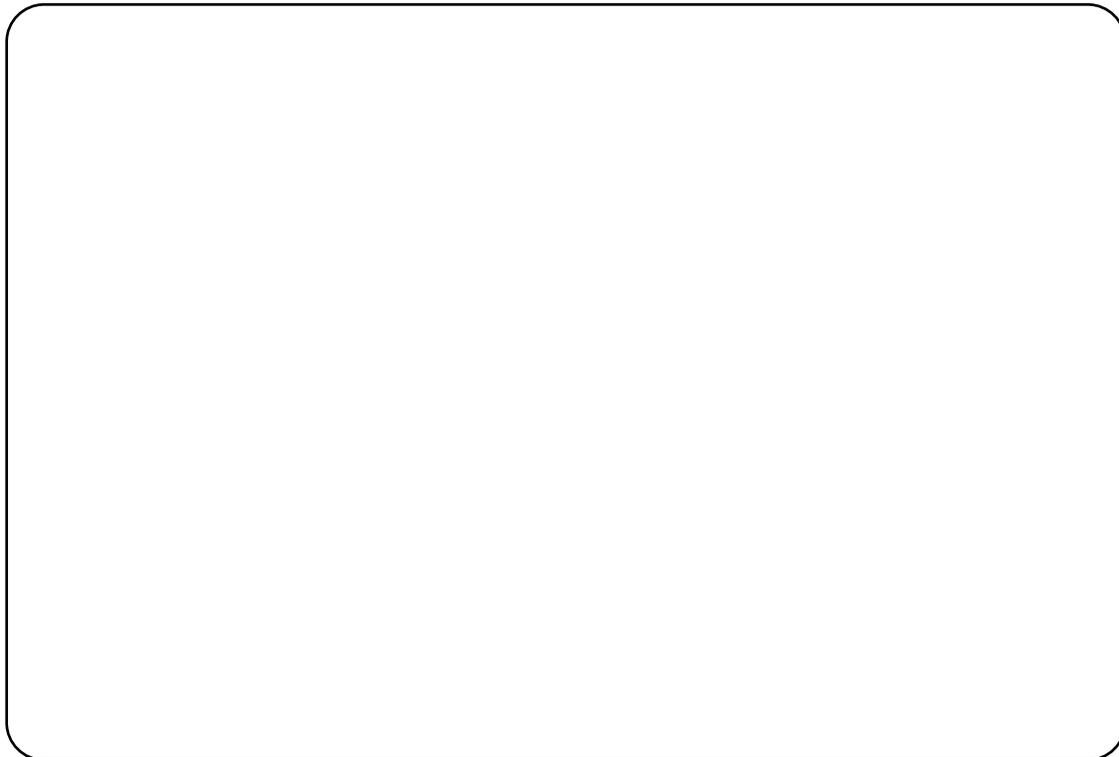
- EventRequest,
- EventAck,
- et EventWaiting.

Une tâche se met en attente d'une interruption (signal EventWaiting), le gestionnaire d'interruptions avertit le processeur qu'une interruption est demandée (signal EventRequest), le processeur la traite en envoyant un signal EventAck et récupère le numéro d'interruption. Une seule interruption peut donc être prise en compte (active sur un front montant). Il n'y a aucune gestion des pertes d'interruption, il faut donc que la fréquence des événements externes soit inférieure à celle du traitement. Pour traiter plusieurs interruptions, il faut intercaler un gestionnaire d'interruption. Celui-ci peut être réalisé en logique discrète (PAL, GAL (Generic Array logic), etc.) ou par un circuit spécialisé (ASIC ou autre).

Notre choix s'est porté sur un gestionnaire d'interruptions classique, le 8259 de la société Intel pour bénéficier de sa bibliothèque de programmes de gestion. Deux contrôleurs ont été mis en cascade pour offrir quinze entrées d'interruption, de IRQ0 (la plus prioritaire) à IRQ14 (figure II-4).

5. Le robot mobile "Tatou"

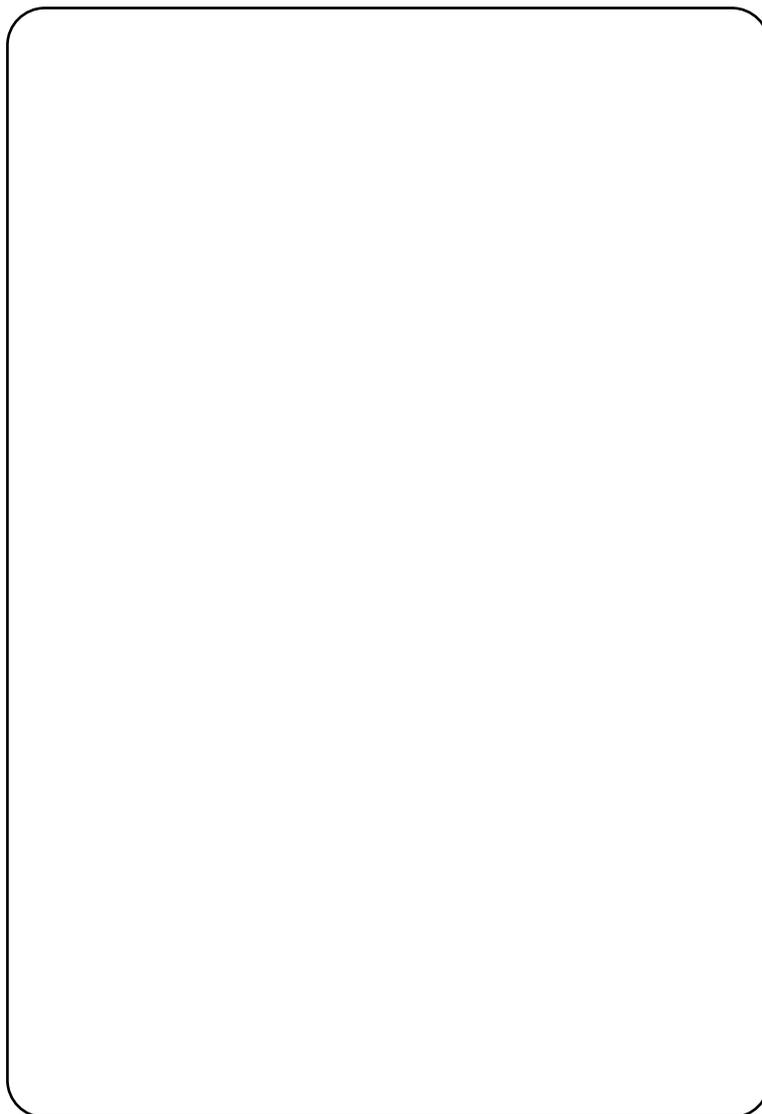
Pour valider le concept d'acteur robotique mobile, un robot mobile nommé 'Tatou' a été développé au laboratoire (photographies II-2 et II-3). C'est un véhicule holonome à rayon de giration non borné. Ce premier prototype est un acteur sériel, c'est-à-dire que son unité de calcul ne traite qu'un seul message à la fois.



Photographie II-2 : Robot mobile - vue de face

Il possède les fonctions suivantes :

- Communication.
- Transport : une plate-forme s'adaptant sur le haut du mobile permet de transporter de petits objets. Ce plateau, en fonction d'une commande, peut monter ou descendre de huit cm.
- Impression : un stylo monté au centre géométrique du mobile permet de dessiner au sol, ce qui reproduit les fonctionnalités de la tortue de sol de type Logo.
- Détection des collisions.



Photographie II-3 : Robot mobile - vue de côté

L'autonomie n'est pas utilisée au sens classique de la robotique mobile. En effet, le robot ne peut pas pour l'instant être autonome au niveau comportement dans un environnement connu ou inconnu. Nous redéfinissons donc cette notion dans ActNet en ne considérant l'autonomie que du point de vue de la communication et de l'énergie. Le robot mobile par le

SRAM de type radio-fréquence n'est pas relié par un câble au réseau et est donc libre de ses mouvements. Il embarque sa propre énergie, en l'occurrence une batterie, qui permet une autonomie énergétique réelle mais limitée dans le temps.

5.1. Méthode de description

Pour décrire physiquement le robot mobile, nous avons opté pour la représentation objet. Cinq niveaux ont été définis : les niveaux composant élémentaire mécanique ou électronique, objet, module fonctionnel, sous-groupe fonctionnel et groupe fonctionnel. Nous sommes par contre restés lors de la décomposition au niveau objet; le niveau inférieur, c'est-à-dire le niveau "composant élémentaire", est traité dans les rapports techniques de chaque objet.

Un groupe ou sous-groupe fonctionnel rassemble plusieurs modules fonctionnels. Un module fonctionnel remplit une ou plusieurs fonctions bien précises; il peut s'agir d'une fonction (impression, télécommunication, etc.) ou d'un but à atteindre (poursuite, etc.). Un module fonctionnel comporte en général trois sous-ensembles : le sous-ensemble mécanique, le sous-ensemble électro-mécanique et le sous-ensemble électronique. Un sous-ensemble est une collection d'objets composés de composants élémentaires. Pour un objet mécanique, il s'agira de composants standards (roulement à bille, boulon, etc.) ou de composants usinés. Les objets électroniques seront composés d'un circuit imprimé où des composants passifs (résistance, condensateur, self, etc.) et des composants actifs (transistor, circuit intégré, etc.) seront soudés. Chaque objet du système est repéré par un identificateur unique appelé `identificateur_objet` dont la structure est la suivante :

(`Id_groupe`, `Id_sous-groupe`, `Id_module fonctionnel`, `Id_sous-ensemble`, `Id_objet`)

Les objets logiciels sont des logiciels pilotes actifs ou acteur. Lorsque des périphériques matériels sont associés, nous parlons d'acteur matériel. Nous n'avons donc volontairement pas créé un sous-ensemble logiciel qui se serait chargé de contrôler les sous-ensembles matériels, comme cela aurait été le cas dans une description classique, fonctionnelle ou systémique. Il existe une bijection entre le module fonctionnel et l'acteur logiciel.

5.2. Caractéristiques mécaniques générales

Le robot est de forme cylindrique, d'un diamètre de 33 cm et d'une hauteur de 16 cm. L'axe des roues motrices passe par le centre de la plaque de base, ce qui permet de dissocier les fonctions 'déplacement' et 'rotation'.

La vitesse du mobile a été fixée à 20 m/mn soit 0,33 m/s, pour une pente maximale de 10 % (5,7°); ce choix a été effectué pour prendre en compte le fait que des chocs éventuels trop violents pourraient endommager la cloche de protection du mobile.

Son poids est de 10 kg, 14 kg avec la batterie.

La garde au sol est de 1 cm. La plaque de base a quatre points d'appui au sol : les deux roues et deux billes de manutention montées à l'envers. La présence de deux billes porteuses de type manutention rend le système mécanique hyperstatique, mais elle limite l'écrasement des pneus en caoutchouc, tout en permettant une adhérence optimale de ceux-ci.

Le système de motorisation est composé de deux moteurs électriques pas à pas et d'un système de réduction à courroie crantée. L'utilisation de ce type de moteur permet d'avoir une commande dite "en boucle ouverte" sans fermeture de la boucle d'asservissement par un capteur de position axiale. Cette solution simplifie la commande et l'électronique. La solution réduction courroie crantée permet d'offrir un jeu minimal au niveau de la transmission tout en restant simple de mise en oeuvre. Une réduction était nécessaire car le moteur ne présentait par le couple nécessaire pour mouvoir l'ensemble. Une rondelle ondulée rattrape un jeu axial éventuel dans le cas d'une dispersion des caractéristiques de fabrication des roulements à billes ou des pièces mécaniques supports.

Un stylo en son centre géométrique permet d'écrire au sol afin de reproduire la fonction impression de la "tortue Logo".

Pour les deux premiers prototypes, la coque a été réalisée en fibre de verre imprégnée de résine. Pour la version définitive, nous avons choisi un plastique technique résistant aux chocs puisque l'appréhension du monde extérieur se fait par contact, et transparent pour l'esthétique. Il s'agit du polycarbonate, qui est utilisé notamment pour les pare-chocs de voiture.

5.3. Constitution générale

Le tatou est constitué d'un ensemble de modules fonctionnels classés par groupes et éventuellement par sous-groupes fonctionnels. En rapport avec les missions que doit remplir le robot mobile et les fonctionnalités de celui-ci, nous avons identifié sept groupes fonctionnels :

- le groupe contrôle,
- le groupe entrée/sortie,
- le groupe propulsion,
- le groupe capteurs,
- le groupe impression,
- le groupe énergie,
- le groupe transport.

Après avoir défini l'ensemble des groupes intervenant dans la constitution du robot mobile, détaillons maintenant ceux-ci dans les sections suivantes.

5.4. Groupe Contrôle

La réalisation a été prévue pour s'adapter aux deux solutions pré-citées concernant l'architecture matérielle. La carte générique a été scindée en trois cartes de type mezzanine.

Les cartes filles ou module “piggy-back” ROM et E/S viennent se superposer sur la carte processeur (figure II-5).

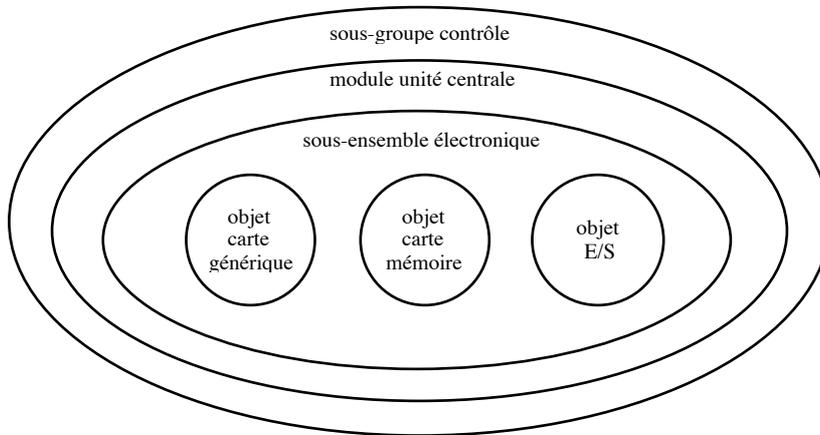
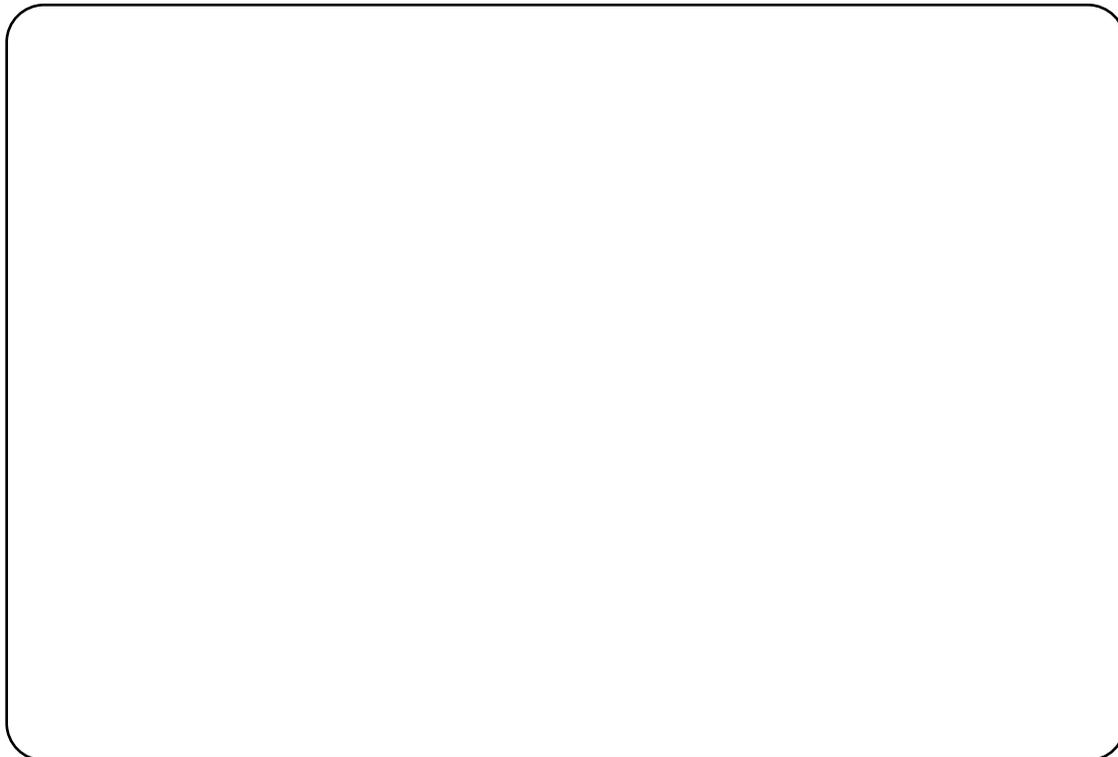


Figure II-5 : Groupe contrôle



Photographie II-4 : Carte générique nue

La carte générique (photographie II-4) se compose d'un transputer T800 ou T425, de 4 Mo de RAM statique et des circuits spécialisés de gestion du bus qui relie toutes les cartes.

La logique de contrôle de la carte a été développée avec des circuits logiques programmables, en l'occurrence des GALs. Un bus composé de 108 signaux répartis sur deux connecteurs permet de relier la carte système aux cartes mémoire morte et entrées-sorties. Un signal d'initialisation (Reset) est nécessaire au démarrage du système. Un circuit spécialisé fournit ce signal dès que la tension d'alimentation est dans sa plage nominale. Associé à ce signal, comme pour tout système embarqué, un système de garde (Watchdog) veille au bon fonctionnement du système. Ce 'chien de garde' est réarmé par le système périodiquement. En cas de défaillance du réarmement, une initialisation est donc démarrée. Un contrôle de la tension d'alimentation de la carte système permet de lancer une initialisation si la plage de tension d'alimentation n'est pas respectée. Périodiquement, le logiciel doit réarmer le chien de garde pour prouver que le logiciel de contrôle a le comportement prévu et n'est pas dans une boucle infinie. Dans le cas contraire, un signal d'initialisation est généré.

La carte mémoire morte est composée de 4 Mo de mémoire morte de type EPROM ou EEPROM. Les composants EEPROM ont été sélectionnés pour faciliter le développement. La tension de programmation est de 5 volts, ce qui évite la tension de programmation de 12,5 volts ou de 21 volts. L'effacement se fait électriquement, donc il est inutile d'effacer par exposition aux rayons ultra-violet, ce qui évite des manipulations de la part de l'utilisateur.

5.5. Groupe Entrée-sortie

Le groupe entrée-sortie (figure II-6) est chargé de mettre en relation la partie système avec le monde extérieur. Il se décompose en deux sous-groupes : le sous-groupe contrôleur E/S et le sous-groupe communication. Les termes entrée et sortie caractérisent le flux de données.

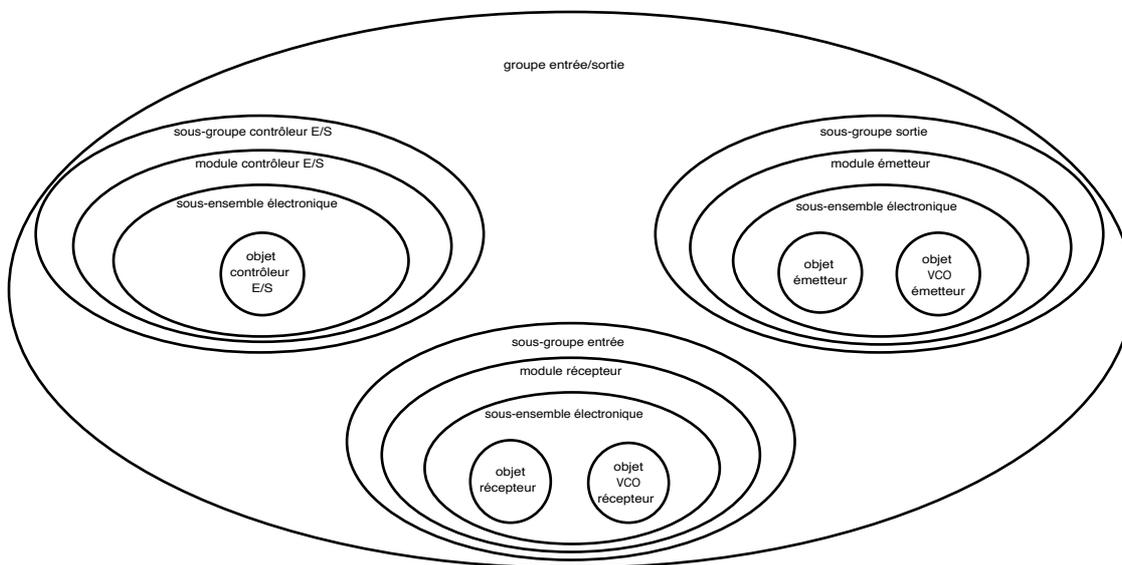


Figure II-6 : Groupe entrée-sortie

Les modules émetteur et récepteur sont décrits dans le chapitre communication.

Nous détaillons par les figures II-7 à II-9 les entrées-sorties. Un module composé de quatre circuits timer gère le temps. Quatre interfaces série compatibles RS232-C permettent au robot de communiquer avec l'extérieur. Un dernier module gère les entrées-sorties parallèles.

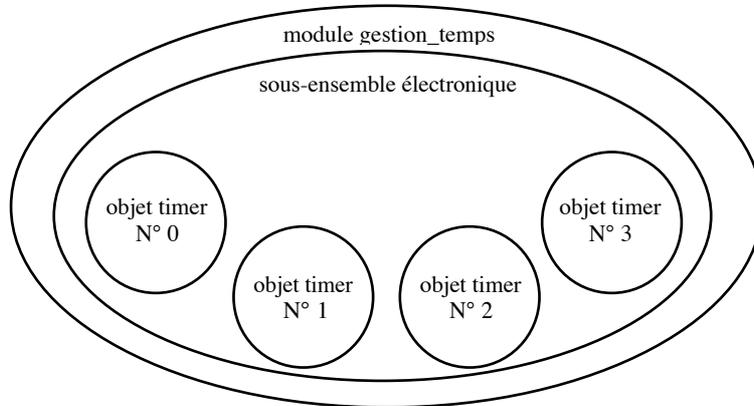


Figure II-7 : Module gestion du temps

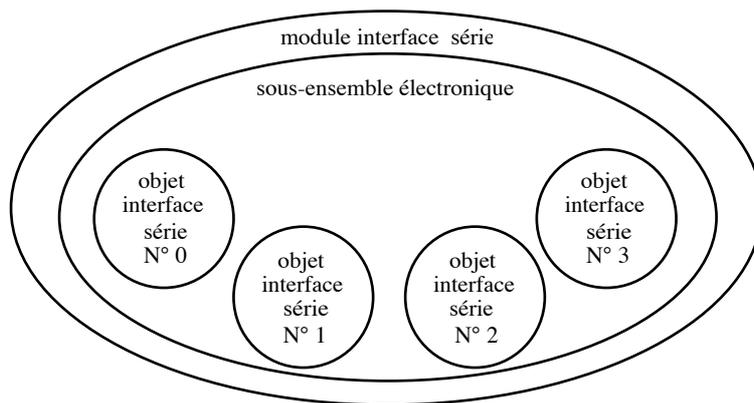


Figure II-8 : Module interface série

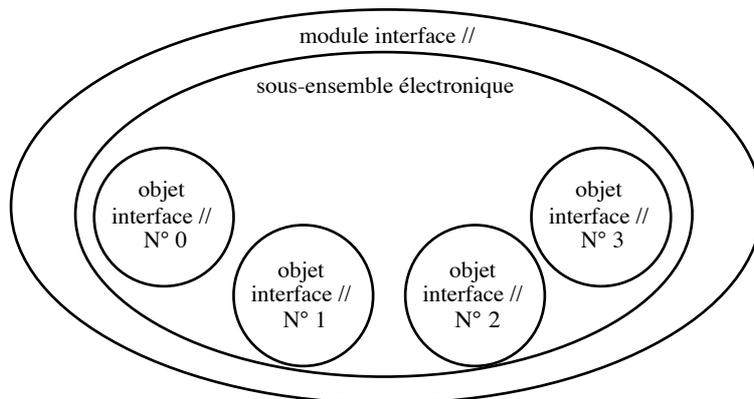
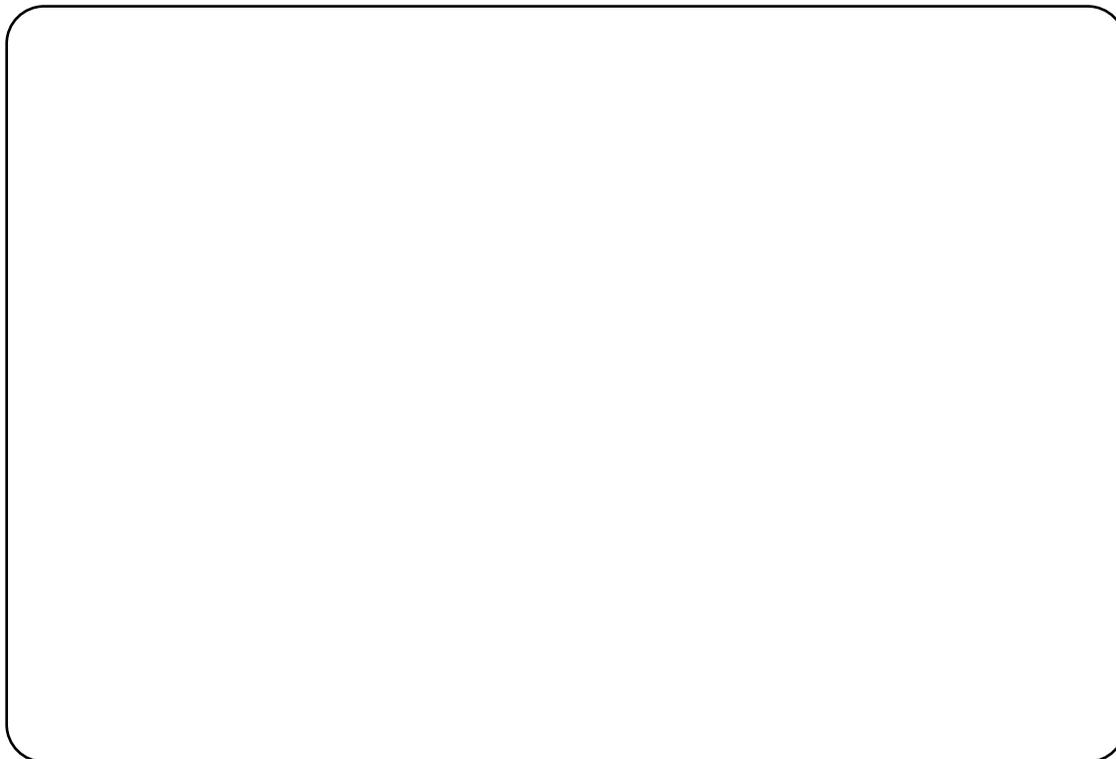


Figure II-9 : Module interface parallèle

La carte d'entrée-sortie peut être remplacée par une carte contrôleur d'entrée-sortie qui communique avec la carte de contrôle générique par une interface série ou une interface parallèle ou d'un autre type.

5.6. Groupe Propulsion

Au niveau motorisation électrique, nous avons le choix entre le moteur à courant continu et le moteur pas à pas. Le premier nécessite une commande très simplifiée se résumant à un signal marche-arrêt. Par contre, pour connaître la position exacte de l'axe, la vitesse, et par intégration la distance parcourue, il nous fallait utiliser un codeur ou un capteur de vitesse, ce qui compliquait l'architecture matérielle et augmentait la consommation par ajout d'une électronique supplémentaire. La deuxième solution, que nous avons retenue, permet, en compliquant la commande et en prenant quelques précautions au niveau de la commande pour éviter des pertes de pas ou des glissements, de supprimer la présence d'un capteur de position.



Photographie II-5 : Carte contrôleur moteur pas à pas

Le groupe propulsion (figure II-10) est donc composé de deux modules motorisation, eux-mêmes constitués des sous-ensembles mécaniques. Ces sous-ensembles sont composés principalement des roues, du système de transmission, électro-mécanique, de l'objet moteur pas à pas qui assurent la motorisation, du sous-ensemble électronique composé des objets contrôleur électronique associés (photographie II-5).

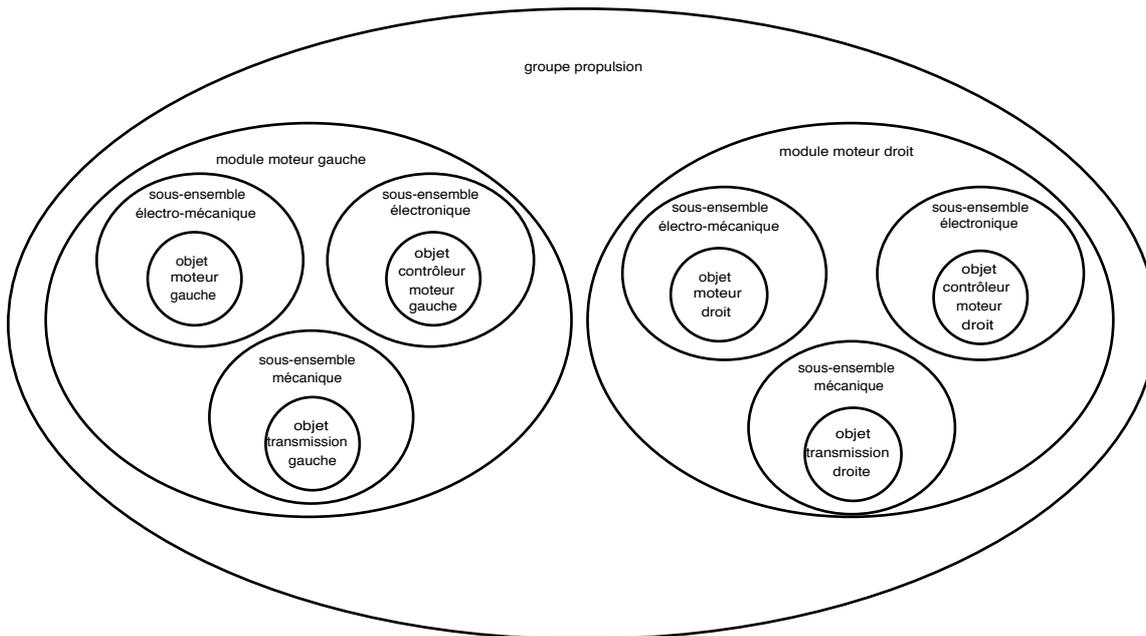


Figure II-10 : Groupe propulsion

Nous avons utilisé un circuit spécialisé (L6223) dans la gestion des moteurs pas à pas, reposant sur la technique à découpage de la société SGS Thomson Microelectronics. Chaque phase du moteur est alimentée en 12 volts et une limitation en courant/tension est mise en oeuvre. Ce principe permet d'augmenter le temps de montée des signaux de contrôle pour une meilleure efficacité de la commande.

Le module moteur se caractérise par un vecteur d'état décrivant :

- son état interne: mode actif ou mode veille ou non alimenté, rotation ou arrêt,
- son sens de rotation (horaire ou anti-horaire),
- la position de l'axe par rapport à une position initiale (position angulaire),
- la consigne vitesse de rotation,
- la consigne accélération.

Ces informations seront reprises par l'acteur logiciel associé.

5.7. Groupe Capteurs

Les capteurs permettent d'appréhender l'environnement interne et externe du robot. Par rapport à la classification de [Coiffet 86] (partie II.2.1.), nous englobons les capteurs de sécurité dans la classe des capteurs internes, car ils renseignent sur des états internes extrêmes. Les capteurs peuvent fournir une information analogique ou tout ou rien. L'information délivrée par les capteurs analogiques doit généralement être convertie en une donnée physique représentant la grandeur physique à appréhender. Cette information est incertaine et doit être en général interprétée. Les capteurs se subdivisent en deux groupes : les capteurs simples et les capteurs intelligents. Avec le capteur simple, l'interprétation est faite

au niveau du système de manière centralisée. Pour l'autre type, une intelligence intégrée dans un contrôleur effectue le traitement du signal (conversion, filtrage) localement et peut gérer l'interconnexion à un réseau de communication.

Le groupe capteur est composé du sous-groupe fonctionnel détection de collision proche. Il existe aussi quatre sous-groupes optionnels (figure II-11) qui sont :

- le sous-groupe mesure de distance,
- le sous-groupe détection humaine,
- le sous-groupe détection différentielle de lumière,
- le sous-groupe vision.

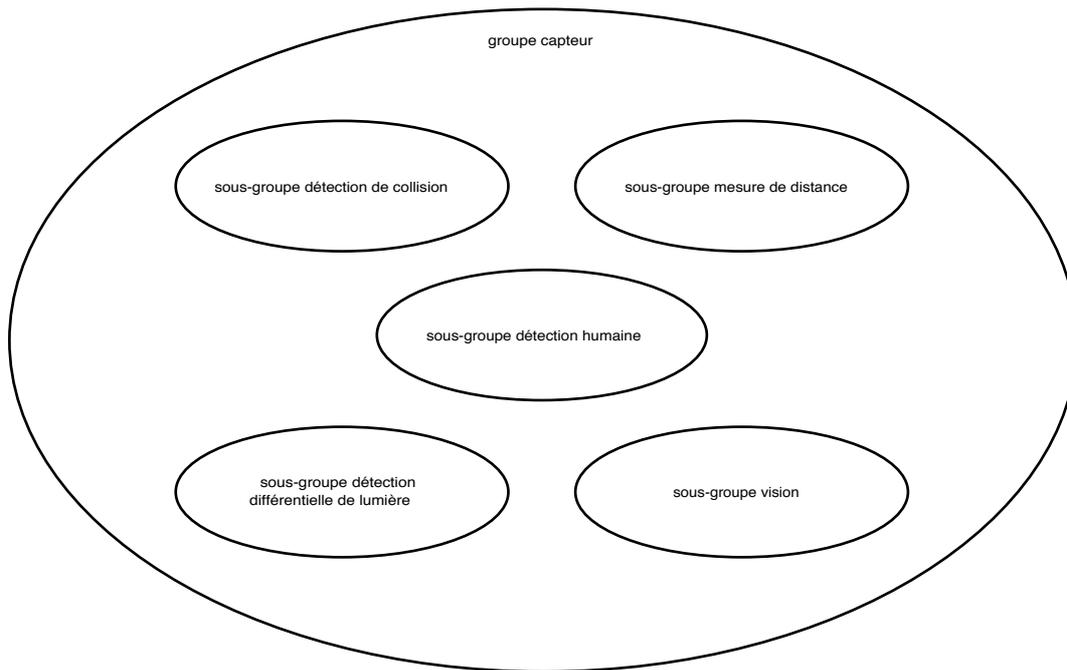


Figure II-11 : Groupe capteur

Le robot mobile est équipé de capteurs de collision. L'ensemble de la surface externe du robot est sensible par basculement de l'enveloppe de protection [Tanguy 87]. En effet, la cloche peut pivoter lors de collisions éventuelles avec le décor ou d'autres acteurs et une information logique est renvoyée à l'acteur. Ce sous-groupe se compose de six capteurs extéroceptifs répartis uniformément sur la périphérie de la base du robot (tous les 60°). Par recoupement d'informations provenant des capteurs, nous disposons de 12 zones angulaires de détection de 30° chacune.

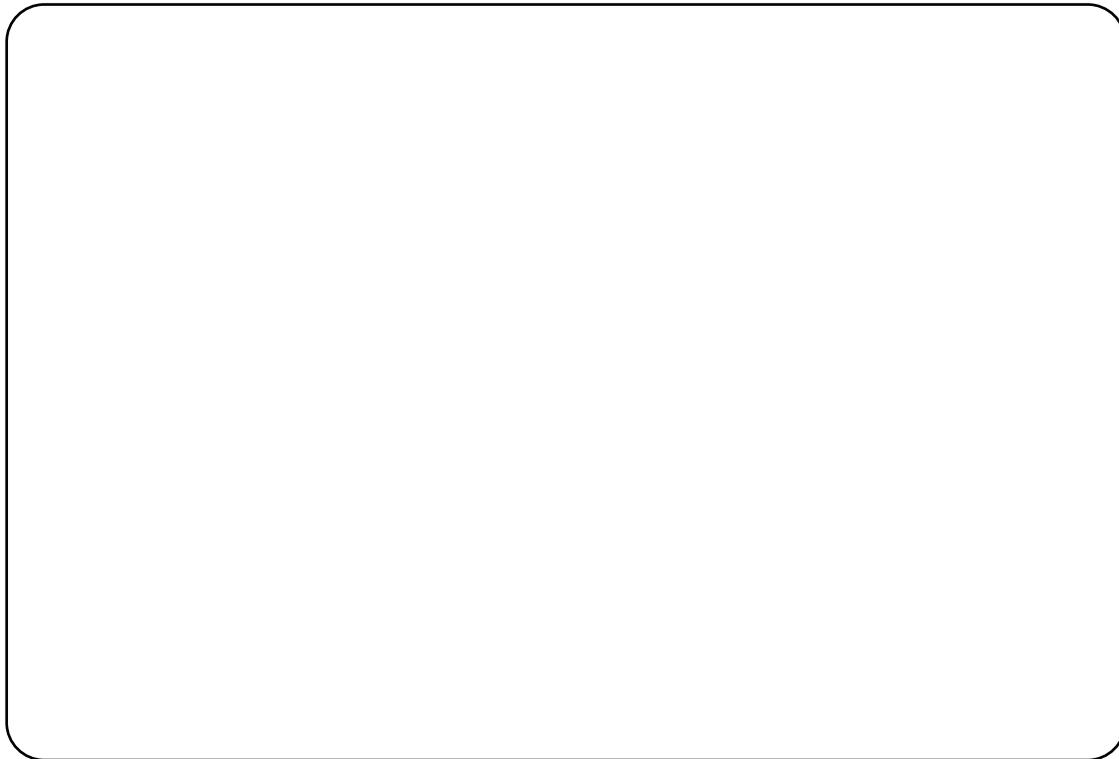
Le composant de base du capteur est un générateur de tension Hall qui est un élément sensible à une intensité de champ magnétique ou à une variation de celle-ci. Une chaîne de traitement composée d'un amplificateur de signal, d'un trigger de Schmitt et d'un transistor émetteur commun traite le signal et fournit un signal logique de type collecteur ouvert qui est envoyé sur le bus. Cette solution permet d'éviter des capteurs à contact mécanique. Les

contacts mécaniques ont une durée de vie limitée et présentent des phénomènes de rebondissement à la commutation nécessitant des circuits électroniques de filtrage à base d'une bascule RS ou d'un trigger de Schmitt. Nous avons choisi un bus parallèle pour relier ceux-ci.

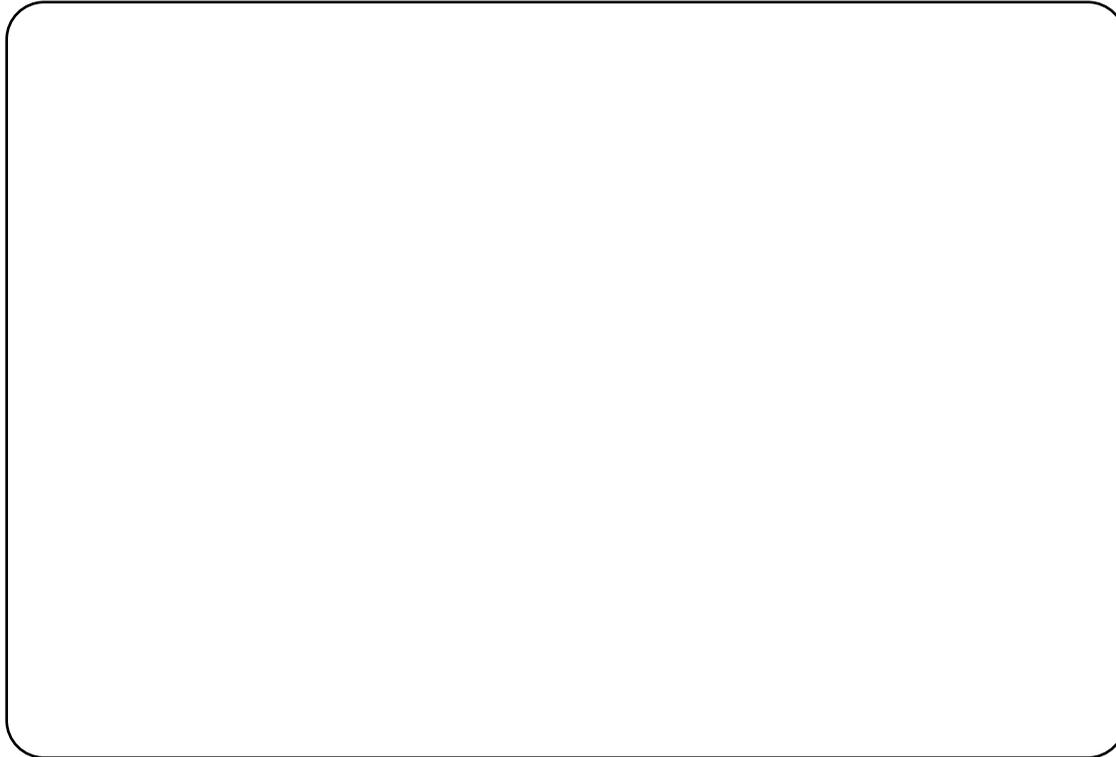
Dans le cas de non utilisation de la cloche, nous utilisons des capteurs infra-rouges qui sont sensibles à la réflexion. La source infra-rouge est modulée, et une détection synchrone est effectuée, ce qui permet de rendre le capteur insensible aux rayonnements parasites.

En résumé, le sous-groupe capteur de collision se compose des six modules fonctionnels, de module capteur_collision_0 à module capteur_collision_5.

La photographie II-6 représente le côté composant du capteur de collision et la photographie suivante, le capteur à effet hall côté soudure en situation sur le robot mobile.



Photographie II-6 : Carte capteur de collision



Photographie II-7 : Capteur de collision en situation

5.8. Groupe Impression

Le groupe impression est composé du module imprimeur qui se trouve au centre géométrique du robot, et qui permet de dessiner au sol. Le module imprimeur se compose du sous-ensemble mécanique, du sous-ensemble électro-mécanique composé d'un servo-mécanisme fonctionnant en mode tout ou rien et du sous-ensemble électronique de contrôle (figure II-12, photographie II-8). Par une commande binaire, le servo-mécanisme actionne un stylo qui monte ou qui descend.

Le sous-ensemble mécanique se compose d'un stylo de forme standard, d'un support mécanique porte-stylo qui est déplacé grâce à une pièce de levage couplée mécaniquement au système de levage. Deux solutions étaient envisageables pour maintenir une pression suffisante de la pointe du stylo sur le sol :

- le poids du stylo seul ou muni d'un poids supplémentaire permettant une pression suffisante au sol,
- un ressort maintenant une pression suffisante au sol.

La deuxième solution présente un avantage indéniable. La pression est supérieure, sans rajouter de poids supplémentaire à l'ensemble.

Nous avons utilisé un servo-mécanisme pour lever le porte-stylo. Le système mécanique est irréversible, ce qui permet de couper l'alimentation du système en position de repos.

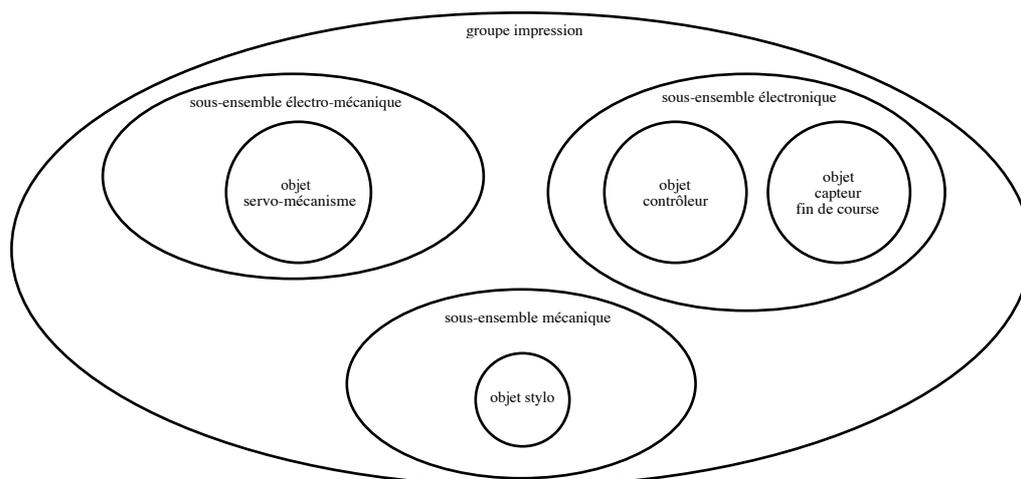


Figure II-12 : Groupe impression



Photographie II-8 : Groupe impression

Deux types de commandes sont envisageables (figure II-13) : proportionnelle et tout-ou-rien. Pour comparer, il faut regarder la commande et le coût. La première solution nécessite une commande de type impulsionnelle et est plus complexe à gérer : le régime impulsionnel

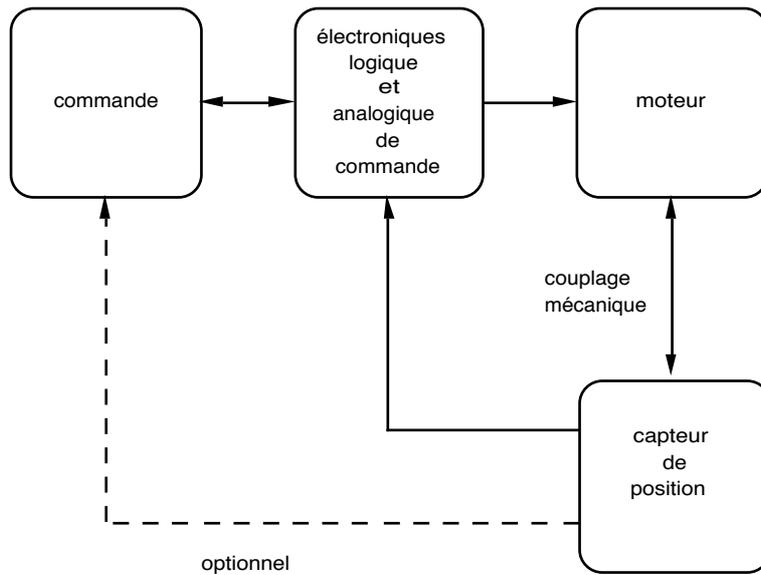
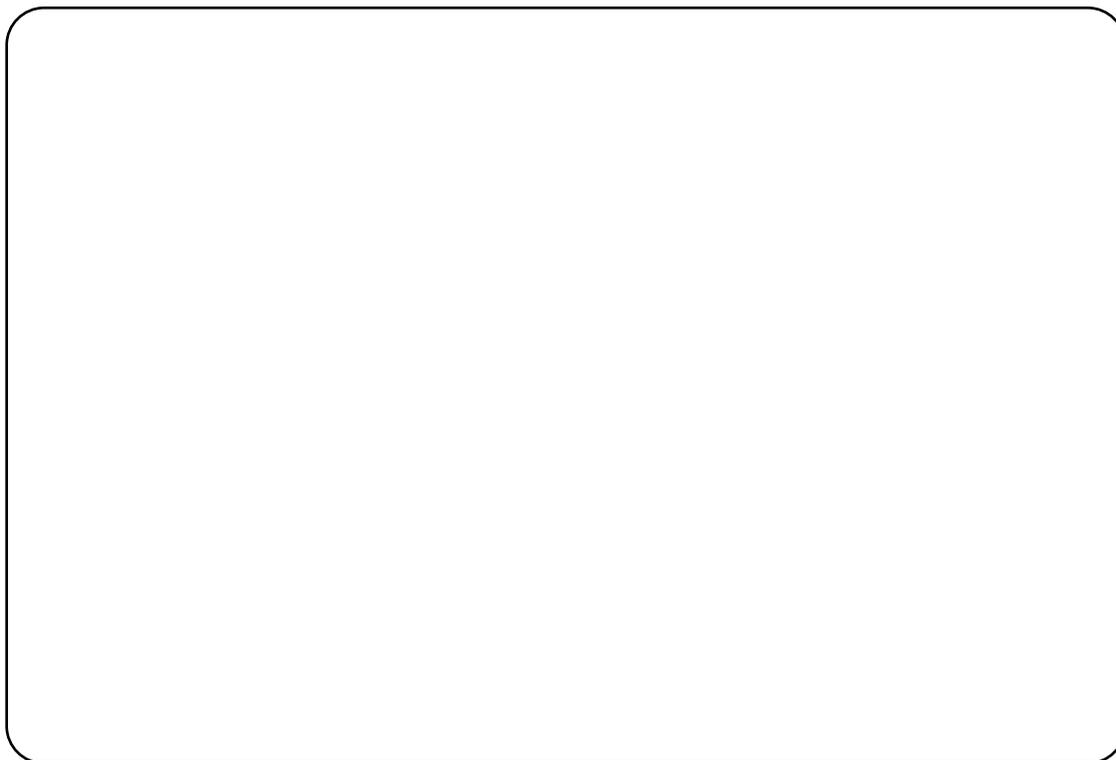


Figure II-13 : Asservissement



Photographie II-9 : Carte contrôleur stylo

lors de la mise en route pose problème, car le stylo bouge de sa position, de plus en cas de non alimentation ou d'une déconnexion côté commande, le système risque d'avoir un comportement aléatoire. De plus, les capteurs de fin de course ne sont pas en général intégrés dans l'électronique de commande. Par contre, ce type de commande est très répandu.

Le deuxième type de commande est plus simple au niveau conception. Si le circuit de commande n'est plus connecté ou en cas de déconnexion, la protection est plus simple car combinatoire. Le coût est par contre plus élevé du fait d'un développement spécifique. La deuxième solution a été retenue pour la simplicité de la commande et un comportement prévisible en cas de déconnexion. Ce servo-mécanisme n'est donc pas équipé de son électronique d'asservissement classique, puisque le système fonctionne en tout ou rien. Il n'y a pas de position intermédiaire. Des éléments électroniques détecteurs de fin de course du même type que le capteur de collision assurent ce type de fonctionnement en coupant l'alimentation du moteur dès que les positions haute et basse sont détectées, par souci d'économie d'énergie (photographie II-10). La position du stylo peut être déterminée éventuellement avec précision en utilisant le potentiomètre à loi ohmique linéaire qui est couplé mécaniquement avec l'axe du moteur. Un système de contrôle en logique discrète (photographie II-9), facilement compilable dans un circuit de logique programmable, gère l'automatisme. En cas de déconnexion du port d'entrée-sortie ou des capteurs de fin de course, le système soit revient en position de repos (position haute du crayon), soit coupe l'alimentation du servo-mécanisme. Un signal d'erreur est alors envoyé au système de commande.



Photographie II-10 : Carte détecteur de fins de course stylo

Nous avons opté pour des capteurs à effet Hall. Un aimant collé sur une des deux crémaillères renseigne les capteurs de fin de course de sa position. Lorsque l'aimant est en

bout de course, le système de contrôle est informé et le moteur est court-circuité afin qu'il soit freiné efficacement.

5.9. Groupe Energie

Depuis l'avènement des ordinateurs portables, une multitude de circuits spécialisés dans la gestion de l'énergie embarquée ont été développés : régulateur faible chute de tension, régulateur/convertisseur à découpage haut rendement, circuit de supervision d'alimentation, et chargeur de batterie.

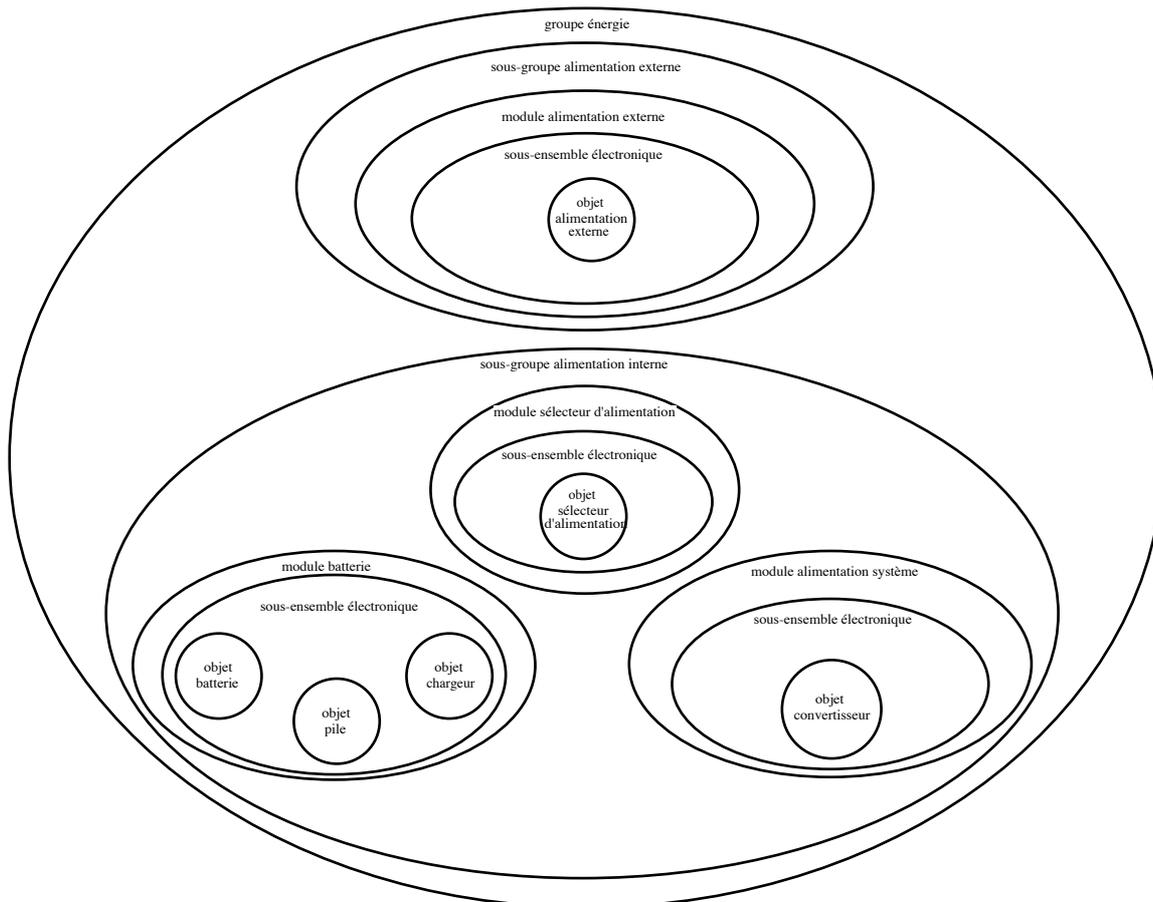


Figure II-14 : Groupe alimentation

Mais un système électronique embarqué est un tout indissociable, et rien ne sert d'optimiser la gestion de l'énergie embarquée si les autres modules électroniques consommateurs ne prennent pas en compte le critère de minimisation de la consommation. Les technologies des composants évoluent (Bipolaire, C-MOS, BIC-MOS, etc.) et de nouveaux mécanismes sont mis en oeuvre comme la mise hors tension ou le mode veille de certains éléments électroniques, ou reconception du circuit intégré pour, par exemple, les amplificateurs opérationnels "micro-power". Nous avons intégré les derniers progrès dans la conception des objets des autres sous-ensembles électroniques.

Trois modules internes et un module externe gèrent l'énergie embarquée du robot (figure II-14). Les modules internes sont constitués du module batterie, du module alimentation système et du module sélecteur d'alimentation.

5.9.1. Sources d'énergie

5.9.1.1. Sources régulées externe et interne

Une source d'alimentation régulée se caractérise par sa plage de fonctionnement en tension : tension nominale, tensions extrêmes (seuils de sur-tension et sous-tension) et sa plage de fonctionnement en courant (courants minimum et maximum qu'elle est susceptible de fournir). En cas de sur-tension ou en cas de sous-tension, un dispositif matériel doit déconnecter le système. Cette surveillance est dédiée au système de contrôle. Lorsque la cause de dysfonctionnement a disparu, le système doit être réinitialisé.

5.9.1.2. Source externe

La source externe est une alimentation tension continue non régulée de 20 volts. La source interne est une batterie qui sert à alimenter le robot dans sa phase autonome. A cause d'une consommation élevée due aux moteurs et aux cartes systèmes, l'utilisation de piles était exclue. Lorsque la batterie est déchargée, l'alimentation externe, si elle est connectée, prend la relève et permet ainsi au robot de fonctionner. De plus, un chargeur embarqué s'alimentant via l'alimentation externe recharge le robot en cinq heures environ. L'horloge temps réel et la mémoire statique C-MOS contenant les paramètres systèmes nécessitent une alimentation permanente afin d'assurer son fonctionnement lors de l'arrêt du robot. Une batterie avec un convertisseur DC/DC pourrait alimenter ce dispositif, mais le risque de décharge complète fait que la présence d'une source auxiliaire, ici une pile, s'impose.

5.9.1.3. Sources internes

L'énergie embarquée se compose d'une batterie (source interne) et d'une pile (source auxiliaire).

Une batterie se définit par ses caractéristiques mécaniques (dimensions, type de raccordement électrique, cosse ou fils), ses caractéristiques électriques (tension, capacité en Ah, courbes de charge et de décharge) et des caractéristiques diverses (technologie, durée de vie, gamme de température, etc.). La technologie des batteries est liée au couple électro-chimique utilisé. Actuellement, quatre technologies sont utilisées : Plomb/acide (Pb) à électrolyte gélifié ou liquide, Nickel/Cadmium (NiCd) et récemment Nickel/métal hydride (NiMH) et Lithium/ion. Il y a en fait deux grandes catégories : celle à base de plomb et celle à base de nickel, qui caractérisent l'utilisation. La technologie au plomb est destinée à l'alimentation des systèmes de secours car elle présente une faible auto-décharge pour une charge relativement lente, contrairement à l'autre catégorie qui est destinée à l'alimentation immédiate d'un système après recharge rapide ou normale, l'alimentation de type flottante

n'étant pas recommandée. Les technologies Lithium/ion trop récentes ou spécifiques à un domaine (spatial ou militaire par exemple) qui sont très peu répandues, donc onéreuses et souvent polluantes ont été écartées.

Selon la technologie, les batteries présentent des avantages et des inconvénients résumés dans le tableau II-1. La recharge est spécifiée en C (capacité nominale).

type de la batterie	Plomb/acide (Pb)	Nickel/Cadmium (NiCd)	Nickel/métal hydride (NiMH)
énergie	identique	identique	2 * Capacité NiCd
tension de base de la cellule	2 V	1,2 V	1,2 V
temps de charge	10 h	20 mn min.	1 h
charge	1/10 C	rapide à 1/10 C	rapide à 1/10 C
type de charge classique	tension	courant	courant
décharge	linéaire	plate puis abrupte	plate puis abrupte
auto-décharge	faible	rapide	rapide
risque de l'auto-décharge		destruction	destruction
utilisation	flottante	immédiate	immédiate
positionnement	aucune importance si électrolyte gélifié	aucune importance si électrolyte gélifié	aucune importance si électrolyte gélifié

Tableau II-1 : Les batteries

La qualité de la recharge d'une batterie est primordiale pour sa durée de vie. Deux techniques peuvent être mises en oeuvre : découpage ou linéaire. Habituellement, la technique à découpage est choisie pour les forts courants de recharge car le rapport puissance/volume est plus élevé. La méthode durant la première période de charge est fonction de la technologie de la batterie, ici la charge en tension. Le problème est de déterminer la fin de charge qui se caractérise par le respect d'un ou plusieurs critères parmi les suivants :

- (1) le temps maximum de charge,
- (2) la température maximale de la cellule,
- (3) le différentiel de température de la cellule $-\Delta\theta/\Delta t$.
- (4) la tension maximale de la cellule en charge (critère le plus couramment utilisé) ,
- (5) la tension de la cellule au repos,
- (6) le différentiel de tension $-\Delta V/\Delta t$ lorsque la tension de la batterie devient négative.

Seuls les critères (1), (2) et (4) sont généralement utilisés pour la recharge rapide. Le critère (6) n'est pas utilisé car le $-\Delta V/\Delta t$ est trop petit. Le critère (5) est utilisé uniquement pour une recharge par impulsion. Nous avons mis en oeuvre les critères (1), (4) et (6).

Comme le robot doit être utilisé dans un environnement pédagogique, la batterie doit rester pendant plusieurs mois au repos sans qu'elle puisse se détériorer. La technologie à base de Nickel, à la vue du tableau précédent, présente un fort taux de décharge et est donc écartée, car une non utilisation entraînerait un risque de décharge complète et une destruction de l'élément. De plus, la tension de la batterie ne sera pas utilisée directement, mais convertie et régulée, la forme de la courbe de décharge n'a donc pas d'importance. Mécaniquement, nous désirons ne pas être contraints par le positionnement de la batterie, nous avons donc choisi une technologie à électrolyte gélifié.

Nous devons, lors d'une perte d'alimentation interne sans secours externe, sauvegarder les paramètres systèmes importants du robot et alimenter l'horloge temps réel. Une pile présente un taux de décharge négligeable et est donc particulièrement adaptée à l'alimentation sous faible courant d'une horloge temps réel et à la sauvegarde mémoire.

5.9.1.4. Couplage source externe et sous-groupe alimentation interne

Pour éliminer la liaison externe entre le robot et la source d'alimentation externe, deux solutions étaient envisagées :

- les cellules solaires,
- un couplage magnétique entre une borne fixe et une bobine montée sous la plaque de base (liaison galvanique) par courant haute fréquence. L'utilisation de connecteurs serait trop complexe en ce qui concerne le positionnement et la fiabilité des contacts, solution pourtant retenue par le robot "ROBART II" [Everett and Gilbreath 88].

L'utilisation de cellules solaires a été exclue en l'état actuel de la technologie pour le rendement de l'ordre de 20 % qu'elles offrent. La solution du couplage magnétique est donc la meilleure solution.

5.9.2. Le système de surveillance

Des informations doivent parvenir au système informatique afin qu'il puisse gérer finement son alimentation. Chaque source d'alimentation est contrôlée par un système de surveillance qui fournit les informations binaires suivantes :

- au niveau batterie : tension haute, tension moyenne, tension basse,
- au niveau alimentations externe et logique (VCC) : sur-tension, tension nominale, et sous-tension,
- et les informations analogiques suivantes pour toutes les sources : tension et courant.

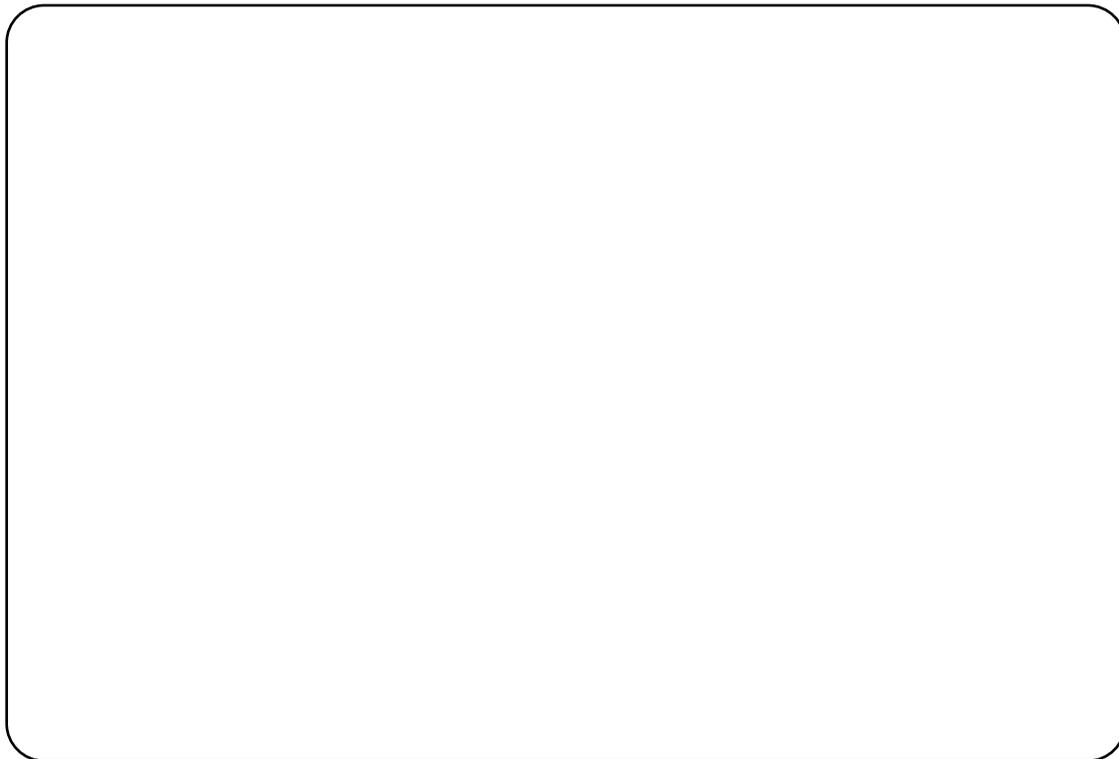
Pour une batterie, ces informations analogiques sont très importantes pour les sources autonomes car elles peuvent, après intégration par rapport au temps, nous indiquer le taux de charge, de décharge, la capacité instantanée et donner une indication du vieillissement pour une maintenance préventive en mémorisant certains paramètres comme :

- la date d'acquisition,
- le nombre de charges,
- les dates de charge et tension en début et fin de charge.

Le système doit être capable, lors d'une anomalie, de fournir un signal d'initialisation au système informatique.

5.9.3. Le sélecteur d'alimentation

L'alimentation externe prend la relève de la batterie embarquée lorsqu'elle est déchargée, cette sélection alimentation externe ou interne étant faite par le sélecteur d'alimentation (photographie II-11). La source, externe ou interne, est sélectionnée sur un critère de tension supérieure. Cet élément doit présenter, à l'état passant, la plus faible tension de déchet possible et, à l'état bloqué, le plus faible courant de fuite possible. La technologie Schottky a donc été utilisée.

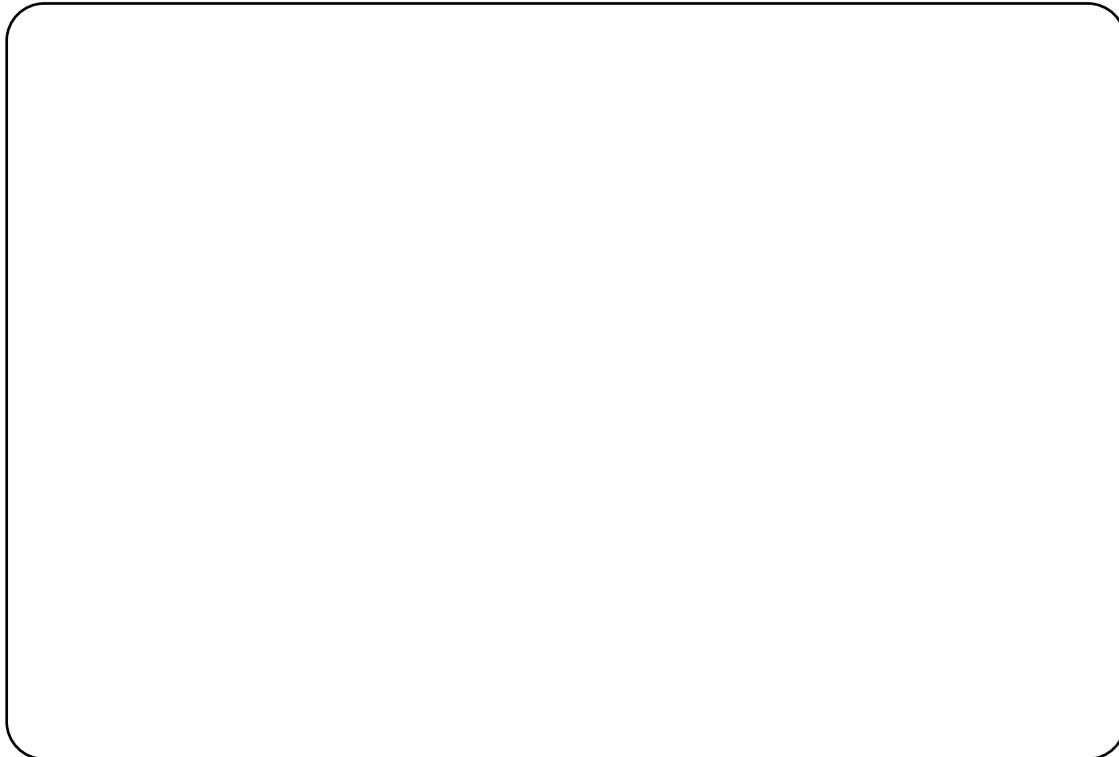


Photographie II-11 : Carte sélecteur d'alimentation

5.9.4. Le chargeur

Le critère de rendement n'est pas à retenir car le sous-ensemble est alimenté de manière externe. Bien évidemment, les pertes par effet Joule sont quand même à limiter pour ne pas élever la température interne du robot et sortir ainsi des limites de fonctionnement thermiques. Nous avons conçu un chargeur linéaire, de conception et de mise au point plus simple que celles d'un chargeur à découpage (photographie II-12).

Pour une batterie complètement déchargée, le cycle comporte trois états : une charge à courant constant faible pour évaluer l'état de charge de la batterie qui se poursuit par une charge à tension constante avec limitation en courant, le dernier état correspond à une charge à tension constante dite *de floating*. Un commutateur électronique déconnecte le chargeur lorsque la source externe n'existe plus afin d'éviter qu'un courant parasite décharge inopinément la batterie.



Photographie II-12 : Carte chargeur en situation

Pour résumer, le synoptique de la figure II-15 montre l'organisation du groupe énergie.

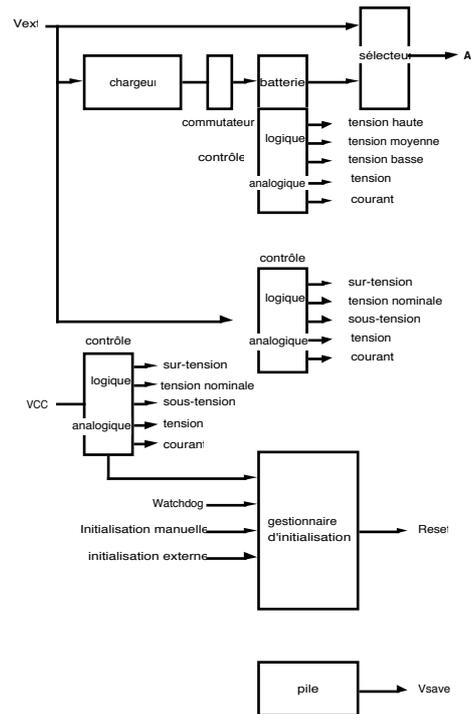
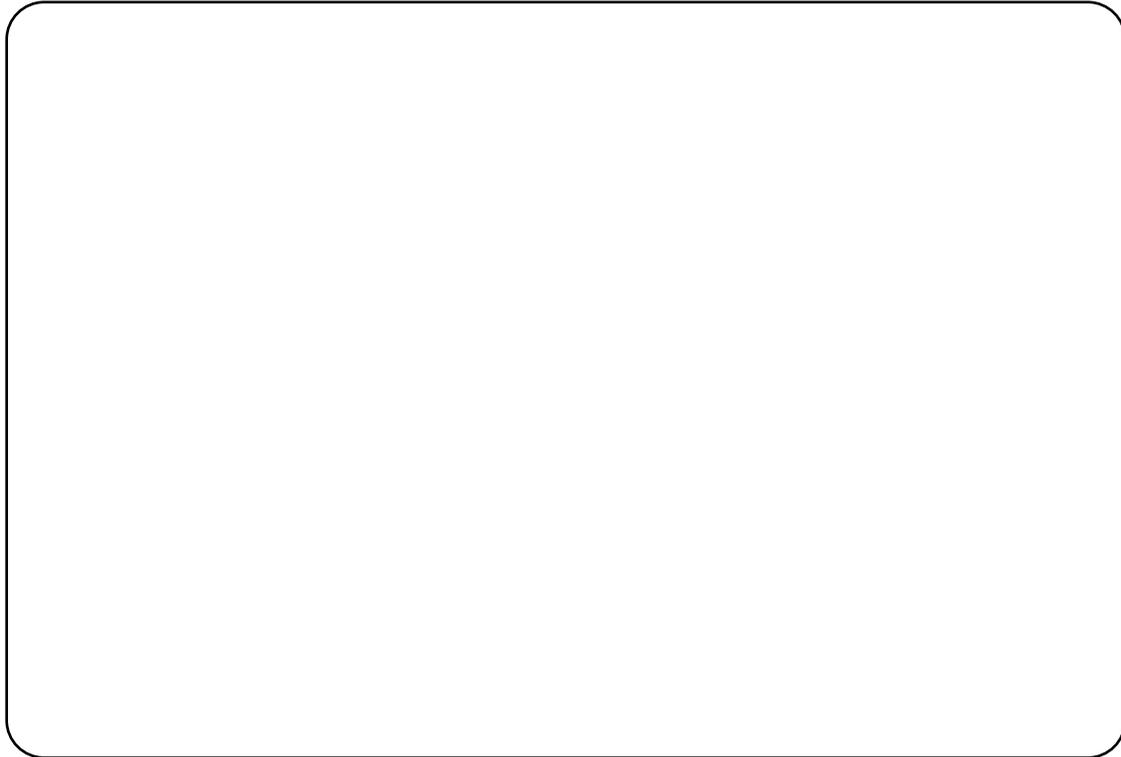


Figure II-15 : Gestion de l'énergie

5.9.5. Le convertisseur

Ce module (photographie II-13) est chargé de réguler la tension d'entrée provenant du sélecteur de tension à une tension régulée compatible avec la tension d'alimentation de la logique, c'est-à-dire 5 volts. Comme il est embarqué, il est important d'obtenir le meilleur rendement de conversion. Une régulation linéaire pour une tension d'entrée de 12 volts offrirait un rendement d'environ 40 %, un convertisseur à découpage, lui, offre un rendement, selon la technique de conversion, de près de 80 %. La puissance de sortie (30 W) a été prévue pour pouvoir supporter une ou deux caméras CCD, futurs capteurs du robot mobile.



Photographie II-13 : Carte convertisseur DC/DC en situation

5.10. Modélisation

Nous utilisons pour définir tous nos objets une approche orientée objet qui combine les aspects statiques et dynamiques de ceux-ci. Le modèle conceptuel s'appelle le *Model_Unit* (*M_U*) [Bachatène 92a] [Bachatène 92b]. Nous avons adapté ce modèle à la robotique en incluant les événement temporels dans les événements internes.

Nous définissons le modèle *Actor_Unit* après avoir défini le modèle objet conceptuel générique *Model_Unit* qui a servi de modèle de base.

5.10.1. Le *Model_Unit*

Le *Model_Unit* est un modèle conceptuel générique encapsulant la concurrence. Son comportement interne, qui est un ensemble d'états possibles, est défini par son cycle de vie. Il communique avec le monde externe par les services demandés et offerts. Des événements internes et externes peuvent interrompre les services en cours (figure II-16). Ces *Model_Units* se décomposent en deux sous-familles :

- les *M_U* actifs, composés des gérants d'entrée, des gérants de sortie et des synchroniseurs,
- les *M_U* passifs.

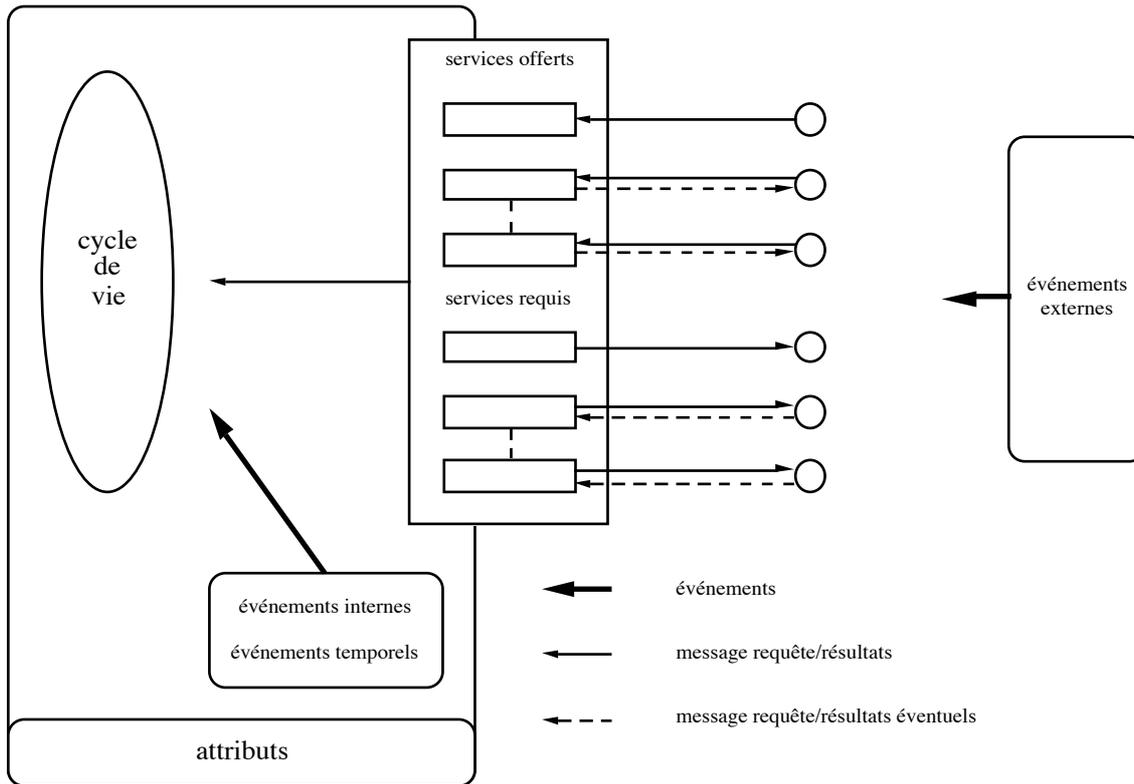


Figure II-16 : Le Model_Unit

Ce modèle possède des limites pour la modélisation de l'acteur matériel. Il n'y a pas de dissociation explicite dans le cycle de vie, du mode de fonctionnement normal et des cycles système et interruption. De plus, les événements internes et temporels sont dissociés. Le message formalise les interactions comme les demandes pour certains services et les résultats éventuels mais les événements ne sont pas encapsulés sous la forme d'un message mais déclenchent un événement interne. Ce modèle doit donc être adapté pour décrire un acteur matériel.

5.10.2. L'Actor_Unit

Contrairement au Model_Unit, nous assimilons les événements temporels à un sous-ensemble appartenant aux événements internes. Les événements sont de deux types : interne et externe. Il peut s'agir d'un signe ou d'un signal. De plus le cycle de vie est nommé **automate application**, et deux automates supplémentaires nommés **automate système** et **automate interruptif** sont ajoutés. Les trois comportements que possède l'acteur matériel sont par priorité croissante de traitement :

- le comportement application,
- le comportement système,
- le comportement interruptif.

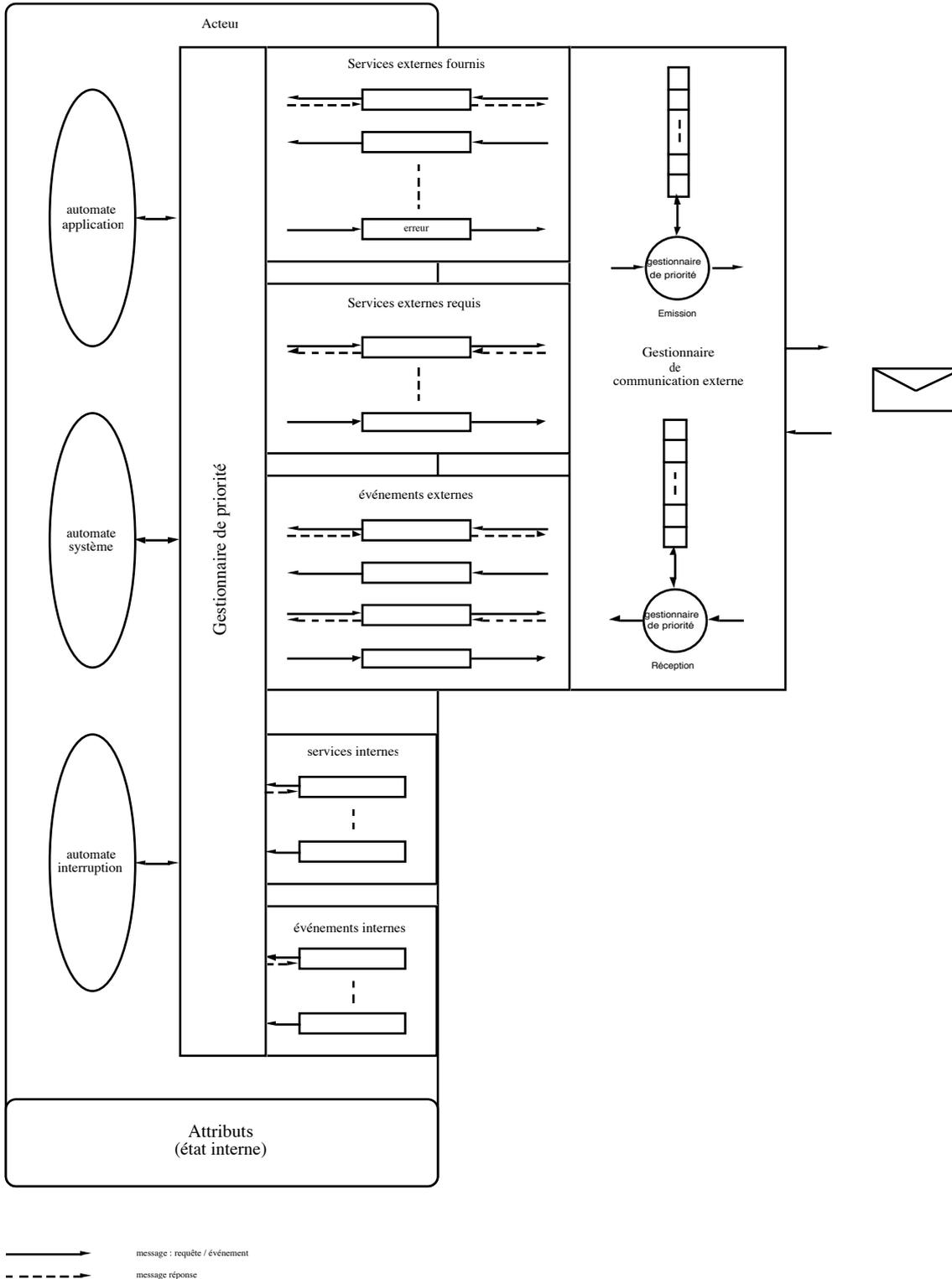


figure II-17 : Organisation interne d'un acteur robotique

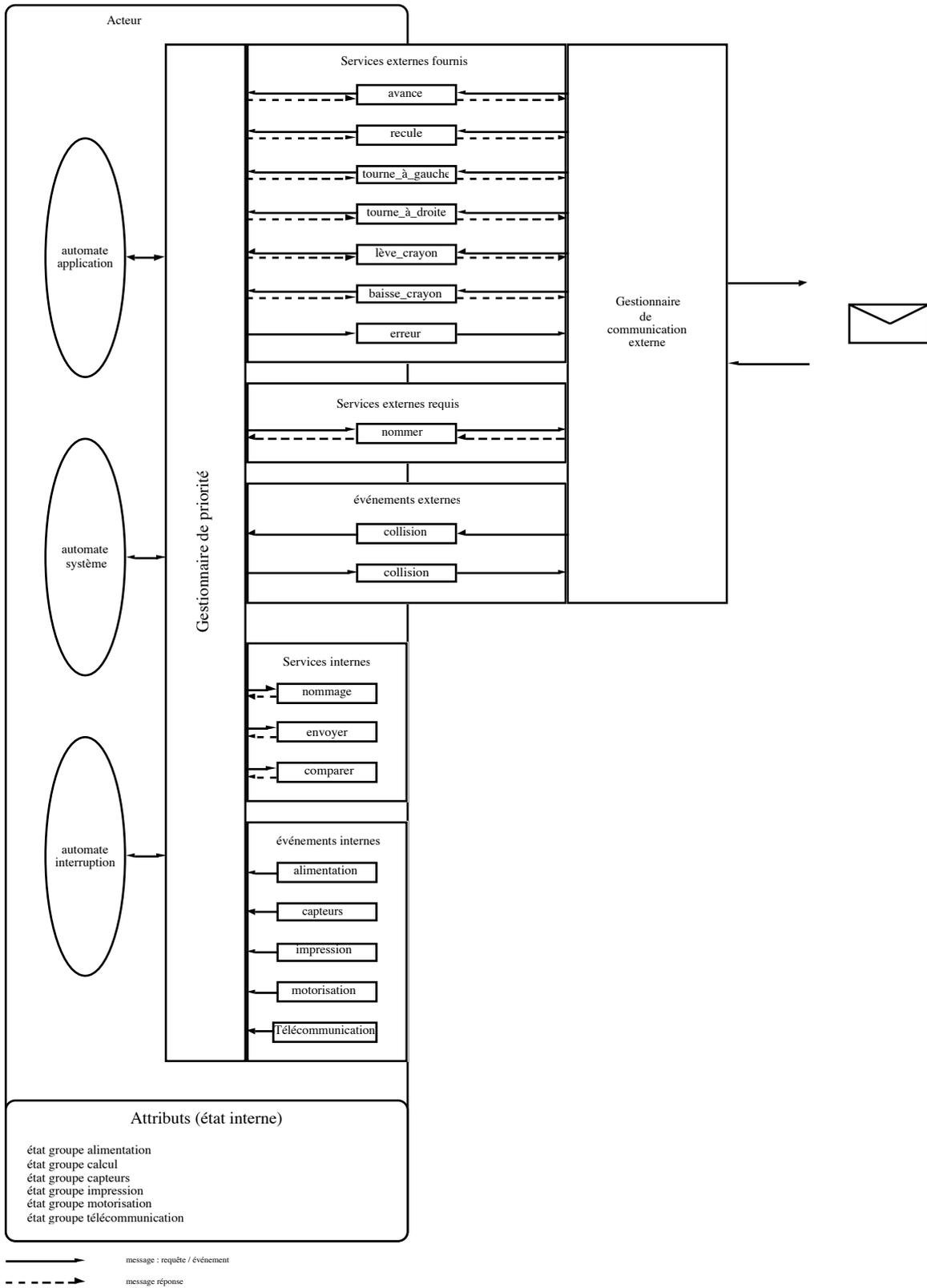


Figure II-18 : Modélisation de l'acteur robotique mobile "tatou"

Les services et les événements sont représentés par des rectangles (figure II-17). A chaque service ou événement est associée une file d'attente. Au niveau communication, une file *messages en entrée* et une file *messages en sortie* régulent les communications. Un gestionnaire de priorité gère l'ordre d'évaluation des méthodes. A tout instant, une méthode en cours d'exécution peut être suspendue, pour une méthode plus prioritaire ou une méthode d'un autre comportement plus prioritaire. L'état interne du robot est représenté par les attributs regroupés par groupe fonctionnel.

5.10.3. Modélisation du robot mobile

Nous reprenons le modèle Actor_Unit pour modéliser le comportement de l'acteur robotique "Tatou".

Les services externes au nombre de six (avance, recule, tourne_à_gauche, tourne_à_droite, lève_stylo, baisse_stylo) sont ceux de la tortue Logo (figure II-18).

6. Le robot bras

Nous avons sélectionné un robot pédagogique commercial à poste fixe : le robot "Youpi" (photographie II-14) [J.D. Productique 86] pour son rapport qualité-prix. Il a été sélectionné pour équiper les collèges et les grandes écoles. Comme le robot est destiné à l'enseignement de la robotique, sa conception a été prévue robuste, sans pour autant sacrifier la précision et la puissance du mouvement.

6.1. Mécanique

Ce robot est de type bras à cinq degrés de liberté. La motorisation s'effectue par six moteurs pas à pas (200 pas par tour) commandés en mode demi-pas. La transmission mécanique, sauf pour la pince, s'effectue via des systèmes à courroie crantée et pignons à denture droite. La résolution des axes après démultiplication est de 0,04°. La vitesse angulaire maximale est de 40°/seconde. Chaque axe est équipé d'un capteur de position initiale. Aucun capteur de position n'est couplé aux axes. La pince est bidigitale de type parallélogramme. Le serrage s'effectue par l'intermédiaire d'un système vis-écrou irréversible. La charge utile d'un diamètre maximal de 80 mm est de 400 g à vitesse maximale.



Photographie II-14 : Le robot sédentaire “Youpi”

6.2. Interface électronique

Le robot est commandé par une interface spécialisée d'entrée-sortie de type parallèle compatible TTL collecteur ouvert. L'électronique de puissance est intégrée au socle et la logique d'interface (décodage d'adresse, arbitrage de bus, mémorisation/amplification) et de contrôle (mémorisation et comptage) est montée sur une carte périphérique compatible bus PC/AT. Des capteurs de positions extrêmes optoélectroniques renvoient une information binaire vers l'interface. Ce capteur est constitué d'un pignon recouvert de deux demi-disques noir et blanc et d'une fourche opto-électronique (diode électro-luminescente infra-rouge non modulée et photo-transistor). De par sa conception, celui-ci est très sensible aux lumières parasites (lumière ambiante naturelle ou artificielle), car la source infra-rouge n'est pas modulée. Pour connaître la position des axes, il est nécessaire de compter les impulsions moteur, en espérant que la commande ne génère pas des pertes de pas ou que des glissements mécaniques n'interviennent pas. Les moteurs sont donc pilotés en boucle ouverte. Le robot est équipé d'une alimentation 220 volts intégrée qui fournit une tension continue à l'interface de puissance.



Photographie II-15 : Carte d'adaptation robot-E/S

6.3. Commande

Nous avons conçu une carte prototype pour réaliser l'interface entre le robot bras et la carte générique de commande (photographie II-15). Cette interface est de type externe à commande de bas niveau (interface type 1) puisqu'elle adapte uniquement les signaux logiques de commande et ceux provenant des capteurs aux exigences électriques de la carte générique.

7. Uniformisation des E/S

L'étude précédente fait donc ressortir un besoin d'indépendance de l'acteur vis-à-vis des entrées-sorties, afin de n'être plus tributaire du matériel et de faciliter l'évolution de l'architecture. Nous devons spécifier des caractéristiques génériques s'adaptant à tous types de matériels robotiques. Il faut aussi proposer une carte de contrôle intelligente offrant un maximum de fonctionnalités pour un coût intéressant. Elle doit s'adapter à tout type de matériel robotique, soit directement, soit par le biais d'interfaces ayant un minimum de composants. Nous avons reporté le plus possible les fonctionnalités au niveau du logiciel pour baisser les coûts matériels. Nous proposons une interface de type 2 ou 4 appelée système NIOS (Normalized Input-Output System). Nous allons étudier les différents périphériques pouvant exister, et ensuite, à partir de cette étude, nous proposerons une architecture générique.

7.1. Etude matérielle

Pour appréhender le monde réel, un transducteur est nécessaire pour traduire le phénomène ou la grandeur physique en un signal compréhensible par l'ordinateur selon la chaîne d'acquisition suivante : monde réel ou micro-monde <--> grandeur physique <--> grandeur électrique <--> monde discret <--> monde interprété. L'information du monde réel à recueillir est logique ou analogique. Un transducteur établit la relation entre la grandeur physique et la grandeur électrique. Pour passer de cette dernière au monde discret et, inversement, une conversion est nécessaire (A/N ou N/A). L'interprétation se fait au niveau de l'application. Rappelons qu'un signal électrique élémentaire se caractérise principalement par sa polarité, son amplitude en tension et en courant et sa fréquence. Pour un signal logique, il faut en plus définir les seuils de basculement.

S'il s'agit d'appréhender une grandeur physique, nous appellerons le transducteur, capteur. Le tableau II-2 donne la correspondance entre des grandeurs physiques usuelles et les capteurs à informations logique / analogique associés :

grandeur physique	capteur logique	capteur analogique
lumière		photo-cellule résistance, diode, transistor
température	thermostat	thermistor, thermo-couple
son		microphone
distance		capteur à ultra-son laser
vitesse		cinémomètre resolver

Tableau II-2 : Correspondance grandeur physique / capteur

Les périphériques de sortie sont commandés, soit de manière logique, soit de manière analogique. Le tableau II-3 donne la correspondance entre transducteur et grandeur physique.

grandeur physique	transducteur
lumière	lampe, LED
température	résistance
son	haut-parleur
divers mécanique	moteur

Tableau II-3 : Correspondance grandeur physique / transducteur

Les périphériques d'entrée-sortie sont principalement des périphériques de communication série, parallèle ou réseau (Ethernet). Il peut s'agir aussi de périphériques des types précédents associés : un servo-moteur par exemple.

7.2. Proposition d'interface uniformisée

Le système NIOS est représenté en figure II-19.

7.2.1. L'alimentation

L'alimentation est placée à l'extérieur pour pouvoir s'adapter facilement à chaque standard international. La tension principale est fixée à +5 Volts pour la logique avec une sortance en courant de 10 A maximum et les tensions auxiliaires à +/-12 Volts avec une sortance maximale en courant de 1 A. Ces tensions auxiliaires sont présentes si les conversions A/N et N/A le nécessitent. La tension auxiliaire -12 Volts est générée en interne par un convertisseur DC/DC car une entrée auxiliaire pour une batterie externe +12 Volts doit être prévue.

7.2.2. Les entrées-sorties

Nous avons fixé un nombre maximum de 256 entrées et sorties logiques et analogiques. Chaque entrée-sortie, selon sa catégorie, est repérée par un numéro physique (P D/A I/O N : Physical Digital/Analog Input/Output Number) allant de 0 à 255.

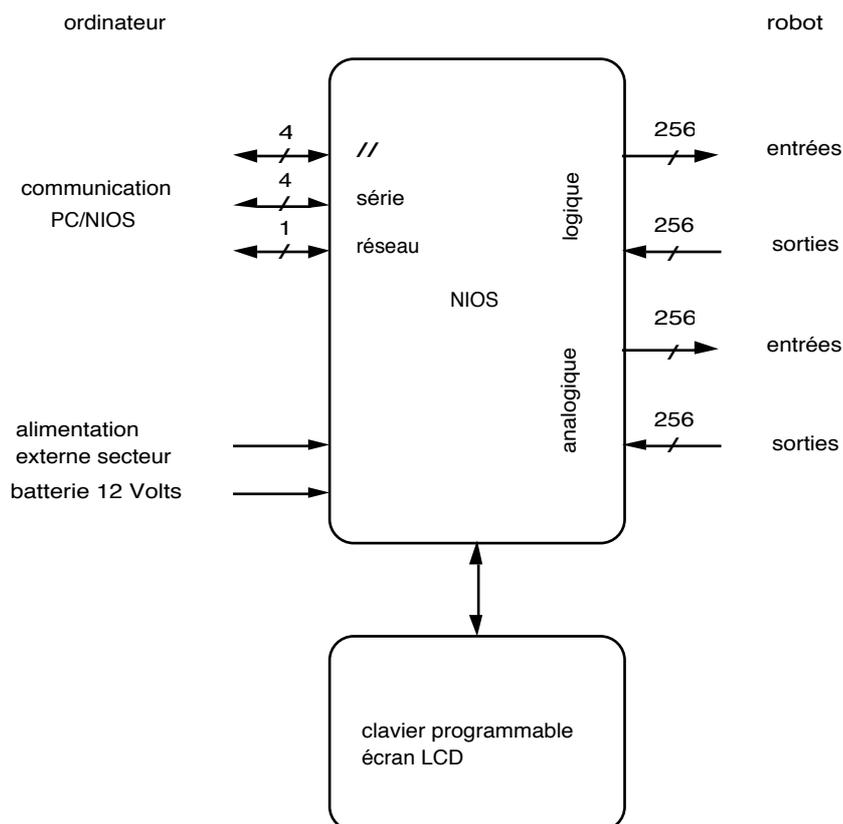


Figure II-19 : Chaîne de contrôle d'un robot pédagogique

Les entrée-sorties logiques doivent être compatibles C-MOS - TTL 5 Volts au niveau des tensions et courants. L'interface de puissance, si elle est nécessaire pour adapter les signaux, sera externe. Une sortie logique se caractérise par son type : totem-pole, collecteur-drain ouvert, trois états. Le type sortie trois états semble le plus adapté car il permet, lorsque deux systèmes électroniques sont couplés, de couper l'alimentation de l'un sans perturber l'autre. Pour des tensions de sortie logique différentes ou s'il y a nécessité d'isolation galvanique, il faudra utiliser soit une interface de type collecteur-drain ouvert, soit un opto-coupleur, soit un relais.

Les tensions autorisées sont celles de l'alimentation auxiliaire (+/-12 Volts). Un système de protection des entrées et des sorties doit être prévu. Une conversion de type 8 bits paraît, pour la plupart des applications, suffisante. Pour les sorties, un courant de +/-10 mA est fixé, c'est-à-dire qu'aucune puissance n'est disponible. Une conversion se caractérise par une écriture d'une valeur numérique à convertir pour une conversion N/A, par un lancement de la conversion, par un temps de conversion, et par une mise à la disposition d'une valeur numérique ou analogique selon le type de conversion.

A chaque entrée-sortie est associé un événement. Il peut concerner les entrées-sorties logiques ou analogiques. Une fonction simple ou complexe peut être associée à cet événement. Un événement simple logique se caractérise par un déclenchement sur front montant, descendant ou les deux, et/ou sur niveau bas, haut, ou les deux. Il est repéré par un numéro physique (SDEN : Simple Digital Event Number) allant de 0 à 511. Un événement simple analogique tout-ou-rien se caractérise par un déclenchement sur valeur seuil avec ou sans hystérésis. Il intervient par déclenchement sur front montant, descendant ou les deux, et/ou sur niveau supérieur, inférieur ou les deux. Un événement analogique est repéré par un numéro physique (SAEN : Simple Analog Event Number) allant de 0 à 511. Un événement complexe, en fonction de sa catégorie, est une fonction logique d'événements simples.

7.2.3. Interfaces de communication

Nous supportons au maximum 4 ports parallèles compatibles Centronics, 4 ports série compatibles RS232C et un port réseau Ethernet.

Un clavier 16 touches sera prévu, ainsi qu'un afficheur LCD pour pouvoir dialoguer en autonome avec le système.

7.2.4. Interface logicielle

Trois modes de fonctionnement de l'interface sont disponibles : système, autonome et application. Un automate permet d'interpréter les commandes qui sont regroupées en un jeu standard et un jeu non standard extensible. L'ensemble de ce vocabulaire constitue le dictionnaire. L'ensemble minimum du vocabulaire de commande des entrée-sorties côté robot est présenté en annexe. L'automate dialogue avec le PC via l'interface série avec le protocole Xon-Xoff. L'élément de base de communication est l'octet et les caractères sont représentés

selon le code ASCII étendu. Au niveau application, une bijection sera établie entre les numéros physiques des entrées-sorties et des événements, et les noms logiques.

8. Conclusion

Après une brève présentation de l'état de l'art concernant les robots mobiles et la robotique pédagogique, nous avons présenté l'architecture matérielle des acteurs matériels. Nous avons développé une carte générique de commande pour les contrôler. Cette carte permet de s'adapter à tout type de robots pédagogiques. Un prototype a été réalisé en technique "wrapping".

Nous avons sélectionné pour la réalisation le transputer. ce micro-processeur n'autorisant qu'une interruption matérielle, nous avons proposé une amélioration matérielle pour prendre en compte plus d'une interruption.

Pour valider le concept d'acteur matériel, nous avons développé un robot mobile prototype le "Tatou" dont nous avons détaillé l'architecture électronique et mécanique.

Un modèle de description est proposé, l'Actor_Unit, s'inspirant du Model_Unit et nous avons décrit le comportement du robot mobile "Tatou" lors de son utilisation dans un exemple de tri réparti proposé dans [**Darche and Nowak 93**].

L'étude de la connexion d'un robot pédagogique avec un système de contrôle a soulevé le problème de l'interface. Nous avons proposé d'uniformiser l'interface afin de ne plus être tributaire du matériel. Cette proposition prend la forme d'un module adaptateur, le système NIOS, qui permet d'adapter le robot pédagogique au système de commande en utilisant des interfaces standards. Un langage de programmation spécialisé dans les entrées-sorties sera présenté dans le chapitre *Architecture logicielle*.

L'architecture proposée est suffisamment générique pour pouvoir s'adapter à des robots plus complexes. L'utilisation envisagée d'ActNet est l'IAD et la robotique mobile classique. Nous proposons dans le dernier chapitre une architecture logicielle permettant cela.

L'acteur matériel communique avec ses congénères par messages via des sous-réseaux locaux. Le chapitre III décrit l'architecture matérielle de ces sous-réseaux.

Chapitre III

Le système de communication

- 1. Introduction**
- 2. Techniques radio-fréquences**
- 3. Etat de l'art des réseaux radio-fréquences**
- 4. Sous-réseau d'acteurs mobiles**
- 5. Sous-réseau d'acteurs sédentaires**
- 6. Passerelle**
- 7. Conclusion**

1. Introduction

Pour communiquer, les acteurs matériels et logiciels font transiter leurs messages par un réseau. ActNet est un réseau virtuel qui permet de faire communiquer des acteurs matériels et logiciels. Physiquement, il se compose de trois sous-réseaux : le sous-réseau utilisateur, le sous-réseau d'acteurs mobiles et le sous-réseau d'acteurs sédentaires.

Nous avons réalisé un réseau hétérogène par leur média. Le sous-réseau utilisateur est un réseau commercial. Le sous-réseau d'acteurs mobiles est un réseau radio-fréquence à topologie programmable et celui des acteurs sédentaires est un réseau filaire de type anneau à jeton.

Après avoir présenté les techniques spécifiques au domaine de la radio-fréquence, nous détaillons, dans ce chapitre, les spécificités techniques de chaque sous-réseau. Nous désirons une communication duplex intégral.

1.1. Modélisation d'une communication

La communication est un transfert d'information d'une source vers un destinataire à travers un milieu appelé canal. La communication peut être unilatérale ou bilatérale, bi ou multi-locuteurs. La figure III-1 représente une communication unilatérale, l'extension aux autres formes pré-citées étant naturelle.

Le codage permet de transporter l'information de manière fiable en fonction des contraintes physiques. On distingue deux types de codage :

- le codage de source qui adapte le signal au mode de transmission,
- le codage de canal qui adapte le signal au support de transmission.

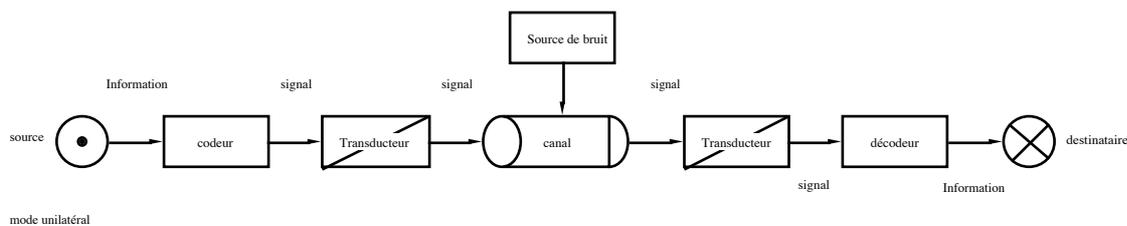


Figure III-1 : Modélisation de la communication

Si le codage de canal n'existe pas, nous sommes en présence d'une transmission en bande de base.

Le canal n'est pas un support idéal. Il modifie et perturbe le signal transporté. Les perturbations sont modélisées par une source de bruit. Le codeur de canal doit en tenir compte. Le transducteur transforme la nature du signal pour l'adapter au milieu de

transmission (onde électrique <--> onde électro-magnétique, ultrasonore, infrarouge, etc.). Pour un canal radio-fréquence, le transducteur est une antenne.

1.2. Rappel des contraintes

Les équipements électroniques de transmission étant qualifiés d'embarqués, nous avons considéré lors de la conception, les critères suivants en ce qui concerne la communication :

- l'autonomie énergétique et l'autonomie de la communication pour l'acteur robotique mobile,
- la fiabilité de la transmission,
- la vitesse de communication,
- la facilité de mise en oeuvre,
- les distances mises en jeu,
- et le coût.

Les systèmes embarqués nécessitent d'être équipés de modules électroniques consommant le moins possible de courant. C'est ce que nous avons appelé le critère d'autonomie énergétique. Ce critère est moins important pour un acteur sédentaire qui est connecté à une source permanente. Par contre, il est primordial pour un acteur mobile.

Les acteurs mobiles nécessitent une communication sans lien physique pour répondre au critère d'autonomie de la communication, alors que les acteurs sédentaires, ne se déplaçant pas, n'en ont pas besoin.

Le critère de fiabilité est plus difficile à résoudre pour le réseau radio-fréquence puisque le canal, en l'occurrence l'atmosphère, est accessible par tous et peut donc être perturbé. Il peut s'agir d'un acte volontaire d'intrusion ou de perturbation, ou d'un acte involontaire. Une liaison par câble est plus fiable par définition malgré des perturbations électro-magnétiques comme la diaphonie ou l'atténuation.

La vitesse de communication est plus élevée pour un réseau filaire. En effet, la ressource canal radio-fréquence est limitée en nombre et en bande passante. La conception du réseau doit prendre en compte cette limitation.

D'un point de vue matériel, la conception et la mise en oeuvre d'un réseau filaire sont plus aisées que celles d'un réseau sans liaison physique, car les communicateurs sont moins complexes.

La distance de connexion est limitée. Elle est principalement due à l'atténuation du support.

2. Techniques radio-fréquences

La technologie de la transmission filaire est bien stabilisée, alors que celle de la radio-fréquence est en pleine expansion. Les comités de normalisation travaillent actuellement sur

ce principe pour la téléphonie et les réseaux numériques, avec en particulier le comité IEEE (Institute of Electrical and Electronics Engineers) 802.11 qui s'intéresse aux couches matérielles (1) (MAC : Medium Access Control) et liaison (2) (LLC : Logical Link Control) du modèle OSI/ISO (Open Systems Interconnection Basic reference Model/ International Standards Organization) pour les réseaux locaux. Mais il n'existe pas à la date d'aujourd'hui de normes pour les réseaux locaux. Nous détaillons dans cette partie les techniques mises en oeuvre pour le réseau radio-fréquence. Après avoir présenté les modulations analogiques discrètes, nous détaillons les principes d'émission-réception, ainsi que la synthèse digitale de fréquence.

2.1. Modulations

A ce point de notre étude, il serait intéressant de savoir si une modulation d'un signal par l'information à transmettre doit être mise en oeuvre. Certaines lignes de transmission peuvent transmettre tel quel le signal, par exemple un câble coaxial ou une fibre optique. Il s'agit d'une **transmission dite bande de base**. La bande passante du canal est supérieure à celle du signal à transmettre.

Considérons d'abord le transducteur comme idéal, c'est-à-dire qu'il transmet et reçoit fidèlement l'information d'un point de vue fréquentiel. L'hypothèse est faite que le signal représentant l'information est à large spectre. Si plusieurs acteurs conversent au même instant, il devient impossible de distinguer les informations des divers interlocuteurs : il y a collision dans le canal. Des protocoles du type CSMA s'accommodent parfaitement de ce phénomène puisqu'ils le détectent (option détection de collision CD). Mais nous devons tenir compte de la nécessité d'établir des communications en duplex intégral. Une transformation ou plutôt une transposition fréquentielle du signal spécifique à chaque acteur s'impose. Il est vrai que l'information en elle-même pourrait contenir une information identificateur locuteur.

Considérons maintenant l'antenne réelle. Cette dernière est assimilable à un filtre passe-bande et elle ne peut pas être adaptée pour une gamme étendue de fréquences. Il est donc nécessaire, grâce à un opérateur comme la modulation, de concentrer le spectre de fréquence caractérisant l'information dans la bande passante du transducteur.

La **modulation** est une opération qui consiste, comme son nom l'indique, à moduler un signal porteur (ou porteuse) par l'information à transmettre (signal primaire ou modulant), le signal résultant étant nommé signal secondaire ou modulé (figure III-2). La fréquence de la porteuse doit être très supérieure à celle du signal modulant.

Comme il s'agit d'une transmission radio-fréquence, la porteuse est une onde sinusoïdale car elle seule permet un transport optimal de l'information dans l'atmosphère. Nous avons donc affaire à une modulation analogique. Comme l'information est de nature discrète, nous parlerons d'une **modulation analogique discrète**.

Pour obtenir une communication en duplex intégral, l'espace des fréquences sera découpé en bandes identiques appelées **canaux radio-fréquences**. Une bande sera affectée à

chaque acteur communicant et nous considérerons la fréquence médiane de cette bande c'est-à-dire la fréquence de la porteuse, comme l'**adresse fréquentielle** de l'acteur mobile communicant.

Deux types de modulation existent : la modulation analogique et la modulation numérique.

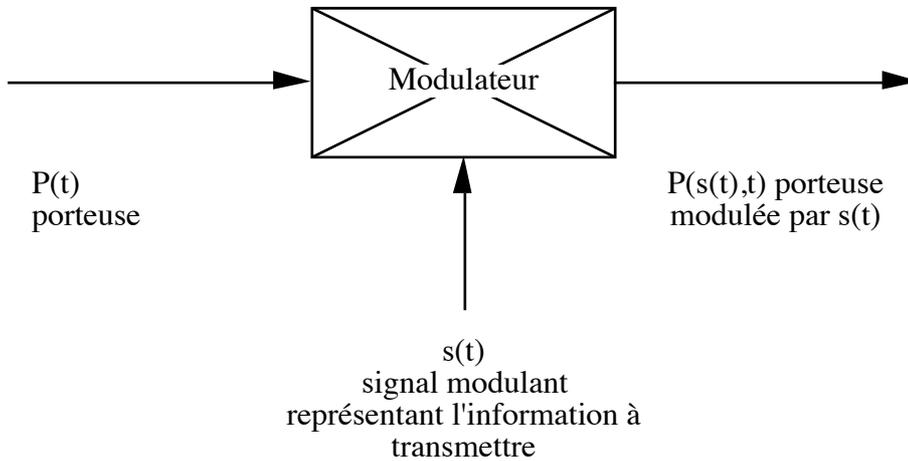


Figure III-2 : La modulation

Le premier type sert à la transmission numérique et analogique. Les signaux analogiques (son, vidéo, etc.) sont en général compressés et/ou pré-accentués avant transmission, puis après réception décompressés et/ou désaccentués. Pour les signaux numériques, l'information binaire module une sous-porteuse basse-fréquence, mais le débit est généralement limité à 9600 bauds dans la pratique.

Un signal électrique peut être représenté par une fonction alternative du temps d'expression suivante :

$$y(t) = a_0 \sin(\omega_0 t + \varphi_0) = a_0 \sin(2\pi f_0 t + \varphi_0)$$

où :

- a_0 représente l'amplitude,
- ω_0 est la pulsation,
- f_0 est la fréquence,
- φ_0 est la phase initiale ou angle de phase,
- $\omega_0 t + \varphi_0$ est la phase instantanée.

Au regard des variables, trois types de modulation peuvent être envisagés :

- la **modulation d'amplitude** (AM),
- les **modulations angulaires ou exponentielles** avec la **modulation de fréquence** (FM), et la **modulation de phase** (PM).

De plus, pour une onde électro-magnétique, le signal peut être polarisé dans l'espace (orientation verticale ou horizontale du vecteur champ électrique). Il s'agit d'un quatrième type de modulation : la modulation de polarisation qui n'est pratiquement pas employée. De plus dans notre application, notre antenne a une position fixe dans l'espace.

La modulation d'amplitude est très sensible aux perturbations de toutes sortes et n'est pas utilisée pour des transmissions de qualité [Pelat 86]. C'est pour cela que nous lui préférons les modulations angulaires.

2.1.1. Modulations discrètes

Suite à la présentation des différentes modulations analogiques et en tenant compte du fait que la modulation est discrète, nous avons :

- la modulation à décalage d'amplitude dite **ASK** (Amplitude Shift Keying) à m valeurs. Dans notre cas particulier, m est égal à 2, nous parlerons ainsi de la modulation d'amplitude tout ou rien dite **OOK** (On-Off Keying),
- des modulations hybrides comme la modulation **QAM** (Quadrature Amplitude Modulation) avec un espace d'états à deux dimensions amplitude-phase,
- la modulation **FSK** (Frequency Shift Keying), modulation à saut de fréquence qui présente une immunité élevée aux bruits et un faible coût, mais limitée en vitesse de modulation,
- la modulation **PSK** (Phase Shift Keying) avec en particulier la modulation **DPSK** (Differential Phase Shift Keying) avec un espace d'états à deux dimensions amplitude-phase, où seule la phase varie. Le nombre d'états est fonction du débit désiré et de la complexité des modulateurs/démodulateurs.

Pour la modulation FSK, lorsque le signal numérique n'est pas parfait, le glissement d'un état à l'autre n'est pas brusque mais varie de manière continue. Nous parlons alors de modulation CPFSK (Continuous Phase Frequency Shift Keying).

Pour transporter des signaux numériques, nous avons utilisé un canal de largeur de bande définie. Il s'agit d'un canal audio-fréquence, ayant une bande passante de 3 KHz environ. Nous pouvions utiliser un modem qui utilise le plus souvent une modulation de phase multipôles ou concevoir directement un modulateur. Cette solution s'impose pour minimiser le nombre d'éléments (critère d'encombrement) et, par la même occasion, la consommation.

2.1.2. Modulation par étalement de spectre

La **modulation à spectre élargi** nommé encore **à étalement de spectre (Spread-Spectrum Carrier (SSC))** repose sur la distribution des informations dans une bande de fréquence qui est très supérieure à celle nécessaire pour la transmission de celles-ci. La redondance des signaux permet de récupérer l'information, même dans un milieu très perturbé. Cette modulation a été développée à l'origine pour l'armée pour résister à l'écoute

et l'intrusion dans les radio-communications. Il existe deux techniques : le saut fréquentiel et la séquence directe.

Dans la modulation à saut fréquentiel ou à échappement de fréquence, le message est découpé en espaces temporels et transmis à des fréquences aléatoires selon une séquence. Le code pseudo-aléatoire module la fréquence instantanée de la porteuse. La présence d'une fréquence ne dépasse pas la demi-seconde. La démodulation est synchrone et un corrélateur qui connaît la séquence aléatoire redonne les données initiales.

Dans la modulation de séquence directe à large spectre, la sortie modulée est une fonction du signal contenant l'information avec un code pseudo-aléatoire (PRN : Pseudo-Random Noise) à fréquence plus élevée que celle du signal à transmettre. La nature pseudo-aléatoire du code conduit à un spectre large et plat. La démodulation est cohérente et un corrélateur permet de redonner les données transmises. En terme d'efficacité au niveau occupation fréquentielle, pour les exemples donnés plus loin (facteur d'élargissement : 15), ce type de modulation occupe 15 fois plus de place que la bande nécessaire, mais la largeur de bande peut être partagée, car chaque utilisateur considère le signal ne le concernant pas comme du bruit. La séquence pseudo-aléatoire personnalisée peut devenir alors l'adresse de l'acteur.

Cette technique est en pleine évolution et les composants électroniques associés sont en début de commercialisation. Nous lui avons donc préféré une modulation plus classique où la technologie est établie.

2.1.3. Les modulations MFSK et GMSK

Nous avons sélectionné une modulation de type CPFASK (Continue Phase Frequency Shift Keying) ou MFSK (Minimum Frequency Shift Keying) ou FFSK (Fast Frequency Shift Keying), car elle optimise l'occupation spectrale et présente une grande facilité de mise en oeuvre. L'indice de modulation m est égal à 0,5. La variation de phase est continue.

Une étude comparative [**Amoroso 80**] a montré que la modulation MFSK était optimale au niveau occupation spectrale par rapport aux modulations de type BPSK et SFSK (bande passante définie à 50 % de la puissance du signal, définition adoptée par la Federal Communications Commission (FCC)). Seule la QPSK la dépasse mais sa mise en oeuvre technique est plus complexe. 99 % de l'énergie du signal est contenue dans la bande passante, ce qui correspond à une occupation spectrale de 1,18 Hz/bit/s.

La modulation GMSK (Gaussian Minimum Shift Keying) est une amélioration qui consiste à utiliser un filtre gaussien pour le signal NRZ (Non Retour à Zéro) afin de diminuer l'encombrement spectral en supprimant les lobes secondaires. En général, cette variation de phase continue peut être obtenue en utilisant des diodes de type varicap et en les commandant par une tension continue.

2.1.4. Conclusion

L'information à transmettre est de nature discrète et binaire (signal digital à $m = 2$ valeurs discrètes). Elle transitera par la liaison série de la carte de commande.

Pour la transmission radio-fréquence, le milieu ambiant sera le canal de transmission et les ondes radio-fréquences ont été choisies comme support de la transmission. Les propriétés du canal disponible interdisent une transmission numérique en bande de base. Une modulation analogique discrète est donc nécessaire. Les modulations analogiques discrètes permettent de transposer les caractéristiques fréquentielles du signal pour l'adapter aux contraintes de l'émission/réception radiofréquence, c'est-à-dire en ce qui concerne :

- les antennes, pour l'encombrement et la bande passante,
- le canal, pour la propagation, la largeur de bande disponible du canal et pour les perturbations (bruit, etc.), et déformations éventuelles,
- la communication, pour le mode duplex intégral.

Nous avons sélectionné la modulations MFSK pour ses propriétés.

2.2. Principe d'émission-réception

Il est nécessaire d'utiliser un émetteur-récepteur pour qu'il y ait dialogue entre acteurs mobiles. Nous détaillons dans cette partie les principes régissant ce type de communication.

2.2.1. Emission

Le présent paragraphe explique le fonctionnement d'un émetteur générique dont le synoptique général est représenté en figure III-3.

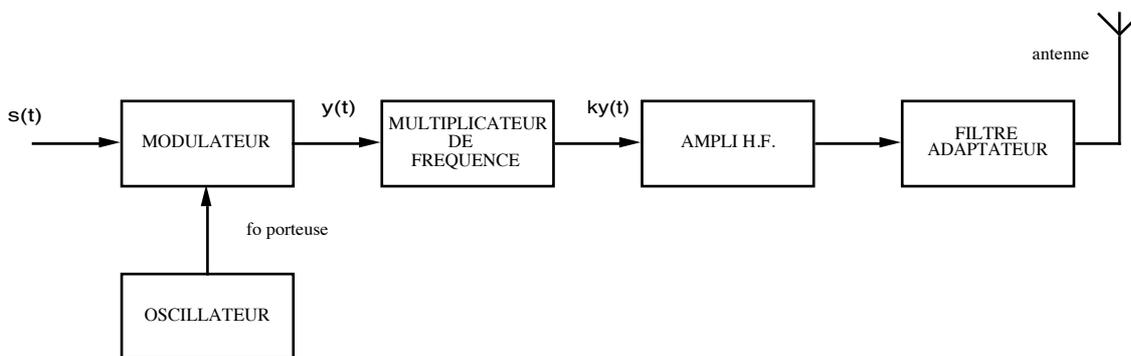


Figure III-3 : Synoptique de la chaîne émission

Le synoptique se compose de cinq blocs fonctionnels :

- le modulateur,
- l'oscillateur,
- le multiplicateur de fréquence,
- l'amplificateur haute-fréquence,

- le filtre adaptateur.

Le modulateur est un modulateur de fréquence, l'oscillateur de référence, dont la fréquence centrale f_p est égale à l'adresse fréquentielle de l'acteur à joindre, génère la porteuse. Le modulateur, de par son rôle, fournit le signal modulé $y(t)$, résultat de la modulation de la porteuse f_p par le signal à transmettre $s(t)$.

Le multiplicateur de fréquence est un bloc facultatif. Il faut savoir que le signal $y(t)$ peut ne pas avoir la fréquence désirée (à cause de la fréquence de travail imposée par le modulateur par exemple) et qu'il est parfois nécessaire d'intercaler un multiplicateur de fréquence pour l'obtenir.

L'amplificateur haute-fréquence permet d'amplifier le signal modulé pour pouvoir transmettre à des distances raisonnables. On définit le rendement n comme le rapport puissance de sortie P_s sur puissance fournie p_f par l'alimentation, le rendement idéal étant bien sûr égal à l'unité. Selon le mode de fonctionnement du transistor, des classes de fonctionnement ont été définies (tableau III-1).

Les rendements pratiques sont bien évidemment moins élevés (baisse de 20% en général). Il existe d'autres classes d'amplification (classes D, E, F, G, H par exemple), mais elles ne n'ont pas été étudiées ici car elles sont plus complexes, parfois protégées par des brevets et le gain de rendement ne serait pas substantiel par rapport à la complexité de la réalisation.

Un paramètre important pour caractériser un amplificateur est le facteur de distorsion.

Type d'amplificateur	type de liaison	rendement théorique
Classe A	à transformateur	0,5
	directe	0,25
	capacitive	< 0,25
Classe B	à transformateur	0,78
	directe capacitive	
Classe C	à transformateur directe capacitive	$0,78 < n < 1$

Tableau III-1 : Classes courantes d'amplificateurs radiofréquences

Nous sélectionnons la classe C qui présente un rendement de près de 70 % en pratique, ce qui satisfait notre critère de rendement en puissance. Par contre, il nécessite un filtre de sortie, ce qui implique une largeur de bande de fonctionnement fixe et étroite.

L'émetteur possédera un limiteur de consommation réalisé par une mise en veille (arrêt de l'émission HF).

L'adaptation permet de transmettre la puissance maximale à l'étage suivant, c'est-à-dire l'antenne. Dans le cas d'une désadaptation, la puissance perdue est éliminée par effet joule, ce qui diminue le rendement de l'émetteur. L'adaptation se mesure grâce au taux d'onde stationnaire. Le filtre adaptateur remplit donc ce rôle : adapter l'antenne à l'amplificateur haute-fréquence pour un rendement maximum de la chaîne d'émission.

2.2.2. Réception

Il existe deux types de chaîne de réception : le **changement de fréquence** et la **conversion directe (DC)**.

La figure III-4 représente le schéma classique d'un récepteur à double changement de fréquence.

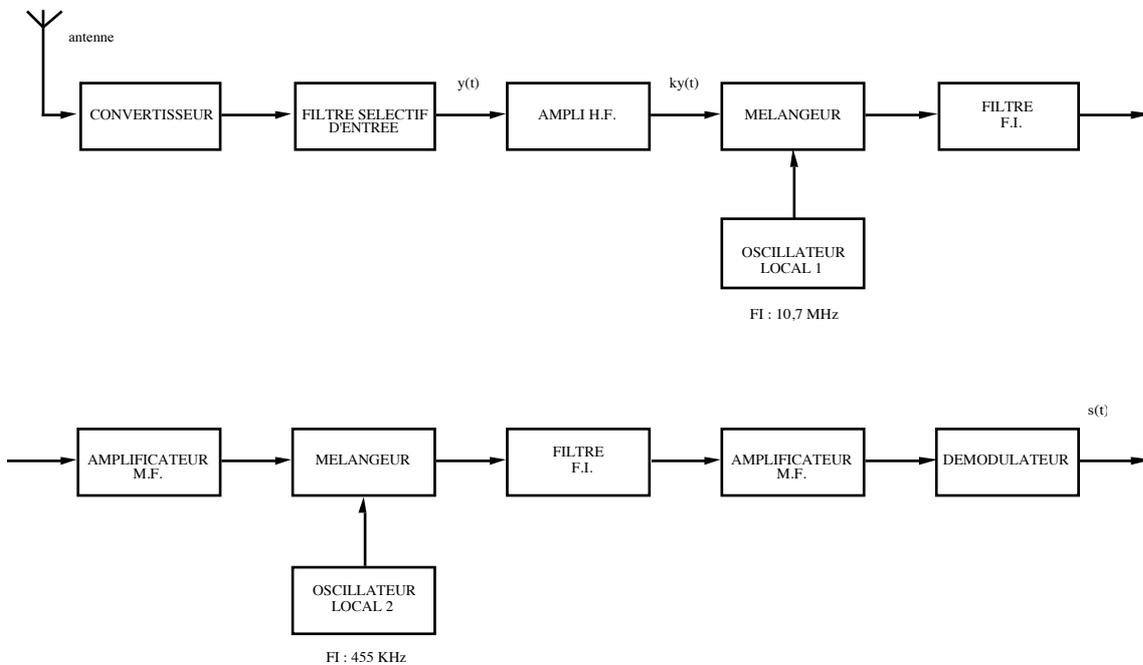


Figure III-4 : Synoptique de la chaîne réception version double changement de fréquence

L'avantage du système superhétérodyne est de permettre d'abaisser la fréquence de travail des circuits électroniques en translatant le spectre haute-fréquence vers le domaine fréquentiel de travail de l'oscillateur local, car il faut savoir que les filtres hautes fréquences sont difficiles à concevoir (réglage, dérive, etc.). Cette opération de translation s'appelle le **changement de fréquence**. De plus, le réglage de la fréquence ne se fait plus en modifiant la fréquence centrale du filtre passe-bande, mais en modifiant la fréquence de l'oscillateur local.

Deux changements de fréquence améliorent la qualité de réception. En sortie du mélangeur, on retrouve des fréquences images qu'il faut éliminer. Cette élimination se fait plus facilement à haute qu'à basse fréquence, pour une pente de filtrage donnée, par exemple pour une FI à 10,7 MHz, il faut rejeter les fréquences supérieures, en particulier à 21,4 MHz.

L'amplificateur haute fréquence n'est pas nécessaire si le signal reçu est d'amplitude suffisamment élevée.

Le choix des fréquences intermédiaires a été fait en fonction des standards en vigueur au niveau de l'émission-réception. Cette solution permet de sélectionner des composants standards du marché, en particulier les filtres (quartz, céramique ou LC).

La **syntonisation** est faite au niveau du premier mélangeur, l'avantage de cette solution est de limiter les bandes passantes des premiers filtres et amplificateurs FI car la syntonisation par le deuxième oscillateur demanderait une bande passante plus large à haute fréquence d'où une réalisation plus compliquée.

Il existe deux types de conversion :

- la conversion infradyne avec

$$f_{IO} < f_s \text{ donc } f_s = f_{IO} + 10,7 \text{ MHz,}$$

- et la conversion supradynne avec

$$f_{IO} > f_s \text{ donc } f_s = f_{IO} - 10,7 \text{ MHz.}$$

Un récepteur à conversion directe (DC) est un récepteur superhétérodyne dont la moyenne fréquence est nulle. La chaîne de réception s'en trouve simplifiée (figure III-5). Après deux blocs fonctionnels classiques (le filtre sélectif d'antenne et l'amplificateur haute-fréquence), un mélangeur translate le spectre du signal reçu dans la bande de fréquence du signal utile, le détecteur étant chargé de démoduler le signal.

Ce type de récepteur ne se prête qu'à la réception de signaux de télégraphie et de type bande latérale unique (BLU), et ne sera donc pas retenu.

Un récepteur est caractérisé principalement par sa sensibilité. La sensibilité est le niveau minimal d'un signal à l'entrée du récepteur qui permet d'avoir en sortie un rapport signal sur bruit et distorsion acceptable. Elle s'exprime en μV . Les récepteurs actuels ont une sensibilité de l'ordre du μV , voire inférieure pour une entrée adaptée en général à 50 Ω .

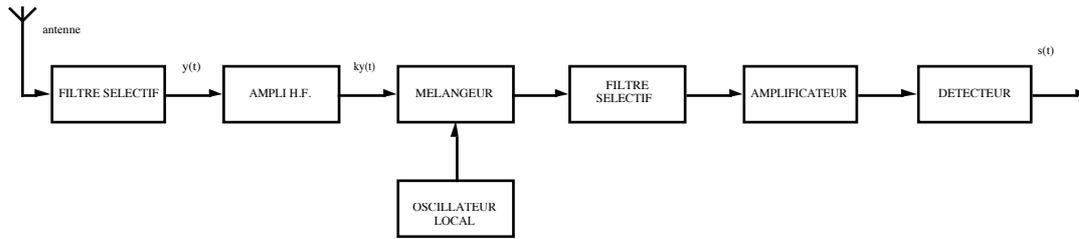


Figure III-5 : Synoptique de la chaîne réception version conversion directe

2.2.3. Antenne

L'antenne est le transducteur qui permettra le passage de l'information d'un milieu vers un autre milieu. Elle est assimilable à un filtre passe-bande et ne peut être adaptée pour une gamme large de fréquence (filtre à bande étroite). Cette antenne est reliée à l'émetteur ou au récepteur par un dispositif de couplage, qui peut comporter plusieurs sections destinées à l'adaptation. Ce dispositif est généralement un câble, mais peut être un guide d'onde, le choix étant lié à la fréquence de travail.

La détermination du canal et du transducteur a pour conséquence de modifier la modélisation de la communication, d'où la figure III-6.

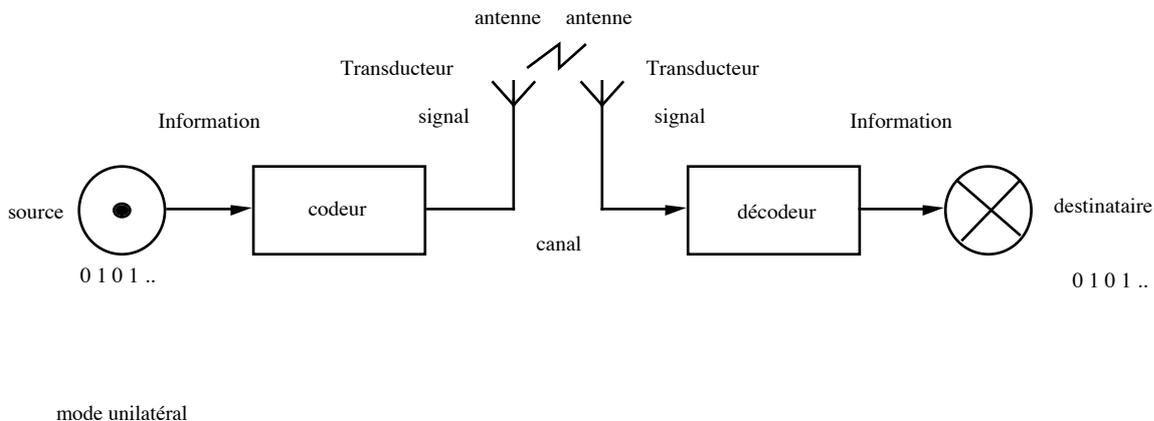


Figure III-6 : Synoptique d'une communication radio-fréquence

L'antenne est l'élément principal de la chaîne d'émission-réception en radio-fréquence. Elle peut être modélisée comme un dipôle. Son encombrement physique est lié à la longueur d'onde, et sera soit un sous-multiple, soit un multiple d'un quart d'onde. Comme elle doit rayonner l'énergie HF, l'antenne sera considérée comme un élément résonnant caractérisé par une fréquence propre qui est sa fréquence de résonance. L'antenne, la ligne de transmission et l'émetteur-récepteur seront adaptés pour obtenir la puissance maximale, ceci étant surtout vrai lors de l'émission, la chaîne de réception pouvant être adaptée ou avoir une entrée haute impédance. A l'adaptation, l'antenne est vue comme un élément résistif composé d'une résistance radiative (radiation resistance) en série avec une résistance de perte (loss

resistance). Dans le cas contraire, il y aura des pertes d'adaptation et des pertes dans le conducteur.

Ces caractéristiques électriques sont :

- son impédance à la fréquence de travail,
- sa bande passante référencée habituellement par rapport à son taux d'onde stationnaire (égale à 2), car une antenne est vue comme un filtre passe-bande,
- son gain dont le diagramme de rayonnement en est la représentation dans l'espace,
- sa polarisation qui est déterminée par la position de l'élément radiant par rapport à la terre. Les polarisations possibles sont les polarisations horizontale, verticale et elliptique (la polarisation circulaire étant un cas particulier).

Les dimensions physiques de l'antenne sont fonction de la fréquence de travail et de la bande passante désirée.

L'antenne doit répondre à des critères contradictoires qui sont les suivants :

- encombrement minimum,
- bande passante maximale,
- portée maximale,
- gain maximum dans les deux directions ox et oy .

Un préamplificateur d'antenne est parfois nécessaire. Cet élément joue deux rôles : il amplifie le signal provenant de l'antenne dans la bande fréquentielle de travail en atténuant les autres fréquences (amplificateur sélectif). Il permet aussi d'adapter les impédances d'entrée et de sortie (adaptateur d'impédance). Nous avons vu que l'antenne devait être adaptée à l'impédance de sortie de l'émetteur, ici 50Ω , pour réduire les pertes en ligne (rendement de l'émetteur) et éviter éventuellement la destruction de l'étage de sortie. Lorsque l'émetteur ne travaille pas dans la même bande de fréquence que le récepteur, et que ces bandes de travail ne sont pas multiples, l'antenne ne peut être adaptée aux deux éléments. Le préamplificateur peut servir alors d'isolateur en ayant une impédance d'entrée très élevée, et ainsi ne pas perturber l'émetteur. Il peut aussi présenter les caractéristiques additionnelles suivantes : sensibilité élevée et meilleure sélectivité, sans pour autant ajouter un facteur de bruit supplémentaire. Technologiquement, l'étage d'entrée est souvent confié à un transistor à effet de champs pour sa haute impédance d'entrée.

Les antennes retenues sont de deux types : une antenne flexible et une antenne télescopique. Ces antennes sont de longueur inférieure à $\lambda/2$ et présentent donc une réactance négative. Leur impédance est capacitive.

2.2.4. Duplexeur

Un duplexeur est constitué de deux filtres F1 et F2 (figure III-7). Si nous considérons une entité communicante de type 1, le filtre F1 présente l'atténuation la plus faible possible pour la bande fréquentielle B1 (faible perte d'insertion de l'ordre de 0,4 dB), et une très forte atténuation pour la bande fréquentielle B2 (de l'ordre de 90 dB), afin de ne pas perturber la

réception lors d'une émission. Le signal de l'émetteur doit représenter un bruit pour le récepteur, afin qu'il ne sature pas celui-ci et qu'il puisse être rejeté. Les filtres à réponse de Tchebysheff ou de type elliptique présentent les pentes d'atténuation les plus abruptes, mais présentent des ondulations dans la bande passante. Le délai de groupe doit être constant afin de ne pas rajouter de la distorsion de phase au signal utile. En prenant des filtres de type Butterworth qui présentent des pentes d'atténuation moyenne et une linéarité moyenne au niveau des temps de propagation, nous faisons un bon compromis.

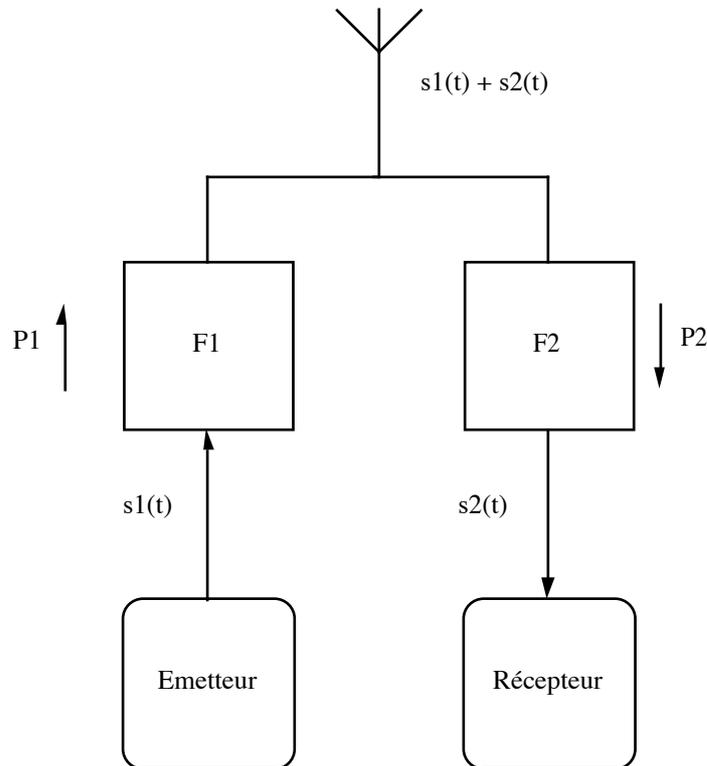


Figure III-7 : Duplexeur d'antenne

La puissance réjectée au niveau du récepteur doit être en-dessous de la sensibilité du récepteur.

Les filtres F1 et F2 peuvent être de tout type : passe-bas et passe-haut, passe-bande, ou coupe-bande.

2.3. Synthèse digitale de fréquence

Deux techniques de synthèse de fréquence sont usitées : la synthèse par boucle à verrouillage de phase et la synthèse numérique directe (SND) ou Direct Digital Synthesis (DDS). Les techniques de synthèse numérique directe émergent seulement du point de vue technologique.

2.3.1. Boucle à verrouillage de phase

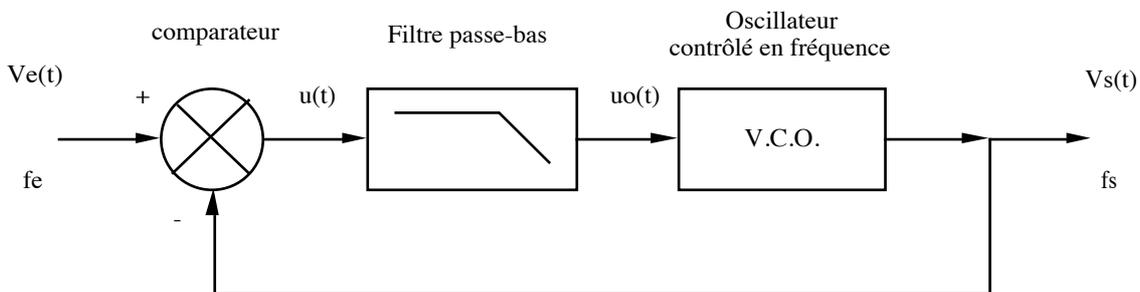
Les fréquences intermédiaires sont fabriquées au moyen d'une technique digitale appelée boucle à verrouillage de phase (ou PLL - Phase Locked Loop). Il s'agit en fait d'un asservissement de phase sur un oscillateur LC.

Nous avons étudié les principes d'émission-réception. L'émetteur nécessite un générateur de fréquence programmable dont le décalage en fréquence est contrôlé par le signal modulant. Le récepteur nécessite aussi un générateur de fréquence programmable pour pouvoir sélectionner la fréquence centrale f_0 du canal de réception. A cause de la largeur des canaux (quelques dizaines de KHz), les générateurs doivent présenter une grande stabilité en fréquence et seuls des oscillateurs à quartz (précision supérieure à 10^{-6}) le permettent.

Le problème soulevé est la programmation des fréquences. En effet, un quartz doit être affecté à chaque canal et le nombre de quartz est donc une fonction linéaire du nombre désiré de canaux. Les techniques digitales disponibles actuellement permettent de s'affranchir de ce problème. Elles sont basées sur l'utilisation du principe de la boucle à verrouillage de phase. Etudions d'abord cet élément fonctionnel avant de voir les différentes techniques de fréquence utilisant cet élément.

Le principe était connu depuis 1932 [De Bellescize 32], mais seul le progrès technologique d'aujourd'hui a permis de les utiliser.

Il s'agit d'un asservissement de phase. La phase d'un oscillateur de référence est comparée à celle du signal de sortie. Si un écart de phase, positif ou négatif, apparaît, alors un signal d'erreur est généré pour corriger l'écart.



- f_e : fréquence du signal d'entrée (signal de référence)
- f_s : fréquence du signal de sortie
- $V_e(t)$: signal d'entrée (référence)
- $V_s(t)$: signal de sortie
- $u(t)$: signal d'erreur de phase
- $u_o(t)$: signal d'erreur filtré

Figure III-8 : Schéma de base d'une PLL

Le schéma de base d'une PLL est représenté en figure III-8. Elle se compose de trois éléments :

- le comparateur de phase,
- le filtre passe-bas,
- l'oscillateur contrôlé en tension (VCO - Voltage Controlled Oscillator).

Le comparateur de phase fournit un signal $u(t)$ d'erreur de phase entre $v_e(t)$ et $v_s(t)$ lorsque la boucle est verrouillée ($f_e = f_s$) avec :

$$v_e(t) = v_e \sin \omega_e t$$

$$v_s(t) = v_s \sin (\omega_s t + \varphi)$$

Si les deux signaux ont même fréquence ($f_e = f_s$), l'erreur de phase est nulle. Dans le cas contraire, l'erreur de phase est :

$$(\omega_e - \omega_s) t - \varphi$$

Le filtre passe-bas, actif ou passif, est chargé de récupérer la composante continue $u_0(t)$ du signal d'erreur $u(t)$, en filtrant les harmoniques.

Ce signal continu d'erreur permet de piloter en fréquence l'oscillateur. Il s'agit donc d'un oscillateur contrôlé en fréquence. La relation qui lie la tension et la fréquence est en général linéaire et de la forme :

$$f_s = f_0 + k u_0(t)$$

- f_0 est appelée la fréquence centrale et est déterminée lorsque $u_0(t)$ est égale à 0 Volt. On dit que l'oscillateur est vobulé.
- k est la sensibilité de l'oscillateur et s'exprime en Hz/V.

Ce générateur est en général peu stable car il est basé sur le principe du circuit oscillant LC (self - condensateur). Mais lorsque cet oscillateur intervient à l'intérieur de la boucle, il est en permanence corrigé par le signal d'erreur du comparateur et la fréquence ainsi générée est très stable. A ces éléments de base, peuvent être ajoutés des diviseurs de fréquence et un amplificateur, mais leur rôle sera précisé dans le paragraphe suivant. Il faut tout de même savoir que les diviseurs permettent de programmer les fréquences de comparaison et de sortie, et que l'amplificateur adapte les caractéristiques de transfert du comparateur de phase à celle de l'oscillateur contrôlé en tension. La figure III-9 représente le nouveau synoptique.

Deux caractéristiques importantes sont à retenir :

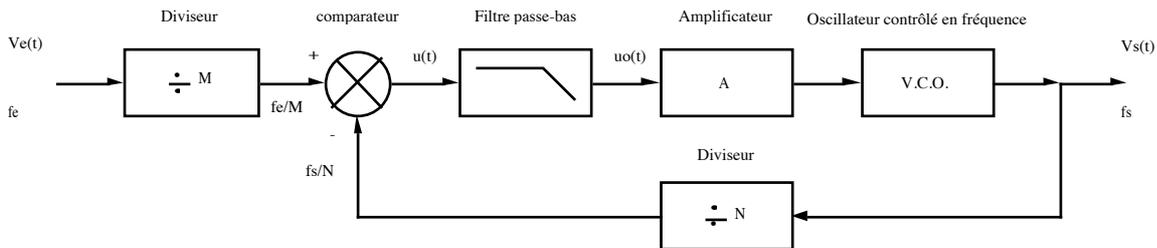
- la plage de verrouillage (ou de maintien, de poursuite (tracking range, lock range)) qui se définit comme la bande de fréquence pour laquelle la PLL reste verrouillée,

- la plage de capture (d'accrochage (capture range, acquisition range, pull in range)) qui est la bande de fréquence pour laquelle la PLL se verrouille.

Nous avons toujours la relation suivante :

$$\text{plage de capture} \leq \text{plage de verrouillage}$$

Les plages pré-citées sont centrées en f_0 .



f_e : fréquence du signal d'entrée (signal de référence)
 f_s : fréquence du signal de sortie
 $V_e(t)$: signal d'entrée (référence)
 $V_s(t)$: signal de sortie
 $u(t)$: signal d'erreur de phase
 $u_o(t)$: signal d'erreur filtré
 M, N : pas de division
 A : gain de l'amplificateur

Figure III-9 : Schéma de base d'une PLL à diviseur

2.3.2. L'oscillateur contrôlé en fréquence

Le circuit de base est un circuit bouclé auto-oscillant qui oscille à la fréquence :

$$f_0 = 1 / 2\pi (LC)^{-1}$$

Une diode varicap permet de faire varier la fréquence de l'oscillateur grâce à sa capacité interne. En effet, ce type de composant présente une capacité variable qui est fonction de la tension inverse qui lui est appliquée. Cette tension est bien sûr la tension d'erreur, filtrée et amplifiée si nécessaire, du comparateur de phase. La fréquence d'oscillation f_0 devient :

$$f_0 = 1 / 2\pi (LC_{eq})^{-1}$$

$$\text{avec } C_{eq} = C + (C_1 // C_{D1})$$

2.3.3. Principe de la synthèse de fréquence

Nous n'avons pas retenu la synthèse directe qui n'utilise pas les boucles à verrouillage de phase et qui donc présente un nombre élevé de composants. Ce type de synthèse utilise les opérateurs fréquentiels suivants :

- l'addition, la soustraction,
- la multiplication, la division.

Elle est surtout présente dans les appareils professionnels.

Nous proposons la deuxième solution qui se nomme la synthèse indirecte. Elle consiste à utiliser un diviseur dans la boucle de retour pour réaliser une multiplication fréquentielle.

Il faut distinguer deux types de synthèse :

- la synthèse à sortie directe (principe utilisé),
- la synthèse à sortie indirecte.

Le principe du générateur programmable de fréquence est un oscillateur LC par exemple à l'intérieur de la boucle d'asservissement de phase.

Un circuit LC est un circuit bouclé auto-oscillant (circuit bouchon) dont la fréquence d'oscillation est donnée par la formule :

$$f_0 = 1 / 2\pi LC^{-1}$$

Malheureusement, la fréquence d'oscillation n'est pas très stable. La PLL va essayer de rattrapper les erreurs du circuit oscillateur en comparant sa phase avec un oscillateur de référence. La référence en fréquence est un quartz; ce composant présente une erreur de l'ordre de 10^{-6} sur sa fréquence d'oscillation et sa stabilité dans le temps est très grande. La composante continue d'erreur va ensuite polariser une diode varicap qui a la propriété de faire varier sa capacité en fonction de la tension inverse qu'on lui applique. Ainsi on modifie la capacité totale du circuit oscillant, et donc sa fréquence. La présence de diviseur de fréquence permet de faire osciller l'ensemble à des fréquences autres que la référence, et comme il s'agit de registres programmables, la fréquence d'oscillation l'est aussi. Noter que le circuit diviseur en sortie de VCO peut être remplacé par un mélangeur, ce qui permet de baisser la fréquence de travail et donc la consommation.

Les circuits intégrant des PLL ne fonctionnent pas à très haute fréquence. Pour travailler au-delà de la fréquence maximale, il est nécessaire de mettre en oeuvre le principe du mélangeur ou du diviseur de fréquence. Cet artifice permet de ramener la fréquence de travail dans la plage de travail autorisée de la boucle d'asservissement.

la fréquence de sortie est donnée par la formule :

$$f_s = f_0 + N / M f_e$$

on veut : $f_e / M = f_s / N \implies f_s = N/M f_e$

L'utilisation d'une boucle à verrouillage permet de synthétiser des fréquences avec un seul quartz en changeant uniquement les coefficients de la boucle par programmation. Cela évite d'avoir tout un ensemble de quartz, un par canal désiré. Cette technique a permis de simplifier l'émission-réception et d'obtenir un gain de place d'autant plus élevé que le nombre de canaux l'était.

2.3.4. Synthèse numérique directe

La synthèse numérique directe est une technique en pleine expansion. Elle consiste à synthétiser une onde sinusoïdale à partir de données numériques stockées dans une mémoire vive ou morte. Ces données sont présentées périodiquement à un convertisseur numérique/analogique (N/A) et le signal analogique ainsi généré est ensuite filtré grâce à un filtre anti-repliement (passe-bas). Le signal devient identique à celui généré par une PLL.

L'intérêt de cette technique est d'effectuer le traitement du signal, en particulier la génération et la modulation, de manière entièrement numérique. La technique digitale étant employée, la dérive des caractéristiques du signal (amplitude, phase, fréquence, distorsion, etc.) en fonction de paramètres comme le temps ou la température est négligeable. Cela tient au fait que très peu de composants passifs et actifs analogiques interviennent dans la chaîne de synthèse. La résolution en fréquence est très élevée ($\gg 1$ Hz !) et le verrouillage en fréquence extrêmement bref. Au niveau technologique, la reproductibilité est idéale et il n'y a pas de réglage délicat.

Par contre, la fréquence de travail ne dépasse pas actuellement quelques dizaines de MHz. La consommation électrique est plus élevée que pour des techniques mi-numériques mi-analogiques comme la PLL, et ceci malgré l'utilisation d'une technologie C-MOS (les technologies C-MOS et bipolaire se rejoignent au point de vue consommation lorsque la fréquence de travail des circuits est élevée). Des erreurs, comme les erreurs de quantification ou les erreurs de non linéarité, sont générées, dues principalement à la conversion N/A. Un bruit de phase peut aussi être généré à cause de l'instabilité de l'horloge système.

3. Etat de l'art des réseaux radio-fréquences

Les progrès réalisés en électronique haute-fréquence, en particulier grâce à des techniques numériques comme la boucle à verrouillage de phase et la synthèse numérique directe ont permis d'améliorer l'électronique RF embarquée en ce qui concerne :

- les performances techniques (sensibilité des récepteurs, facteur de bruit, etc.),
- la consommation électrique,
- l'encombrement physique.

Ainsi des systèmes portables de radio-communications autonomes du point de vue énergétique ont pu être développés.

Actuellement, le secteur des radiocommunications, c'est-à-dire la téléphonie analogique, la téléphonie numérique et les réseaux numériques, du point de vue économique et industriel, est en pleine expansion. Deux catégories de systèmes existent : ceux pour communiquer à faible distance, dans un local par exemple, et ceux pour communiquer à grande distance. Dans la première catégorie se trouvent les réseaux locaux radio-fréquences (WLAN : Wireless Local Network), et dans la seconde, sont rangés la radio-téléphonie de proximité (systèmes CT2, DECT, etc.), la radio-téléphonie cellulaire (systèmes GSM (Groupe Spécial Mobile), DCS 1800, etc.), les réseaux radioélectriques (le futur système Tetra (Trans European Trunked Radio), le système radio-amateur packet radio, etc.). Lorsque ces réseaux sont destinés au transport de données, ils sont désignés 3RD (Réseau Radio-fréquence Réservé aux Données).

Ces réseaux sont tous à commutation de messages ou de paquets, le réseau à commutation de circuits n'étant absolument pas adapté à ce média. Les WLANs utilisent la plupart du temps un seul canal pour plusieurs utilisateurs et le type d'accès est alors en général du type Aloha ou Slotted-Aloha.

Dans les systèmes de communication à grande distance, les canaux fréquentiels sont partagés entre plusieurs utilisateurs. La zone géographique de communication est partagée en cellules, afin de limiter la puissance d'émission et d'éviter des interférences. Un centralisateur par cellule alloue une ou plusieurs fréquences à un mobile, et cette allocation ne change pas pendant la communication. Les cellules sont reliées entre elles par lignes spécialisées. Lorsqu'il y a une seule cellule, nous sommes en général en présence de systèmes privés (réseaux locaux). Ce type de réseau est nommé 3RP (Réseau Radio-fréquence à Ressource Partagée).

Il existe en radiocommunication cinq types de communications **[Rémy et al. 88]** : en communication bilatérale, le duplex, le simplex ou l'alternat (half-duplex) et en communication unilatérale, la communication du mobile vers un site fixe et inversement (tableau III-2).

Dans ces réseaux radio-fréquences, les voies de communication doivent avoir les fonctions suivantes :

- voie d'appel,
- voie de demande de connexion,
- voie de trafic,
- voie d'échange de signalisation.

Ces voies peuvent être partagées temporellement ou fréquentiellement.

Trois types d'accès existent:

- Accès Multiple à Répartition de Fréquence (AMRF) ou Frequency-Division Multiple Access (FDMA),
- Accès Multiple à Répartition dans le Temps (AMRT) ou Time-Division Multiple Access (TDMA),
- Accès Multiple à Répartition dans les Codes (AMRC) ou Code-Division Multiple Access (CDMA).

type de communication	nombre de voies	mode de communication	application type
bilatérale simplex	1	alternat	canal à collision
bilatérale duplex	2	simultané	téléphonie mobile
bilatérale alternat	2	alternat	relais
unilatérale : fixe vers mobile	1	-	radio-messagerie
bilatérale : mobile vers fixe	1	-	téléalarme

Tableau III-2 : Type de communication en radiofréquences

Dans le premier type d'accès, une ou plusieurs fréquences sont allouées à un utilisateur. Dans le cas d'un accès de type AMRT, un utilisateur a un intervalle de temps alloué sur un canal à grand débit. Pour le dernier cas, l'utilisateur a une succession d'intervalles (temps, fréquence) alloués.

Lorsque le média est partagé, les protocoles rencontrés sont :

- CSMA/CD (Carrier Sense Multiple Access / Collision Detect),
- CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance)

Ces protocoles sont utilisés lorsque plusieurs noeuds partagent l'accès au même canal. Dans le premier cas, l'émetteur émet et détecte une collision éventuelle de son message avec ceux d'autres locuteurs. Dans le deuxième cas, il évite la collision par une écoute préalable.

3.1. Les normes

En France, en ce qui concerne les transmissions de données, les deux premières normes qui régissaient les communications radio-fréquences concernaient les systèmes de transmission à bande étroite ($\Delta f = 10$ kHz), il s'agit des normes 123 (janvier 1980) et 1382 (Avril 1988) éditées par le CNET.

Ces normes ont été éditées pour des applications professionnelles de type télécommande, télémessure et transmission de données.

La télécommande de modèles réduits ((26 - 41 - 72 -144 et 436 MHz) - P max. = 5 Watts) n'est pas soumise à une homologation, mais à une licence. L'utilisation professionnelle est interdite.

norme	puissance apparente rayonnée (PAR) en mW	vitesse max en bauds	protocole imposé	licence
123	< 100 > 100	9600	non oui 5 mn d'arrêt pour 20 s d'émission	oui
1309	< 5		non	
1542	< 5 < 100	38400	non	non
1382	< 2000	1200	oui avec silence	oui

Tableau III-3 : Caractéristiques des principales normes de radio-transmission de données

norme	bande de fréquence	largeur de canal	date
123	30,525 à 40,6 68,4625 à 82,475 151 à 161,475 406,1 à 428 441,500 à 470	canal de 12,5 KHz	janvier 1980
1309	12 fréquences entre 30 et 500 MHz		
1382	450-470 MHz		avril 1988
1542	223,5 - 225 MHz	6 canaux de 200 KHz	mai 1985

Tableau III-4 : Caractéristiques des principales normes de radio-transmission de données (suite)

Il n'existe pas de normes mondiales pour les réseaux radio-fréquences. La réglementation mondiale concerne uniquement l'allocation de fréquences. Les tableaux III-3 et III-4 non exhaustifs donnent pour information les caractéristiques des principales normes françaises pour la transmission de données.

En ce qui concerne la communication infra-rouge, aucune norme nationale n'existe. Des institutions de normalisation travaillent actuellement sur la normalisation des réseaux locaux. Il s'agit en particulier de l'IEEE (Institute of Electrical and Electronics Engineers) avec le comité 802.11 qui s'intéresse aux couches matérielles (1) (MAC : Medium Access Control) et liaison (2) (LLC : Logical Link Control) du modèle OSI/ISO pour les réseaux locaux.

3.2. Présentation de réseaux

Pour illustrer notre propos, nous présentons quelques réseaux radio-fréquences représentatifs des techniques et des applications futures.

3.2.1. Le réseau DECT

Le téléphone numérique sans fil européen DECT (Digital European Cordless Telecommunication) est un réseau radio-fréquence à bande étroite. C'est un système orienté connexion comme pour le téléphone. Il est composé de noeuds centraux (HUB) qui supportent jusqu'à quatre stations de base et une station de base gère un utilisateur à 400 kbits/s (couverture : 50 à 200 m). Chaque noeud central est relié par un câble de type "paire torsadée non blindée" et forme un réseau de type bus à collision Ethernet ou de type anneau token ring.

Chaque station de base transmet et reçoit sur une des dix fréquences, chacune fournit un canal à 1,152 Mb/s. Chaque canal est divisé en 24 intervalles temporels de 48 Kbits/s. Le protocole utilisé est le Time Division Duplex/Time Division Multiple Access (TDD/TDMA). Cette technique de multiplexage utilise des intervalles de temps séparés sur une porteuse pour transmettre et recevoir les signaux. Un intervalle dure 10 ms, 5 ms pour la transmission et 5 ms pour la réception. Ces sous-intervalles sont divisés en 12 intervalles de temps de 480 intervalles temporels (416,67 ms). Les intervalles de temps sont alloués par paire (émission/réception). Un utilisateur peut utiliser jusqu'à 8 paires d'intervalle de temps (==> 384 Kbits/s pour émettre et 384 Kbits/s pour recevoir). La modulation employée se nomme Gaussian-Shaped Frequency Shift Keying (GSFSK). Le taux de transfert est de 1,152 Mb/s et la largeur du canal (1,728 MHz) limite l'efficacité spectrale à 0,67 bit/s/Hz. La bande de fréquences allouée va de 1,88 GHz à 1,9 GHz, et la puissance d'émission est de 250 mW.

3.2.2. Le réseau cellulaire digital

La technique CDPD (Cellular Digital Packet Data) utilise des canaux de largeur 30 KHz, comme pour le réseau cellulaire vocal. Une telle largeur fréquentielle de canal offre un débit de 19200 bauds. Cette technique sacrifie le débit pour le taux d'erreur. En comparaison d'un protocole filaire V.32 qui offre 4 bits par baud, le CDPD offre 1 bit par baud, ceci est lié au canal utilisé très bruité. Le code de correction d'erreurs utilisé est un code basé sur Reed-Salomon, qui permet une détection-correction sans retransmission de la donnée. Un protocole d'identification et de cryptage est aussi inclus. La contrepartie est un débit peu

élevé de l'ordre de 10 Kbits/s. Les stations de base CDPD s'additionnent à celle qui gère le réseau vocal. Une station de base CDPD gère tous les canaux disponibles. Le principe de communication est le suivant : dès qu'une station de base reçoit une demande de communication, elle alloue un canal non occupé et envoie les données, mais une demande pour un canal vocal est prioritaire et peut retirer l'allocation pour un envoi digital.

3.2.3. La norme Tetra

Cette norme concerne les réseaux radio-électriques privés qui véhiculeront la voix et les données. Au niveau données, deux modes existent : les modes circuits protégé et non protégé, et les modes paquets en mode connecté (X25), et non connecté. La norme n'est pas finalisée, mais la modulation serait de type QPSK ($\pi/4$), avec un accès TDMA. La largeur de bande serait de 25 KHz et le débit des données en mode circuit de 19200 Kbits/s.

3.2.4. Le réseau domotique CEBus

L'Association de l'Industrie Electronique (Electronic Industry Association - EIA) a choisi le protocole SSC comme standard de communication radio-fréquence pour le CEBus (Consumer Electronics Bus), réseau domotique qui utilise divers médias : radio-fréquence, réseau électrique usager, infra-rouge, câble électrique (coaxial ou paire torsadée) et fibre optique.

Une trame commence par un train court en modulation OOK (On Off Keying) pour résoudre les problèmes d'accès au canal. Une fois le problème de contention résolu (accès résolu), le paquet est envoyé en utilisant une modulation PSK (Phase Shift Keying). Deux types de paquets ont été définis : un contenant un champ adresse et des informations de contrôle, et l'autre contenant un champ adresse et des données.

Lors de la transmission, l'énergie du signal est répartie uniformément selon la technique à étalement de spectre.

4. Sous-réseau d'acteurs mobiles

Nous désirons, pour les mobiles, une communication asynchrone multi-locuteurs. Nous verrons plus loin quelles sont les conséquences de ce choix. La vitesse de modulation choisie est celle d'une liaison série asynchrone classique et varie donc de 1200 bauds à 19200 bauds.

4.1. Spécificités d'ActNet

Les problèmes techniques sont plus complexes que pour un réseau filaire. Le réseau est ouvert, n'importe qui peut se brancher sur le média pour écouter ou perturber la communication (brouillage ou fausses données).

D'un point de vue physique, le médium de communication est soumis en permanence à des perturbations variables. Le milieu de transmission entre acteurs mobiles est l'atmosphère.

La propagation en milieu atmosphérique est difficile à mettre en oeuvre à cause des perturbations extérieures intervenant lors de communications et des modifications provoquées par le canal lui-même (distorsion, atténuation, etc.). Des hypothèses simplificatrices seront faites tout au long de notre discours concernant la propagation et les perturbations éventuelles, du fait des distances mises en jeu entre acteurs. Le périmètre d'évolution des acteurs sera le laboratoire.

Pour communiquer sans lien physique, de part le canal, plusieurs solutions caractérisées par le domaine fréquentiel sont envisageables [**Darche 90**] :

- la liaison sonore,
- la liaison ultra-sonore,
- la liaison infra-rouge cohérente (laser) ou non,
- la liaison par onde lumineuse cohérente (laser) ou non,
- la liaison radio-fréquence.

Les solutions non envisageables sont les suivantes :

- la liaison ultra-sonore, à cause de la fréquence de travail. En général, les transducteurs travaillent entre 40 KHz et 200 KHz. Ils sont trop directifs.
- la liaison de type laser demande trop de directivité,

Les deux solutions à retenir sont la liaison infra-rouge et la liaison radio-fréquence.

Les ondes infra-rouges, généralement plus lentes, ne traversent pas les objets opaques, aucune législation n'existe à ce jour et elles ne sont pas sujettes à des perturbations d'ordre radio-électrique, la portée est limitée.

Nous n'avons pas retenu la liaison infra-rouge pour un problème de distance entre les robots. En effet, dans un environnement industriel, les robots peuvent être séparés par des obstacles opaques au rayonnement infra-rouge comme un mur. De plus, ce type de transmission est relativement sensible à l'environnement. Un simple éclairage peut perturber la communication. Par contre, en environnement scolaire, le micro-monde se trouve dans un seul local : la salle de classe. Ainsi, une communication infra-rouge est envisageable.

Les quatre premières solutions posent le problème de la directivité et ont été écartées. Nous avons donc retenu la liaison radio-fréquence, et par conséquent, la propagation électromagnétique. Le transducteur sera donc une antenne.

Les ondes radio-électriques traversent tous les objets non métalliques, sont sujettes à des perturbations radio-électriques et doivent respecter la législation des radio-communications, sauf dans le cas du réseau à spectre étalé.

L'architecture adoptée permet de s'adapter à n'importe quelle solution envisageable. Seule les modules de communication sont à modifier.

Nous désirons, pour les mobiles, une communication asynchrone multi-locuteurs. La vitesse de modulation est celle de la liaison série asynchrone équipant la carte générique de

contrôle, et celle-ci, réglable, varie de 1200 à 19200 bauds. Un codage de canal est nécessaire pour adapter l'information à transmettre au canal. Dans notre cas, pour obtenir une communication multi-locuteurs, il est nécessaire de distinguer les informations provenant des différents interlocuteurs. Une modulation analogique sera donc employée. Ainsi, l'espace des fréquences sera découpé en bandes identiques appelées canaux radio-fréquentiels et une bande sera affectée à chaque acteur du SRAM, qui sera donc caractérisé par une adresse fréquentielle.

Pour des raisons d'occupation spectrale, une modulation angulaire discrète de type fréquentiel (FM), la modulation par déplacement de fréquence minimum (MFSK), a été sélectionnée, car elle optimise la largeur de bande nécessaire à la transmission d'un signal digital, sans pour autant nécessiter un codage de source élaboré.

Nous utiliserons la bande métrique qui comprend les ondes de longueur d'onde comprise entre 10 m et 1 m, c'est-à-dire les ondes VHF (Very High Frequency - de 30 MHz à 300 MHz).

Ce choix se justifie par les faits suivants :

- plus la fréquence augmente, plus la propagation est réduite et directe,
- plus la fréquence augmente, plus la conception devient difficile :
 - la technologie devient spécifique,
 - le schéma devient moins reproductible,
 - les exigences de précision et de stabilité des oscillateurs sont plus sévères, de plus l'effet Doppler devient sensible,
 - il faut tenir compte des capacités et selfs parasites des composants et de la platine de montage,
 - les réglages sont plus difficiles, et des perturbations internes interviennent,
- l'antenne est moins encombrante dans cette bande puisque sa longueur est fonction de la longueur d'onde du signal à véhiculer,
- le plan de fréquence est très chargé, et la bande de fréquence sélectionnée présente quelques canaux disponibles pour les civils (en particulier, vers 27, 41 et 72 MHz). Mais le risque de perturbations de stations professionnelles d'émission-réception est grand, même si nous avons pris comme hypothèse de départ le fait que le laboratoire était une cage de Faraday. Nous devons, autant que faire se peut, réduire la portée (50 m maximum) pour éviter les perturbations éventuelles. Ceci se fera en réduisant la puissance d'émission. De toute façon, s'il le fallait, nous pourrions prendre une licence (coût réduit de l'ordre de 100 F) pour nous mettre en règle avec la législation.

Cette bande représente donc un bon compromis.

Nous avons fait travailler les communicateurs dans des bandes de fréquences peu élevées pour une facilité de mise au point de l'électronique, et à cause de la législation, nous avons donc sélectionné :

- B₁ = [40 , 45] MHz,
- B₂ = [70 , 75] MHz.

La puissance d'émission est inférieure à 5 mW et nous n'avons pas besoin d'être licencié. Dans le cas d'une portée plus grande, la largeur des bandes serait ramenée à 1 MHz et les bandes fréquentielles utilisées seraient celles réservées aux radio-modélistes avec nécessité d'une licence d'où :

- $B_1' = [41, 42]$ MHz,
- $B_2' = [72, 73]$ MHz.

Les quatre topologies de base les plus couramment employées au niveau d'un réseau local câblé sont le bus, l'étoile (Star Network), l'arborescence (Tree Network), et l'anneau (Ring Network). Des topologies hybrides peuvent être envisagées, mêlant les topologies de base, comme le maillage (Meshed Network). Beaucoup de ces topologies amènent à concevoir un système hiérarchique.

Lorsqu'un mobile se déplace, la communication peut être perturbée principalement par trois phénomènes physiques [**Rémy et al. 88**] qui sont :

- l'effet Doppler fonction de la vitesse du mobile et qui agit sur la fréquence du signal,
- le fading de Rayleigh qui agit sur l'amplitude du signal, dû à des ondes réfléchies ou diffractées, entraînant des évanouissements de signal, dont le temps est inversement proportionnel à la fréquence de travail,
- une dispersion des temps de propagation.

Ces phénomènes engendrent donc respectivement une distorsion de fréquence, d'amplitude et de phase. Pour notre application, nous négligeons ces distorsions sachant que le mobile se déplace à faible vitesse et que son espace de mobilité est le laboratoire.

4.2. Interface d'entrée-sortie

Quelle interface d'entrée-sortie faut-il choisir ? La liaison série s'impose d'elle-même car l'interface parallèle multiplierait d'autant plus le nombre de canaux (pour notre cas : 8). Faut-il utiliser l'interface série asynchrone classique ou les liens du transputer ? Les liens semblent exclus car le taux de transfert des données, par l'intermédiaire de ceux-ci, nécessiterait une bande passante trop élevée, incompatible avec la largeur des canaux disponibles (quelques dizaines de KHz). De plus, le protocole de communication étant câblé (principe du rendez-vous avec accusé de réception), le fait de rajouter un élément communicateur supplémentaire (l'émetteur/récepteur) entraînerait une désynchronisation qui serait difficilement récupérable. Il faut donc utiliser la liaison série classique.

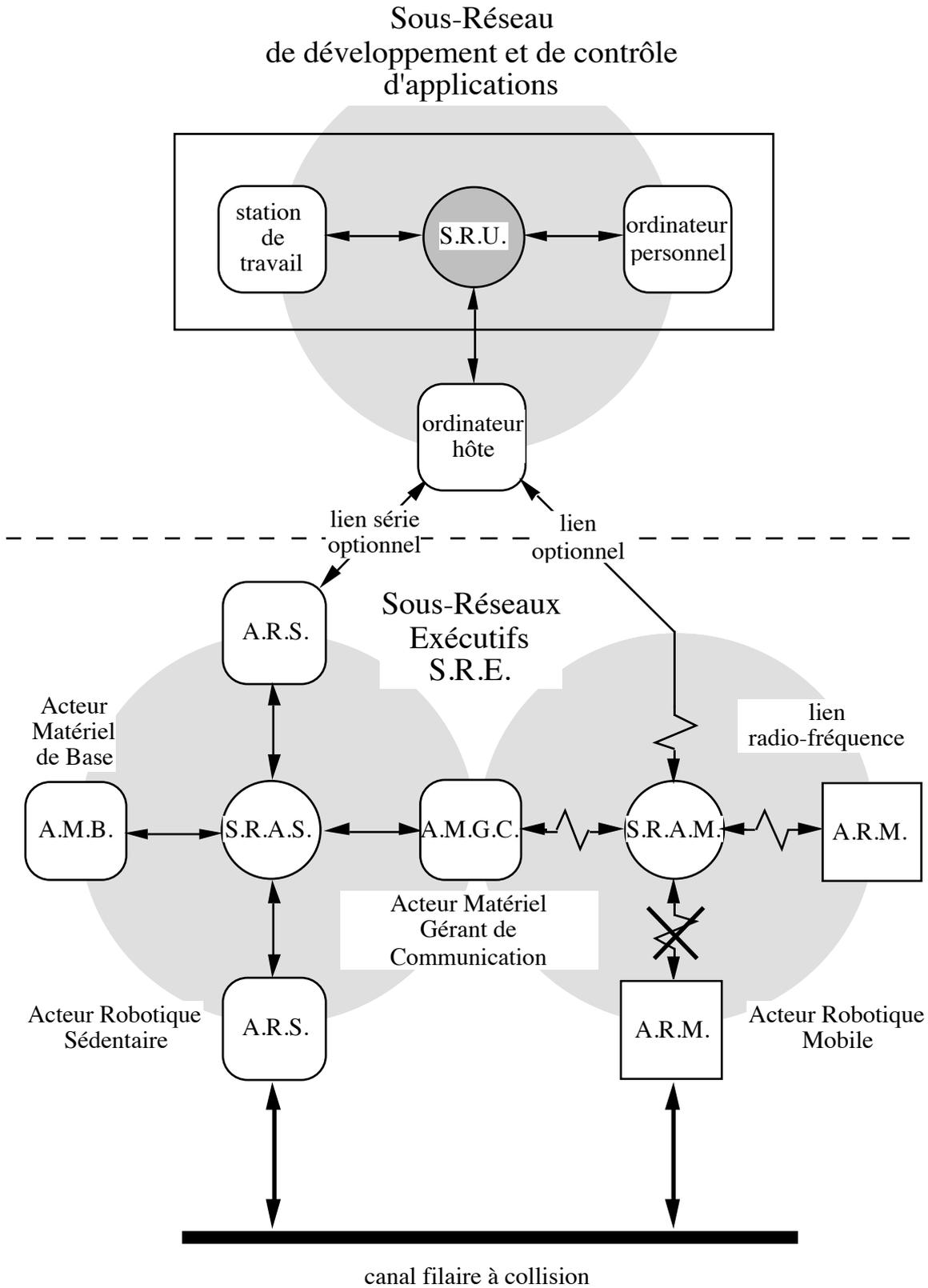


Figure III-10 : Canal de secours (1er cas)

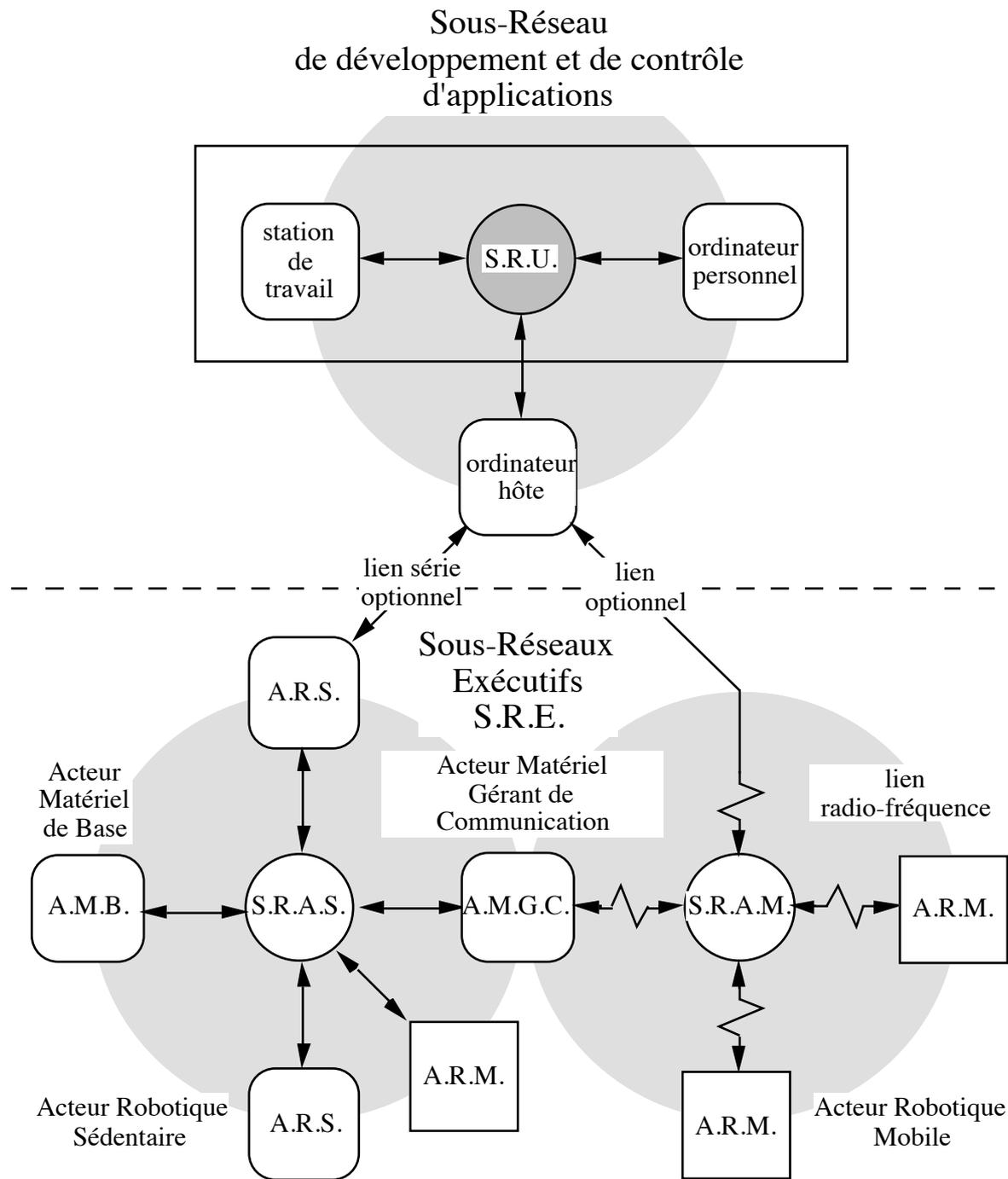


Figure III-11 : Canal de secours (2ème cas)

L'étude particulière du SRAM a été conduite en prenant en compte la tolérance aux pannes. La communication par câble avec les robots mobiles est à considérer seulement dans les cas suivants :

- quand une panne arrive dans le système d'énergie embarquée, une source d'énergie externe est nécessaire pour l'alimentation des cartes électroniques et pour recharger les batteries, la ligne d'alimentation du réseau remplirait ainsi pleinement son rôle,
- dans la phase de développement pour un télé-chargement rapide d'un logiciel prototype ou pour la mise au point des applications,
- quand une panne du système radio-fréquence survient.

Lors d'une connexion filaire, la liaison série peut bien sûr être utilisée comme un canal à collision (figure III-10). Un acteur mobile peut alors se connecter à un acteur matériel sédentaire, celui-ci faisant office de passerelle. Mais il serait intéressant d'utiliser les liens en incluant le mobile dans le sous-réseau en anneau (SRAS), puisque celui-ci est équipé d'une carte générique basée sur le transputer. Ce deuxième cas est illustré par la figure III-11.



Photographie III-1 : Récepteur

4.3. Choix technologiques

Chaque émetteur/récepteur sera doté de deux synthétiseurs de fréquence du type boucle à verrouillage de phase (PLL), pour permettre la programmation digitale des fréquences d'émission et de réception. Ainsi, chaque acteur pourra communiquer avec n'importe quel autre acteur du SRAM, à partir de l'instant où il connaîtra les adresses fréquentielles de ses interlocuteurs. Le circuit intégré utilisé est le MC145170 de la société Motorola.

La deuxième contrainte est l'autonomie en énergie qui impose une limitation en puissance de l'émission, pour limiter la consommation électrique, et donc la portée. L'émetteur sera équipé d'un amplificateur RF de classe C pour augmenter le rendement de transmission et donc la consommation, pour une puissance d'émission donnée. A la demande de l'acteur, l'émetteur pourra se mettre en veille de consommation.

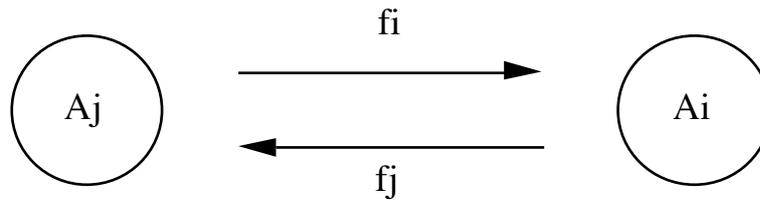
Le récepteur sera de solution intégrée (photographie III-1), en l'occurrence le circuit intégré MC3362. Il s'agit d'un récepteur à double changement de fréquence pour une sélectivité élevée et un rapport signal/bruit amélioré.

4.4. Types de communication

Nous allons étudier dans cette partie les différentes topologies possibles dans ActNet en fonction du type de communication et du nombre d'antennes. Nous considérons un et un seul récepteur et émetteur par acteur mobile. La conception des éléments communicants radio-fréquences a été modulaire. Les entrées-sorties ont été adaptées à l'impédance caractéristique des câbles (50Ω) pour avoir le minimum de pertes en ligne et le maximum de puissance fournie à l'étage suivant. Toutes les connections se font par des connecteurs et il est ainsi aisé de changer rapidement de modules. Cette souplesse permet à l'utilisateur de choisir sa configuration, d'un système de communication à une antenne à un système de communication à deux antennes. Il existe ainsi trois configurations possibles : une configuration multi-bandes à une antenne avec duplexeur, une configuration mono-bande à deux antennes et une configuration mono-bande à une seule antenne et un commutateur d'antenne. Nous allons étudier ces 3 cas et les topologies qui peuvent être réalisées. La communication selon le cas peut alors s'effectuer en duplex, en alternat ou en simplex. Pour notre étude, nous considérons qu'il pourra exister au maximum deux bandes de fréquences B_1 et B_2 à notre disposition et qu'un seul système de communication (émetteur et récepteur) équipe un acteur.

4.4.1. Communication duplex à une antenne

Nous considérons l'émission-réception en duplex à une antenne. La contrainte technique majeure est l'écart duplex Δf_{duplex} . En effet, un certain écart fréquentiel doit être respecté pour que l'émission ne perturbe pas la réception par saturation du récepteur. Et si le signal d'émission a une énergie trop élevée, cela pourrait entraîner la destruction du récepteur. Il faut donc intercaler un filtre sélectif entre l'antenne, l'émetteur et le récepteur pour séparer le signal de l'émission de celui de la réception. Le duplexeur est un élément passif ou actif qui réalise cette fonction. Cet élément reste d'un coût élevé et les dimensions physiques dépendent en général de la longueur d'onde.

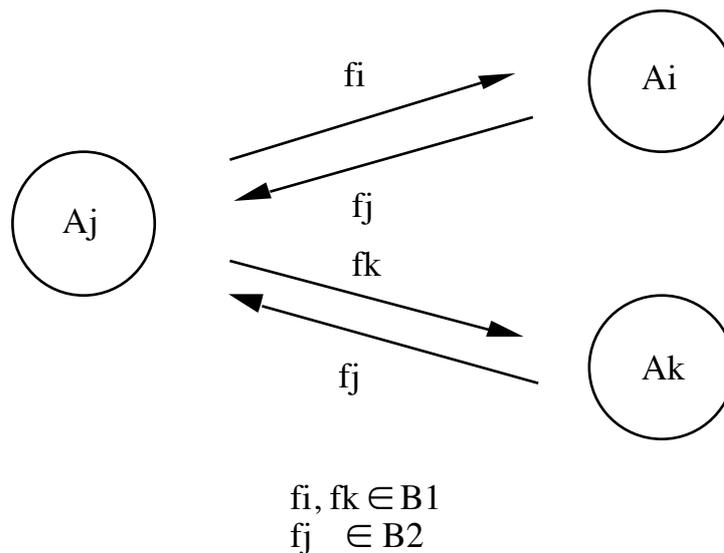


$$f_i \in B_1, B_1 = [40,45] \text{ MHz}$$

$$f_j \in B_2, B_2 = [70,75] \text{ MHz}$$

Figure III-12 : Communication en duplex à une antenne

Nous considérons donc un système de radio-communication bi-bandes B_1 et B_2 . Les entités communicantes sont donc partagées en deux sous-familles : celles qui émettent dans la bande B_1 et qui reçoivent dans la bande B_2 (type 1) et inversement pour l'autre sous-famille (type 2) (figure III-12). On pose $f_{B_2} > f_{B_1}$, et $f_{B_2\min} - f_{B_1\max} > \Delta f_{\text{duplex}}$.



$$\begin{aligned} f_i, f_k &\in B_1 \\ f_j &\in B_2 \end{aligned}$$

Figure III-13 : Diffusion en duplex à une antenne

La diffusion n'est possible que d'un acteur d'une sous-famille vers les acteurs de l'autre sous-famille à condition que ceux-ci aient la fréquence de réception. Pour que la diffusion soit totale, il faut que le message soit relayé par un acteur de l'autre sous-famille ayant reçu la diffusion. Ceci pose un problème de tolérance aux pannes, car dans le cas extrême où $n = 3$ acteurs (figure III-13), un acteur A_j ne peut pas communiquer avec A_k , du même type que lui, lorsque l'acteur A_j , relais de l'autre type, est en panne. Etudions maintenant les topologies les plus classiques, applicables à ActNet.

Les topologies qui peuvent être mises en oeuvre sont le bus et l'étoile. L'arborescence binaire n'est pas possible, car nous avons limité le nombre de modems à l'unité; la communication doit se faire en duplex avec les acteurs père et fils.

La solution bus consiste à constituer deux bus à collision, un par bande de fréquences. Une seule fréquence peut être utilisée par bande ou comme dans le cas de la figure III-14, une fréquence par acteur. Dans le premier cas, la diffusion n'est possible que pour les acteurs de l'autre type. Dans tous les cas, l'acteur ne peut écouter le canal avant émission, puisqu'il émet sur le canal dual. Le protocole est donc de type aloha. Un cas extrême est à considérer, celui où il n'y a que trois communicateurs, avec deux communicateurs de type 1. Si le communicateur minoritaire relais tombe en panne, la communication devient impossible.

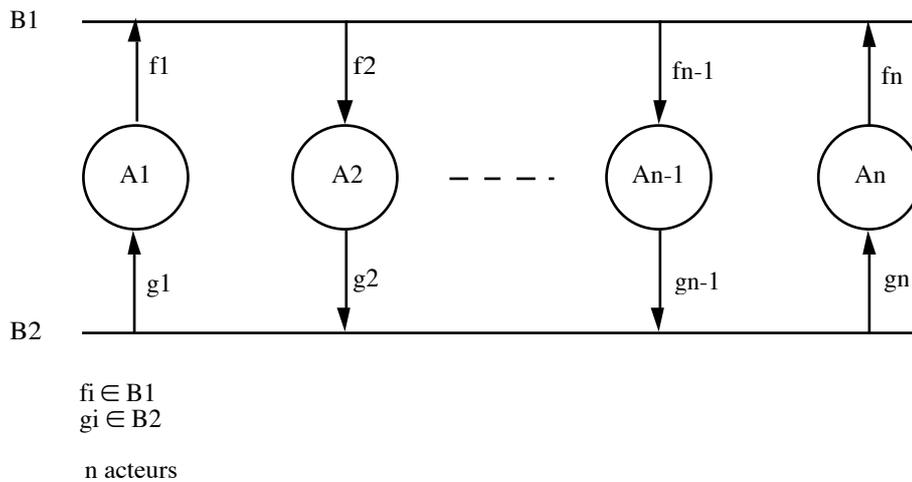


Figure III-14 : Bus en duplex à une antenne

La topologie en étoile (figure III-15) semble la plus viable, solution d'ailleurs retenue pour le téléphone cellulaire. La station de base, appelée aussi concentrateur ou station relais, est de type 1. Les stations satellites sont de type 2. Si l'acteur relais est sédentaire, le nombre d'éléments communicateurs peut être supérieur à l'unité. S'il n'y a qu'un communicateur, les acteurs sont interrogés avec une politique d'interrogation circulaire et une seule fréquence par bande est suffisante. Une diffusion est alors possible. Le problème de la tolérance aux pannes est critique puisque la panne du concentrateur empêche toute liaison.

L'anneau avec une politique à un ou plusieurs jetons (figure III-16) est aussi possible. Par contre, si n est impair, un des acteurs doit être enlevé de l'anneau et la communication avec cet acteur doit s'effectuer par l'intermédiaire d'un acteur passerelle du type opposé. Le principal problème est encore la tolérance aux pannes : lorsqu'un acteur tombe en panne, l'anneau ne peut être reconstitué simplement du fait que deux communicateurs de même type ne peuvent communiquer.

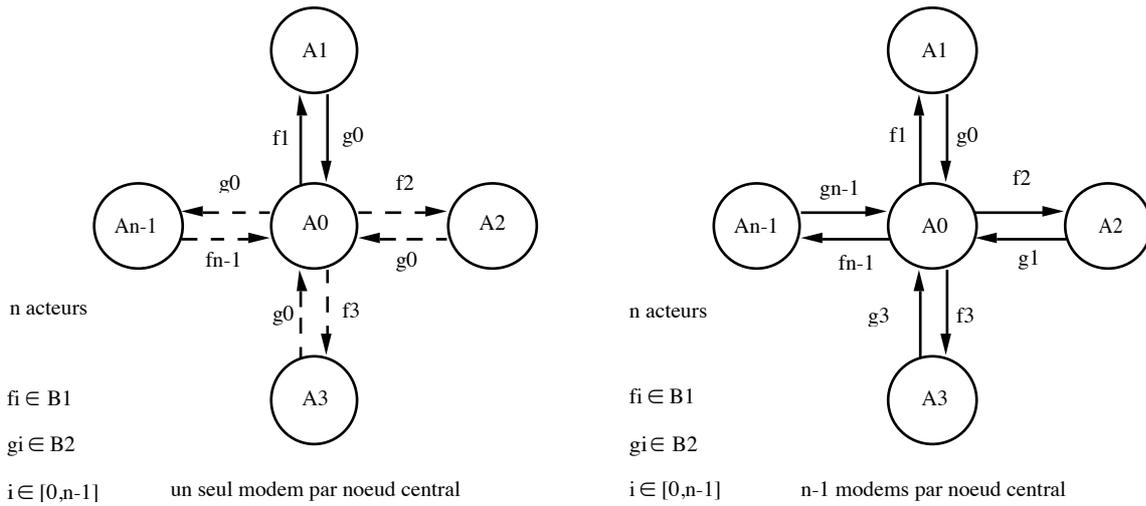


Figure III-15 : Etoile en duplex à une antenne

Deux cas se présentent alors :

- soit le nombre d'acteurs en état de fonctionnement est impair, il faut donc retirer de l'anneau un acteur du type qui est majoritaire en le faisant communiquer par un acteur passerelle,
- dans le deuxième cas, un acteur communiquait déjà par un acteur passerelle. Alors soit les nombres d'acteurs de chaque type sont égaux et il n'y a pas de problème, soit il y aura un acteur supplémentaire qui communiquera par un acteur passerelle.

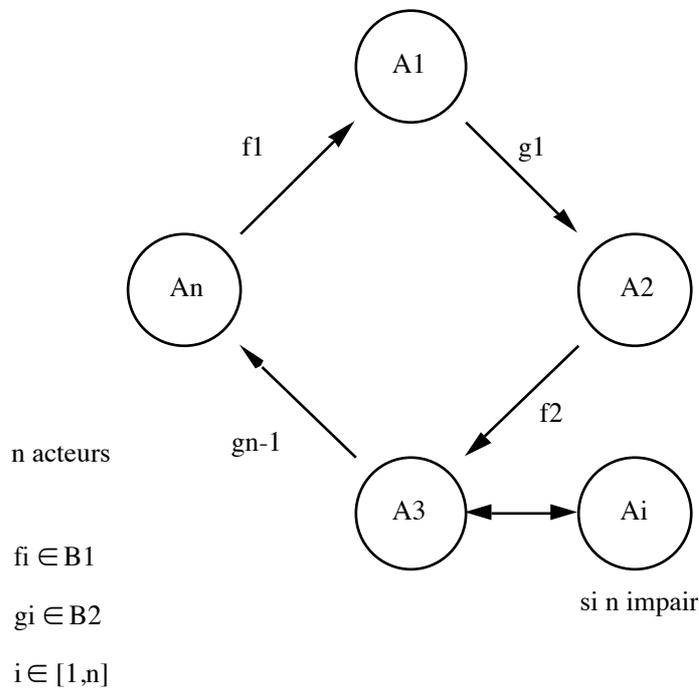


Figure III-16 : Anneau en duplex à une antenne

La topologie de type arborescence (figure III-17) est applicable mais il faut un seul modem par acteur mobile avec une communication duplex. Chaque étage acteur est d'un type opposé au précédent. De plus, un père ne peut être joint par ses fils simultanément, il y a donc risque de collision.

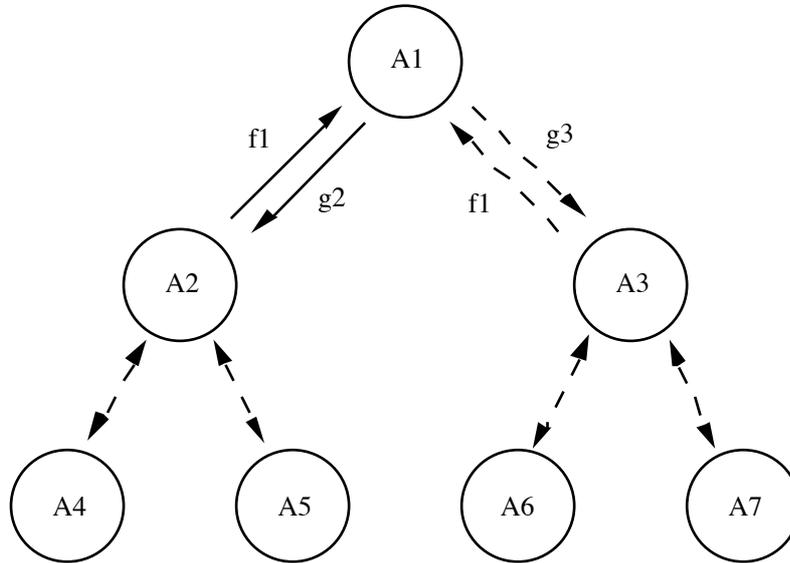
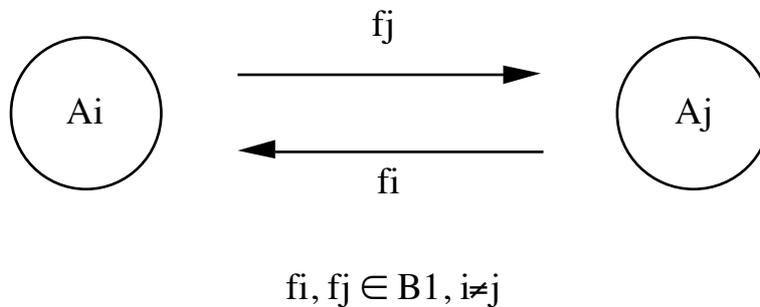


Figure III-17 : Arborescence binaire en duplex à une antenne

4.4.2. Communication duplex à deux antennes



$$B1 = [40,45] \text{ MHz ou } [70,75] \text{ MHz}$$

Figure III-18 : Anneau en duplex à deux antennes

L'utilisateur de deux antennes permet d'émettre dans la même bande sans avoir la contrainte de l'écart duplex (figure III-18). La caractéristique primordiale est le découplage entre antennes de réception et d'émission. Il évite la réception gênante de signaux perturbateurs émis par l'émetteur. Ces signaux sont considérés comme du bruit. On considère que le découplage est horizontal, car les antennes sont montées sur un même plan pour tout type d'acteurs matériels. Ce cas est le moins favorable car le couplage est maximal

par rapport à des antennes montées verticalement l'une par rapport à l'autre [Rémy et al. 88]. Une seule bande est utilisée. Etudions maintenant les topologies les plus classiques applicables à ActNet.

La topologie bus n'est réalisable que dans une configuration bi-bus (figure III-19). Un seul bus n'est pas envisageable car nous voulons une communication duplex, c'est-à-dire deux voies.

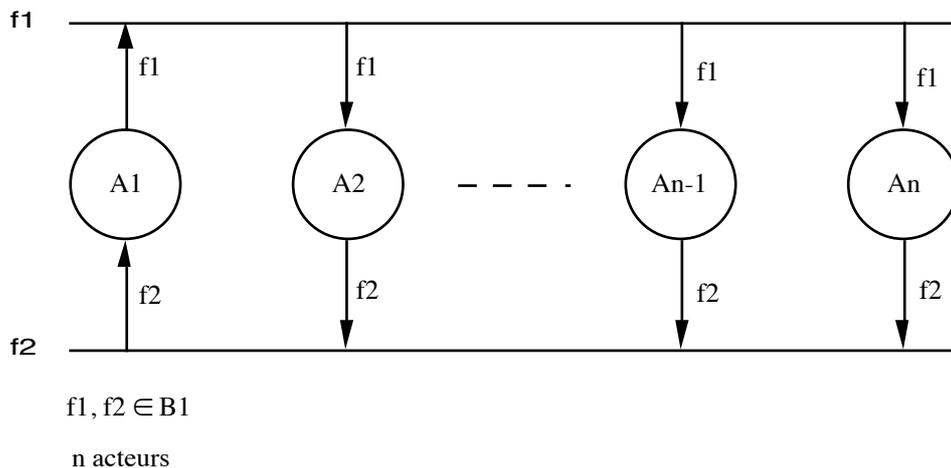


Figure III-19 : Bus en duplex à deux antennes

La topologie en étoile (figure III-20) est possible avec les configurations à un ou plusieurs modems pour le noeud central.

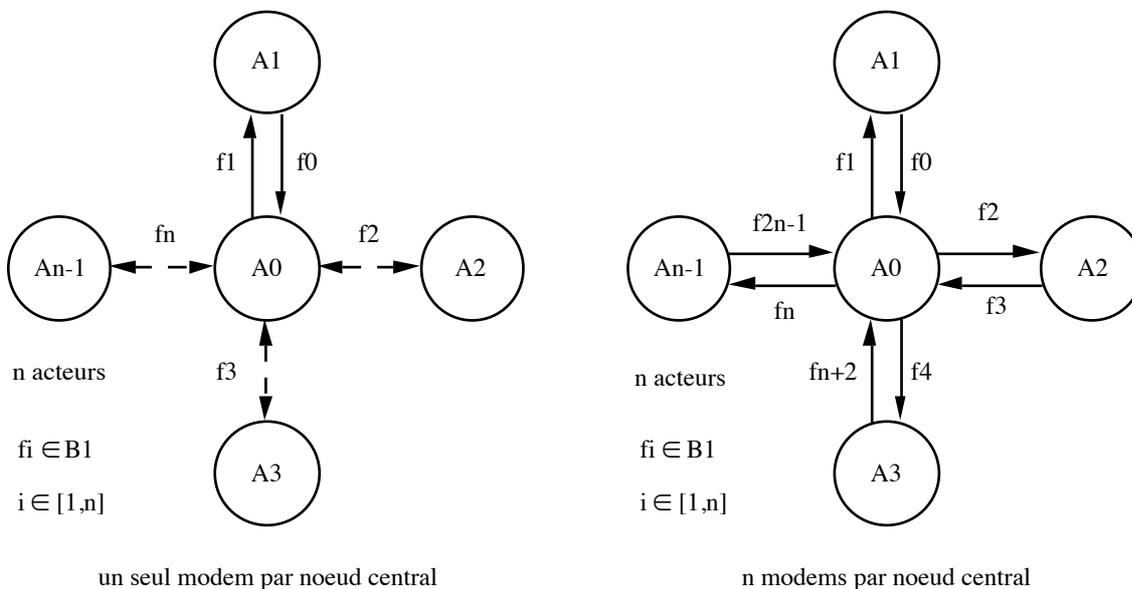


Figure III-20 : Etoile en duplex à deux antennes

La topologie en anneau (figure III-21) est réalisable. L'intérêt de cette configuration est la possibilité d'avoir plusieurs jetons qui tournent en même temps, augmentant d'autant la bande passante du réseau. Ceci est possible puisque la communication s'effectue par des canaux différents.

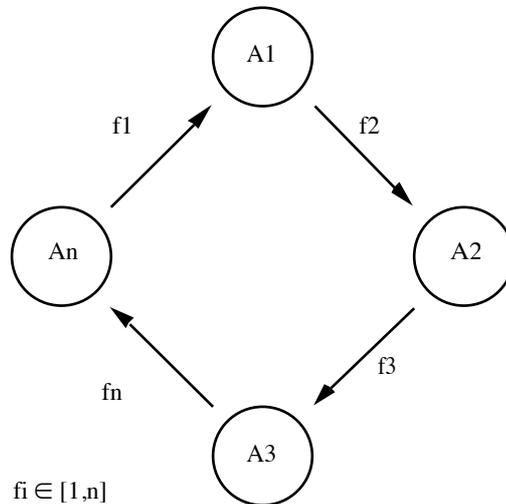


Figure III-21 : Anneau en duplex à deux antennes

La topologie de type arborescence (figure III-22) est réalisable, mais un noeud ne pourra communiquer à un instant donné qu'avec un seul acteur. Lorsqu'un acteur communique avec un acteur, un troisième voulant joindre l'un des deux premiers risque de rentrer en collision.

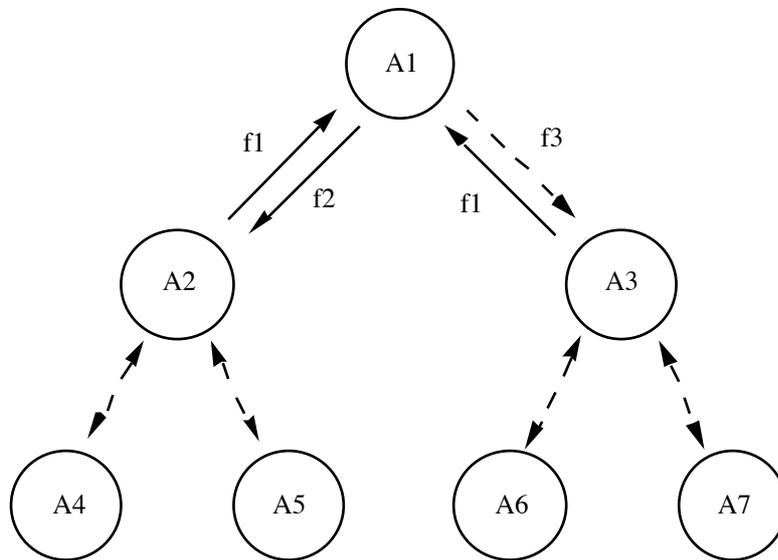
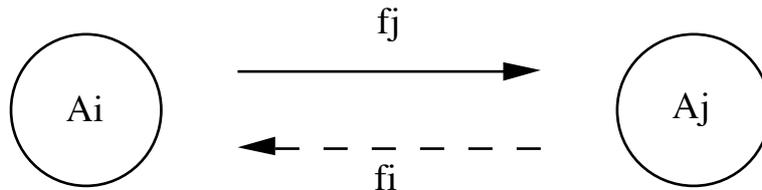


Figure III-22 : Arbre binaire en duplex à deux antennes

Les topologies présentées sont identiques à celles de l'étude précédente, au nombre de bandes près.

4.4.3. Communication simplex ou à l'alternat

Dans le cas où tous les acteurs mobiles ne sont équipés que de communicateurs d'un même type et d'un commutateur émission/réception, l'émission et la réception ne sont pas simultanées. Les éléments communicants travaillent dans une seule bande avec une seule antenne (figure III-23). Les fréquences d'émission et de réception peuvent être égales. La communication est simplex s'il n'y a qu'un seul canal, sinon elle est à l'alternat.



$$f_i, f_j \in B_1, B_1 = [40,45] \text{ MHz ou } [70,75] \text{ MHz}$$

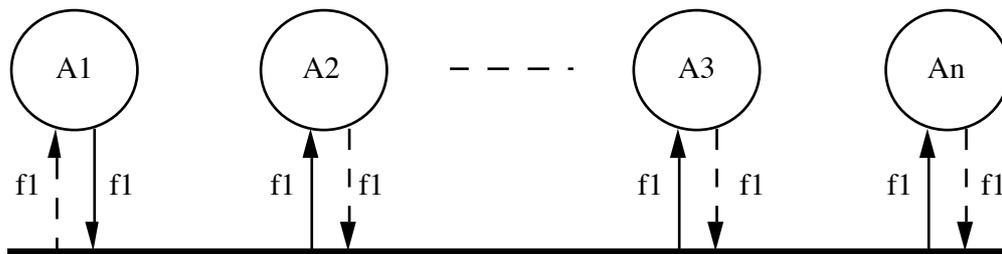
$$i=j \text{ ou } i \neq j$$

Figure III-23 : Anneau en alternat

Le canal est de type canal à collision lorsque deux acteurs veulent communiquer en même temps avec un troisième acteur. L'écoute du canal est possible juste avant l'émission d'un message.

Les topologies possibles sont :

- le bus (figure III-24),
- l'étoile (figure III-25),
- l'anneau à jeton (figure III-26),
- l'arborescence n-aire (figure III-27).



$$f_1 \in B_1$$

Figure III-24 : Bus en simplex

Pour la topologie de type bus (figure III-24), l'acteur pouvant écouter le canal avant transmission, le protocole pour le bus pourrait être de type CSMA/CA. La communication est de type simplex.

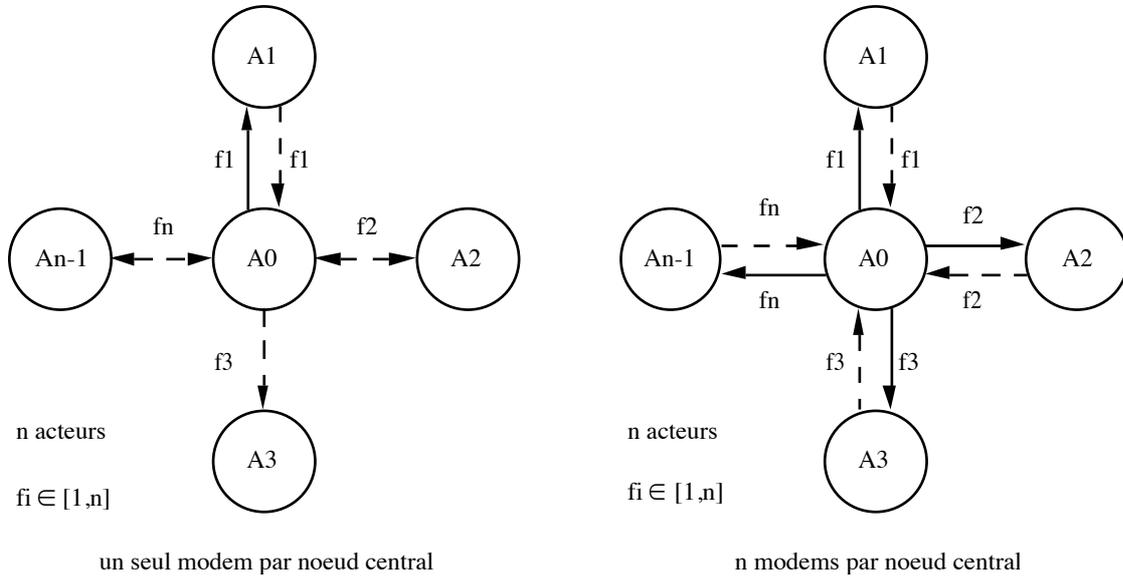


Figure III-25 : Etoile en alternat

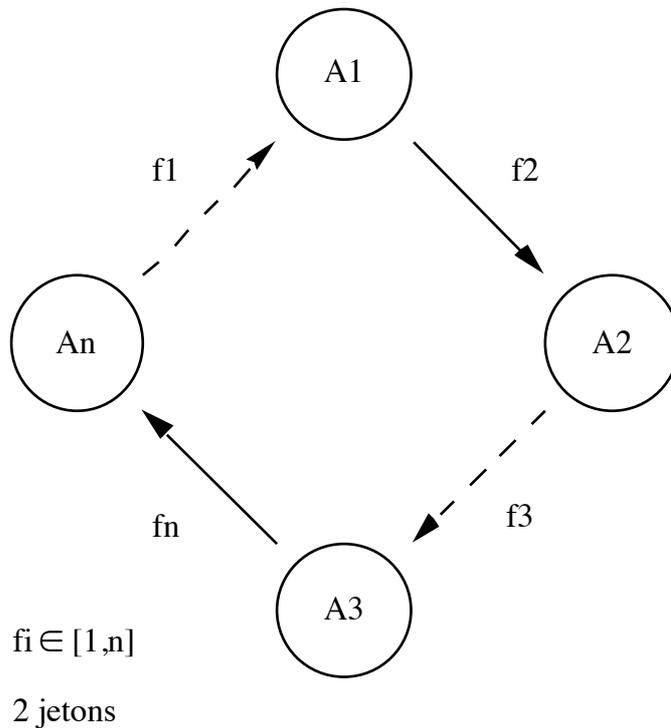


Figure III-26 : Anneau en alternat

L'étoile est possible avec une politique cyclique où le noeud central interroge successivement les acteurs mobiles. Cette topologie est possible avec les configurations à un ou plusieurs modems pour le noeud central. La communication peut être simplex (figure III-25) ou à l'alternat (deux canaux par acteur).

L'anneau (figure III-26) peut tourner avec plusieurs jetons. Mais le risque de perte existe si les deux acteurs sont contigus, le deuxième acteur émettant et ne pouvant donc pas recevoir le jeton du premier.

L'arborescence représentée en figure III-27 est binaire. Le problème est le même que dans le cas de la communication duplex à deux antennes. Un acteur risque de ne pas recevoir la communication d'un troisième acteur alors qu'il communique avec un deuxième.

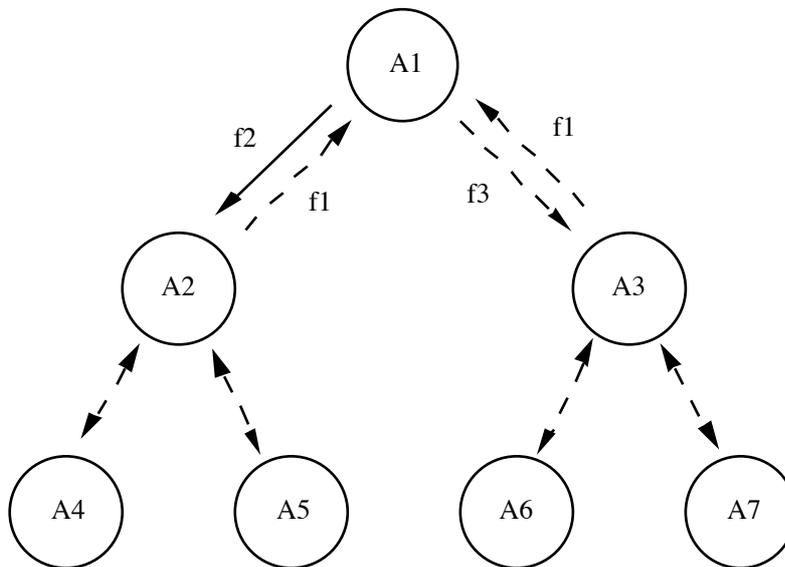


Figure III-27 : Arborescence en alternat

5. Sous-réseau d'acteurs sédentaires

Chaque acteur utilise les liens, c'est-à-dire l'interface série rapide du transputer pour communiquer entre chaque branche du double anneau. Le Transputer possède quatre liens série haut débit à 20 Mbits/s. La transmission est asynchrone, car il n'y a aucune synchronisation entre les horloges d'émission et de réception. Par contre, la communication est synchrone, car il faut que le lien récepteur soit en attente du message et qu'il renvoie dès réception de celui-ci un accusé-réception de 2 bits. Nous avons sélectionné la transmission bande de base. Ces liens peuvent être connectés directement par un câble électrique de type paire torsadée pour des liaisons inférieures à 30 cm. Pour des distances supérieures à 30 cm et jusqu'à 30 m, un amplificateur est utilisé. Cet adaptateur joue aussi le rôle d'adaptateur d'impédance.

La configuration en anneau a été retenue. Il s'agit d'un double anneau contre-rotatif. Cette configuration a l'avantage d'être tolérante aux pannes. En effet, si une liaison est rompue, il est possible de parcourir l'anneau dans l'autre sens pour repérer la liaison défectueuse.

La solution retenue permet un coût de connexion réduit au minimum pour de bonnes performances (vitesse de transmission, longueur maximale des liaisons filaires).

6. Passerelle

Les deux sous-réseaux sont reliés par une passerelle qui fait transiter les messages. cette fonction est réalisée par un acteur matériel spécialisé : la passerelle. L'acteur passerelle est constitué d'une interface SRAM et d'une interface SRAS. Sa deuxième fonction est de réguler le flux de données en faisant office de tampon à messages.

7. Conclusion

Nous avons présenté les modulations radio-fréquences et nous avons retenu la modulation MFSK. Les principes d'émission et de réception radio-fréquences ont ensuite été détaillés.

Pour communiquer, il est nécessaire d'avoir plusieurs voies ou canaux, donc d'avoir plusieurs références fréquentielles. L'utilisation d'oscillateurs à quartz définissant la fréquence centrale du canal s'impose pour des problèmes de stabilité car la largeur des canaux est petite (50 KHz). L'utilisation d'un quartz par canal peut s'avérer coûteux et encombrant dans le cas d'un nombre important de canaux (> 5). Nous avons remédié à ce problème en utilisant une boucle à verrouillage de phase qui, avec un seul référentiel fréquentiel, génère la fréquence centrale de tous les canaux. Une autre solution aurait été envisageable : la synthèse numérique directe, mais la technologie est émergente.

Nous avons ensuite présenté quelques principes de communication dans les réseaux radio-fréquences en présentant des descriptions de réalisations présentes et futures, ainsi que les principales normes nationales réglementant la radio-communication numérique.

Nous avons ensuite décrit le sous-réseau d'acteurs mobiles. Le SRAM est un réseau radio-fréquence VHF pour garantir l'autonomie des mobiles du point de vue des communications [**Darche 91**]. La communication est de type asynchrone, en duplex, à l'alternat ou en simplex grâce à la modularité des éléments communicants. La vitesse de modulation est celle d'une liaison série RS232C équipant la carte de contrôle, et celle-ci varie entre 1200 et 19200 bauds. Un codage de canal est nécessaire pour adapter l'information. Pour obtenir une communication duplex, il est nécessaire de différencier l'information provenant de plusieurs interlocuteurs. Une modulation analogique sera utilisée. l'espace fréquentiel sera découpé en canaux identiques appelés canaux radio-fréquences. Dans le cas d'une communication à deux voies, deux canaux seront alloués à chaque acteur du SRAM, qui sera par conséquent caractérisé par deux adresses fréquentielles, une pour l'émission et l'autre pour la réception. Pour l'occupation spectrale et la fiabilité, une modulation de fréquence (FM) discrète, une modulation angulaire, la modulation BMFSK (Binary

Minimum Frequency Shift Keying) a été choisie, car elle optimise la largeur de bande nécessaire à la transmission d'un signal digital, sans un codage de source élaboré. La vitesse de modulation maximale demandée et le type de modulation employée définissent une largeur de canal de 50 KHz. L'utilisation de la technologie digitale (les boucles à verrouillage de phase, Phase Lock Loop en particulier) nous permet de programmer les fréquences d'émission et de réception de chaque mobile sans multiplier le nombre de références fréquentielles.

Dans le cas de l'utilisation d'un duplexeur d'antenne, la communication est duplex. Mais à cause d'un écart duplex entre les fréquences d'émission et de réception, il existe deux bandes de fréquence distinctes B1 et B2, et donc deux familles d'acteurs, entraînant ainsi des contraintes au niveau de la topologie.

Dans le cas où deux antennes sont utilisées, il n'existe qu'une seule famille d'acteurs et toutes les topologies classiques sont réalisables.

Dans le cas de l'utilisation d'un commutateur d'antenne, les communications de type simplex et mieux, de type à l'alternat, sont réalisables. Les choix techniques effectués nous permettent d'obtenir la meilleure topologie de réseau pour les applications de l'utilisateur (bus, anneau, étoile, arborescence, etc.).

En cas de défaillance des systèmes embarqués de télécommunication ou lors de la phase de développement, les acteurs mobiles pourront communiquer par un canal filaire à collision.

L'utilisation d'une boucle à verrouillage de phase pour générer des références fréquentielles permet aux équipements communicants de s'adapter à une communication de type spectre élargi à saut de fréquence. La prochaine étape de notre travail consistera à mettre en oeuvre cette technique de transmission. Par la suite, il serait intéressant de s'intéresser aux ondes infra-rouges pour développer des réseaux faible coût, donc très bien adaptés aux applications pédagogiques. Les diodes électro-luminescentes infra-rouges se prêtent bien à la modulation de séquence directe à étalement de spectre.

Chapitre IV

Architecture logicielle

- 1. Introduction**
- 2. Etat de l'art des systèmes**
- 3. Architecture logicielle de l'acteur matériel**
- 4. Conclusion**

1. Introduction

L'architecture logicielle de nos robots est complexe car elle intègre à la fois les notions de concurrence, de communication, de temps réel et, dans les extensions futures, les notions de système distribué.

Le modèle acteur nécessite de gérer la concurrence entre entités. La communication doit être gérée entre acteurs d'un même noeud (communication interne) et entre acteurs matériels (communication externe). Les acteurs communiquent entre eux par messages. A cela s'ajoute la nécessité de garantir les temps de réponse de l'acteur de contrôle pour ne pas mettre en péril les procédés mis en oeuvre, voire même entraîner la destruction du robot. Il faut donc gérer en temps réel certains signaux provenant de capteurs et fournir les actions nécessaires en retour.

Les systèmes matériels sont souvent multi-processeurs afin que les temps de réponse du système informatique soient compatibles avec la mission à accomplir. L'informatique étant répartie, ces systèmes sont donc gérés par des systèmes distribués. L'approche système virtuel ActNet insiste encore plus sur la notion de distribution. Dans ActNet, tout est acteur ou objet matériel et/ou logiciel selon le contexte.

Nous présentons un bref état de l'art des systèmes temps réel. Nous comparons ensuite l'approche **système monolithique** et l'approche **micro-noyau**. Nous décrivons ensuite l'architecture logicielle d'un acteur matériel, basée sur le concept de micro-noyau.

2. Etat de l'art des systèmes

Nous présentons dans cette partie un bref état de l'art concernant les systèmes temps réel et les approches logicielles possibles pour le développement d'un système informatique.

2.1. Le temps réel

Un système temps réel (STR) est un système informatique qui gère un procédé. Sa propriété essentielle est la prédictibilité, c'est-à-dire que ses comportements fonctionnel et temporel doivent être déterministes. Il doit par conséquent respecter des échéances de temps.

Il en existe plusieurs catégories [**Mullender 93**] :

- STR à contraintes temporelles relâchées, ou doux (Soft Real-Time (SRT)),
- STR à contraintes temporelles strictes ou dur (Hard Real-Time (HRT)).

Un STR à contraintes temporelles relâchées est un système qui admet une faible probabilité de non respect des contraintes temporelles. Un exemple serait l'acquisition de température avec un procédé évoluant lentement d'un point de vue thermique. La perte ou le retard d'une acquisition n'aurait pas une conséquence catastrophique.

Un STR à contraintes temporelles strictes est un système qui doit respecter les échéances d'exécution de tâches dans n'importe quelles circonstances. Dans le cas contraire, cela pourrait entraîner une destruction du procédé. En exemple, je citerais un système de contrôle d'une centrale nucléaire; outre les défaillances matérielles, le moindre non respect de contraintes temporelles au niveau exécution de tâches peut avoir des conséquences dramatiques.

Les STR à contraintes temporelles relâchées se décomposent en deux catégories :

- ceux à haute intégrité, un système bancaire par exemple,
- et ceux à haute disponibilité, un commutateur téléphonique par exemple.

Les STR à contraintes temporelles strictes se décomposent en deux catégories :

- ceux à sécurité intégrée (Fail Safe), un système de signalisation ferroviaire par exemple,
- et ceux opérationnels en faute (Fail Operational), un système de contrôle aérien par exemple.

Les STR peuvent être dirigés par les événements (Event triggered), ou par le temps (Time Triggered).

Nous distinguons deux types de tâches dans un système temps réel :

- les tâches périodiques,
- les tâches apériodiques.

Une tâche périodique peut par exemple avoir comme fonction la lecture d'un capteur de température ou une mesure de tension de batterie. La détection de collision est typiquement un événement aléatoire, et il est donc traité par une tâche apériodique.

Un autre classement serait :

- les tâches critiques, qui doivent respecter leurs échéances, sinon le procédé risque d'être en péril,
- les tâches essentielles, où on admet de légers retards,
- les tâches non essentielles où il est possible de dépasser les échéances,
- les tâches d'arrière-plan (non temps réel) qui sont exécutées lorsque la ressource processeurs est disponible.

La structuration de la couche logicielle est influencée par ces contraintes de temps réel, notamment au niveau du système d'exploitation qui est l'interface entre les acteurs et le matériel. Le choix de la structure du système d'exploitation est donc un problème délicat.

Parmi les systèmes temps réel généralement décrits, certains sont basés sur un noyau monolithique, d'autres ont choisi une structure à base de serveurs coopératifs au dessus d'un micro-noyau. Les parties suivantes présentent ces deux architectures en comparant leurs avantages et leurs inconvénients.

2.2 L'approche noyau monolithique

De manière traditionnelle, les systèmes d'exploitation ont été conçus sous la forme d'un seul programme appelé noyau s'exécutant en mode superviseur. Celui-ci est chargé de gérer les ressources physiques et de satisfaire les demandes des processus utilisateur. Le système Unix [**Bach 86**] [**Leffler et al. 89**] constitue un bon exemple de cette approche.

Les services assurés par le noyau comprennent entre autres les systèmes de fichiers, l'ordonnancement des processus, la gestion de la mémoire. Les programmes système et utilisateur utilisent les appels système fournis par le noyau pour offrir des fonctionnalités de haut niveau à l'utilisateur.

Le modèle monolithique offre certains avantages:

- les performances du système sont excellentes dans la mesure où les appels système sont exécutés sous forme de trappes;
- le regroupement de tous les services système dans le même espace d'adressage leur permet de communiquer aisément entre eux en partageant des données et en s'appelant directement par des appels de sous-programmes.

L'inconvénient majeur d'un système monolithique est la difficulté de gérer dynamiquement les pannes de matériel (disque, communication, etc.), ce qui est nécessaire dans le cadre robotique. De plus, un tel système est difficile à maintenir et à étendre. Comme les divers services système s'appuient les uns sur les autres et sont interdépendants, l'ajout ou la modification de service entraîne fréquemment des effets de bord.

2.3. L'approche micro-noyau multi-serveurs

Une architecture plus récente propose de séparer les fonctionnalités offertes par le noyau en deux couches, l'une implémentant les fonctions de bas-niveau et d'accès à la machine physique, l'autre regroupant les services de haut niveau offerts à l'utilisateur par le système d'exploitation.

La première couche, appelée micro-noyau continue à s'exécuter en mode superviseur alors que la seconde est répartie dans des serveurs coopérants s'exécutant en mode utilisateur. Les demandes de service des processus utilisateurs sont alors envoyées sous forme de requêtes vers les serveurs spécialisés. Ceux-ci réalisent la demande en faisant éventuellement appel soit à d'autres serveurs, soit directement au micro-noyau.

Les principaux micro-noyaux Amoeba [**Mullender et al. 90**], Chorus [**Rozier et al. 88**], V-Kernel [**Cheriton 84a**] et Mach [**Tevanian and Rashid 87**] partagent certaines abstractions tels que les communications par messages et les processus légers. Chorus ainsi que Mach (Mach-RT) offrent de plus des extensions permettant la gestion du temps réel dans les applications.

L'architecture multi-serveurs permet une extension aisée du système, chaque serveur étant mis au point indépendamment des autres puis intégré dans le système lorsqu'il fonctionne. Il est également aisé d'ajouter ou de retirer dynamiquement un serveur du système. De plus, comme les serveurs s'exécutent en mode utilisateur, ils ne peuvent pas accéder à la machine physique directement. Ainsi, si un serveur contient une erreur, le service qu'il assure sera indisponible mais le reste du système ne sera pas affecté. Le système réparti MASIX [Card 93] est un exemple de système multi-serveurs, basé sur le micro-noyau Mach.

Cette approche permet aussi de masquer la distribution aux utilisateurs, un message de requête pouvant être traité sur la machine locale ou transmis à un serveur sur une machine distante de manière transparente. L'inconvénient majeur de ce type de système réside dans les performances. Les appels système étant implémentés sous forme de requêtes à un serveur, ils entraînent un ou plusieurs changements de contexte. Cette méthode est beaucoup plus coûteuse que le schéma monolithique. Des études [Dean et Armand 92] [Douglass et al. 91] ont néanmoins montré qu'il était possible d'obtenir des performances voisines de celles des systèmes monolithiques.

3. Architecture logicielle de l'acteur matériel

L'organisation logicielle en couches a été choisie pour s'adapter à différentes architectures matérielles et logicielles comme celles de Brooks en robotique ou celles liées à des standards ISO comme en téléinformatique. Une telle organisation permet une souplesse de développement et des modifications aisées. Toute la partie matérielle peut être modélisée, et simulée aisément par logiciel grâce à l'approche objet, avant de passer à la réalisation matérielle.

Rappelons que le système ActNet se caractérise par l'absence d'horloge globale et la non fiabilité du sous-réseau d'acteurs mobiles.

L'architecture logicielle d'un acteur matériel se compose de quatre couches principales (figure IV-1) :

- la couche entrées/sorties avec les sous-couches matérielle et réflexe,
- la couche système composée de deux sous-couches :
 - le micro-noyau gérant les interruptions et les fonctionnalités de bas niveau (accès périphériques, gestion mémoire physique, etc.),
 - et un ensemble de serveurs systèmes pour les fonctionnalités de haut niveau (gestion de fichiers, mémoire virtuelle, etc.),
- la couche application, avec éventuellement deux sous-couches :
 - des applications interpréteur langage, Méta-Vlisp et Logo, si l'application le nécessite (application interprétée), avec une sous-couche objet/acteur.
 - des applications juste au-dessus du système.

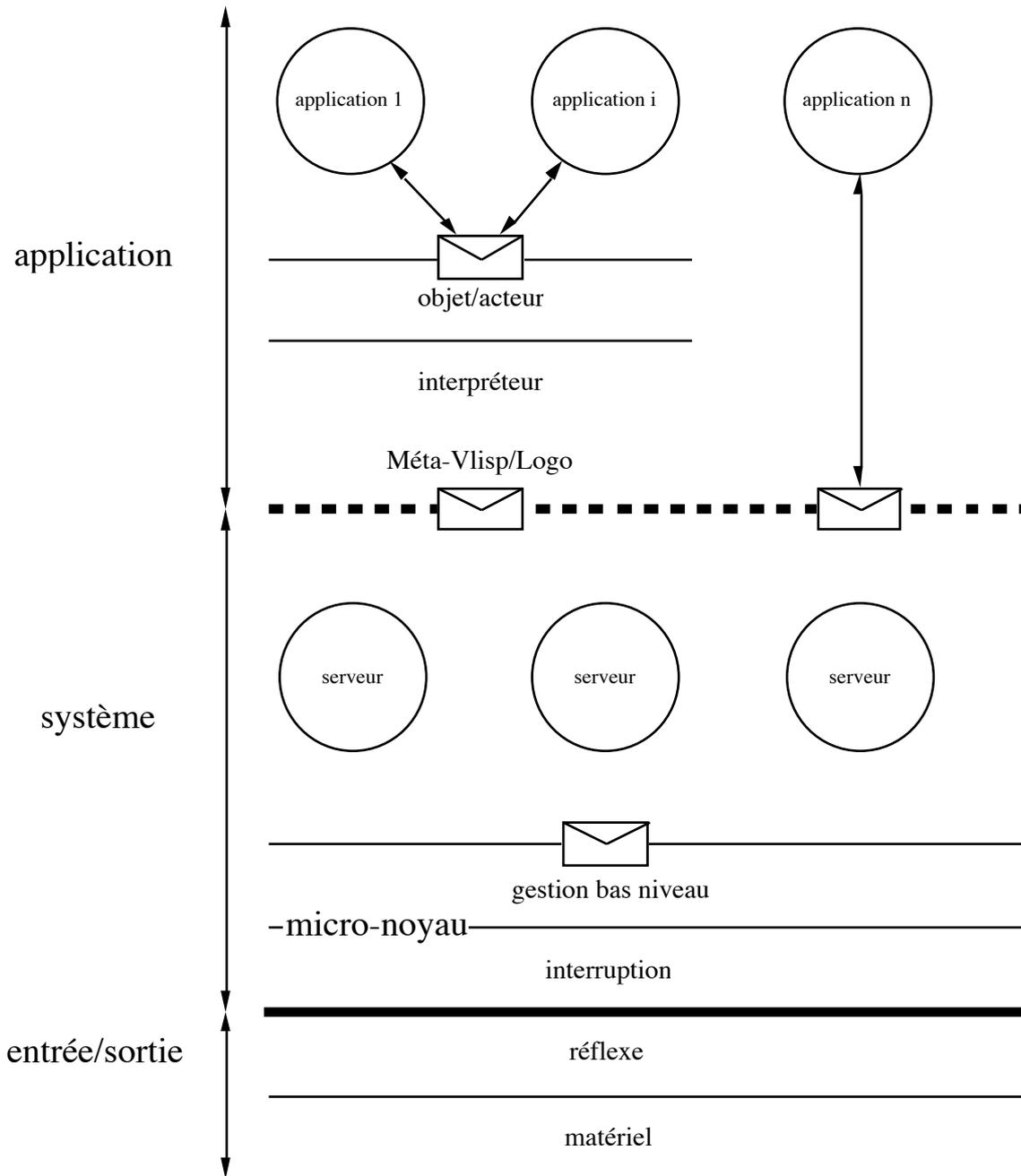


Figure IV-1 : Architecture logicielle d'ActNet

D'un point de vue acteur, les méthodes peuvent être assimilées à des tâches temps réel. Nous aurons donc :

- les **méthodes critiques**,
- les **méthodes essentielles**,
- les **méthodes d'arrière-plan**.

La première couche peut être matérielle ou logicielle. Lors du développement, pour des raisons de temps de réponse, s'il est nécessaire d'accélérer les traitements, le logiciel peut être remplacé par un module matériel. Ceci est vrai en particulier pour la couche réflexe : si un actionneur doit être arrêté rapidement suite à la lecture d'un capteur, un capteur de fin de course par exemple, seul le matériel peut fournir le temps de réponse le plus petit.

Nous présentons les différentes couches intervenant dans l'application.

3.1. Couche entrée/sortie

La couche entrée/sortie comporte des objets de type entrée et de type sortie ou les deux à la fois. Les objets de type entrée sont principalement des capteurs qui permettent d'appréhender le monde extérieur, mais aussi de prendre connaissance d'un état interne (état de charge de la batterie par exemple). Les objets de type sortie sont principalement des actionneurs qui permettent d'agir sur l'acteur ou le monde extérieur. Pour terminer, il existe aussi des périphériques plus complexes de type entrée/sortie qui regroupent les deux types d'objets précédents. Si ces objets sont actifs, nous les appellerons des acteurs.

3.1.1. Sous-couche matériel

La couche matériel avec les différents modules a été décrite dans le chapitre II. Le lecteur désirant des détails s'y reportera.

3.1.2. Sous-couche réflexe

La couche réflexe permet de créer un arc réflexe entre les organes d'entrée et les organes de sorties. Le temps de réponse de cette couche est primordial pour la sauvegarde de l'acteur : lors de collisions effectives ou futures par exemple, l'acteur robotique doit être capable d'arrêter son déplacement très rapidement ou de modifier sa trajectoire. Si la couche est logicielle, des mécanismes de type interruption seront mis en oeuvre. Dans le cas d'une réalisation matérielle, le capteur agit directement sur l'effecteur.

3.2. Couche système

La couche système se compose de deux sous-couches : la sous-couche micro-noyau et la sous-couche serveur.

3.2.1. Sous-couche interruption

La couche interruption gère l'ensemble des objets matériels. Elle gère donc les périphériques robotiques et les périphériques systèmes. Les périphériques robotiques sont ceux des groupes motorisation, impression, capteurs, etc. Les périphériques systèmes sont composés d'éléments classiques (disque dur, lecteur de disquettes, clavier, écran, ports d'entrée-sortie (série, parallèle, réseau), etc.) et d'éléments spécifiques aux systèmes embarqués (énergie, radio-communication).

3.2.2. Sous-couche micro-noyau

Le micro-noyau implémente les fonctions de bas niveau comme :

- la gestion de l'énergie embarquée et externe,
- la gestion du processeur,
- la gestion de la mémoire physique,
- la gestion des entrées-sorties,
- la gestion de bas niveau des acteurs (création, suspension, terminaison, communication interne),
- la gestion des tâches et threads (synchronisation, communication),
- la couche liaison du système de communications entre acteurs matériels.

Il doit résider entièrement en mémoire afin de ne pas mettre en retard l'exécution de certains services non résidents.

Parmi les fonctions citées ci-dessus, les services à temps réel strict sont :

- les services sécuritaires,
- la gestion des communications internes,
- la gestion des ressources critiques : énergie, processeurs, périphériques robotiques,
- l'ordonnancement des tâches,

Ceux-ci doivent être gérés obligatoirement par le micro-noyau sous interruptions pour avoir des temps de réponse compatibles avec les services à réaliser.

3.2.3. Sous-couche serveur

Les serveurs offrent les services suivants :

- les communications entre acteurs matériels et logiciels non hébergés par le noeud,
- la gestion de fichiers,
- la gestion de la distribution des acteurs.

Ce qui caractérise une communication sans lien physique (wireless communication), c'est la non fiabilité de la communication par des ruptures de liaisons (évanouissement par exemple, absorption et autres phénomènes physiques) et la faible vitesse de modulation. Des mécanismes logiciels et des modules matériels doivent être mis en oeuvre afin de pallier à ces manques.

Les systèmes embarqués ont en général une mémoire non volatile. Pour les robots qui ont dépassé une taille que nous qualifierons de critique, le problème de l'encombrement de l'énergie embarquée ne se pose plus. En effet, le volume de la batterie est nettement inférieur au volume utile du robot. La mémoire non volatile peut être un disque dur. Pour des robots de taille inférieure, la mémoire non volatile est une mémoire C-MOS sauvegardée par batterie ou par pile, ou une mémoire de type Flash. La place mémoire est bien sûr restreinte. Grâce à cette mémoire, une gestion de fichiers exécutables et surtout de données est envisageable.

Les acteurs logiciels peuvent migrer entre acteurs matériels. Un gestionnaire de distribution d'acteurs est alors nécessaire.

Les services à temps réel relâché de la sous-couche serveur sont :

- la communication réseau,
- la gestion des ressources systèmes : énergie, processeurs, périphériques robotiques, et sont gérés par des serveurs systèmes prioritaires.

Les services sans contraintes de temps sont :

- la gestion de fichiers,
- la distribution des acteurs logiciels sur les acteurs matériels,
- la désignation des acteurs.

Comme dans les systèmes informatiques où une tâche se désigne à deux niveaux : un nom symbolique et un nom interne, un acteur se caractérise par un nom symbolique et une adresse réseau qui correspond à sa boîte aux lettres. La correspondance nom logique - nom interne est réalisée par un serveur de désignation.

La notion de protection est un point important dans les systèmes distribués. La protection est l'ensemble des mécanismes mis en oeuvre afin de protéger les utilisateurs d'un accès non autorisé. Il faut garantir la protection des ressources, la sécurité de communication des utilisateurs [Mullender 93]. L'authentification du correspondant peut être faite par cryptage/décryptage. L'utilisation d'un réseau radio-fréquence rend le système encore plus vulnérable que lorsque ce dernier utilise un réseau câblé comme support de communication.

En ce qui concerne un réseau non filaire, l'accès au média est très aisé. Lorsque le réseau subit une attaque, nous pouvons classer les actes malveillants en quatre catégories : l'écoute, le blocage, la mascarade, et la modification.

ActNet peut être considéré comme un système distribué. L'hétérogénéité se retrouve à plusieurs niveaux dans ce système :

- au niveau matériel,
- au niveau téléinformatique, à cause de l'utilisation de plusieurs types de réseau d'où divers médiums et protocoles de communication,
- au niveau système d'exploitation.

3.3. Communication

3.3.1. Réseau

Un système de radiocommunication comprend en général deux sous-ensembles [Rémy et al. 88] : un **sous-système gestion et acheminement des messages** (SSGA) et un **sous-système radio** (SSR) qui assure la transmission radio-électrique des messages.

Le SSGA possède les fonctionnalités suivantes :

- interconnexion avec un réseau s'il y a lieu,
- reconnaissance et analyse des demandes de mise en relation,
- recherche des correspondants,
- gestion des ressources en canaux,
- acheminement des messages de l'utilisateur jusqu'au canal de communication radio-électrique; le SSGA ne modifie en rien le message, les traitements comme le cryptage, la compression et les traitements inverses sont effectués en amont,
- supervision des communications (enregistrements de paramètres pour la taxation par exemple).

Le SSR possède les fonctionnalités suivantes :

- transmission des messages par voie radio-électrique,
- supervision de la continuité de la liaison,
- protection contre les tiers.

Les fonctionnalités décrites ci-dessus doivent être intégrées dans les couches téléinformatiques de notre système.

3.3.2. ATM

Nous avons sélectionné le mode de transfert asynchrone (ATM) comme mode de communication entre acteurs matériels. Ce mode est dédié au temps réel et aux hauts débits, mais il peut s'adapter aux débits moins élevés. Le canal radio-fréquence est très perturbé et une longueur de paquets trop grande augmente la probabilité de retransmission.

Des protocoles comme HDLC auraient été trop contraignants, nécessitant des synchronisations entre communicants. De plus, certaines sources ne peuvent pas supporter la synchronisation pour des raisons de temps de réponse du système (matérielles ou logicielles).

L'architecture téléinformatique se décompose en trois couches (figure IV-2) :

- la couche physique (PHY),
- la couche ATM (Asynchronous Transfer Mode),
- la couche d'adaptation (AAL : ATM Adaptation Layer).

La couche physique rassemble l'ensemble des fonctions liées à la transmission. Elle est partagée en deux parties : la couche média physique et la couche de convergence.

La couche d'adaptation ATM sert d'interface avec la couche applicative.

Les messages à transmettre sont de taille peu élevée (une centaine d'octets).

La taille d'une cellule ATM est de :

- 48 octets de données,
- 4 octets d'en-tête
- et un octet de protection.

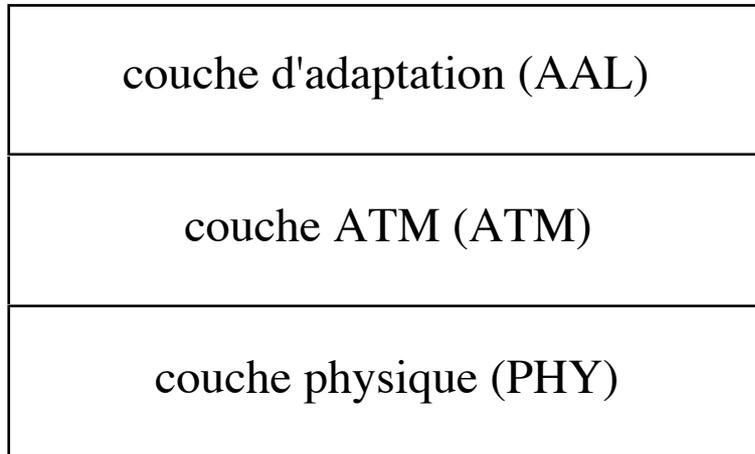


Figure IV-2 : Les couches communication

Mais nous préférons dimensionner la taille d'une cellule dynamiquement en insérant dans l'en-tête une information renseignant sur la taille. En effet, en fonction du type de messages, il nous faut optimiser le temps de transfert. Le problème est critique pour le télégramme. L'information à transporter demande moins de dix octets. Une taille de cellule trop grande augmenterait inutilement le temps de transfert et nuirait à l'efficacité de la communication. De plus, un ajustement dynamique permet d'adapter la transmission par rapport au taux d'erreurs. La contre-partie est une difficulté supplémentaire lors de la modélisation du comportement du réseau avec la théorie des files d'attente due à la taille variable des paquets.

Un acteur matériel s'adresse à ses congénères via leurs adresses réseau. Un acteur logiciel s'adresse via son adresse logique.

La couche ATM permet de rajouter l'en-tête, c'est-à-dire l'adresse fréquentielle de l'acteur et l'adresse logique. Nous avons sélectionné l'acheminement par datagramme, mode d'acheminement adapté à notre application par rapport à l'auto-acheminement et la voie logique.

L'utilisation du mode paquet à taille variable et du transfert asynchrone offre une grande souplesse à l'utilisateur. Ce mode de transfert est original dans le sens où il n'est pas réservé à une classe particulière d'applications. Cela lui permettra entre autres de s'adapter à de futures applications temp réel comme le transport d'images vidéo.

3.3.3. Graphe de liaison

Comme les systèmes de communication sont embarqués, la puissance disponible est limitée, de même que la distance maximale de liaison. De plus, des phénomènes d'ombre font que la liaison peut être interrompue.

Une étude probabiliste a été réalisée pour ce type de communication [Araï et al. 93]. Lorsqu'un lot de robots coopèrent, il est nécessaire d'évaluer le délai de transmission, et de contrôler la diffusion. La zone d'ombre est due à des obstacles, à un manque de puissance et à des phénomènes d'annulations d'onde.

Pour constituer le futur graphe de liaison, deux types d'information sont accessibles :

- les robots ont un chemin déterminé,
- le déplacement est aléatoire.

Un graphe géographique d'obstacles permet par expériences successives de déterminer l'environnement. Si celui-ci est statique, le graphe est complet, sinon il peut l'être sans certitude.

Ceci est d'autant plus vrai pour une communication de type infra-rouge, car les robots peuvent être séparés par des obstacles non franchissables statiques ou dynamiques. Ce graphe est dynamique car les robots sont mobiles.

La fonction relais est un point important. Ce sont des points sûrs de communication, mis à part les pannes matérielles.

Le réseau peut être modélisé par un graphe où les noeuds sont des acteurs et les arcs sont des liens de communication. Trois types de graphes sont nécessaires :

- le graphe des capacités de liaisons physiques de type statique (figure IV-3),
- le graphe des liaisons physiques de type dynamique (figure IV-4),
- le graphe des capacités de liaisons logiques de type statique (figure IV-5),
- le graphe des liaisons logiques de type dynamique (figure IV-6).

Le graphe des capacités de liaisons physiques est constitué à partir des informations suivantes :

- le type de la communication (duplex, simplex, alternat) et donc du type de communicateurs,
- les distances maximales de liaison,
- le nombre maximum d'acteurs.

A partir de ce graphe, on déduit un graphe de liaison physique. Il est constitué à partir des informations suivantes :

- le graphe des capacités de liaisons physiques,
- la zone géographique de déplacement des acteurs matériels mobiles.

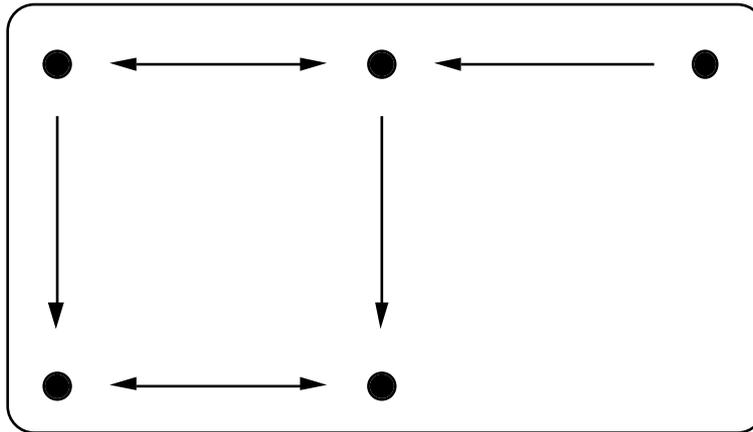


Figure IV-3 : Graphe des capacités de liaisons physiques

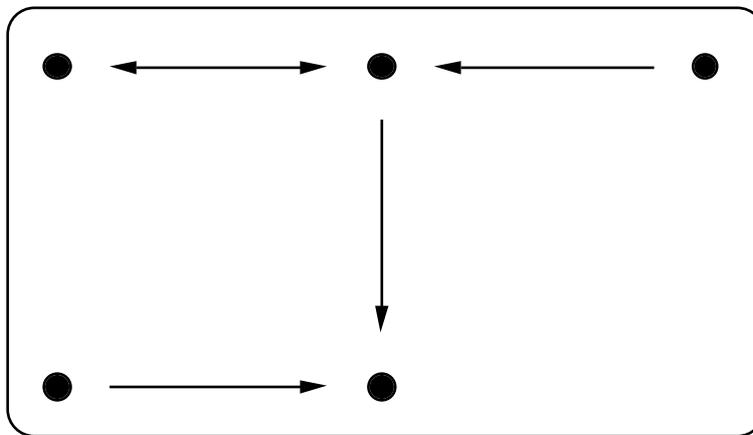


Figure IV-4 : Graphe des liaisons physiques

Le graphe des capacités de liaisons logiques est constitué à partir des informations suivantes :

- le graphe de liaisons physiques,
- la topologie du réseau choisie,

La **zone de couverture** peut ainsi être définie. La zone de couverture est l'ensemble des points où une liaison peut être établie avec un acteur. Cette zone peut être fixe si l'acteur considéré est de type sédentaire. Elle est mobile si l'acteur considéré est mobile. Lorsque l'alimentation électrique n'est pas régulée, les caractéristiques des communicateurs changent et le diamètre de la zone de couverture varie.

Le graphe de liaison logique est dynamique (et modifié en temps réel) s'il y a des mobiles car ceux-ci font déplacer leur zone de couverture. Il peut arriver aussi des pannes ou des perturbations du canal impliquant des ruptures momentanées ou permanentes.

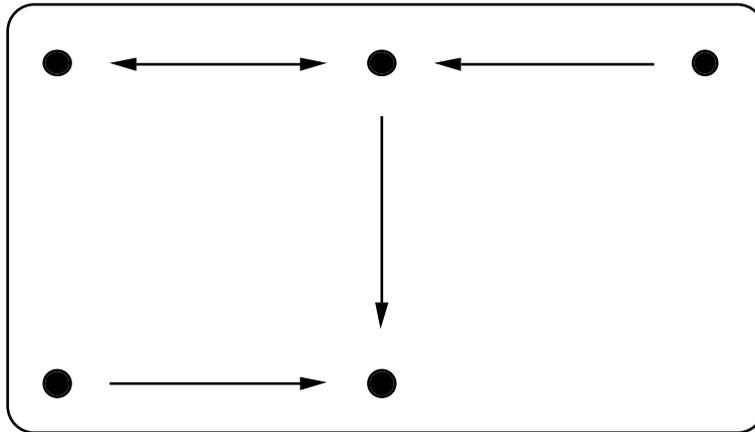


Figure IV-5 : Graphe des capacités de liaisons logiques

Une panne matérielle peut modifier les trois graphes, alors qu'une panne logicielle n'agit que sur les deux derniers.

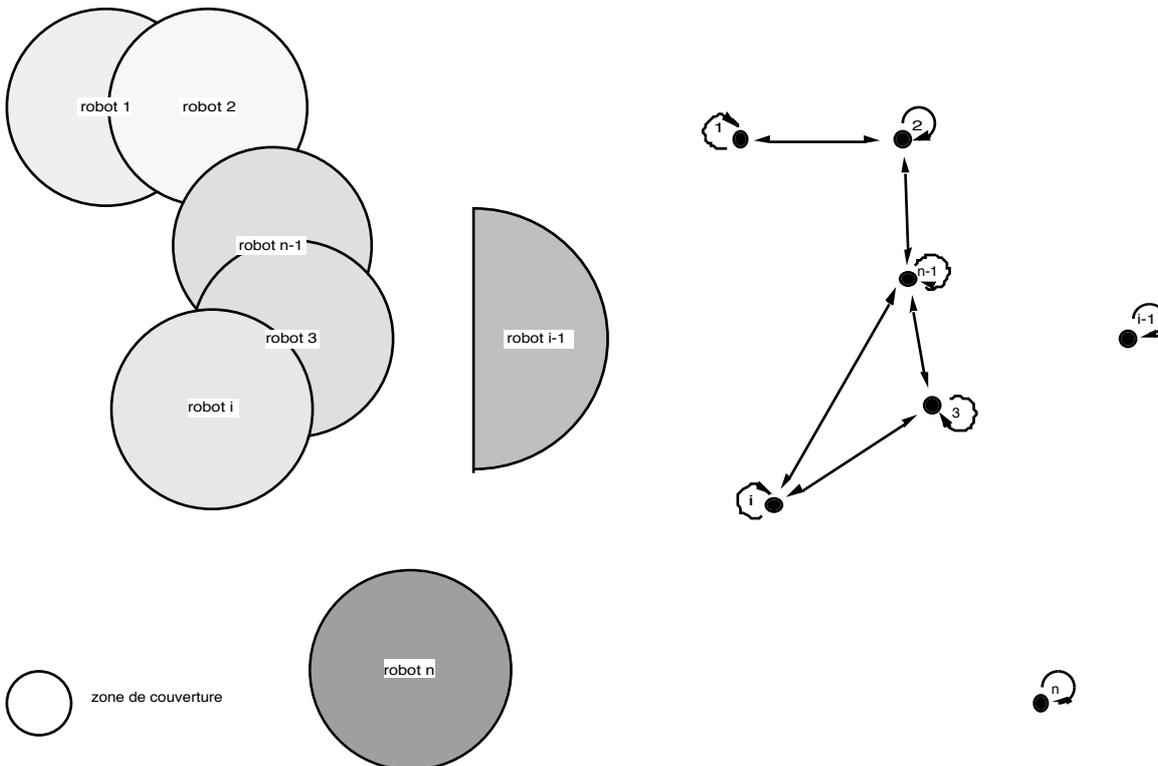


Figure IV-6 : Graphe des liaisons logiques

3.4. Couche application : Méta-Vlisp

Au niveau applicatif, nous proposons deux types d'applications (figure IV-7) : celles qui fonctionnent au-dessus du système d'exploitation et celles qui ont besoin d'un interpréteur.

Citons pour exemple d'applications : interpréteur de commandes (automate), applications comportementales, etc.

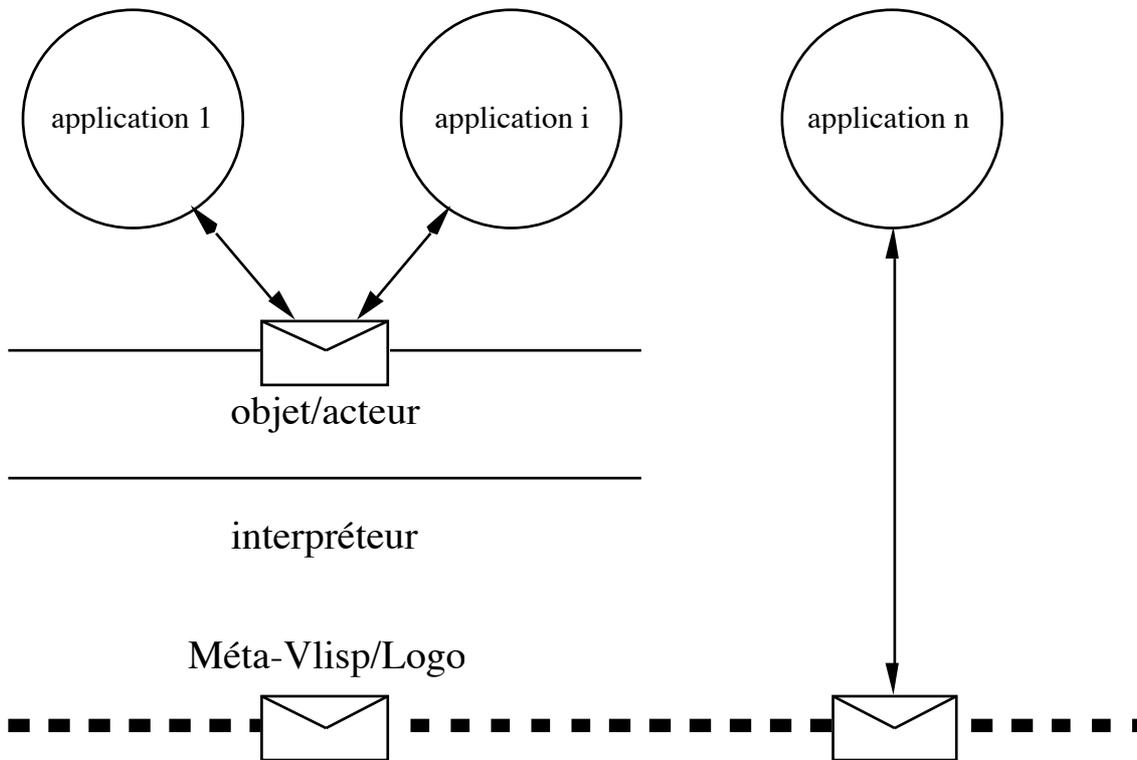


Figure IV-7 : Couche application

Nous avons décidé de nous intéresser au macro-parallélisme ou parallélisme d'applications par opposition au parallélisme "intrinsèque" qui essaye de paralléliser en interne l'interpréteur Lisp. Le modèle acteur se prête bien à ce choix. Grâce à la notion d'acteurs robotiques communiquant par messages et qui peuvent embarquer un interpréteur Méta-Vlisp, chaque acteur exécute la boucle principale :

`(print (eval (read))).`

Il s'agit donc du schéma de coopération d'interprètes concurrents.

Nous avons choisi un langage fonctionnel, car ce type de langage permet de s'adapter à de multiples contraintes de structuration [Nowak et Le Roch 90]. De plus, il est aisé d'aller vers les concepts d'objets [Masini et al. 90] [Meyer 88], d'acteurs [Hewitt 77] [Agha 86] [Bond and Gasser 88] et d'agents réactifs [Ferber and Carle 90]. Il s'agit du langage Lisp, et plus particulièrement de Méta-Vlisp [Saint-James 87] [Méta-Vlisp 88], développé à l'Université Pierre et Marie Curie entre 1985 et 1988 par l'équipe Méta, dirigée par Emmanuel Saint-James, au laboratoire LITP. Son développement se poursuit actuellement au Centre de Recherche de Bull, notamment sous la forme d'un évaluateur partiel et d'une version en C de Méta-Vlisp traitant les mécanismes d'interfaçage

avec les bibliothèques C et l'allocation-récupération de pages. Ce système est un système ouvert issu d'un contrat avec l'Education Nationale [**Saint-James 87**] largement documenté, et entièrement écrit en Lisp. Méta signifie Machine Endomorphique pour la Transparence Applicative.

Le système que nous proposons est dédié à l'apprentissage des mécanismes de communication et de coordination, basé sur la notion d'acteurs logiciels et matériels, et à l'Intelligence Artificielle Distribuée. Par conséquent, nous avons besoin d'un environnement interactif qui offre la possibilité d'activer et d'arrêter des processus avec une gestion des erreurs, et éventuellement restauration après correction de celles-ci. Un interpréteur semble indispensable.

Nous avons choisi Lisp pour deux aspects :

- comme langage d'assemblage pour implémentation de haut niveau (objets, acteurs, agents, système expert et réseaux sémantiques,
- comme langage de haut niveau pour les utilisateurs (sa syntaxe et sa sémantique sont aisées). Lisp est un langage fonctionnel, partiellement redéfinissable et puissant à cause de la récursivité, un langage impératif étant à écarter.

3.4.1. Caractéristiques

Méta-Vlisp (MVL) est un système complet constitué de son interpréteur et d'un environnement.

L'interpréteur est constitué :

- de l'évaluateur et de sa bibliothèque d'environ 200 fonctions primitives servant de noyau, écrits dans un sous ensemble de Méta-Vlisp, d'où le terme méta,
- du lecteur et de l'imprimeur écrits en Méta-Vlisp.

L'environnement est écrit en Méta-Vlisp; il est constitué des outils d'aide à la mise au point (trace, pas à pas, visualisation de la pile), des outils d'édition (de lignes, de pages), de gestion de l'historique de la session, du "garbage collector". L'évaluateur calcule automatiquement une fermeture chaque fois qu'une lamda-expression est évaluée. Cette caractéristique est très utile pour les constructions orientées objets.

Dans un contexte système donné, interpréteur et environnement sont traduits par un assembleur, lui-même écrit en Méta-Vlisp, fournissant dans une première phase des objets prédéfinis (doublets, symboles, chaînes, textes, tables de continuation) et une séquence d'instructions d'une machine intermédiaire nommée Odin, puis dans une deuxième phase une séquence d'instructions du langage support choisi.

Depuis longtemps, la philosophie de portage des systèmes de la famille Vlisp consiste à traduire finalement en langage d'assemblage le résultat de la première phase de traduction. Ce fut aussi le cas pour Méta-Vlisp (Motorola 680x0, Intel 80x86, Motorola 6809) mais l'apparition des processeurs de type RISC (transputer, sparc, etc.) d'une part, et la large

utilisation de C et Ada d'autre part nous ont conduits depuis deux ans, à une meilleure portabilité de Méta-Vlisp, à son développement en Occam pour transputer, en C pour Motorola 680x0, transputer et sparc, en Ada pour sparc.

3.4.2. Implémentation de Méta-Vlisp sur Transputer

Pour implémenter Méta-Vlisp sur transputer, les possibilités d'Occam, le langage de base du transputer, ont été explorées. La qualité du code est très bonne mais Occam a des capacités limitées (peu de structures de données, pas de récursivité), et de plus, ce langage est spécifique au transputer, et ne peut, par conséquent, pas être aisément utilisé pour des développements ultérieurs avec d'autres processeurs. De plus, les possibilités de mixage de code (assembleur en ligne) n'étaient pas suffisantes pour terminer cette implémentation particulière.

Le langage C n'est pas réellement standard, mais semble être un bon choix. Le système Méta-Vlisp [**Nowak et al. 93**] a donc été implémenté en utilisant le langage C (compilateur "C3L" de la société 3L) sur le transputer. L'efficacité est peu élevée, mais l'implémentation de ce système dans un langage évolué est possible. En utilisant le nouveau compilateur C d'Inmos "ICC" au standard ANSI, le gain en vitesse d'exécution a été de 13%. Une implémentation avec "Helios-C" pour un système T-Node est aussi terminée.

Après une grande tentation de définir tout un système/langage, de manière homogène, en Lisp et après avoir vérifié ces idées [**Méta-Vlisp 88**], nous distinguons maintenant d'un côté les réelles caractéristiques spéciales de Lisp et le besoin d'une redéfinition fonctionnelle dynamique, et de l'autre côté, les aspects purement système. Il semble maintenant que le bénéfice principal de Lisp et particulièrement de Méta-Vlisp reste essentiellement sa fonction "eval" et ses 300 ou 500 fonctions prédéfinies. Ecrit en Lisp, l'environnement Lisp : le lecteur, l'imprimeur, le gestionnaire de file, l'interactivité avec les outils de mise au point (trace, pas à pas, la visualisation de la pile) ne sont pas justifiés dans le contexte d'une implémentation d'un Lisp sur transputers, comme pour un réseau d'acteurs. En effet, de tels acteurs connaissent uniquement les canaux de communication et les canaux traditionnels d'entrée/sortie série ou parallèle. En plus, étant données les fréquences d'émission et de réception de messages dans un tel réseau, ces acteurs doivent profiter d'une gestion non interprétée du flot d'entrée/sortie. La messagerie s'appuiera sur la liaison d'arbres de Méta-Vlisp, particulièrement pour la phase de liaison des arguments aux paramètres formels.

4. Conclusion

Dans un contexte pédagogique, le choix de Méta-Vlisp semble être bon. Nous bénéficions d'une longue expérience et le système est maîtrisé. Par contre, dans un contexte robotique pur où la notion de temps réel est primordial, un système à base de micro-noyau, pour garantir les temps de réponse s'impose.

L'architecture logicielle proposée s'adapte parfaitement à d'autres architectures qu'elles soient complexes (architecture classique) ou plus simples (Brooks). Les approches

neuronales ou génétiques peuvent aisément être mises en oeuvre. La couche système peut être un système classique, un système temps réel, un système réactif ou ne pas exister : c'est l'application qui prendrait alors en compte la gestion de la couche entrée-sortie, comme c'est le cas, actuellement, pour Méta-Vlisp.

L'approche micro-noyau permet un développement modulaire. L'ajout ou la suppression d'une entité (serveur, acteur, périphérique) est aisée.

L'utilisation du mode paquet à taille variable et du transfert asynchrone offre une grande souplesse à l'utilisateur. De futures applications temp réel comme le transport d'images vidéo par réseau radio-fréquence pour un traitement déporté seront facilement adaptables.

ActNet peut être considéré comme un système distribué. Dans un contexte de robotique industrielle, la tolérance aux pannes d'un tel système pourrait être renforcée. Dans le cas d'une panne d'un processeur parmi n ($n \neq 1$), des ressources systèmes pourraient être disponibles sur d'autres sites afin de remédier à cette panne. De plus, une partie de ces mêmes ressources pourraient être disponibles pour augmenter la puissance de calcul d'un robot. Une application déportée pourrait être par exemple le traitement d'images qui nécessite une puissance de calcul importante.

Conclusions et Perspectives

Nous avons décrit dans le premier chapitre le modèle acteur de base d'Hewitt et les modèles dérivés en particulier le modèle de calcul d'Agha et ceux implémentés dans les principaux langages d'acteurs, ABCL/1 en particulier. Ensuite, nous avons défini le modèle acteur matériel. Nous avons donné une définition, et nous avons détaillé son comportement et ses spécificités par rapport aux modèles logiciels. Pour rester cohérent avec le modèle acteur logiciel, les acteurs matériels communiquent par messages de trois types : passé, présent et futur. Comme les entités sont matériellement séparées, un réseau local de communication était indispensable. Le réseau virtuel a été décrit. Ce réseau est qualifié de virtuel dans le sens où aucune distinction n'est faite entre un acteur matériel et un acteur logiciel et que toutes ces entités communiquent par un réseau virtuel. Nous avons décrit le système **ActNet**. Nous avons distingué deux types d'acteurs matériels : les acteurs sédentaires et les acteurs mobiles. A chaque type d'acteurs correspond un sous-réseau. Nous avons donc le sous-réseau d'acteurs sédentaires (SRAS) et le sous-réseau d'acteurs mobiles (SRAM). Un troisième sous-réseau nommé sous-réseau utilisateur (SRU) est chargé de gérer le développement et l'exécution des applications. Les topologies de chaque sous-réseau ont été décrites.

Ce travail s'intègre dans le projet de recherche IBP Marsala (Modélisation, Analyse et Réalisation de Systèmes d'Agents par Langages d'Acteurs) [**Marsala 92**]. Ce projet a pour but de définir des méthodes et de concevoir les outils permettant de spécifier et de réaliser des applications utilisant un modèle conceptuel de type "agent" et mises en oeuvre à l'aide d'un mécanisme d'exécution de type "acteur". ActNet se situe en aval pour l'exécution de ces applications.

Après une brève présentation de l'état de l'art concernant les robots mobiles et la robotique pédagogique, nous avons présenté dans le chapitre II l'architecture matérielle des acteurs matériels. Nous avons développé une carte générique de commande pour les contrôler. Cette carte permet de s'adapter à tout type de robots pédagogiques. Un prototype a été réalisé en technique "wrapping".

Nous avons sélectionné pour la réalisation le transputer. Ce micro-processeur n'autorisant qu'une interruption matérielle, nous avons proposé une amélioration matérielle pour prendre en compte plus d'une interruption.

Pour valider notre concept, nous avons développé deux exemplaires d'un acteur matériel robotique mobile "le Tatou". Nous en avons détaillé l'architecture électronique et mécanique.

Pour mener à bien ce développement, nous avons mis en place une chaîne CAO de développement de projets mécaniques et électroniques. Pour la réalisation des pièces mécaniques, nous avons mis en place une collaboration avec les lycées techniques de Sens (89), Noisiel (77), Meaux (77) et Chelles (77). Pour les cartes électroniques, les circuits imprimés ont été réalisés par un industriel. La matière première en mécanique et les composants électroniques ont été achetés par nos soins.

Un modèle de description du modèle acteur robotique a été proposé, l'Actor_Unit, s'inspirant du Model_Unit et nous avons décrit le comportement du robot mobile "Tatou" lors de son utilisation dans un exemple de tri réparti proposé dans [**Darche and Nowak 93**].

L'étude de la connexion d'un robot pédagogique avec un système de contrôle a soulevé le problème de l'interface. Nous avons proposé d'uniformiser l'interface afin de ne plus être tributaire du matériel. Cette proposition prend la forme d'un module adaptateur, le système NIOS qui permet d'adapter le robot pédagogique au système de commande en utilisant des interfaces standards.

Après avoir présenté les techniques spécifiques au domaine de la radio-fréquence, nous avons détaillé les spécificités techniques de la communication de chaque sous-réseau.

Le sous-réseau d'acteurs mobiles (SRAM) est un réseau radio-fréquence VHF garantissant l'autonomie des mobiles du point de vue des communications [**Darche 91**]. La communication est de type asynchrone, en duplex, à l'alternat ou en simplex grâce à la modularité des éléments communicants. La vitesse de modulation est celle d'une liaison série RS232C équipant la carte de contrôle, et celle-ci varie de 1200 à 19200 bauds. Un codage de canal est nécessaire pour adapter l'information. Pour obtenir une communication duplex, il est nécessaire de différencier l'information provenant de plusieurs interlocuteurs. Une modulation analogique est utilisée. L'espace fréquentiel sera découpé en canaux identiques appelés canaux radio-fréquences. Dans le cas d'une communication à deux voies, deux canaux seront alloués à chaque acteur du SRAM, qui sera par conséquent caractérisé par deux adresses fréquentielles, une pour l'émission et l'autre pour la réception. Pour l'occupation spectrale et la fiabilité, une modulation de fréquence (FM) discrète, une modulation angulaire, la modulation binary minimum frequency shift keying (BMFSK) a été choisie, car elle optimise la largeur de bande nécessaire à la transmission d'un signal digital, sans un codage de source élaboré. La vitesse de modulation maximale demandée et le type de modulation employée définissent une largeur de canal de 50 KHz. L'utilisation de la technologie digitale (les boucles à verrouillage de phase, Phase Lock Loop en particulier) nous permet de programmer les fréquences d'émission et de réception de chaque mobile. En

cas de défaillance des systèmes embarqués de télécommunication ou lors de la phase de développement, les acteurs mobiles pourront communiquer par un canal filaire à collision.

Dans le cas d'utilisation d'un duplexeur d'antenne, une communication en duplex est possible. Mais à cause d'un écart duplex entre les fréquences d'émission et de réception, il existera deux bandes de fréquence distinctes B1 et B2, et donc deux familles d'acteurs, entraînant ainsi des contraintes au niveau de la topologie du sous-réseau.

Dans le cas de deux antennes, il n'existe qu'une seule famille d'acteurs, et une seule bande de fréquence est utilisée. Toutes les topologies sont réalisables en duplex. Avec une configuration à une antenne avec un commutateur d'antenne, une communication à l'alternat ou simplex est réalisable. Les choix techniques effectués nous permettent d'obtenir la meilleure topologie de réseau pour les applications de l'utilisateur (bus, anneau, étoile, etc.).

ActNet se caractérise donc par trois mots-clés :

- Généricité,
- Hétérogénéité,
- Modularité et Flexibilité.

Nous avons proposé une carte générique de commande de robots. Elle permet au concepteur de ne pas se préoccuper des problèmes d'interfaçage, ou tout du moins d'en limiter l'ampleur. Tous les acteurs peuvent donc être équipés d'une même carte de commande. En utilisant des interfaces normalisées, l'architecture proposée est suffisamment générique pour pouvoir s'adapter à la commande de robots plus complexes.

Un acteur matériel peut être équipé de n'importe quelle carte processeur. En effet, la seule unité d'échange entre acteurs est le message. Nous avons utilisé les micro-processeurs transputer T800 et T425 de la société Inmos SGS Thomson, mais d'autres micro-processeurs peuvent être utilisés, comme ceux de la famille Motorola 680X0 ou Intel 80X86. Grâce à la généricité des cartes processeurs, l'adaptation de l'acteur matériel à un type de périphériques robotiques, sédentaires ou mobiles, est aisée. La seconde solution de gestion d'entrée-sortie renforce la modularité puisque la seule chose commune est le langage de commande.

La conception des acteurs matériels a été voulue modulaire afin de faciliter les évolutions. Les cartes processeurs sont interchangeable, puisque les interfaces avec les périphériques sont normalisées. Les entités logicielles sont des acteurs et peuvent donc être déplacées ou changées aisément. Les modules communicateurs sont aussi interchangeables. L'utilisateur peut donc choisir, en fonction de son application particulière, la meilleure architecture matérielle et la topologie du réseau de communication la mieux adaptée.

L'architecture décrite présente des performances élevées au niveau de la puissance de calcul, de la capacité mémoire, ou de l'autonomie énergétique, ceci grâce aux choix techniques effectués.

Ces caractéristiques matérielles et logicielles répondent donc aux besoins de laboratoires de recherche spécialisés en Intelligence Artificielle Distribuée (IAD) et offrent ainsi une plateforme de développement et d'exécution pour leurs applications spécifiques. Lisp avec Méta-Vlisp est le langage de développement par excellence pour des styles de programmation récents (par objets, acteurs, agents, règles de la logique). Ainsi, il permet de développer aisément des applications à base d'agents de type réactif en particulier.

Evolutions futures

Nous voulons valider la deuxième proposition de gestion d'entrée-sortie en développant une carte à micro-contrôleur Motorola 68HC11 chargée de gérer les entrées-sorties des acteurs robotiques et qui communiquerait par notre langage standard. La proposition d'architecture est adaptative. Avec la gestion des entrées-sorties par un processeur spécialisé qui dialogue par un langage défini, aucune carte n'est dédiée matériellement. Il sera possible de pouvoir tester de nouveaux processeurs comme le T9000 avec ses communications à 100 Mbbauds de la société Inmos, le processeur de traitement du signal TMS320C40 de la société Texas Instruments ou le PowerPC de la société Motorola, ceci avec un coût minimum pour l'adaptation, montrant ainsi la modularité d'ActNet.

Dans ce cadre, nous adapterons sur d'autres prototypes mobiles et sédentaires des cartes systèmes au standard PC/AT à base de microprocesseur 80486 ou compatible (Cyrix ou AMD). Nous pensons en particulier utiliser des cartes électroniques miniatures au standard PC104, standard en cours de normalisation par l'IEEE. Cette adaptation permettrait d'obtenir un environnement de développement peu onéreux, standard et performant basé sur une architecture répandue.

Actuellement, l'acteur robotique mobile ne possède que des capteurs de collision. Le nombre de capteurs doit être augmenté pour pouvoir appréhender de manière beaucoup plus fine l'environnement. Nous voulons développer de nouveaux capteurs, en particulier un télémètre à ultra-son et un capteur de détection infra-rouge pour la détection de personnes, et ajouter une chaîne d'acquisition/reproduction sonore.

A long terme, nous voulons adapter deux caméras CCD pour pouvoir appréhender l'environnement en vision stéréoscopique. La vision nécessite des traitements temps réels. La présence de celle-ci permettrait d'augmenter la charge système et ainsi de valider les temps de réponse du système.

L'utilisation de boucle à verrouillage de phase permet aux équipements communicants de s'adapter à une communication de type spectre élargi à saut de fréquence. La prochaine étape de notre travail serait de mettre en oeuvre cette technique de transmission. Par la suite, il serait intéressant de s'intéresser aux ondes infra-rouges pour développer des réseaux à faible coût, donc très bien adaptés aux applications pédagogiques. Les diodes électro-luminescentes infra-rouges se prêtent bien à la modulation de séquence directe à étalement de spectre à spectre élargi. Il serait aussi intéressant d'intégrer les dernières technologies comme la synthèse numérique directe (DDS) ou la modulation de phase à n états qui offrent des perspectives techniques intéressantes (performances, intégration, etc.).

Du point de vue logiciel, nous voulons valider l'architecture logicielle proposée dans le chapitre IV. Nous développerons une couche comportementale de type "Brooks" pour donner une dimension réactive à l'acteur matériel. Nous étudierons ensuite et développerons la couche système basée sur la notion de micro-noyau avec, en particulier, la gestion des communications. Ainsi les acteurs matériels offriront un plus, c'est-à-dire la communication, par rapport aux agents réactifs de Brooks.

A long terme, nous prendrons en compte la notion de temps réel dans notre conception. Nous pensons établir des collaborations afin de développer un système distribué temps réel reposant sur un réseau non câblé (wireless network). Cette approche serait appliquée à la robotique. Un système distribué d'acteurs robotiques permet de rendre le système tolérant aux pannes. Toutes les ressources processeurs et mémoire de masse seraient mises en commun et dans le cas d'une panne d'une carte processeur parmi plusieurs, ou d'un périphérique non vital au fonctionnement du robot, celui-ci pourrait disposer de ressources processeurs non utilisées dans le réseau.

Nous n'abandonons pas pour autant la robotique pédagogique. Nous voulons continuer la collaboration initialisée pendant cette thèse concernant la partie pédagogique en implémentant l'interpréteur Méta-VLisp comme une application fonctionnant au-dessus du système. Puis en utilisant celui-ci dans une expérience pédagogique dans une école en relation avec des pédagogues. Le concept d'acteur est un concept pédagogique très puissant. L'expérience consisterait à développer la métaphore du théâtre. Suite à la consigne du professeur, comme par exemple celle de [Darche and Nowak 89 90] donnée pour un tri réparti, l'élève construit un scénario et joue la solution. Si les primitives sont suffisamment de haut niveau, l'élève pourra ensuite programmer l'acteur et lui faire "jouer" sa solution. Notre projet représente un pas qualitatif dans le domaine des micro-mondes dédiés à l'apprentissage, en comparaison avec le classique micro-monde 'tortue Logo'. La notion de micro-monde associée à la notion d'acteurs logiciels et matériels (acteur robotique entre autres) communiquant par messages sont des concepts forts, qui, implémentés dans un système de traitement de données, le convertissent en un puissant outil pédagogique d'apprentissage des mécanismes liés au parallélisme et devraient répondre au besoin en formation de certaines entreprises ou d'institutions.

Ce projet est donc innovant car il répond ou répondra à des préoccupations actuelles : acteur, communication mobile, robotique et IAD, système distribué, et temps réel.

Annexes

Schémas électroniques

- 1. Carte unité centrale**
- 2. Carte mémoire morte**
- 3. Carte entrées-sorties**
- 4. Contrôleur moteur pas à pas**
- 5. Détecteur de collision**
- 6. Contrôleur module stylo**
- 7. Détecteur de fins de course stylo**
- 8. Sélecteur d'alimentation**
- 9. Chargeur de batterie**
- 10. Convertisseur DC/DC**
- 11. Divers**

Références bibliographiques

[Agha 86]

Gul A. Agha : "Actors: A model of concurrent computation in distributed systems". The MIT Press, Cambridge Massachusetts, 1986. ISBN 0-262-01092-5.

[Amoroso 80]

Frank Amoroso : "The bandwidth of digital data signals". IEEE Communications Society Magazine, Vol. 18, N°6, pp. 13-24. November 1980.

[Angle 89]

Colin M. Angle : "Genghis, A Six Legged Autonomous Walking Robot". Thesis in Electrical Engineering and Computer Science. MIT, Cambridge Massachusetts (USA). March 1989.

[Araï et al. 93]

Tamio Araï, Eiichi Yoshida and Jun Ota : "Information Diffusion by Local Communication of Multiple Mobile Robots". Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Systems Engineering in the Service of Humans, Vol. 4, pp. 535-540. October 17-20 1993, Le Touquet (France).

[Bach 86]

Maurice J. Bach : "The Design of the UNIX Operating System". Prentice Hall, 1986. Paru en français : "Conception du système Unix". Masson 1989. ISBN 2-225-81596-8.

[Bachatène 92a]

Haféda Bachatène : "Modeling Units : A High Level Formalism to Specify Complex Distributed Systems". Proceedings of the 7th International Symposium on Computer and Information Sciences (ISCIS VII), pp. 495-498. Erol Gelenbe, Ugur Halici and Nese Yalabik editors. November 1992, Antalya (Turkey). Presses de l'EHEI, Paris (France).

[Bachatène 92b]

Haféda Bachatène : "A Modular Design Based on Stepwise Refinements of Coloured Petri Nets". Proceedings of the Second Annual Software Engineering Research Forum. November 5-6 1992, Melbourne, Florida (USA).

[Barnes et al. 91]

D.P. Barnes, C. G. Downes and J. O. Gray : "The Real Time Control of a Hexapodal Robot Using Multiple Transputers". Applications of Transputers 3, Proceedings of the 3rd International Conference on Applications of Transputers, Vol. 2, pp. 400-405. Edited by Tariq S. Durrani, William A. Sandham, John J. Soraghan and Sheila M. Forbes. IOS Press. 28-30 August 1991, Glasgow (United-Kingdom). ISBN 90-5199-064-2.

[Baudry et Schoefs 89]

Marc Baudry et Yves Schoefs : "S.P.F. : un environnement d'apprentissage des concepts liés à la flexibilité dans la production" Actes du 1er congrès francophone de robotique pédagogique, pp. 155-165. Août 1989, Le Mans (France).

[Bond and Gasser 88]

Alan H. Bond and Les Gasser : "Distributed Artificial Intelligence, Communication, Coordination". Morgan-Kaufman, Lecture notes 1988.

[Bougé 87]

Luc Bougé : "Repeated Snapshots in Distributed Systems with Asynchronous Communications and Their Implementation in CSP". Theoretical Computer Science 49, pp. 145-169. 1987.

[Briot 89]

Jean-Pierre Briot : "Actalk : a testbed for Classifying and Designing Actor Languages in the Smalltalk-80 Environment". Proceedings of the European Conference on Object-Oriented Programming (ECOOP'89) edited by S. Cook, British Computer Society Workshop Series, Cambridge University Press, pp. 109-129. July 1989, (United-Kingdom).

[Brooks 85]

Rodney A. Brooks : "A Robust Layered Intelligent Control System for a Mobile Robot". AI Memo N° 864, September 1985. Artificial Intelligence Laboratory, Massachusetts Institute of Technology (USA).

[Brooks 86a]

Rodney A. Brooks : "A Layered Intelligent Control System for a Mobile Robot". IEEE Journal of Robotics and Automation, RA-2, pp. 14-23. April 1986.

[Brooks 86b]

Rodney A. Brooks : "Achieving Artificial Intelligence through Building Robots". AI Memo N° 899. May 1986. Massachusetts Institute of Technology, Artificial Intelligence Laboratory (USA).

[Brooks et al. 88]

Rodney A. Brooks, Jonathan Hudson Connell and Peter Ning : "Herbert : A Second Generation Mobile Robot". AI Memo N° 1016. January 1988. Massachusetts Institute of Technology, Artificial Intelligence Laboratory (USA).

[Brooks 89]

Rodney A. Brooks : "A Robot that Walks; Emergent Behaviors from a Carefully Evolved Network". AI Memo N° 1091. February 1989. Massachusetts Institute of Technology, Artificial Intelligence Laboratory (USA).

[Brooks 90]

Rodney A. Brooks : "The Behavior Language; User's Guide". AI Memo N° 1227. April 1990. Massachusetts Institute of Technology, Artificial Intelligence Laboratory (USA).

[Brooks 92]

Rodney A. Brooks : "Artificial Life and Real Robots". Proceedings of the First European Conference on Artificial Life, pp. 3-10. December 1991, Paris France. Edited by Francisco J. Varela and Paul Bourguine. The MIT Press 1992. ISBN 0-262-72019-1.

[Card 93]

Rémy Card : "MASIX : un Système d'Exploitation Multi-Environnements utilisant le Micro-Noyau MACH". Thèse de doctorat de l'Université Pierre et Marie Curie (Paris VI - France). Mai 1993.

[Carle 92]

Patrice Carle : "Mering IV : un langage d'acteurs pour l'intelligence artificielle distribuée intégrant objets et agents par réflexivité compilatoire". Thèse de doctorat de l'Université Pierre et Marie Curie (Paris VI - France). Avril 1992.

[Cheriton 84]

David Cheriton : "The V Kernel : A Software Base for Distributed Systems". IEEE Software, pp. 19-42. 1984.

[Coiffet 86]

Philippe Coiffet : "La Robotique, principes et applications". Traité des Nouvelles Technologies, série Robotique. Edition Hermès 1986. ISBN 2-86601-076-0.

[Coiffet 93]

Philippe Coiffet : "Robot habilis, robot sapiens - histoire, développements et futurs de la robotique". Collection perspectives. Edition Hermès 1993. ISBN 2-86601-381-6.

[Connell 89]

Jonathan Hudson Connell : "A Colony Architecture for an Artificial Creature". Technical Report N° 1151. August 1989. Massachusetts Institute of Technology Artificial Intelligence Laboratory.

[Courtrai et al. 92]

Luc Courtrai, Jean-François Roos, Jean-Marc Geib and Jean-François Mehaut : "The Implementation of an Actor Environment on Transputers under Helios, The Communicating Active Components". Poster in Transputers'92 (Advanced Research and Industrial Applications), page 351. 1992, Arc-et-Senans France. IOS Press. ISBN 90-5199-081-2. Also in Publication ERA N° 114. Mai 1992, Laboratoire d'Informatique Fondamentale de Lille (L.I.F.L.).

[Cox and Gehani 89]

Ingemar J. Cox and Narain H. Gehani : "Concurrent Programming and Robotics". The International Journal of Robotics Research, Vol . 8, N°2, pp. 3-16. April 1989.

[Darche 90]

Philippe Darche : "Conception et réalisation d'un environnement de contrôle d'acteurs robotiques". Rapport de D.E.A. Systèmes Informatiques. Septembre 1990, Université Pierre et Marie Curie (Paris VI - France).

[Darche 91]

Philippe Darche : "Red de Actores Robóticos". Robótica Pedagógica Memoria de la Tercera Conferencia internacional. Enrique Ruiz Velasco Sánchez Coordinador, pp. 71-89, Mexico (Mexique), Août 1991. ISBN 968-36-2759-5.

[Darche and Nowak 92]

Philippe Darche and Gérard Nowak. "ActNet : A Network of Actors based on Transputers for Apprenticeship of Parallelism". Proceedings of Transputers'92 (Advanced Research and Industrial Applications), pp. 293-306. 1992, Arc-et-Senans (France). IOS Press. ISBN-90-5199-081-2.

[Darche and Nowak 93]

Philippe Darche and Gérard Nowak : "ActNet : A Heterogeneous Network of Actors for Learning of Parallelism, Communication and Synchronization". In Control Technology in Elementary Education, NATO ASI Series F : Computer and Systems Sciences, Vol. 116, pp. 289-307. Edited by Brigitte Denis. Springer Verlag Berlin (Germany) 1993. ISBN 3-540-56710-0 and 0-387-56710-0. Paru aussi en rapport de recherche MASI N° 93-22. Université Pierre et Marie Curie (Paris VI - France).

[Darche 93]

Philippe Darche : "L'acteur robotique : étude et proposition d'interface générique de contrôle". Regards sur la robotique pédagogique, Actes du quatrième colloque international sur la robotique pédagogique, pp. 232-248. 5-8 Juillet 1993, Liège (Belgique). Technologies nouvelles et éducation. Institut National de Recherche Pédagogique (INRP). ISBN 2-7342-0391-X.

[Dean and Armand 92]

Randall W. Dean and François Armand : "Data Movement in Kernelized System". Proceedings of the workshop on Micro-Kernels and Other Kernel Architectures, pp. 243-261. April 1992.

[De Bellescize 32]

De Bellescize : "La réception synchrone". L'onde électrique. 1932.

[Demazeau and Müller 89]

Yves Demazeau and Jean-Pierre Müller : "Decentralized Artificial Intelligence". Proceedings of the first European Workshop on Modelling Autonomous Agents in a Multi-agent World, pp. 5-13. August 16-18 1989, Cambridge England. Edited by Yves Demazeau and Jean-Pierre Müller. Elsevier Science Publishers 1990. ISBN 0-444-88705-9.

[Douglass et al. 91]

Fred Douglass, M. Frans Kaashoek, Andrew S. Tannebaum, and John K. Ousterhout : "A Comparison of Two Distributed Systems : Amoeba and Sprite". Computing Systems, Vol. 4, N° 3, pp. 353-384. December 1991.

[Etzioni 93]

Oren Etzioni : "Intelligence without Robots: A Reply to Brooks". AI Magazine Vol. 14, N° 4, pp. 7-13. Winter 1993.

[Everett and Gilbreath 88]

CDR H.R. Everett, Gary Gilbreath : "A Supervised Autonomous Security Robot". Robotics and autonomous Systems, Vol. 4, N° 1, pp. 209-232. March 1988.

[Ferber 89a]

Jacques Ferber : "L'ECO résolution ou comment résoudre un problème par interactions". Colloque Franco-Espagnol de la Méta-connaissance. Septembre 1988. Barcelona (Espagne). Cahier du Laforia, Avril 1989. Paru en anglais, rapport Laforia N° 5/90 (Janvier).

[Ferber 89b]

Jacques Ferber : "Objets et Agents : une Etude des structures de Représentation et de Communication en Intelligence Artificielle". Thèse d'état. Juin 1989, Université Pierre et Marie Curie (Paris VI - France).

[Ferber 90]

Jacques Ferber : "The Framework of ECO-Problem Solving". Proceedings of the 2nd European Workshop MAAMAW'90, pp. 103-114. August 1990, Saint-Quentin-en-Yvelines (France).

[Ferber and Carle 90]

Jacques Ferber and Patrice Carle : "Actors and Agents as Reflexive Concurrent Objects: a Mering IV perspective". IBP report N° 18/90. Juin 1990, Université Pierre et Marie Curie.

[Flynn 87]

Anita M. Flynn : "Gnat Robots (And How They Will Change Robotics)". IEEE Micro Robots and Teleoperators Workshop. November 1987. Hyannis, Massachusetts (USA).

[Flynn et al. 89]

Anita M. Flynn, Rodney A. Brooks and William M. Wells III : "Squirt : The Prototypical Mobile Robot for Autonomous Graduate Students". AI Memo N° 1120. July 1989. Massachusetts Institute of Technology, Artificial Intelligence Laboratory (USA).

[Freedman 91]

Paul Freedman : "Time, Petri Nets, and Robotics". IEEE Transactions on Robotics and Automation. Vol. 7, N° 4, pp. 417-433. August 91.

[Giralt et al. 79]

Georges Giralt, Ralph Sobek and Raja Chatila : "A Multi-level Planning and Navigation System for a Mobile Robot; a first approach to Hilare". Proceedings of the 6th International Joint Conference on Artificial Intelligence (IJCAI), pp. 335-337. August 20-23 1979, Tokyo (Japan).

[Greussay 80]

Patrick Greussay : "Les outils de communication" dans Jean-Claude Simon : "L'éducation et l'informatisation de la société, rapport au Président de la République". Fayard 1981. ISBN 2-213-01008-0. Annexe 1 : les voies de développement, Contributions des groupes de travail (groupe apprentissage autonome). La Documentation Française, pp. 221-226. 1980. ISBN 2-11-000-623-4.

[Hewitt et al. 73]

Carl Hewitt, Peter Bishop and Richard Steiger : "A Universal Modular Actor Formalism for Artificial Intelligence". Proceedings of the 3rd International Joint Conference on Artificial Intelligence (IJCAI), pp. 235-245. August 20-23 1973. Stanford University, Stanford California (USA).

[Hewitt 77]

Carl Hewitt : "Viewing Control Structures as Patterns of Passing Messages". Artificial Intelligence, Vol. 8, N° 3, pp. 323-364. June 1977.

[Hirose 82]

Shigeo Hirose : "Mobile Robot Technology". J. Soc. Instrum. Control Eng., 21(12), pp. 1129-1134. 1982.

[Hirose and Morishima 90]

Shigeo Hirose and Akio Morishima : "Design and Control of a Mobile Robot with an Articulated Body". The International Journal of Robotics Research, Vol. 9, N°2, pp. 99-114. April 1990.

[Hoare 87]

C.A.R. Hoare : "Processus Séquentiels Communicants". MASSON Paris. ISBN 2-225-81100-8.

[Inmos 89]

INMOS Corporation : "The Transputer Data Book.". Ref. 72 TRN 203 01. 1989.

[Inmos 91]

INMOS Corporation : "The T9000 Transputer Products Overview Manual". Ref. 72 TRN 228 00. 1991.

[J.D. Productique 86]

Société J.D. Productique : "Documentation du robot 'Youpi'". 1986.

[LAAS 87]

Le séminaire du LAAS-CNRS : "Les Robots Mobiles Autonomes". pp. 167-196. Derniers développements en Automatique Informatique Robotique Micro-électronique. 1987. ISBN 2-85428-101-0.

[Leffler et al. 89]

Samuel J. Leffler, Marshall Kirk McKusick, Michael J. Karels, and John S. Quaterman : "The Design and Implementation of the 4.3 BSD UNIX Operating System". Addison Wesley, 1989. ISBN 0-201-06196-1.

[Le Roch et Nowak 85]

Marie-France Le Roch et Gérard Nowak : "En amont de Logo. Apprentissages au Cours Moyen : Communication, Coordination". Compte rendu d'expérimentation. Paris 1985, Laboratoire MASI, Université Pierre et Marie Curie.

[Le Roch et Nowak 90]

Marie-France Le Roch et Gérard Nowak : "Synchronisation de robots avec ADA". Compte rendu de projet DESS ISI. 1990, Laboratoire MASI, Université Pierre et Marie Curie.

[Le Roch et Nowak 90]

Marie-France Le Roch et Gérard Nowak : "Moteurs! Acteurs!". Actes du Deuxième Congrès International sur la Robotique Pédagogique, pp. 165-180. Les Publications de la Faculté des Sciences de l'Éducation, collection Actes de Colloque. Édités par Pierre Nonnon et Martial Vivet. Août 1990, Montréal (Canada). ISBN 2-020208-70-4.

[Le Roch and Nowak 91]

Marie-France Le Roch and Gérard Nowak : "Robotic Communicating Actors". proceedings of the 2nd France-Israel Symposium on Robotics - Advances in Non-Manufacturing Robotics, session 6, pp. 6.1-6.14. 8-10 April 1991, Saclay (France).

[Masini et al. 90]

Gérald Masini, Amedeo Napoli, Dominique Colnet, Daniel Léonard et Karl Tombre : "Les langages à objets." Inter Editions, 1990. ISBN 2-7296-0275-5.

[Marsala 92]

"Rapport d'activité et travaux MASI dans le cadre du projet MARSALA". Rapport interne. Laboratoire MASI, Université Pierre et Marie Curie (Paris VI - France). Décembre 1992.

[Méta-Vlisp 88]

"Méta-Vlisp. Manuel d'utilisation de l'interprète". LITP, Université Pierre et Marie Curie. 1988.

[Meyer 88]

Bertrand Meyer : "Object-Oriented Software Construction." - C.A.R. Hoare Series Editor, Prentice Hall 1988.

[Meystel 91]

A. Meystel : "Autonomous Mobile Robots, Vehicles with Cognitive Control". World Scientific Series in Automation, Vol. 1. World Scientific 1991. ISBN 9971-50-088-4.

[Minsky 88]

Marvin Minsky : "La Société de l'Esprit". InterEditions, Paris, 1988. ISBN 0-7296-0188-0. Publié en langue anglaise par Simon and Schuster, New-York, 1985, sous le titre "The Society of Mind".

[Mondada and Verschure 93]

Francesco Mondada and Paul F.M.J. Verschure : "Modeling System-environment interaction : The complementary roles of simulations and real world artifacts". Proceedings of European Conference on Artificial Life. Brussels (Belgique).

[Mouaddib et al. 93]

Abdel-Allah Mouaddib, François Charpillet and Jean-Paul Haton : "Real-time Message Engine for a Multi-agent Architecture". Proceedings of the Thirteenth International Conference on Artificial Intelligence, Expert Systems and Natural Language, Vol. 1, pp. 589-599. May 24-28, Avignon (France).

[Mullender et al. 90]

Sape Mullender, G. van Rossum, A. Tanenbaum, R. van Reneese, and H. van Straveren. "Amoeba : A Distributed Operating System for the 1990s". Communications of the ACM, Vol. 15, N° 5, pp. 44-53. May 1990.

[Mullender 93]

Sape Mullender : "Distributed Systems". Edited by Sape Mullender. ACM Press, Frontier Series, Addison Wesley . New York (USA). 1993. ISBN 0-201-62427-3.

[Narasimhan et al. 88]

Sundar Narasimhan, David M. Siegel, and John M. Hollerbach : "A Standard Architecture for Controlling Robots". AI Memo N° 977. Massachusetts Institute of Technology, Artificial Intelligence Laboratory. July 1988.

[Nehmzow et al. 89]

Ulrich Nehmzow, John Hallam and Tim Smithers : "Really Useful Robots". Intelligent Autonomous Systems 2, pp. 284-293. Edited By T. Kanade, F.C.A. Groen and L.O. Hertzberger, Amsterdam (Hollande). 1989. ISBN 90-800410-1-7.

[Noreils 93]

Fabrice R. Noreils : "Toward a Robot Architecture Integrating Cooperation between Mobile Robots : Application to Indoor Environment". The International Journal of Robotics Research, Vol . 12, N°1, pp. 79-98. February 1993.

[Nowak et Le Roch 90]

Gérard Nowak et Marie-France Le Roch : "Méta-Vlisp, langue maternelle pour acteurs "robotiques" à base de transputers". Premières Journées Francophones sur les Langages Applicatifs (J.F.L.A.), pp. 159-169. Janvier 1990, La Rochelle (France).

[Nowak et al. 93]

Gérard Nowak, Marie-France Le Roch et Philippe Darche : "Méta-Vlisp sur transputer, application à un réseau d'acteurs". Rapport interne. Université Pierre et Marie Curie (Paris VI - France). Juin 1993.

[Papert 80]

Seymour Papert : "Mindstorms, Children, Computers and Powerful Ideas". Basic Books inc. Publishers New-York (USA), 1980.

[Papert 87]

Seymour Papert : "Microworlds : Transforming Education in Artificial Intelligence and Education". R.W. Lawler and M. Yazdani eds, Vol. 1. 1987.

[Pelat 86]

Alain Pelat : "Modulation d'amplitude et modulation angulaire". Collection technologies, série électronique. Masson 1986. ISBN 2-225-80714-0.

[Premvuti and Yuta 93]

Suparek Premvuti and Shin'ichi Yuta : "An Experiment in Realizing Autonomous and Cooperative Behaviors between Multiple Mobile Robots". IEEE International Workshop on Emerging Technologies and Factory Automation (Technology for the Intelligent Factory), pp. 301-304, R. Zurawski and T.S. Dillon Editors. August 11-14 1994. Melbourne (Australia). CRL Publishing Ltd, London (United Kingdom).

[Rémy et al. 88]

Jean Gabriel Rémy, Jean Cueugnet et Cedric Siben : "Systèmes de radiocommunications avec les mobiles". Collection technique et scientifique des télécommunications. Eyrolles 1988.

[Rozier et al. 88]

Marc Rozier, Vadim Abrossimov, François Armand, Ivan Boule, Michel Gien, Marc Guillemont, Frédéric Herrmann, Claude Kaiser, Sylvain Langlois, Pierre Léonard, and Will Neuhauser : "CHORUS Distributed Operating Systems". Computing Systems, Vol. 1, N° 4, pp. 305-370. December 1988.

[Saint-James 86]

Emmanuel Saint-James : "Lisp comme sa propre machine virtuelle". Bigre + Globule N° 50, pp. 134 -149. 1986.

[Saint-James 87]

Emmanuel Saint-James : "De la Méta-Récurtivité comme outil d'implémentation". Thèse d'Etat. Décembre 1987, Université Pierre et Marie Curie.

[Saint-James 91]

Emmanuel Saint-James : "Un assembleur pour une Machine Endomorphique". Rapport de recherche N° RT/91009. Mars 1991, Bull Company.

[Shoham 93]

Yoav Shoham : "Agent-oriented programming". Artificial Intelligence N° 60, pp. 51-92. 1993.

[Smith and Hewitt 75]

B.C. Smith and Carl Hewitt : "A Plasma Primer". Draft, AI Lab. September 1975, MIT, Cambridge MA, USA .

[Tanguy 87]

Roger Tanguy : "Un réseau de mobiles autonomes pour l'apprentissage de la communication". Thèse de doctorat de l'Université Pierre et Marie Curie (Paris VI - France). Décembre 1987.

[Tevanian and Rashid 87]

Avadis Tevanian and Richard Rashid : "Mach: A basis for future UNIX Development". Technical Report CMU-CS-87-139. 1987. Scholl of Computer Science, Carnegie Mellon University (USA).

[Verissimo and Rodrigues 93]

Paulo Verissimo and Luis Rodrigues : "Group Orientation : a Paradigm for Modern Distributed Systems". Extended and revised version of a paper appeared in Proceedings of the ACM SIGOPS 1992 Workshop, Mont Saint-Michel (France). Esprit Basic Research Project 6360, First Year Report. Vol. 1 of 3, October 1993.

[Vivet 90]

Martial Vivet : "Robotique Pédagogique! Soit, mais pour enseigner quoi ?". Deuxième colloque de Robotique Pédagogique, pp. 7-16. 1990, Montréal (Canada).

[Vivet et al. 91]

Martial Vivet, Jérôme Bruneau and Christophe Parmentier : "Learning with Micro-Robotics Activities". Integrating Advanced Technology into Technology Education, NATO ASI Series F : Computer and Systems Sciences, Vol. 78, pp. 139-147. Edited by Michael Hacker et al. Springer Verlag. Berlin (Germany) 1991. ISBN 3-540-54275-2/0-387-54275-2.

[Yakoh and Anzai 93]

Takahiro Yakoh and Yuichiro Anzai : "Physical Ownership and Task Reallocation for Multiple Robots with Heterogeneous Goals". Proceedings of International Conference on Intelligent and Cooperative Information Systems, pp. 80-87. Rotterdam (The Netherlands). May 12-14 1993.

[Yonezawa et al. 90]

"Object-Oriented Concurrent Programming". Edited by Akinori Yonezawa and Mario Tokoro. The MIT Press, Cambridge Massachusetts, 1990. ISBN 0-262-24026-2.

[Yonezawa 90]

"ABCL : An Object-Oriented Concurrent System". Edited by Akinori Yonezawa. The MIT Press, Cambridge Massachusetts, 1990. ISBN 0-262-24029-7.

GLOSSAIRE

AAL	:	ATM Adaptation Layer
AL	:	Acteur Logiciel
AM	:	Acteurs Matériels
AM	:	Amplitude Modulation - modulation d'amplitude
ActNet	:	Actors' Network
AMB	:	Acteur Matériel de Base
AMGC	:	Acteur Matériel Gestionnaire de Communication
AMRC	:	Accès Multiple à Répartition dans les Codes
AMRF	:	Accès Multiple à Répartition de Fréquence
AMRT	:	Accès Multiple à Répartition dans le Temps
AN	:	Application Number
A/N	:	Analogique/Numérique
ARM	:	Acteur Robotique Mobile
ARS	:	Acteur Robotique Sédentaire
ASK	:	Amplitude Shift Keying - Modulation d'Amplitude Discrète (MAD)
ATM	:	Asynchronous Transfer Mode
BLU	:	Bande Latérale Unique (SSB = Single Side Band)
BMFSK	:	Binary Minimum Frequency Shift Keying
BPSK	:	Binary Phase Shift Keying
CAC	:	Communicating Active Component - Composant Actif Communicant
CCD	:	Coupled Charge Device
CDMA	:	Code-Division Multiple Access
CDPD	:	Cellular Digital Packet Data
CEBus	:	Consumer Electronics Bus
CP	:	Continue Phase
CPFSK	:	Continuous Phase Frequency Shift Keying
CSMA/CA	:	Carrier Sense Multiple Access / Collision Avoidance
CSMA/CD	:	Carrier Sense Multiple Access with Collision Detect. Accès Multiple à Détection de Porteuse avec Détection de Collision
DPSK	:	Differential Phase Shift Keying
DC	:	Direct Conversion - Conversion Directe
DECT	:	Digital European Cordless Telecommunication
DDS	:	Direct Digital Synthesis
EIA	:	Electronic Industry Association
E/S	:	Entrée-Sortie
FCC	:	Federal Communications Commission
FDMA	:	Frequency-Division Multiple Access
FI	:	Fréquence Intermédiaire - Intermédiaire Frequency (IF)
FIFO	:	First In, First Out, premier entré, premier sorti
FM	:	Frequency Modulation - Modulation de Fréquence (MF)
FFSK	:	Fast Frequency Shift Keying - (MDFR) Modulation par Déplacement de Fréquence Rapide, autre dénomination : MSK
FSK	:	Frequency Shift Keying - (MDF) Modulation par Déplacement de Fréquence
GAL	:	Generic Array Logic - Logique à zone générique
GMSK	:	Gaussian Minimum Shift Keying
GSFSK	:	Gaussian-shaped Frequency Shift Keying
HRT	:	Hard Real-Time

HF	:	High Frequency - Haute Fréquence
IAD	:	Intelligence Artificielle Distribuée
LED	:	Light Emitting Diode
LIFO	:	Last In, First Out, dernier entré, premier sorti
LC	:	Self Condensateur
LCD	:	Liquid Crystal Display
LLC	:	Logical Link Control
LO	:	Local Oscillator - oscillateur local
MAC	:	Medium Access Control
MF	:	Moyenne Fréquence
MFSK	:	Minimum Frequency Shift Keying - Modulation par Déplacement de Fréquence Minimum (MDFM), autre dénomination : FFSK
N/A	:	Numérique/Analogique
NRZ	:	Non Retour à Zéro
OOK	:	On-Off Keying - modulation tout ou rien
PAL	:	Programmable Array Logic - logique à zone programmable
PLL	:	Phase Locked Loop - boucle à verrouillage de phase ou boucle d'asservissement de phase
PM	:	Phase Modulation - modulation de phase
PPN	:	Parallel Port Number
PRN	:	Pseudo-Random Noise
PSK	:	Phase Shift Keying - Modulation par Déplacement de Phase (MDP)
QAM	:	Quadrature Amplitude Modulation
QPSK	:	Quaternary Phase Shift Keying
RAM	:	Random Access Memory - mémoire à accès aléatoire (mémoire vive)
RAR	:	Réseau d'Acteurs Robotiques
RF	:	Radio-Fréquence
RMA	:	Robot Mobile Autonome
RN	:	Register Number
ROM	:	Read Only Memory - mémoire à lecture seule (mémoire morte)
SAEN	:	Simple Analog Event Number
SDEN	:	Simple Digital Event Number
SFSK	:	Sinusoidal Frequency Shift Keying
SPN	:	Serial Port Number
SRAM	:	Sous-Réseau d'Acteurs Mobiles
SRAS	:	Sous-Réseau d'Acteurs Sédentaires
SRE	:	Sous-Réseau Exécutif
SRT	:	Soft Real-Time
SRU	:	Sous-Réseau Utilisateur
SSC	:	Spread-Spectrum Carrier
SSGA	:	Sous-Système Gestion et Acheminement des messages
SSR	:	Sous-Système Radio
STR	:	Système Temps Réel
TDD	:	Time Division Duplex
TDMA	:	Time Division Multiple Access
VCO	:	Voltage Controlled Oscillator - Oscillateur Contrôlé en Tension (OCT)
VHF	:	Very High Frequency - Très Haute Fréquence (THF)
WLAN	:	Wireless Local Network